Reliable wireless data transfer for smart sensor shorts in football

MSc Thesis

Zakaria Abdellaoui

Delft University of Technology

Reliable wireless data transfer for smart sensor shorts in football

by

Zakaria Abdellaoui

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Monday August 29, 2022 at 13:30.

Thesis committee:	Dr.ir. A. Bossche Prof.dr. P.J. French Dr. ir. Arjan van Genderen Ing. J. Bastemeijer MSc A.S.M. Steijlen
Faculty:	Faculty of Electrical Engineering, Mathematics, and Computer Science, Delft

TU Delft

Abstract

In football, most of the injuries occur in the lower extremities of the athletes. The leading cause of this is high muscle stress during explosive actions. To be able to prevent injuries, the Dutch Football Association (KNVB) and Dutch Hockey Association (KNHB) are working on smart sensor shorts in collaboration with the Delft University of Technology. The human movement scientists from the VU and RUG use these sensor shorts to develop a model which can predict potential incidents. The product's goal is for the physical trainer to use it to monitor the team in real-time during a game or training. The physical trainer will use a computer to analyze all the data. The sensor shorts consist of 3 Inertial Measurement Units (IMUs). Each IMU measures the angular velocity (dps), linear acceleration (g), and magnetic field (µT) with a sampling rate of 250 Hz. The high volume of data due to the high sampling rate induces challenges for wireless communication and design.

The sensor shorts can, at the moment, read out the IMUs and compress the sensor data. Besides, possible antennas for the sensor shorts have already been looked at. This thesis will continue with the communication link between the sensor shorts and the computer used by the physical trainer. The thesis will be focused on designing a reliable system for wireless communication. Multiple challenges must be faced to develop a reliable wireless communication system with minimal size. The challenges are the minimum amount of resources available, high data rate, large area to cover for communication, and obstruction due to the human body.

According to literature about monitoring devices currently described in the literature used for football, eight base stations are needed around the football field. The goal is to minimize the number of base stations so that the system can be easily set up.

For the development of the system, first, the different Data Collection Protocols(DCP) are examined. Based on the network topology corresponding to the DCP, we want to choose a DCP that is the most efficient and reliable. The Direct Delivery protocol was chosen as most suited in combination with WiFi operating at 2.4 GHz. WiFi can deal with the high data rates and communicate over large distances.

The number of base stations needed is determined based on tests performed on the football field. The choice was made to place the antenna of the sensor shorts on the back of the athlete because there it has the most negligible chance of obstruction due to the athlete's limbs. The athlete's own body could cause attenuation of 10 dBm. Additionally, it was chosen to have the base station at 2.2 m height, to have fewer chances of obstruction due to the athletes on the football field.

Multiple tests have been performed to determine how many base stations are needed. First, the possible signal strength values of the sensor shorts around the football field have been measured for a single base station. After that, a relation between the measured signal strength and possible loss and delay has been established. These results made it clear that more than one base station is needed. In tests with two base stations, it became clear that data might get lost when there is no base station in the radiation plane of the patch antenna of the sensor shorts. Therefore the choice was made to have four base stations, one on each corner of the football field, so that there is always a base station in the radiation plane of the patch antenna. Based on the test results, the choice was made to have four base stations, one on each corner of the football field. Assuming that the sensor shorts' RSSI values are always higher than −80 dBm, the system with four base stations would have a loss of less than 5%. The upper boundary for the delay would be 5 s.

Preface

This thesis is written as part of my graduation project to obtain a master's degree in Embedded Systems at the Delft University of Technology. I am pleased and grateful to arrive at this huge milestone. Therefore I would like to thank the people who have supported and guided me toward this achievement.

First, I want to thank my parents for their support, love, and care during my education. Besides, I want to thank my brothers for their time and patience, especially during the sometimes cold testing days. For the test days, Murat opened the football club for me in his spare time. I appreciate that, thank you very much.

I want to thank the people who guided me during the project, Andre Bossche, Annemarijn Steijlen, and Jeroen Bastemeijer. Thanks for the guidance and discussions throughout the project. I enjoyed working together during this project. I would also like to thank the people who have been part of the CAS meetings.

Lastly, I would like to thank my friends. Especially Koen, who has been a valuable partner during the master courses and from who I learned a lot. Besides, I would like to thank Dewwret for his support during my graduation project.

Zakaria Abdellaoui
Delft, August 2022

Contents

Nomenclature

Abbreviations

Abbreviation	Definition
ACK	Acknowledgement
AEDT	Adaptive Energy Aware Data Aggregation Tree
AP	Clear to Send
CSMA/CA	Carrier-Sense Multiple Access with Collision Avoidance
CTS	Clear to Send
DCP	Data Collection Protocol
DOF	Degrees of freedom
DTC	Data Transmission Cycle
EM	Electromagnetic
FIFA	Fédération Internationale Football Association
GND	Ground
ID	Identity
IIAB	The International Football Association Board
IMU	Inertial Measurement Unit
IP	Internet Protocol
KNHB	Koninklijke Nederlandse Hockey Bond
KNVB	Koninklijke Nederlandse Voetbal Bond
LEACH	Low-Energy Adaptive Clustering Hierarchy
LLC	Logical Link Control
LOS	Line-of-sight
MAC	Medium Access Control
MFELACS	Modified Fast and Efficient Lossless Adaptive Compression Scheme
MIMO	Multiple Input Multiple Output(MIMO)
NHE	Nordic hamstring exercise
OSI	Open System Interconnection
PL	Path Loss
RSSI	Received Signal Strength Indicator
RTS	Request to Send
TBRP	Tree-Based Routing Protocol
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
UART	Universal Asynchronous Receiver-Transmitter
UDP	User Datagram Protocol
UWB	Ultra-wideband

1 Introduction

Football is the most popular sport around the world. More than 100 million licensed players are registered with the Fédération Internationale Football Association (FIFA) [1]. Football is an intensive game; hence it has a high injury rate. Football players suffer 710 reportable injuries per 100 000 hours of training and competition, this number is higher than most occupations in the United Kingdom [2]. Football players are mostly injured in the lower extremities [3]. One of the most common injuries in the lower extremities is the hamstring injury. Despite various methods proposed to prevent hamstring injuries, the incidence of hamstring injuries had increased by 4% per year from 2001 till 2014 [4]. The injuries of football players present high burdens to football clubs in the financial and performance aspect. It has been analysed that a team's performance decreases when dealing with a higher injury burden and lower availability of the football players [5]. Because of the decrease in performance, teams might lose money when they lose games. It was calculated that the average cost of missing a first-team player for one month is around ∈ 500.000 [6]. Due to the high risk of injuries and the negative effects of it, much research is done on the prevention of injury and monitoring systems of the players. For example Horst et al. proposed a Nordic hamstring exercise (NHE), which strengthens the eccentric muscle strength [7]. It was observed that doing these exercises significantly decreases the rate of hamstring injuries. Despite the prevention methods, there are still too many injuries.

To attack this issue, the KNVB (Koninklijke Nederlandse Voetbal Bond) and the KNHB (Koninklijke Nederlandse Hockey Bond) came up with an idea, which should monitor the explosive actions of the players on the field. Because most injuries occur on the lower extremities, the idea was to monitor the movements of the lower body. By monitoring the lower body, models can be developed to find the risks of specific movements related to injuries. To monitor the lower body, KNVB wants to design sensor shorts in collaboration with the TU Delft. Movement scientists from the University of Groningen and VU Amsterdam will use the sensor shorts to create the models for predicting injuries. Those sensor shorts will be equipped with three Inertial Measurement Units (IMUs), one will be located on the trunk, and the other two will be placed on the right and left upper leg. The IMUs measure the angular velocity (dps), linear acceleration (g) and magnetic field (μT).

The doctor or coach of the team should be able to monitor the measured data live. The measured data should be sent to a central computer to achieve this. It is impossible to have the players connected to the central computer by wires. A wireless connection is a logical alternative. For the wireless connection, a wireless network needs to be developed, in which the sensor shorts data is sent wireless to a central computer.

1.1. Current Prototype

The design of the sensor shorts has been going on for three years now [8]. Various prototypes have already been developed. This thesis will describe a new prototype with extra functionalities based on the latest prototype. In the latest sensor shorts prototype, there are three IMUs. Two of the IMUs are located at the upper part of both legs and one at the trunk. The IMUs are a combination of a 3-axis magnetometer, gyroscope and accelerometer. The accelerometer and gyroscope form one sensor module (Figure 1.1), which is the ICM-20649 [9], with a range of ±4000 dps and ±30 g, respectively. The magnetometer is from AsahiKASEI and is called the AK8963, with a range of ±4900 μT [10]. The magnetometer is connected to the ICM-20649 using the I^2C protocol. Through this coupling, the two sensor modules will work as one sensor with 9 degrees of freedom (DOF), and it can be read out using an SPI connection. The two sensor modules together form the IMUs, Figure 1.1 shows the internal layout of the IMU.

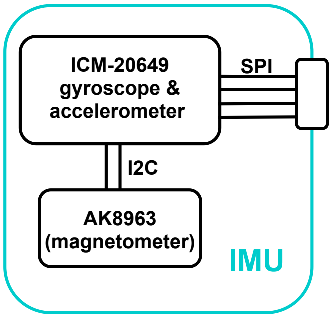


Figure 1.1: IMU internal layout

The three IMUs send data to the main PCB (also called the Central PCB), which is located at the trunk. For the placement of the main PCB, the trunk has been chosen because it will introduce the least amount of obstruction when wearing and will most likely cause the least amount of damage in case of falling. The PCB at the trunk is responsible for reading out the sensors and storing this data on the SD card. It is also responsible for the power supply in the system. The main PCB is equipped with a microcontroller which is an Arm Cortex-M4 core from the STM32F411 Nucleo-64 development board [11]. The following are the most relevant specifications of the microcontroller:

- running at 100MHz max
- 512kB of FLASH
- 128kB of SRAM
- 125 DMIPS (or 1.25 DMIPS / MHz)

The power supplied to the PCB, and the IMUs comes from a Grab' n Go (GNG-109) 1000 mAh power bank. The battery is connected with a Micro USB connectory and provides a voltage of 5 V. The real-time operating system FreeRTOS [12] is used on the microcontroller. In FreeRTOS, tasks can be created, and FreeRTOS ensures that those tasks will be executed within a specified time constraint. A task consists of an activation-, start-, computation- and finishing time. The current prototype is divided into the following tasks:

- dataCompression: this task will compress the data stored in the buffer. So that the amount of data sent by the transmitter will be reduced.
- initTask: this task will create a new file to write data to by the storeSDTask. It will also handle the buttons' input and provide feedback using the LEDs.
- readAccGyroTask: this task will read out the accelerometer and gyroscope for all IMUs and passes the values to the storeBuffTask.
- readMagTask: this task will read out the magnetometer from the IMUs and passes the values to the storeBuffTask.

- storeBuffTask: this task will store the IMU readings from the accelerometer, gyroscope and magnetometer into a Ping-Pong buffer. Each buffer has a size of 4 kB.
- storeSDTask: this task will write the Ping-Pong buffer when filled to the SD card.

In the latest sensor shorts research, Burgers performed tests with two antenna types [8]. The goal of the tests was to find what type of antenna provides the most reliable and efficient wireless communication between transmitter and receiver. Besides, tests with a dual-antenna configuration were performed to determine whether this might improve the communication between transmitter and receiver. For testing the antenna, the xPico240 was chosen. This is a WiFi module which supports a dual-antenna configuration.

The conclusion of Burgers was that it is worth using a dual-antenna configuration [8]. With a dual antenna configuration, pattern diversity can be applied. Pattern diversity could be used to overcome the effect of a blocked signal. When a signal from one antenna is blocked due to obstructions, the other antenna might provide a better connection. The antenna chosen for the latest prototype was the W3230 from PulseLarsen patch antenna because of its radiation pattern (Figure 1.2). Besides the patch antenna provides a better signal strength.

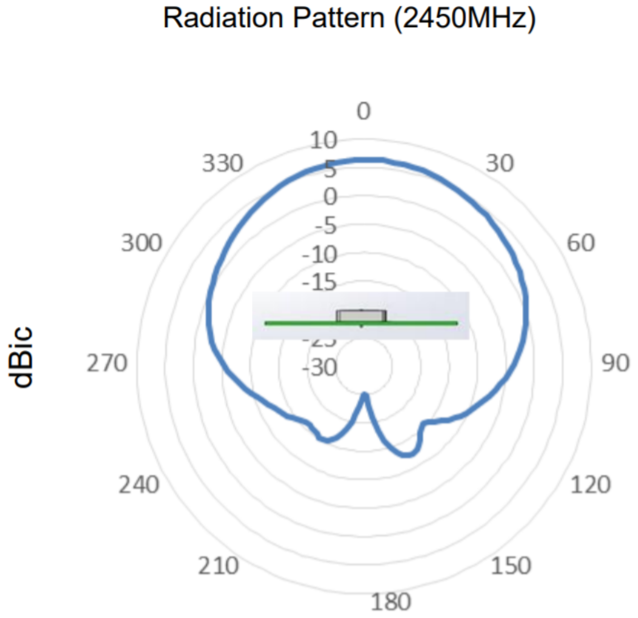


Figure 1.2: The radiation pattern of the W3230 patch antenna [13]

The activation of the tasks happens at a fixed frequency. Every task has a particular priority declared. When multiple tasks are activated simultaneously, the task with the highest priority of all activated tasks will be executed first. There is a preemptive highest priority scheduling scheme used. When a task with a lower priority is executed, it can be interrupted by a higher priority task. The scheduling scheme for the other tasks in the last prototype is shown in Figure 1.3.

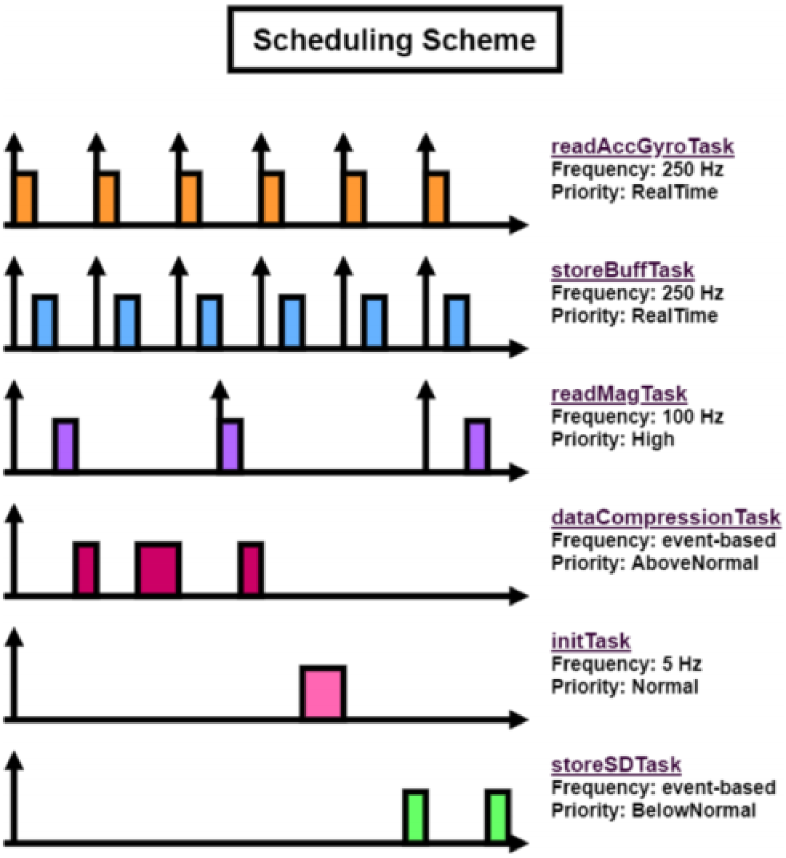


Figure 1.3: The scheduling scheme of the latest prototype, which had the dataCompressionTask as extra functionality [8]

From Figure 1.3, it can be seen that the dataCompressionTask runs when all sensors are read out and safely stored in a buffer. The accelerometer and gyroscope data are read out at 250 Hz. The magnetometer is read out with a frequency of 100 Hz because that is its limit. Each of the sensors senses over three different axes, with a sample length of 16 bits per axis. This results in an overall load of 86.4 kb/s for sensor shorts with 3 IMUs. For sensor shorts with 5 IMUs, the overall load is 144 kb/s.

When a team of 11 players are wearing the sensor shorts with 5 IMUs, the total load is 1.6 Mb/s. A compression algorithm was used in the previous prototype to reduce the size of the data. The algorithm used is the modified fast and efficient lossless adaptive compression scheme (MFELACS). In the tests done using MFELACS, the compression rate in the most intensive moments of a soccer game is 23 %. This means that the maximum load of one player is 66.5 kbit/s and 111 kbit/s for sensor shorts with 3 IMUs and 5 IMUs, respectively.

1.2. Next Prototype

The goal of the sensor shorts is to collect the measured data of the athletes in (near) real-time on a central computer. The central computer is also referred to as the central server. On the central computer, the team doctor should be able to analyse the athletes' data. The team doctor needs to get the data in real-time to warn an athlete as soon as possible. With the latest prototype, it is not yet possible to wirelessly transmit/receive data on a central computer. It is essential to have a reliable and robust connection to the central computer from the athletes. The wireless network should be able to cover the whole football field and handle the data of all players on the football field. Because it has to be made sure that all the athletes' data arrive without errors and in real-time, it is important to choose a network topology that can provide this. This network topology determines how the different network elements are connected. It is also important that this network architecture can support the data load of the athletes.

A data collection protocol is needed to collect the athletes' data on a central server. The Data Collection Protocol (DCP) describes how to get the data from the athlete to the end destination [14]. The DCP could also be described as a routing protocol, but DCP will be used for this report. In this thesis, a DCP will be proposed for the sensor shorts. It is essential to make a well-considered choice for the DCP, because the choice of the DCP affects the delay in receiving the data on the central computer. The DCP and network topology are related. The DCP describes which nodes to use to get the data from the athlete to the central computer. Thus it says how the nodes in the network are connected. To have a robust and reliable low delay data transfer, the best DCP for the sensor shorts application has to be found.

Finally, we will look at the communication protocol for the chosen DCP and the corresponding network topology.

1.2.1. Requirements for the next prototype

The movement scientists concluded that only the trunk and the two upper leg IMUs are necessary for predicting hamstring injuries. Due to this fact, the next prototype will be focused on sensor shorts with 3 IMUs. Here are the requirements listed for the sensor shorts:

1. *Coverage:* The systems area coverage should be scalable for football and hockey. However, since a football field is larger, we focus on football covering football fields. Therefore our system should cover a pitch of at maximum 120 m × 90 m area [15]. This is the maximum size of a football field following the rules of The International Football Association Board (IIAB).

2. *Number of tracked players:* There are 11 players on the field during a football match. This requires the system to be able to track 11 players at the same time.

3. *System setup:* The system should be easy to set up, and its functioning should be independent of the area.

4. *Operational time:* The system should be able to last for 90 minutes which is an entire football game.

5. *Data rate:* The total data produced in the system in the case of 11 players wearing sensor shorts with 3 IMUs is 950 kb/s. The system's throughput should be able to process this.

6. *Communication:* Data delivery to the central server should be energy efficient to last for a game. Besides, it should survive when packets are missing. Indeed we want the packet loss to be as low as possible. An acceptable packet loss during a game is below 5%. This means that during a game of 90 minutes, 5 minutes of data loss is acceptable. Besides, the delay between measurement and arrival time at the central server should be low. A maximum delay of 30 seconds is acceptable for our system.

1.2.2. Research Questions

In the previous section, it was stated that to have a robust and reliable low delay data transfer, a suitable DCP has to be found. Literature research will be performed to explore the possible DCP that could be used. The research questions that the literature should answer are:

1. Which DCP is most suitable for low delay, reliable and robust data transfer?
2. What is the corresponding network topology of the DCP found?
3. What communication protocol to use for the chosen DCP and the corresponding network topology?

1.2.3. Thesis outline

The thesis is outlined in the following way. Chapter 2 presents some wearable monitoring devices for sports athletes found in the literature. Those devices are then compared to our device. In Chapter 3, we will look at how the data can get from the sensor shorts to the central computer. Different methods to do this from literature are presented and compared. Besides, we discuss the most common communication protocols to find out which one might suit our application. In Chapter 4, the system is designed, which will be tested on the football field in Chapter 5. The thesis ends with the conclusion and discussion in Chapter 6 and Chapter 7, respectively.

2

State of art of wearable monitoring devices in sports

Before looking into the DCP and the corresponding network topology, we will first look into current wearable monitoring devices to get a general idea of the topic. There are already several works of literature on wearable monitoring devices that are used in sports. This section will discuss some of the literature.

In 2008, Ermes et al. and Morris et al. were one of the first who mention wearable sensors in their papers [16, 17]. Ermes et al. stated that it is valuable to have a wearable monitoring device which recognises activity [16]. Because with activity recognition, the user can be provided with recommendations regarding physical activity and sports to promote a more active lifestyle. In [16], they collect data from 3D accelerometers placed on the hip and wrist, and GPS information to recognise the activities. The sampling frequency for the accelerometer is 20 Hz. The paper is mainly about the performance of the activity recognition. Morris et al. presented a new wearable device, namely one using sensors to monitor sweat [17]. These types of sensors can sense parameters like sweat pH and sodium levels. This can be used to get information about the hydration of athletes.

Like our sensor shorts, the monitoring device presented by Bräysy et al. is also equipped with an accelerometer [18]. The device they propose is designed to track movements of sports team players and also gather sensor data. The players are equiped with a nanoLOC AVR Module hardware unit, these units consist of an AVR microcontroller and nanoLOC TRX transceiver. The communication protocol operating on the nanoLOC TRX transceiver is Zigbee. There is already a microcontroller for the sensor shorts, but for the wireless communication, we should also add a transceiver as done by Bräysy et al. The wearable monitoring device proposed by Bräysy et al. consists of an IMU with a 3D-accelerometer which is sampled at 10 Hz [18].

Bräysy et al. performed measurements to explore the wireless communication. For example, they compared the amounts of packets missing or erroneous packets when one or two sensor units are operating simultaneously. In the case of two sensor units operating simultaneously, there are more erroneous packets. This might be caused due to the collision occurring between the data packets of the two sensor units. A Medium Access Control (MAC) protocol can be used as a solution to prevent this. MAC protocols describe which transceiver is allowed to communicate in the wireless medium. MAC protocols should prevent collisions of data packets due to simultaneous communication. Bräysy et al. provided a MAC protocol in which each sensor unit gets its time slot assigned. The sensor unit can only communicate in its dedicated time slot [18].

The goal for our sensor shorts is to be sampling at a frequency of 250 Hz, this is a high sampling rate compared to the previous papers in this section. The monitoring device with the highest sampling frequency is designed by Liu et al., the sampling frequency of their device is 148 Hz [19]. The monitoring device of Liu et al. is designed for badminton, meant to measure the training load of the badminton athletes. A high sampling frequency might bring challenges in processing the new data and saving it on the microcontroller, another challenge might be to get the data as fast as possible to the central server.

All wearable monitoring devices have a general system architecture. The architecture was researched to understand the monitoring devices' architecture better. In [20] and [21], the architecture of the monitoring device built is explained. Figure 2.1 shows the general architecture based on those two papers. The core of a wearable sensor node is the microcontroller. The sensor units and a wireless transceiver are attached to the microcontroller. The task of the microcontroller is to read the sensor data and send it to the central server via the transceiver. The arrows in Figure 2.1 show the direction of the communication. For powering the system, a battery is connected to the microcontroller. In the previous prototypes of this project, all blocks except the transceiver were chosen. This thesis chooses the transceiver module.

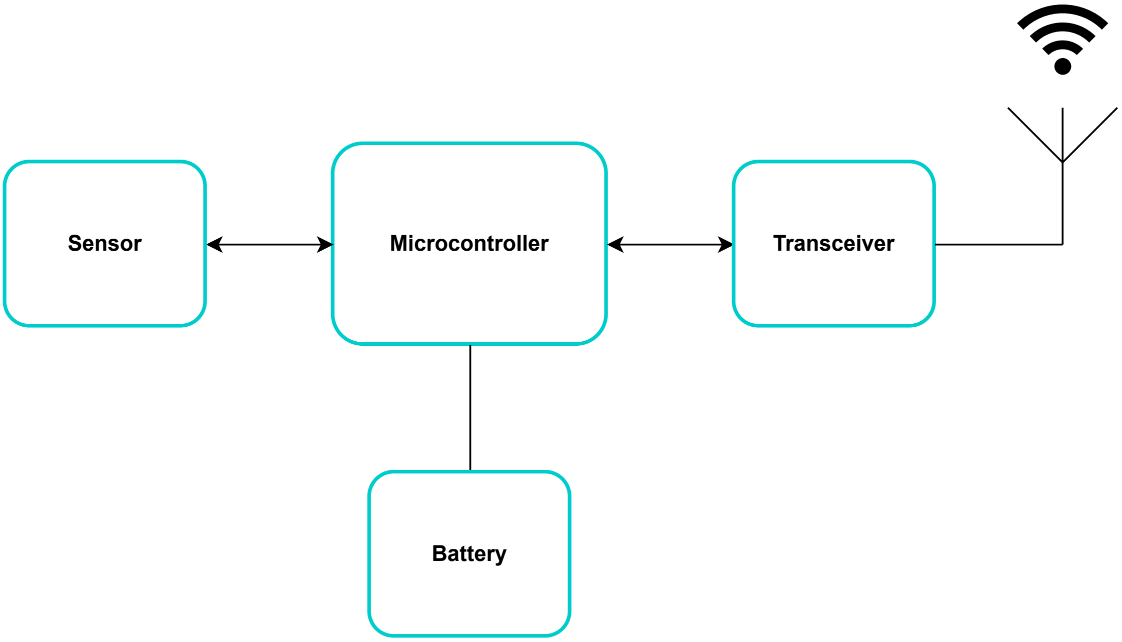


Figure 2.1: The general architecture of a wearable sensor system with wireless connection

3
Literature research on Data Collection Protocols in sensor networks

In this chapter, first, the Data Collection Protocols found in the literature will be described and discussed. After that, we are going to choose one of the DCPs for the sensor shorts. The chosen DCP should be able to support the needed bit rate, and it has to provide a low delay and reliable data transfer. After the selection is made, we will look at which communication protocol is the best to use for the choice made and for our application. The choice will mainly be based on the expected delay, energy consumption, use of resources and the complexity of the protocol.

3.1. Data from base stations to central server

For monitoring devices used to monitor athletes real-time, there is a general network outside the sports field. Like in the MarathonNet project, in this project, runners during a marathon are monitored real-time, so the spectators have information on the athletes' performance [22]. Pfisterer et al. did research on the dependencies of connectivity [22]. In order to get the athletes' data to a central point, so-called base stations are used. The base stations consist of a transceiver connected to the central server. In a marathon, players run a predetermined track. The base stations are positioned along the track. With the help of those base stations, the data from the athletes can be transmitted to the central server. Pfisterer et al. connected the base station and central server by a wide or local area network technologies like WLAN, GPRS or a wired network [22].

Like the MarathonNet project, base stations are used for football applications too. For example, Martinelli et al. developed an ultra-wideband (UWB) positioning system that monitors football athletes, especially for five-a-side football [23]. In this system, the players are equipped with a UWB transmitter located on the upper back of the body. Martinelli et al. call the base stations "anchors" or "receiving antennas". Martinelli et al. used four receiving antennas, which are placed on the corners of the field. Besides they analysed the height of the receiving antenna and its effect on the system's performance. This effect is analysed by considering three heights for the receiving antenna: 1 m, 1.6 m, and 2 m.

There are four parameters used to make a decision. The most interesting parameter is the signal power detected by each receiver. The signal power can tell us something about the best height for the receiving antenna to get the best possible signal power at the receivers outside the field. The received signal power should be above a particular power level, indicated as the receiver sensitivity. The signal cannot be detected when the power is below the receiver sensitivity of the receiving antenna. So to have a better chance of detecting the signal at the receiving antenna, the height of the antenna might be essential. Martinelli et al. concluded that the receiving antenna height of 1.6 m results in a better reception power for the four different receivers [23]. This height is around the same height as the transmitter antenna on the upper back. Because they are both at the same height, the line-of-sight (LOS) path is shorter and therefore, less power will be lost than having the receiving antenna at 1 m or 2 m. So for a more reliable path between the transmitter and receiver, it is essential to place them at the same height.

However there is one note to this conclusion. The conclusion is based on five-a-side football. The field in five-a-side football is much smaller than a regular football field, it is about 10 times smaller. For five-a-side football, the size is around 39 m × 20.2 m. Therefore Martinelli et al. suggested to analyse the antenna heights for a regular football field. The results for a typical field might be different. For example, there might be fewer obstructions because the distances between players are larger.

Another thing to remember is that a smaller area has to be covered by the base stations for the five-a-side football. For a larger field, the area that should be covered is more extensive, resulting in more base stations needed. Sivaraman et al. designed a monitoring device to monitor players on a standard football field [24]. The entire football field is covered by eight base stations, which is twice the amount as in [23]. The positions of those base stations are shown in Figure 3.1. As it can be seen from the figure, the size of the playfield in this scenario is 93 m × 70 m. In this application, one base station of the eight is the master base station, which extracts the data from the other base stations.

