Smart Farming For Emerging Economies

A LoRaWAN Based Approach

Stephen Simei Kimogol

Thesis submitted for the degree of
Master in Informatics: Programming and System Architecture
(Distributed Systems and Networks)
60 credits

Institute of Informatics
Faculty of mathematics and natural sciences

UNIVERSITY OF OSLO

Autumn 2019

Smart Farming For Emerging Economies

A LoRaWAN Based Approach

Stephen Simei Kimogol

Smart Farming For Emerging Economies

http://www.duo.uio.no/

Printed: Reprosentralen, University of Oslo

Summary

Population growth and climate change pose a real threat to food security. In addition, current food production practices have created a huge ecological footprint. In spite of this, food production has to increase globally. There is a need for smallholder farmers to adopt technology to facilitate efficient food production. The recent advancement of sensor technology, computational power, and the emergence of low power wide area networks (LPWAN) for the Internet of Things(IoT) connectivity enables the collection, transmission, processing, and analysis of data. These technologies in combination with low-cost devices and the availability of open-source software enable the development of smart solutions that can make sectors like farming efficient. In this thesis, we propose a LoRaWAN based smart farming for emerging economies.

Acknowledgement

I would first like to thank my supervisor Professor Josef Noll of the Institute of Technology Systems (ITS) and internal supervisor Professor Olaf Owe of Institute of Informatics (IFI) for their support. I would like to thank Prof. Josef for his guidance, encouragement, and willingness to give his time so generously throughout this thesis.

Secondly, I would like to extend my gratitude to Pål G. Solheim of the Institute of Technology Systems (ITS) for his advice and provision of the necessary equipment to fulfill my work. My acknowledgment also goes to David Gureya for proofreading this thesis.

Finally, I would like to express my profound gratitude to my family and friends for their support. A special thanks to my wife for her love and support and my daughter for being such a good child making it possible for me to complete this thesis.

Contents

List of Figures

List of Tables

1 Introduction

The rapid growth of the world population and climate change poses a great threat to food security. Population growth affects the capacity of the environment to produce food due to changes in land use and the limited availability of arable land. On the other hand, changes in land use and agriculture contribute to a quarter of the total greenhouse gas emissions [1]. Fluctuations of weather patterns have an impact on food production and this affects the livelihoods of people, especially the poor who live in rural areas. In spite of this, global food production has to increase by 70 % by 2050 to feed the growing population [2]. The problem of food security, population growth, and climate change are intertwined and there is a need to promote farming practices that are cognizant of these challenges. Technology can transform agriculture and help smallholder farmers adopt new farming approaches making them resilient to climate change.

Smallholder farmers need to produce sufficient food to match the food demand of the growing population. To achieve this and practise sustainable farming smallholder farmers need information. Furthermore, they need to understand how different crops perform in changing weather patterns and varying availability of water. Fortunately, there is technological advancement that is driving the fourth industrial revolution which is impacting all sectors by connecting physical and digital worlds [1]. Agriculture is also experiencing this development with the adoption of Information and Communication Technology (ICT). For smallholder farmers in developing countries, this development is particularly suitable as they can leverage this technology to adjust to climate change and produce food efficiently. Adoption of technologies like the Internet of Things (IoT), edge and cloud computing coupled with sensor technology and low cost and energy-efficient wireless communication could help them in the realization of information-driven farming. Moreover, farmers get full control of their farm operations, monitor soil and crop growth at different stages. In addition, they can monitor their animal's health and based on the farm data farmers can optimize their farm inputs. Also, such technologies can help them diversify farming and enable them to not only rely on rain-fed agriculture but also adopt other techniques like irrigation and hydroponic systems to complement traditional agriculture.

1.1 Motivation

The use of ICT in agriculture in developing countries is focused on the use of mobile technology to dispatch information to farmers i.e. market information, early weather alert and interaction with agricultural officers for advisory purposes [3]. To fully realize the potential of ICT and foster efficient food production, there is a need to leverage advanced technologies. However, the use of advanced technologies in agriculture has not found significant space in developing countries. This has mostly been hindered by high cost, limited internet connectivity and lack of efficient communication that offer long-range and power efficiency. The rapid development of IoT, low power wide area networks (LPWAN) together with the increased availability of low-cost hardware and open-source software offers new opportunities to design new solutions that can help farmers in rural areas.

1.2 Problem statement

There is a general consensus that smallholder farmers in developing countries have to adopt technology to facilitate efficient food production and enable them to be more resilient to climate change. In addition, the use of technology has to go beyond the current mobile phone-based solutions that facilitate sharing of information regarding weather, financial services and market prices into putting technologies in the farms to get data that promote smart farming. Furthermore, smallholder farmers are less likely to benefit from commercial solutions and they could be at risk of being ”further marginalized and disadvantaged as the last in line to benefit from the data revolution” [4]. As such, there is a need to develop open and low-cost smart farming solutions that meets their needs. This thesis examines LoRaWAN based smart hydroponic farming as one potential way of adopting technology for efficient food production. Besides, frugal use of resources in hydroponic farming makes it more resilient to climate change and it can complement traditional farming.

1.3 Thesis outline

This thesis is organized as follows. Section 2.1 gives a piece of background information on population, food security and adoption of technology in agriculture. In section 2.2 we present digital dimensions of agriculture. In section 2.3 a use case used in this thesis is presented. An outline of the requirements of smart farming in resource-constrained regions is given in section 2.4. We also give an introduction to the different technologies that enable smart farming in section 3 and in section 3.4 we give a brief overview of the related work. Section 4 builds on the technologies discussed in section 3 and presents the system architecture and implementation of smart farming in developing countries. Section 5 gives an evaluation and directions of future work. In the final section, we give our conclusion.

2 Background

Population growth and climate change are a global challenge. United Nation's 2030 agenda, defines 17 Sustainable Development Goals (SDG) [1], among them eradication of hunger and responsible production and consumption. The demographic, climatic and environmental changes call for the use of innovative technologies to address food security problems. There is a need to use technology to regulate the consumption of depleting resources, increase productivity and enhance resilience. ICT has the most impact on development, particularly on innovation, efficiency and effectiveness in all sectors[5].

Figure 1: Sustainable Development Goals.

2.1 Agriculture and Technology in developing countries

2.1.1 Population, climate change and agriculture

Besides providing food, agriculture is a source of livelihood for 36 % of the world's task force with 40-50 % of Asia and the Pacific population and two-thirds of people in sub-Saharan Africa relying on it to make a living [6]. Climate change affects food production and this is felt mostly by the people in emerging economies who rely on agriculture is the main source of livelihood. Since most people depend on agriculture, which is sensitive to rainfall variability and temperature change, hunger is a significant threat in the face of climate change. In

---

[1] `https://www.un.org/sustainabledevelopment/sustainable-development-goals/`

addition, farmers in these areas are resource-limited and vulnerable to climate change.

Climate change will only exacerbate water scarcity and unpredictability of water supply due to changes in weather patterns. Currently, 70 % of freshwater in the world is used for agriculture and there will be growing competition for water between agriculture, industries, and consumption in the cities [7]. Water scarcity in the face of climate change will affect most rural communities in sub-Saharan Africa and South Asia where water problem is already a challenge and have low capacities to adapt to changes in climate. There is a need of using technology to help the farmers in these regions to adapt to climate change and practise farming techniques that are cognizant of the current problems caused by climate change.

The United Nations (UN) projected in 2017 that the world population will reach 9.8 billion in 2050 and over half of this population growth (1.3 billion) and 750 million will occur in Africa and Asia respectively [8]. Yet, according to the UN Food and Agriculture Organization (FAO), 821 million (one person out of every nine in the world) are currently undernourished [9] and it is estimated that food production in Africa has to increase by 260 % by 2050 to provide food for the expected population [10]. The increase demand in food production to feed the growing population will have more effect as the current agricultural production approaches have already created a large ecological footprint as shown in figure 2 [11]. To address food security problem and at the same time reduce ecological footprint associated with food production, agriculture has to be transformed.

Figure 2: Food demand vs ecological footprint (Source: [11]) .

2.1.2 Uptake of agricultural technologies

In this section we will discuss the uptake of agricultural technologies in developing countries, causes of low uptake and opportunities technological developments offer.

Agricultural engineering and mechanization contributed to rise of large-scale farming and increased production and transformation of countries from agriculture to industry-based economies [12]. With the use of modern agricultural approaches like irrigation and fertilizers, the cereal production in East Asia increased by 2.8 % a year between 1961 and 2004 while there was stagnation of yields in sub-Saharan countries that did not adopt those approaches[13].

Heavy investment is needed for mechanization of farming to increase production like in developed countries, but this is not realistic for most smallholder farmers. As mentioned earlier, the fourth industrial revolution has driven agriculture towards the use of IoT and sensor technology that is facilitating efficient farm practices. Smallholder farmers can capitalize on the benefits brought by the fourth revolution to increase their production sustainably. However, some of these new technologies are not yet mature and challenges summarised in figure 3 hinder their adoption.

Figure 3: Major barriers that limit implementation of new technologies in agriculture (Source: [14])

Whilst the above mentioned factors applies to all, developing countries are not keeping pace with the rest in uptake of technologies due to the following additional barriers.

2.1.2.1 Socio-economic factors

The social-demographic and socio-economic factors affect the adoption of new technologies [15, 16]. Farmer's education level, age and computer confidence are among the factors that hinder farmers choice of technology. The knowledge to existence of technology is also an important factor in the adoption of technology [17] and in many cases even the existing knowledge and technologies have not reached farmers in developing countries [18].

2.1.2.2 Infrastructure

Adoption of technology in developing economies is mostly hindered by insufficient or lack of infrastructure. Access to communication infrastructure and the Internet are key enablers in the adoption of technology in agriculture. Information and communication technologies keep farmers informed about the recent technologies in agriculture, weather conditions, financial services and enable connection with buyers [19]. However, according to the International Telecommunication Union (ITU), 53 % of the world's population are still unconnected to the Internet and they could not benefit from the aforementioned benefits [20]. Internet connection is not given in the most emerging economies – of the 6000 gateways that are operational in the world, only 100 are in Africa inhibiting access to open and free network [21]. The UN has acknowledged the indispensability of access to information and the critical role played by communication technology. In the recently launched UN's SDGs, one of the targets of the ninth goal is to 'increase access to information and communications technology and strive to provide universal and affordable access to the Internet in least developed countries by 2020' [2]. Several mobile services are already offered to farmers, but uptake and use of more advanced devices and services e.g cloud-based services are influenced by battery life of devices and access to fast internet [3].

