Delft University of Technology
Master's Thesis in Embedded Systems

Solar Powered Passive Wireless Moisture Sensor with Cloud Communication

Luis A. Moreira Cardoso

IBM

POSEIDON

embedded software

TU Delft Delft University of Technology

Challenge the future

Solar-Powered Passive Wireless Moisture Sensor with Cloud Communication

Master's Thesis in Embedded System

Embedded Software Section
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology
Mekelweg 4, 2628 CD Delft, The Netherlands

Luis A. Moreira Cardoso
l.a.moreiracardoso@student.tudelft.nl

11^{th} May 2017

Author
Luis A. Moreira Cardoso
(l.a.moreiracardoso@student.tudelft.nl)
Title
Solar-Powered Passive Wireless Moisture Sensor with Cloud Communication
MSc Presentation
11^{th} May 2017
Graduation Committee

Prof. dr. K. G. Langendoen	Delft University of Technology
Dr. R. R. Venkatesha Prasad	Delft University of Technology
Dr. Arjan van Genderen	Delft University of Technology

Company Supervisor

Robert-Jan Sips	IBM

Abstract

Studies have shown that fresh water is getting scarcer worldwide day by day. Most of the fresh water is being wasted on land fields for irrigation. Farmers waste so much fresh water, when watering their crops. To conserve water, measurements need to be done efficiently. However, soil moisture sensors currently available on the market are quite expensive and have high power consumption to be used in developing countries extensively.

Currently, ICT technologies moving at a rapid pace has resulted in a broad domain called, Internet of Things (IoT), which is becoming more and more ubiquitous. IoT is being deployed as part of smart solutions for purposes as remote monitoring and automation, to solve individual as well as social needs.

In this work we present a new solar-powered passive moisture sensor, that is a low-cost alternative moisture sensor with low power consumption. To reduce the power consumption we took two approaches. Firstly, we target to reduce the power consumption during operation and secondly, we aim to harvest solar energy to prevent battery depletion.

The Passive moisture sensor measures the volumetric water content in soil by means of two electrodes composed of different metals, such as copper and zinc. The potential difference between the metals changes with the change of moisture in the soil. Since, the measurements are based on the potential difference between the metals there is no energy consumed by the node. We show that measurements performed by the passive moisture sensor correlates with the measurements performed by commercial moisture sensors, such as, the Decagon EC5. Our passive moisture sensor solution is at least 20% less expensive than other solutions currently available on the market.

To reduce the energy consumption as much as possible, we let both main components of the sensor node, microcontroller and radio, duty cycle separately, since each of these components has its own independent task. Therefore, we have developed the Harmonic-Medium Access Control (H-MAC) protocol, which ensures that the duty cycle of both components do not conflict with each other, and allows the communication between the sensor nodes and the Cloud to be bi-directional. H-MAC makes the network scalable and resilient to failure. With H-MAC the radio has lower duty cycle, lower energy consumption and has lower latency than X-MAC.

Further, to ensure that the batteries of the nodes last for as long as possible we powered the nodes with solar energy. Since, solar energy is lacks consistency and reliability we need to predict when will the sun provide enough energy to power the sensor nodes. We show that by using the weather forecast and inserting information about solar radiation and sunshine duration into our prediction model we get an improvement of the error rate of about 10% on the prediction of solar energy availability.

In this work we provide convincing results that our passive moisture sensor is a low cost, low power alternative for the market. Our solution is affordable for developing countries.

Preface

I always dreamed to make the world a better place by making it smarter. I believe that science and technology is the right path to achieve my dream. With this thesis I complete my Master degree at TU Delft and with that bringing me one step closer to my dream. All the knowledge and experience that I have acquired at TU Delft are of utmost valuable for my journey here after.

When I was employed at IBM, I went to a presentation given by Robert-Jan Sips. In this presentation he showed how the world is running out of fresh water. He stated that most of the fresh water is wasted in land field by farmers. Farmers in developing countries have no means to measure the moisture level in their field. Current solutions on the market are too expensive for them. My grandfather is a farmer in Cape Verde, Africa. Cape Verde has two seasons; dry season, in the first half of the year; and rain season in the second half of the year. He also does do not have any tool to measure the moisture level in the soil. Therefore, he ends up using too much water when irrigating his field. Fresh water in Cape Verde is scarce, often my grandfather does not have enough water for the whole dry season.

Hence, I decided to join this project as part of my thesis. With this work I want to open doors to a new way of thinking of lower cost alternatives of moisture sensors. And expand my idea to other fields as well.

Acknowledgements

I would like to thank the people that supported me during my thesis. First and for most I would like to thank my supervisor R. R. Venkatesha "VP" for his support and inputs. I would like to thank Robert-Jan for giving me the opportunity to work on this project. I would like to thank Rolf Hut for allowing me to use his lab to conduct part of my experiments. I would like thank Agatha Volkmann and Jan-Jaap Dekkers for providing input at the beginning of my thesis.

I would like to thank Arjan van Genderen for his coordination and lectures through out my studies. Many thanks to Koen Langendoen and ES Department of TU Delft for the lectures and support during my studies.

And especial thanks to my family for the support in all these years.

Contents

Chapter 1

Introduction

Freshwater is essential for life and industry. We need fresh water to drink, to cook, to clean, and to wash clothes. In the Netherlands each person consumes on an average 119 litre per person per day.

In industry a huge amount of water is used, for example to clean and cool the machinery and in the meat industry 4500 litres water is used to make 500 gram of beef!

In agriculture, farmers use freshwater to irrigate their fields, each day a huge amount of water is used for irrigation.

Figure 1.1 shows the global water footprint between 1996 and 2005, and we can see that agriculture is by far the greatest consumer of water. Thus it is mandatory to reduce fresh water consumption everywhere because only 2.5% of water is fresh water globally.

In this chapter we introduce our motivation of this thesis and what our contributions are. We will show what the current situation is and how we are going to create the desired situation.

1.1 Motivation

Freshwater is getting scarcer, due to climate change and due to growth of the world population [35, 17, 20, 25]. According to studies, one third of the world biggest basins around the world have dried out [27, 32].

Most freshwater is used in land fields by farmers to irrigate their crops.

It is crucial to increase the efficiency of water usage during irrigation, to avoid wastage of water, and to ensure that enough water is being consumed by the crops.

Over-irrigation causes leaching of the fields affecting the health of the crops. Under-irrigation causes the crops to dry out and die [12]. Having said this, a precise moisture sensor is required to monitor the moisture level of the fields and irrigate the crops in a controlled manner.

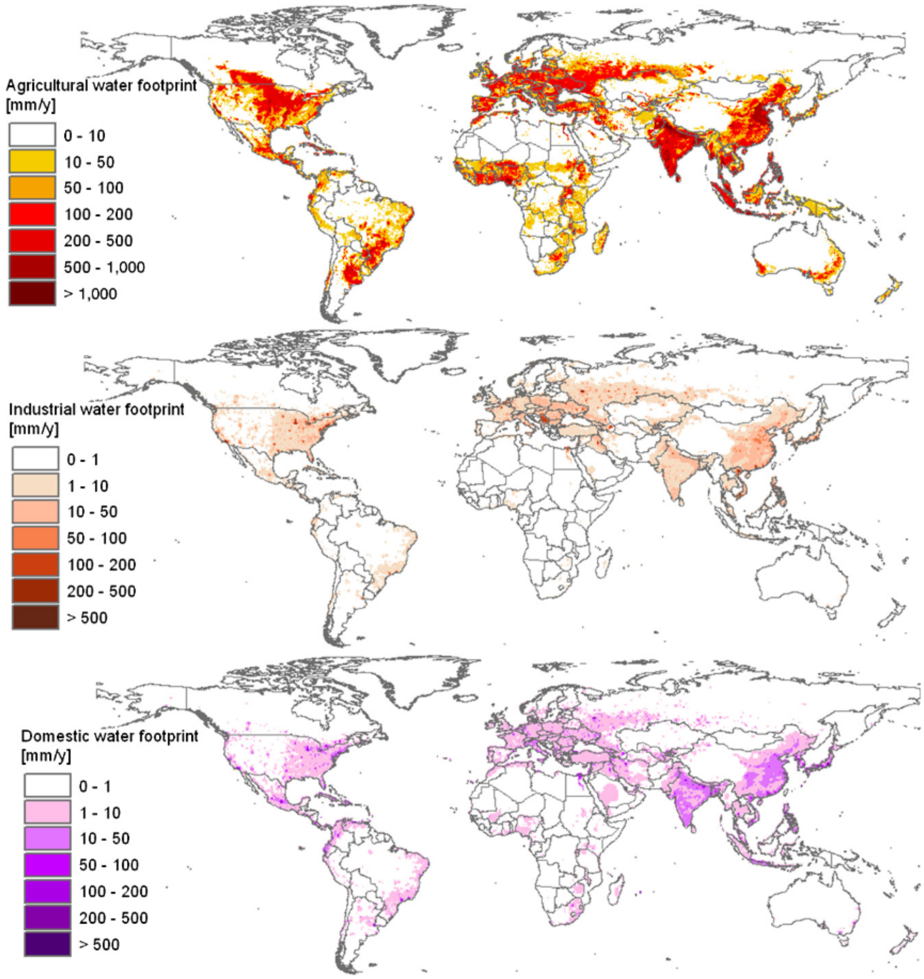

Figure 1.1: The water footprint within nations in the period 1996–2005, shown by sector: the total water footprint of agricultural production (above), the total water footprint of industrial production (mid), and the total water footprint related to domestic water supply (below). The data are shown in millimetre per year on a 5 × 50 grid. Data per grid cell have been calculated as the water footprint within a grid cell (in cubic meters per year) divided by the area of the grid cell (in $10^3 m^2$). [22]

Smart applications and services using ICT technologies are being used in many areas. The field of Internet of Things (IoT), which involves sensing, actuation and data analysis is growing every year. By 2020 it is expected that 50 billion IoT devices around the world would be deployed.

IoT devices often come embedded into products, such as appliances, making those products smarter. They have the ability to monitor the environment and perform computation on the data gathered, or send the data to a base unit via internet [26]. We expect that IoT can be used to help solve many world challenges, such as wastage of fresh water.

1.2 Poseidon

Poseidon is a project started by the IBMer Robert-Jan Sips. The focus of the Poseidon project [2] is to make people aware of the scarcity of freshwater, so that they can adapt the way they treat water, for example by not staying too long under shower and the like. Volunteers in this project give speeches at events and high schools to make people aware of the global shortage of water. Another goal of Poseidon is to provide moisture sensor kits to high school students for educational purposes. With these kits students can built their own irrigation system.

Currently, microcontrollers such as Arduino and Raspberry Pi are at the core of these kits and they are powered by batteries. Poseidon also aims to build a wireless moisture sensor that monitors and controls the irrigation of plants automatically. This away, the amount of water wasted in irrigation can be measured and may be reduced. Unlike, Arduino and Raspberry Pi this new moisture sensor must be powered by renewable energy [33].

Since, this project will be rolled out to third world countries in the future, the new moisture sensor should be affordable, meaning around the same price as Arduino and Raspberry Pi, preferably less expensive.

In this work we present a low-cost passive wireless moisture sensor powered by solar energy with Cloud communication. At the core of this new moisture sensor is a Photon module developed by Particle [9].

1.3 Solution Requirements

Moisture sensors require a high supply voltage, ranging between 1.8V and 5V. Electric batteries that supply power to sensors that run at that voltage range usually have short lifetime. To ensure that the sensor node is online continuously it must be attached to a continuous power source, such as the power grid. To connect the sensor node to a power grid would require the use of cables, but since moisture sensors are placed in the field and underneath the ground and are deployed with multiple nodes, it would be cumbersome to connect each sensor node to the grid by cable. Further, for many areas in Africa grid power is hardly available even.

Most moisture sensors are stand-alone devices meaning they are not connected to any network. This means that to know the soil moisture level at any given point in time one must be on site to make the measurements and report.

When wireless sensors are used they require to be within the range of the gateway.

Farmers from developing countries do not have enough resources, such as devices and electricity to deploy a complete automated moisture sensor network solution.

To provide an affordable wireless moisture sensor with longer lifespan the solution must satisfy the following requirements:

1. Sensor nodes must be wireless, since multiple moisture sensor nodes could be deployed in a field it would be cumbersome to connect them all by wire.
2. Sensor nodes must harvest energy from the environment to ensure long lifespan, since sensor nodes are meant to be put underneath the ground it is convenient to prolong the need of digging the sensor nodes up for battery replacement.
3. The power consumption of the sensor nodes must be minimized as much as possible, to prevent the depletion of the batteries before their next recharge by solar energy.
4. Sensor nodes must be able to communicate with the Cloud, so that external loggers are not required and the user can follow the soil condition from anywhere at any time.
5. Sensor nodes must be affordable, so that farmers in the developing countries can afford to purchase a moisture sensor solution. The solution should be less expensive than an Arduino module.

1.4 Research Goal

The goal of this project is to develop a technique for measuring soil moisture with the lowest-possible power consumption. While increasing the lifespan of the sensor. Thus, the main research question that is raised in this project is:

What hardware and network architecture is most suitable for improving the lifespan of moisture sensors, while keeping the sensor affordable for anyone?

To be able to answer this question, we must first break down the question and answer each sub-question individually.

1. Which method is affordable enough and provides acceptable soil moisture measurements?
2. What method minimizes energy consumption when measuring the moisture level in the soil?

3. What is the best method to maximize the usage of environmental energy?
4. What protocol allows the nodes to communicate with each other with minimal power consumption?

These questions will be answered in chapter 3, 4, and 5.

1.5 Contribution of the Thesis

The main contributions of the thesis are:

1. Building a passive probe made of two different metals, copper and zinc. This passive probe generates energy when it comes into contact with water. This implies that no external power supply is required.
2. Calibrating this low cost sensor with a precision measurement sensor.
3. Energy harvesting with prediction based on forecast, this feature allows sensor nodes to adjust their behavior according to energy available.
4. H-MAC protocol, which takes into account the sampling rate and the transmission rate of the nodes and prevent them from conflicting with each other.

The final result of this project is a moisture sensor that is composed of a passive probe made of two distinct metals, copper and zinc. The moisture sensor is able to communicate with other sensor nodes in the same area with our developed H-MAC protocol that is used for dynamic duty cycling and supports the duty cycling of the microcontroller and the radio. The sensor can connect to the Cloud to transmit its collected soil moisture samples. The connection between the sensor node and the Cloud is bi-directional, upon transmitting its data the sensor node receives the new duty cycle from the Cloud for the next slot. The end-user will be able to access the data via the Cloud.

1.6 Thesis Outline

This work is organised in the following manner: Chapter 2 presents the related work, in Chapter 3 we present the design of passive sensors.

In this chapter we show how to create a passive probe based on two distinct metals, copper and zinc. We show its characteristics and how to calibrate the passive probe. In Chapter 4 we present our work on using H-MAC protocol. We show how H-MAC differs from other MAC protocols.

Chapter 5 discusses prediction of solar energy availability based on forecast. In this chapter we show how we improved the solar energy prediction. We combine everything together in Chapter 6. We conclude this work with a conclusion and future work, in Chapter 7.

Chapter 2

Related Work

2.1 Sensor Nodes in Agriculture

In agriculture sensors can be used to monitor the soil conditions as well as the health of crops. By making use of open-source systems, such as Android, Linux, Apache, MySQL, PHP, Perl or Python (LAMP) [30], monitoring systems can be made cheap. The monitoring system in [30] is based on mesh networks that facilitates real-time data acquisition.

The sensors measure the soil condition and send the information to a central node. The sensors use 6LoWPAN technology. Upon receiving the data from the sensors, the central node forwards that information to the server.

The LAMP server stores the data in a MySQL database and displays it to the clients via the web or the mobile application. Wireless sensor networks (WSN) are suitable for smart irrigation systems [11, 21]. Their system comprises soil humidity sensors to acquire moisture-level information and sends it to the PC, electro-valves that controls the watering of the plants, and the PC that receives the data and stores it in a MySQL database.

WSN improves the water used by monitoring the moisture level in the soil and automatically irrigating the plants with a precise amount of water when the moisture level has passed a pre-determined threshold. The sensor is powered by photovoltaic panels to recharge the batteries and eliminate the need of replacing batteries.

Drip irrigation is another system where WSN can be applied to improve irrigation efficiency. The authors in [18] use a base station unit (BSU), valve units (VUs) and sensor units (SUs) as the main components of their system, where SUs collect data about the moisture level in the soil and send it to the BSU, wirelessly. The SUs are powered by photovoltaic panels. The BSU receives the data from the SUs and computes whether the valves should be open or stay close. The BSU is responsible for controlling the state of the VUs.

The advantage of the above discussed solutions is that they automate the irrigation of the plants and help preventing water wastage. The disadvantage is that the solutions make use of conventional probes that consume energy. And the data is only accessible locally, since they are stored on a local server, that implies that the user needs to be on site to be updated about the condition of the soil. In the next section we discuss different types of methods to measure soil moisture.

2.2 Moisture Sensor: Overview

The level of moisture in soil is expressed as volumetric water content. Water content is the quantity of water contained in soil. Water content expresses the ratio between a completely dried soil and a saturated soil. The ratio for dried soil is expressed as 0 percent, while the ratio for saturated soil is expressed as 100 percent. Soil saturation means that all pores of the soil are filled with water.

In this section we present different types of techniques [19, 44] to measure the water content in the soil.

2.2.1 Gravimetric Water Content

Gravimetric water content, is a oven-drying technique that implies taking a sample of the soil, weigh it and then put it in the oven to let it dry for about 24 hours. When the soil is completely dried, it is weighed again to compare its weight against the measurement before drying the soil and determine the moisture in the soil. This technique is used to calibrate other soil moisture measurements techniques. Equation 2.1 shows how the gravimetric water content is calculated.