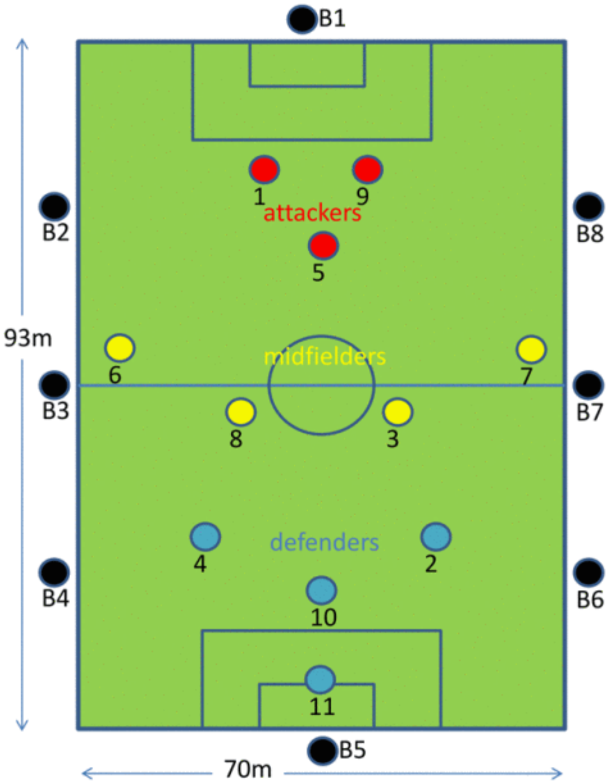


Figure 3.1: Overview of the base stations positions outside the football field in [24]

The setup and operation of those stations is a major effort before each training session or football game, besides the cost increases with every base station added. Because of this, the goal is to keep the base station at a minimum. However, this should not impact the connection between a pair of sensor shorts and the base stations. When too few base stations are used, this might cause players not to be able to communicate with the base station. When a player can not communicate with the base station, the data will not be sent to the central computer, or it will arrive with a delay once the player gets in the range of the base station. So one of the trade-offs for the number of base stations is a higher delay but a lower effort for setup and costs, or a higher cost and more effort but a lower delay and more coverage.

Another critical design choice for the sensor shorts is how the system gets the athletes' sensor data to the base stations. The Data Collection Protocols describe how the data is sent to the base stations. These protocols also affect the number of base stations needed to cover the football field. Therefore in the next section, we will look into interesting Data Collection Protocols for the sensor short.

3.2. Data from athletes to base station

This section will discuss the different possible Data Collection Protocols and the corresponding network architecture. The Data Collection Protocols and network architecture are judged based on the expected delay of receiving data, the energy use, and the reliability of the network architecture. An essential characteristic of a wearable sensor device is that it is limited in capacity in terms of computing power, memory, data aggregation and more. So the DCP should also be judged on the use of resources. We want to minimise the use of the resources of the sensor nodes. In this section, the goal is to find a relative score for those parameters. The choice for the DCP will depend on the mentioned parameters.

3.2.1. Direct Delivery

Dhamdhere et al. tested a couple of DCPs for a monitoring device in football [25]. The proposed protocols will be discussed in these subsections. The setup used for the DCPs is shown in Figure 3.1. One of the most simple DCPs proposed by Dhamdhere et al. is the Single-Hop DCP. In Single-Hop, the sensor data goes from the athlete immediately to the base station. The data does not have to pass any other node in the sensor network. Single-Hop is also referred to as Direct Delivery. For this DCP, the performance depends on the athlete's position on the football field. As can be imagined, when an athlete is further away from the base station, the performance is worse, for example, in terms of delay. The data sent may experience more delay because it has to travel a longer distance.

For evaluating this DCP, the delay experienced by individual samples is measured. The transmission of the samples takes place once every second. When the data does not arrive at the base station in the first second, the data will be sent again at the next instance. It appeared that for most players, the delay lies above 50 seconds. Nevertheless, not all players have a delay above the 50 seconds. The goalkeeper, number 11 in Figure 3.1, has a smaller delay because it is close to the base station.

For some data samples, the delay measured was 200 seconds, and for player three in Figure 3.1, the mean delay was around 200 seconds. Therefore, the conclusion of Dhamdhere et al. is that the data can not be delivered with an acceptable delay to a base station [25].

How these significant delays are caused is not discussed in the paper, but an apparent reason might be that some of the players are, for some time, not within reach of a base station. Therefore the data has to wait until an instant when there is a base station in the reach. Another problem that occured during the tests performed by Dhamdhere et al. is that one base station stopped working and therefore did not receive any data. This might explain why there was no base station in reach some of the time. So to solve the high delay, one of the solutions might be to have more base stations or transmit at higher power. Another solution is to use Multi-Hop protocols instead of Single Hop routing. Dhamdhere et al. also explored Multi-Hop protocols besides the Single Hop protocol, these protocols will be discussed in the following sections.

The network architecture when using the Direct Delivery method is a double star architecture, as shown in Figure 3.2. The squares in the image are the base stations, the base stations are connected to a central server. This central server is where all athlete data is collected and analysed. As seen from the image, when one of the sensor shorts stops working, it only affects its connection. When the base station stops working, it affects all shorts for whom the distance to the stopped base station is the shortest relative to the other base stations. Those shorts have now to communicate over more considerable distances or aren't able to communicate at all because there is no base station within its range.

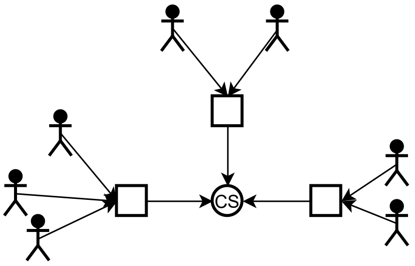


Figure 3.2: Global overview of the network architecture in case of Single Hop routing, to communicate to the central server, the sensor shorts are connected with the closest base station.

3.2.2. Full Flooding routing

As said in the previous section, a solution for the high delay experienced when using Single Hop routing is using a Multi-Hop routing protocol. Dhamdhere et al. proposed and tested different types of Multi-Hop protocols [25]. One of the Multi-Hop protocols is the Full Flooding routing protocol. In this DCP, the data is continuously transmitted every second till confirmation of reception is received. In this protocol, the data can also be transmitted to neighbour athletes. The data is transmitted to all athletes and base stations in the range. When a neighbour athlete receives this data, it is re-transmitted by the neighbour. Once the base station receives the data, the base informs the player that the data has been received successfully. The athletes' wearable sensor module will delete this data since it has already been received and should not be transmitted again. The problem with this protocol is that it is intensive in terms of memory usage and data rate. The reason is that besides their regular data also, the neighbouring data has to be transmitted and saved. A consequence of the additional data to be sent is that more power is needed, resulting in the energy consumption being higher than the previous DCP.

However, since every available node participates in forwarding the data, this protocol guarantees the lowest delay. Using this routing protocol resulted in a delay of between 7 and 18 seconds. Dhamdhere et al. also analysed how often a sample is transmitted until a base station receives it. The Full Flooding protocol needed 60 times more transmission than the Single Hop routing protocol [25]. Because the sensor nodes transmit much more in this DCP, the energy consumption is higher. There are no numbers mentioned about energy consumption, but this is an obvious consequence of the power needed for wireless communication transmission.

The Full Flooding DCP is also more complex than the previous DCP in terms of complexity. Because in this DCP, the data can also be sent to other players. Therefore, the sensor modules should also listen for data and know what to do with the received data. So it needs some logic in the system to know how to process the received data. Also, the complexity of the network architecture has increased with this routing protocol. Because everything in the range of a specific player's module is connected to the module, the field network consists of several local mesh networks. Because of this, there are multiple paths from a specific athlete to a base station. Therefore using the Full Flooding routing protocol is more reliable than the Single Hop routing protocol.

3.2.3. Random Forwarding

The previous DCP proposed in subsection subsection 3.2.2 is too resource-intensive, making it unusable for practical implementation. The goal of the next DCP is to limit the use of resources. The Random Forwarding protocol is a protocol that limits the use of resources. The limitation is achieved by not transmitting the data to all the neighbours in the Random Forwarding protocol. Limiting the use of resources is done by probabilistically forwarding the data to a neighbouring player at the end of a time slot if the data cannot be delivered to the base station. In the Random Forwarding DCP, when the device of a player can hear n other players, the data is sent to one of the neighbouring players with a probability of $1/(n+1)$. This procedure is repeated at the end of every time slot till a base station is hit. With this DCP, the sparsely connected players experience some of the advantages of the Full Flooding protocol by using the players who are better connected. Since the data is sent to the neighbouring players based on probability, the data delivery delay performance is better than the Direct Delivery protocol. When using this DCP, all the players have a mean delay below 50 seconds. Because the data is not sent to all neighbours, the use of resources has become less than the Full Flooding DCP. The reduction of resource consumption results in less energy consumption for wireless communication since the number of transmissions has decreased. Besides less memory is needed compared to the Full Flooding protocol. This DCP is more complex than the previous ones because the number of neighbours should be known, and calculations must be done before transmitting.

Because the wireless links between the athletes are probabilistically determined, it is difficult to define the network topology for Random Forwarding. But we know that depending on probability, there might be a link with neighbouring athletes. If a base station is within the range of the sensor node, there is an immediate link with the base station. So in this network, the base station could be reached via the neighbouring sensor nodes.

3.2.4. LEACH-Mobile

The previous DCPs, are DCPs explored and tested by Dhamdhere et al., and are intended for a sensor network in football [25]. It is also interesting to look at other DCPs, which have not yet been deployed or tested in a sensor network on the football field. Heinzelman, Chandrakasan, and Balakrishnan proposed an interesting DCP, called LEACH (Low-Energy Adaptive Clustering Hierarchy) [26]. LEACH is a clustering-based protocol, it was designed to minimise the energy consumption of the sensor nodes in a sensor network. The idea behind this protocol is that there will be local coordination and control for setting up a cluster. Every cluster consists of a Cluster Head, which will be randomly changed. The Cluster Heads are the only sensor nodes that communicate with the base station. By doing this, the energy consumption for the other nodes is reduced. The distance from the base station to the sensor nodes is typically large in sensor networks using LEACH. Therefore by having the Cluster Head collect the data of all the sensor nodes in the cluster, the sensor nodes save energy because the transmit distance is now much smaller. The energy usage will be uniformly distributed by rotating the Cluster Head randomly. There are multiple variants of the LEACH protocol one of them is the LEACH-Mobile protocol [27]. This protocol is interesting in our case because LEACH-Mobile is designed for wireless mobile sensor networks. In contrast, the classic LEACH protocol is designed for static sensor nodes.

The LEACH-Mobile protocol consists of two phases. The first phase is the set-up phase, in this phase the clusters are organised. The cluster heads are selected in the set-up phase, and the associated sensor nodes are determined. The non-cluster head nodes transmit a cluster joint-request. All the cluster heads which hear this join-request, respond with a cluster head advertisement. Based on the power of the received advertisement, the non-cluster head node will join the cluster head with the highest power. To let the cluster head node know that the non-cluster head will be joining the cluster, it responds by transmitting data to the cluster head. The cluster head keeps a list of the mobile sensor nodes that are members of the cluster. The cluster head makes a TDMA (Time Division Multiple Access) schedule based on this list. In the TDMA schedule, every mobile node gets a slot assigned by the cluster head. The mobile node can only transfer the data to the cluster head in its assigned time slot.

When the set-up phase has finished, the steady-state phase will begin. In the steady-state phase, the sensor data transfer will start. In the steady-state phase, time is broken into frames. During a frame of a cluster head, every mobile node has its time slot according to the TDMA schedule. In LEACH-Mobile, all nodes should be time-synchronised to synchronise all the frames. During the steady-state phase, the cluster head sends a request for data message to the non-cluster head node according to the TDMA schedule. When the non-cluster node receives the message, it will send the sensed data to the cluster head. If the non-cluster head does not receive this message, it will wait until the next frame. When it again does not receive the request data message, it will try to join another cluster by transmitting a join request as in the procedure followed during the set-up phase. The cluster head keeps a list of the nodes which did not transmit anything. At the end of the frame, this list is updated. When a mobile node does not transmit two times, it will be deleted from the list, and the TDMA schedule will be rescheduled. The node will be deleted because it is assumed that it has moved on and will join another cluster. The cluster head will also transmit the TDMA schedule to the cluster member nodes. After a specific time, which is predetermined, a new round begins with each node determining if it should be a cluster head for the next round.

LEACH-Mobile would result in a network topology as shown in Figure 3.3. The arrows in Figure 3.3 indicate in which direction the sensor data travels. As can be seen from the figure there are 2 clusters, in which the cluster head receives the data from the non cluster head athletes. The cluster head is connected to the closest base station. From this figure we see that LEACH-Mobile may sometimes be less efficient or cause problems. This is the case when an athlete which is far from the base station is the cluster head, for example athlete "a" in cluster 2.

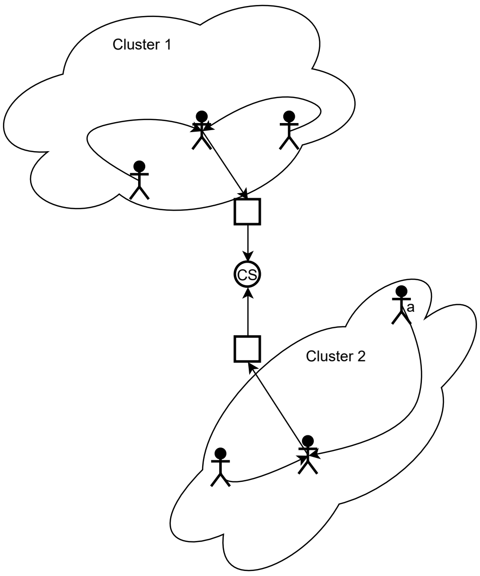


Figure 3.3: General overview of the network topology in case of LEACH-Mobile, with two clusters.

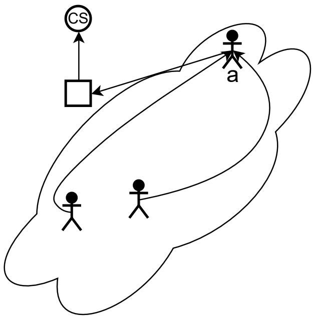


Figure 3.4: Athlete "a" of Figure 3.3 is now the cluster head, in this situation the cluster is not energy efficient

In Figure 3.4, athlete "a" of Figure 3.3 is now the cluster head. As it can be seen from the figure, LEACH-Mobile is now not that efficient anymore. Because the non-cluster head node has to transmit the sensor data over a much larger distance than the distance to the base station, this problem could be caused by multiple factors. Which is the number of sensors, the area of the sensor network field and the number of base stations. With a few sensor nodes on a large area field and a few base stations, the chances are higher that a sensor node with a large distance to the base station becomes the cluster head. Furthermore, the distance between the sensor nodes might be large because of the large area and few sensor nodes. The LEACH-Mobile protocol seems to be more efficient when there are many sensor nodes. Heinzelman, Chandrakasan, and Balakrishnan, they simulated the LEACH protocol in a network of 200 m x 200 m, with 100 sensor nodes [26]. There are no results shown with fewer sensor nodes, but due to the problems that might occur with fewer sensor nodes, the expectation is that LEACH-Mobile is not energy efficient in the smart sensor shorts project. However, it might be better than the previous collection protocols regarding energy consumption.

In the literature, there is not much focus on the delay of the LEACH-Mobile protocol. However, because of its characteristics, the data delay will be higher than the Direct Delivery technique because the data is not directly sent to the base station. Of course, it will also have a higher delay than the Full Flooding technique, Since Full flooding uses maximal resources. Comparing LEACH-Mobile and Random Forwarding is more difficult because the range of the communication protocol plays an important role. For LEACH-Mobile, the assumption was made that all the sensor nodes should be able to reach the base stations. Under this condition, it might be fair to say that Random Forwarding will perform better in terms of delay. Because in Random Forwarding, the data will be sent directly to the base station when it is within its range.

LEACH-Mobile is the most complex DCP because all sensor nodes need to be synchronised in LEACH-Mobile. Also, the set-up phase needs some computing. Besides the computation needed, the memory consumption is also high for the cluster heads since it needs to store all the data of the non-cluster heads.

3.2.5. Tree Based Routing Protocol

Another type of network topology is the tree topology. The tree topology creates logical links between sensor nodes based on a tree structure. There are different roles for the sensor nodes in a tree structure, the roles a node can have is a root, intermediary and leaf node. The typical structure of such a topology is shown in Figure 3.5. As shown in Figure 3.5, leaf nodes send data to the connected nodes, but they do not receive data from other nodes. Intermediate nodes receive data from nodes on the higher level and transmit data. In the end, all the sensor node data is collected at the root node. The root node is responsible for sending all the sensor data to a base station.

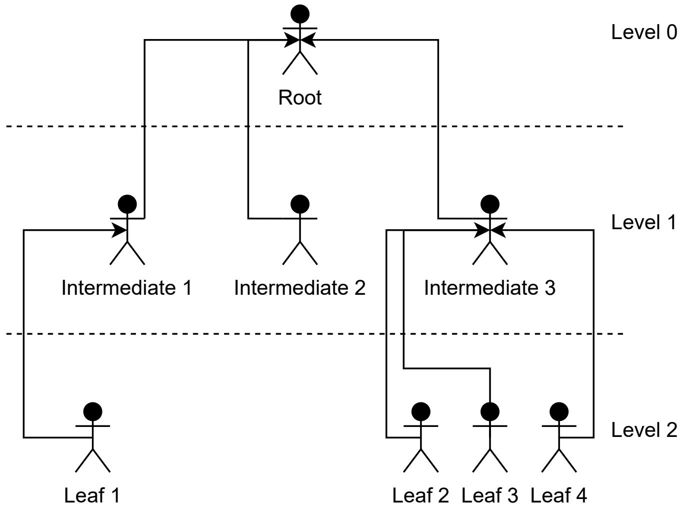


Figure 3.5: Network topology of a Tree Based DCP, where the root node should be connected to a base station

A DCP which uses the tree topology as shown in Figure 3.5, is the Tree-Based Routing Protocol (TBRP) [28]. In TBRP, the network forms a tree with different levels. These levels are based on the distance to the base station. Depending on the level of a node, it has some degree of constraint. The degree of constraint constrains the maximum amount of connected nodes a node can have.

The Tree-Based Routing algorithm consists of three phases: the tree formation phase, the data collection and transmission phase, and the last one is the Purification phase.

The tree formation phase starts with the base station sending an *init* (initialisation) message to all sensor nodes. This message contains information about the position of the base station. The sensor nodes calculate their distances to the base station after receiving the position of the base station. According to the calculated distance, the node assigns a level to itself. Once the level has been determined, the node broadcasts a *join_req* packet. This packet contains information about the node ID and the level of the node.

Some nodes in the network might receive this message. After receiving the *join_req* message, there are three conditions checked. The listening node should have a parent node, the level of the listening node should be lower than the level in the *join_req* message and the last one is that degree should be lower than the degree constrain. The listening node responds with a *join_ack* message if all three conditions are met. The requested node joins the node from which it gets the first acknowledgement. These formation steps are repeated until a whole tree has been formed.

After the three formation phase is finished, the data collection and transmission can start. First, a TDMA schedule is created by the father node. A father node is a node which is nodes connected with nodes of a higher level. The higher-level nodes are the children of the father node. The child node sends its data in its time slot to the parent node. The parent node aggregates its data with the data received from the children and then forwards it toward its parent node. The root node, which is near the base station, collects all data of the whole network to send it to the base station.

The last and final phase, as mentioned before, is the purification phase. This phase corrects the graphs in case of node movement or failure. When a node changes position, it has to calculate its new position and based on the new position, the level will be updated. Because of the graph change, the children of the moving node might become unconnected. So as a consequence of the moving parent node, also the child nodes need new connections to the graph again.

As the TBRP is discussed and explained, we can now look at the performance and the expected performance in the sensor shorts case. In the results from [28], it can be seen that in terms of energy consumption, the TBRP performs the best with more sensor nodes. When there are more nodes, the time till the first node dies is longer. Because with fewer nodes, the nodes are more scattered. More control messages have to be transmitted when sensor nodes are scattered far from each other. That is also why the tree formation phase consumes more energy than the LEACH-Mobile setup phase.

The nodes with lower levels die first because the nodes in the lower levels have more data to transmit, as they also transmit data from higher-level nodes. Another consequence of the network topology from TBRP is that the delay in the higher levels has a higher delay than the data in lower levels. The delay for TBRP is expected to be higher than LEACH-Mobile because, in TBRP, the base station receives the data after the node has received the data of the complete network. A solution for this might be transmitting the received data immediately, but in TBRP, the amount of hops before reaching the base station is higher than LEACH-Mobile. However, TBRP and LEACH-Mobile have the advantage that the route to the central server is known, unlike the Full Flooding and Random Forwarding technique. So there will not be any unnecessary use of resources for finding the path to the base station.

By separating the sensor nodes based on the locations in different levels, the transmit distance of the sensor nodes is approximately the same for all the nodes. However, when all the nodes are closely clustered at a considerable distance from the base station, the transmit distance of the root node is relatively large because it should transmit data to the base station. This would invalidate the use of these levels because now the root node has to transmit the data of all sensor nodes over a considerable distance. Consequently, the energy consumption would be much more for the root node compared, for example, to the Cluster Head in LEACH-Mobile, because in LEACH-Mobile, there are multiple nodes connected to the base stations. In this kind of situation, it might be more efficient if the sensor nodes deliver their data directly to the base station.

The TBRP, as described by Singh et al. does not consider the energy of the sensor nodes when forming the network [28]. Virmani, Sharma, and Sharma proposed an alternative of the TBRP, the Adaptive Energy Aware Data Aggregation Tree (AEDT), which is an alternative to the TBRP [29]. In AEDT, the tree is based on the energy of the nodes. The node with the highest energy is selected as the root node. Using this technique in TBRP, will make TBRP more energy efficient.

A disadvantage for the TBRP is that we need a system to calculate an athlete's position on the field. Due to the extra computations, more power is consumed. Besides, more memory space is needed as the root node processes all the sensor nodes' data.

3.3. Wave propagation

To understand how to get a reliable wireless connection between the athletes and the base station, we should better understand how the radio wave propagates. This can help us choose the suitable communication protocol and Data Collection Protocol. It can also help determine how many base stations we might need to cover the entire football field.

3.3.1. Path Loss

As shown in Figure 2.1, we need a transceiver and an antenna to communicate with the base station. The antenna produces Electromagnetic (EM) waves that the base station's antenna can receive. This EM wave contains the information to be communicated between the two nodes. The medium through which the EM wave travels is also known as the wireless radio channel. In this medium, the communication signals are susceptible to noise, interference, and other channel impediments, but those impediments are changing over time due to user movements [30].

The most simple model for characterizing the received signal power over distance takes the loss of power due to path loss (PL) in regard. This model is called the Path Loss model [30]. Path loss is the dissipation of power due to the transmitter's radiation or as an effect of the propagation channel.

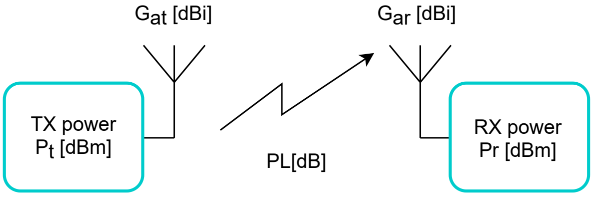


Figure 3.6: Wireless communication system with transmitted power and received signal power P_t and P_r respectively, the gain of the transmitter and receiver is indicated with G_{at} and G_{ar} respectively

For the path loss model, a system is considered with no obstruction, and the signal propagates along a straight line between the transmitter and receiver. This signal is called the line-of-sight (LOS) signal. In Figure 3.6 the signal transmission system is shown for the path loss model. From the figure, we see the antenna gain is indicated with G_{at} and G_{ar}, the unit for antenna gain is dBi. The gain of the transmitting antenna and the receiver antenna is G_{at} and G_{ar}, respectively. In Figure 3.6, P_t and P_r represent the transmitted power and received signal power, respectively. In wireless communication, the unit used for power is typical dBm. dBm indicates a power level in decibels with reference to one milliwatt. The Path loss is indicated with PL in Figure 3.6, the unit for PL is dB.

An antenna consists of electrical conductors designed to radiate an electromagnetic field from alternating current, or vice versa, producing electrical current from electromagnetic waves. The radiation of an antenna depends on the design of the antenna. An antenna which transmits the signal equally in all directions is an isotropic antenna. An isotropic antenna is an ideal antenna which is not feasible [31]. The i in the antenna gain unit dBi stands for isotropic. The gain is the increase of the output signal compared to the output signal of an isotropic antenna. In a real case scenario, the antenna gain is different in every direction.

Now we have seen the important parameters in Figure 3.6 and now what they mean, the Received power (P_r) can be calculated with the following equation:

$$P_r\text{dBm} = P_t\text{dBm} + G_{at}\text{dBi} + G_{ar}\text{dBi} - PL\text{dB}. \tag{3.1}$$

The parameters in Equation 3.1, are the same as in Figure 3.6.

Before Equation 3.1 can be used the value for the path loss (PL) should be calculated. The dissipation of power of a signal depends on the distance the wave has to travel and on the frequency of the signal. This results in the following equation for PL:

$$PL = 20\log_{10}\frac{4\pi d}{\lambda}\text{dB} \tag{3.2}$$

In Equation 3.2, d is the transmission distance and λ is the wave length. The wave length can be calculated using the propagation speed (which is approximately $c = 3 \cdot 10^8[\text{ m/s}]$) and the frequency of the EM wave:

$$\lambda = \frac{c}{f} \tag{3.3}$$

In the above equation, c is the speed of light, and f is the frequency of the EM wave.

Another important aspect of wireless communication is receiver sensitivity. The receiver sensitivity indicates the level above which the received signal power should be. So it is the minimum received signal power that the receiver can correctly receive.

Now we know the important parameters and equation for the free space path loss model, we can combine Equation 3.1 and Equation 3.2 to find the maximum distance between a receiver and transmitter. Combining Equation 3.1 and Equation 3.2 and applying the logarithm rules results in:

$$P_r = P_t + G_{at} + G_{ar} - 20\log_{10}(4\pi) - 20\log_{10}(d) + 20\log_{10}(\lambda). \tag{3.4}$$

From Equation 3.4 the range between two can be determined by rewriting the equation. Rewriting gives the following equation for the range:

$$d = 10^{\left(\frac{\mathrm{P_t - P_r + G_{at} + Gar} - 20\log_{10}(4\pi) + 20\log_{10}(\lambda)}{20}\right)} = 10^{\left(\frac{\mathrm{P_t - P_r + G_{at} + Gar} + 20\log_{10}(\lambda) - 21.98}{20}\right)}. \tag{3.5}$$

3.3.2. Two-Ray model

So when we have chosen the necessary electronics for wireless communication, the maximum transmission distance can be calculated using Equation 3.5, which is derived from the free-space path loss model. The path loss model assumes there are no obstructions and there is one single signal arriving: the LOS signal. However, in a real-life scenario, there are obstruction or surfaces which causes the signal to get absorbed, reflected, scattered and diffracted (Figure 3.7).

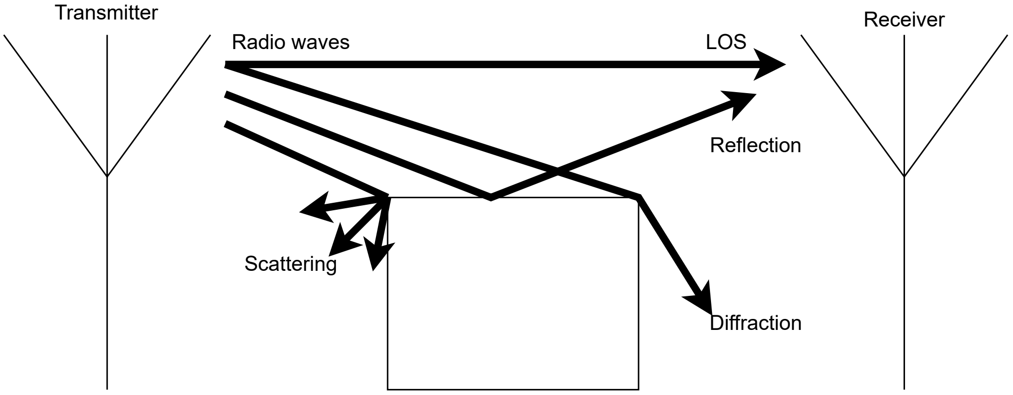


Figure 3.7: Due to obstruction radio waves can get reflected, scattered or diffracted. The wave arriving at the receiver in direct path is the LOS wave.

In Figure 3.7 it can be seen that for example due to reflection there arrives a second signal at the receiver. This signal arrives later than the LOS signal and often it is also out of phase from the LOS signal. Because of the different phases most of the time the reflected signal causes destructive interference with the LOS signal. This phenomena of two or more waves arriving at the receiver is called multipath.

Applying Equation 3.2 on Equation 3.1, it can be seen that there is a logarithmic relationship between the distance and the received signal strength in the case of the path loss model. The Path Loss model was designed under the assumption that the signal is transmitted across clear air and with no reflections. However, as we saw in Figure 3.7, there is always some reflection. For example, there might be reflections on a football field due to the field. Because of multipath, the received signal is often distorted compared to the transmitted signal.

There are various models of different complexities which can describe the wave propagation in the case of reflections. One of the most simple model that can do this is the Two-Ray model [30]. The Two-Ray model assumes that there there is a single ground reflection which dominates the multipath effect. If we only have the transmitter and receiver on a football field, with no players on the field, the Two-Ray can be used to see what behaviour we can expect. Because then the reflection from the field is dominant in the multipath effect. The set-up assumed for the Two-Ray model is shown in Figure 3.8.