2.1.2.3 Cost and ownership of technology

Further, there is a disparity in the research, development and ownership of new technologies since public and private investment in such technologies is concentrated in high- income countries thus limiting access to emerging countries [19]. Also governments in developed countries are giving subsides and invest in projects that facilitate the adoption of technology in agriculture. For example, the European Union gave €95 billion for modernization of agriculture in Europe between 2007 and 2013 [22]. In low-income countries, public spending on research and development for science and technology is below the recommended 1% of the agricultural gross domestic product (GDP)[19].

---

[2] `https://www.un.org/development/desa/disabilities/envision2030-goal9.html`

2.2 Digital Dimension of Agriculture

The use of advanced technologies has been integrated to farming and new concepts like precision farming/agriculture and smart farming concepts have emerged. While these concepts all revolve around modernization and use technology in agriculture, they have some differences.

2.2.1 Precision Farming

According to a report by European Parliament on Precision agriculture and the future of farming in Europe, precision agriculture is defined as: “a modern farming management concept using digital techniques to monitor and optimise agricultural production processes” [23]. The focus is optimization of farm inputs. It ranges from application of correct amount of fertilizers to the specific part of the field based on soil properties, precise water use and to giving the correct amount of feed to a specific animal. Sensor, satellite navigation and positioning technology are an indispensable part of Precision Agriculture. Precision farming commenced when GPS signals were made available for the general public [24]. Precision farming has successfully been implemented in large-scale farms in Central and Northern Europe, the USA and Australia with use of Controlled Traffic Farming (CTF) and auto-guiding systems showing clear benefits [25].

According to Wolfert, Goense and Sørensen, the development of precision agriculture is as a result of growth of farm enterprises and move from scaling of farm assets to optimization of assets [26]. With the increase of cost of the farm inputs and regulations e.g. use of fertilizers and unpredictability due to climate and market prices, different systems that collect and manage data were developed to help farmers in making right decisions. Precise monitoring and control are done to manage spatial and temporal variability of crops, animals and soil factors [11]. It differs from traditional farming by accurately identifying variations and relating spatial data to management activities [27].

2.2.2 Smart Farming

Smart farming is a recent phenomenon that came into being with the inclusion of computing technologies and the transmission of data in agriculture [28]. It overlaps with technologies like precision farming and management information systems that have been derived from farm management information systems (FMIS) [28]. It extends precision agriculture, where management is based not only on the location and on field variability but also on data that is triggered by real-time events [11]. This requires use of different technologies as depicted in figure 4.

Figure 4: A smart farming technologies (Source: [22]).

In smart farming, the focus is on the utilization of ICT in the cyber-physical farm management cycle [29]. This is enhanced by the advancement of nanotechnology in the last decade which enables production of small and inexpensive sensors [30]. Moreover, cloud computing and IoT promote the development of smart farming [11]. The use of sensors enables data collection and monitoring and events triggered by analysis done in the cloud.

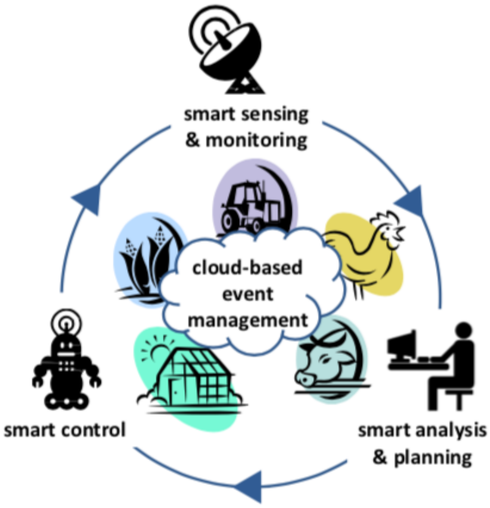


Figure 5: Smart farming management cycle (Source: [26]).

Figure 5 shows a smart farming as cycle of sensing, monitoring, analysis and cloud based control of farm events. The harvesting of data from sensors deployed in the fields aid decision making process on animal health, remote monitoring and accurate diagnosis of the soil and crop conditions and timely interventions. Farmers will also have access to historical data of weather and other inputs and they can make informed decisions. This will result in less waste, efficient use of resources and effective food production thus reduction of the ecological footprint [31].

ICT is viewed as an enabler of climate-smart farming which is "agriculture that sustainably increases productivity, resilience (adaptation), reduces/removes GreenHouse Gases(GHGs) (mitigation), and enhances achievement of national food security and development goals."[18]. But the adoption of technology by smallholder farmers in developing countries, especially in Africa, mostly revolves around the use of mobile phones and services provided through it. This includes sharing of agriculture-related information, provision of financial services, weather and market price information[3]. To further improve the practice of climate-smart farming, we can leverage technological advancement and help farmers diversify farming practices. However, this requires a holistic approach and involvement of different agents to achieve it. Indeed, as Walter et al. points out that "only if aspects of technology, diversity of crop and livestock systems, and networking and institutions (i.e. markets and policies), are considered jointly in the dialogue, should farming in the digital era be termed 'smart farming'"[31]. Figure 6 shows these four factors.

We endorse their (Walter et al.) opinion as this approach is necessary for emerging economies where planning and implementations of policies are mostly disjointed due to lack of resources and poor governance. Inspired by this view of smart farming and taking into account the challenges in the adoption of technology in emerging economies as discussed in section 2.1.2, in this thesis, we focus on following technological aspects of smart farming:

- The integration of information and communication technology into farming management systems and leverage (advancement in technology) low-cost sensors to monitor farm systems for efficient use of resources and sustainable food production.

- Making sensor data and information on smart farming accessible to farmers and sharing of data among different stakeholders. This entails the storage of data in local servers and periodic transmission to cloud for remote access for extension officers and other agents. In developing countries, farmers rely on agricultural extension officers on issues related to farming, which is usually done through field visits. Using online platforms that store data from farms will give new interaction between farmers and extension officers. This will enable a timely response from agricultural officers and save on costs related to fieldwork for data collection. Early warning and timely information about farm conditions and advice from extension officers can foster effective response and measures by farmers. Augment data collected from the farms with information like the weather forecast to help farmers and extension officers in decision making and generation of actionable information.

- Use technology to diversify farming systems and introduce practices that were not possible or required skills e.g. hydroponic farming (monitoring nutrient solutions) or precise irrigation to reduce water consumption (time for water and site-specific needs) thus, reconciling the need for increased food production and sustainability. This thesis used hydroponic farming as an example of diversifying farming with the help of technology.

.

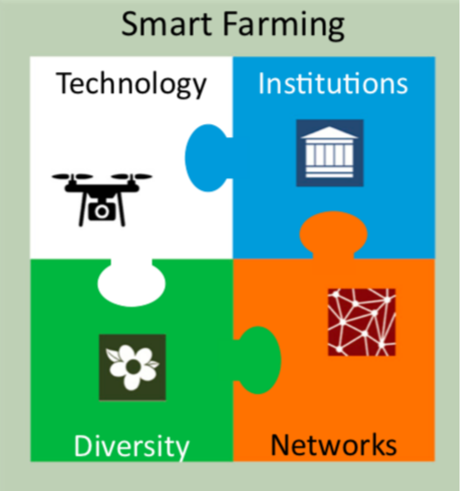


Figure 6: Smart farming in digital era. source: [31]

2.3 Use Case: Hydroponic farming

The objectives of using hydroponic farming as use case scenarios are two fold: 1) extract the functional requirements of smart farming in resource constrained setting (technologies) and 2) the need of using technology for efficient food production in the face of climate change and population growth (shortage of water and arable land).

Growing plants without soil has been practiced for a long time. This method of food production has been practiced earlier e.g hanging gardens of Babylon, the floating gardens of the Aztecs of Mexico [32]. The term Hydroponics, however, is recent and was first used by W.F. Gericke of the University of California in early 1930s [32]. Hydroponics can be defined ”as the science of growing plants without the use of soil, but by the use of an inert medium, such as gravel, sand, peat, vermiculite, pumice, perlite, coco coir, sawdust, rice hulls, or other substrates, to which is added a nutrient solution containing all the essential elements needed by a plant for its normal growth and development” [32]. In hydroponic systems plants can either grow in an aqueous media or substrate [33]. In substrate approach plants grow in pots filled with growing medium e.g. sawdust while in aqueous approach there are three designs used: nutrient film technique(nutrient solutions flow through the plastic pipes with holes on which plants are placed), deep water culture(plants roots are in the nutrient solution which is aerated) and aeroponics (roots of the plants are suspended in air and are sprayed with nutrient solution continuously) [33]. Figure 7 shows nutrient film technique and deep water culture.

Hydroponics farming is classified as either open (nutrient solution is not reused) or closed (where solution is recovered, replenished and recycled) [34]. Hydroponics has several advantages over the traditional farming: it can be used in areas where in-ground farming is not possible e.g due to climate (cold and desert), areas with water scarcity and conditions where complete control of nutrient content is required and there is a need for increased productivity/crop yields [35]. Table 1 compares soil less culture (hydroponic) and soil(traditional).

Hydroponics, if adopted can address challenges faced by smallholder farmers in developing countries like scarcity of water, limited arable land, labour cost and reduced long growth periods [36]. In optimal growing conditions hydroponic greenhouse far out-yield varieties produced on the field e.g Tomatoes production increased in yields by 4- 10 times [32] and for the production of fodder 50 sq. m. area could produce 600 kg maize fodder in seven days compared to 1 ha of land needed to produce the same amount of fodder [36]. The major limitation to adoption of hydroponics is the initial capital required [35] especially for smallholder farmers in developing countries. However, the cost can be reduced by low cost devices/construction material [36]. Floating hydroponic system used in South East Asia is an example of low cost approach [33].

Hydroponic farming is a relatively new practice in some of the farmers in developing countries with smallholder farmers barely having knowledge about it. Most smallholder farmers practice mixed farming: farmers grow crops and keep animals. Hydroponic farming is, as such, an approach that can be used to produce food crops and fodder for farm animals. Closed hydroponic could address problems faced currently due to scarcity of water and rainfall variability. The recycling of water could affect production and necessary measurements and monitoring need to be done for the farm to be economically viable. Moreover as stated in table 1 hydroponic system needs higher knowledge on technology as compared to traditional farming. IoT could solve these problem. Farmers do need to rely on experts as the information they need to make decisions is made available to them by data from sensors. Sensors can collect data of the ingredients of the solutions and this can help farmers make informed decisions at the right time. Nutrient imbalance can easily be identified and necessary action taken at the right time.