The advantage of the oven-drying technique is that the measurement is accurate; is independent from salinity and soil type; and the calculation is simple.

The disadvantages of the oven-drying technique is that it is a destructive test, since a sample of the soil has to be removed from the field; it takes about 24 hours before getting the results; it requires manual action; and it requires a lab, proper equipment and knowledge about the subject.

$$\text{GravimetricWaterContent} = \frac{\text{MassOfWetSoil} - \text{MassOfDrySoil}}{\text{MassOfDrySoil}} * 100 \tag{2.1}$$

2.2.2 Volumetric Water Content

Volumetric water content, is an oven-drying technique like the gravimetric water content, but it requires an extra step. It requires to measure the bulk density (2.2), beforehand. The bulk density is the expression of mass of dry soil per unit volume of soil. It is commonly expressed in grams per cubic centimeter of soil. It relates to the porosity and compaction. Porosity is the void space in soil and compaction is the filled space with moisture in soil. The volumetric water content can then be calculated by using the bulk density and gravimetric water content (2.3).

$$\text{BulkDensity} = \frac{\text{MassOfDrySoil}}{\text{VolumeOfSoil}} \tag{2.2}$$

$$\text{VolumetricWaterContent} = \frac{\text{BulkDensity}}{\text{DensityOfWater}} * \text{GravimetricWaterContent} \tag{2.3}$$

2.2.3 Resistive Sensor

This method can be applied in two manners, the first manner is by measuring the resistance between two electric probes connected to a porous material that is made of gypsum or fiberglass. When buried in the soil this material allows moisture to flow in and out. This moisture flow changes the resistance between the probes, from this change the volumetric water content can be derived, after calibrating the sensor.

The second manner is to measure the resistance of the soil by using two electric probes and stick them directly in the soil. By applying a DC current on one of the probes the resistance can be read on the other probe. Resistance of the soil changes according to soil moisture level.

The advantage of this method is that it offers a low cost solution for measuring the volumetric water content. And it does not need to be removed from the soil.

The disadvantage of this method is that each porous material has a small different characteristic, which implies that each unit must be calibrated individually. The probes have a shorter life time than the capacitive probes.

2.2.4 Capacitive Sensor

This sensor uses two electrodes plates and measure the changes in the dielectric of the soil between the plates. The dielectric constant is directly proportional to the moisture level in the soil. Most soils such as sand, clay and organic matter, have a dielectric constant from 2 to 4. Whilst water has dielectric constant of 78.

The advantages of the capacitive sensor are, it has instantaneous response time; it can deliver measurements with high precision; and it is suitable for remote solutions. The disadvantage of this sensor is that its stability might become compromised in the long term.

2.2.5 Neutron Probe

Neutron scattering emits neutrons from a radio active source into the soil. Neutrons collide with hydrogen atoms in the soil moisture that makes them bounce back to a detector. The number of neutrons counted is linearly related to the moisture content in the soil.

The advantages of this method are, it allows to measure a large soil volume; it is non-destructive; and it delivers measurements with high precision. The disadvantages of this method are, it is expensive; it adds the risk of radiation hazard.

2.2.6 Tensiometer

A tensiometer measures the tension of the soil moisture. It comprises a sealed water-filled tube, and a vacuum gage on the upper side and a porous ceramic cup on the bottom. The water contained in the tube moves through the ceramic cup to the soil when the soil is dry. Or water can move into the tube through the ceramic cup when the soil is wet. The in and out water flow changes the vacuum in the tube. This change can be read from the gage.

The advantages of this technique are, it is affordable; it does not need to be removed; it is not destructive.

The disadvantages are, limit range up to 80 centibar that makes it not suitable for drier soil conditions; it requires periodic service; it is not easy to translate data to volume water content.

In Chapter 3 we introduce a new method of measuring soil moisture.

2.3 Prediction Methods: Overview

In this section we present techniques for the prediction of solar energy availability.

2.3.1 Forecast-based prediction

Predicting the solar energy availability implies collecting weather data and create an energy model. The authors in [36, 37] collected and analyzed extensive traces of past forecast and observational weather data from the National Weather Service (NWS), as well as fine-grain solar and wind energy harvesting and observational weather data from their own deployment. Since, we are only concerned about solar energy we will continue this part only discussing their work on solar energy. They used the traces to quantify how well weather forecasts and the immediate past predict the weather phenomena - sky condition- that most impact solar energy harvesting at time-scales ranging from 3 hours to 72 hours. They found that NWS forecasts in the regions they examine are a better predictor of the future than the immediate past at the time-scales for sky condition.

They used observational data to correlate (I) weather forecasts for their entire region with their own local weather observations and (II) their own local weather observations with the energy harvested by their deployed solar panel. They used the data set to formulate a simple model that predicts how much energy their solar panel will harvest in the future given weather forecasts every 3 hours from 3 hours to 72 hours in the future.

They base their model for solar energy on a simple premise: if the sky condition reports a cloud cover of N% then the observed solar radiation, as well as our solar panel's power production, will be (100-N)% of the maximum possible under ideal cloudless skies. For example, if the 3 hour forecast predicts a sky condition with 50% cloud cover, and the maximum possible solar power production is 60 watts over that 3 hour interval, then the solar power prediction for that 3 hour interval will be 60*0.5=30 watts. The authors incorporated cloud coverage into their solar energy prediction model using their weather station's traces of solar radiation in 3 steps:

1) Computing Solar Power From Solar Radiation: They first derive the relationship between the solar radiation their weather station observes and the power their solar panel produced using the data trace. They used linear regression to convert the solar radiation observed by their weather station to the solar power produced by their solar panel, where power is in units of watts and solar radiation is in units of watt/m^2, Equation 2.4.

$$\text{SolarPower=0.0444*Radiation-2.65} \tag{2.4}$$

2) Computing the Maximum Possible Solar Power: They next derived the estimate for the maximum solar power possible at a given time of the day and year. They use a profile for a single sunny day in each month of the year as the baseline for computing the ideal maximum power on any day of that month. The authors observed that the predicted power harvested—using (2.4)—appeared quadratically related to time of the day, whereas around noon the highest power peak was reached. Therefore, they fitted a quadratic function (2.5) to their observation, where a, b and c are the parameters of the quadratic function. The authors predetermined the values of the parameters for each month, since the daylight hours change throughout the year, the quadratic function changes accordingly.

$$\text{Maxpower} = a * (\text{Time} + b)^2 + c \tag{2.5}$$

3) Solar Model: They created their solar-energy prediction model (2.6), where the cloud coverage is incorporated by the parameter SkyCondition that is provided by the forecast.

$$\text{SolarPowerPrediction=MaxPower*(1-SkyCondition)} \tag{2.6}$$

In the simulation the model (2.6) showed better performance than PPF (Past Predict Future) model. With the model (2.6) less than 5% of the days one node depleted its batteries, while with the PPF model nearly 50% of the days at least one node depleted its batteries. The PPF model is based on the previous weather condition that is used to predicted the future solar energy availability.

The advantage of predicting solar energy is that one can prevent the depletion of the batteries. The only disadvantage that we noticed in the model is that it relies on the forecast of cloud coverage. Cloud coverage does not tell whether the clouds will be covering the sun or not. Different days can have the same cloud coverage but different sunshine duration. In Chapter 5 we show our improvement in this area.

2.4 Medium Access Control: Overview

The main concern of MAC protocols for WSN is to reduce energy consumed by the radio of battery-powered sensor nodes. Along the years many MAC protocols have been developed to satisfy different requirements. MACs protocols come in different categories, synchronous and asynchronous and the combination thereof also known as hybrid. What type to deploy is completely dependent on the application.

Synchronous Sensor nodes share their schedule with their neighbors and negotiate when they should be awake for data exchange. By specifying the time when nodes must be awake in order order to communicate reduces the time and energy wasted in idle listening.

Asynchronous Sensor nodes rely on low power listening (LPL) also known as preamble sampling to connect a sender with a receiver. When the sender has data, the sender transmits a preamble that is at least as long as the sleep period of the receiver. The receiver will wake up, detect the preamble, and accept the data. The receiver only wakes up periodically for a short time to sample the medium, thereby limiting idle listening.
The main advantage of asynchronous low power listening protocols is that the sender and receiver can be completely decoupled in their duty cycles.

In this section we present a bit of a history of MAC protocols. We are not going to discuss all MAC protocols, and we will focus more on the asynchronous protocols, since these protocols are related to our application.

2.4.1 S-MAC

(Sensor) S-MAC [43]—published in 2002—is a synchronous one of the first MAC protocols designed to reduce energy consumption. The authors first identified the source of energy waste, which are, idle listening, collision, overhearing and control overhead. The main features of S-MAC are, (i) periodic listen and sleep, (ii) collision and overhearing avoidance, and (iii) message passing.

In S-MAC each node wakes up periodically and check if there are any node that wants to transmit data to it. It is thus required that sensor nodes stay synchronized.

To keep the nodes synchronized two methods are used, (i) all timestamps that are exchanged are relative rather than absolute, (ii) the listen period is significantly longer than clock error or drift. Nodes exchange their schedules by broadcasting it to its neighbors. When multiple nodes want to access the same node they make use of the RTS (Request To Send) and CTS (Clear To Send) packets.

Periodic listen and sleep Sensor nodes are in idle for long time when no sensing event happens. S-MAC lets the nodes go into periodic sleep.

Schedules The node chooses randomly a time to go to sleep, when it doesn't receive any schedule from other nodes during its listening period. It immediately broadcasts its schedule in a SYNC packet.
When the node receives a schedule from its neighbor before choosing a random time, it agrees on that schedule by setting its schedule to be the same.
When the node receives a schedule from its neighbor after choosing a random time, it adopts its own schedule and that from its neighbor.

Synchronization Nodes need to update each others schedule periodically, to prevent long-time clock drift. The update is done by sending SYNC packets. The SYNC packet contains the address of the sender, and the time of its next sleep. Receivers adjust their timers immediately after receiving the SYNC packet.

Collision Avoidance Since multiple nodes may want to communicate with the same receiver at the same time, sensor nodes need to content for the same medium to avoid collision.
To avoid collision S-MAC uses the RTS/CTS mechanism. It includes both virtual and physical carrier sense. In the virtual carrier sense the node stores a duration field that is transmitted in each packet indicating the transmission duration, in an variable NAV (Network Allocation Vector). This value decreases with time. Each time the node has data to send, it checks if NAV has value zero, meaning that the medium is free.
In Physical carrier sense happens at the physical layer by checking the channel for potential transmissions. The sender determines if the medium is free when both virtual carrier sense and physical carrier sense indicate that it is free.

Overhearing Avoidance When nodes hear an RTS or CTS packet that are not meant for them they will go immediately to sleep, so they will not overhear unwanted DATA packets.

Message Passing The long message is fragmented into small fragments and transmitted in burst. Hereby only RTS packet and one CTS packet is sent. These packets ensure that the medium of the receiver is free before starting the transmission. For each transmitted fragment the sender waits for the ACK packet from the receiver. If the sender does not receive the ACK packet from the receiver it immediately re-transmit the same fragment.

The authors evaluated their protocol on a testbed nodes. They showed that the traffic is heavy when the message inter-arrival time is less than 4s. S-MAC consumes less than half the energy consumed by protocol 802.11. S-MAC achieve energy saving mainly by avoiding overhearing and efficiently transmitting long messages.

2.4.2 T-MAC

(Timeout) T-MAC [41]—published in 2003—is a synchronous protocol and is an improvement on S-MAC, it reduces the idle listening time of the sensor nodes by transmitting all messages in bursts of variable length, and sleeping between bursts. The length of the bursts are determined dynamically.

In T-MAC every node wakes up periodically, to exchange data with its neighbor. To avoid collision and increase the transmission reliability the scheme RTS, CTS, DATA, ACK is used during communication.

The node only listens and transmits during its active period, outside this period they are at sleep. An active period end when no activation event has occurred for a time TA. An activation event is: (i) the firing of a periodic frame timer; (ii) the reception of any data on the radio; (iii) the sensing of communication on the radio; (iv) the end-of-transmission of a node's own data packet or acknowledgement; (v) the knowledge, though overhearing prior RTS and CTS packets, that a data exchange of a neighbor has ended.

With T-MAC the authors achieved in their implementation as much as 96% of energy saving comparing with S-MAC.

2.4.3 B-MAC

(Berkeley) B-MAC [31]—published in 2004—is an asynchronous carrier sense media access protocol for wireless sensor networks that provides a flexible interface to obtain ultra low power operation, effective collision avoidance, and high channel utilization.

With B-MAC when a sender wants to transmit its data it turns on its radio and sends a long preamble. The preamble size is at least as long as the sleep time. When a receiver wakes up, it checks if there are any senders willing to transmit their data by sampling the medium for preambles. If there is a preamble detected, then the node will receive those messages and stay awaken until the transaction is finished. After reception the receiver checks if there are any other message to be received. If there are no messages left the node goes back to sleep.

The authors claim that B-MAC is suitable for monitoring applications and its flexibility allows other services and applications to be realized efficiently.

To avoid collision this protocol employs software automatic gain control for estimating the noise floor, because ambient noise changes according to the environment.

B-MAC is configurable, which makes it flexible and scalable. It can be adapted to network conditions. This way the protocol can always provide optimized power consumption, latency and throughput to support services that rely on it.

The authors showed that B-MAC surpasses other protocols such as S-MAC and T-MAC in terms of energy savings, latency and throughput. Their protocol had over 98.5% packet delivery.

2.4.4 X-MAC

X-MAC [15]—published in 2006—is an asynchronous low power listening MAC that employs a short preamble to reduce energy consumption as well as to reduce latency.

X-MAC introduces a series of short preamble packets each containing target address information, thereby reducing the overhearing problem of low power listening, saving energy on non-target receivers.

X-MAC inserts pauses into the series of short preamble packets, creating a strobed preamble that enables the target receiver to shorten the strobed preamble via an early acknowledgment, thereby achieving additional energy savings at both the sender and receiver, as well as a reduction in per-hop latency.

The sender sends a short preamble containing the address of the target node. This gives the overhearing neighbors means to discover that they are not the target node and thereby they are allowed to turn off their radio.

The short preamble is followed by a short idle time, on which the sender waits for a acknowledgement from the receiver. If the sender does not receive the acknowledgement of the receiver it sends a short preamble again, and will repeat this process until it receives an acknowledgement from the target node. After receiving an acknowledgement the transmission takes place, after that the sender goes immediately to sleep, whilst the receiver waits a little while to sense if there are any more packets to be received.

The duty cycle of X-MAC is adaptable to the traffic load of the network. Energy consumption remains relatively constant as network density increases.

The authors evaluated their protocol by implementing it on top of Mantis Operating System (MOS). X-MAC reduces latency by 50% on a chain topology of 8 nodes. It receives approximately 90% or more of the packets for all network densities and sleep times. X-MAC uses 10% less energy than B-MAC.

2.4.5 Ri-MAC

RI-MAC (Receiver-Initiated-MAC) [38]—published in 2008—is an asynchronous protocol and it works in the opposite of X-MAC, the transmission is always initiated by the intended receiver. When the sender needs to transmit its data, it wakes up and stays idle waiting for the receiver to wake up.

Each receiver has its own schedule, and wakes up periodically according to its schedule to check if there is any data to be received. The receiver does so by immediately broadcasting a beacon notifying the neighbors that it is awake and that it is ready to receive a data frame.

Upon receiving the beacon from the receiver the sender starts transmitting its data frame. The reception is acknowledged by the receiver after the transmission. The acknowledgement has two roles, (i) it acknowledges the correct receipt of the sent data frame, (ii) it requests a new data frame transmission to the same receiver. When there is no other incoming data the receiver goes to sleep.

The sender that transmitted the data goes to sleep as well. Other senders that still have to transmit their data to the same receiver perform a CCA (Clear Channel Assessment) check and initiate a retransmission of data.

RI-MAC reduces the amount of time a pair of node occupy the medium before they reach a rendezvous time for data transmission. This implies that RI-MAC increases the capacity of the network and thereby increases the throughput.

The authors evaluated their protocol in TinyOS on MICAz motes. According to the authors, RI-MAC achieves higher throughput, higher packet delivery ratio, and greater power efficiency under a wide range of traffic loads compared to X-MAC.

2.4.6 ORiNoCo

ORiNoCo (Opportunistic Receiver-initiated NO-overhead COllection) [40]—published in 2012—is an asynchronous protocol and is an opportunistic, resource- and energy-efficient collection protocol for sensor networks. This protocol is based on RI-MAC with a path weight added on top of it. This path weight is used to build a tree routing structure to the RI-MAC beacons. The weight represents the cost for sending data to the sink. Upon receiving the beacon from the sender **s**, the receiver **r** calculates the path weight via **s** using the cost metric,

$$\phi_{\mathbf{r},\mathbf{s}} \leftarrow \phi_{\mathbf{s}} + \kappa_{\mathbf{r},\mathbf{s}} \tag{2.7}$$

Whereby, $\phi_{\mathbf{s}}$ is the weight carried by the beacon from **s** and $\kappa_{\mathbf{r},\mathbf{s}}$ is a measure of the cost (calculated by **r**) for sending a packet to **s**. Each **r** saves solely the best known path weight that is updated only if a path with lower weight is found. Initially, $\phi_{\mathbf{r}} = \infty$ for all nodes **r**, except for sink k that is always $\phi_{\mathbf{k}} = 0$.

During initialization phase, a node **r** that has not yet received a path weight from a neighbor is in the initialization phase : **r** switches on its radio and starts listening to the wireless channel. When **r** receives a beacon from a neighbor **s**, **r** calculates an initial weight using (2.7). Once node **r** has accepted a beacon and assigned a value to $\phi_{\mathbf{r}}$, **r** proceeds to the packet transmission phase and starts the RI-MAC protocol. The initial assignment of weights shapes an implicit tree-like routing structure that is not necessarily optimal in term of ϕ.