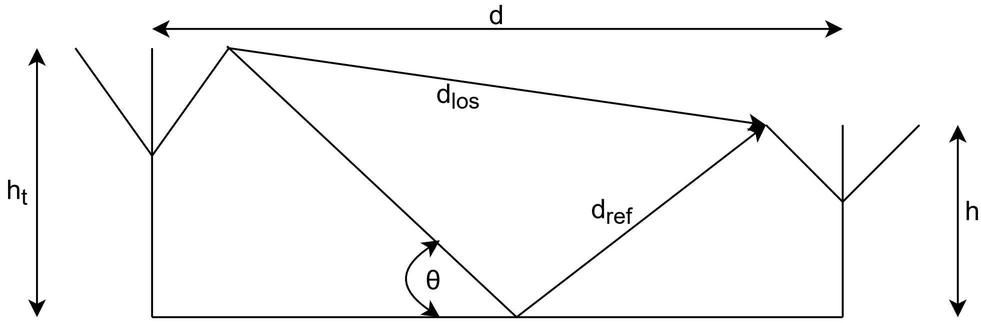


Figure 3.8: The setup in a Two-Ray model

In Figure 3.8 the height of the transmit and receive antenna is noted with h_t and h_r respectively, d_{LOS} and d_{ref} are the distance of LOS signal and the reflected signal respectively. The angle of incident is indicated with the symbol θ. The equation for d_{LOS} and d_{ref} can be geometrically derived [32]:

$$d_{\mathrm{los}} = \sqrt{d^2 + (h_{\mathrm{t}} - h_{\mathrm{r}})^2}, \tag{3.6}$$

$$d_{\mathrm{ref}} = \sqrt{d^2 + (h_{\mathrm{t}} + h_{\mathrm{r}})^2}. \tag{3.7}$$

The phase difference can be calculated using the length of the paths the two signal have to travel and the wavelength:

$$\varphi = 2\pi \frac{d_{\mathrm{los}} - d_{\mathrm{ref}}}{\lambda}. \tag{3.8}$$

The attenuation due to reflection is captured in a reflection coefficient, the reflection coefficient is noted with the symbol $\Gamma_\perp$. The reflection coefficient is dependent on the relative permittivity (ε_{r}) and the incidence angle. Only the sine and cosine of the incidence angle θ are needed for the reflection coefficient. For θ, $\sin\theta = (h_{\mathrm{t}} + h_{\mathrm{r}})/d_{\mathrm{ref}}$ and $\cos\theta = d/d_{\mathrm{ref}}$. Using these, the reflection coefficient can be calculated with the following equation:

$$\Gamma_\perp = \frac{\sin\theta - \sqrt{\varepsilon_{\mathrm{r}} - \cos^2\theta}}{\sin\theta + \sqrt{\varepsilon_{\mathrm{r}} - \cos^2\theta}}. \tag{3.9}$$

Now it is known how to describe the reflected signal this can be added into Equation 3.2, which results in the following PL equation for a Two-Ray model:

$$PL[\mathrm{dB}] = 20\log_{10}\left(4\pi\frac{d}{\lambda}\left|1+\Gamma_{\perp}e^{i\varphi}\right|^{-1}\right) \tag{3.10}$$

So for the Two-Ray model PL is given in Equation 3.10, to calculate the received power for the Two-Ray model Equation 3.1 can be used.

3.3.3. Body shadowing

Attenuation caused due to absorption, reflection, scattering, and diffraction is called shadowing. On a sports field, the body of the surrounding athletes could also cause shadowing. In [33], the effect of body shadowing is analysed. The effect on the PL increases when the human body is closer to the transmitter. A human body between the transmitter and receiver can cause a break in the connection. The effects of body shadowing depend on multiple things, for example, the surrounding, the frequency of the signal, the position of the antennas, etc. Therefore there are various studies analysing different conditions to body shadowing [34, 35, 36].

Januszkiewicz used computer modelling to analyse the human body shadowing [33]. The first analysis compared the free space path loss (no obstructions) and the body shadowing path loss. There was a body between the receiver and transmitter for the body shadowing PL. The results conclude that when the receiving antenna is very close to the human body, the path loss is the largest. With the receiver next to the body it caused an increase of 20 dB or higher depending on the position of the transmit antenna. When the receive antenna is at a more significant distance, the path loss comes close to the free space path loss, there is a difference of around 6 dB. Also, the effect of having more people in between is analysed. It has been shown that having more people increases path loss.

So the variation in the path loss depends on the athletes' position on the field. When two athletes have physical contact, that can cause one athlete to obstruct the other athlete's antenna. In this scenario, the path loss will probably be the highest. When there is some distance between two athletes, the increase in path loss due to body-shadowing will be minimal. However, it can be high because of all the other athletes on the field.

3.4. Communication Protocols

For wireless communication, we are going to need a communication protocol. The choice of the communication protocol has also effect on our system. Because the communication protocol determines, for example, the data rates allowed and the transmit power. Therefore, finding the best communication protocol for our use case is essential. For the choice of the communication protocol, ZigBee WiFi and Bluetooth have been explored. Those three are the most well-known and used communication protocols because they operate in the unlicensed frequency band of 2.4GHz.

Communication protocols determine the Transport, Network and Data Link Layer of the Open System Interconnection (OSI) model [37]. The OSI model defines seven functional layers in a data communication network that describe the different required functions in a data network. Each layer has its protocol to perform its task.

It is interesting to look at the communications protocols in general instead of the effect the communication protocol has on the different layers in the OSI model. The goal is to determine the data rates for a specific communication protocol and how much energy it consumes.

3.4.1. ZigBee

ZigBee is a communication protocol built on the IEEE 802.15.4 standards. ZigBee is simple and low cost, it is also defined as a low power wireless communication technology. According to [38], ZigBee has a consumption of 10 mW. Zigbee is meant for networks with low data rates. ZigBee supports only data rates up to 250 kbit/s. This data rate is enough to get the data of a single athlete in time to the base station. However, this is not enough when multiple athletes are connected to the same base station. The data load will be larger than 250 kbit/s, which will cause increasing delays as every second more data is stacking up. This might also explain the high delays in [25], in [25] ZigBee was used. ZigBee can support multiple different types of topologies like mesh, star and tree [39].

The transceiver used in [25] which supports ZigBee, is MICAz [40]. According to the datasheet, MICAz has a maximum transmit power of 0 dBm and can cover an outdoor distance of 75 m to 100 m. During transmissions at maximum transmit power, the current draw is 17.4 mA. We can use this information to compare the different communication protocols.

3.4.2. Bluetooth

Bluetooth is based on the IEEE 802.15.1 standard. According to literature, unlike ZigBee, Bluetooth is designed for short-range applications [38]. Because it is designed for short-range, we can also expect that the energy consumption might be lower than ZigBee. According to [38] the power consumption is 1 dBm. Like ZigBee, Bluetooth can also support mesh star and tree topologies.

Looking at the products available on the market, some transceivers can cover longer ranges because higher transmit power is possible in new Bluetooth transceivers. For example, the CC256 transceiver series of Texas Instruments [41]. The CC256 series have a transmit power up to 10 dBm and a data rate up to 3 MB/s. Because of the higher transmit power, the CC256 should be able to cover larger distances than the MICAz mote. During transmission, the current draw is 18.5 mA.

3.4.3. WiFi

WiFi is based on the IEEE 802.11 standard. WiFi has the advantage that it has high data rates and can cover long distances [42]. However, this goes at the cost of more energy consumption. WiFi is based on IP addressing [43]. When using WiFi, an Access Point (AP) or router is needed to find the route to the destination IP. With WiFi, it is not possible to have a direct connection with the other athletes, the connection goes via an AP or router. So in WiFi, there is a star topology possible.

We saw before the xPico 240 from LANTRONIX, this is a transceiver which delivers WiFi connectivity [44]. The xPico can deliver a data rate of 150 Mbps, and a transmit power of 17 dBm. When transmitting, the current drawn may be up to 494 mA.

3.5. Conclusion

Based on the literature, it can be concluded that several base stations are needed outside the field. The base stations will collect the data from the athletes and send it to the central server. At the base station, the data will be arriving by wireless communication. For the connection between the base station and the central server, a wired connection is chosen because the communication does not get interfered with by other wireless signals.

Different Data Collection Protocols haven been looked into for getting the data from the athletes to the base station. Table 3.1, shows a general overview of the different Data Collection Protocols and their characteristics. The Data Collection Protocols are characterised by the expected delay, energy consumption, use of resources and complexity. The ratings are done using plus signs. The rating is relative to each other, the one with the worst rating gets a single plus, and the one with the best rating gets four pluses.

Table 3.1: Rating the different DCPs using pluses, the rating is relative to each other

	Delay	Reliability	Energy consumption	Use of resources	Complexity
Direct Delivery	+++	+++	++	++++	++++
Full Flooding routing	++++	++++	+	+	++
Random Forwarding	+++	+++	++	++	+++
LEACH-Mobile	++	++	+++	+++	++
Tree Based Routing Protocol	+	++	++++	++	+

To choose one of the DCPs, first, the communication protocols should be considered. In section 3.4, the communication protocols looked into are Bluetooth, WiFi and ZigBee. We have been looking at the available transceiver of the communications protocols and defined the data rates, energy consumption and the transmit power of those transceivers. In Table 3.2 an overview of the values found are given for the different communication protocols. For the energy consumption, the current drawn by the transceiver during transmission is used because, at that moment, energy consumption is the highest.

Table 3.2: Overview of the maximum data rate, transmit power and current draw during transmission for the communication protocols

	Bluetooth	WiFi	ZigBee
Maximum data rate	3 Mb/s	150 Mb/s	250 kb/s
Maximum TX power	10 dBm	17 dBm	0 dBm
Peak transmit current	18.5 mA	494 mA	17.4 mA

We saw that in the case of sensor shorts using compression, the total load for an entire team is above 250 kb/s. Because ZigBee has a maximum data rate of 250 kb/s, ZigBee is not feasible in the sensor shorts application. Table 3.2 shows that the highest data rate and transmit power comes from WiFi, but this goes at the cost of high energy use. Because during transmissions, the current drawn using WiFi is 494 mA. In the previous prototype of the sensor shorts, a battery supply of 1000 mAh was used. Using this battery together with WiFi, the sensor short should be able to last $1000/494 \approx 2\,hours$. For our use case, an operation time of 2 hours is good enough as we need an operation time of 90 minutes.

In Table 3.2 also the transmit power of the communication protocols is shown. In section 3.3 it is shown using the transmit power and the path loss model, the distance than can be covered can be calculated. The distance that can be covered is calculated using Equation 3.5, the equation depends on the wavelength, the antenna gain and the transmit power. For the considered communication protocols, the wavelength is the same. If identical antennas are also used for the different communication protocols, the communication protocol with the highest transmit power has the highest range. So this means that WiFi will cover the longest distance compared to Bluetooth and ZigBee.

Because WiFi has a high data rate and can cover large distances, WiFi might be an interesting communication protocol. However, as said before, the athletes should be connected to the base station in WiFi because only Direct Delivery is possible. The data can not get via the other sensor shorts to the base station.

Dhamdhere et al. concluded that Direct Delivery has a high delay [25]. However, as said before, this might be caused due to the communication protocol used. The communication protocol used is ZigBee. ZigBee has a low data rate and low transmission range compared to Bluetooth and WiFi. It might be possible by using WiFi to get a much better delay compared to [25].

Besides, in [25] there were, in total, eight base stations used to cover the whole field. In this thesis, we want to minimise the use of the base stations, so that cost and time will be saved when setting up the sensor network. For WiFi, the reach is more extensive compared to ZigBee, which is used in [25].

section 3.3 showed that due to the complexity and variability of the radio channel, it is difficult to obtain an accurate deterministic channel model. In our case, there is a lot of movement and channel variety due to the movement of players. Because of this, it is difficult to get a model which can describe the channel on a football field.

4 Design of a reliable wireless communication connection

4.1. Research questions

The previous section covered the different types of DCP and communication protocols. The choice was made to use WiFi because of the expected gains. It can transmit over long distances and supports high data rates. The chosen DCP is Direct Delivery. When there are enough base stations to cover the entire field, it is expected that using Direct Delivery will result in the lowest possible delay. Besides, it is more reliable and robust because the data transfer is mainly dependent on the base stations and not on the other sensor nodes. Finally, DCP minimises resources as it does not use other sensor nodes.

In order to develop a good system, several research questions have been formulated that will help in this regard. These research questions will be answered during this chapter and through tests in the next chapter.

The questions that will be used as a guideline throughout the design process are listed below.

1. How to create a wireless link between the sensor shorts and the central computer?
2. How to get the data from the sensor shorts to the central computer?

   The first two questions are used to develop a design that can send the data from the sensor shorts to the central computer. With the first question we want to find out what kind of protocols or electronics we needs to have wireless communication between two devices. In the second question we want to find out how the data gets from the sensor shorts to the central server. For example what devices are needed between those two endpoints? How has data to be processed and transferred?

3. Is a single base station able to cover an entire football field?
4. How does a human body affect the received signal strength?
5. Is it better to have the base station at a more elevated position for the large field application?

   The previous questions are to get an idea if one base station is enough to cover an entire field. In section 3.1, it was shown that for five-a-side football, the best height for the base station is the same height as the antenna on the athletes. However, one note was that it might be better to place the base stations slightly higher in a large field. Therefore we want to investigate this.

6. What effect has the received signal strength or/and distance on the data delay, error or loss?

   When a weaker signal is received, it is expected that the quality of the signal will be worse. The weaker signal might affect the delay or packet loss/error. Does it have such an effect that extra base stations are needed?

7. Is it beneficial to use a dual antenna configuration for the sensor shorts?

   Burgers came to the conclusion that the dual antenna configuration provides advantages [8]. Because when one antenna is blocked due to obstruction, the other antenna might provide a better signal. In this thesis, further research is performed on the advantage of the dual antenna configuration.

8. How many sensor nodes can connect to the base station simultaneously?

9. What is the effect of multiple sensor shorts being active at the same time?

   There is research needed on the scalability of the system. Does having more athletes connected to the system affect, for example, the delay or packet loss/error.

10. How much energy is consumed by the sensor shorts when using WiFi and Direct Delivery?

    Indeed, we want to know if our design can last a complete football game, so this question needs to be answered.

4.2. Wireless connectivity of the sensor shorts

This section will look into how a wireless connection is achieved between the sensor short and the central server. First, the connection between the sensor shorts and the base station will be looked into. Second, the connection between the base station and the central server will be discussed. Finally, the end to end connection will be reviewed.

4.2.1. The connection between the transceiver and Central PCB

Figure 2.1 of chapter 2 shows a general design for a wearable monitoring device with a wireless connection. A transceiver has to be connected to the microcontroller. The transceiver has an antenna connected which transmits or receives the wireless signals. This sub section will discuss the connection between the chosen transceiver and the Central PCB.

In section 3.5, the choice was made to use WiFi in combination with Direct Delivery for the wireless communication of the sensor shorts. So the transceiver should support WiFi. Additionally, the transceiver should support a dual antenna configuration because tests will be done with the dual antenna configuration.

The xPico 240 from LANTRONIX supports WiFi with a dual antenna configuration. The xPico 240 is an Embedded WiFi Gateway, with a size of 22 mm × 35.5 mm × 2.73 mm. A gateway is a device that can communicate and send data back and forth. In Figure 4.1 an xPico 240 module is shown. In an attempt to show the scale of the xPico, a coin is placed next to the xPico module.

In Figure 4.1 the two antenna connectors are indicated with an arrow. Burgers had already done research to two types of antennas, the dipole antenna and the patch antenna [8]. From the research it was shown that the patch antenna covers longer ranges. Because of this the patch antenna is also used in this design. The patch antenna used is the W3230 from PulseLarsen Antennas [13]. In Figure 4.1, the two antenna connectors are indicated with an arrow. I

Section 1.1 discussed the design of the electronics on the previous smart sensor shorts prototype. In addition to the previous design, a transceiver is connected to the Central PCB (Figure 4.2). **Figure 4.2** shows the complete design of the electronics on the sensor shorts. A power supply is connected to the Central PCB to keep the system powered. The power supply used is the GNG-109 battery pack, with 2600 mAh at 5 V. The main PCB consists of the microcontroller. This microcontroller runs all the tasks on the sensor shorts. Such as collecting data from the sensors and saving it on the SD card. The data from the three IMUs is sent over the SPI connection to the Central PCB. Before the sensor data are transmitted wirelessly, they must be sent to the xPico. The connection used between the central PCB and xPico is based on the universal asynchronous receiver-transmitter (UART).

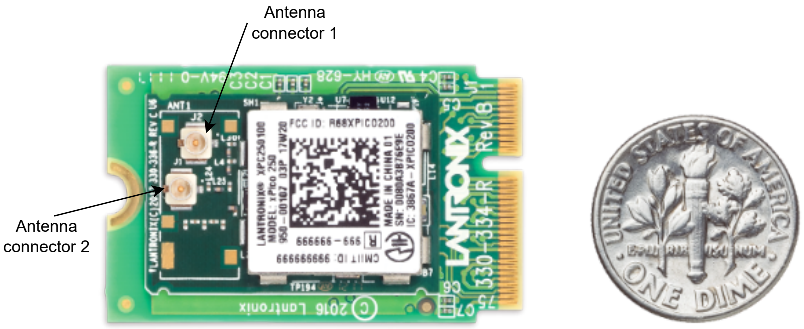


Figure 4.1: The xPico 240, which is used as a transceiver in the smart sensor shorts [44]

The data received on the xPico from the Central PCB is saved in a buffer before transmission. The data in the buffer will be deleted when data is sent to the central server, and the receiver acknowledges the reception. As a result, it might occur that when no acknowledgements are received, the buffer will be packed so that no new data can be written from the central PCB to the xPico. When the buffer is full, and the central PCB continues sending new data to the xPico, this data will be lost. Flow control inside the sensor shorts system should be used to prevent overwhelming the receiver.

Flow control is a method which will tells the central PCB to stop or resume sending data over UART. There are two types of flow control, hardware flow control and software flow control. In hardware flow control, extra wires are used, where the logic level on these wires defines whether the transmitter should keep sending data or stop. Special characters are sent over the standard data lines in software flow control to start or stop the transmission. The special characters are Xon and Xoff. Xon resumes the transmission, and Xoff stops the transmission.

Software flow control checks whether the incoming data on the xPico 240 consists of a Xon/Xoff character. When sensor data happens to be equal to one of the characters, the sensor data will be interpreted as a Xon/Xoff character. This is undesirable, so we opt for the hardware flow control. Another advantage of hardware flow control is that the possible data speeds are much higher. The data rate is set to 4 Mb/s. As mentioned before, extra wires are needed for the UART connection for hardware flow control. In Figure 4.3 a simple overview is shown of the connection between the xPico and the Central PCB.

In Figure 4.3, the data communicated between the xPico and Central PCB goes from the TX (transmit) to RX (receive). Both devices' ground (GND) are connected to have the same reference voltage. The additional wires in Hardware Flow Control are the RTS (Request to Send) and CTS (Clear to Send). When the xPico can receive data, the RTS line of the xPico is asserted. The RTS line is de-assert when the xPico cannot receive more data because its buffer is full. Figure 4.3 shows that the RTS is connected with the CTS. However, the CTS of the xPico is connected to the GND because the data flow from the xPico is minimal, and there are no overflows expected on the Central PCB. By connecting the CTS of the xPico to GND, the data from the xPico is always "Clear to Send".

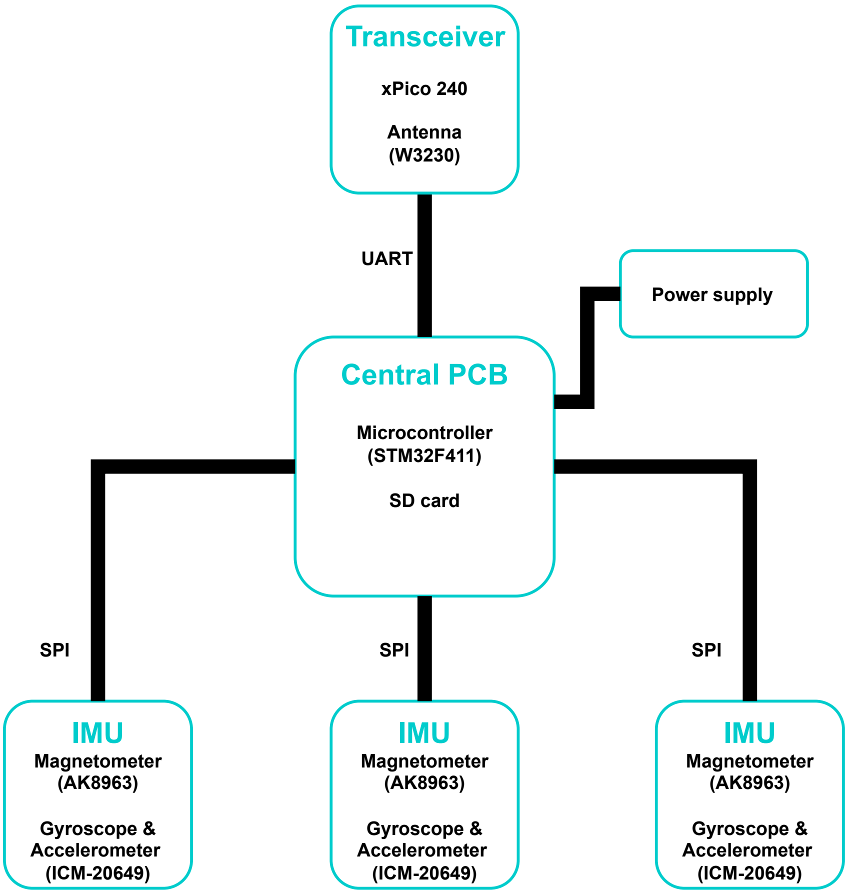


Figure 4.2: Overview of the different hardware units in the sensor shorts and how they are connected

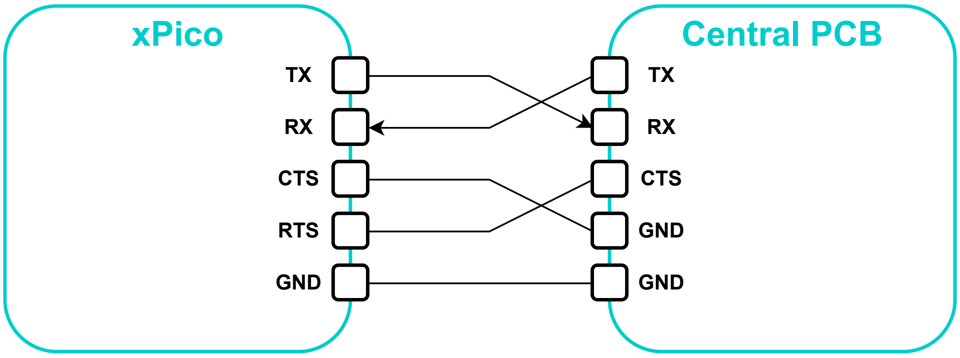


Figure 4.3: A simple overview of the UART connection between the xPico and Central PCB, with the hardware flow control

Figure 4.4 shows a diagram of how the hardware flow control works. Depending on the state of the buffers on the xPico, the CTS of the central PCB has a value of either 1 or 0. Those two values for CTS represent the two states on the Central PCB in which the UART might be. Figure 4.4 shows that the buffer of the xPico being full or not full causes the transition between the two CTS states. The software uses a boolean *PauseUart*, this boolean determines whether data will be sent over UART or not. When the value of CTS is equal to 0, *PauseUart* is set to false, and sensor data is transmitted to the xPico. When CTS is equal to 1, *PauseUart* is true, and no data is sent to the xPico because the buffer has not have enough space to store new data. The figure also shows that no new data is sent to the buffer in the case of a full buffer.

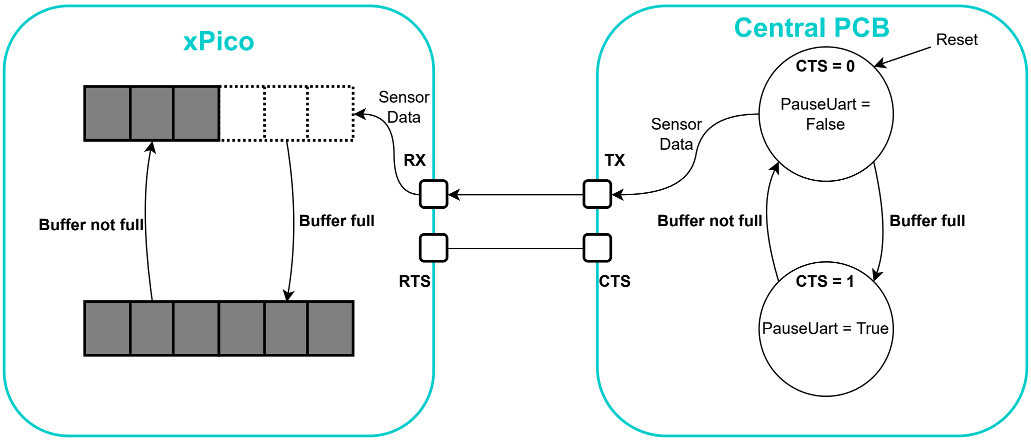


Figure 4.4: Overview of how the hardware flow control works in software

For the CTS on the Central PCB an interrupt pin is used. The interrupt pin calls a callback function when a rising or falling edge is detected. The name of the callback function is *HAL_GPIO_EXTI_Callback*. The system starts with the *PauseUart* being true, as shown in Figure 4.4 by the reset arrow. Because, at the start, the xPico buffer is empty. The xPico controls the value on the CTS pin of the Central PCB by changing the value on the RTS pin. The rising and falling edge triggers the state transition on the Central PCB. In Figure 4.4, transitions are indicated by the "buffer full" and "buffer not full" arrows.

4.2.2. Connection between sensor shorts and base station

From section 3.1, it is shown that before the data from the sensor shorts reaches the central server, it should pass the base station. The wireless link between the sensor shorts and the base station will be discussed in this subsection.

The xPico 240 provides different wireless standards, the provided standards are IEEE 802.11 a/b/g/n. Of the four standards, IEEE 802.11n is the newest and, therefore, the best in terms of data rates. Theoretically, the IEEE 802.11n standard could provide a maximum data rate of 150 Mbps. Because IEEE 802.11n is the most advanced and has the highest maximum speed, we will be using the IEEE 802.11n standard.

The data from the sensor shorts is sent to a base station. The base station should be able to receive data from the sensor shorts and transmit data to the sensor shorts. Indeed the communication should be wireless. Besides, the base station should relay the data from the sensor shorts to the central server. A device which can provide this is an Access Point (AP). Therefore an access point is used as a base station.

A suitable candidate as a base station is the EAP225-Outdoor from TP-Link. The EAP225-Outdoor is one of the cheapest AP on the market. Additionally, it can cover large distances and provide high data rates. The EAP225-outdoor is equipped with two omnidirectional antennas. Omnidirectional antennas have equal radiation in all directions. The benefit of having two antennas is that they can cover larger areas. Another benefit is that it can transmit and receive simultaneously, one antenna will be transmitting, and the other will be receiving.

It has been mentioned before that the wireless connection between the base station and the sensor shorts will be based on the IEEE 802.11n standard. The IEEE 802.11n standard provides functionality that determines the physical and data link layer of the OSI model (Figure 4.5) [45]. In the Data Link layer, the IEEE 802.11n determines the MAC sublayer that interacts with the Logical Link Control layer. As shown in Figure 4.5, all the 802.11 standards share the same MAC protocol, which is the Carrier-Sense Multiple Access with Collision Avoidance (CSMA/CA) [46]. The difference between the standards is mainly in the Physical Layer.

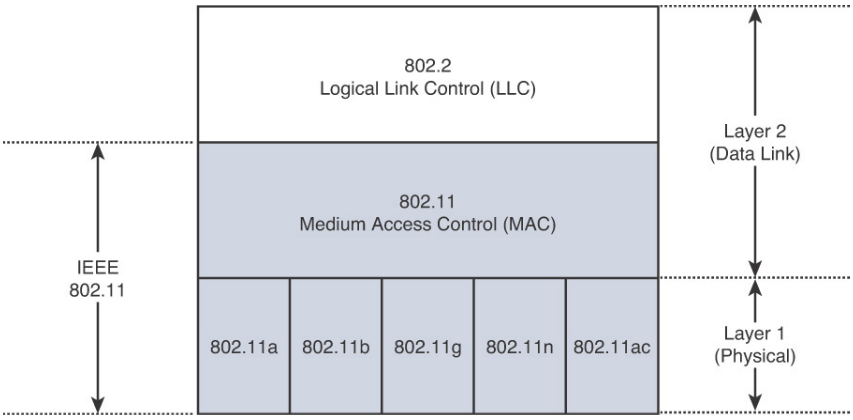


Figure 4.5: IEEE 802.11 standards like IEEE 802.11n standard provide functions for the Data Link and Physical layer of the OSI model [45]

To prevent signals of multiple sensor shorts from interfering, CSMA/CA is used in 802.11n. In CSMA/CA, once one of the sensor shorts wants to transmit, it senses the channel to find out whether there are already sensor shorts transmitting or not. When there are transmissions ongoing, the sensor shorts refrains from transmitting and will try again later on. However, when the sensor shorts senses an idle channel, the sensor shorts should first send a RTS message. The receiver will respond with a CTS message. The sender must refrain from sending and try again later when no CTS is received. This mechanism is introduced to reduce the collision probability.

So, only one sensor short can transmit data at a time. When there is already one sending data, it has to wait till a later moment to transmit. In high data traffic and high density client networks, this might result in clients having to wait for a long time before they can transmit their data. Clients are referred to devices which are connected to the AP, for example, the sensor shorts. According to the datasheet, EAP-225 Outdoor can handle more than 220 concurrent clients. So there are no problems expected when monitoring a complete team. However, because of the possible effect of having multiple sensor shorts connected, we will look at the effects we see when multiple shorts are monitoring.