(a) deep water culture (b) Nutrient Film Technique

Figure 7: Hydroponic Systems ([33])

	Soil	Soilless
Farming in new areas	Not always possible. Depends on the type of soil, fertility, salinity	Agriculture possible in any condition
Cultivation	Constant preparation of soil, need of machines, fuel intensive	Not needed, substrates preparation or positioning on troughs/ground
Intensification of production	Limited. Monoculture brings “soil tiredness” and already decreases yields after two successive crops. Soil tiredness requires crop rotation, fallow or soil sterilization, which is time consuming and interrupts crop cycles for 2–3 weeks	Monoculture is possible with no decadence of performances Substrates could be sterilized with simple means and no crop interruptions Inert media or water do not face risk of any fertility losses due to their characteristics
Plant nutrition	Variable delivery. The release depends on soil characteristics. Some deficiencies are possible. The precise delivery of nutrients according to the plant growth stage is not possible	Real time distribution of nutrients and pH according to the growth stage of the plants. Real-time control of the levels of nutrients required by plants
Nutrient use efficiency	Fertilizers broadcasted broadly, High dispersal through leaching and runoff in outdoor conditions	Minimal amount required due to microirrigation and containment of media. Water and nutrients monitoring avoid the loss of nutrients
Water use efficiency	Efficiency affected by soil texture and irrigation system	Optimal delivery trough microirrigation supported by sensors
Weed control	Need continuous control	No need of any control
Diseases and pests	Affected by soil-borne diseases and pests. Needs sterilization, crop rotation	Not affected because of no use of soil
Quality	Product characteristics depends on of the type of soil and management	Standardized production with full control of nutrients. Optimized growth
Production costs	Normal, but use of machinery necessary for soil cultivation and higher use of inputs (water). Higher costs if greenhouses/nethouses are used	Higher costs due to more expensive setting in greenhouses/nethouses and the presence of a monitoring system,
Farm management	Standard level	Expert level. Needs higher knowledge for the higher technology used

Table 1: Soil versus soilless production (Source: [33]).

2.4 Requirements of smart farming in resource constrained regions

As explained in section 2.1.2 infrastructural, socio-economical and cost are some of the factors that contribute to low uptake of the technologies in emerging economies. As such, the technical requirements suggested for the smart farming countries have to put these factors into consideration. For the sustainability and enhanced use of the technologies, the solutions should be easy for the local communities and give new meanings in their own context. In addition, they should foster local digital capacity and innovations.

2.4.1 Low cost device

Smallholder farmers have limited resources to invest in technology. As such, computing and sensor devices for smart solutions have to be low cost.

2.4.2 Low power device

Power connectivity is not given in most of these regions and if it is available, power outages are frequent. Rechargeable batteries and the solar panels should thus be used to power the system or act as a back-up in case of outage. Furthermore, use of solar panels is a cheap, clean and sustainable source of energy.

2.4.3 Cost-efficient communication

Internet connectivity is unavailable, intermittent, slow or costly in most of the developing countries. Consequently, the solutions needed should include use of unlicensed bands for IoT connectivity. Communication between devices and particularly wireless communication is power consuming, thus solutions that offer efficient communication, low power consumption and routing protocols with low memory requirement are required [37]. Also a cost-efficient communication is required for sending data to the cloud. Since bandwidth is limited, data mitigation techniques [27] are required in such areas to reduce the amount of bandwidth needed to send data to the cloud.

2.4.4 Software

There are many commercial and open source software for IoT with respective strengths and weaknesses. Cost is a limiting factor when considering proprietary software. As such open source software should be used. Open software enables researchers to replicate the design and customize it to meet specific needs of the context [21]. Most IoT devices are resource constrained and battery powered. Therefore, software used should be low memory consuming.

2.4.5 Computation and storage

The data collected by the sensors need to be processed and stored. Cloud computing offers limitless on-demand storage and computation capacity. A key problem with the use of cloud computing is a need for connectivity to the internet which is not realistic in most of the developing countries because of cost and limited network coverage. Edge computing can substitute in areas with no coverage and complement cloud computing in areas with limited network coverage. Edge enables storage and processing of data locally and make it accessible to the users [38, 21].

2.4.6 Scalability

Scalability involves ability of system to adapt to changes e.g. increase in number of devices connected while giving optimal performance. In this case, the system should be able to accommodate the connection of new hydroponic farms and scale with increase in data from devices.

2.4.7 Ease of use and sustainability

Given that most smallholder farmers are not tech savvy, they need a system that is easy to operate without continuous technical support. The system should also be adaptable to different farm sizes and a short learning curve for farmers [11]. In addition, the system should equip farmers with skills and build the capacity of the communities through provision of access to information about smart farming.

3 Enabling Technologies and Related Work

Heterogeneous technologies enable smart farming by facilitating sensing, transmission, analysing and storage of data. In this section, we will discuss how different technologies can be applied to meet the requirement of smart farming in emerging economies. In section 3.1 we discuss the IoT and how it promotes connection of things and data collection. In section 3.1.1 we introduce different IoT protocols. Section 3.2 discusses the different wireless technologies and evaluate their applicability in smart farming. We discuss the trend on LPWANs and the opportunities they offer for IoT. Section 3.3 discusses the role of cloud and edge computing and how they contribute to data processing and storage and the sharing of information among different stakeholders. In the last section, we discuss a selection of related work on smart farming and how they make use of the technologies discussed in the previous section.

3.1 Internet of Things

The term 'Internet of Things' was coined in 1999 by Kevin Ashton and is generally viewed as interconnected devices, objects, people and software. Internet of Things is rapidly developing, and it continues to receive much attention due to many markets and applications scenarios it offers. CISCO estimates that there will be 50 billion devices connected by 2020 [39] and McKinsey Global Institute estimated in 2015 that IoT will have an economic impact of between \$3.9 trillion to \$11.1 trillion per year in 2025 [40]. IoT is a combination of technological push, human pull for connectivity between the immediate and wider environment and it emerged from development in identification technologies e.g. RFID and barcodes and from development of networked sensors and actuators [41].

There is no agreed definition for the Internet of things. According to European Research Cluster on the Internet of Things (IERC), IoT is

"A dynamic global network infrastructure with self-configuring capabilities based on standard and interoperable communication protocols where physical and virtual "things" have identities, physical attributes, and virtual personalities and use intelligent interfaces, and are seamlessly integrated into the information network, often communicate data associated with users and their environments” [42].

A user centric definition is given by Gubbi et al. IoT is “Interconnection of sensing and actuating devices providing the ability to share information across platforms through a unified framework, developing a common operating picture for enabling innovative applications. This is achieved by seamless large scale sensing, data analytics and information representation using cutting edge ubiquitous sensing and cloud computing” [41].

IoT has many applications areas and Asghari, Rahmani and Javadi have given a comprehensive taxonomy of different applications including health-care, environmental, smart city, commercial, industrial and general aspects[43]. Smart farming/agriculture is a subsection of environmental application scenario. IoT platforms are used different agricultural sectors and the following are some examples: a henhouse to monitor and control environmental factors (temperature, humidity, carbon dioxide, ammonia levels) [44], hydroponic greenhouse [45], monitoring and control of irrigation system in rural communities [46], smart irrigation in tunnel farming [47], smart animal farm [48].

A generic three-layer IoT architecture consisting of sensing, transport and application layer is depicted in Figure 8 and it can also be extended to five layers with inclusion of network and processing layers between the second and third layer [49, 37]

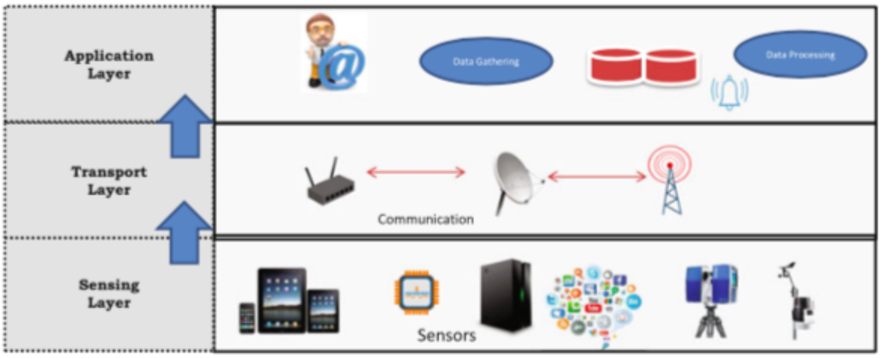


Figure 8: IoT based agricultural framework (Source: [49]).

IoT in agriculture consists of several layers of interconnected things and interfaces. Ray provides a six layer framework for a fully fledged agricultural solution based on IoT [30]. Figure 9 shows these six layers and interconnection between them. However, the service layer in this framework doesn't include edge plane and data is directly sent to the cloud and no analysis of data is done either at this stage.

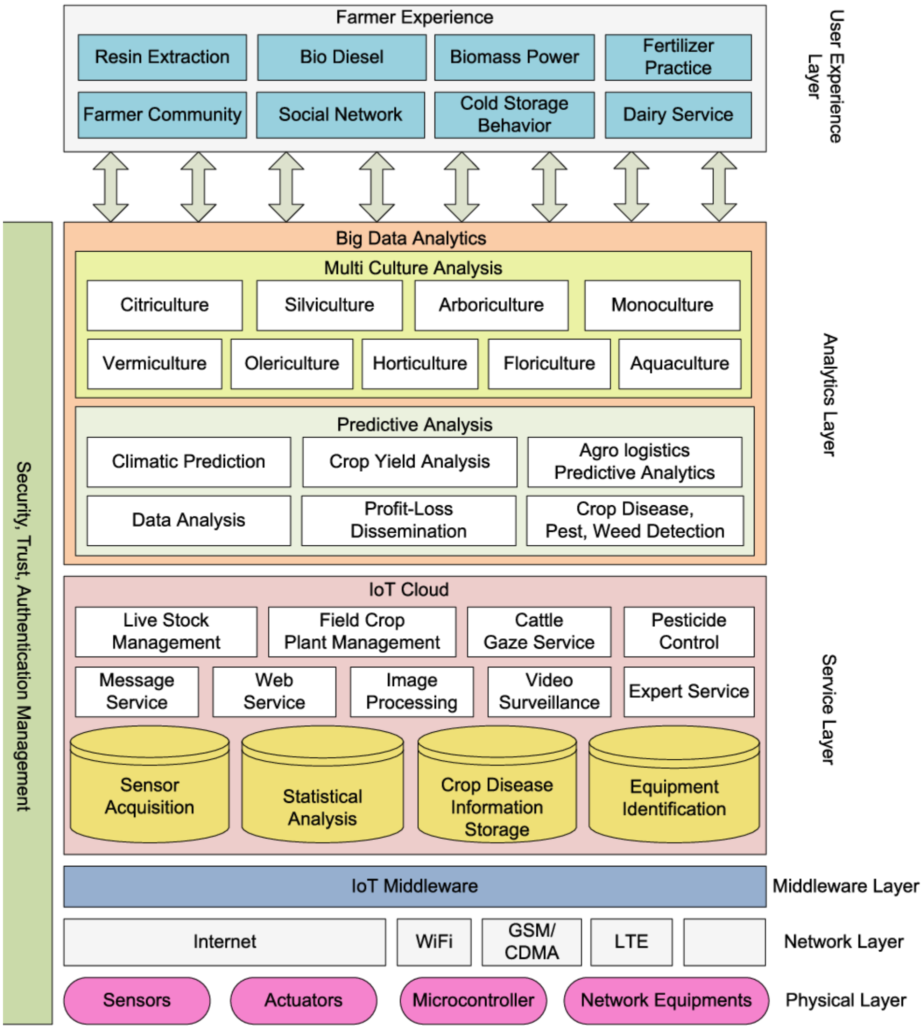


Figure 9: IoT based agricultural framework (Source: [30]).