During the packet transmission phase, each node broadcasts beacons with an inter-beacon time chosen randomly from interval $[(1-\alpha) * T_{slp}, (1+\alpha) * T_{slp}]$, where $0 < \alpha < 1$. This interval is a generalization of the inter-beacon time used with RI-MAC.

A node willing to transmit data starts listening to the channel and waits for a beacon from an arbitrary node **s** satisfying

$$\phi_{\mathbf{r},\mathbf{s}} \leftarrow \phi_{\mathbf{s}} + \kappa_{\mathbf{r},\mathbf{s}} \leq \phi_{\mathbf{r}} \tag{2.8}$$

that guarantees routing progress towards the sink. If (2.8) holds, node **r** transmits its packet to **s** and waits for an acknowledging beacon from **s**. Only if that beacon is received, **r** updates its weight $\phi_{\mathbf{r}}$ with $\phi_{\mathbf{r},\mathbf{s}}$. The goal is to prevent **v** from adapting to a path with low cost but poor connectivity. If no acknowledging beacon is received, **v** waits for the next beacon (possibly from a different node).

In order to favor links with high delivery probability , link-quality estimates, such as LQI or RSSI, should be used to ignore beacons from neighbors offering a poor link.

ORiNoCo enables **r** to forward packets to the first available parent satisfying (2.8), so that the expected waiting time per node is decreased. Hence, radio usage and therefore energy-consumption are reduced, while packet latency is decreased.

The authors evaluated their protocol in a simulation ORiNoCo finds a good trade-off between power-efficiency and responsiveness, while maintaining high reliability due to its self-stabilizing property. In scenarios of single-sink data-gathering the protocol outperforms CTP and passes the test on average power consumption of 0.3mW while providing and end-to-end latency below 2.5s for a network of 200 nodes.

2.4.7 RoCoNo

RoCoNo (Receiver-initiated Opportunistic Data Collection and Command) [34]—published in 2015—is an asynchronous protocol and is an extension of ORiNoCo [40] that reconfigures subsets of nodes during runtime. The main purpose of RoCoNo is to improve the collection of data in distributed sensor network and improve the dissemination of control commands.

While ORiNoCo is developed to send data from the node to the sink in the most energy efficient manner, RoCoNo is developed to send data from the sink to a node or a group of nodes without losing the energy efficiency. To make that possible it extends the data packet of ORiNoCo with three fields: address of the destination; command; version. Whereby the addresses of the target nodes are stored in a Bloom filter [14].

A field of one byte is added to each control message. An example of a command could be to change the duty cycle.

To avoid having nodes execute the same command twice a two byte field is added containing the version number of the command. This field is incremented whenever the sink has made any changes to the address of destination field or command field. With the version number it can be identified whether the nodes in the network contain the new command or it needs an update. Since the version numbers are assigned solely by the sink, it guarantees consistency throughout the network.

Potential receivers send beacons periodically, these beacons serve both as an invitation to send data as well as an acknowledgement for data reception. These beacons are leveraged to disseminate control commands throughout the network.

RoCoNo adds a new type of message to the protocol, confirmation message. This message is regular data packet and it acknowledges the sink that the node has received and executed the command.

	B-MAC	X-MAC	Ri-MAC	ORiNoCo	RoCoNo	S-MAC	T-MAC
Scalable	Yes	Yes	Yes	No	No	No	No
Synchronous	No	No	No	No	No	Yes	Yes
Adaptive Duty Cycle	Yes	Yes	Yes	No	Yes	No	Yes
Radio vs Microcontroller*	No	No	No	No	No	No	No

Table 2.1: MAC Comparison: * Radio and microcontroller duty cycle, separately. The microcontroller has a sampling rate that dictates how often to sample the soil in a period of time. The radio has the transmission rate that dictates how often to exchange data in a period of time.

The authors evaluated their protocol in a testbed of 54 TelosB nodes. They showed that RoCoNo consumed on average 1.4mW, that is the same power consumption as that of ORiNoCo. The command reception rate ranged from 96% to 100%. The percentage of command confirmations received at the sink ranged from 99% to 100%.

Table 2.1 summarizes the properties of each MAC, addressed above. Our application requires asynchronous protocols, therefore S-MAC and T-MAC are disregarded. ORiNoCO and RoCoNo both calculate the best path from source to sink, the calculation itself consumes energy and further more the network in our application is configured in chain topology that implies that there is only one path.

We can see that B-MAC, X-MAC and Ri-MAC have the same characteristics. The disadvantage of B-MAC is that it uses long preambles, this results in higher energy consumption. With Ri-MAC, nodes wake up periodically to check whether there is data to be received even when they have no data to send, this also results in waste of energy. With X-MAC the sender only wakes up when it has data to transmit, the energy consumed depends on the next wake up time of the receiver.

X-MAC has closer relation to our application, since our application requires nodes to only wake up when they have data to transmit. Our application also requires nodes to duty cycle the microcontroller as well as the radio and that is not supported by X-MAC. In this work we introduce our protocol that solves the shortcomings of X-MAC.

2.5 Dissemination Protocols: Overview

There are networks where sensor nodes need to be reprogrammed or perform a certain action on demand. To do so, nodes need to be able to receive new code and perform the action required. The dissemination protocol is meant to disseminate data packets containing new code, message or command throughout the network. Often, a data packet is targeted to a single node or a group of sensor nodes. In contrast to the MAC protocol where the data packet is transmitted from source to drain, with the dissemination protocol the transmission of the data packet starts at the drain.

Our application requires the network to be updated with a new transmission rate, periodically. In this section, we discuss several dissemination protocols.

2.5.1 DRIP

The DRIP [39] protocol—published in 2005—is built upon Trickle [28], it allows sensor nodes to disseminate and receive new messages. To accomplish this it provides two interfaces to sensor nodes, i) a transport layer interface, to disseminate messages throughout the network; and ii) a standard reception interface, to receive new messages from the network. New messages contain updates that sensor nodes have to install. Nodes that wish to receive a new message are required to subscribe with a specific identifier on a channel of the source node. Once the source node receives a new message for that specific channel, the message is directly transmitted to all nodes subscribed to that channel. DRIP uses a sequence number to determine whether a received message is new. If the message is new the node installs the new update, otherwise the update is disregarded.

2.5.2 DIP

The DIP (DIssemination Protocol) protocol [29]—published in 2008—is a hybrid protocol between data detection protocol and dissemination protocol. DIP detects and identifies data on the network that has changed. The detection of the changed data is done by means of comparing its own hash with the incoming hash. During identification phase DIP determines whether the incoming data has a newer version number or an older version number. When the income data is a newer version, DIP disseminates the newer data throughout the network to keep the network up-to-date. The new data is installed by all nodes that still have the older version of the data.

2.5.3 DHV

The DHV (Difference detection, Horizontal search and Vertical search) protocol [16]—published in 2009—detects the code that has changed by comparing at the bit level solely the last few least significant bits of its version number with that of the incoming version number. This implies that it is not required to transmit the whole version number through the network for comparison. In comparison with DRIP and DIP, DHV transmits less data through the network.

To find out which bits of the currently installed code differ from the incoming code, DHV operates in two phases, i) detection phase, in this phase DHV detects whether both codes differ or not by comparing the hash of its currently installed code with the hash of the incoming code; ii) identification phase, in this phase DHV identifies which bits of the code differ by means of comparing the checksum of both codes. Once the bits have been identified, only these bits will be transmitted to the node. The node on its turn will update its code with the new bits upon reception of the bits.

The above discussed dissemination protocols allow sensor nodes to disseminate and receive new messages containing updates. Nodes can install these updates, but before installing the updates each node has to find out whether the receiving update is new or not. The operations performed by the nodes to determine whether the incoming data is new, take time and consume energy.

In our application it is obsolete to check if a new incoming message contains a new update, since all packets received from the Cloud are regarded as a new update. Therefore, we disregard these protocols in our work.

2.6 Dynamic Duty Cycle

Solar energy is not constant, to cope with this fact harvesting sensor nodes should only be active when enough energy is available to prevent them from depleting their batteries. This means that harvesting sensor nodes should adapt their duty cycle according to the availability of solar energy.

The authors in [23] present their adaptive duty cycle algorithm. Their algorithm makes use of historical data to predict the future behavior of the energy source.

Their system comprises the following components:

Energy Generation Model This component provides a model of the energy available to the system in a form that can be used for making power management decision.

Energy Consumption Model This component provides detailed information about the energy usage characteristics of the system, at various performance levels.

Harvest Energy Tracking This component makes data measurements. This data is used to predict future energy availability. A good prediction model should have a low prediction error and provide predicted energy values for durations long enough to make meaningful performance scaling decisions.

Energy Storage Model This component represents the model for the energy storage technology. Since, not all the harvested energy may be used immediately, the harvesting system will store this excessive energy in a storage component, such as a battery or ultra-capacitor.

Harvesting-aware Power Management All previous mentioned components provide input to this component so it can determine the suitable power management strategy for the system.

They showed that their algorithm can use up to 58% more of the solar energy compared to the case when harvesting-aware power management is not used.

In the next chapter we will present our work. Our work is built on the works addressed in this chapter. The MAC protocols do not satisfy our requirements which are: i) support scalability, ii) adaptive duty cycle, iii) support multiple duty cycles, and iv) synchronous.

2.7 Summary

In this chapter we have shown what has been developed on the field of irrigation systems, we showed that there are different methods to measure the volumetric water content in a soil. All these methods consume energy. This is not convenient for a sensor node, since they deplete the batteries and thereby decrease the lifetime of the sensor node.

We showed that solar energy prediction is based on solar radiation and cloud coverage. The behavior of the clouds are unpredictable when the sky is not overcast, meaning that one cannot predict when a cloud will be covering the sun and when not. This makes difficult to create a solar energy prediction model.

We also presented various MAC protocols that do not support multiple schedules inside one single node. A sensor node consists of two main components namely the radio and microcontroller, each of these components need to duty cycle according to their own schedule.

Dissemination protocols allow the network to be updated with new code. To accomplish that, a dissemination protocol performs several operations to detect and identify the new update. These operations are time and energy consuming. Our application does not require to detect and identify new updates.

In the next chapters we will present our work that improves the techniques discussed above. In Chapter 3 we present our passive probe that does not consume any energy when measuring the soil moisture level.

In Chapter 4 we present our MAC protocol that we have developed, this protocol takes into account the duty cycles of the radio and the microcontroller.

In Chapter 5 we present our solar energy prediction model that trumps the model based on solar radiation and cloud coverage.

Chapter 3

Sensor Design

One of the requirements for our moisture sensor is that it must be powered by renewable energy, rather than batteries. There are numerous renewable energy technologies, these technologies generate electricity with no impact on the greenhouse effect. Electricity generated by renewable energy technology is considered as “green”. The current list of commercialized renewable energy consists of solar, wind, hydro, biomass, geothermal.

Solar energy can be harnessed to generate electricity by converting solar light or artificial light to electricity; Wind energy converts the wind power to electricity; Hydro energy converts water force into electricity; Electricity can also be generated from biomass and heat.

For the purpose of this project solar energy technology is most suitable, since it is available on the market in smaller scales with higher performance and low price.

This section presents the solar-powered passive wireless moisture sensor with Cloud communication.

3.1 Hardware

For this project we used 10 Photon microcontrolers from Particle [9]. The microcontrollers contain a WiFi module for wireless connection with the internet.

The passive moisture sensor probe is composed of two different metals, copper and zinc. Our passive probe attaches to the microcontroller with the zinc plate always connected to the ground pin and the copper plate is always connected to the one of the analog pins.

Each microcontroller is powered by a monocrystalline solar panel that produces 5V with 400mA of current. Since, solar energy lacks consistency and reliability we added a rechargeable battery pack to our solution. The battery pack can be recharged by the solar energy.

Figure 3.1 shows the components of our wireless moisture sensor solution.

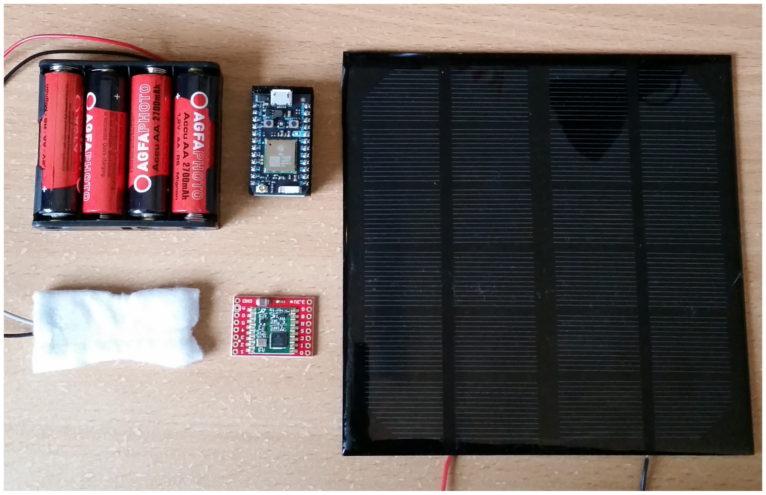

Figure 3.1: Prototype Moisture Sensor. On the top left are the batteries to supply energy to the sensor node when no solar energy is available. On the left bottom the passive probe. At the centre top Photon microcontroller. At the centre bottom RFM69. And on the right the solar panel to harvest solar energy.

	Voltage	Current
Operation (Wi-Fi on)	3.6V	100mA
Operation (Wi-Fi off)	3.6V	40mA
Sleep	3.6V	2mA
Deep Sleep	3.6V	100μA

Table 3.1: P0 Particle Power Consumption Specification

3.1.1 P0 Particle Module

P0 is a module from Particle [9], with the purpose to simplify the prototyping of internet of things (IoT) applications. It allows wireless internet connection via its on-board Broadcom BCM43362 WiFi chip, 802.11b/g/n. P0 also contains a powerful STM32F205 120MHz ARM Cortex M3 microcontroller, with 1MB flash, 128KB RAM. It consists of an on-board status LED that shows which mode P0 is operating. It runs on FreeRTOS real-time operation system. P0 requires a power supply between 3.6V and 5.5V. It consumes on average 100mA during normal operation.

P0 is also available as a PCB solution, called Photon. Photon is open source, and can be integrated in any product.

	Voltage	Current
Receive	3.3V	16mA
Transmit (+13dBm)	3.3V	45mA
Sleep	3.3V	1μA

Table 3.2: RFM69 Power Consumption Specification

3.1.2 RFM69 module

The module RFM69 module from Hoperf [4] is used for local communication between the nodes. This module operates at 868 MHz and is connected to the Particle P0 module via SPI in slave mode. It runs on 3.3V that is supplied by P0 Particle through one of its digital pins. It has a power output capability of +20dBm at 100mW. This module has a range of about 400 to 500 meters. RFM69 consumes about 60% less energy than the WiFi chip.

3.2 Potential Difference of Metals

Each type of electrode has its own atomic composition having different numbers of atoms and electrons. The difference in the number of electrons creates a potential difference between two electrodes [13]. An electrode is an electrical conductor, such as metals.

Electrodes are categorized as anode and cathode. The electrons depart from the anode into the cathode. This is also known as the redox reaction. Redox reaction is short for reduction-oxidation, and it refers to the chemical reaction in which the states of atoms are changed.

The chemical reaction involves the transfer process of electrons from anode into cathode. This process is called oxidation and reduction and it happens when both electrodes are dipped in a electrolyte.

An electrolyte is a solution that conducts electricity, such as water.

The electrode undergoing oxidation is losing electrons, while the electrode undergoing reduction is gaining electrons.

Figure 3.2, shows the prices and the standard electrolyte potential of different metals. The standard electrolyte potential is the redox reaction of metals with respect to standard hydrogen electrode that forms the basis of the thermodynamic scale of oxidation-reduction potentials.

We chose two metals, copper and zinc due to their high availability in the market, low cost and they have good redox reaction with each other.

3.3 Passive Probe

Different metals such as zinc and copper have different electric potential. This property can be harnessed to measure moisture level in the soil. By putting two different metals plates against each other separated by a isolator, the voltage

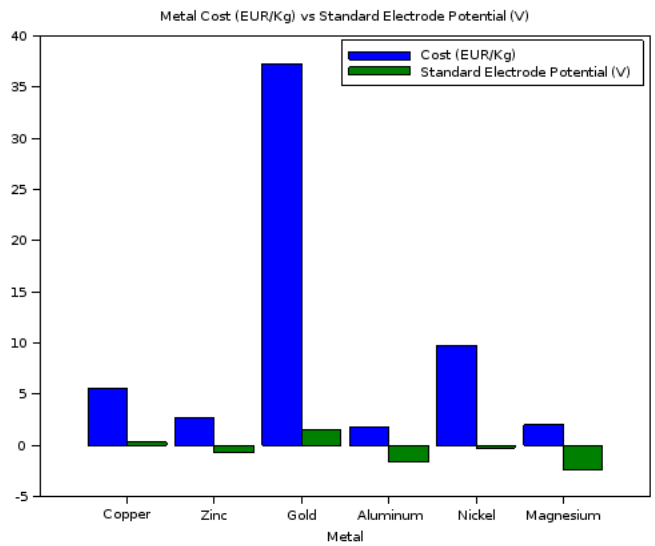

Figure 3.2: Metals comparison

difference between the two metal plates will change when the isolator gets wet by the moisture in the soil.

The isolator is made of a materials that absorbs moisture well, such as paper or cotton. The passive probe is a galvanic cell that generates electricity, whereas the electrodes are the two metal plates and water acts as an electrolyte.

To build such a passive probe we first need to gather a piece of copper, zinc and cotton, as shown in Figure 3.3. The copper metal is always connected to the analog input pin of the microcontroller, while zinc metal is always connected to ground. We then put both metals against each other separated by a piece of cotton, Figure 3.4. The piece of cotton should be big enough to wrap both pieces of metals, Figure 3.5.