In Figure 4.5, it is shown that the other sub-layer of layer 2 is the LLC layer. In wireless communication, there is a greater chance of receiving incorrect data. Incorrect data might be caused due to interference from other signals. The LLC layer tries to prevent errors in the data [46]. At the receiver, it is checked whether the message has been received correctly. If this is the case, an acknowledgement (ACK) is sent to the sender. If nothing is sent back, the sender will transmit its data again later.

The advancement of the 802.11n standard over the older standards is that it can use Multiple Input Multiple Output (MIMO) technology. subsection 3.3.2 discussed that multipath occurs due to reflections, scattering, and diffraction. These signals arrive out of phase and then interfere with one another. MIMO systems employ multiple antennas to use these reflected signals as additional simultaneous transmission channels. In short, MIMO knits the disparate signals together to produce a single, more robust signal. Both ends of the wireless link should support MIMO to use this technology. Unfortunately, this technology can not be used on the xPico 240 because it supports only one transmission or reception at a time.

The main difference in the different WiFi standards is in the Physical Layer (Figure 4.5). The Physical Layer is responsible for transmitting the data over the wireless medium, in the case of WiFi. The general operations of the Physical Layer include carrier sense, transmit and receive functions. Each of the standards performs these functions differently. One different aspect is the modulation of binary messages to wave signals. Over time, modulation techniques and hardware have been developed, due to these developments, high data rates are achievable in 802.11n. As mentioned before, due to the modulation scheme used in 802.11n, data rates up to 150 Mb/s can be achieved.

The Physical Layer determines also the band in which the signals have to be transmitted. WiFi is based in the 2.4 GHz band, this band is split in 13 different channels (Figure 4.6). The bandwidth of the channels are 20 Mhz.

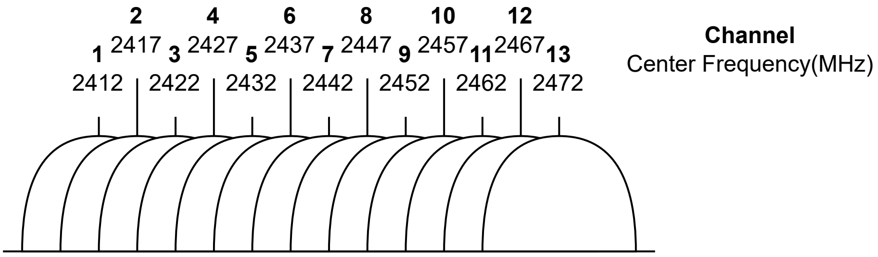
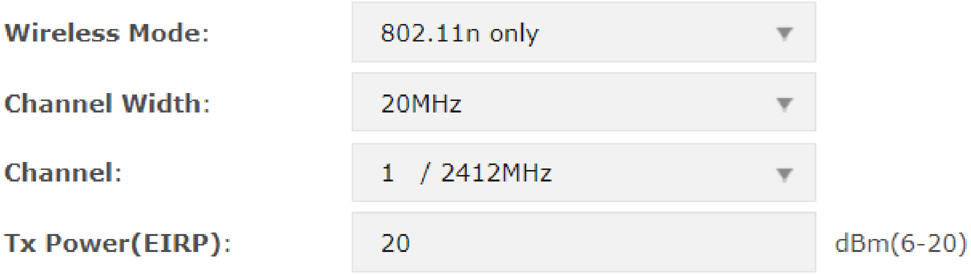


Figure 4.6: The 13 channels in the 2.4 GHz band that is used for WiFi, with a bandwidth 20 Mhz

Figure 4.6 shows that there is an overlap between the channels. For example, if a neighbouring network operates in channel two and our network operates in channel three, our network will experience interference from the neighbouring network. Channels 1 and 13 will have the least overlap with neighbouring channels. Because of this, the choice is made to have our system operate in channel 1. Figure 4.7 concludes the settings of the AP. The transmit power is set to its maximum, which is 20 dBm.



Wireless Mode:	802.11n only ▼	
Channel Width:	20MHz ▼	
Channel:	1 / 2412MHz ▼	
Tx Power(EIRP):	20	dBm(6-20)

Figure 4.7: The settings of the AP for the wireless connection

For the AP, there is one last thing to consider: the orientation of the antenna. According to the datasheet of the AP, the best orientation for the antennas is to have it in an angle of 45 degrees as shown in Figure 4.8 [47]. This configuration is used to cover the largest possible area with the antennas.

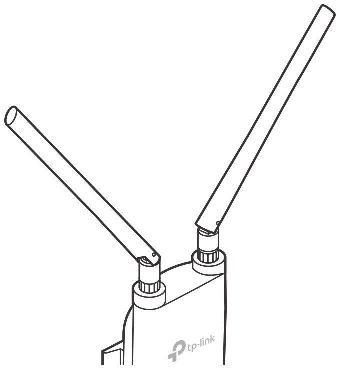


Figure 4.8: The configuration of the EAP225-Outdoor antenna in 45° angle [47]

4.2.3. Connection between base station and central server

An access point provides wireless connections for the sensor shorts, but a wired connection is preferred between the central server and the base stations. Wired connections are more reliable than wireless connections because wireless networks are subject to interference which can distort the wireless signal and cause misinterpretation of the signal [48].

A router is needed to interconnect the wireless network to the wired network. The router used is the TL-R470T+ from TP Link. The TL-R470T+ router provides a data rate of 100 Mb/s.

Routers operate on the Network Layer of the OSI model. The protocol used by WiFi in the Network Layer is the Internet Protocol (IP) [49]. The Network Layer is responsible for delivering the data to its intended destination. The IP assigns an IP address to each node. The IP address consists of four bytes, and it is denoted as $x.y.z.w$ where $x, y, z, w \in [0, 255]$. In the Network Layer, one of the things added to the data is the destination IP address. The destination IP address can be used in the Network Layer to find the destination node.

The router is responsible for assigning an IP address to the nodes and AP in our network. Assigning the IP address can be done statically or dynamically via the DHCP server. For the sensor shorts, the IP address is set dynamically, and the IP address of the central server is set statically. The following subsection will explain why having a static IP address for the central server is helpful.

4.2.4. End to end connection

The end to end connection refers to the connection between the sensor shorts and the central server or computer. The end to end connection is described in the fourth layer of the OSI model, the Transport Layer. The two primary protocol of the Transport Layer is the User Datagram Protocol (UDP), and the Transmission Control Protocol (TCP)[49]. For the design, a choice between the two has to be made. Therefore both protocols will be reviewed, and a decision will be made

The transmission of a data set from one computer to another is not sent as one piece. It is sent in pieces, called frames. The receiving computer puts those frames together. Some frames could be lost during transmission. Lost frames indeed result in having an incomplete data set at the receiver. What to do when this happens is described by TCP and UDP.

TCP provides reliable data transfer. When there is a frame missing, TCP will recover the missing frame. TCP is a connection-oriented protocol. This means that a connection between the nodes has to be established to communicate between two nodes. When a connection is established, the transmission of data can start. In TCP, when frames are missing, or there are errors, the receiver will request the transmitter to resend the data.

UDP, in contrast to TCP, provides unreliable data transfer. UDP does not check for missing data or errors. When using UDP, data might get lost or contain errors. The receiver will not request the transmitter to resend the data.

By looking at the characteristics of both protocols and the requirements in subsection 1.2.1, TCP is the most obvious to use. Because we ideally want no data losses, and TCP provides this. However, a disadvantage of TCP is that it may result in delays due to the guarantee of reliable communication. Because when data is transmitted and does not arrive correctly, it has to be transmitted again until it is received correctly. New data has to wait till the receiver acknowledges previous data.

So when using TCP, it has to be taken in mind that it might result in more latency. Latency could be prevented by assuring a strong connection over the entire field. Having more base stations could ensure a strong connection over the entire field. In the next chapter, we will look into the delays and the amount of base station that is needed.

The TCP connection between the central computer and sensor shorts is established in the web interface of the xPico. To establish a connection, the sensor shorts that initiate the connection need to know the IP address and port it has to connect to. To keep the IP address constant for the server, it got an IP address assigned statically.

In terms of hardware, the end to end connection is shown in Figure 4.9. The figure shows a wireless connection between the sensor shorts and the access point. The access point is connected to the router, and the router also has an Ethernet connection with the server. In our case, the server is a computer.

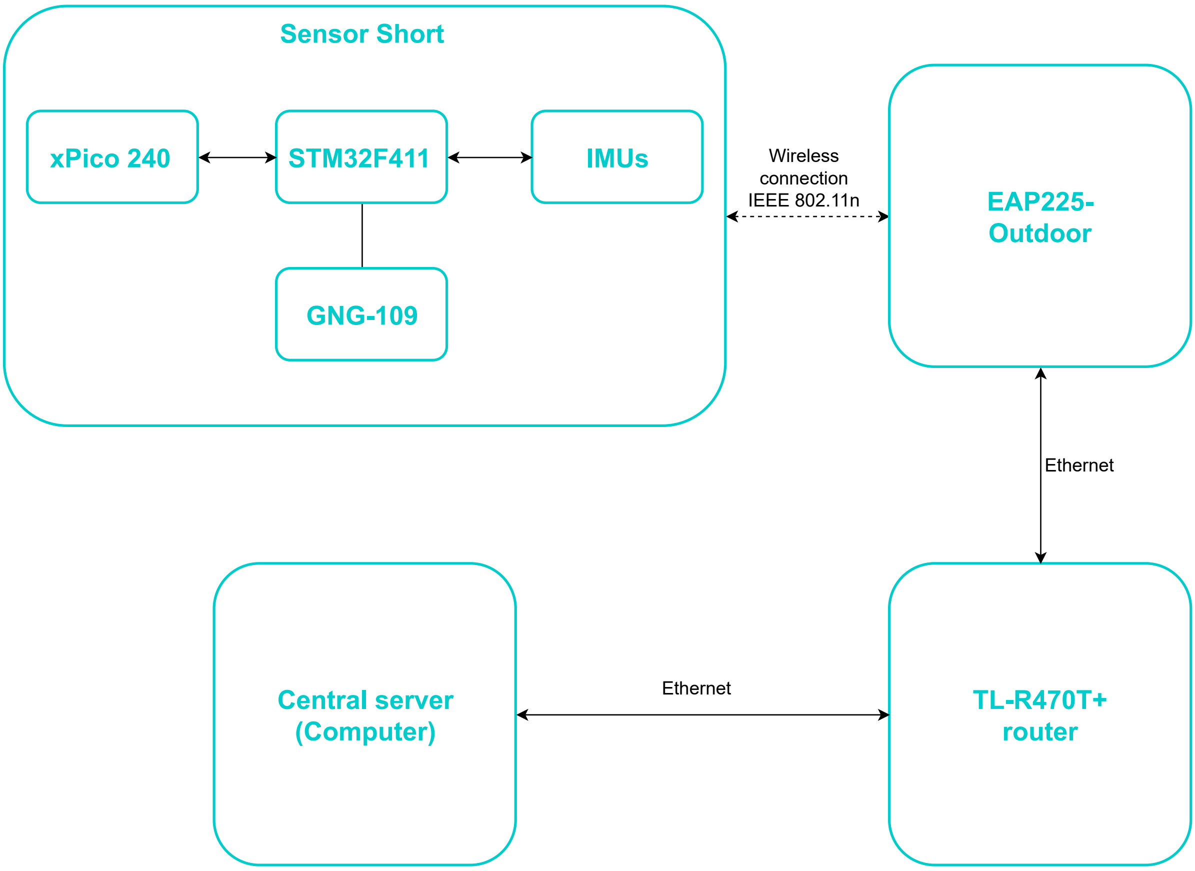


Figure 4.9: A schematic overview of what the end-to-end connection looks like, with all the hardware used.

4.3. Implementation of the data delivery procedure

This section is focused on the software of the data delivery procedure. It will explain how the software takes care of the data transmission and reception. The software implementation can be divided into two parts, software for the sensor shorts and software for the server, running on a computer. First, the software on the sensor shorts will be discussed, and in subsection 4.3.2 the software for the server will be discussed.

4.3.1. Software sensor shorts

For the software of the sensor shorts, the following programs are used, Atollic TrueStudio, FreeRTOS, and STM32CubeMX. Atollic TrueStudio provides the environment to write, build and debug the software for the sensor shorts. The code for the sensor shorts is written in C++. FreeRTOS has already been mentioned in section 1.1, it is a real-time operating system which schedules and executes the tasks in a system. STM32CubeMX is used to configure the in- and outputs of the microcontroller.

In FreeRTOS, the code consists of multiple components or tasks, a function represents those tasks. Each task has a frequency so that the tasks will be executed periodically. A timer is used to trigger the tasks at the indicated frequency. Besides, each task gets a particular priority assigned. In case multiple tasks are triggered at the same frequency, tasks with higher priority will be executed first. If a low priority task is running and a higher priority task is triggered. The lower priority task will be paused, and the higher priority task will be executed. When the higher priority task has finished, the lower priority task will resume its execution. This algorithm is also known as the preemptive priority scheduling algorithm. Preemptive refers to pausing a task with lower priority to execute the higher priority task.

In section 1.1 we saw the tasks *readAccGyroTask*, *readMagTask*, *storeBuffTask*, *storeSDTask*, *initTask*, *dataCompression*. We continue using these tasks except for the *dataCompression* task. This task was created to be able to handle the high data rates. However, because we are using WiFi, the data rates from the sensor shorts are not expected to cause any problems. We are going to test if this is indeed the case. In addition to these tasks a new task has been created:

- *wifiModuleTask*: this task checks whether the UART connection with the xPico can be used or not. If data can be sent over UART, it sends the sensor data to the xPico with a speed of 4 Mb/s.

An overview of the tasks and their priorities is shown in Table 4.1. The priorities of the tasks are rated from 4 to 0, with 4 being the highest priority. The tasks (*readAccGyroTask* and *readMagTask*) which read out the sensor values have the two highest priorities because these tasks are the most important. The *readAccGyroTask* has priority 4 and *readMagTask* has priority 3. This was chosen so that the *readAccGyroTask* is executed first and eadMagTask second when activated simultaneously. Because the *readAccGyroTask* also saves the timestamp of readout, when having equal priorities, there might be irregularities.

After the sensor values have been read, the data must be stored in the buffer. To make sure that this happens correct, the task *storeBuffTask* has the highest priority after the sensor readout tasks. Because the accelerometer and gyroscope are sampled at 250 Hz, *storeBuffTask* has an activation frequency of 250 Hz. The activation will then be as fast as the readout.

The *initTask* is set on a frequency of 5 Hz. In the previous prototypes, this appeared to be high enough to handle the user input. Because this task is less critical than the previous tasks, it has priority 1. The *wifiModuleTask* has the same priority because it is less critical than the tasks with the higher priorities. In the ideal case, *wifiModuleTask* has to be executed only once every 60 ms. Later on, it will be shown that the sensor data is sent as packets of 15 samples to the xPico. It takes 60 ms to collect 15 samples on the microcontroller.

In Table 4.1 for *storeSDTask* and *wifiModuleTask* the listed frequency, is event-based. An event-based task is not periodic but depends on certain conditions. The *wifiModuleTask* for example, depends on if the UART buffer is full and if there are no ongoing transmissions. More about the transmission of the UART buffer will be explained later on in this section.

The *storeSDTask* is the least critical task. That is why it has the lowest priority. If data needs to be saved on the SD, it uses the time that no tasks are performed to save data on the SD.

Table 4.1: The different tasks with their priorities and frequency.

Task	Priority	Frequency
initTask	1	5 Hz
readAccGyroTask	4	250 Hz
readMagTask	3	100 Hz
storeBuffTask	2	250Hz
storeSDTask	0	Event-based
wifiModuleTask	1	Event-based

The data that has to be sent over UART has first to be saved in a buffer. The task *storeBuffTask* was previously used to save the data in a buffer for saving it on the SD. Now there is an extension to this task because it also executes the function *FillWifiBuffer*. This function converts the data to 8 bits and stores it in a buffer which is used for the UART transmission.

For the buffer which will be used for the UART transmission, a structure is made called *UartPing* and *UartPong*. Those structures consist of the following group of variables:

- txdata: is the buffer that will contain the data sent over UART. It can store up to 500 sensor data samples.
- halffull: is a Boolean, set to true when txdata is filled with 15 samples.
- full: is a Boolean, set to true when txdata is full.

- init: is a Boolean, set to true when the structure is initialised.
- transmitted: is a Boolean that is set to true when the data is transmitted over UART.
- bytesFilled: is a variable used to track how many bytes of txdata are filled.

UartPing and *UartPong* are so-called Ping-Pong buffers. Ping-Pong buffering is an effective way to increase the data transfer rate. When one of the buffers is being transmitted, the other buffer is used to store data.

The sensor data is saved within *txdata*, in *txdata* there is place for 500 samples. One sample consists of values, from the accelerometer, gyroscope and magnetometer of the three-axis, for all the IMUs. After each sample is read out, the function *FillWifiBuffer* will be executed. The boolean *fillping* is used to indicate which buffer has to be filled. If this boolean is true, the *UartPing* is used to store data.

In Figure 4.10 a state diagram for the function *FillWifiBuffer* is shown. The function starts initialising the *txdata* of *UartPing* and *UartPong*. Each sample is enclosed in a predefined packet to distinguish between different data samples. Those packets have a total size of 59 bytes. Three bytes are used for the encapsulation. The three bytes are the first two and the last element of the packet. The encapsulation used is:

"{D ... }"

In the place of the dots, the sensor data is placed. Besides, we are also sending a sequence number consisting of two bytes. The sequence number will be used to check if the data is received in the correct order. This sequence number might be left out at a later stage. A data packet can now be distinguished by its encapsulation and packet length. In the initialisation state of *FillWifiBuffer* the encapsulation for the data is placed at the right place.

The program can start saving the actual measured data when the initialisation is done. Because *fillping* is true at the start, the first buffer that will be filled is the *UartPing*. It is decided to send 15 samples at a time to the xPico, this is a randomly chosen number, but the size of the xPico's buffer was considered. The total size of 15 samples is 885 bytes, which is less than the buffer size on the xPico.

The filling of the first 15 space in the *UartPing* and *UartPong* buffer is shown in Figure 4.10 as "Filling Ping buffer 1" and "Filling Ping buffer 2" respectively. When for example, the program is in "Filling Ping buffer 1" and the first 15 data entries are filled, the buffer is half full. Depending on whether the *UartPong* is free to use, the program goes to "Filling Pong Buffer 1". The *UartPong* is free to use in the case *fillping* is False and *UartPong* is not being transmitted i.e is false.

It might happen that *UartPing* is half full but *UartPong* is being transmitted. In this case, new data can not be saved on *UartPong*, i.e. it is not possible to go from "Filling Ping buffer 1" to "Filling Pong Buffer 1" as shown in Figure 4.10. For this reason, there is space for 500 samples in the buffer instead of only 15. Because we have space for 500 samples in the buffer, we can use this space for such situations. Saving the data in the additional space is represented by the states "Filling Ping buffer 2" to "Filling Pong Buffer 2" in Figure 4.10. If the program is in the "Filling Ping buffer 2" state and the *UartPong* buffer becomes available, the program goes into the "Filling Pong Buffer 1" state.

Another situation that might occur is that the complete *UartPing* buffer becomes filled, while the *UartPong* buffer is still not available. In this case, the program goes from the "Filling Ping buffer 2" state into the "Ping buffer full" (Figure 4.10). This state will cause data losses because the *UartPing* buffer is full, and the *UartPong* buffer is being used. Therefore new data cannot be saved. When the *UartPong* buffer becomes available while the program is in "Ping buffer full", the program goes into the "Filling Pong Buffer 1" state.

Figure 4.10 shows that in the "Ping buffer full" state the *UartPong* is at least half full. But the *UartPong* buffer might also be full.

In Figure 4.10 some of the transitions to "Filling Ping buffer 1" or "Filling Pong Buffer 1" do not have the transition conditions listed. For the transitions to "Filling Ping buffer 1" or "Filling Pong Buffer 1", all transitions have the same condition. The condition listed means that when the filling buffer is half full, and the other buffer is free to use, fill the other buffer so that the filling buffer can be transmitted to the xPico.

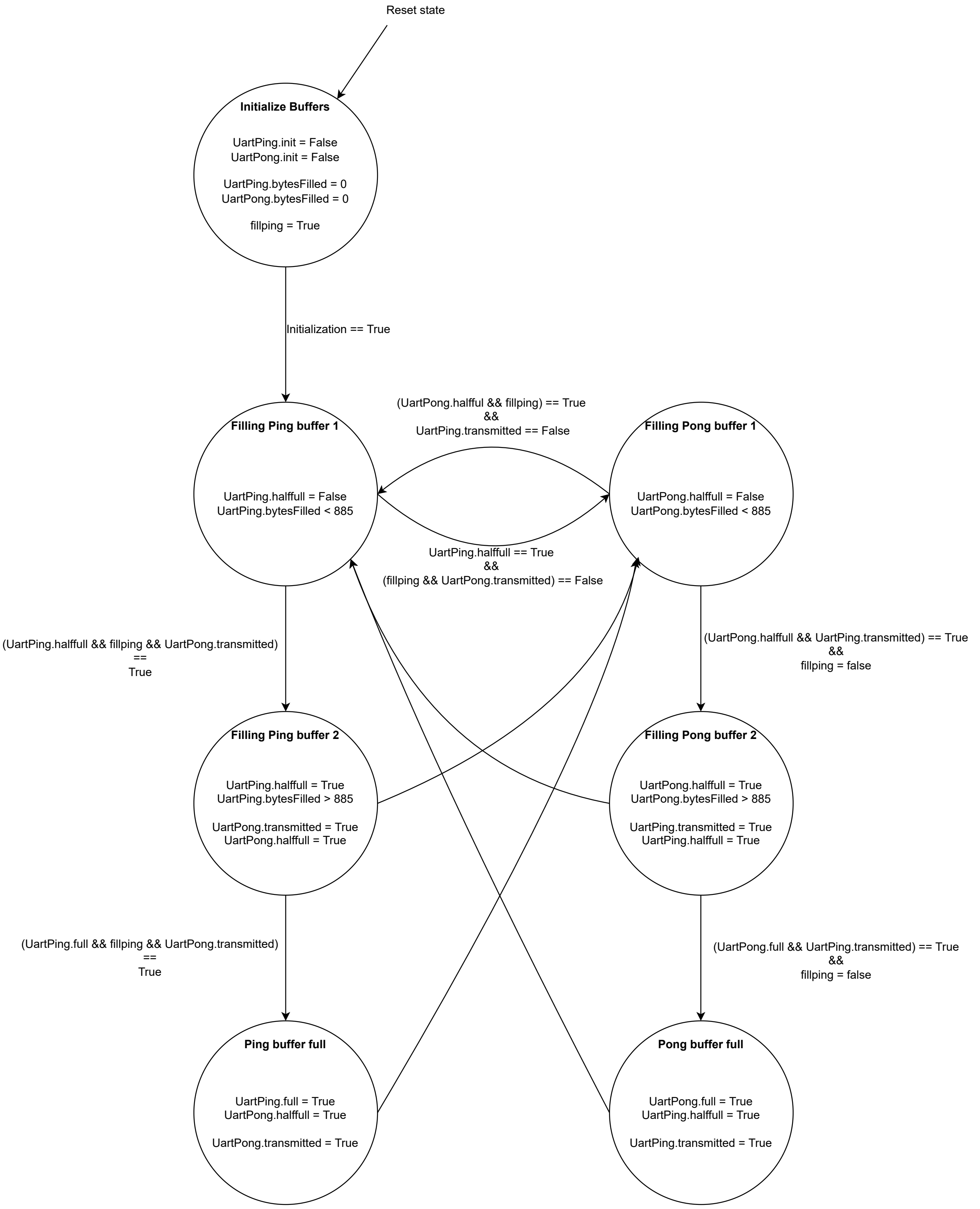

Reset state
Initialize Buffers
UartPing.init = False
UartPong.init = False
UartPing.bytesFilled = 0
UartPong.bytesFilled = 0
fillping = True
Initialization == True
Filling Ping buffer 1
UartPing.halffull = False
UartPing.bytesFilled < 885
Filling Pong buffer 1
UartPong.halffull = False
UartPong.bytesFilled < 885
(UartPong.halfful && fillping) == True
&&
UartPing.transmitted == False
UartPing.halffull == True
&&
(fillping && UartPong.transmitted) == False
(UartPing.halffull && fillping && UartPong.transmitted)
==
True
(UartPong.halffull && UartPing.transmitted) == True
&&
fillping = false
Filling Ping buffer 2
UartPing.halffull = True
UartPing.bytesFilled > 885
UartPong.transmitted = True
UartPong.halffull = True
Filling Pong buffer 2
UartPong.halffull = True
UartPong.bytesFilled > 885
UartPing.transmitted = True
UartPing.halffull = True
(UartPing.full && fillping && UartPong.transmitted)
==
True
(UartPong.full && UartPing.transmitted) == True
&&
fillping = false
Ping buffer full
UartPing.full = True
UartPong.halffull = True
UartPong.transmitted = True
Pong buffer full
UartPong.full = True
UartPing.halffull = True
UartPing.transmitted = True

Figure 4.10: State diagram of the function *FillWifiBuffer*, which saves data in the *UartPing* and *UartPong* buffer.

When the data is saved in the UART buffer, and the buffer is ready to be sent over UART, the *wifiModuleTask* comes into action. Therefore the next thing we will look into is the *wifiModuleTask* function. This task does the actual transmission of the data to the xPico. Another responsibility of the *wifiModuleTask* is the correct alternation of the UART buffer.

In Algorithm 1 the pseudocode of *wifiModuleTask* is shown. In line 1 of Algorithm 1 it is shown that every time *wifiModuleTask* is executed we check if there is something to be received.

The microcontroller communicates over UART using the interrupt functions. The advantage of the UART interrupt function is that the reception or transmission happens in the background so that the system can continue with the other tasks. Once the UART transmission/reception is finished, the code is interrupted, and a callback function is called. For transmission and receiving, there are separate callback functions.

In line 2, the program checks whether the measurements have started and if there is no stop message received. The stop message is a message from the central server, which tells the sensor shorts to stop measuring. Line 2 also checks whether there is an ongoing transmission. That is done by using the variable *opentx*, which keeps track of the "open transmission". With open transmissions, we mean transmissions that have not yet been finished. When there is no open transmission *opentx* is zero, in this case, data transmission over UART is allowed. The last condition that is checked on line 2 has been discussed before. The program needs to check whether it can transmit over UART according to the hardware flow control.

In lines 3 and 7, which of the two buffers must be transmitted is checked. When a buffer is half full, i.e., filled with at least 15 samples, and has not been transmitted yet, the buffer will be transmitted.

In line 4, it is made sure that the new data will now be saved in the *UartPong* buffer by making *fillping* false. Because in line 6, the *UartPing* will be transmitted and is thus not available anymore for saving new data. In line 8, the reverse happens as it is transmitting *UartPong* in line 10.

In line 5 and line 9, we add a 1 to *opentx* because there will be one ongoing transmission in the next line. Once a UART transmission has been completed, a callback function will be called. In this callback function the *opentx* value is decreased. In the callback function, the buffer that was transmitted will be reset.

Algorithm 1 An algorithm with caption

```
 1: UART receive
 2: if Measurment Started and no stop_messagge received and opentx is 0 and PauseUart is False then
 3:     if UartPing.halffull is True and UartPing.transmitted is False then
 4:         $fillping \leftarrow False$
 5:         $opentx \leftarrow opentx + 1$
 6:         UART Transmit UartPing.txdata
 7:     else if UartPong.halffull is True and UartPong.transmitted is False then
 8:         $fillping \leftarrow True$
 9:         $opentx \leftarrow opentx + 1$
10:         UART Transmit UartPong.txdata
11:     end if
12: end if
```

4.3.2. Software server

The data from the sensor shorts will be sent to a computer. This computer is called the central server in the network. The server should be able to receive data from multiple sensor shorts and save this data for future use. These services are provided by a program which is written in C++. In this section, the software for the server will be explained.

Between the server and sensor shorts, different types of messages are exchanged. There are pre-arranged communication agreements made to distinguish between the different messages. There are five different messages which are listed in Table 4.2. All messages are enclosed within the opening ("{") and closing semicolon ("}"). The first message of Table 4.2 was discussed in the previous section. This message contains a "D" to notify the server, which receives this message that the packet contains sensor data. The dots represent the sensor data.

Table 4.2: The pre-defined messages that are used for the communication between the sensor shorts and server.

	Message	Receiver
1.	{D...}	Server
2.	{C}	Sensor shorts
3.	{I...}	Server
4.	{S}	Sensor Short
5.	{E}	Both

Number two of Table 4.2 is sent to the sensor shorts. This is sent to the sensor shorts once a connection is established between the server and shorts. It is used to notify the sensor shorts that it is connected to the server.

We want to be able to distinguish between the different sensor shorts. Since the STM32F411 microcontroller has a unique ID stored in its registers, this ID is used to identify the sensor shorts. The ID of the microcontroller is linked to the short. When a sensor short receives message two of Table 4.2, it responds by sending its ID to the server. The format of this response is shown by number 3 of Table 4.2, the "I" indicates that the data after the "I" is the ID.

Indeed the sensor shorts do not immediately start measuring and sending data to the server. A measurement starts by transmitting message four from Table 4.2 to the sensor shorts. The measurement ends when the sensor short receives message five from the sensor shorts. When a sensor short has received this message, it responds with the same message. This is done to prevent the server from waiting for new sensor data.