Even though 20 years have passed since IoT was first introduced, there is no unified IoT architecture and different protocols and standards are used to connect IoT parts depending on the IoT use case. In the next section we will discuss existing IoT application layer protocols that are currently used by developers and researchers.

3.1.1 IoT application layer protocols

Application layer protocols are used to update the online servers with the current readings of the sensor nodes and also carry commands from applications to the sensor nodes [50]. Figure 10 illustrates the communication between end devices, online servers and applications. Several application layer protocols have been suggested and these include Constrained Application Protocol (CoAP), Message Queuing Telemetry Transport (MQTT), Extensible Messaging and Presence Protocol (XMPP), RESTFUL Services (Representational State Transfer), AMQP (Advanced Message Queuing Protocol - a corporate messaging protocol that emerged from financial industry [50]), Websockets and HTTP (designed for WEB and not optimal for IoT as it is heavy weight protocol [51]). In this section we will only consider MQTT and CoAP, which are the most common protocols in IoT systems.

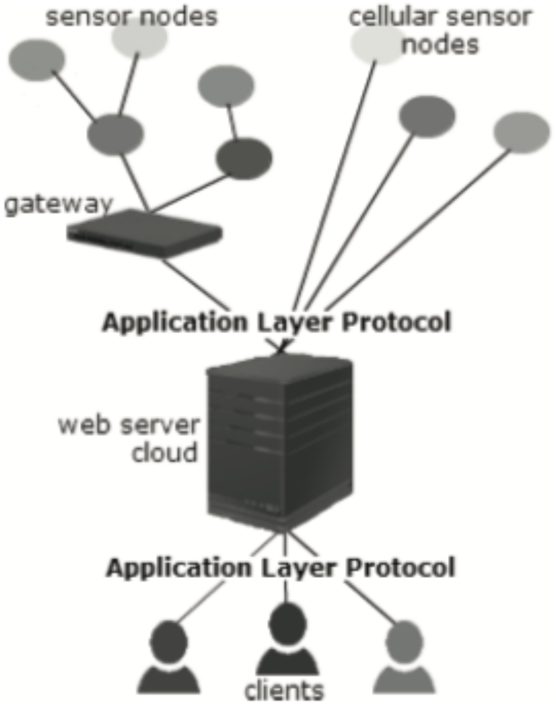
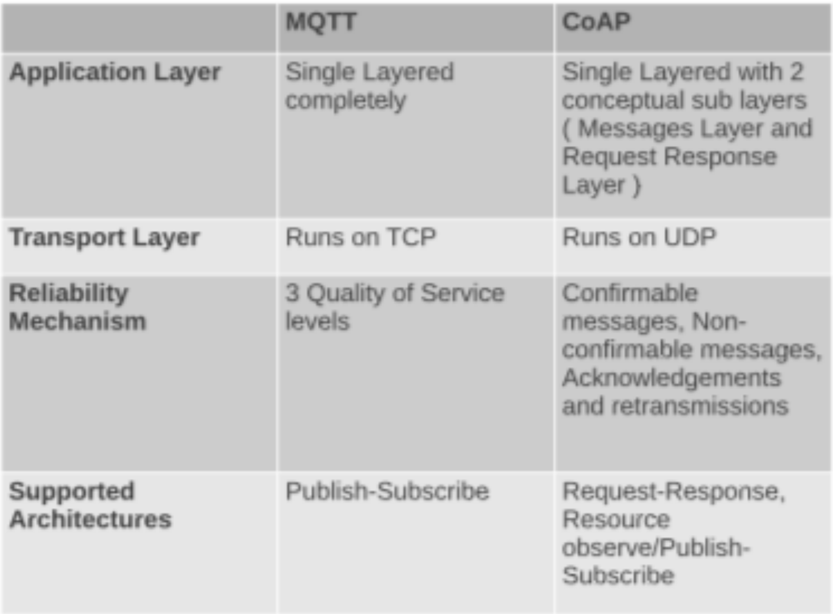


Figure 10: Application layer protocols source ([50]).

3.1.1.1 CoAP The Constrained Application Protocol (CoAP) was designed by Internet Engineering Task Force (IETF) to address the requirements of resource constrained devices[50]. It uses request/response and resource/observe (variant of publish/subscribe) architecture making it interoperable with HTTP [51]. It is uses Universal Resource Identifier (URI) rather than topics thus publishing and subscription are done to a specific URI. It is a UDP based protocol, Datagram Transport Layer Security (DTLS) is used for security and to achieve reliability and Quality of Service (QoS), it utilizes four message types: Confirmable(message needs acknowledgement by the receiver), Non-Confirmable(message doesn't need acknowledgment), Acknowledgment(reception of confirmable message confirmed) and Reset (message received but couldn't be processed) [50]. Authors in [50] argue that even though CoAP is designed for IoT, its use of DTLS for security increases network traffic as DTLS handshakes require additional packets and computation resources thus affecting the battery lifespan on the end devices.



	MQTT	**CoAP**
Application Layer	Single Layered completely	Single Layered with 2 conceptual sub layers (Messages Layer and Request Response Layer)
Transport Layer	Runs on TCP	Runs on UDP
Reliability Mechanism	3 Quality of Service levels	Confirmable messages, Non-confirmable messages, Acknowledgements and retransmissions
Supported Architectures	Publish-Subscribe	Request-Response, Resource observe/Publish-Subscribe

Figure 11: Difference between CoAP and MQTT. source ([52]).

3.1.1.2 MQTT Message Queuing Telemetry Transport Protocol is a light-weight publish/subscribe protocol that uses topics as the addresses where the messages are published to and subscribed to by the clients [51]. Topics are contained in a broker [50] - these are servers that publishers send messages to and where clients automatically receive updates on the topic they subscribed to. MQTT runs on TCP and uses TLS/SSL for security [51].

3.1.1.3 Perfomance evaluation Naik did an in-depth comparative study of four (HTTP, AMQP, MQTT and CoAP) application layer protocols [51]. According to the author [51] CoAP requires lowest power and lower bandwidth than MQTT in transferring same payload under same network conditions. However, MQTT does better in terms of Quality of Services and reliability. In addition, MQTT is used by large number of organizations in the world but not yet global standard as HTTP. In [52], performance analysis between MQTT and CoAP shows that performance of the protocols depend on the network condition: MQTT packets have low delays for lower packet loss but CoAP performs better if the value of packet loss increase due to smaller UDP headers as compared to TCP headers required in retransmission of message. They also suggest that difference in performance can be exploited at the gateway by detecting network condition and using the protocol that gives best performance depending on prevailing network conditions. Whereas smart gateway has the above mentioned advantages, we have not implemented it in this thesis. The choice of these protocol also depends on the conditions and requirements of the IoT system. In this thesis, we have used MQTT in our implementation.

3.2 Wireless Communication Standards

Traditionally, connectivity in IoT has mainly been provided by short-range multi-hop technologies based on the unlicensed spectrum or long-range cellular networks. A new promising solution for IoT wireless connectivity is the Low Power Wide Area Network (LPWAN), which offers long-range [53]. This section will look at these three approaches and discuss their feasibility in smart agriculture in emerging economies.

3.2.1 Short-range communication

The most common short-range wireless technologies include Bluetooth, ZigBee, near field communications (NFC), radio frequency identification (RFID), 6LoWPAN, Thread, Wi-Fi and Z- wave which is a proprietary system [53].

These technologies are from different vendors, and one of the biggest challenges is interoperability. This problem is addressed by different organizations that define standardization procedures and testing to guarantee interoperability between devices [54].

Short-range technologies have the advantage of low power consumption- a requirement in IoT but they have limited coverage, which hinders its application in some IoT scenarios. These technologies are primarily used in personal area networks or local area networks. In the following section, we will discuss the different features of four of the common short-range communication technologies.

3.2.1.1 Bluetooth Bluetooth is a wireless communication technology operating on 2.4Ghz and was previously standardized as IEEE 802.15.1 but currently maintained by Bluetooth SIG [54]. It is mainly used in personal area network with a range of up to 10 meters. It uses star network topology. It is a low power technology and devices are mostly battery powered. It has a throughput of up to 2MBps. Bluetooth Low Energy (BLE) is a new standard aimed at reducing power consumption and increasing the life-time of the coin cell batteries while the downside of this is low data throughput [54].

3.2.1.2 ZigBee ZigBee is based on IEEE 802.15.4 link layer standard and is managed by ZigBee Alliance. It is a low power, low cost and low throughput (up to 250KBps) with a mesh network topology making it possible to connect with thousands of nodes [54]. ZigBee network requires an application-level gateway to connect to the Internet. ZigBee has a low-duty cycle and is suitable for agricultural applications where periodic information update is needed such as irrigation management, pesticide and fertilizer control, and water quality management [55].

3.2.1.3 6LoWPAN 6LoWPAN 6LowPAN (IPv6 over Low power Wireless Personal Area Networks) is a standard by the 6LoWPAN working group of the Internet Engineering Task Force (IETF). Compared to the other standards above, 6LoWPAN enables devices to directly communicate over the Internet [54]. It operates on 2.4-GHz and the 868MHz/915MHz ISM bands and it uses mesh network topology. It only supports IP version 6 (IPv6) thus it requires an IPv6-to-IP version 4 (IPv4) conversion protocol in the gateway [54].

3.2.1.4 Wi-Fi Wi-Fi is based on the IEEE 802.11 standard. It operates on 2.4 GHz and 5 GHz with a star topology and access point (AP) as a gateway. It has a range of 100m and a throughput of up to 72Mbps [56]. Most of the new devices come with Wi-Fi software and the TCP/IP software making integration easier. The downside of this standard is that it has high power consumption mainly due to high data rate and coverage. However, advanced sleep protocols and power management design mechanisms have been included to increase the lifetime of battery-powered devices [54]. In agricultural applications, WiFi enables the connection of multiple types of devices through heterogeneous architectures over an ad-hoc network [55].