The soil moisture level can be deduced by analyzing the change in the potential difference between the metal plates. This implies that the passive probe makes it possible to do measurements without the need of any power supply, as conventional moisture probes do.

The more moisture the isolator absorbs the higher the voltage difference between the two metal plates. The maximum voltage measured on the passive probe is 0.85V at 1mA.

Figure 3.3: Copper, Zinc, and cotton

Figure 3.4: Zinc is connected to ground. Copper is connected to analog pin. Cotton separates both metals.

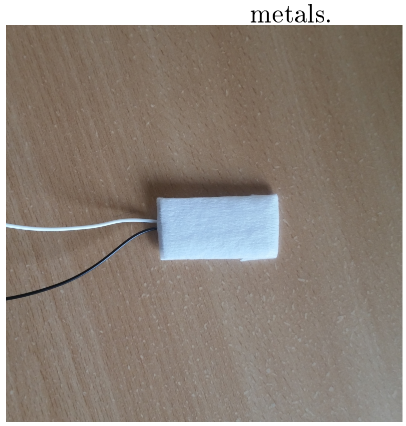

Figure 3.5: Both metals are wrapped with cotton

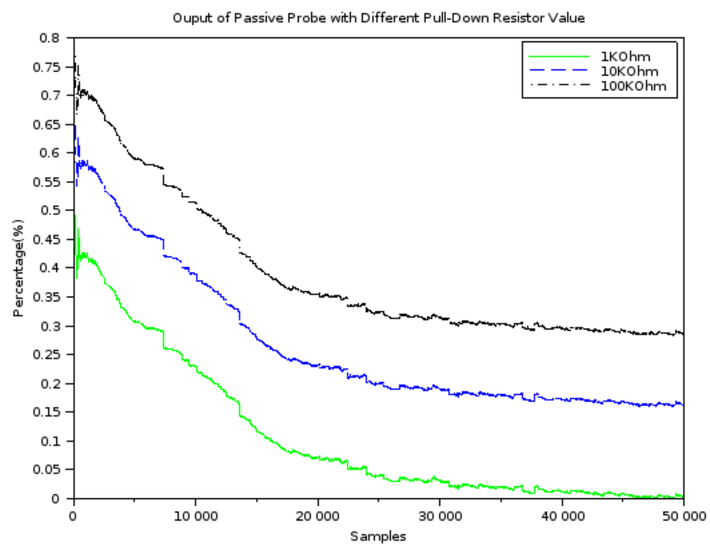

Figure 3.6: Passive Probe comparison of different value of pull-down resistors

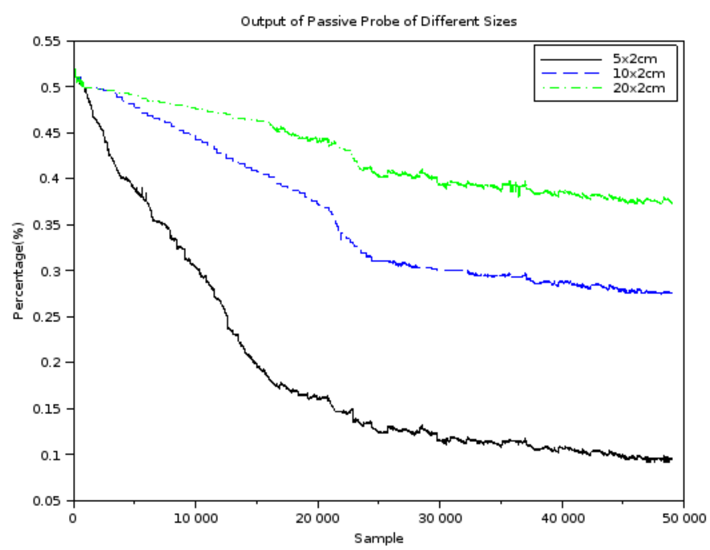

Figure 3.7: Passive Probe of different sizes

3.4 Passive Probe Characterization

This section shows the characterization of the passive probe. We conducted a experiment whereby we put our 5x2 cm passive probe under the soil in a sandbox of the size 30x30x15 cm. The probes where connected to Photon microcontroller that reads the input values from the passive probes. Photon was connected to a Raspberry Pi for data storage.

To find out if there is any difference in the size of the resistor connected between the ground and the analog pin to which the probe is connected to, we connected the passive probe on three pull-down resistors, 1 KOhm, 10 KOhm, 100 KOhm.

Figure 3.6 shows the plot of our experiment. We can see that the output of the probe changes with the size of the resistor. The higher the resistor the higher the output of the probe. However a resistor of 100KOhm gives the highest output.

The reason for the change in the output is because the resistor prevents electrons from flowing between ground and the analog pin. The higher the resistor less electrons will flow through it to the analog pin, and thus more electrons will flow through the passive probe. See Appendix A, for more details about the design of the sensor.

The size of the probe also affects the output of the probe, as shown in Figure 3.7. Probes with larger area take more moisture in and therefore need more time to lose it.

3.5 Experimental Setup

The experiment was conducted in the Water Management Lab at TU Delft, Figure 3.8, and it lasted for about 10 weeks. We attached the passive probe to Photon microcontroller that would read in the values from the passive probe and stored on the Raspberry Pi to which Photon was connected to.

For comparison a Decagon Soil Moisture Sensor EC5 was used in the same experiment. Decagon soil moisture sensor measures the volumetric water content that tells the percentage of moisture in the soil.

This additional sensor is already calibrated by the manufacturer Decagon [1]. Decagon EC5 was connected to the logger HOBO. Both sensors were buried around 10 cm deep in a 100x100x30 cm sandbox. The probe was kept for 10 weeks buried in sand. During that period the sand was repeatedly watered and let dried, several times.

To increase the drying process we installed a fan next to the sandbox to ventilate the environment.

Figure 3.9 shows the results obtained from the experiment. The graph shows the volumetric water content measured by EC5 sensor. And the voltage generated by the passive probe in percentage, s, that is the measured voltage, v_m, divided by the maximum possible voltage, v_{max}, that can be generated by the passive probe, Equation 3.1. In the first week of the experiment we did not

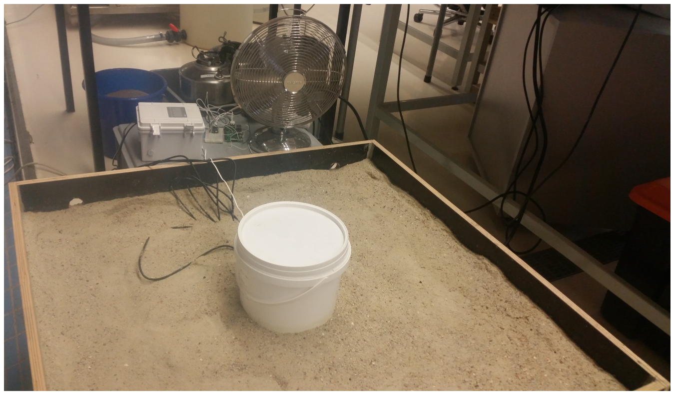

Figure 3.8: Lab at TU Delft

connect an capacitor between the input pin and ground, which caused some noise on the output. The noise can be seen on the left part of Figure 3.9 on the output of our passive probe. The graph shows that the reaction time of the passive probe to a change in the soil condition is equal to the reaction time of EC5 probe. The value s can be inserted in Equation 3.2 to determine the volumetric water content.

$$s = \frac{v_m}{v_{max}} \tag{3.1}$$

3.6 Sensor Calibration

The voltage percentage measured by the passive probe does not tell very much about the amount of moisture in the soil.

Therefore, the passive probe needs to be calibrated, so that the volumetric water content can be derived from the amount of voltage generated by the passive probe.

We calibrated the passive probe by means of linear regression method, Equation 3.2. Linear regression is a simple and often used technique to reveal the relationship between a dependent variable and an independent variable. As shown in Figure 3.10 the values measured by the passive probe correlates with the measures taken by EC5. This means that the passive moisture probe is a good alternative for measuring soil moisture.

$$\text{VolumetricWaterContent} = 2.40s + 0.0087 \tag{3.2}$$

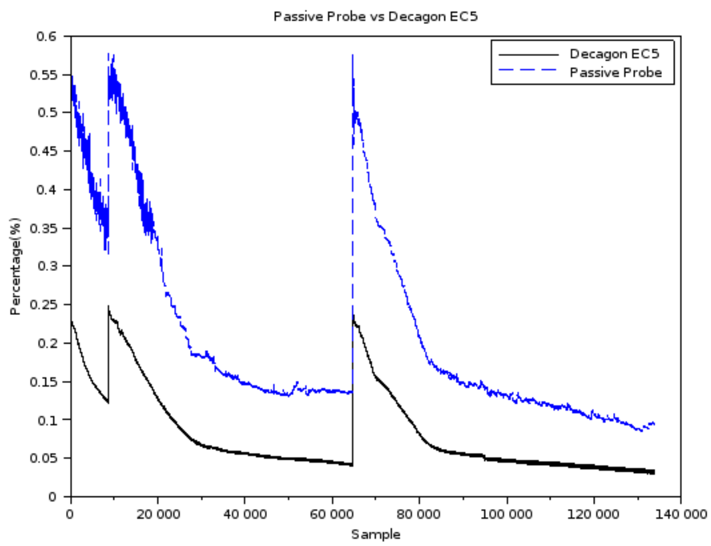

Figure 3.9: Soil Moisture Level: Decagon EC5 vs Passive Probe

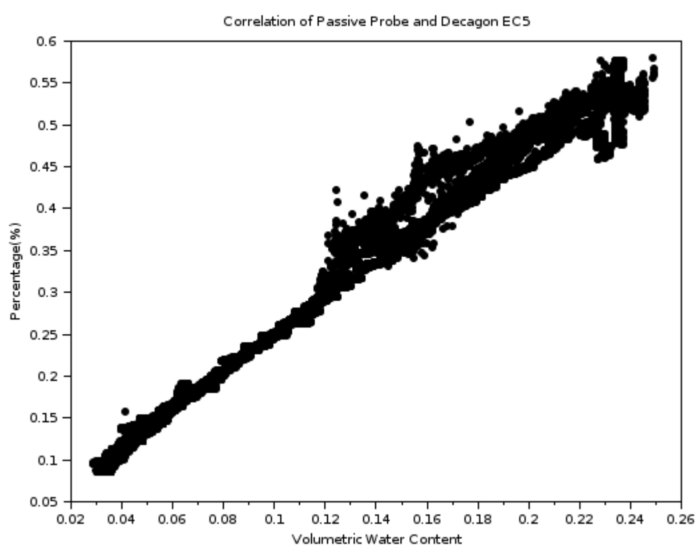

Figure 3.10: Linear Regression: Decagon EC5 vs Passive Probe

Component	Price
Metal Zinc (2*5cm)	€ 0.16
Metal Copper (2*5cm)	€ 0.05
Isolator per piece	€ 0.06
Total	**€ 0.27**

Table 3.3: Passive Probe Cost

Component	P0 Particle	ESP8266	Arduino	Raspberry Pi
Microcontroller	€ 10	€ 10	€ 15	€ 25
Radio WiFi	-	-	€ 5	€ 10
RFM69	€ 3	€ 3	€ 3	€ 3
Probe	€ 0.27	€ 4	€ 4	€ 4
Total	**€ 13.27**	**€ 17**	**€ 27**	**€ 42**

Table 3.4: Price Comparison of Components

3.7 Cost

The cost of our passive probe is extremely low, as shown in Table 3.3, the total cost of its materials is only € 0.27 in total. The two copper and zinc metals are quite inexpensive. They cannot be bought in small custom size pieces, they can only be bought in larger sizes that we have to cut in smaller pieces, afterwards. The smallest piece that can be bought of the zinc metal is a plate of 20 x 20 cm for about € 5. This plate can be cut into 30 to 40 pieces of 2.5x5 cm that brings the price down to € 0.16 per piece. Copper is available as a tape of 2.5 x 1000 cm for about € 10. By cutting it into 200 smaller pieces of size 2.5 x 5 cm brings the price down to € 0.05 per piece. For the isolator we choose wipe surface cleaning that is available on any supermarket as a packet of 80 pieces for about € 5. This brings the price down to € 0.06 per piece. The cheapest moisture probe that we could find on the internet was a Grove moisture sensor that costs, about € 4, that is much expensive than our passive probe.

The price of this passive moisture sensor solution is lower than the current moisture sensors currently available on the market. As shown in Table 3.4, the most expensive component is the P0 module from Particle that comes with a built-in WiFi chip, it costs about € 10. The radio RFM69 is the second most expensive component, it costs € 3.

The ESP8266 module has about the same price as P0 Particle, but it only has a 10-bit ADC, while P0 has a 12-bit ADC. And also it is not possible to duty cycle its microcontroller separately from its radio.

Arduino is slightly more expensive, and other drawbacks are that it requires a 5V power supply, it has a 10-bit ADC, and it does not have a built-in WiFi chip. In sleep mode Arduino consumes 25 mA of current, while P0 consumes 80 μA, and it can operate at 3.6V.

The Raspberry Pi is the most expensive microcontroller in the table, but it also has other drawbacks, such as it requires a 5V power supply, and it does not have any ADC input pin nor a built-in WiFi chip.

3.8 Summary

In this chapter we showed how to build a passive probe consisted of two different metals, such as copper and zinc. We have shown that the potential difference between the two metals correlates with the volumetric water content of the soil. The potential difference increases with the increase of the volumetric water content and decreases when the volumetric water content of the soil decreases. We showed that the value measured by our passive probe can be adjusted by changing the value of the pull-down resistor. And showed also that the characteristic of the passive probe changes with the size of its area.

By applying linear regression we observed that the measures performed by our passive probe correlate with the commercial moisture sensor Decagon EC5. This implies that the passive probe is a good low-cost alternative moisture sensor.

Our passive probe consumes no energy at all. And it costs as low as € 0.27 that brings the cost of the whole solution to be at least 20% less expensive than currently commercial solutions.

In the next chapter we present our protocol to reduce the power consumption during operation time.

Chapter 4

Harmonic MAC (H-MAC) Protocol

Since the sensors should save their energy as much as possible and the availability of solar energy is uncertain it is required to have both the microcontroller and the radio turned off as much as possible.

The challenge with the sensor nodes is that their microcontroller and radio are duty cycling separately from each other. At each cycle the microcontroller takes samples and go to sleep, keeping the sample in memory, whilst the radio allows nodes to transmit their samples to each other and eventually send it to the Cloud. This can lead to a mismatching of duty cycles between the microcontroller and the radio, and having the sensor node missing its turn to receive and transmit samples.

Sensors that are placed in an environment where the moisture level changes drastically will operate at a higher harmonic, meaning they will wake up the microcontroller with more frequency to take samples, the radio stays off until the first harmonic is reached.

The network must be flexible and scalable such that new probes can be added to the network at any given time. And the network must be resilient enough to prevent the network from going offline when a node occurs a failure or is removed from the network. The network must allow messages to be delivered to the Cloud from sensor nodes and vice versa.

To satisfy all these requirements we developed H-MAC. H-MAC is a hybrid of MAC protocol and dissemination protocol. It is similar to X-MAC, it is asynchronous allowing nodes to be loosely coupled, operating independently from each other. This makes the network scalable. And it keeps the network up-to-date by disseminating messages through the network.

However, X-MAC is suitable to nodes of which the processor does not duty cycle while the radio has a fixed duty cycle. H-MAC has a dynamic duty cycle and is suitable for nodes that contains a processor and a radio that duty cycle, separatly. The processor and radio duty cycle in harmony.

And dissemination protocols generally perform all kinds of periodic tasks to discover whether data items on the network has changed. For our application H-MAC only broadcasts updated messages after the sensor nodes have communicated with the Cloud. It is assumed that the message from the Cloud is always new. The message from the Cloud includes for example the new transmission rate of the nodes.

4.1 Forbidden State

Each sensor node contains a microcontroller and a radio. The challenge with the sensor nodes is that their microcontroller and radio are duty cycling separately from each other. At each cycle the microcontroller takes samples and go to sleep, keeping the sample in memory, whilst the radio allows nodes to transmit their samples to each other and eventually send it to the Cloud. This can lead to a mismatching of duty cycles between the microcontroller and the radio, and having the sensor node mis the transmission window. As shown in Table 4.1, at state 0 both microcontroller as well the radio are turned off. At state 1 the microcontroller is turned on while the radio is turned off, in this state the microcontroller is allowed to take a sample of the soil and go back to state 0. The sample is kept in its memory. At state 2, the microcontroller is on as well as the radio. In this state the microcontroller is allowed to take a sample of the soil and immediately after that transmit its samples to its neighbors.

However, at state 3 where the microcontroller is off while the radio is on is an impossible state. The radio is driven by the microcontroller, thus it cannot turn itself on. So, for the radio to turn on the microcontroller must be on and turn the radio on.

H-MAC solves this problem by letting the microcontroller and the radio duty cycle in harmony. This means that the microcontroller will duty cycle at the n^{th} harmonic of the duty cycle of the radio.

In the next section we will discuss how H-MAC synchronizes both microcontroller and the radio.

4.2 Synchronization Between Microcontroller and Radio

The radio is driven by its host, microcontroller. It is important to have them both synchronized to prevent the microcontroller from failing to turn on the radio, and therefore the node will fail to communicate with its neighbors.

H-MAC synchronizes the microcontroller and the radio by means of harmonic duty cycling. The duty cycle of the radio is consider to be the first harmonic. All nodes have the same first harmonic, this is when they transmit their samples to each other. The duty cycle of the microcontroller is a n^{th} harmonic of the radio's duty cycle. This ensures that the microcontroller will always be able to match the duty cycle of the radio, and hence the microcontroller will drive the radio on time.