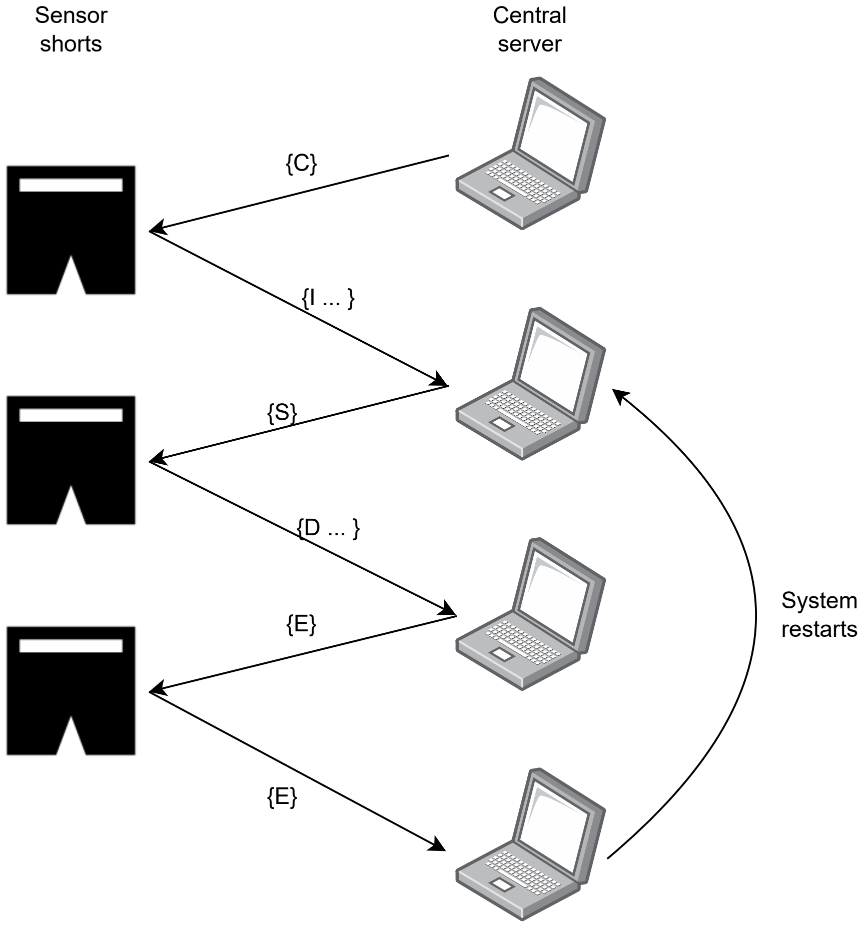


Figure 4.11: Overview of the messages sent between the sensor shorts and server.

To conclude the communication between the server and the sensor shorts, Figure 4.11 shows an overview of the communication between the server and the sensor shorts. When a measurement ends, the system restarts and a new measurement can be started.

Now that the types of messages that could be exchanged between the server and sensor shorts have been discussed. We will look into the actual software for the server. Figure 4.12 describes a state diagram of the implemented software.

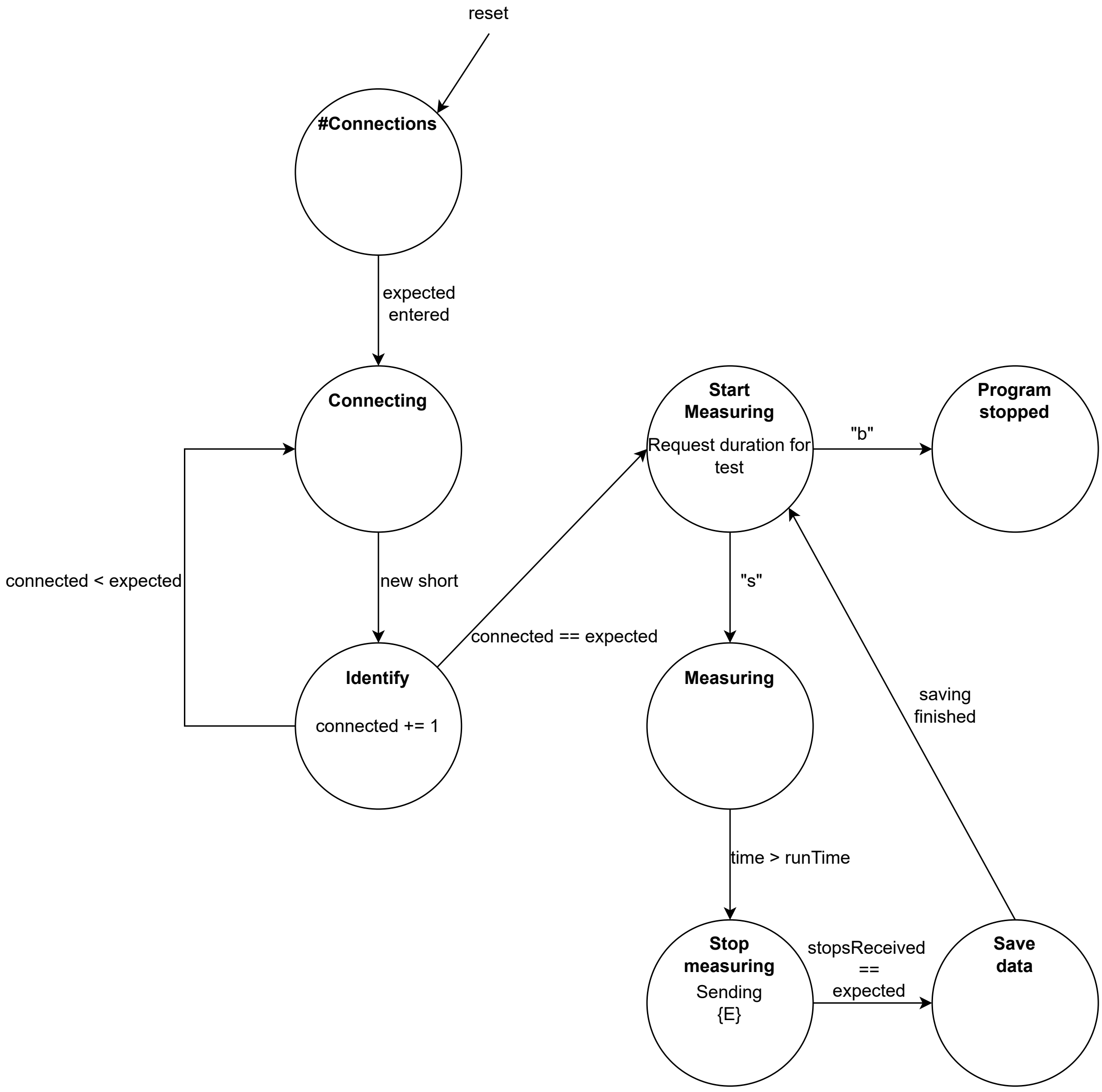


Figure 4.12: State diagram of the server.

The program starts in the "#Connections" state. In this state, the program requires input from the user. The server wants to know how many sensor shorts will be connected. The number of sensor shorts that will be connecting to the server is saved in *expectedShorts*, in Figure 4.12 shown as *expected*. When the user enters the number of expected sensor shorts, the program goes to the next state.

In the Connecting state, the server listens for a TCP connection. Listening, in this case, means it checks whether there are devices in the network that want to connect to the server. If it finds a new device that wants to connect, the identification state is next. In the Identify state, it sends message two from Table 4.2 to the new short. The server should then receive the sensor shorts ID. The server couples the corresponding short number to a new socket. Sockets are the endpoints of wireless communication.

In the software, there is a variable *connectedShorts*, in Figure 4.12 shown as connected, which keeps track of the amount of shorts that are connected. If this number is not equal to the expected amount, the server goes back to the "Connecting" state.

If all the sensor shorts are connected, the server goes from "Identify" to the "Start measuring" state. In the "Start measuring" state, a measurement can be started. This state requires two inputs from the user. One is the time duration of the measurement. The entered time duration is stored in *runTime*. The second input is an "s" or "b" character. The "b" character can be entered to break or stop the program. With the "s" character, the program starts measuring.

While the server is in the "Measuring" state, the server receives sensor data from the sensor shorts. To be able to handle multiple clients, the *select* [50] function is used. The function select is provided with a list of sockets of the sensor shorts. The server checks the status of the sockets in the list. A flag is set for a specific socket when the selection function sees a socket has received data. Based on the socket's flags, the program knows that there is new data from a certain socket.

In the "Measuring" state, besides receiving data, the program also keeps track of the time. When the "time" the program is running exceeds the *runTime* set, the program goes into the "Stop measuring" state. In this state, message five from Table 4.2 is sent to the sensor shorts. As shown in Figure 4.11 the sensor shorts confirm by replying with the same message. The server waits till it has received a stop message from all the shorts. The variable *stopsReceived* keeps track of how many stops message has been received. For now, this implementation was good enough. However, this might not be ideal for the actual deployment of the sensor shorts. Because when a sensor short becomes inactive, the system gets stuck because the system can not communicate anymore with the inactive sensor short.

When a stop message has been received from all sensor shorts, the data received from the sensor shorts will be saved in the "Save data" state. The data is saved in a separate Excel file for every sensor short. When the data is stored in the Excel file, a new measurement can be started.

In the states "Identify", "Measuring", and "Stop measuring" (Figure 4.12), data from the sensor shorts is being received. In each state, another type of data is received. As mentioned before, when the *select* function detects activity on one of the sockets, a flag will be set for the specific socket.

After the flag is set, the data will be read and stored in a buffer. When the data is stored in the buffer, the function *interpret_message_received* is called. As the function says, it will interpret the message received, i.e. it will look at what type of message is received. The function is provided with the socket number to know what socket it has to handle. Based on the formats shown in Table 4.2 and the size each message has, the function can determine what kind of message it is receiving.

4.4. Lab validation

Now that design is finished, the validation is next on the list. The lab validation checks whether the design behaves as expected, and we confirm whether everything is working.

First, we will be looking at the scheduling of the tasks and the chosen priorities. The execution and planning of the tasks have been analysed to prove that all tasks can be executed properly. First, the execution time of the different tasks has been analysed. The results of these tests are shown in Table 4.3. The execution time is determined by using *xTaskGetTickCount* to get the start and end time of a task. With *xTaskGetTickCount* function, the resolution that can be measured is 1 ms. Table 4.3 shows that the execution time of the tasks is in the range of the quantisation steps. Therefore we can not exactly determine the execution time. The execution time is rounded up, which gives an estimation of the maximum execution time. Table 4.3 shows that the task with the largest execution task is *storeSDTask*, it takes up to 11 ms.

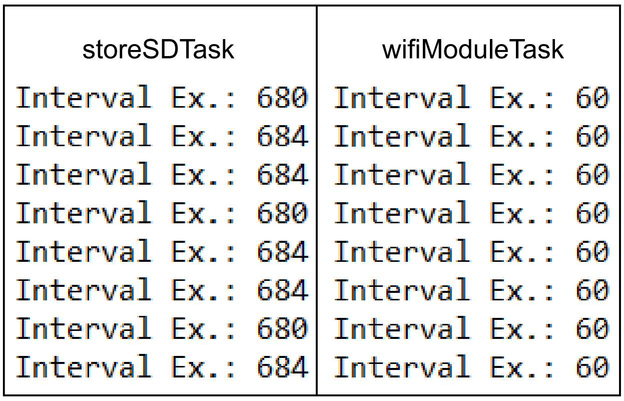
Table 4.3: Maximum execution time of each task and overview of which tasks are preempted by a specific task.

Task	Execution Time[ms]	Preempted
initTask	1	storeSDTask
readAccGyroTask	1	storeSDTask
readMagTask	2	storeSDTask
storeBuffTask	1	storeSDTask
storeSDTask	11	-
wifiModuleTask	1	storeSDTask

Due to the relatively long execution time of *storeSDTask* it is obvious that this task will be preempted. In Table 4.3 it is shown that *storeSDTask* is preempted by all the other tasks. This is, of course, one of the reasons that it has the largest maximum execution time. Additionally, writing data to the SD card is another reason. Nevertheless, the relatively large execution time is no problem because for a buffer of 16 384 B, where each stored sample is 96 B, the buffer stores around 170 samples. Those 170 samples represent $170 \cdot 4 = 680ms$ of data. So around every 680 ms a buffer is full and *storeSDTask* has to be executed. Obviously, with an execution time of 11 ms, 680 ms is enough to have the task finished. In Figure 4.13 the difference in start time of successive executions of the *storeSDTask* is shown. For this test, the sensor short was transmitting data for 20 s, and there is only a part taken of the results printed. From the figure, it can be concluded that the time between the successive executions varies between 680 ms and 684 ms. That is because the buffer does not store exactly 170 samples. Sometimes it also takes a part of a sample and therefore has to wait for extra 4 ms.

For the *wifiModuleTask*, it is expected that it will be executed every 60 ms, because the data is sent per 15 samples. The time interval of 60 ms is also enough time for the *wifiModuleTask* to be executed. The *wifiModuleTask* has a quite low execution time because the transmission is non-blocking. For the *wifiModuleTask*, the time interval has also been measured. The result is as expected 60 ms and shown in Figure 4.13. However, when the hardware flow control is blocking UART, more than 15 samples may be transmitted over UART. Blocking will result in larger intervals between the successive executions of the *wifiModuleTask*. But for the scheduling and the executions, this should have no effect.

As the activation frequency for the *readAccGyroTask* and *storeBuffTask* is 250 Hz, and the activation frequency for the *readMagTask* is 100 Hz. The tasks with 250 Hz are activated at $0, 4, 8, 12, 16, 20, ...ms$, while the task of 100 Hz is activated at $0, 10, 20, ...ms$. The activation of the tasks at 8 ms and the *readMagTask* task at 10 ms, is the closest to each other. In such case the sensor short should first execute *readAccGyroTask*, and because there is no new magnetometer sensor data at 8 ms, the second task to be executed is *storeBuffTask*. But the risk of the task priorities as assigned in Table 4.1, is that the activation of *readMagTask* might interrupt the *storeBuffTask*. But luckily it is validated that this is not possible as the total maximum execution time of the *readAccGyroTask* and *storeBuffTask* is $1 + 1 = 2ms$ (Table 4.3). To get a better idea of how the task activation and scheduling looks like, the scheduling scheme is shown in Figure 4.14.



```
        storeSDTask           wifiModuleTask
Interval Ex.: 680     Interval Ex.: 60
Interval Ex.: 684     Interval Ex.: 60
Interval Ex.: 684     Interval Ex.: 60
Interval Ex.: 680     Interval Ex.: 60
Interval Ex.: 684     Interval Ex.: 60
Interval Ex.: 684     Interval Ex.: 60
Interval Ex.: 680     Interval Ex.: 60
Interval Ex.: 684     Interval Ex.: 60
```

Figure 4.13: Time interval between successive executions in milliseconds for *storeSDTask* (left) and *wifiModuleTask* (right)

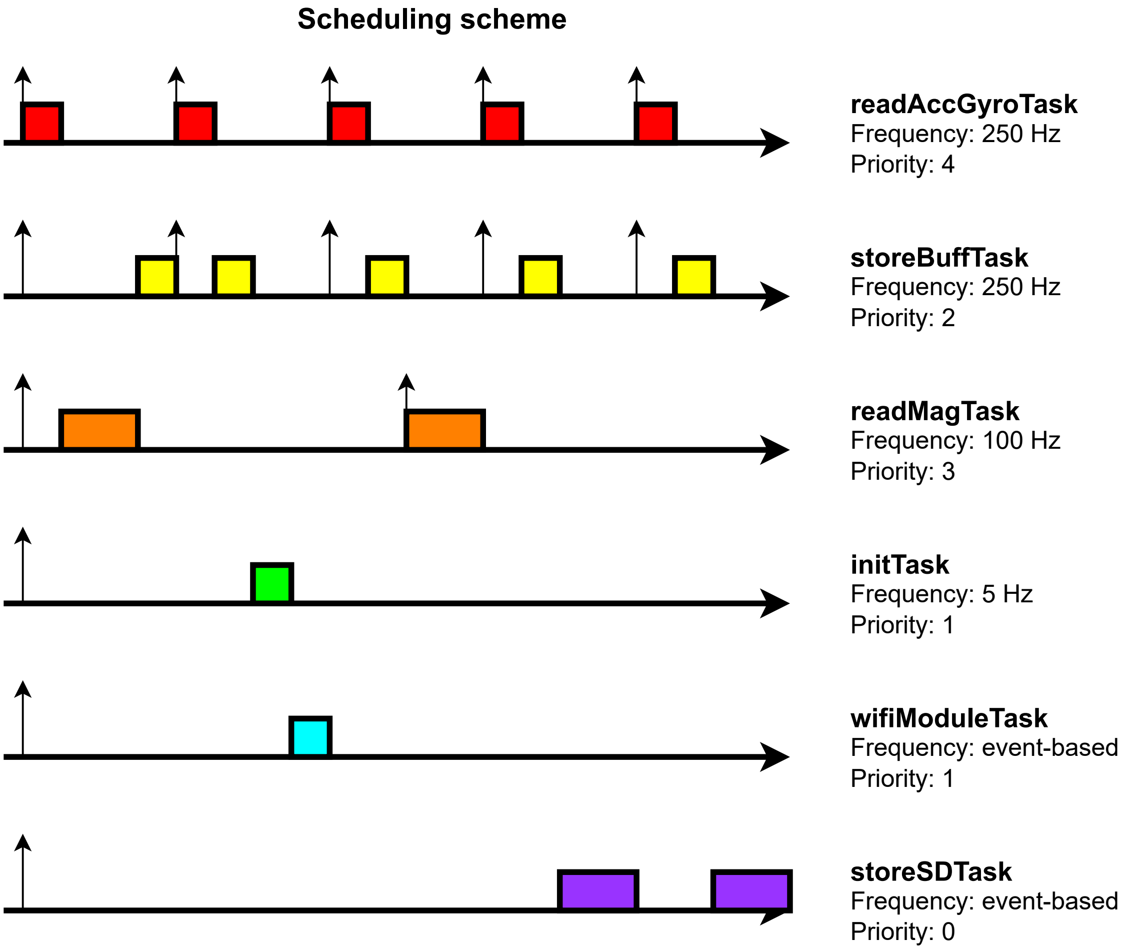


Figure 4.14: The scheduling scheme based on lab validation tests

In Figure 4.14 only a part of the task schedule is shown. At the beginning of the scheduling scheme shown, all tasks are activated simultaneously, but this is a very rare situation. Besides, in the scheduling scheme shown, a task is running at all times, but this is not always the case.

Now that the system is working, power measurements can be done. The system should be able to last for at least 90 minutes. A digital USB power meter is used (Figure 4.15) to measure the power consumption. The USB power meter had to be used because the GNG-109 battery has a micro USB connector. The USB power meter was connected in series with the sensor shorts and the GNG-109.

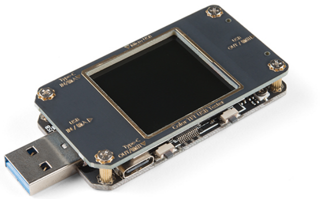


Figure 4.15: The USB power meter used for the power measurements.

For the power consumption test, the sensor shorts was transmitting for one hour long to the server. After one hour, it appeared the sensor shorts consume 219 mAh. The measured consumption is much less than what was expected in section 3.5. That is because we made an assumption based on the peak current according to the data sheet of the xPico. With a consumption of 219 mAh, the sensor shorts is able to last for 11.8 h.

5

Testing and results of the DCP

Up to this point, the design of the data delivery procedure has been implemented and validated. The sensor shorts can communicate with the central server in a lab environment, in which the communication range is limited. On a football field, we might need more base stations. The wireless connection needs to be tested to understand how the connection behaves in the field. The acquired field test results are used to determine how many base stations are needed and how to make the design less sensitive to environmental changes. Those tests and the acquired results will be discussed in this chapter.

5.1. Test setup

Before going into the tests performed and the results, the test setup will be discussed. Figure 4.9 showed an overview of the end to end connection between the server and the sensor shorts. The three modules: central server, TL-R470T+, and EAP225-Outdoor from Figure 4.9 are the part of the system which receives the sensor data. Figure 5.1 shows the setup for the receiver part of the system. For all the tests done, this is generally what the setup looks like. The server and the person controlling the server are close to the base station due to the limited Ethernet cable length. However, the seats are placed such that it does not obstruct the path between the AP and sensor shorts.

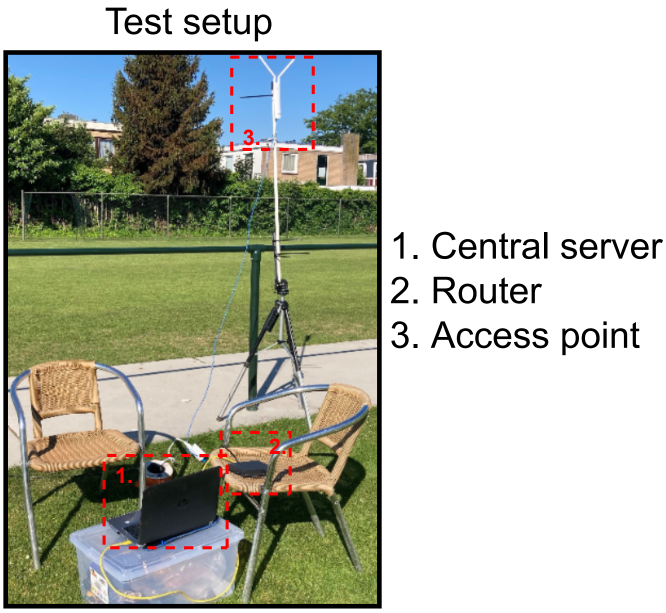


Figure 5.1: Overview of the general setup at the field tests

5.1.1. Antenna placement

Another critical aspect of the setup is the position the antenna will be placed. The ideal position for the antenna of the sensor shorts is determined by performing tests. Before starting, it was known that the antenna would be placed on the waist. Nevertheless, the exact position on the waist had to be determined.

Section 3.3.3 showed that an electromagnetic wave could attenuate due to human obstructions. Even the athlete wearing the sensor shorts could obstruct the signal. So the antenna should be placed where the signal experiences minor interference. We could expect obstructions from the arms when the antenna is placed on the side of the waist. In the following test, a comparison is made between having the antenna on the side and the back. For the comparison, the Received Signal Strength Indicator (RSSI) of the xPico module is measured. The measured value can be retrieved from the AP web management page.

For the test, the sensor shorts is worn by a male test person of 1.80 m and 72 kg. The main PCB of the sensor shorts is placed at the center of the back. The antennas on the back and side are placed left of the main PCB and on the right side, respectively. The height of the antenna on the body is approximately 1.10 m.

In this test, the test person will stand at 2 different distances from the receiver. The distances are 5, and 40 m from the base station. At these distances, the antenna is pointing in three different directions. Figure 5.2 shows the different antenna pointing directions. The antenna is radiating in different directions for these positions.

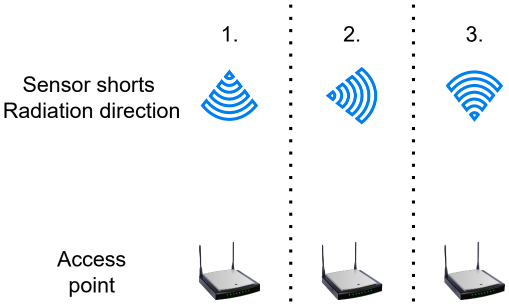


Figure 5.2: The different directions the patch antenna radiates used for the test

Table 5.1 shows the test results for the different distances and antenna pointing directions. The results are denoted as (x, y), with x and y the results of having the antenna on the back and the side, respectively.

Table 5.1: Measured RSSI at 5, and 40 meters with the antenna, on the back and side, pointing in directions 1,2, and 3.

(back, side)	**Direction 1**	**Direction 2**	**Direction 3**
5 m	(-43,-54) dBm	(-46,-55) dBm	(-63,-65) dBm
40 m	(-54,-66) dBm	(-57,-65) dBm	(-71,-76) dBm

Section 3.3 showed that the RSSI would be the strongest when the antenna on the sensor shorts points to the AP and there is no obstruction between the two. Standing with the antenna on the back in direction 1 (of Figure 5.2), such a scenario is simulated. Table 5.1 shows that, in the mentioned scenario, the RSSI is the strongest. Indeed there is a difference between the two distances for this scenario due to the PL.

While standing in direction one, there is a difference between measured RSSI when having an antenna on the back or the side. In direction 1, an arm is in front of the antenna when the antenna is on the side. With the arm in front of the antenna, the wireless signal experiences an obstruction. Table 5.1 shows that the arm causes a significant decrease in the measured RSSI. At 5 m the RSSI measured is −54 dBm when the antenna is on the side and thus an arm is obstructing the signal. When the antenna is on the back, i.e., without obstruction, the measured RSSI is −43 dBm. The measured RSSI is a difference of more than 10 dBm compared to when the antenna is on the side.

In direction 3, we would expect a weaker RSSI compared to direction 2. One of the reasons is the radiation pattern shown in Figure 1.2. Another reason is the body, which is between the antenna and the base station. The body partially absorbs the signal going to the base station. Indeed, Table 5.1 shows that the RSSI measured for direction 3 is weaker than direction 2.

Figure 1.2 shows that the patch antenna has a maximum directivity in the direction perpendicular to the patch antenna. Therefore it could be expected that for direction 2 the measured RSSI is weaker than in direction 1. However, there is one notable observation, as we look at Table 5.1. There is only a minimal difference between the measured values for the antennas on the side in directions 1 and 2. In this case, the receiver benefits from the signal radiating in the patch antenna's horizontal plane.

Because the antenna on the side might experience obstruction due to the arm along the body, it is desirable to place the antenna on the back. On the back, the obstruction due to the body is minimal. Figure 5.3 shows the placement of the antenna on the back. The blue square is the main PCB in the center of the back, and the antenna (red square) is on the left of the main PCB. Because we will test the dual antenna configuration, one antenna is also on the front of the body.

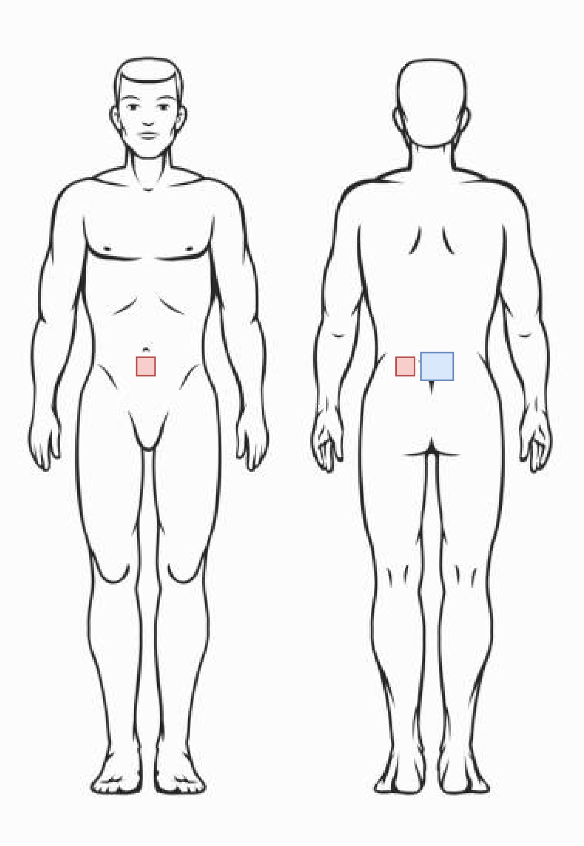


Figure 5.3: Placement of the antennas (red squares) and the main PCB (blue square) on the human body

5.2. Measuring RSSI

Section 3.3 showed that the RSSI decreases when the distance between the transmitter and receiver increases. Besides, the human body also causes the RSSI to decrease. This section will look into the effects of distances and human body obstructions on the RSSI to predict the expected effects on the football field.

For the RSSI measurements, the tests are performed on a field of approximately 100 x 65 m. The base station is placed outside the field facing the halfway line (Figure 5.4). If the test person stands at the halfway line, he is in front of the base station. The field is divided into four boundaries representing the extremes when the base station is at the center of the sideline. Figure 5.4 shows the boundaries of the field named tracks, and it shows where the base station is positioned. The figure also shows the antenna's orientation on the test person. In the 0 degrees orientation, the patch antenna is pointing to the sideline where the base station is positioned. In the 90 degrees orientation, the patch antenna points to the field's right half.

We are only testing with one antenna for the RSSI measurements, and the antenna being used is the one on the back. These tests have to give an idea of what RSSI values might be expected and should help to characterize the expected RSSI. As we are measuring over the boundaries, we might get an idea of the minimum and maximum RSSI values over the field.

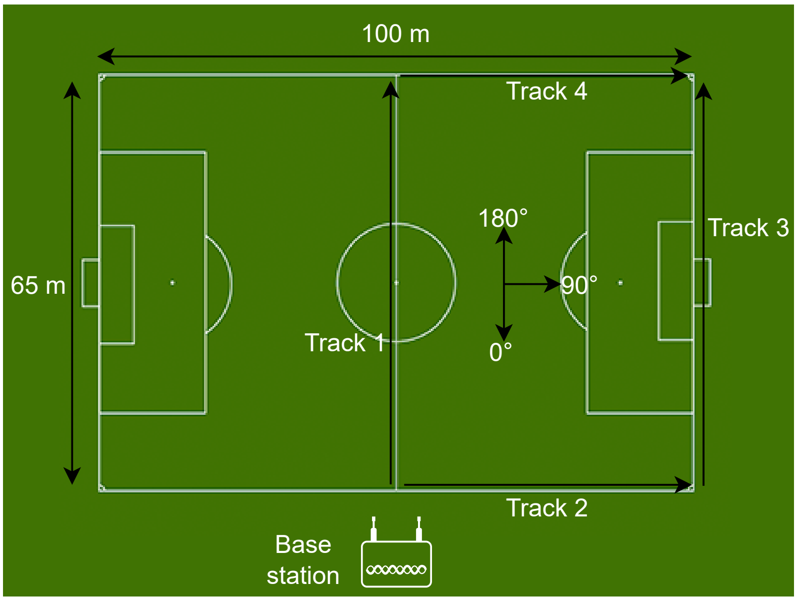


Figure 5.4: For the RSSI measurements the base station is placed pointing to the halfway line, RSSI measurements are performed over 4 different tracks. The angles in the figure indicate the direction the patch antenna is facing.