3.2.2 Low-Power Wide-Area Networks (LPWANs)

Low-Power Wide-Area Networks utilize unlicensed frequency bands (2.4 GHz, 868/915 MHz, 433 MHz, and 169 MHz depending on region) and have star network topology [57]. There are known for low power consumption and wide area coverage hence there are termed as Low-Power Wide-Area (LPWA) technology. The new physical layer design aimed at very high receiver sensitivity enables short-range devices to have coverage of about 10-15 and 2–5 km in rural and urban areas respectively [54]. According to [53], the use of this paradigm in IoT connectivity with long-range and low data rates is encouraged by the sporadic transmissions of very small packets by the IoT devices. The end devices connect to the Internet through a gateway. Some LPWAN solutions include LoRa, Sigfox, Ingenu-RPMA, DASH7, Weightless [53, 57]. DASH7 and Weightless are open sources while the rest are proprietary systems. In the following section, we look at three of the most common LPWANs.

3.2.2.1 Long Range Radio (LoRa) LoRa is a spread spectrum modulation technique developed by Semtec [3], which is based on chirp spread spectrum (CSS) technology [58]. LoRa physical layer enables long-range communication and it operates on different frequencies depending on the region: 902–928 MHz

---

[3] `https://www.semtech.com/lora/what-is-lora`

band (United States) and 863– 870 MHz band (Europe). However, it can also work on lower ISM bands at 433 MHz and 169 MHz [53]. LoRa is a proprietary product and one of the most used communication protocols built above the LoRa is LoRaWAN. LoRaWAN is an open communication protocol and network system architecture [59] by LoRa Alliance[4], a nonprofit association. LoRaWAN network architecture consists of the end nodes, gateway, and network server. The network server handles all the complexities related to packets de-duplication and decoding [58]. The end devices communicate with the gateway using LoRa and from the gateway, packets are forwarded to the network server through backhaul interfaces like 3G or Ethernet [27].

3.2.2.2 SigFox SigFox is based on ultra-narrowband technology (UNB) and it uses 915MHz ISM band (United States) and 868MHz (Europe) [56]. It was first released in 2009 and IoT service provider as its business model thus no documentation is openly available [53]. The communication range is up to 30 km and this is achieved by transmitting at very low data rates (up to 100bps) [56].

3.2.2.3 Ingenu-RPMA Ingenu-RPMA is a proprietary technology by On-Ramp Wireless which developed 802.15.4k standard and owns the right to Random Phase Multiple Access technology [53]. According to [57] Ingenu-RPMA achieves higher throughput and capacity compared to other technologies that operate on sub-GHz band because of its flexibility in using spectrum across different regions. It has a typical uplink data rate of 50 kbps [60].

3.2.3 Cellular Network

The cellular network is an established worldwide system with a potential for providing ubiquitous access. These include GSM, UMTS, and LTE networks. It is considered as a prominent candidate in the provision connectivity to IoT due to its capillary geographical coverage, technological maturity and cost-effectiveness due to high revenue it generates from other services like video, voice, and data [61]. However, due to the expected growth of IoT devices and the sporadic nature of traffic generated by them, the current cellular network

---

[4] `https://lora-alliance.org`

could collapse because of signaling traffic from these devices [61, 53]. To address these shortcomings, revamping of second generation/ Global System for Mobile Communications (2G/ GSM) [53] and LPWA solutions have been introduced to cope with the requirements of IoT. The solutions introduced by Third Generation Partnership Project (3GPP) include Extended Coverage – GSM – Internet of Things (EC-GSM-IoT), enhanced machine-type communication (eMTC),Long-Term Evolution (LTE) and narrowband Internet of Things (NB-IoT) [57]. Fifth generation (5G) standards have been released in 2018 and the earliest deployment is expected in the second quarter of 2019 whereas sixth generation (6G) is just on its start in terms of research and artificial intelligence (AI) is seen as the driver for 6G [62].

3.2.4 Connectivity with alternative low cost networks

Bringing Internet connectivity to remote regions does not make a good business case for mainstream network providers. Alternative Networks have emerged and deployed in areas where that traditional network could not cover because of high initial and operational costs, privacy concerns and limited connection to the power grid [63]. Alternative networks are mostly small scale, individuals and other interested stakeholders share the cost of setting up and maintenance expenses. Most smallholder farmers in developing countries are not connected to the Internet. This hinders access to information related to smart farming and emerging technologies in general. The provision of internet connectivity is important in solving this information asymmetry.

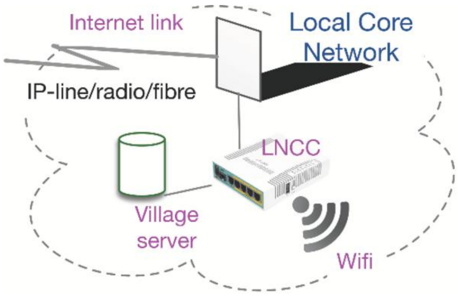
Internet Lite is a solution by Basic Internet Foundation aimed at addressing the digital divide challenge [64]. It aims at providing affordable internet access to the residents of the developing countries and thereby bridging the digital divide and working towards the achievement of the UN sustainable development goals(SDG) where the Internet is seen as an enabler in attaining these goals. The broadband service provided by traditional mobile service providers continues to be expensive and limited thus limiting the opportunities offered by the Internet to attain SDG.

To achieve this, the Basic Internet Foundation used a low-cost network infrastructure that includes a local core network, a local network, a centralized core, and backhaul network [65, 66]. Figure 12a illustrates these components. Sudhir and Noll have defined InfoInternet standard that is aimed at making access to information free (text and pictures) [66]. This is implemented in the Local Network Control Centre (LNCC)[5]. Internet Lite solution complies with the net neutrality requirement by restricting the content type, not the content. Contents are filtered depending on the number of bits consumed. This approach accommodates both the users of basic Internet and users with paid subscriptions. For the users of Internet Lite, the dynamic content e.g. video is filtered out while the text and pictures are allowed while if a user has a voucher, then all content is allowed.

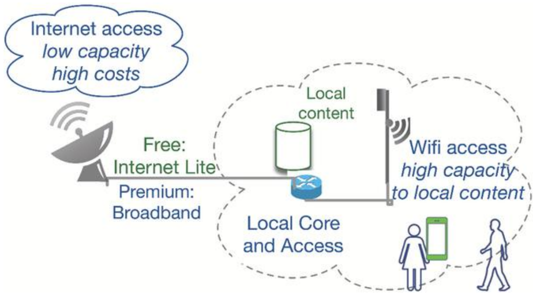
The solution offered by Basic Internet is relevant to smart farming in developing countries for these two reasons. 1) Basic Internet solution has WiFi access point (Information spots) where farmers can access information as shown in figure 12b. 2) Internet Lite offers free connectivity to the Internet for text and pictures. This can be used to share data from farms to other stakeholders i.e. for transmitting data stored at the local servers to the cloud where other stakeholders can access it. How this solution is integrated into the solution proposed in this thesis will be discussed in section 4.

---

[5] `https://its-wiki.no/wiki/BasicInternet:Home/NOsolutions`



(a) Local core network



(b) WiFi access point

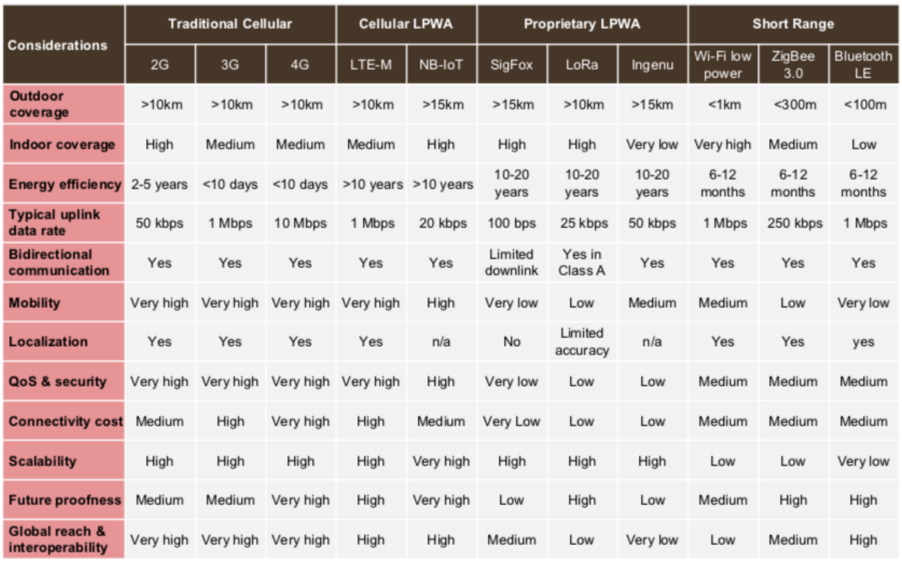
Figure 12: Internet Lite source ([64])

3.2.5 Applicability in Smart farming

Figure 13 depicts a comparison of the main wireless communication technologies and parameters such as transmission range, data rate, energy consumption, and cost. All these technologies have their strengths and weaknesses and therefore a choice depends on the application scenario. In this thesis, we are considering a smart hydroponic farm in the resource-constrained region. Hydroponic system requires monitoring of nutrient solutions and other factors within the greenhouse for efficient food production. In addition, farmers do not afford to install a complete monitoring system due to cost. We, therefore, consider a scenario where farmers have shared infrastructure such that the sensors deployed in individual farms transmit data to community-owned gateway and local server that hosts the network server. In such scenario range of wireless technology becomes a vital factor to consider because hydroponic farms owned by smallholder farmers are mostly located in different parts of a village. Moreover, high energy efficiency and low cost is a requirement in such scenario.

	Spectrum cost	Deployment cost	End-device cost
Sigfox	Free	>4000€/base station	<2 €
LoRa	Free	>100€/gateway >1000€/base station	3-5€
NB-IoT	>500 M€/MHz	>15000€/base station	>20 €

Table 2: A cost comparison of LoRa, SigFox and NB-IoT (Source: [67]).



Considerations	Traditional Cellular			Cellular LPWA		Proprietary LPWA			Short Range		
	2G	3G	4G	LTE-M	NB-IoT	SigFox	LoRa	Ingenu	Wi-Fi low power	ZigBee 3.0	Bluetooth LE
Outdoor coverage	>10km	>10km	>10km	>10km	>15km	>15km	>10km	>15km	<1km	<300m	<100m
Indoor coverage	High	Medium	Medium	Medium	High	High	High	Very low	Very high	Medium	Low
Energy efficiency	2-5 years	<10 days	<10 days	>10 years	>10 years	10-20 years	10-20 years	10-20 years	6-12 months	6-12 months	6-12 months
Typical uplink data rate	50 kbps	1 Mbps	10 Mbps	1 Mbps	20 kbps	100 bps	25 kbps	50 kbps	1 Mbps	250 kbps	1 Mbps
Bidirectional communication	Yes	Yes	Yes	Yes	Yes	Limited downlink	Yes in Class A	Yes	Yes	Yes	Yes
Mobility	Very high	Very high	Very high	Very high	High	Very low	Low	Medium	Medium	Low	Very low
Localization	Yes	Yes	Yes	Yes	n/a	No	Limited accuracy	n/a	Yes	Yes	yes
QoS & security	Very high	Very high	Very high	Very high	High	Very low	Low	Low	Medium	Medium	Medium
Connectivity cost	Medium	High	Very high	High	Medium	Very Low	Low	Low	Medium	Medium	Medium
Scalability	High	High	High	High	Very high	High	High	High	Low	Low	Very low
Future proofness	Medium	Medium	Very high	High	Very high	Low	High	Low	Medium	High	High
Global reach & interoperability	Very high	Very high	Very high	High	High	Medium	Low	Very low	Low	Medium	High

Figure 13: A comparison of different wireless technologies (Source: [60]).