Each node have a default first harmonic that is 10 minutes. The harmonics are measured in seconds, that represents sleep cycle. For example, a first harmonic of 600 seconds (10 minutes) means that the radio will be turned off for 10 minutes, periodically. The default first harmonic is derived from the data that we gathered in the lab where we conducted the experiment with the passive probe, discussed in Section 3.3. From that data we noticed that the moisture level didn't had any significant change within a period of 10 minutes, so it would be waste of resources to do anything before that time.

The first harmonic can change according to the predicted solar energy availability, this is discussed in the next section. In Section 4.4 we discuss the sampling rate of the microcontroller.

4.3 Radio Transmission Rate

The transmission rate is considered to be the first harmonic at which all nodes transmit their data samples to each other and eventually one of the nodes will transmit all collected data samples to the Cloud. It is thus paramount to have all sensor nodes agreed on the same first harmonic.

The first harmonic is determined on the Cloud and it is determined based on predicted solar energy for the next time slot. On the Cloud weather forecast is used to predict the future solar energy availability with the help of the decision tree.

If the expected solar energy availability is high enough then high transmission rate will be assigned to the moisture sensor nodes. Likewise, if the expected solar energy availability is low then low transmission rate will be assigned to the moisture sensor nodes.

Each time slot has a period of 1 hour. The size of the time slot is chosen based on the weather forecast provided by the online platforms [7, 8]. Online platforms provide forecast from hourly up to few days in the future.

After computing the first harmonic the Cloud sends the results back to the sensor nodes. The response packet include the following fields, transmission rate and maximum sampling rate. These fields are described in minutes.

Algorithm 4.1 shows the complete procedure for computing the first harmonic. The solar energy predicted p for the next time slot is gathered and from that predicted energy the number of possible transmissions by the sensor node is calculated by dividing it by p_t, that is the energy required to transmit one moisture sample. p_t is large enough to cope with the load unbalance.

If the predicted energy p in the time slot is not sufficient for at least one transmission, then the time slot is skipped and the next time slot is fetched. This iteration will proceed until a time slot is found in the future with sufficient predicted energy for at least one transmission by the sensor node.

Once, the number of transmissions t has been computed, the radio duty cycle can be calculated by multiplying the slot duration by the number of slots skipped i and dividing everything by t.

The remaining energy r is calculated. From the remaining energy r the maximum number of moisture samples s that can be taken by the moisture sensor nodes is computed by dividing r by the energy required to take one sample p_s.

Algorithm 4.1 Algorithm calculates the transmission rate and the maximum number of samples that the microcontroller is allowed to generate

```
Input: Current time slot
Output: Transmission period and max samples
  P = {p_1, p_2, p_3, ...}                ▷ Energy predicted per slot
  i = 1
  for p in P do       ▷ Calculate number of transmissions possible in next slot
      t = ⌊p/p_t⌋
      if t == 0 then
          i++
          skip slot
      end if
      d = slot_size(i)/t                  ▷ Calculate duty cycle of radio
      r = p − (t ∗ p_t)                   ▷ Calculate remaining energy in this slot
      s = ⌊r/p_s⌋                         ▷ Calculate max number samples allowed
       return d, s
  end for
```

4.4 Microcontroller Sampling Rate

The sampling rate is the rate at which the moisture sensor is sampling the soil. Each sensor node samples the soil at its own rate, meaning that they do not need to take other nodes into account.

However, within each sensor node the microcontroller needs to take into account the transmission rate of its co-existent radio, as discussed in Section 4.2. Thus, the sampling rate must be a n^{th} harmonic of the transmission rate.

The minimum sampling rate is 10 minutes. The minimum sampling rate is derived from the analysis that we have done on the data that we gathered from the lab, as discussed in Section 3.5. From the data we saw that within the period of 10 minutes there is no significant change in the volumetric water content of the soil.

The sampling rate changes according to the condition of the soil. If the soil is drying quickly then a higher sampling rate will be applied. Likewise if the soil is losing moisture slowly then a lower sampling rate is applied.

The maximum sampling rate is determined by the amount of solar energy predicted. Algorithm 4.1 shows the procedure how to compute the sampling rate as discussed in the previous section.

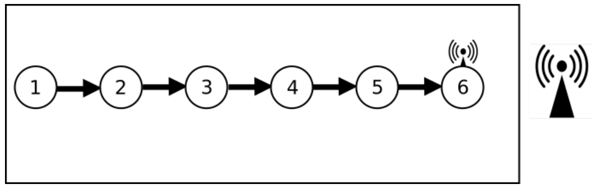

Figure 4.1: Topology: Each node sends its data to the next node. Each next node collects the data from the previous node and forward it along with its own data to the next node in the chain. The last node in the chain collects all the data and sends it to the Cloud. The nodes in the network only wake up when the previous node is ready to transmit, that implies that nodes further down the chain are allowed to sleep a bit longer, saving their energy.

4.5 Cloud

The Cloud receives the samples from the sensor nodes and makes it available for the user in a visual manner. The user can access the information from anywhere at anytime from any device. And eventually take any action required.

The Cloud gathers the latest weather forecast and determines if the next hour there will be enough solar energy available to power the sensor nodes. The Cloud responds with a packet containing information about solar energy availability, a multiplier of the first harmonic, Figure 4.2. The Cloud data packet contains the fields: node id, transmission rate, back-off time, maximum sampling rate, and drain. The node id corresponds to the id of each node on the field, the transmission rate field contains the transmission rate for all nodes, back-off time field specifies the back-off time for each node. Nodes are not allowed to sample more that what is specified in the maximum sampling rate field. The drain field specifies which node is allowed to communicate with the Cloud.

The user can access the Cloud to get informed about the current soil condition.

4.6 Initialization procedure

In this application we walk through the field and turn on each node, sequentially and bury it underground. As shown by Figure 4.3 when node n_i is turned on, at time t_0, for the first time it starts sending intermittent RTS signals. If the node does not find any neighbor within 15 seconds, then it jumps to listening mode. The next node n_{i+1} that wakes up and send intermittent RTS signals, but now n_i receives the RTS from n_{i+1}. Node n_i acknowledges the reception of the signal by replying with ACK, telling that it is ready to receive the data packet from n_{i+1}. Upon reception n_i goes to sleep, while n_{i+1} goes to listening mode, and wait for the next node to wake up. After 15 seconds n_{i+1} will go to sleep if no other node has been detected. If n_{i+1} is the last node in the network,

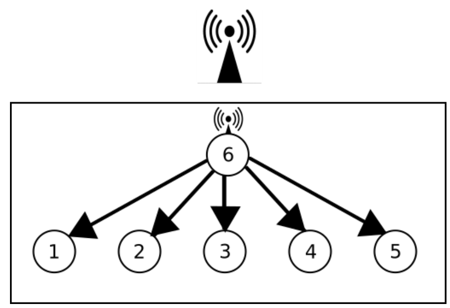

Figure 4.2: Cloud Response: The node communicating with the Cloud receives the response from the Cloud. Since, the response packet from the Cloud must be shared among all nodes, it is broadcasted throughout the network.

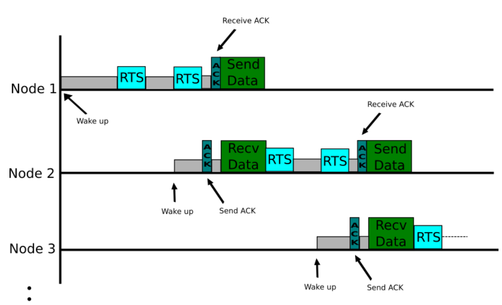

Figure 4.3: H-MAC Initialization

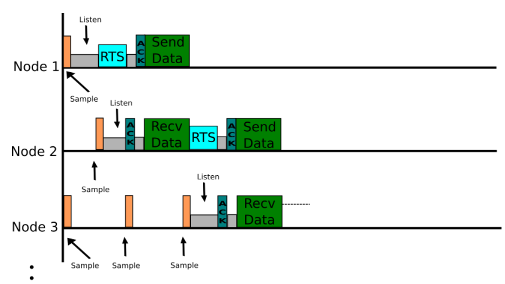

Figure 4.4: H-MAC Steady State

this node will be the node that will communicate with the Cloud.

RTS packet contains the packet type such that receiving nodes can determine what action to take upon receiving the RTS packet. The packet type field specifies the type of the packet the node wishes to send that can be an initial data packet, data sample packet, or cloud response. The node can respond with an ACK or ignore it. If the node is expecting the response from the Cloud it will ignore all other packets sent by its neighbors.

The initial data packet is a token containing fields such as, node id, and rssi value. The node id is used by the Cloud so it knows which samples belong to which sensor node. With the rssi value of each node, the Cloud knows which nodes have access to internet, and thus are able to connect to the Cloud. The initial data packet and sample data packet are received by the nodes and forwarded to the neighbor, while the Cloud data packet is read and information corresponding to the node is retrieved. Data sample packet contains the fields: node id, sample and rssi value.

4.7 Steady State

During steady state nodes turn on their microcontroller regularly to take samples. The wake up frequency of the microcontroller is a n^{th} harmonic of the wake up frequency of the radio.

In steady state the nodes wake up their radio at the first harmonic, one after the other like in the initialization phase but with shorter listening time. This manner collision can be prevented. As shown by Figure 4.4, when they are awaken they share their sample data with each other. The sample data will travel from the first node until the last node in the network. The last node transmits the sample data to the Cloud.

The Cloud receives the data samples from the node and responds with the prediction of solar energy availability for the next hour.

The sampling rate is dynamic and the transmission rate is dynamic as well. This can lead to a situation where the schedule of the microcontroller and the schedule of the radio conflict with each other. The different states of the both schedules are shown in Table 4.1, whereas state 0 means sleeping mode and state 1 means awaken mode. As can be seen at state 3, the microcontroller is sleeping and the radio is turned on. This is clearly not possible, the radio can only be turned on when the microcontroller is turned on. So state 3 has to be prevented. As shown by Figure 4.2, the nodes start at state 0, that is when they are turned off or at sleep. From there they jump between state 2 and state 1.

To avoid state 3, nodes have to agree on a first harmonic at which they will transmit their data packet. The first harmonic is determined by the energy availability at that moment. The sampling rate is determined by the soil condition at that moment. The sampling rate is always a multiple of the first harmonic, this ensures that conflicts between schedules of microcontroller and radio do not occur.

State	Microcontroller	Radio
0	0	0
1	1	0
2	1	1
3	**0**	**1**

Table 4.1: Sampling rate vs Transmission rate

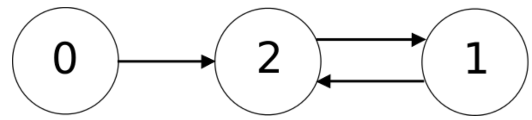

Table 4.2: State Diagram

4.8 Collision Avoidance

The network is configured in a chain topology, because in our application the number of nodes that will be in the network is uncertain as well as the number of nodes within the range of the WiFi gateway. We assume that there is always at least one node with the connection to the Cloud. To avoid collisions among sensor nodes a back-off time is calculated by the Cloud for each node. Each node will be assigned with its own back-off time. From the very first time the nodes communicate with the Cloud, they will receive a response packet containing the back-off time and the transmission rate. When nodes wake-up for transmission they will wait for the period of its back-off time. The back-off time dictates the order in which nodes will transmit their data.

The Back-off time is calculated by Equation 4.1, where b_i is the back-off time that will be assigned to node i, t is the amount of time that it takes to transmit a single sample, s is the total number of samples that can be generated by the nodes during the sleeping period of the radio, calculated in Algorithm 4.1.

$$b_i = ts + b_{i-1} \tag{4.1}$$

4.9 Network Resiliency

Sensor nodes will be put in a field where people and machinery constantly move around. Since, the sensor nodes are buried underneath the soil people will be hardly aware of their presence. This implies that sensor nodes might be stepped upon, which can bring damage to the node. It might happen that the node at the end of the network chain fails to report to the Cloud due to failure, or a node has been removed from the network. It might also be the case that some nodes have no connection to the Cloud, because their are out of range of the WiFi gateway. We assume that in the network there is always at least one node with access to the Cloud.

To cope with such situation neighbor nodes that are waiting for the response will communicate with the Cloud, after the timeout expires. The Cloud will be aware that there is a node missing and report to the user. The remaining nodes receive their response back from the Cloud that contains the message in which order the nodes must wake up to ensure that at least one node can report to the Cloud.

4.10 Network Flexibility and Scalability

It might happen that sensor nodes are deployed in a field with already an existing network. This implies that at first the new sensor node will not be able to communicate with the rest of network. Assuming that this new node has access to the Cloud, it will receive a response from the Cloud specifying its transmission rate, such that it will be able to communicate with the sensor nodes in the already existing network.

4.11 Experimental Setup

H-MAC was implemented in each node based on Photon that runs on FreeRTOS. FreeRTOS is an open source, real time operating system developed by Real Time Engineers Ltd [10].

Each node can turn off their WiFi radio by calling WiFi.off() and turn it back on by calling WiFi.on(). To put the entire node into sleep mode we have to call System.sleep(SLEEP_MODE_DEEP, seconds). During the sleep mode data will be kept in EEPROM. The RFM69 module can be put to sleep as well by configuring its 8-bit register RegOpMode with the values 000 on bits 4-2. RFM69 can be awaken in transmitter mode as well as in reciever mode with the corresponding configuration of the bits 4-2 on its register RegOpMode, 011 and 100, respectively. Further details about the modules are described in Section 3.1.1 and Section 3.1.2.

4.12 H-MAC Performance

To evaluate the performance of H-MAC we conducted several experiments in a chain topology. The experiments were performed in different locations to satisfy the requirements of each experiment. To evaluate H-MAC on nodes in different distances we have conducted our experiment in a open field with a length up to 500 metres in Amsterdam near IBM's headquarters building. During this experiment we solely used 2 sensor nodes, because more than that we would require more space and time. All other experiments related to network size were performed on a field in Rotterdam. For this experiment we created a network up to 10 sensor nodes.

In our application nodes performed a maximum of 6 sample measurements and 6 transmissions, in a period of one hour. This implies a maximum of one sample and one transmission per 10 minutes. These values are derived from the data gathered from the experiment conducted in the TU Delft lab at the water management department, as discussed in Section 3.5, we observed that the volumetric moisture content decreased slightly every 10 minutes, before that

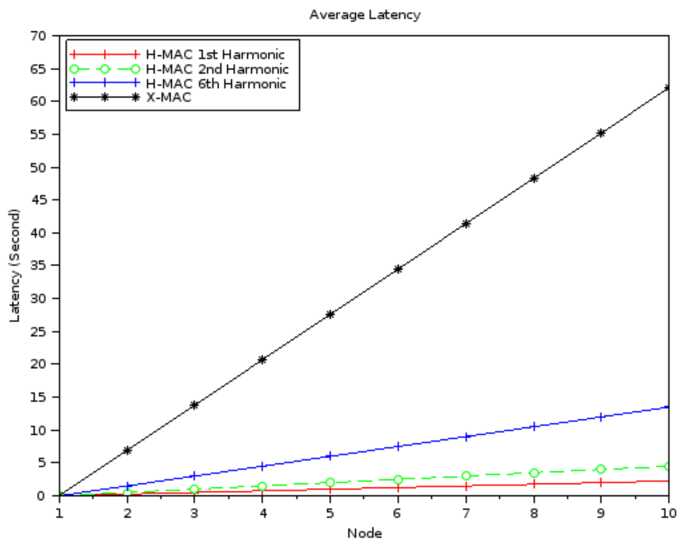

Figure 4.5: Average Latency with respect to network size

period the volumetric water content is about the same, therefore there is no need to sample at cycles less than 10 minutes.

4.12.1 Latency

To evaluate the latency of H-MAC and X-MAC we conducted an experiment in a chain topology of 10 nodes. Since, in our application the end-user firstly puts the sensor nodes in place before turning them on one by one. This sequential start up of the sensor nodes generates a start up delay between nodes. This start up delay can vary from seconds to minutes. Since, H-MAC has a dynamic transmission rate this network delay can be reduced, which is not the case with X-MAC. When H-MAC and X-MAC have the same network delay they show the same latency, therefore we decided to not show this result. Figure 4.5 shows the latency of H-MAC with a network delay of 100 milliseconds and X-MAC with a network delay of 5 second. The average latency of H-MAC increases at higher harmonics, which makes sense, since at higher harmonics more samples have to be transmitted. We can see how the network delay affects the latency, even in high harmonics H-MAC still has lower latency than X-MAC. This delay stays fixed with X-MAC, while the delay of H-MAC is adjusted by the back-off time calculate by the Cloud, as discussed in Section 4.8.

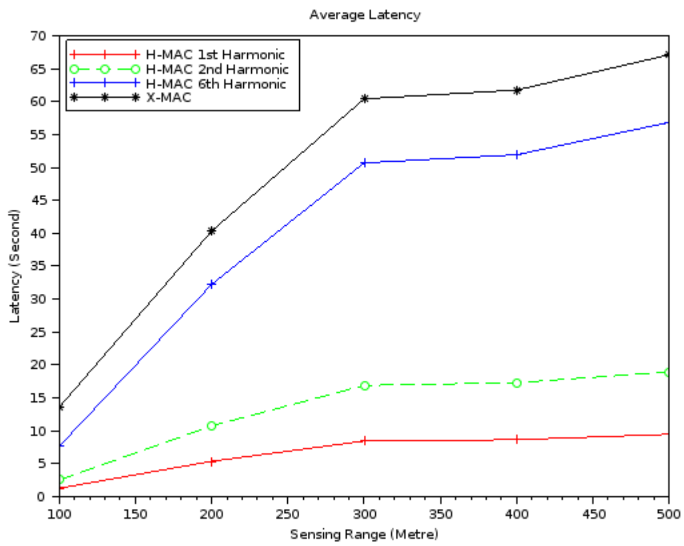

Figure 4.6: Average Latency of a network of 2 nodes with respect to sensing range

Figure 4.6 shows that the latency also increases when the nodes are further apart from each other. The increase of the latency is mostly caused by the number of packets dropped, see figure 4.9 and 4.10.