5.2.1. LOS RSSI measurement

The first measurements were done over track 1 (Figure 5.4) with the base station at the height of 1.1 and 2.2 m. Section 3.3 explained that for the LOS the received power could be described using the Free space or 2-Ray model, therefor this section will perform the LOS RSSI measurements. Using these measurements a model could be defined for our system case. Based on this model a range estimation could be made for the LOS, in case there are no obstructions. Over track 1, there is a pawn every 5 meters from the base station up to 65 meters. At every pawn, the LOS RSSI is measured three times. Three measurements are done to compensate for human errors. Because when the test person is slightly oriented differently, there is already a difference in RSSI. The orientation of the the antenna is 0 degrees (Figure 5.4) because we want to measure the LOS signal. The Free space and 2-ray model will be fitted for the average of the three measurements. Figure 5.5 and Figure 5.6 shows the results for the base station at 1.1 and 2.2 m, respectively.

The Figures show the result of the measured and the mean RSSI over the different distances. In the top graph of the figures, the Path Loss model has been fitted over the mean RSSI. The bottom plot shows the result of fitting the 2-Ray model over the mean values.

The logarithmic function shown in Equation 5.1 is used to fit the models. Equation 5.1 is a simplified version of Equation 3.1, $b \cdot \log(x)$ of Equation 5.1 represents the Path Loss variable of Equation 3.1. For the Path Loss and 2-Ray model, the Path Loss variable is shown in Equation 3.2 and Equation 3.10, respectively. The variable a consists of the remaining variables of Equation 3.1.

For the 2-Ray model, a few parameters that determine the variable x of Equation 5.1 must be determined. The parameters discussed in subsection 3.3.2 that must be determined are h_t, h_r, and ε_r. The parameters h_t and h_r represent the height, which is 1.1 m and 1.1 m or 2.2 m, respectively. The parameter ε_r was set to 3.

$$f(x) = a + b \cdot \log(x) \tag{5.1}$$

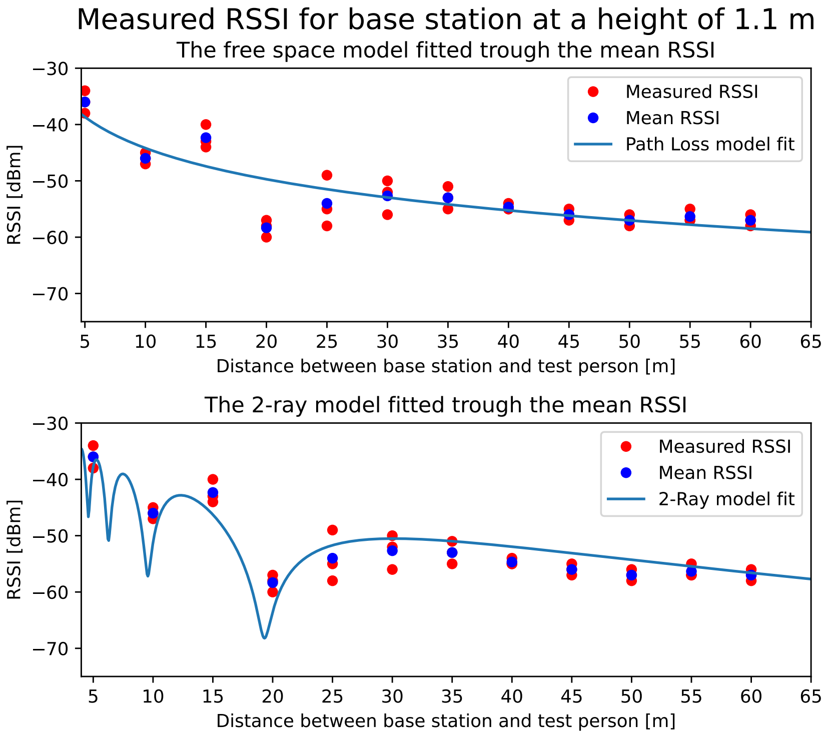


Figure 5.5: LOS RSSI measurement over track 1 with the base station at a height of 1.1 m.

Figure 5.5 and Figure 5.6 show that there is variation in the RSSI measurements for a specific distance. The variation in RSSI values might be caused due to different orientations at the independent measurement moments. The wave will propagate differently when the orientation at different measurement moments is slightly different.

The Path Loss model fits reasonably well through most mean values, but some mean RSSI values do not match the Path Loss model's trend line. For example, the mean values at 15 m, 20 m, and 25 m in Figure 5.5 and the mean at 25 m, 35 m, and 40 m in Figure 5.6. section 3.3 mentioned that the Path Loss model does not consider reflections. However, in our test case, there are reflections because of the soil ground. The 2-Ray model shown at the bottom of Figure 5.5 and Figure 5.6 considers the reflection. For those graphs, the mentioned points do better fit the curve.

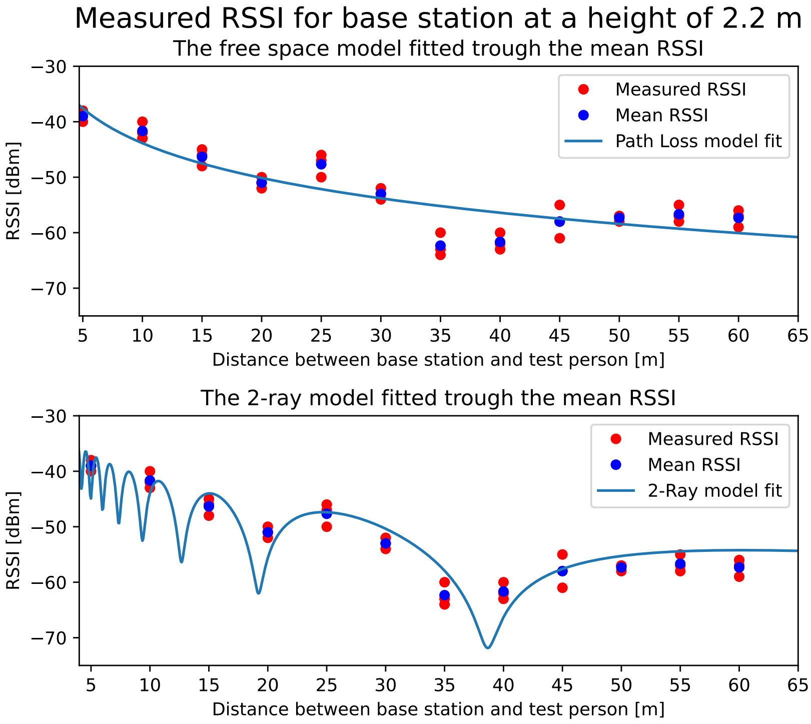


Figure 5.6: LOS RSSI measurement over track 1 with the base station at a height of 2.2 m.

The receiver sensitivity for the AP is −90 dBm, meaning a signal with an RSSI lower than −90 dBm cannot be correctly interpreted. The graphs in Figure 5.5 and Figure 5.6 do not reach this value. According to the fitted models, the system should easily cover the maximum distance on the football field, which is 120 m.

5.2.2. Obstruction RSSI measurement

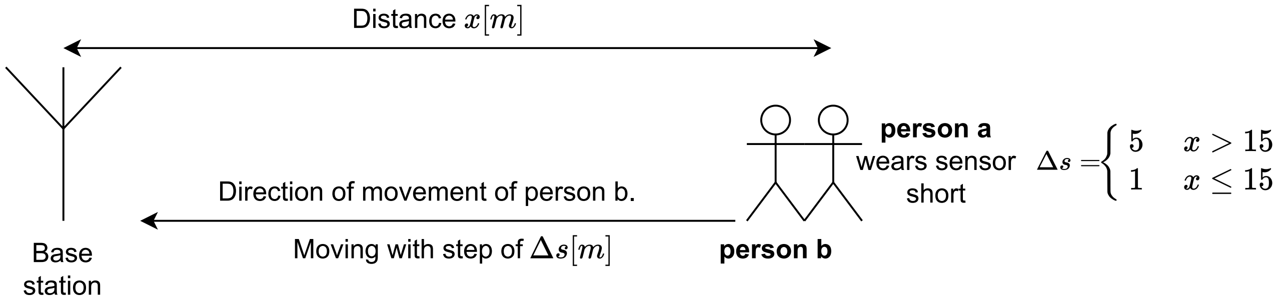

Distance $x[m]$
person a
wears sensor
short
$\Delta s = \begin{cases} 5 & x > 15 \\ 1 & x \leq 15 \end{cases}$
Direction of movement of person b.
Moving with step of $\Delta s[m]$
person b
Base
station

Figure 5.7: An simple overview of the test with obstruction

In the subsequent tests, we will examine the effect of a human obstruction on the RSSI. The tests performed are the same as in the previous test. However, now there will be a person between the base station and the transmitting short. The person wearing the sensor shorts will be called "person a", and the person who causes the obstruction is "person b" as shown in Figure 5.7. Person a stays in a fixed position while person b stands on multiple positions between person a and the base station. Person b always starts directly next to person a. Person b obstructs the signal by facing its belly to person a's back, where the patch antenna is mounted. When person a stands at 15 m or less, or larger than 15 m from the base station, the distance between person a and b increases 1 m or 5 m, respectively. Figure 5.7 gives a simple overview of how the movement of person a, relative to person b takes place. After each time person b arrives at the base station, person a increase his distance with 5 m, and person b starts again next to person a.

In these tests, the base station is at 1.1 m and 2.2 m. The reason to test on two heights is to see if a difference is measured in RSSI. Because the base station at 2.2 m is higher than the athlete, the LOS signal might experience less obstruction due to the human body. An attenuation is expected in the case of an obstruction relative to the case without obstruction.

The mean attenuation is plotted in Figure 5.8 for the two heights to evaluate the difference between the two heights. The attenuation is the subtraction of the mean RSSI (measured in the previous test) by the measured RSSI in case of obstruction. There are multiple attenuation values for a specific position of person a because of the different positions where person b will stand. These different values are taken together to calculate the mean attenuation at a specific position person a stands. So the mean attenuation shows the deviation from the actual measured mean RSSI from the previous tests due to the obstruction.

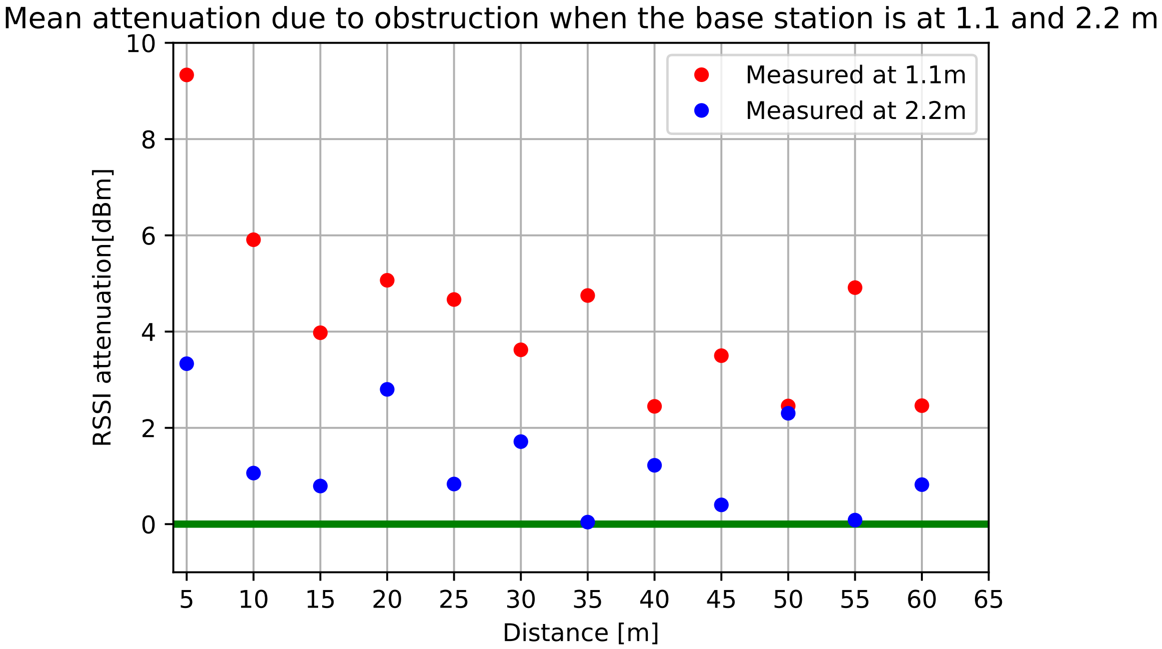


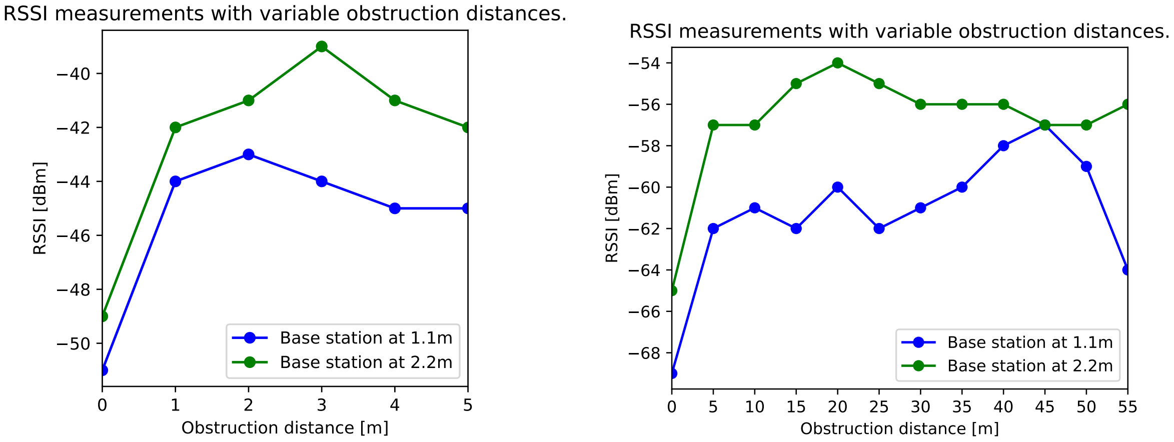
Figure 5.8: The mean attenuation for a player at a certain distance

Figure Figure 5.8 shows that more attenuation is measured for the base station at 1.1 m. The figure shows that when the base station is lower, the system is more affected by the human body obstruction. Typically the mean attenuation for the base station at 2.2 m is between 4 dBm and 0 dBm. The base station at 1.1 m attenuates between 6 dBm and 2 dBm if the data point at 5 m is left out. At 5 m, the mean attenuation is significant since person b is very close to the base station and to person a.

Because the base station at the height of 2.2 m is less affected by a human obstruction, it is chosen to continue with the base station at 2.2 m. For further research, it might be beneficial to investigate if a height of more than 2.2 m would deliver better results.

Figure 5.9 shows the results for two of the fixed distances of person a. Figure 5.9a and Figure 5.9b have a fixed distance of 5 m and 55 m, respectively. The x-axis refers to the distance between person b and person a. At 0 m, person b stands exactly next to person a. The figures show the result for the base station at the height of 1.1 m and 2.2 m. As expected, when the obstruction stands next to person a, the RSSI is the lowest. However, there is already a difference between the two heights at this obstruction distance. It might be caused because the LOS for the lower base station is blocked by the body part, which contains a higher volume of water fluids than the part that blocks the LOS for the more elevated base station.

As the distance increases between person a and b, the RSSI increases because the obstruction is less. But for the base station at 1.1 m, the RSSI decreases slightly when the obstacle comes closer to the base station. In Figure 5.9a and Figure 5.9b, we see that happens at 4 m and 50 m, respectively. This occurrence slightly shows the advantage of a more elevated base station. Because the base station at 2.2 m is higher than the obstacle's length, the obstacle doesn't stand in the LOS of the base station when it is close to the base station.



(a) RSSI measurements with the test person at 5m from the sideline.

(b) RSSI measurements with the test person at 55m from the sideline.

Figure 5.9: Test person stays at fixed distance and x-axis shows the distance between the test person and obstruction.

5.2.3. Measuring RSSI over field borders

As mentioned at the beginning of section 5.2, measurements are performed over four tracks shown in figure 5.4. Those tracks represent the extreme position cases when the base station is placed at the halfway line. Those tests will indicate the expectations for a minimum and maximum RSSI value and how the antenna behaves over the different tracks.

The first measurement is performed over track 1. The measurement starts at the sideline, where tracks 1 and 2 intersect (Figure 5.4). Every 5 meters, there is a pawn till the other field sideline. The distance between the first pawn and the base station is approximately 3.8 m, and the distance between the first and last pawn is 65 m.

The measurement starts at the closest pawn to the base station. The RSSI for the different orientations, as shown in Figure 5.4, are measured at every pawn. First, the RSSI with the antenna pointing to the base station (0 degrees). Second, the antenna points to the right half of the field (90 degrees), and last, the antenna points away from the base station (180 degrees). After those three have been measured, the test person goes on to the next pawn, 5 m further away from the base station. This process repeats till the last pawn has been reached. During the tests, the test person wears the antenna on his back.

Figure 5.10 shows the results of the described tests. The plot shows the measured RSSI for the different distances and orientations. From the radiation pattern of the patch antenna, we could expect that when the radiation plane is pointing more and more away from the base station, the RSSI decreases. Therefore in track 1, we would expect the best performance from the 0 degree angle and the worst from the 180 degree angle. The results in Figure 5.10 show that this is indeed the case. The 0 and 90 degree orientations have minor differences compared to the 180 degree scenario. The significant difference is mainly caused due to the radiation pattern and the test person's body, which is between the antenna and base station in case of the 180 degree orientation. The significant difference shows a drawback of choice for a patch antenna on the back. When the base station is not in the 180 degree plane of the patch antenna, the signal will be less substantial.

In the 180 degree orientation, the RSSI starts to increase when the test person comes closer to the sideline on the other side of the field. The expectation is that this happens due to the fences along the football field. The fences consist of reflective material, which could cause the strong signal radiated in the fence's direction to reflect to the base station.

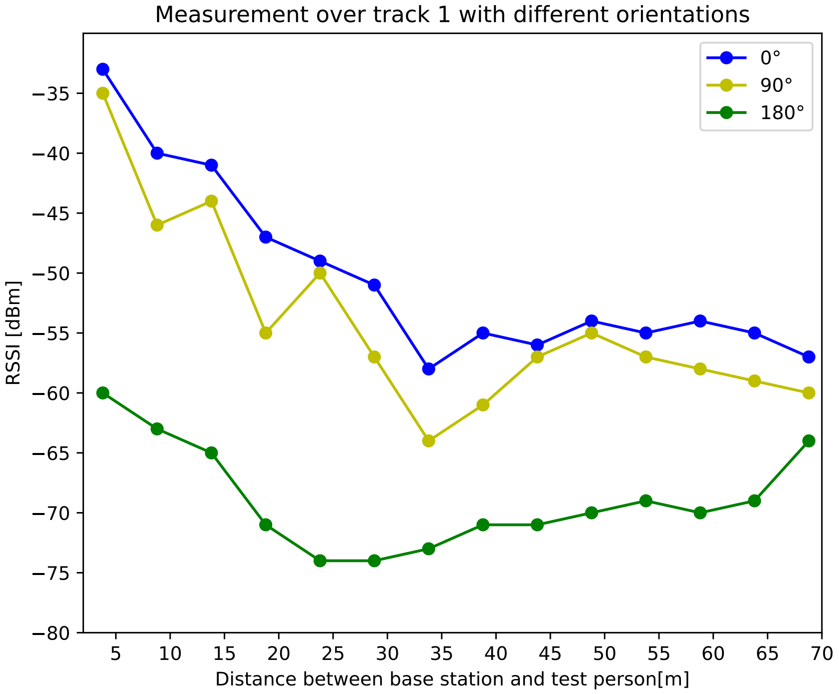


Figure 5.10: The measured RSSI values over track 1 (Figure 5.4) for three different orientations of the patch antenna, with 0 degree the antenna perpendicular to the base station.

The mean difference for the 0 degree and 180 degree orientations is 18 dBm, which is a significant difference between the two orientations. The following section will investigate what this difference means in terms of wireless communication. However, it seems evident that a second antenna will be needed on the front or an additional base station on the other side of the field. From the figure it becomes clear that the smallest measured RSSI is close to −75 dBm.

For track 1, we have not measured the RSSI in the case of -90 degrees. The -90 degree orientation was ignored because the assumption that the RSSI value for -90 and 90 degrees will be somewhat similar. Besides, we want to know the lowest RSSI values possible; it is not expected that the -90 degree orientation has a weaker RSSI than the 180 degree orientation. Therefore to save measurements, the -90 degree orientation was ignored.

For tracks 2, 3, and 4, from the radiation pattern and previous measurements, it could be expected that the orientation of -90 degrees will not give the lowest possible RSSI. It is expected that the antenna orientation with an angle of 90 or 180 will have the lowest RSSI for tracks 1, 2, and 3. Therefore we are focused on the results for the 90 and 180 degrees orientation of the antenna. But for reference, the 0 degree antenna orientation is still plotted in the results shown in Figure 5.11. Figure 5.11 shows the result of the three different tracks.

As expected, the 0 degree orientation provides the best RSSI values for all three tracks. There is something remarkable about the 0 degree orientation in the results from tracks 2 and 3. In track 2, there is a significant drop in the RSSI as the distance increases, while in track 3, the RSSI increase when the distance increases. The increase of RSSI in track 3, even when the distance increases, is because the path angle between the base station and patch antenna decreases as the distance increases. Because of the increasing angle, the signal arriving at the base station comes more from the center. This observation could again be explained by the radiation pattern of the patch antenna (Figure 1.2).

Another remarkable observation from the results on track 3 is that after 25 m, the 90 degree orientation has a higher RSSI value, up to 45 m. This could be caused due to the large fence behind the goals. Due to these fences, strong signals get reflected and could yield a better RSSI.

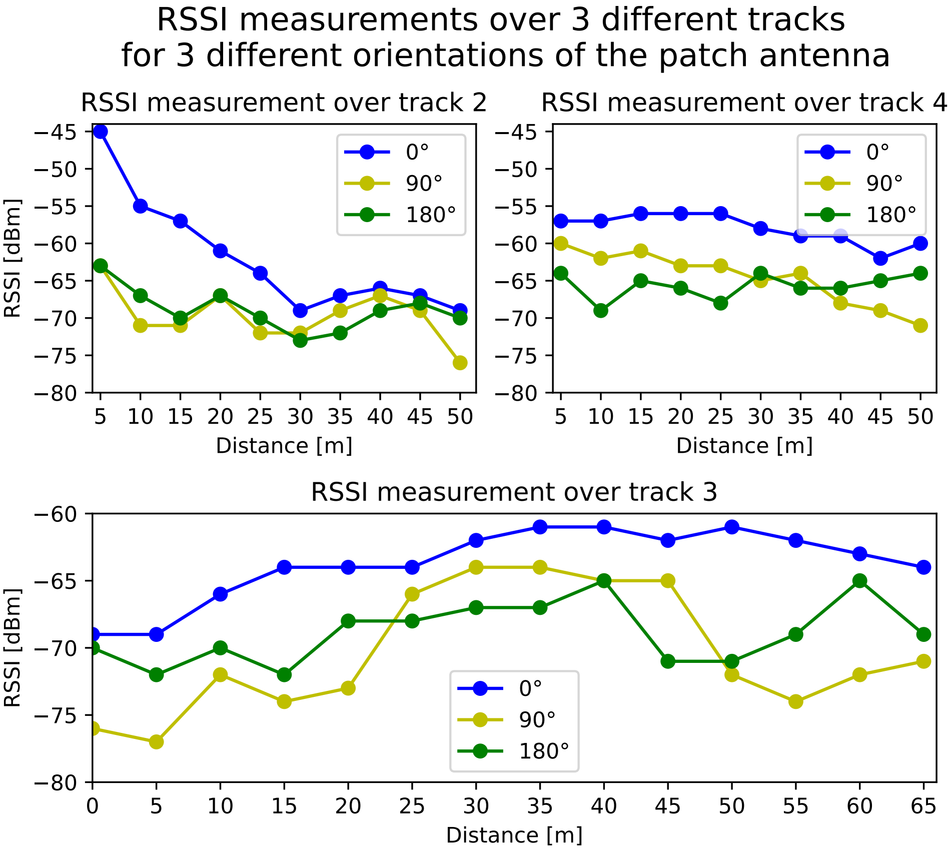


Figure 5.11: The measured RSSI values over track 2, 3, and 4 (Figure 5.4) for three different orientations of the patch antenna, with 0 degree the antenna perpendicular to the side line where the base station is positioned.

Figure 5.11 shows that the weakest RSSI is from the 90 degree or 180 degree orientation. For both orientations, the base station is not in its 180 degree plane, which results in the RSSI being weaker than the 0 degree orientation.

To conclude from the measurements performed, the RSSI varied from -33 dBm to -77 dBm. The weak signals are mainly measured when the base station is not in the 180 degree plane of the patch antenna.

5.3. Delay and packet loss

The previous section gave an idea of the behavior of the RSSI over the field. However, we do not know yet what effect the signal strength has on the data transmission. It is known that a weaker received signal strength causes the receiver to be unable to understand the signal. The signal needs then to be retransmitted. This section will investigate the link between the RSSI and data communication. We want to determine what the measured values in the previous section mean regarding sample loss and delay. Based on these results, a conclusion could be drawn about the number of base stations needed.

5.3.1. Packet loss

In the following tests, the test person has again to stand still in a fixed position. The distance to the base station is not essential this time for these tests, but the RSSI of a specific position. The measurement is done for the RSSI of −85 dBm till −40 dBm. A position is determined where the desired RSSI is obtained. For example, for an RSSI of −85 dBm, the test person had to stand at 100 m from the base station with the antenna pointing away from the base station. At the determined position, the sensor shorts will be monitoring the athlete for one minute. The athlete has to stand still to try to keep the RSSI constant. Because it is hard to stand still, the measurements have been performed three times. The results in this subsection are the average of the three tests.

For measuring the packet loss, every sample has a sequence number. We can obtain the number of packets lost by analyzing the sequence numbers. Between each successive sample, the sequence number increase by 1. When a sequence number is missing, the sample has been lost. Figure 5.12 shows the percentage of samples lost for the different RSSI values. It is known during one minute (60 · 250 =) 15000 samples are expected. Using this information the percentage of samples lost can be calculated.

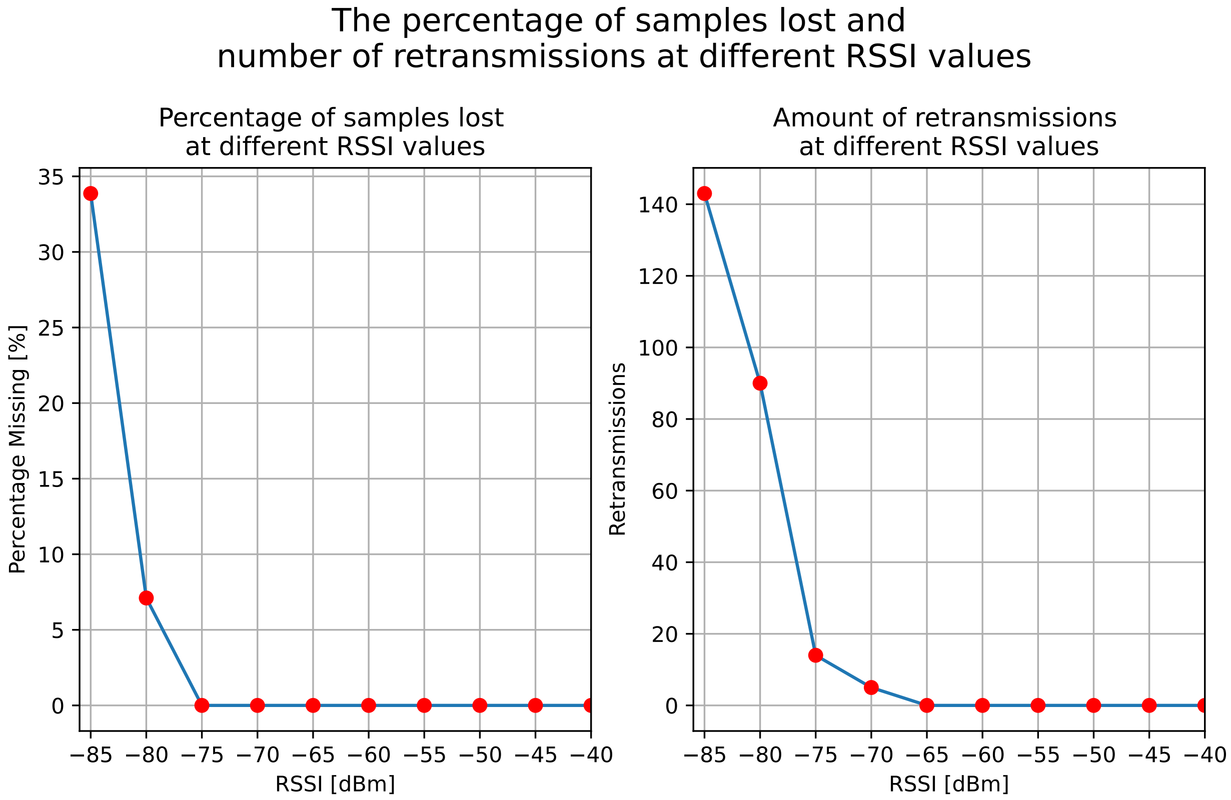


Figure 5.12: Sample loss for different RSSI values, during a measurement of one minute

Figure 5.12 shows that for a signal lower than −75 dBm packets start to get lost. At −85 dBm, approximately 34% of the packets are missing. The missing packages are caused due to the weak connection. The weak connection causes packet errors, and(/or) messages get lost. Because of the TCP connection, the server requests retransmission when packet loss or error occurs. As we saw before, the xPico holds the data in its buffer until an acknowledgment is received. When retransmission is requested, the xPico's buffer keeps holding the data in its buffer. When the buffer on the xPico is full, the sensor shorts uses its buffer on the microcontroller to store the data. Data will be lost when the buffer on the sensor shorts is full because it can not be saved anymore on the sensor shorts. The buffer on the sensor shorts can save 500 samples, so if the buffer on the xPico is full for more than 2 seconds, data gets lost.