With this in mind, we first consider the feasible wireless technology to connect the devices to the gateway and then backhaul connectivity between the edge layer and cloud. Bluetooth, ZigBee, 6LoWPAN, and WiFi all have a short communication range. This will require high node density to cover a small area which also adds complexity and reduce battery lifetime. As such, there are not suitable for farming that requires a shared infrastructure where long-range is a requirement.

A comparative study of LWPAN technologies is given [67, 68]. Mekki et al. compare large-scale deployment of LoRa, SigFox and NB-IoT [67]. From this comparison, Lora and SigFox are considered as cost-effective as spectrum and deployment cost for NB-IoT is high. SigFox end devices are cheaper but the deployment cost is high and on the other hand, LoRa end devices are slightly expensive but its deployment cost is lower. Table 2 shows this comparison.

Even though LoRa is a proprietary product, its upper layer, LoRaWAN is open, operator and subscription-free making it simple to deploy and manage its infrastructure whereas, in SigFox, users purchase end devices and subscription for the devices from the network operators [68]. In terms of cost, openness, and availability SigFox is currently not feasible in most developing countries. Ingenu-RPMA has several private deployments that require a yearly subscription and upfront payment per application and device [69]. This makes it unsuitable for deployment in rural areas in developing countries.

LWPANs generally offer a longer range and a limited throughput. LoRa offers long-range and low bandwidth and it compliments and fills the gap between cellular and short-range technologies to meet the requirements of IoT use case scenarios. Figure 14 shows this comparison. This makes it suitable for scenarios like smart farming in rural areas where farms are spread in a large area and the data from sensors in farms are short and sporadic. Transmission of data can also be limited to when a certain threshold is met.

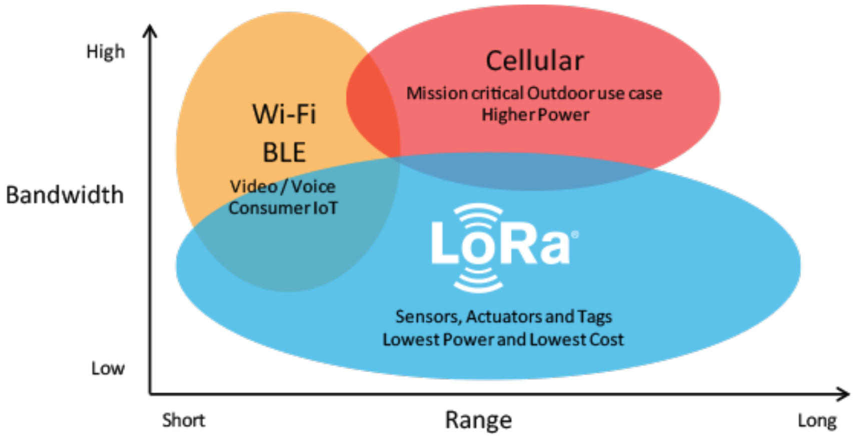


Figure 14: A range comparison of short range technologies, cellular and LoRa.

The flexibility offered by LoRa ecosystem makes it suitable for local deployment [67] and is ideal for deployment in rural areas. Low-cost single board computers and micro-controllers like Raspberry Pi and Arduino can be used to construct gateways and end devices to reduce cost [70]. The proliferation of low-cost hardware, availability of open-source software and initiatives like Sparkfun[6] and Adafruit [7] has led to the third wave of Do-It-Yourself(DIY) which is seen

---

[6] `https://www.sparkfun.com/categories/23`

[7] `https://www.adafruit.com`

as revolutionary, enabling anybody anywhere to create innovative solutions and this suits well regions where industrial manufacturing infrastructure is lacking [71].

Cellular is widely available in most developing countries. The technology is mature, secure and it offers a high quality of service. The disadvantage is that devices need sim cards to connect to the network and data plans offered in developing countries are very expensive. Cellular LWPANs are not yet deployed in most of these countries and are thus not feasible for smart farming in the near future. In addition, there are not cost-effective e.g NB-IoT as shown in table 2. However, cellular is suitable for backhaul connectivity. From the edge server, the data can be consolidated and sent to the cloud regularly depending on the needs of the smart farm ecosystem. Despite wide coverage and its presence in almost every part of the world, it continues to be expensive, especially in developing countries. Thus low-cost connectivity provided by Basic Internet can complement it.

Table 3 shows a comparison of different wireless technologies that are applicable to smart farming. We considered four main factors based on the application scenario: cost, power efficiency, range, availability and openness. The following symbol legends are used in table 3.

more suitable: ++, suitable:+

less suitable: −, least suitable − −

Table 3: Cost, Energy efficiency and range

Wireless technology	Cost	Energy efficiency	Range	Availability and openness
Bluetooth	++	++	− −	++
ZigBee	++	++	− −	++
6LoWPAN	++	++	− −	++
Wi-Fi	++	− −	− −	++
LoRa	++	++	++	++
SigFox	+	++	++	− −
Ingenu-RPMA	++	++	-	− −
Cellular-LWPN	− −	++	++	− −

As shown in table 3 features of LoRa are favourable for the implementation in rural areas. It offers longer range, cost, and power efficiency. It also has LoRaWAN protocol which is open and enables deployment of network anywhere. From this analysis, we consider LoRa as a suitable solution for IoT connectivity. Solutions offered by Basic Internet architecture can then complement cellular for backhaul connectivity. Basic Internet solution also offers WiFi access points that are suitable for access to farm data that is stored locally.

3.2.6 Fundamentals of LoRa

As described in section 3.2.2.1, LoRa, a technology by Semtech, uses chirp spread spectrum(CSS) modulation to achieve long-range while maintaining low power usage. The carrier signals in LoRa have chirps that enable signals to travel long-range and still be demodulated even if the signal power is 20dB below the noise floor [72]. LoRa offers configuration parameters that can be modified to achieve different power consumptions, transmission distance, and data rates. According to Bor and Roedig, LoRa device configuration involves a combination of different bandwidth, spreading factors, coding rate and transmission power resulting in over 6720 settings combinations [73]. In the following section, we look at these four parameters, what they mean in LoRa and the inevitable trade-off because of different combinations of these factors. Figure 17 shows combination of different BW,CR,SF and payload size and the resulting time on air. Table 4 gives a summary of this parameters.

Spreading Factor (SF): SF refers to how spread a chirp is and the *spreadness* is dependent on the numbers of bits in a chirp [74]. LoRa offers a spreading factor of between SF6 and SF12. An increase in SF reduces the transmission rate by half and doubles the airtime of the packet, thus increase in power consumption [73]. However, the increase of transmission time gives the receiver enough chances to sample the signal which results in higher signal-to-noise ratio(SNR) increasing the probability of decoding correctly [74]. SF6 is used when the receiver is close to the transmitter and to decode received signals with power as low as -136 dBm due to long distance or obstacles in the path, a SF of 12 is used [75].

Coding Rate(CR): CR is a forward error correction code aimed at increasing resilience against interference [74]. These are 4/5, 4/6, 4/7 or 4/8.

Settings	Values	Effect
Bandwidth	125 . . . 500 kHz	Higher bandwidths allow for transmitting packets at higher data rates (1 kHz = 1 kcps), but reduce receiver sensitivity and communication range.
Spreading Factor	$2^6 \ldots 2^{12}$ $\frac{chips}{symbol}$	Bigger spreading factors increase the signal-to-noise ratio and hence radio sensitivity, augmenting the communication range at the cost of longer packets and hence a higher energy expenditure.
Coding Rate	4/5 . . . 4/8	Larger coding rates increase the resilience to interference bursts and decoding errors at the cost of longer packets and a higher energy expenditure.
Transmission Power	4, . . . , 20 dBm	Higher transmission powers reduce the signal-to-noise ratio at the cost of an increase in the energy consumption of the transmitter.

Table 4: LoRa configurations and effects on communication performance (Source: [72]).

In LoRa, 4/5 CR means that for four bits of data 1 bit is added. Higher CR leads to higher transmission time due to the increased number of bits but offers improved protection from interference [73].

Bandwidth (BW): BW is a range of frequencies between the upper and lower frequencies of the transmission band. High bandwidth gives a higher rate thus shorter air time but with a lower sensitivity [73]. 125kHz, 250kHz, and 500kHz are mostly used in LoRa.

Transmission Power (TX): LoRa permits adjustment of transmission power like other wireless radios. Transmission power directly affects the amount of power required to transmit a packet. Therefore, higher TX increases the SNR thus improving chances of the packet being received and survival against attenuation caused by the environment at the cost of increased energy usage at the transmitting end.

LoRaWAN: While LoRa defines the physical layer which is responsible for long-range communication, LoRaWAN defines the system architecture for the network and the communication protocol. Figure 16 shows the LoRa and LoRaWAN protocol stack. According to LoRaWAN specifications, the network architecture consists of the end nodes, gateways, network server and the application as shown in figure 15. To avoid the complexity and battery effect of mesh network architecture, LoRaWAN employs a star topology [59]. The end nodes are not associated with any gateway. Because of this, multiple gateways can receive data from end nodes. The network server has the purpose of deduplicating the packets sent by end devices, data authentication, and sending acknowledgements.