4.12.2 Duty Cycle

To evaluate the performance of H-MAC on nodes with separately duty cycling processor and radio, we performed an experiment in a chain topology network of varying size. In a period of 60 minutes we measured the operation time spent on each of the first 3 states mentioned in Table 4.1. We observed that state 0 can be neglected, since the microcontroller as well as the radio are both sleeping. State 1 has a negligible operation time compared to state 2, therefore we decided to add its operation time to state 2. By doing that we get the total operation time of the sensor node. Figure 4.7 shows the duty cycle derived from the total operation time. We observe that H-MAC has a lower duty cycle than X-MAC, due to the fact that H-MAC makes less use of the radio, since it goes immediately to sleep after the microcontroller has taken a sample, keeping the samples in memory to send them all at once, later on. While X-MAC, for each sample that the microcontroller takes, it immediately sends the sample to its neighbor, and thus using the radio more frequently. This implies that X-MAC spends more time transmitting preambles, than H-MAC. We also see that the duty cycle increases in larger networks for both protocols, although H-MAC

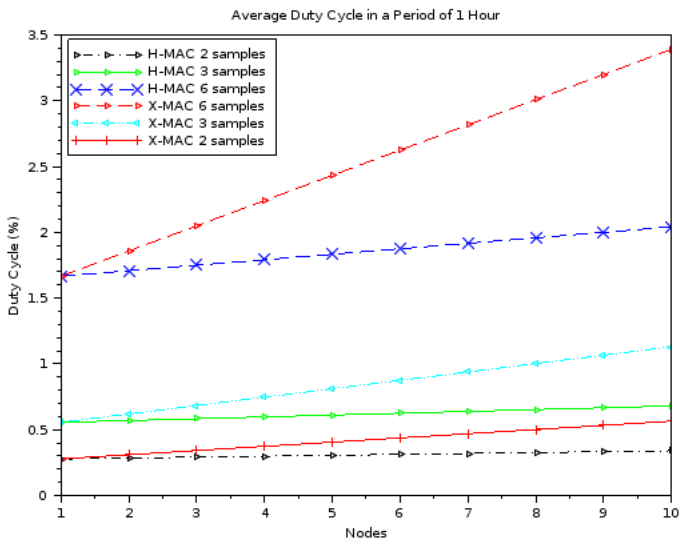

Figure 4.7: Average Duty Cycle with respect to network size

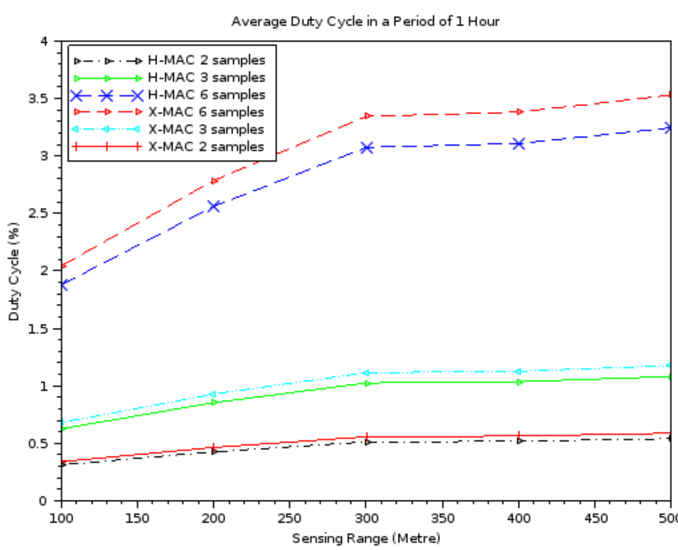

Figure 4.8: Average Duty Cycle—in a network of 2 nodes—with respect to sensing range

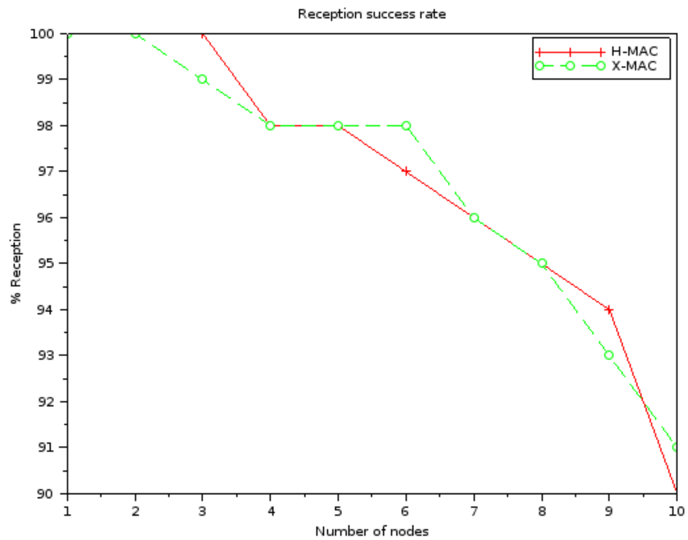

Figure 4.9: Reception success rate, 1 packet per second with respect to network size

shows a better performance, than X-MAC.

Figure 4.8 shows that the duty cycle increases with the distance between the sensor nodes, this is most due to the number of the packet drop.

4.12.3 Transmission Success Rate

To evaluate how many transmitted packets were successfully transmitted from the beginning of the network chain to the end, we conducted an experiment with varying network size. Figure 4.9 shows the results achieved when sending one packet per 10 seconds. We can see that the higher the number of nodes in the network the lower the reception success rate. Both protocols have the about the same success rate.

Figure 4.10 shows that the number of successfully received packets decreases with the distance between the nodes. After 300 metres we observed that the packet drop leveled off, we suspect before 300 metres both nodes where in a region with higher interference, and after 300 metres one of the nodes was in a region with lower interference.

4.12.4 Energy consumption

To evaluate the energy consumed by the nodes we measure the time the nodes spend operating and multiplied that time with their power usage according to

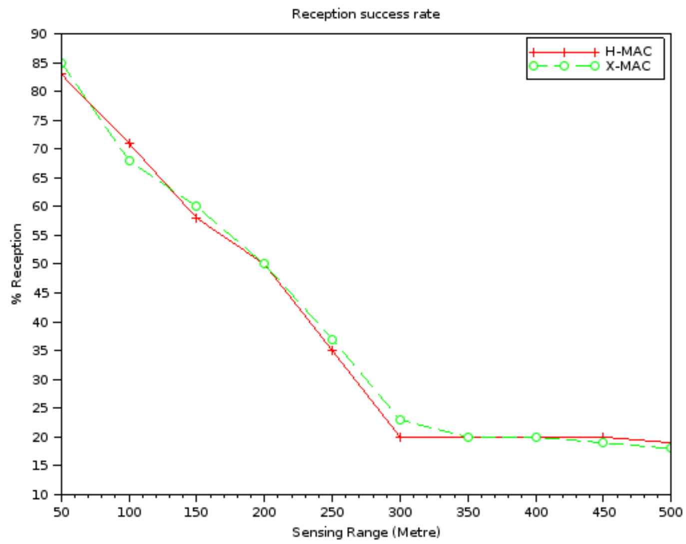

Figure 4.10: Reception success rate, 10 packets per second with respect to sensing range

their specifications. The energy consumption of the microcontroller and radio are calculated separately.

Since H-MAC only turns on the radio at the 1^{st} harmonic its energy consumption is low. While X-MAC shows higher energy consumption, because the radio has a higher duty cycle. As expected the higher the transmission frequency the higher the power consumption, which is the case of X-MAC when transmitting 6 samples in a period of an hour.

4.13 Summary

In this chapter we introduced our protocol H-MAC. We have shown that H-MAC is suitable for nodes that contain a microcontroller and radio duty cycling separately from each other. When the microcontroller and the radio duty cycle separately they might conflict with each other, since the radio is driven by the microcontroller. And also the network must be resilient to failures and scalable to be able to add new sensor nodes to the network at any time.

Therefore, we developed H-MAC. H-MAC is a hybrid between MAC protocol and dissemination protocol. It allows messages to be transmitted from the sensor nodes to the Cloud and vice versa.

In our experiment with a network with a varying size, up to 10 nodes H-MAC shows a overall better performance than X-MAC. The latency of H-MAC is lower than that of the X-MAC, since H-MAC is able to decrease its network delay between by having the Cloud compute a new back-off time, while X-MAC has a fixed network delay. Since, H-MAC uses the radio less frequently than X-MAC, it has a overall lower duty cycle than X-MAC. And also H-MAC shows a better transmission rate than X-MAC. Further, we have shown that H-MAC has lower duty cycle, lower energy consumption in comparison with X-MAC, this result is achieved by reducing the usage of the radio.

In the next chapter we address the usage of energy harvesting to avoid battery depletion.

Chapter 5

Energy Harvesting Prediction

The sensor nodes are powered by a battery pack and solar energy. Solar energy should prevent the depletion of the batteries, but it is known that the availability of solar energy is not constant. To prevent the depletion of the batteries, we must cope with the uncertainty of the availability of the solar energy.

One method that copes with the uncertainty of solar energy is to predict the availability of solar energy in the near future. To do such prediction weather forecast data is required. The authors in [36] used cloud coverage forecast data to improve their model, whilst [37] improve the model by applying machine learning, that shows that sun radiation is also an important factor for predicting solar energy in the near future. The author also showed that solar intensity depends on multiple parameters, that makes the development of a prediction model complex.

In this work we show that a model based on sunshine duration and solar radiation trumps models based on cloud coverage. Since we derive our model parameters empirically, it depends on the specific characteristics of our deployment, and are not directly useful in other deployments. The results reported in this work are dependent on the specific characteristics of our location's climate. The nodes were deployed in a open area, we did not consider the effect of local conditions, such as shade from foliage.

With the prediction sensor nodes can adjust their transmission rate, accordingly. When a high solar energy is predicted nodes can transmit more often and take more samples, while when a low solar energy is predicted for the near future sensor nodes will transmit less often and take less samples.

5.1 IBM Watson Analytics

Watson is a artificial intelligence platform developed by IBM [3]. IBM is on the leading edge on the development of artificial intelligence. Watson is the

IBM's latest generation in the field of artificial intelligence platforms, it is the successor of Deep Blue that is known by winning the World Champion chess player in 1997. Watson has proved its power in 2011 in a TV show Jeopardy where he won the challenge against top players.

In this work we used the power of IBM Watson to generate a decision tree shown in Figure 5.3 from which solar energy availability can be predicted based on current weather information.

IBM Watson Analytics creates two types of decision trees, CHAID Classification Tree and CHAID Regression Tree that we will address in the next section.

5.2 Decision Tree

From the forecast data that we gathered we generated a decision tree, shown in Figure 5.3 from which future solar energy availability can be predicted. The decision tree is created by IBM machine learning platform, Watson. From the current weather forecast information the decision tree can tell us what solar energy availability to expect in the near future.

The decision tree uses a hierarchical classification of records to predict the solar energy availability. The hierarchy is based on the values of the input fields regarding i) solar radiation, ii) sunshine period, iii) power harvested.

To create a tree, the records are segmented into groups that are called nodes. Each node contains records that are statistically similar to each other with respect to the target field, in our case the power harvested field.

Because the target field's values are statistically similar for records in a node, the node can be used to predict a target field value.

There are two types of decisions trees, CHAID (CHi_squared Automatic Interaction Detection) [24] Classification Tree and CHAID Regression Tree.

CHAID Classification Tree is a type of decision tree that uses CHAID to classify records in the target field's categories, in our case power harvested field by our solar panel.

predictions are based on the combinations of values in the input fields. The decision tree assigns predicted target category for each node in the tree. It is applicable when the target field is categorical.

CHAID Regression Tree is a type of decision tree that also uses CHAID and regression analysis to predict values of the target field, in our case power harvested by our solar panel.
The decision tree assigns predicted mean values for each node in the tree. It is applicable when the target field is continuous.

5.3 CHi-squared Automatic Interaction Detection

The most common used algorithm to create a decision tree is CHAID [24]. The advantage of CHAID over other decision tree algorithms is that CHAID is non-parametric, that means that no distributional knowledge of the dataset is required. CHAID only requires a large dataset for it to be able to produce a good decision tree.

This algorithm makes use of the chi-squared test of independence that requires the expected frequency in each cell of the relevant cross-table to be at least 5.

Appendix B shows the complete algorithm of CHAID.

5.4 Experimental Setup

The experiment was conducted in two stages in a back yard for about 7 months, from March to September 2016. One 14.5x14.5 cm solar panel of 2W.

During the first stage of the experiment we used the first 6 months to collect weather data from the Internet [8, 5, 6, 7] and measured the power harvested by the solar panel to create a energy profile, Figure 5.1.

The weather metrics collected were; i) time of day, ii) solar radiation, iii) sunshine duration; iv) cloud coverage. And from that create a decision tree, Figure 5.3.

The prediction tree was create by IBM Watson that is an artificial intelligence platform from IBM. In the last month we conducted the second part of the experiment whereby the first week we used the prediction tree to evaluate the solar energy predictions of both models. And in the last three weeks we conducted the experiment with 10 sensor nodes to measure their energy consumption using the prediction model.

5.5 Evaluation

To find out the correlation between the parameters we applied linear least square regression. In Figure 5.4 we see that there is no correlation between the time of day and the energy harvested, that makes sense since at any given time the weather can be favorable for energy harvesting or not.

Cloud coverage does not correlate with energy harvesting as well. As it can be seen in Figure 5.5, at any given cloud coverage forecast any energy could be harvested.

This means that the amount of energy that can be harvested cannot be predicted based on cloud coverage. But when looking at Figure 5.6 we see a bit of correlation between sun radiation and energy harvesting.

When the sun radiation is higher the harvested energy tend to be higher as well. This brings us to assume that sun radiation parameter is an appropriate candidate for energy harvesting prediction. A little bit less obvious is the correlation between sunshine and power harvested, Figure 5.7.

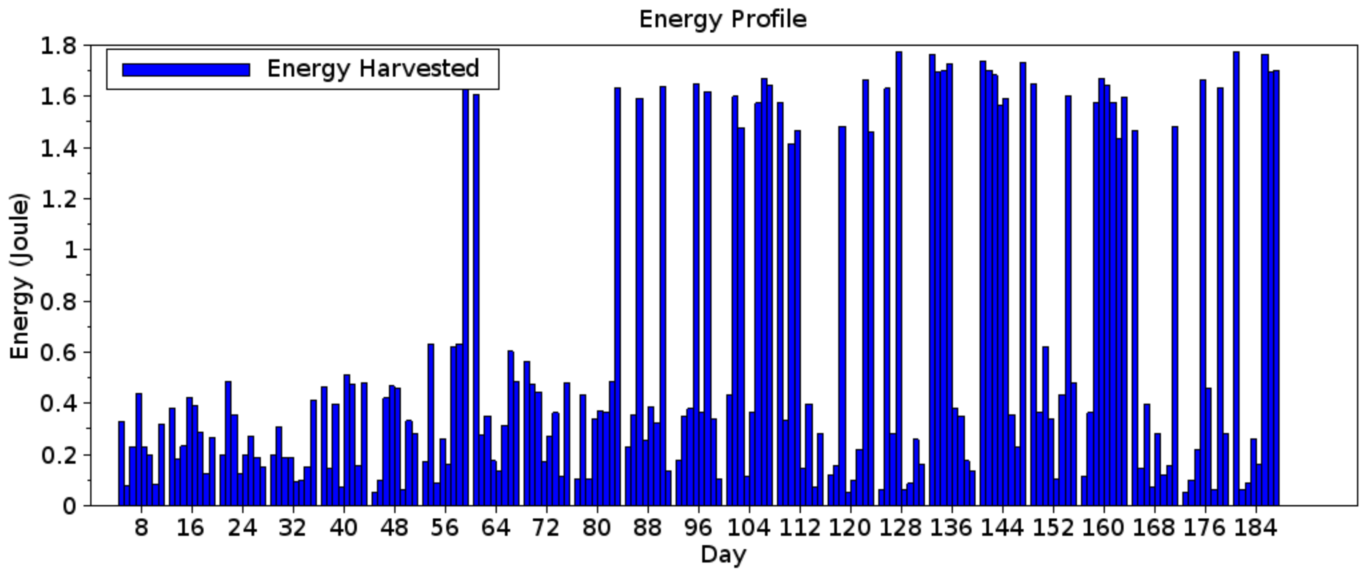

Figure 5.1: Energy Profile of the months March till August 2016

We can see that there is a tendency to harvest more energy when sunshine duration is longer and less energy is harvested when sunshine duration is short.

This brought us to ask if there is any correlation between solar radiation and sunshine duration, and sure enough as it can be seen in Figure 5.8, the higher the solar radiation the higher the sunshine duration and vice versa.

5.5.1 Comparing Results

When comparing the results, Figure 5.2 shows that predictions based on sunshine duration and solar radiation data have 10% lower error rate than predictions based solely on cloud coverage.

Both approaches predict are about the same when the weather is stable, meaning when the sky is clear throughout out the day or the sky is overcast. And they vary slightly from each other when the weather is dynamic throughout the day.

The reason for this discrepancy is that when clouds are in the sky, not overcast, they can either cover the sun or not cover the sun for some period of time. The cloud coverage data gathered from the online weather forecast only tells the percentage of cloud coverage in the sky, that stays the same regardless of clouds covering the sun or not. On the other hand data about sunshine duration provides better information about the sun availability. This gives a more precise data that improves the error rate of the prediction model by 10%.