Figure 5.12 shows in the right graph the number of retransmissions. The retransmissions are measured using Wireshark; it is a program that is used for network analysis. Wireshark captures the communication between the sensor shorts and the central server.

In the results obtained by capturing the traffic (Figure 5.12), it is observed that retransmission starts at an RSSI of −70 dBm, but it has no effect yet on the packet loss. When the signal becomes weaker, the amount of retransmissions increases. At −80 dBm, the retransmissions start to casue the packet loss.

From the two graphs in Figure 5.12, we can conclude that a signal of less than −75 dBm should always be prevented. Signals between −75 dBm and −65 dBm are less desirable, but the system can handle those signals.

The data lost represent a particular time that is lost. For example, when two samples are lost, in the case of a sampling frequency of 250 Hz, 8 ms of data is lost. This loss duration is analysed because it is essential for sports scientists. When the loss duration is too long, the model used by the sports scientists can not reestablish the movement of the athlete it is monitoring. Table 5.2 shows the loss duration characteristics. Indeed we expect a higher loss duration for the lower RSSI value, which is the case according to the table. The maximum measured loss duration is 2004 ms and 5948 ms for −80 dBm and −85 dBm, respectively.

Table 5.2: Time characteristics of the loss duration

	-80 dBm	**-85 dBm**
Max	2004 ms	5948 ms
Mean	920 ms	2112 ms
Median	612 ms	1808 ms

5.3.2. Delay

A weaker connection also affects the delay of the data delivery. We will look into the effects it has on the data delivery delay. Unfortunately, we can not measure the exact delay because to be able to measure it, the server and the sensor shorts need to be synchronised. Nevertheless, based on how the system is designed, we could come up with an estimate which expresses the upper boundary of the data delivery delay.

Figure 5.13 is used to explain how the upper boundary is determined. The image shows the timeline of the Central PCB, xPico, and server. For the timeline, the unit is in milliseconds. Between the Central PCB and xPico, the data is communicated via UART transmissions, followed by transmission over WiFi from the xPico to the server. Every 4 ms, the Central PCB samples the sensor data, the data sampled is expressed as a vector of x in Figure 5.13.

The Figure shows three different cycles, and we will call these cycles the data transmission cycles. Data transmission cycle 1 represents an ideal cycle. Because as we know, in an ideal scenario, every 60ms, 15 samples are sent from the sensor shorts to the xPico. Cycle 1 transmits the first 15 samples at 60 ms to the xPico. The data has to be saved on the xPico, which causes the buffer to become full. The data is transmitted over WiFi to the server when the buffer is filled. The data received on the server have then to be acknowledged by the server. When the xPico receives an acknowledgment, the xPico knows that it has successfully transmitted the data. Upon successful transmission, the xPico deletes the data from the buffer. In cycle 1, the data is deleted before the Central PCB has sampled the following 15 samples. A cycle finishing before there are 15 new samples available is the characteristic of an ideal cycle. Because cycle 1 finishes in time, the 15 new samples on the Central PCB can immediately be transmitted.

Cycle 2 represents a situation where the data transmission cycle is not ideal. Figure 2.1 shows much more time between the WiFi transmit and the acknowledgment compared to cycle 1. The time difference is, for example, caused due to weak signal strength. Therefore retransmissions are needed, and the transmission takes longer compared to cycle 1. In cycle 2, the xPico receives the acknowledgment at 185ms. At 185ms, more than 15 new samples are already available at the Central PCB. However, because the xPico's buffer is not empty yet, it can not immediately transmit the new samples to the xPico. The central PCB has to wait till the previous data transmission cycle finishes. When the cycle finishes, the Central PCB transmits all new data to the xPico.

When denoting the time of the UART transmissions as t_n with $n \in \{1, 2, ...\}$ and the sampled time of a sample as $x_{n,m}$ with $m \in \{1, 2, ...\}$, n represents which transmission the samples belongs to and m is the id of the sample. For every new sample m increases by 1, every sample has an unique value for m. From the figure it becomes clear that for a sample transmitted at $n = i$ with $i \in \{1, 2, ...\}$, its data transmission cycle finishes at $n = i + 1$. So for calculating the data transmission cycle (DTC) the following equation can be used:

$$DTC[ms] = t_{i+1}[ms] - x_{i,j}[ms]. \tag{5.2}$$

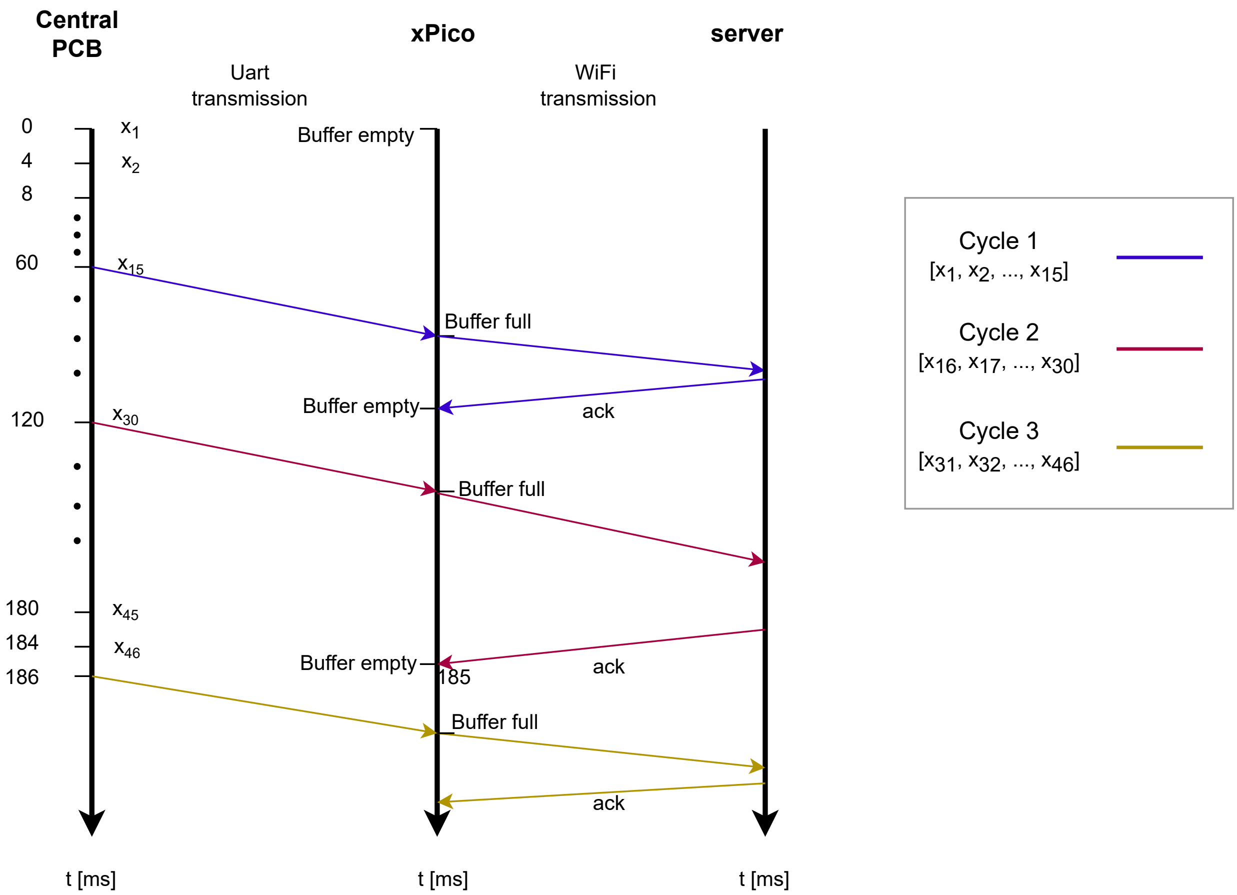


Figure 5.13: Number of re-transmissions for different RSSI values, during a measurement of one minute

Next, we will use Equation 5.2 to examine the data transmission cycle for the measurements performed in subsection 5.3.1. The retransmission plot in Figure 5.12 shows that there are no retransmissions for the RSSI values from −65 dBm to −40 dBm. The data transmission cycle delay for those RSSI values all look similar. To highlight one of the results, Figure 5.14 shows the data transmission cycle delay of −50 dBm. The Figure shows how many samples are located in a certain interval. For −50 dBm, all the samples are between 60 ms and 120 ms and equally distributed over 15 bins. The reason that it is all between 60 ms and 120 ms is that each cycle is ideal. When the cycle is ideal, it means that $t_{i+1} - t_i = 60ms$. In words, between each UART transmission, there is always a difference of 60 ms. So every transmission consists of 15 samples; for the first sample, sampled at 0 ms and transmitted at 60 ms, Equation 5.2 results in 120 ms. For the fifteenth sample, sampled at 60 ms and transmitted immediately after being sampled, equation Equation 5.2 results in 60 ms. Therefore the samples between the first and fifteens sample fall between 60 ms and 120 ms, with 4 ms between each sample. It should also be noted that the mean is 88 ms. Increasing the amount of simultaneously active sensor shorts to five had no effect on the DTC shown in Figure 5.14. We expect the mean to increase when we start having retransmissions.

Figure 5.15 shows the data transmission cycle delay for −70 dBm and −75 dBm in the top and bottom graphs, respectively. Figure 5.10 showed that there are retransmissions for those two RSSI values but no losses yet. As explained before, the retransmissions cause more delay in the systems, which increases the mean data transmission cycle delay. For −70 dBm and −75 dBm, the mean is 121 ms and 152 ms, respectively. Nevertheless, for −70 dBm, there is a more extensive spread in the data transmission cycle delay. The larger spread could be caused because one of the cycles needed more time than the cycles in −75 dBm.

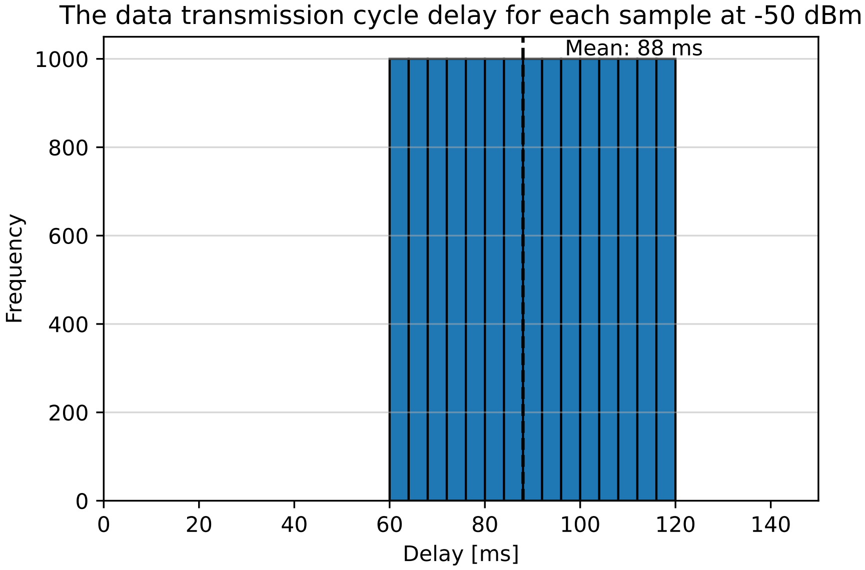
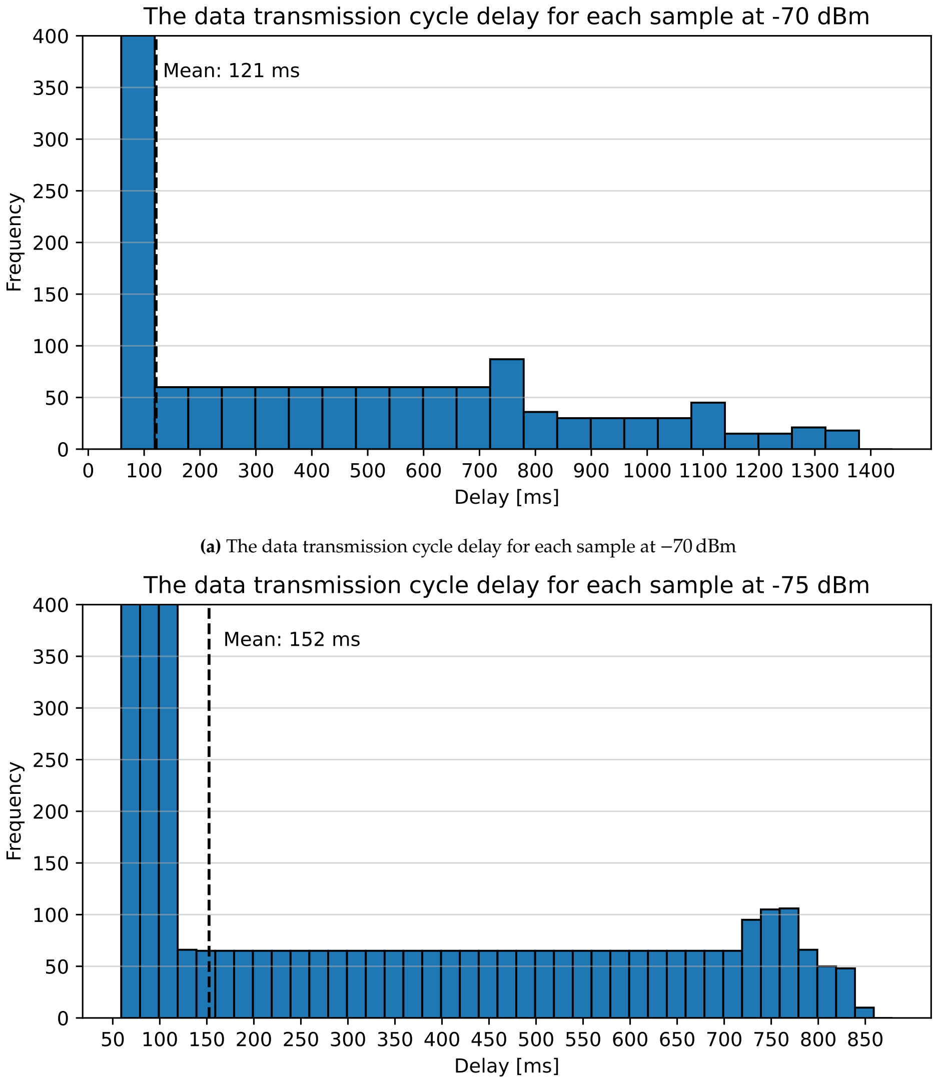


Figure 5.14: The data transmission cycle delay for each sample at −50 dBm

Figure 5.16 shows the data transmission cycle delay for −80 dBm and −85 dBm in the top and bottom graphs, respectively. These RSSI values experience packet losses and significantly more retransmissions, which can also be seen from the figures. Because the delays are more significant, for −80 dBm and −85 dBm, the mean is almost 5 s and 17 s, respectively. The gaps in the time axis, like in the top graph of Figure 5.16, are the consequences of data losses. The data losses are a consequence of the considerable data transmission cycle delay. Because a more significant delay causes the buffers on the sensor shorts to be full, and no new data can be stored anymore.

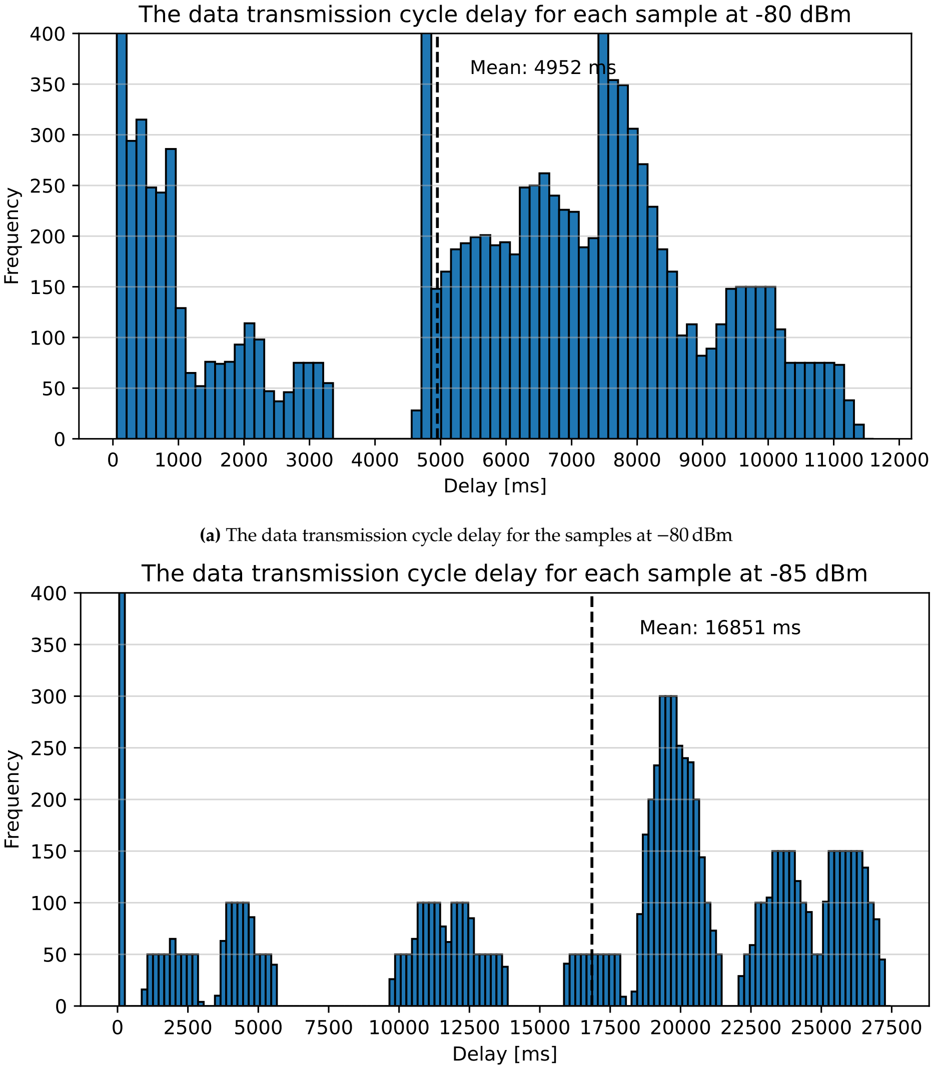
As we can see from the results, the RSSI values lower than −80 dBm cause significant delays and even data losses. Therefore we should ensure an RSSI minimal of −85 dBm over the football field.



(a) The data transmission cycle delay for each sample at −70 dBm

(b) The data transmission cycle delay for each sample at −75 dBm

Figure 5.15: The data transmission cycle delay for the samples at −70 dBm and −75 dBm



(a) The data transmission cycle delay for the samples at −80 dBm

(b) The data transmission cycle delay for the samples at −85 dBm

Figure 5.16: The data transmission cycle delay for the samples at −80 dBm and −85 dBm

5.4. Dynamic tests

The previous section showed that when the base station is placed on the halfway sideline and the athletes are equipped with only one antenna, there is a chance of data loss. To solve this problem, we could use a second antenna that should cover the area on the front of the body or extra base stations. The following subsections will look into these solutions.

Up to this moment, only static tests have been performed. For the following tests, dynamic tests are performed. Dynamic tests are needed to ensure a situation in which an antenna or base station switch is necessary. By analyzing these situations, a conclusion can be drawn about whether a specific solution is feasible.

5.4.1. Dual antenna configuration

In this subsection, the dual antenna solution is examined. The dual antenna solution consists of two antennas mounted on the sensor shorts. In Figure 5.3, the location of the two antennas is shown. One antenna is located on the back, the second on the front. The xPico has been chosen as the WiFi module because it supports a dual antenna configuration. We will examine whether the dual antenna configuration can solve the problem occurring when the sensor shorts have only one antenna.

For examing the antenna switching, a test case has to be created in which it is necessary to switch between antennas. Figure 5.17 shows the setup used to create such a test case. The base station is now positioned at the back line of the field so that we can do measurements over longer distances. The test person starts at position 1, with his face pointing to point number 2 in Figure 5.17. The test person starts walking to position 2, turns around once it reaches 2, and walks back to the base station.

The expected behavior for the described test case is that the back antenna is first connected to the base station because the back antenna points to the base station when walking to position 2. When the test person reaches 2, he has to turn around. After turning around, the back antenna no longer points to the base station. Here we would expect the front antenna to be used for communication. Because the xPico checks each antenna port approximately every 310 ms when the signal level is below −20 dBm and when the average number of received packets is low. After checking, it selects the best antenna to use, i.e., the antenna with the highest RSSI. When the back antenna is pointing outside the field at position two (Figure 5.17), the RSSI is expected to be lower than −75 dBm, based on performed results. At the same time, the front antenna, which points to the base station, has a much stronger RSSI. Therefore it has to switch to the front antenna after turning around at point 2.

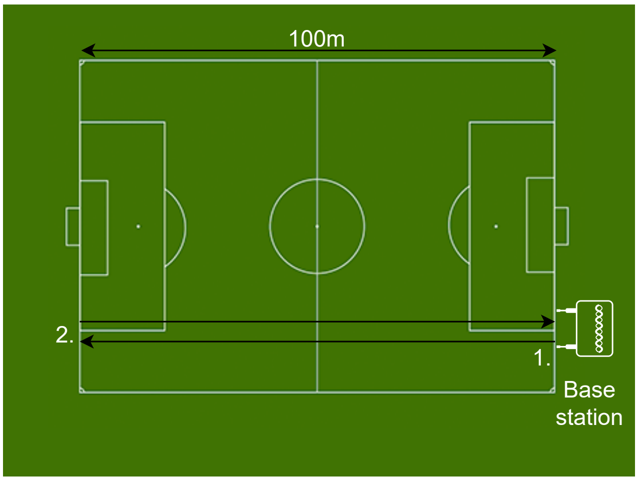
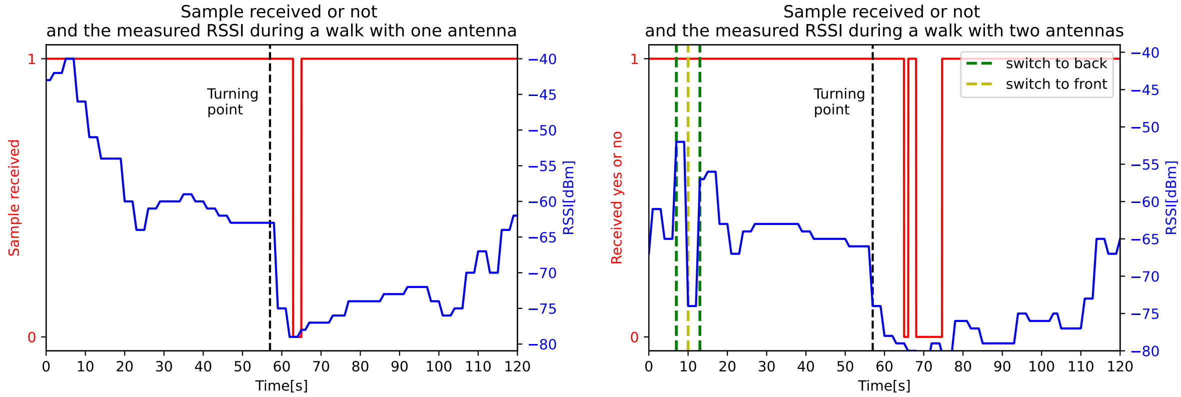


Figure 5.17: The test setup used for testing the dual antenna sensor shorts

Before going into the pattern diversity test, a null test will be performed. In the null test, only one antenna is used, located on the back of the test person. In the null test, we expect data loss after turning around at position 2 (Figure 5.17). Because at that moment, the connection between the base station and sensor shorts is too weak. From the test, it appeared that this was indeed the case. In total, there were 529 samples lost during measurement of 2 minutes, which is 1.7% of the total.

Figure 5.18 shows the results from the test for the single and dual antenna cases. The graphs show with the red lines whether the samples have been received or not. When the red line has a value of 1, the sample at the corresponding time has been correctly received by the central server. With the blue line, the RSSI value during the measurement is shown.

At 0 s, the test person starts walking to position 2 (Figure 5.17). The moment of reaching point 2 is indicated with a vertical black line. For both graphs, the turning point is at 57 s.



(a) Test results in case 1 antenna is used.

(b) Test results in case 2 antennas are used.

Figure 5.18: Results of a test walking over the field, with on left the result in case one antenna is used, and on the right in case 2 antenna are used. The graphs show whether a sample is received i.e. sample received is 1, and the RSSI during the test.

In Figure 5.18a, the RSSI decreases as the test person get further away from the base station. At the turning point, the RSSI experiences a significant drop in the RSSI value. This drop is expected because the test person's antenna no longer points to the base station. The RSSI reaches almost −80 dBm, and the connection becomes so weak that it loses data. As the test person walks back, the signal recovers, and no data is lost anymore. One thing that should be noticed is that between the start and end RSSI, there is almost a difference of 20 dBm. The difference is because the antenna points to the base station at the start, and at the end, it is not. When using the dual antenna configuration, we expect minimal difference because the front antenna could be used to connect with the base station.

In Figure 5.18b, the difference between end and start is insignificant. However, the RSSI at the end of Figure 5.18b is somewhat similar to that of Figure 5.18a. Besides, the RSSI at the start of Figure 5.18b is significantly different from the start of Figure 5.18a. Therefore it seems that the antenna switching algorithm of the xPico does not function properly. At the start, the front antenna is first connected to the base station, while the back antenna should be connected. After 7 s, the xPico switches to the back antenna, and at 10 s, it switches to the front antenna, and from 13 s onwards, it uses the back antenna. From 13 s onwards, the RSSI of figure Figure 5.18b follows the same trend as Figure 5.18a, while the desired behavior is that the xPico switches its antenna at the turning point. However, this is not the case, resulting in data losses.

This test shows that xPico's antenna switching algorithm does not function properly. After further investigation of the switching algorithm, it seems it does not work correctly. The extra tests performed for the switching algorithm test are described in Appendix A. Therefore using the dual antenna configuration of the xPico is not feasible. In the next section, we will examine the multi-base station solution.

5.4.2. Multi-base stations

The previous section showed that the dual antenna configuration on the xPico does not function properly. Another solution to get a more reliable wireless connection is to use multiple base stations. Luckily the base station and xPico provide IEEE standards which make fast roaming possible. The IEEE standard is IEEE 802.11r, a wireless protocol that allows the xPico to roam from one base station to another faster without needing to re-authenticate.

For a system with multi-base stations, the base stations form a mesh network, with one of the base stations having a wired connection to the router. The base station with a wired connection is the master node in the mesh and makes sure that data received by the other base stations is transmitted to the router. The unwired mesh network has the advantage that no long wires are needed from the base station to the router.

Unfortunately, when having a multi-base station setup, the ability to measure the RSSI of the xPico is lost because the web manager of the access points can not be accessed anymore. A central controller controls the access points, and this controller does not provide the ability to measure the RSSI. Therefore we could look only at the losses and analyze the packet loss to evaluate the performance of a multi-base station setup.

The test person has the antenna on his back for the following tests. Again a test case has to be created in which we ensure that roaming is needed. Therefore the same test setup can be used as in subsection 5.4.1, but now at the other end of the field, there is a second base station. For an overview of the setup, see Figure 5.19.

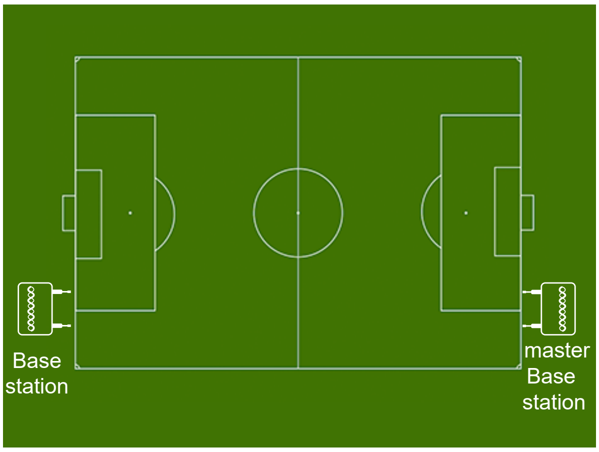


Figure 5.19: The test setup used in the multi-base station tests. The master base station has a wired connection to the router to transmit its and the other base station data to the router.

For the setup shown in Figure 5.19, the same test has been performed as in subsection 5.4.1. For the multi-base station setup, no loss has been measured this time. The expected behavior is that the test person is connected to the other base station fast enough when he turns around. Besides, as long as a base station is in the 180 degree plane of the patch antenna, the connection is reliable enough. When the switch of base stations exactly happens can not be determined with the tools used.

More challenging tests have been performed to verify the expectations. The test person will not walk between the two base stations but will run a predetermined, more complex trajectory.