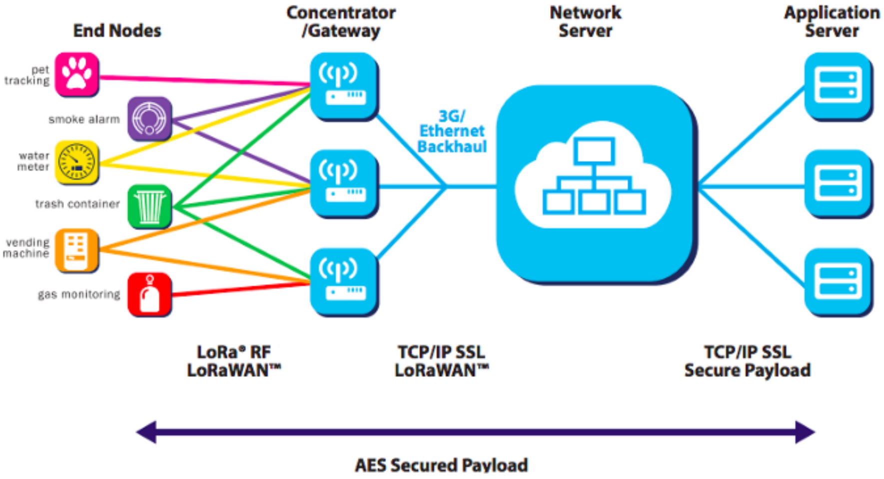
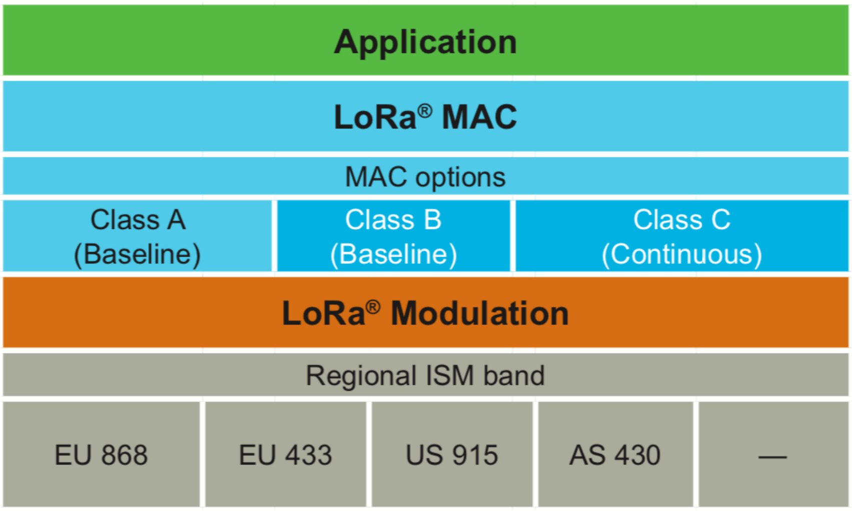


Figure 15: LoRaWAN network architecture (Source: [59]).


Application

LoRa® MAC

MAC options

Class A (Baseline) | Class B (Baseline) | Class C (Continuous)

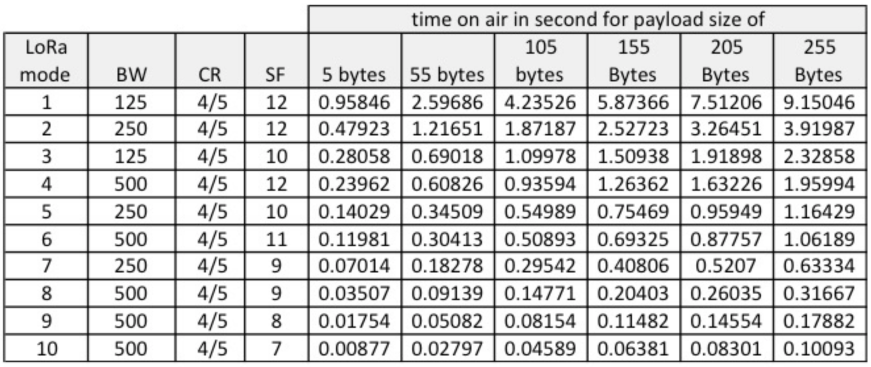
LoRa® Modulation

Regional ISM band

EU 868 | EU 433 | US 915 | AS 430 | —

Figure 16: LoRaWAN protocol stack (Source: [59]).

The LoRaWAN network server optimizes data rates and battery lifetime using adaptive data rates (ADR) scheme [59]. The ADR is determined by maximum SNR of the last 20 received uplink messages and from this, the network server optimizes the message duration to ensure as lowest TX power is used [76]. ADR also enhances the overall capacity of the network and scalability. With ADR the network is scalable i.e. increased number of nodes supported as compared to default LoRaWAN settings [68]. The scalability is also affected by the regulatory constraints on the use of physical medium since LoRa is using ISM bands. The imposed duty cycle for LoRa is 1%.



				time on air in second for payload size of					
LoRa mode	BW	CR	SF	5 bytes	55 bytes	105 bytes	155 Bytes	205 Bytes	255 Bytes
1	125	4/5	12	0.95846	2.59686	4.23526	5.87366	7.51206	9.15046
2	250	4/5	12	0.47923	1.21651	1.87187	2.52723	3.26451	3.91987
3	125	4/5	10	0.28058	0.69018	1.09978	1.50938	1.91898	2.32858
4	500	4/5	12	0.23962	0.60826	0.93594	1.26362	1.63226	1.95994
5	250	4/5	10	0.14029	0.34509	0.54989	0.75469	0.95949	1.16429
6	500	4/5	11	0.11981	0.30413	0.50893	0.69325	0.87757	1.06189
7	250	4/5	9	0.07014	0.18278	0.29542	0.40806	0.5207	0.63334
8	500	4/5	9	0.03507	0.09139	0.14771	0.20403	0.26035	0.31667
9	500	4/5	8	0.01754	0.05082	0.08154	0.11482	0.14554	0.17882
10	500	4/5	7	0.00877	0.02797	0.04589	0.06381	0.08301	0.10093

Figure 17: Time on air for different payload sizes (Source: [70]).

Device Type: LoRa also offers better energy efficiency and is suitable for rural areas as the connection to the power grid is not guaranteed. LoRaWAN has three end device classifications [59]. (Fig 18):

- Class A: end device transmission followed by two short download windows.
- Class B: scheduled receive slots through synchronization by gateway beacon.
- Class C: continuously listening.

According to a predictive model by Liando et al. on the lifetime of end nodes, battery efficiency, and longevity can be increased by choosing carefully the microcontrollers unit used in end nodes and using the right combination of spreading factor, transmission power and duty cycle [74]. Choosing the right combination of hardware and settings is particularly important in rural areas where connection to power is not guaranteed and also reduces the cost of replacing batteries often.

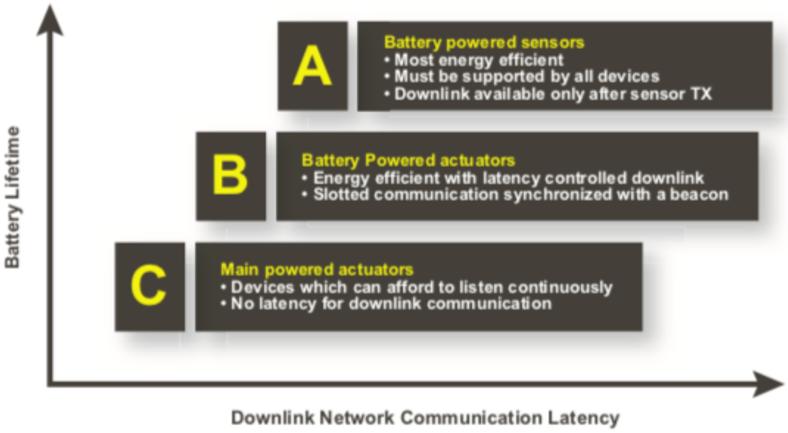


Figure 18: LoRa device classes and power consumption [59].

In this section, we have introduced different wireless technologies. We have discussed technologies such as the short-range technologies, cellular, and LPWANs that offer a new paradigm in IoT connectivity. We have also introduced alternative low cost communications for remote unconnected areas. From the general requirement of the use case, LoRa was identified to fulfill the requirement and alternative networks for backhaul to complement cellular. In the last section we have introduced the basics of LoRa and LoRaWAN.

3.3 Cloud vs Edge Computing

Cloud computing was seen as one of the computing paradigms that could deliver utility computing vision, namely, computing to be commoditized and offered like other utilities such as water, electricity, gas, and telephony [77]. citeauthorvaquero2008break have analysed over 20 definitions of cloud computing and they have proposed the following definition:

”Clouds are a large pool of easily usable and accessible virtualized resources (such as hardware, development platforms and/or services). These resources can be dynamically reconfigured to adjust to a variable load (scale), allowing also for an optimum resource utilization. This pool of resources is typically exploited by a pay- per-use model in which guarantees are offered by the Infrastructure Provider by means of customized (Service-Level Agreements) SLAs" [78].

The National Institute of Standards and Technology (NIST) of the U.S. Department of Commerce has defined cloud computing as "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction " [79].

This ubiquitous and on-demand access to storage and computational resources has made cloud computing gain copious usage in different sectors. In addition, cloud centralized architecture offers effective economies of scale [80]. In the agricultural sector, the usage of cloud computing has grown due to usage of ICT and sensor technologies. This has enabled data to be collected and pushed to the cloud for storage and analysis. Production of big data from farms and storage in cloud give insights to farm operations and facilitate real-time decision making [29]. This also enables the sharing of data between different stakeholders and remote control of farming operations.

Cloud computing has enabled users to obtain computing and storage resources provided by data centres at anytime and from anywhere [81]. Cisco Internet Business Solutions Group predicted that there would be 50 billion devices connected to the Internet by 2020 [39]. The data produced by these devices at the edge of the network pose a challenge to networks and central cloud computing. The increase in number of devices and rapid advancement of Internet technologies comes with its own unique set of challenges such as latency issues for time critical applications, storage of sensitive data at external service providers raises privacy issues and limited bandwidth to transmit large amounts of data produced by the devices [82].

Edge computing has been emerging approach in distributed computing in the last few years. It extends traditional cloud computing to the edge of the network. It is worth noting that fog computing and edge computing are used interchangeably in literature. However, they are some authors that make distinction between these two paradigms. OpenFog consortium defines fog computing as a "system-level horizontal architecture that distributes resources and services of computing, storage, control and networking anywhere along the continuum from the cloud to things. Fog computing is different from edge computing and provides tools for distributing, orchestrating, managing, and securing resources and services across networks and between devices that reside at the edge. Edge architecture places servers, applications, and small clouds at the edge. Fog jointly works with the cloud, while edge is defined by the exclusion of cloud ” [83]. Yousefpour et al. made in-depth comparison of edge and fog computing and other related paradigms. From this, edge is viewed as one of the immediate first hop from IoT devices like WiFi access points or gateways[84].

Edge computing sits at the peak of Gartners Hype Cycle for Cloud Computing, 2018 [85]. Disillusionment and false starts are to be expected before standardization and wide adoption. However, it has the potential to complement and decentralize the current centralized cloud architecture and legacy data centres [86].