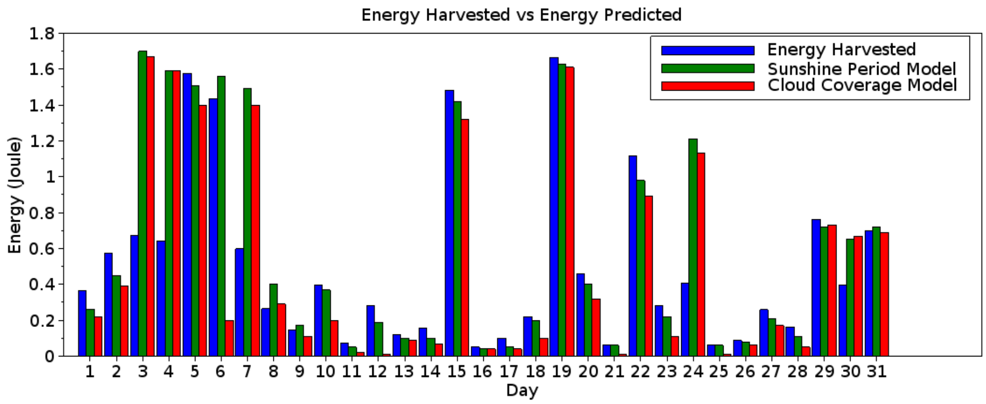

Figure 5.2: Power Harvested vs Power Predicted

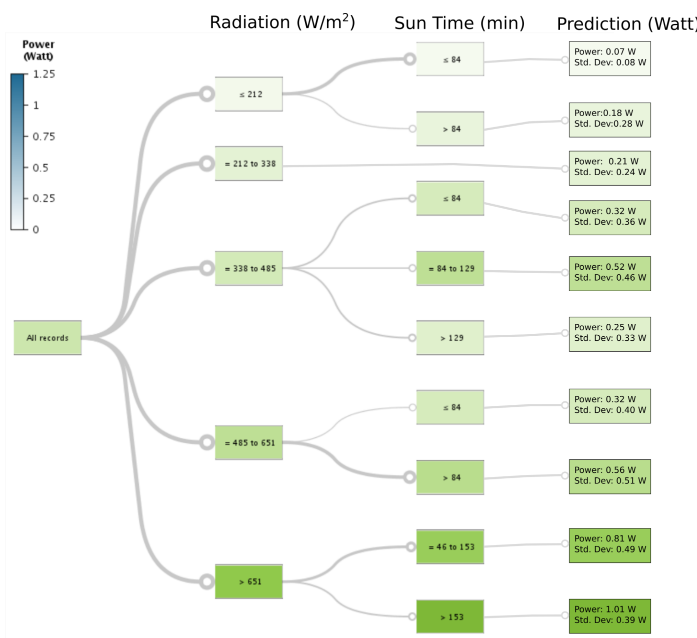

Figure 5.3: Decision Tree: Power Prediction

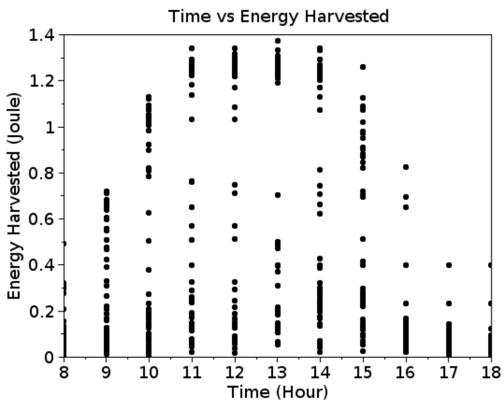

Figure 5.4: Time vs Energy Harvested

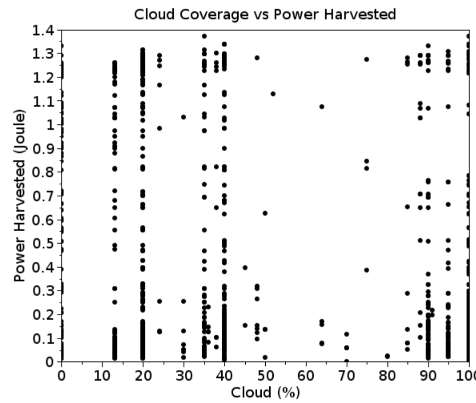

Figure 5.5: Cloud Coverage vs Energy Harvested

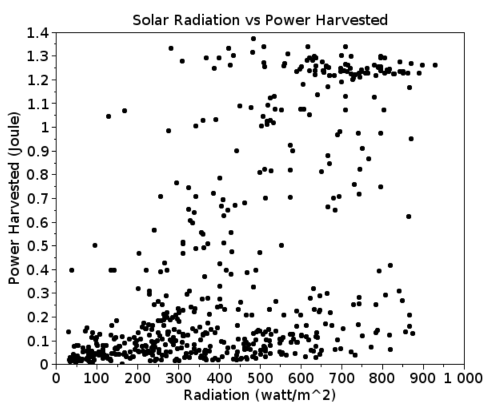

Figure 5.6: Solar Radiation vs Energy Harvested

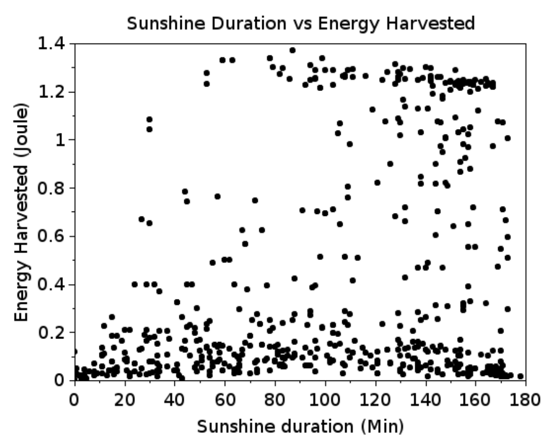

Figure 5.7: Sunshine duration vs Energy Harvested

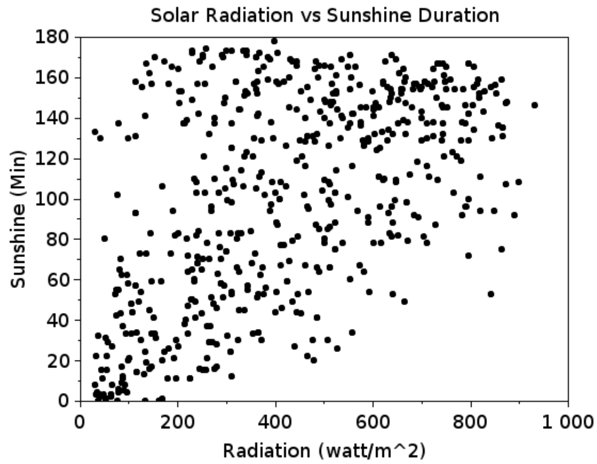

Figure 5.8: Solar Radiation vs Sunshine duration

5.6 Summary

In this chapter we showed how we harvested solar energy to reduce battery depletion. First, we created a weather profile by collecting weather forecast. We then apply linear regression to find out which metrics do correlate. We then observed that metrics such as solar radiation and sunshine duration correlates with power harvested. We also observed that the metrics sunshine duration and solar radiation do correlate with each other.

For the nodes and user to be able to make a decision based on the weather forecast we created a decision tree. The decision tree is stored on the Cloud. On the Cloud the current weather forecast is fetched to predicted the solar energy availability in the near future. Based on the prediction the duty cycle of the nodes will be decided.

By incorporating forecast information about solar radiation and sunshine duration into the prediction model, the error rate of our prediction model improved about 10%. Our model showed good results during days with clear sky and overcast sky. This was to be expected, since there is no change in the sky that could interfere with the sun radiation. In cloudy days the prediction model performed poorly, as it is hard to predict when a cloud is going to cover the sun. When the sun is covered the solar panels cannot deliver enough energy to the nodes, although the weather forecast claims that there is enough solar radiation throughout the day.

In the next chapter we put everything discussed up to this chapter together.

Chapter 6

Putting Everything Together

In this chapter we explain how it all comes together. Our goal is to extend the lifetime of the sensor nodes. To achieve our goal we took two approaches, i) reduce the power consumption, and ii) harvest energy to prevent the battery depletion.

In Chapter 3 and Chapter 4 we dealt with the reduction of power consumption. In Chapter 3 we developed a passive probe that measures volumetric water content in soil by means of potential difference between the two electrodes. This method does not consume any power at all, this is a reduction of hundred percent.

In Chapter 4 we introduced H-MAC protocol that keeps the moisture sensor node in sleep mode as much as possible, while taking into account the duty cycle of the radio and the microcontroller. Our protocol also ensures that the power consumption remains low.

In Chapter 5 we showed the solar energy prediction model. The moisture sensor nodes adapted their duty cycle according to the predicted solar energy availability for the near future.

In this chapter we combine the previous discussed solutions to extend the lifetime of the moisture sensor nodes as much as possible.

6.1 Experimental Setup

At the end of our research project we put everything together, in a field we deployed 10 passive moisture sensor nodes. As a power source each moisture sensor node has a 4.5V battery pack, and a 14cm x 13cm solar panel of 5V at 400mA. Further details about the hardware is described in Section 3.1.

The experiment was conducted in two days with different weather conditions. Day one was a sunny day with clear sky. Day two was overcast and raining.

6.2 Experimental Results

In this section we discuss the results that we attained from our experiment.

6.2.1 Soil Moisture Content

To evaluate the relation between the soil moisture level and weather condition we analyzed the moisture level during a sunny day and a raining day. Figure 6.1 shows the relation between the volumetric water content and the weather condition. We can see that when the day is sunny the volumetric water content decreases during the day, while in raining days the volumetric water content in the soil remains about the same, as expected.

We also observed that the volumetric water content measured by our passive moisture sensor remained around 60% during raining day. This is due to the fact that the water falling from the sky is absorbed by the soil and infiltrates dipper into the ground and thus not remaining on the surface.

During sunny days we observed that the soil did not lose its moisture immediately, rather gradually. This has to do with many factors, such as how much direct solar radiation is hitting the soil, air moisture content, temperature, wind and the like. We observed an unexpected rise in the output of the passive sensor. The sensor node was placed in a non-controlled environment, therefore we suspect the there was some kind of interaction between the probe and the environment including people, animals, and the like. We did not had time to repeat the experiment, since we would have to wait for another sunny day to arrive.

6.2.2 Duty Cycle of Sensor Node

To evaluate how dynamically sensor nodes change their duty cycle we analyzed the change of the duty cycle with respect to the weather condition. Figure 6.2 shows that during the sunny day the duty cycle is much higher when the solar radiation is high. Since, the nodes are duty cycling for at least every 10 minutes their duty cycle stay low. We also noticed that the sensor nodes were powered by the solar panel and thus saving the batteries.

During sunny day the sensor nodes showed a higher duty cycle after noon, this is expected, since at that period the sun has the highest radiation and the soil loses moisture faster.

During raining days the sensor nodes remained at sleep mode most of the time, every now and than they wake up to transmit their sample. During raining day the sensor nodes were powered by the battery pack. The fact, that the sensor nodes are less active in raining days does not affect the quality of service, since the soil moisture level remains about the same throughout the day.

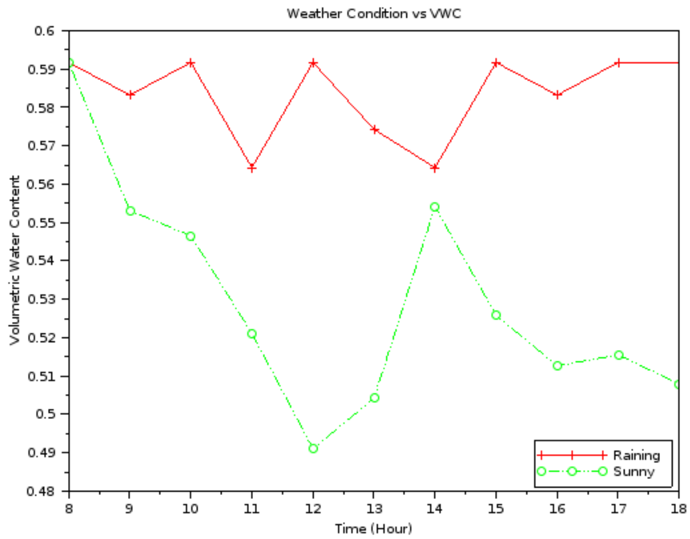

Figure 6.1: Volumetric Water Content behavior with respect to weather condition

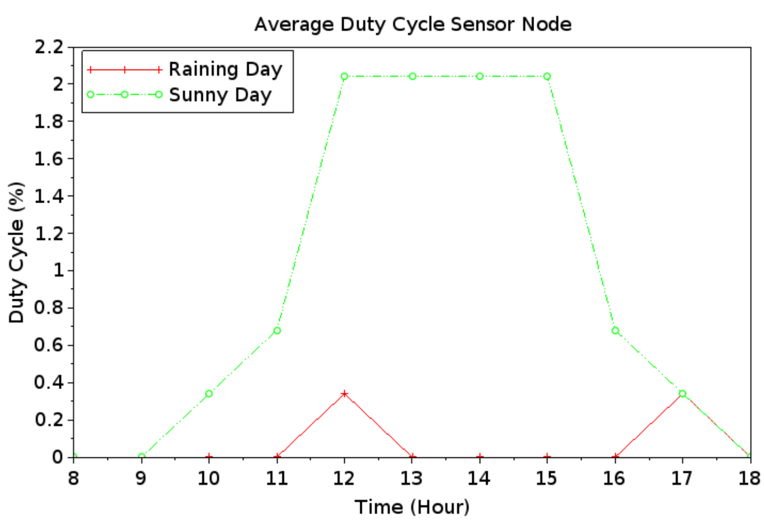

Figure 6.2: Duty Cycle with respect to weather condition

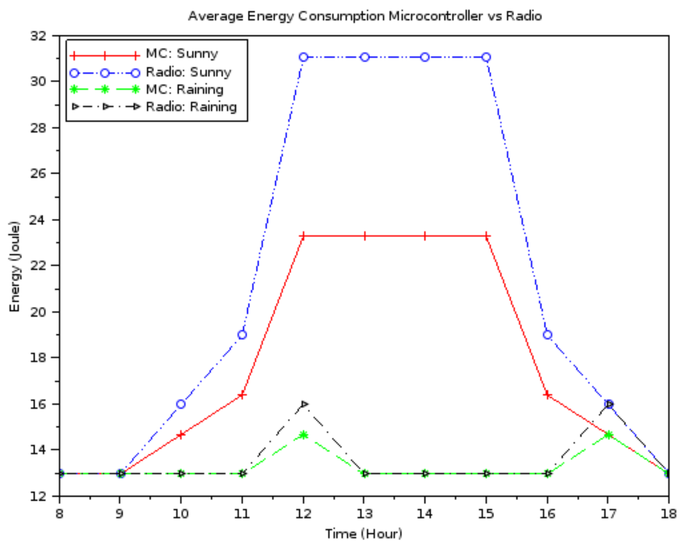

Figure 6.3: Energy Consumption with respect to weather condition

6.2.3 Energy Consumption

To evaluate the power consumption of the sensor nodes we analyzed the power consumption in a sunny day and the power consumption in a raining day. Since, the sensor nodes adjust their duty cycle according to the weather condition we observed a reduction in power consumption. Although, Figure 6.3 shows higher energy consumption during sunny days, sensor nodes are completely powered by solar energy that prevents the depletion of the battery pack, that means that 100% of the energy is saved.

6.3 Summary

In this chapter we combined all aspects discussed in previous chapters. The combination of the passive probe introduced in Chapter 3, H-MAC discussed in Chapter 4, and the prediction of solar energy addressed in Chapter 5 produced a soil moisture sensor solution with lower power consumption and it harvests solar energy. These achievements help prolong the lifetime of sensor nodes.

Here, we showed that the sensor nodes are able to change their duty cycle dynamically according to weather condition. Since, sensor nodes wake up at most once every 10 minutes, their duty cycle remain low throughout the day. During sunny days the sensor nodes are solely powered by the solar energy. This implies that no energy is being drawn from the batteries. And their highest duty cycle is after noon, when the sun radiation is higher. During raining days the sensor nodes are powered by the battery pack, but they manage to keep their energy consumption low by keeping their duty cycle low. The fact of keeping the duty cycle low during raining days does not affect the quality of service of our solution, since the volumetric water content in the soil remains about the same throughout the day.

In the next chapter we conclude our research.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In this thesis, we answered the questions posed in Section 1.4. We have made a complete wireless moisture sensor solution composed of microcontroller, radio and passive moisture probe. The hardware components that we have chosen for our solution are P0 Particle, RFM69 radio module and a passive probe made of two electrodes of different metals, copper and zinc. The P0 Particle module contains an built-in WiFi module that allows the node to communicate with the Cloud.

The radio module RFM69 allows sensor nodes to communicate with each other, it operates at a lower frequency 868 MHz, consumes less energy than WiFi chip and provides longer range. We showed how to build a low cost passive moisture sensor that measures the volumetric water content in soil by means of two electrodes of different metals, copper and zinc. The combination of our components used in our solution is at least 20% less expensive than any other low cost soil moisture sensor solution currently available on the market.

The potential difference of the electrodes changes with the level of moisture between the two metals. An isolator made of cotton that protects the metals from the environment and vice versa, facilitates the absorption of moisture between the metal plates of the electrodes. The highest potential measured on the passive probe was about 0.85V.

We showed that the potential difference between the electrodes change with the amount of moisture the metals are in contact with. If the moisture level between the metal plates is high then the potential difference will be high as well.

We showed that the moisture measurements of the passive moisture sensor correlates with the measurements of the manufactured commercial moisture sensor Decagon EC5.

The output of the passive probe can be adjusted by adjusting the pull-down resistor. The higher the value of the resistor the higher the output. With a resistor of 0 Ohms the probe gives a constant output 0V, since there would be a direct connection between the analog input pin and ground. A resistor with value higher than 100 KOhm do not provide any noticeable better output, since such resistors have a value higher than the impedance of the probe.