In the first test, we want to verify that all packets will be received when a base station is always in the 180 degree plane of the patch antenna. The circuit to verify the hypothesis is shown in Figure 5.20 (Test 1). The test person starts at the master base station and runs to the outer point of the penalty arc on the other half of the field. Once the penalty arc is reached, the test person turns around and returns to the base station. As expected, there was no packet loss for this trajectory.

The second trajectory is a more complex test and is shown in Figure 5.20, noted as Test 2. The letters A, B, and C indicate the locations of the field where the test person has to make a turn. The test person starts at the master base station and follows the arrows in Figure 5.20. Figure 5.21 shows the result for Test 2. The figure indicates whether the samples have been received or not with a value of 1 or 0, respectively. Figure 5.21 shows that some of the samples have been lost after turn B.

In total, three samples were lost during Test 2. The loss occurred after the test person made turn B. From the previous test, the expectation was that loss would occur after turn B because the patch antenna doesn't face any of the base stations at that moment. So this could also be the reason for the loss of the three samples.

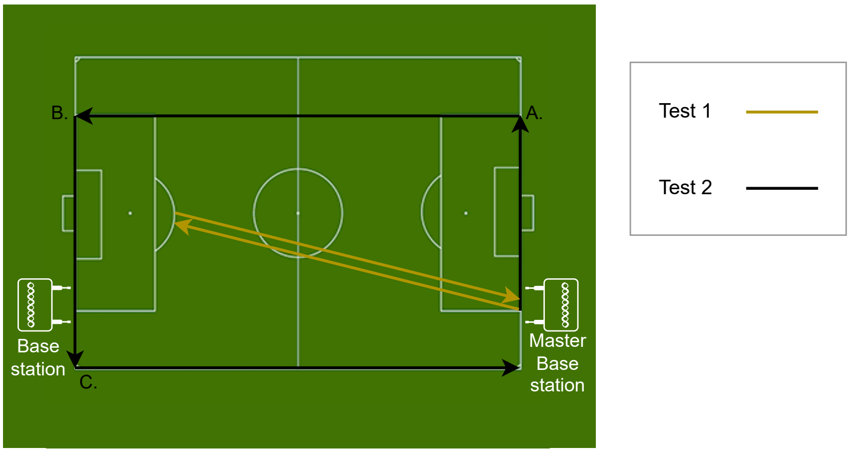


Figure 5.20: Two test scenarios for testing the multi-base station design. For test 2 the turning points are indicated by the letters A, B and C.

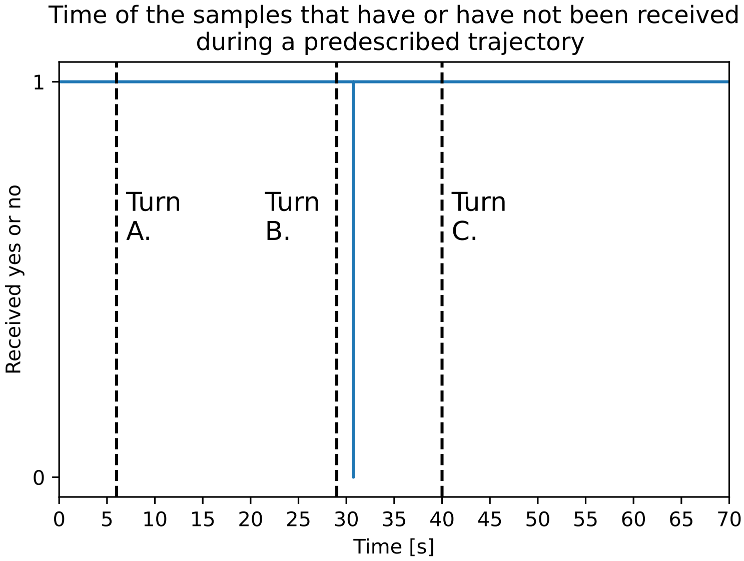
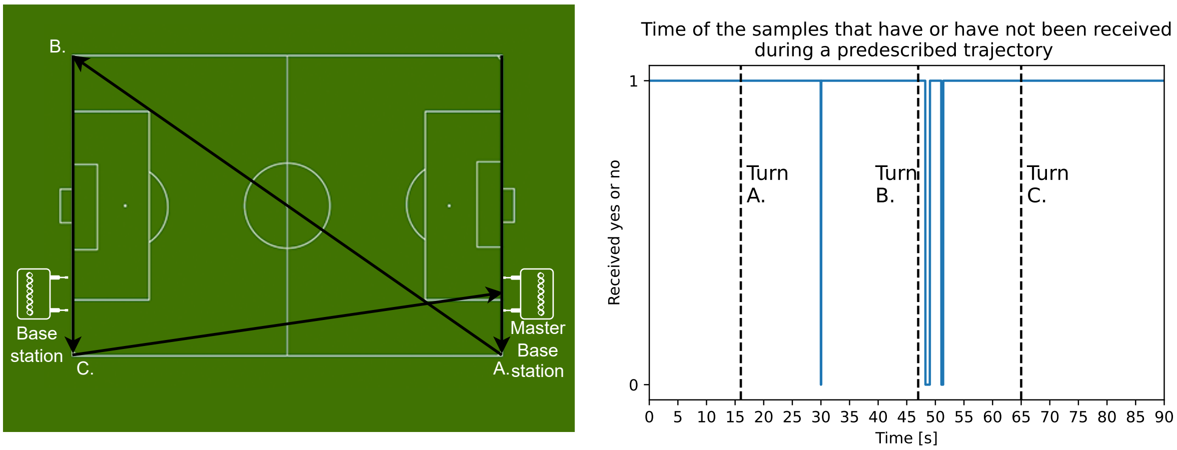


Figure 5.21: The samples being lost during Test 2 (Figure 5.20)

The same results have been analysed for an even more difficult trajectory, called Test 3. Figure 5.22a shows the more challenging course of Test 3. The turning points on the field are noted as A, B, and C. For this course, again, sample loss has been measured. Figure 5.22b shows an overview of the sample receptions.

In Test 3, 298 samples were lost, 1.3% of the total packets expected. Beforehand losses were expected on the path between the start and A and between B and C (Figure 5.22a). Because at those paths, the antenna faces away from the base station. However, the results in Figure 5.22b show no loss between the start and point A. The connection was good enough to transmit all data on this path to the base station. The figure shows an unexpected behavior on the route between locations A and B. Data loss was detected for this route, while no loss was expected.

Table 5.3 shows the samples lost for the different paths. The sensor shorts lost 0.08% of samples on the route between A and B. Apparently, the connection was too weak, while the expectation was that the sensor shorts should be connected to the master base station and have a good wireless connection. It might be that sensor shorts switched from the base station and was not associated with the master base station for a short time. Overall in case, the antenna faces the base station, no data loss is measured. The most considerable loss occurred on the path between B and C.



(a) The trajectory the test person runs for Test 3 in the multi-base station tests. The points A, B, and C are the locations the test person has to turn.

(b) The samples lost during Test 3.

Figure 5.22: Results of a test walking over the field, with on the left the trajectory. The right image shows the result in terms of loss.

Table 5.3: The number of samples lost over the different paths in Test 3 (Figure 5.22a), with the percentage missing of the total expected samples.

	Start - A	**A - B**	**B - C**	**C - End**
Samples lost	0	18 (0.08%)	280 (1.2%)	0

These results show that a multi-base station setup is needed to cover the entire field, and there should always be a base station in the 180 degree plane of the patch antenna. Therefore the systems should consist of four base stations, one on each corner of the field.

6 Conclusion

This thesis developed a system to establish reliable wireless communication for the smart sensor shorts project. For the wireless connection between the sensor shorts and central server, an access point is needed. The base station relays the data from the sensor shorts to the central server and vice versa. The connection between the server and sensor shorts is based on TCP. TCP provides reliable and error-checked data transmission. The data transmission tests showed that this is indeed the case. When a sample has been received, it is always correct.

It is crucial to consider the location of the patch antenna on the sensor shorts. The patch antenna must not experience obstruction from the person wearing the shorts. The arm of the test person could cause attenuation of 10 dB of the signal. The antenna will be placed on the back of the athlete's body to prevent attenuation of the signal due to his own body. The patch antenna's radiation plane is perpendicular to the test person's back.

To test the sensor shorts' line-of-sight reach, the RSSI has been measured for every 5 meters up to 60 m. From the results obtained, the 2-Ray model seems to be a good model to describe the RSSI measured, as it follows the trend of the measured data points. From the obtained fitted model and the receiver sensitivity of the AP, it seems that for the LOS cases, the sensor shorts should be able to reach the base station from every point on the field.

However, during a football game, the players move around freely, and the chances are minimal that the radiation area of the patch antenna is perpendicular to the base station. Therefore RSSI measurements have been performed over the field with different orientations. It became apparent that the RSSI decreases when the patch's antenna radiation area points more and more away from the base station. The worst RSSI at a specific location is measured when the antenna completely faces the opposite direction. The difference between facing an not facing the base station is 18 dB. In our design, this has to be taken into consideration.

Another factor that could affect wireless communication during a game is the obstruction caused by the players on the field. The tests showed that having a base station at 2.2 m is more favorable than a base station at lower heights. At 2.2 m, less attenuation is measured due to the human body. So the height of the base stations will be 2.2 m.

The RSSI measured on the field varied from −77 dBm to −33 dBm. The RSSI value does not tell much about wireless data communication regarding delay and losses. The losses and delay have been measured for different RSSI values to know what the measured RSSI values mean. From −70 dBm, the communication starts to experience the effects of a weak connection. The connection needs retransmissions to ensure the correct delivery of data. From −80 dBm and lower, the connection is so weak that samples are being lost. With 34% of samples lost at −85 dBm.

Around the football field, a minimum RSSI of −77 dBm is measured without obstructions. The average attenuation caused by obstruction is 5 dB on average. Due to the attenuation of human bodies, it is possible to have RSSI values of under −80 dBm when only one base station is used and one patch antenna.

The use of pattern diversity has been considered on the sensor shorts to prevent a weak connection. However, the antenna switching algorithm on the xPico does not function properly. Another solution investigated to cover the entire field is the multi-base station solution.

The results in subsection 5.4.2 show that the multi-base station solution seems ideal for covering the entire football field. From the results, it became clear that for a stable connection between the sensor shorts and base station, there should be a base station in the 180 degree plane of the patch antenna. For this reason, the setup shown in Figure 6.1 is the recommended setup with the devices chosen. Four base stations will be used, with one on each corner. By doing this, we ensure a base station in the 180 degree plane of the patch antenna for most of the time.

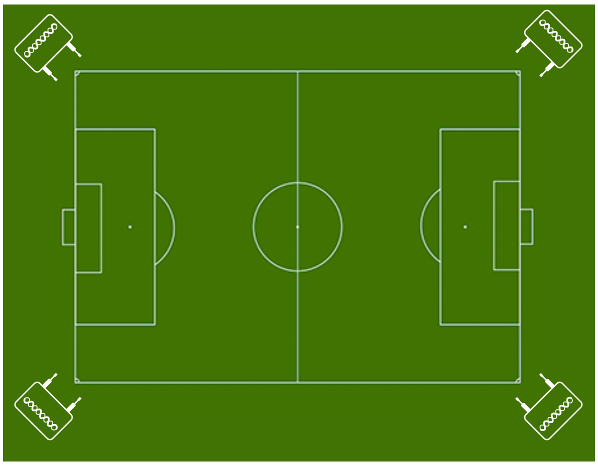


Figure 6.1: For a reliable wireless connection with the sensor shorts, four base stations should be used, one on each corner.

The energy consumption of the sensor shorts is 219 mAh. With the GNG-109 battery, the sensor shorts should be able to last for 11.8 h. A duration of 11.8 h is enough to last a complete football game. We could even look at smaller batteries for the sensor shorts.

With the data rates possible and according to the AP data sheet, a full team could be monitored with the model designed. The AP can deal with more than 220 concurrent clients. A test has been performed with five concurrent sensor shorts; this had no effect on the loss or delays measured.

So for a stable RSSI value, i.e., a value higher than −70 dBm, the data transmission cycle delay will look as shown in Figure 5.14, and there will be no loss. We examined whether having more sensor shorts connect and communication will affect data loss or transmission cycle delay. A test with five sensor shorts didn't seem to affect the transmission cycle delay or loss. According to the datasheet of the AP and WiFi data rates, we did expect the same as was observed. The AP can handle more than 220 clients.

7

Discussion

One of the disadvantages of our system at the moment in case of weak connections is that the sensor data buffer is minimal. When a wireless transmission takes longer than 2s, data will be lost because the buffer on the sensor shorts can save a maximum of 2s of sensor data. There are multiple ways to store more data, which means that there will be less loss. Currently, the accelerometer and gyroscope sensor are read out at 250 Hz. The stack that saves the sensor data has a fixed length for each row of 59 bytes, and the stack rows consist of the data from all sensors and its sequence number. The data of all sensors is saved on the stack every time the accelerometer and gyroscope sensor are read out. Even the data of the magnetometer is saved after each readout of the accelerometer and gyroscope sensor while it samples at 100 Hz. Because the sampling frequency of the accelerometer and gyroscope sensor is much higher, old and unnecessary magnetometer data is saved every four ms. By leaving out the old and unnecessary magnetometer data, we could save 18 bytes of the 59 bytes. But the data could only be saved for the rows which consist of old magnetometer data. When sampling for 20 ms, with the first sample at 4ms and the last at 20 ms, five rows of 59 bytes are typically needed. Ignoring old magnetometer data requires only three rows of 59 bytes and two rows of 41 bytes. By ignoring old magnetometer data, we have 12% more memory available, which means we can save 2.277s of data instead of 2s, equal to 69 more samples that could be saved.

Besides, at the moment, memory is used for the SD card data. If we use this memory for the buffer used for the wireless connection, we have 16384 extra bytes available. The 16384 extra bytes are equal to 1.265s additional sensor data that can be saved in the buffer, similar to 316 samples. So by not using the SD card memory and not storing the old magnetometer data, the sensor should be able to save up to 3.542s of sensor data, equal to 885 samples instead of 500.

We could also look into making the code more efficient in terms of memory. Besides, we could use the compressions algorithm from the previous prototype. Using the compression algorithm, we should have at least 23% more memory available, meaning that approximately 4.3s of sensor data can be stored in the buffer.

By increasing the size of the buffers, we could decrease the packet loss. So it is essential to consider the previously mentioned suggestions for improving the buffer size.

In this thesis project the pattern diversity on the xPico did not function properly. A solution might be to have an antenna splitter that splits the output for the antenna connection on the xPico into two outputs. Or another transceiver should be found which supports pattern diversity or MIMO. The splitter will result in a decrease of 3 dB transmit power, because it divides the power over the two antennas. Another specification of the transceiver to consider for the choice of a transceiver is the buffer size for the serial data coming from the sensor shorts. Because the xPico has only space for 1420 bytes. Because of the small buffer on the xPico, we need large buffers on the sensor shorts.

By having the xPico support pattern diversity or radiating using two antennas in case of the antenna splitter, two base stations might be sufficient. When each base station is placed at the halfway line, there is always a base station in the 180 degree plane of the antenna.

Another thing that might be considered for further research is having the patch antenna higher on the body and having the base station at a higher height. By placing the two more elevated, the systems might experience less obstruction from the athletes on the field.

To make the sensor shorts more user friendly, it is advised to use smaller batteries. With the batteries used at the moment, the sensor shorts could easily last for 11.8h. But the battery's disadvantage is that it is relatively large, making it nearly impossible to fit the xPico, central PCB, and battery in the pocket.

Currently, no tests have been performed with four base stations or with multiple field players. For testing the design during a game first, a mesh network should be established via the Omoda controller provided by TP-Link. Only one of the four base stations needs to be connected with ethernet to the router. But for a more reliable network, it is suggested to have all four base stations connected to the router with ethernet. A new router is needed with 5 LAN ports to connect four base stations over ethernet. One of the LAN ports is used for the central server, but the central server could also connect over WiFi to the AP. For testing, it is advised to look for a tool to trace which AP the shorts are associated with and measure RSSI. These tools can help to analyze and get a better understanding of how the model works.

In this thesis, the effects of interference with other signals have been ignored. For further research, it might be to look at the impact of interference on the performance of our system. For example, in a packed stadium, more interference might be caused by the audience's phones. It would be helpful to know the effects of these interferences. More parameters could affect wireless communication, for example, humidity and temperature. These are also interesting parameters to investigate in later research.

As far as known, this thesis is the first which provides a wireless data transfer system for wearable monitoring sensor devices in football using WiFi. Using WiFi, we can deal with the sensor shorts' relatively large sample rates and can cover large distances. Because of this, four base stations will be enough to cover the entire field. The expected loss will be less than 5% and the delay will be less than 5 s in our system.

A

Testing the antenna selection of the xPico

From the results obtained in subsection 5.4.1, it became clear that the antenna diversity on the xPico didn't work as expected. The xPico didn't select the antenna with the best connection to the base station at certain moments. According to the manual: the xPico will check the received signal strength on each antenna port approximately every 310 ms when the signal level is below −20 dbm and when the average number of received packets is low.

In our scenario, the signal level is always below −20 dbm, so that might not prevent the xPico from selecting the antenna with the highest signal level. So the only other condition that could cause problems is the amount of data traffic. According to the manual, the xPico refrains from switching antenna to maintain throughput when the traffic is too high. Additional tests have been performed to check whether this was the problem during the tests of subsection 5.4.1. The additional tests were performed in the lab.

For the additional test, no data was transmitted to ensure that all the conditions for selecting the best antenna were satisfied. A test person wore the sensor shorts with a patch antenna on the front and back. The access point was mounted on one side of the room during this test. The test person stood on the other side of the room with one antenna facing the base station and the other facing the wall and covered with the test person's hands. By doing this, the antenna which faces the base station will be forced to have the highest signal level.

There is a big difference between the two antennas the antenna facing the base station has an RSSI of higher than −40 dBm, and the antenna facing the wall and covered by hands has an RSSI lower than −50 dBm. Because of this difference, we can easily derive which antenna is selected.

At the start of the test, the back antenna is facing the wall and covered by the test person's hands. After 15 s, the test person turns around and covers the front antenna with his hands. We expect the front antenna to be selected at the start up to 15 s after the start. After the turn at 15 s, the xPico should choose the back antenna.

Figure A.1 shows the result of the test. From the RSSI values, we can derive which antenna is selected by the xPico. At the start, up to 15 s, the RSSI is below -50 dBm, which means the wrong antenna has been selected. Because the back antenna is selected at the beginning instead of the front antenna, this area is colored red. After the turn at 15 s, the back antenna should be chosen. From the RSSI values, it can be seen that, indeed, the back antenna is selected from 15s up to 21 s. At 21 s, the xPico switches to the front antenna, while the back antenna should be selected.

The results show that the pattern diversity on the xPico doesn't work as described in the manual. For pattern diversity, we should look into other devices which can provide this. The xPico can still be used with a single antenna configuration.

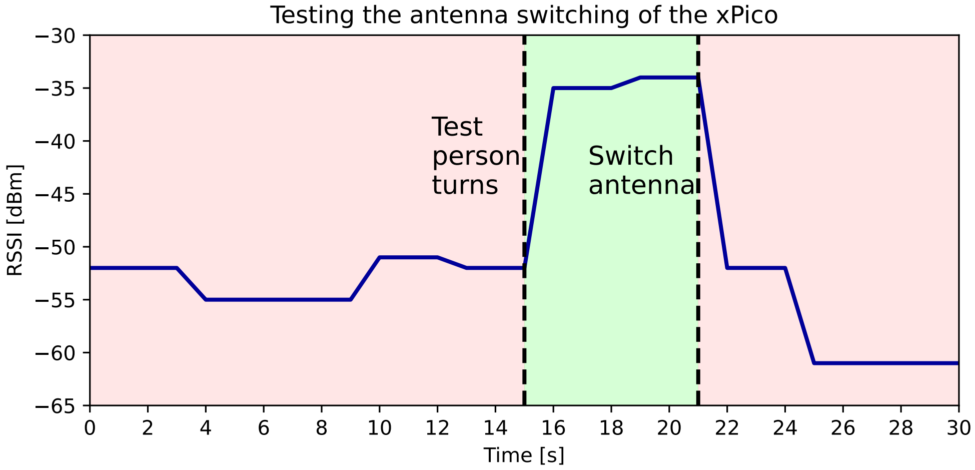


Figure A.1: Test results of the antenna diversity test, at 15 s the test person turn around. Green area indicates the right antenna is selected, red area means wrong antenna is selected.

Bibliography

[1] Thor Einar Andersen et al. "Football incident analysis: a new video based method to describe injury mechanisms in professional football". In: *British Journal of Sports Medicine* 37.3 (2003), pp. 226–232.

[2] S Drawer and CW Fuller. "Evaluating the level of injury in English professional football using a risk based assessment process". In: *British journal of sports medicine* 36.6 (2002), pp. 446–451.

[3] Patricia Wong and Youlian Hong. "Soccer injury in the lower extremities". In: *British journal of sports medicine* 39.8 (2005), pp. 473–482.

[4] Jan Ekstrand, Markus Waldén, and Martin Hägglund. "Hamstring injuries have increased by 4% annually in men's professional football, since 2001: a 13-year longitudinal analysis of the UEFA Elite Club injury study". In: *British journal of sports medicine* 50.12 (2016), pp. 731–737.

[5] Martin Hägglund et al. "Injuries affect team performance negatively in professional football: an 11-year follow-up of the UEFA Champions League injury study". In: *British journal of sports medicine* 47.12 (2013), pp. 738–742.

[6] Jan Ekstrand. *Keeping your top players on the pitch: the key to football medicine at a professional level*. 2013.

[7] Nick van der Horst et al. "The preventive effect of the nordic hamstring exercise on hamstring injuries in amateur soccer players: a randomized controlled trial". In: *The American journal of sports medicine* 43.6 (2015), pp. 1316–1323.

[8] Bastiaan Burgers. *Designing a wireless communication system for smart sensor shorts in football: Using lossless data compression and pattern diversity*. Jan. 2023. URL: http://resolver.tudelft.nl/uuid:d2125dd5-8bc9-47c3-96c3-e8354bd42aeb.

[9] *ICM-20649 Datasheet*. DS-000192. Rev. 1.1. TDK InvenSense. Jan. 2021.

[10] *AK8963, 3-axis Electronic Compass*. MS1356-E-02. Rev. 1.1. AKM Semiconductor. Oct. 2013.

[11] *STM32F411xC STM32F411xE*. 026289. Rev. 7. STMicroelectronics. Dec. 2017.

[12] *Market leading RTOS (real time operating system) for embedded systems with internet of things extensions*. Dec. 2021. URL: https://www.freertos.org/.

[13] *TECHNICAL DATASHEET*. W3230. PulseLarsen Antennas.

[14] Koppala Guravaiah, Arumugam Kavitha, and Rengaraj Leela Velusamy. "Data collection protocols in wireless sensor networks". In: *Wireless Sensor Networks-Design, Deployment and Applications*. IntechOpen, 2020.

[15] IFAB. *Laws of the Game 20/21*. June 2020. URL: https://digitalhub.fifa.com/m/5371a6dcc42fbb44/original/d6g1medsi8jrrd3e4imp-pdf.pdf.

[16] Miikka Ermes et al. "Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions". In: *IEEE transactions on information technology in biomedicine* 12.1 (2008), pp. 20–26.

[17] Deirdre Morris et al. "Wearable sensors for monitoring sports performance and training". In: *2008 5th International Summer School and Symposium on Medical Devices and Biosensors*. IEEE. 2008, pp. 121–124.

[18] Vinski Bräysy et al. "Movement tracking of sports team players with wireless sensor network". In: *2010 Ubiquitous Positioning Indoor Navigation and Location Based Service*. IEEE. 2010, pp. 1–8.

[19] Tsung-Han Liu et al. "Better position for the wearable sensor to monitor badminton sport training loads". In: *Sports Biomechanics* (2021), pp. 1–13.

[20] Chen Xijun, MQ-H Meng, and Ren Hongliang. "Design of sensor node platform for wireless biomedical sensor networks". In: *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*. IEEE. 2006, pp. 4662–4665.

[21] Malek ALRASHIDI and Nejah NASRI. "Wireless Body Area Sensor Networks for Wearable Health Monitoring: Technology Trends and Future Research Opportunities". In: ().

[22] Dennis Pfisterer et al. "Marathonnet: Adding value to large scale sport events-a connectivity analysis". In: *Proceedings of the first international conference on Integrated internet ad hoc and sensor networks*. 2006, 12–es.

[23] Alessio Martinelli et al. "Ultra-wide Band Positioning in Sport: How the Relative Height Between the Transmitting and the Receiving Antenna Affects the System Performance". In: *International Journal of Wireless Information Networks* 27.1 (2020), pp. 18–29.

[24] Vijay Sivaraman et al. "Experimental study of mobility in the soccer field with application to real-time athlete monitoring". In: *2010 IEEE 6th International Conference on Wireless and Mobile Computing, Networking and Communications*. IEEE. 2010, pp. 337–345.

[25] Ashay Dhamdhere et al. "Experiments with wireless sensor networks for real-time athlete monitoring". In: *IEEE Local Computer Network Conference*. IEEE. 2010, pp. 938–945.

[26] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. "Energy-efficient communication protocol for wireless microsensor networks". In: *Proceedings of the 33rd annual Hawaii international conference on system sciences*. IEEE. 2000, 10–pp.

[27] Do-Seong Kim and Yeong-Jee Chung. "Self-organization routing protocol supporting mobile nodes for wireless sensor network". In: *First International Multi-Symposiums on Computer and Computational Sciences (IMSCCS'06)*. Vol. 2. IEEE. 2006, pp. 622–626.

[28] Mrityunjay Singh et al. "A tree based routing protocol for mobile sensor networks (MSNs)". In: *International Journal on Computer Science and Engineering* 2.1S (2010), pp. 55–60.

[29] Deepali Virmani, Tanu Sharma, and Ritu Sharma. "Adaptive energy aware data aggregation tree for wireless sensor networks". In: *arXiv preprint arXiv:1302.0965* (2013).

[30] Andrea Goldsmith. *Wireless communications*. Cambridge university press, 2005.

[31] Shahin Farahani. "Chapter 5-RF propagation, antennas, and regulatory requirements". In: *ZigBee Wireless Networks and Transceivers* (2008), pp. 171–206.

[32] Christoph Sommer, Stefan Joerer, and Falko Dressler. "On the applicability of two-ray path loss models for vehicular network simulation". In: *2012 IEEE Vehicular Networking Conference (VNC)*. IEEE. 2012, pp. 64–69.

[33] Łukasz Januszkiewicz. "Analysis of human body shadowing effect on wireless sensor networks operating in the 2.4 ghz band". In: *Sensors* 18.10 (2018), p. 3412.

[34] Łukasz Januszkiewicz. "Model for ray-based utd simulations of the human body shadowing effect in 5g wireless systems". In: *International Journal of Antennas and Propagation* 2018 (2018).

[35] M Fakharzadeh et al. "The effect of human body on indoor radio wave propagation at 57–64 GHz". In: *2009 IEEE Antennas and Propagation Society International Symposium*. IEEE. 2009, pp. 1–4.

[36] Andrea Duarte Carvalho de Queiroz and Luiz Cezar Trintinália. "An analysis of human body shadowing models for ray-tracing radio channel characterization". In: *2015 SBMO/IEEE MTT-S International Microwave and Optoelectronics Conference (IMOC)*. IEEE. 2015, pp. 1–5.

[37] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. "Wireless sensor network survey". In: *Computer networks* 52.12 (2008), pp. 2292–2330.

[38] Salim Jibrin Danbatta and Asaf Varol. "Comparison of Zigbee, Z-Wave, Wi-Fi, and bluetooth wireless technologies used in home automation". In: *2019 7th International Symposium on Digital Forensics and Security (ISDFS)*. IEEE. 2019, pp. 1–5.

[39] Shadi Al-Sarawi et al. "Internet of Things (IoT) communication protocols". In: *2017 8th International conference on information technology (ICIT)*. IEEE. 2017, pp. 685–690.

[40] *MICAz, WIRELESS MEASUREMENT SYSTEM*. 6020-0060-04. Rev. A. Crossbow Technology, Inc.

[41] *CC256x Dual-Mode Bluetooth® Controller*. SWRS121E. Rev. E. TEXAS INSTRUMENTS. Feb. 2016.

[42] Eko Nugroho, Alvin Sahroni, et al. "ZigBee and wifi network interface on Wireless Sensor Networks". In: *2014 Makassar International Conference on Electrical Engineering and Informatics (MICEEI)*. IEEE. 2014, pp. 54–58.

[43] Shuang Song and Biju Issac. "Analysis of WiFi and WiMax and wireless network coexistence". In: *arXiv preprint arXiv:1412.0721* (2014).

[44] *xPico 200 Series*. 900-818. Rev. H. LANTRONIX. Oct. 2019.

[45] Jim Geier. *Designing and deploying 802.11 wireless networks: a practical guide to implementing 802.11 n and 802.11 ac wireless networks for enterprise-based applications*. Cisco Press, 2015.

[46] James F. Kurose and Keith W. Ross. *Computer networking: A top-down approach*. Pearson, 2017.

[47] *TP-link Omada*. 2021. URL: https://static.tp-link.com/upload/product-overview/2021/202107/20210730/EAP%2020Datasheet.pdf.

[48] Rahul Malhotra, Vikas Gupta, and RK Bansal. "Simulation & performance analysis of wired and wireless computer networks". In: *Global Journal of Computer Science and Technology* (2011).

[49] Piet van Mieghem. *Data communications networking*. Purdue University Press, 2006.

[50] *Select function (winsock2.h)*. Oct. 2021. URL: https://docs.microsoft.com/en-us/windows/win32/api/winsock2/nf-winsock2-select.