With these electrodes we managed to build a low cost passive probe that does not consume any energy. The output measurements of the passive probe are acceptable for any soil moisture application.

Sensor nodes need to transmit and receive data from the Cloud, through the network. WiFi module consumes a large amount of power when operating, therefore we opted to make use of RFM69. Sensor nodes pass their data to one another in a chain topology and eventually the node at the end of the chain sends the data to the Cloud with its WiFi module.

The two main components of the sensor node, microcontroller and radio, have their own duty cycle, sampling rate and transmission rate, respectively. The sampling rate dictates how often in a period of time the microcontroller samples the soil moisture. The transmission rate dictates how often in a period of time the radio should transmit data. Both components are sleeping when they are not sampling the soil moisture nor transmitting data.

Since, the radio is driven by the microcontroller a conflict may occur when the microcontroller is sleeping according to its sampling rate and at the same moment the radio has to be turned on according to the radio's transmission rate.

The network must be flexible and scalable such that new probes can be added to the network at any given time. And the network must be resilient enough to prevent the network from going offline when a node occurs a failure or is removed from the network. The network must allow messages to be delivered to the Cloud from sensor nodes and vice versa.

To solve these problems we developed H-MAC. H-MAC is a hybrid of MAC protocol and dissemination protocol. It is similar to X-MAC, it is asynchronous allowing nodes to be loosely coupled, operating independently from each other and has short preamble. This makes the network scalable. And it keeps the network up-to-date by disseminating messages from the Cloud through the network. This protocol allows the microcontroller and the radio to work in harmony while maintaining all sensor nodes with the same transmission rate.

However, X-MAC is suitable to nodes of which the processor does not duty cycle while the radio has a fixed duty cycle. H-MAC has a dynamic duty cycle and is suitable for nodes that contains a processor and a radio that duty cycle, separately. The processor and radio duty cycle in harmony.

And dissemination protocols generally perform all kinds of periodic methods to discover whether data items on the network has changed. For our application H-MAC only broadcasts updated messages after the sensor nodes have communicated with the Cloud. It is assumed that the message from the Cloud is always new. H-MAC shows a lower duty cycle, lower energy consumption and lower latency than X-MAC. With H-MAC sensor nodes use their radio less frequently that is reflected in the reduction of energy consumed by the sensor nodes.

To improve the lifetime of our sensor nodes we leverage weather forecast to predict the solar energy availability in the near future. By letting the sensor nodes only transmit data when there is enough solar energy prevents the depletion of the batteries.

We showed that the error rate of the prediction model can be improved by 10%, when weather information about solar radiation and sunshine duration is taken into account instead of just cloud coverage. In days of clear sky or overcast sky we observed that both models predicted about the same solar energy availability. Our model showed better results in days when the sky has small percentage of cloud coverage. But still, we observed that our model does not always make the right prediction when there are some clouds in the sky, since it is hard to predict when a cloud will cover the sun.

We believe that our passive wireless moisture sensor solution is affordable for everyone even for those in the developing countries.

7.2 Future Work

To test the measuring accuracy of the volumetric water content by the passive probe we conducted experiments in a sandbox. The results gathered does not reflect the behavior of the passive probe in other soil types. To extend this study further experiments should be conducted in other soil types to discover its characteristics in those soils.

The passive probe should be put through a durability test which implies conducting the same experiment for at least one year. From this extended experiment the lifespan of the passive probe can be determined.

H-MAC shows promising results, nevertheless it should be tested in a bigger field containing a higher number of sensor nodes. It should be tested in a massive network to prove its reliability and resiliency.

The communication from the Cloud to the sensor nodes can be improved by letting the nodes sleep while waiting for the response of the Cloud. At this moment the nodes stay awake waiting for the response to arrive.

Further, the feeding of the Cloud with weather forecast should be automated for a complete autonomous solution.

For marketing purposes there should be a world wide research on the acceptance of metals in soil, whether it is considered as a safe application or harmful for the environment.

Bibliography

[1] Decagon Devices. Soil Moisture Sensors. Hydrology Services. www.decagon.com.

[2] Help worlwide freshwater shortage. www.poseidonproject.org.

[3] IBM Watson, Cognitive Technology. https://www.ibm.com/watson/.

[4] Manufacturer of IOT Sensor and RF Core Component Design. http://www.hopesensor.com.

[5] Online Weather and Forecasts. http://openweathermap.org.

[6] Online Weather and Forecasts. www.accuweather.com.

[7] Online Weather and Forecasts. www.buienradar.nl.

[8] Online Weer, weersverwachting Nederland en buien radar. www.weer.nl.

[9] Particle: Connect your Internet of Things (IOT) devices. https://www.particle.io.

[10] Standard and Cross Platform Real Time Operating System. http://www.freertos.org.

[11] Constantinos Marios Angelopoulos, Sotiris Nikoletseas, and Georgios Constantinos Theofanopoulos. A Smart System for Garden Watering using Wireless Sensor Networks. *MobiWac*, 2011.

[12] J. Balendonck. Farm Level Optimal Water management: Assistant for Irrigation under Deficit. Technical report, Praktijkonderzoek Plant en Omgeving BV, 2010.

[13] Allen J. Bard, Roger Parsons, and Joseph Jordan. *Standard Potentials in Aqueous Solution*. Marcel Dekker, Inc, 1985.

[14] Burton H. Bloom. Space/Time Trade-offs in Hash Coding with Allowable Errors. *Magazine Communications of the ACM*, 3:422–426, 1970.

[15] Michael Buettner, Gary V. Yee, Eric Anderson, and Richard Han. X-MAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks. *SenSys0́6*, Proceeding SenSys 0́6 Proceedings of the 4th international conference on Embedded networked sensor systems:307–320, 2006.

[16] Thanh Dang, Nirupama Bulusu, Wu chi Feng, and Seungweon Park. DHV: A Code Consistency Maintenance Protocol for Multi-Hop Wireless Sensor Networks. *EWSN 2009: Wireless Sensor Networks*, 327-342:5432, 2009.

[17] Petra Döll. Vulnerability to the impact of climate change on renewable groundwater resources: a global-scale assessment. *IOPScience*, 4, 2009.

[18] Mahir Dursun and Semih Ozden. A wireless application of drip irrigation automation supported by soil moisture sensors. *Academic Journals*, 2011.

[19] Juan M. Enciso, Dana Porter, and Xavier Périès. Irrigation Monitoring with Soil Water Sensors. Technical report, The Texas A&M University, 2007.

[20] Steven M. Gorelick and Chunmiao Zheng. Global change and the groundwater management challenge. *Water Resources Research*, 51:3031–3051, 2015.

[21] Joaquín Gutiérrez, Juan Francisco Villa-Medina, Alejandra Nieto-Garibay, and Miguel Ángel Porta-Gándara. Automated Irrigation System Using a Wireless Sensor Network and GPRS Module. *Instrumentation and Measurement, IEEE Transactions on*, 2013.

[22] Arjen Y. Hoekstra and Mesfin M. Mekonnen. The water footprint of humanity. *PNAS*, 9:3232–3237, 2012.

[23] Jason Hsu, Sadaf Zahedi, Aman Kansal, Mani Srivastava, and Vijay Raghunathan. Adaptive Duty Cycling for Energy Harvesting Systems. *Proceeding ISLPED '06 Proceedings of the 2006 international symposium on Low power electronics and design*, pages 180 – 185, 2006.

[24] Gordon V. Kass. An exploratory technique for investigating large quantities of categorical data. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 29:119–127, 1980.

[25] Byung Sik Kim, Hung Soo Kim, Byung Ha Seoh, and Nam Woon Kim. Impact of climate change on water resources in Yongdam Dam Basin, Korea. *Stochastic Environmental Research and Risk Assessment*, 21:355–373, 2007.

[26] Hermann Kopetz. *Real-Time Systems*, chapter 13 - Internet of Things, pages 307–323. Springer US, 2011.

[27] Marc J. Leblanc, Paul Tregoning, Guillaume Ramillien, Sarah O. Tweed, and Adam Fakes. Basin-scale, integrated observations of the early 21st century multiyear drought in southeast Australia. *Water Resources Research*, 45, 2009.

[28] Philip Levis, Neil Patel, David Culler, and Scott Shenker. Trickle: A Self-Regulating Algorithm for Code Propagation and Maintenance in Wireless Sensor Networks. *NSDI '04*, 2004.

[29] Kaisen Lin and Philip Levis. Data Discovery and Dissemination with DIP. *Information Processing in Sensor Networks, 2008. IPSN '08. International Conference on*, 2008.

[30] Francisco G. Montoya, Julio Gómez, Alejandro Cama, Antonio Zapata-Sierra, Felipe Martínez, José Luis De La Cruz, and Francisco Manzano-Agugliaro. A monitoring system for intensive agriculture based on mesh networks and the Android system. *Computers and Electronics in Agriculture*, 2013.

[31] Joseph Polastre, Jason Hill, and David Culler. Versatile Low Power Media Access for Wireless Sensor Networks. *SenSysÓ4*, pages 95–107, 2004.

[32] Maryam Pournasiri Poshtiri and Indrani Pal. Patterns of hydrological drought indicators in major U.S. River basins. *Climatic Change*, 134:549–563, 2016.

[33] Saifur Rahman. Green Power: What Is It and Where Can We Find It? *IEEE Power & energy Magazine*, 1:30–37, 2003.

[34] Andreas Reinhardt and Christian Renner. RoCoCo: Receiver-Initiated Opportunistic Data Collection and Command Multicasting for WSNs. *European Conference on Wireless Sensor Networks*, pages 218–233, 2015.

[35] Alexandra S. Richey, Brian F. Thomas, Min-Hui Lo, John T. Reager, James S. Famiglietti, Katalyn Voss, Sean Swenson, and Matthew Rodell. Quantifying renewable groundwater stress with GRACE. *Water Resources Research*, 51:5217–5238, 2015.

[36] Navin Sharma, Jeremy Gummeson, David Irwin, and Prashant Shenoy. Cloudy Computing: Leveraging Weather Forecasts in Energy Harvesting Sensor Systems. *2010 7th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, pages 1–9, 2010.

[37] Navin Sharma, Pranshu Sharma, David Irwin, and Prashant Shenoy. Predicting solar generation from weather forecasts using machine learning. *Smart Grid Communications (SmartGridComm), 2011 IEEE International Conference on*, pages 528 – 533, 2011.

[38] Yanjun Sun, Omer Gurewitz, and David B. Johnson. RI-MAC: a receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks. *Proceeding SenSys '08 Proceedings of the 6th ACM conference on Embedded network sensor systems*, pages 1 – 14, 2008.

[39] Gilman Tolle and David Culler. Design of an Application-Cooperative Management System for Wireless Sensor Networks. *Wireless Sensor Networks, 2005. Proceeedings of the Second European Workshop on*, 2005.

[40] Stefan UnterschÃŒtz, Christian Renner, and Volker Turau. Opportunistic, Receiver-Initiated Data-Collection Protocol. *9th European Conference, EWSN 2012, Trento, Italy, February 15-17, 2012. Proceedings*, pages 1–16, 2012.

[41] Tijs van Dam and Koen Langendoen. An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks. *Proceeding SenSys Ó3 Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 171–180, 2003.

[42] Merel van Diepen and Philip Hans Franses. Evaluating chi-squared automatic interaction detection. *ScienceDirect*, 31:814–831, 2006.

[43] Wei Ye, John Heidemann, and Deborah Estrin. An Energy-Efficient MAC Protocol for Wireless Sensor Networks. *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 2002.

[44] Fedro S. Zazueta and Jiannong Xin. Soil Moisture Sensors. Technical report, University of Florida, 1994.

Appendix A: Eagle schematic of Moisture Sensor

Figure 7.1 shows the eagle schematic of the soil moisture sensor. On the left side we see the Photon module with the passive probe attached to it. The probe is connected the Ground pin and Analog pin of Photon. Zinc metal is suppose to be always connected to the Ground pin, while copper metal is suppose to be connected to one of the analog pins of Photon.

Between the ground pin and the analog pin a resistor R1 and capacitor C1 is placed. The purpose of the capacitor is to smooth the signal coming from the probe. The resistor creates a reference with ground.

On the right side of the schematic we find the RMF69 radio module that is connected to Photon through the SPI pins.

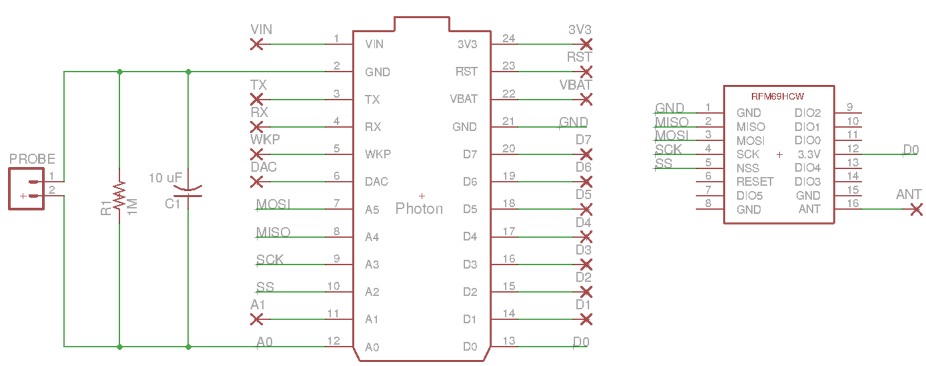

Figure 7.1: Eagle Schematic

Appendix B: CHAID Algorithm

The CHAID algorithm [42] works as follows:

1. For each predictor in turn, cross tabulate the categories of the predictor with the categories of the dependent variable and do steps 2 and 3.

2. Find the pair of categories of the predictor (only considering allowable pairs as determined by the type of the predictor*) whose sub-table is least significantly different. If this significance does not reach a critical value, merge the two categories, consider this merger as a single compound category, and repeat this step.

3. For each compound category consisting of three or more of the original categories, find the most significant binary split (constrained by the type of the predictor) into which the merger may be resolved. If the significance is beyond a critical value, implement the split and return to step 2.

4. Calculate the significance** of each optimally merged predictor and isolate the most significant one. If this significance is greater than a critical value, subdivide the data according to the (merged) categories of the chosen predictor.

5. If the stopping criterion*** is not met yet, return to step 1, for each partition of the data that has not yet been analysed.

*CHAID can process three types of categorical predictors: monotonic, free and float. The monotonic predictor contains some kind of natural ordering, that should be maintained.

This implies that only contiguous categories may be merged. The free predictor is purely nominal, so that all categories may be merged. The float predictor is similar to the monotonic predictor with the exception of one category.

This category may be merged with all other categories and it usually represents “no opinion” or “don’t know”. The type of each predictor affects the merging algorithm and hence the Bonferroni adjustment formula.

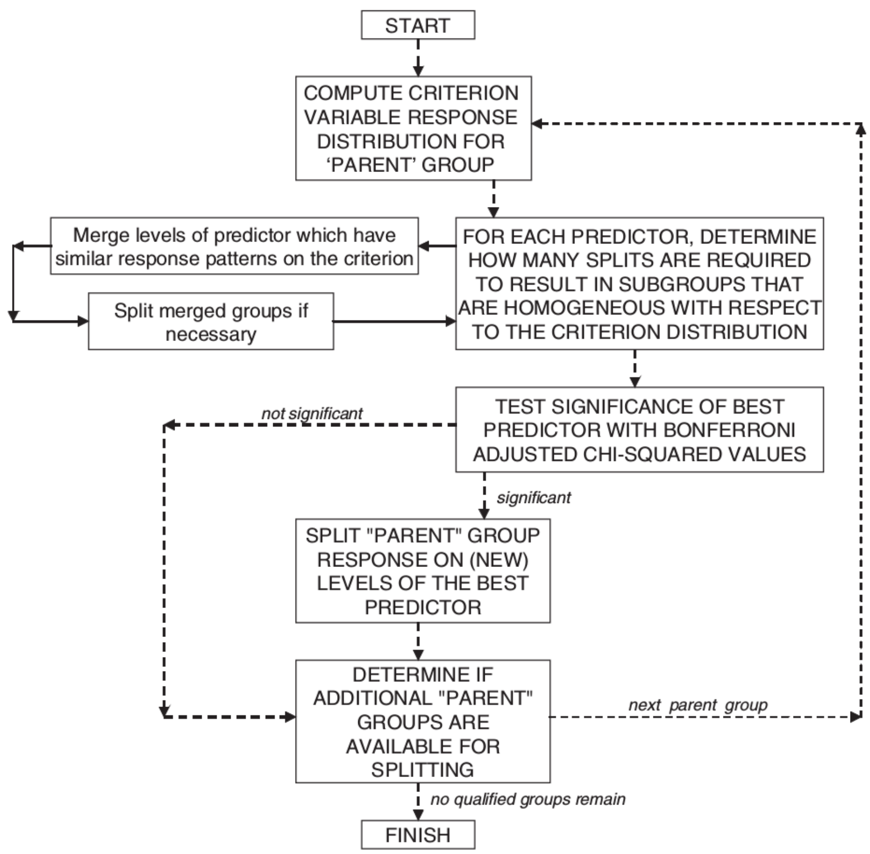

Figure 7.2: CHAID Algorithm [42]

**This significance is computed according to the appropriate Bonferroni adjustment.

***A node will not be split if any of the following conditions is met: (i) all cases of a node have identical values for all predictors; (ii) the node becomes pure, i.e., all cases in the node have the same value of the target variable; (iii) the depth of the tree has reached its pre-specified maximum value; (iv) the number of cases constituting the node is less than a pre- specified minimum parent node size; (v) the split at the node results in producing a child node whose number of cases is less than a pre-specified minimum child node size; and (vi) no more statistically significant split can be found at the specified level of significance. All but the first two of these rules could be user specified.

Figure 7.2 shows the flow diagram of the CHAID algorithm.