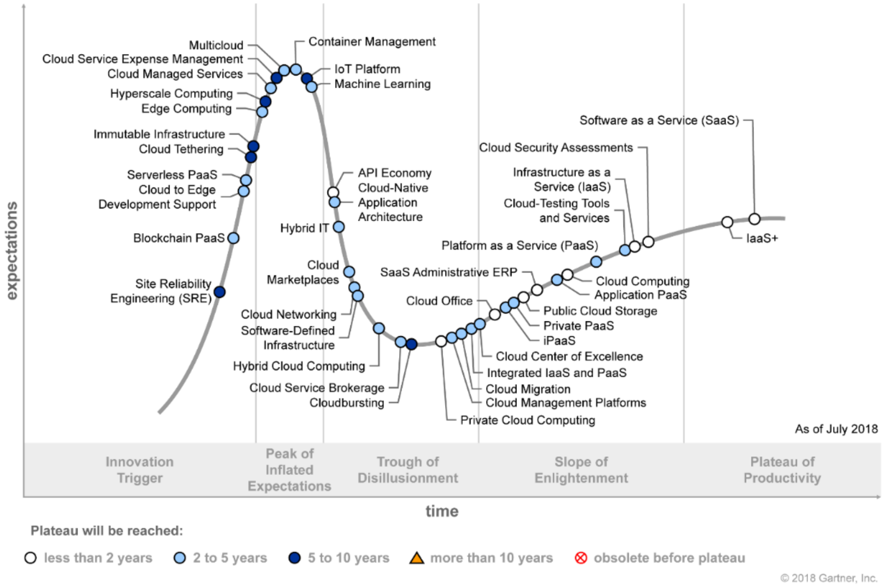


Figure 19: Gartner’s Hype Cycle (Source: [85]).

Edge computing architecture is built on edge servers that offer storage, computing and networking services and enable communication and cooperation between decentralized devices without supervision by a third party [87]. This new paradigm extends the cloud services and has the potential to address the aforementioned challenges related to latency and privacy.

Traditionally IoT applications have stringent requirement of low latency, but this is not the case in smart and precision farming as network performance requirements are less stringent [45]. Furthermore, most areas in emerging economies where small-scale farmers reside is associated with insufficient infrastructure and limited bandwidth. In this context, the benefits of edge computing include: filtering, pre-processing, analysing and aggregation of raw data before forwarding to cloud [88]. This reduces bandwidth usage, local caching for retrieval robustness and reducing the need for communication with cloud [88]. This also saves the user bandwidth if they depend on carriers data plan and also it gives the possibility of users to evaluate which connection and speed they can use at the edge [82]. Analysis can also be done at the edge. Edge analysis is "any data analysis task performed within an edge device (or leaf node) can be identified as edge analytics" [38] e.g. smart plug instead of sending data to cloud every second can analyse data and only send when there is fluctuation in the energy consumed. In addition parameters like sampling frequency and communication frequency can be optimized to reduce bandwidth and storage cost and elongate the lifetime of the device. Knowledge inferring can also be done at the edge by comparing data collected from faulty sensor to the nearby sensor [89]. Storage and analysis at the edge improves privacy as no personal information data is stored in the centralized servers [86]. In agriculture, this is also important as regulations regarding data protection and privacy has not yet been adopted in most emerging economies. Data storage at edge will enable farmers to have control over the data from their farms.

So far we have discussed how edge computing can reduce the cost of communication by reducing the amount of data transmitted to the cloud and reduce power consumption. But one of the fundamental elements that edge offers is putting humans in the control loop giving them control over their system and network links [80]. Such user centric design are important in smart farm as they put humans in loop making them part of the decision making process relating to the farm [90]. Since smart farming is data driven and decisions are based on analysis made on this, socially aware system with humans in control loop and local access to data will encourage adoption of such technologies. Adoption of technological innovations is influenced by farmers perceptions on the effectiveness and accrued benefits [91]. From this, the perception that farmers get from being in control due to benefits offered by computation done at the edge and being in the control loop and decision making could help adoption- same couldn't be said if computation is done at cloud and especially if farmers technological understanding is limited. However, the benefits offered by cloud computing in the general smart farming ecosystem shouldn't be overlooked- as it offers storage and remote access to important data to other stakeholders i.e agricultural extension officers and other experts for analysis and contribute decision making process.

Whereas the advantages of edge computing are many, in this thesis we intend to use edge analysis and storage so as to reduce the cost of transmitting data to cloud and local storage to ease access of data. Also governance policies related to transmission of data to cloud and frequency of transmission can be set depending on the needs of farmers and other stakeholders that consume the data produced at the farms. In addition, simple analysis will be performed on the edge on the data received before notifications are sent to the farms. In this thesis, we therefore harness the benefits offered by edge and cloud solutions to meet the requirements of smart farming in emerging economies. In deed, in most IoT applications one size rarely fits all.

3.4 Related Work

The earlier applications of technology in precision and smart farming focused mostly on automating farm systems based on the data collected by sensors. Zamora-Izquierdo et al. argue that control area in agriculture have developed gradually and significant improvement has been achieved after integration of information and communication system into farm management system [45]. As such, there is a vast amount of literature on greenhouse and hydroponic smart farming and different approaches to monitoring of plants using sensors have been proposed. Cambra et al. have presented a smart system that monitors the state of water that provides nutrient solution to the plants in hydroponic farming [92]. It also presents auto calibrated pH sensors and use of wireless networks to monitor their functioning. Crisnapati et al. presents a hydroponic monitoring and automation system with a responsive web framework [93]. Different wireless technologies are used in depending on the requirements of the agricultural applications scenario. In [94] a wireless control system for Tomato hydroponic farm using the 400 MHz band and IEEE 802.15.6 standard is described. The authors used 400MHz band as it is less affected by plants than 2.4GHz band.

The inclusion of intermediary processing layers (edge or fog) has been recently introduced to smart farming implementations. Caria et al. have proposed a smart farming for animal welfare monitoring with fog layer that enables farmers to locally access the system, manually control parameters and actuators [95]. Ferrández-Pastor et al. have presented edge computing and IoT paradigms in agriculture and they implemented the system in a real hydroponic farm [96]. A more advanced approach with edge computing and virtualization is presented by Zamora-Izquierdo et al. [45]. In their approach edge computing layer is enabled by Network Function Virtualization (NFV) technology so as to increase flexibility in deployment of control modules. Truong have proposed a software component to enable edge analytics on LoRaWAN [97]. The author argues that this is suitable for monitoring of environment and farmers in developing countries where network connectivity and cost are the key constraints and that data is consumed locally reducing the need for pushing data to cloud. Pham, Rahim and Cousin have presented a low cost IoT solution based on LoRa gateway with local storage and access for rural African villages [70]. The solution suggested is part of European Union-Africa project[8] and is applied in monitoring of storage and farming facilities and it targets small and medium scale deployment scenarios in sub-Saharan Africa.

The above solutions show the implementation of different smart farming components such as IoT, edge and cloud computing. A wide range of factors ranging from lack of infrastructure, high cost, limited access to technology to lack of technical know-how hinders the adoption of technologies in agriculture in emerging economies. As such, smart farming solutions for such environments should consider the above factors for the effective use of technology in food production and for the sustainability of the said system. In the next section, we will discuss the solution proposed in this thesis.

---

[8] `https://www.waziup.eu`

4 Implementation

As already mentioned, agriculture is not exempted from the transformation caused by the fourth industrial revolution. The use of IoT and LPWANs in agriculture has already gained momentum in developed countries. In emerging economies, however, mobile-based services are the main services that farmers have adopted. The information farmers get from these services is mostly from other stakeholders. It is therefore important for farmers to get data from their farms to enable them to fully benefit from opportunities that smart solutions brings to farming. Adoption of technology by smallholder farmers can help them improve their productivity, enable efficient use of natural resources and help them adapt to climate change [4].

Considering the requirements of smart farming described in section 2.4 and the need for efficient food production, we propose a smart farming solution that draws upon IoT, edge computing and LPWANs. Recall section 3.2.5, LoRa was considered to conform with the requirements of cost, power consumption and long-range. In addition, it offers the flexibility of deploying networks as it is not operator based and it allows the use of low-cost hardware and open-source software. This is particularly important as it can enable local individuals to design solutions that meet their needs. Indeed as Fox described the availability of open-source software and hardware drives the third generation of DIY, where individuals anywhere can invent solutions to meet their needs, especially in regions with minimum infrastructure [71].

In the proposed solution, the farms have a shared infrastructure. This will mainly entail the gateway and local server. The purpose is to reduce the cost as it will be expensive and technically complicated for individual farmers to deploy the whole network. As such the farmers within the range of the network coverage will share the installation and maintenance cost.

The rest of this section is as follows. Section 4.1 describes the architecture of the system used in this thesis. We will also discuss the various components of the system and its implementation in section 4.2. This section will also explain the experimental set-up used. In section 4.3 we discuss the results and the importance of building capacity of the farmers and their communities.

4.1 System architecture

The smart farming solution proposed here is aimed at helping smallholder farmers in rural areas to better monitor their hydroponics system. LoRaWAN network architecture is generally distributed with centralized cloud-based data aggregation centers that do not promote edge analytics making it unsuitable for developing countries due to the high cost of internet connectivity limiting pushing of data to cloud [97]. The solution proposed here incorporates the edge layer and the system essentially comprises three layers:

- IoT end devices layer,
- LoRa gateway and local server which hosts the edge layer and
- Cloud layer

.

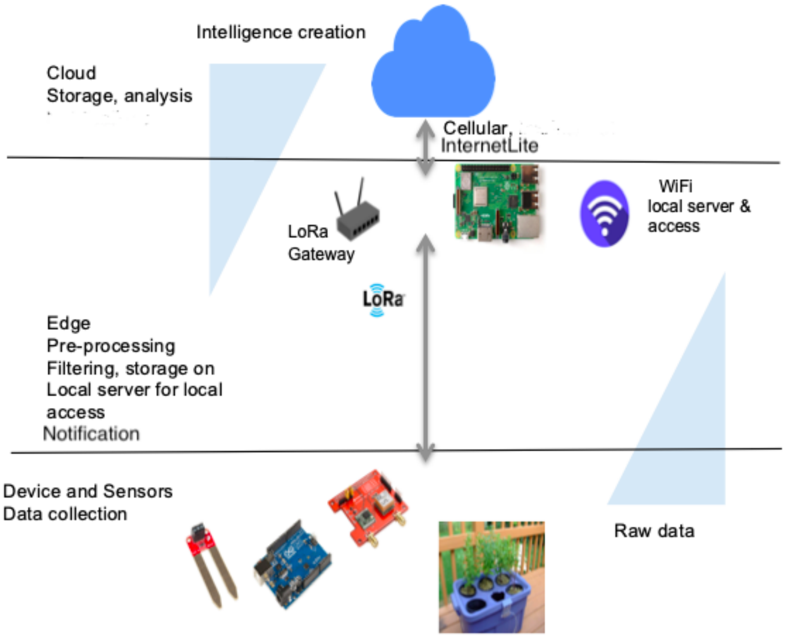


Figure 20: A three-layer architecture

Figure 20 shows the three layers (Three-layer IoT architecture). The IoT end devices are located in the Hydroponic farm in the proposed solution. The edge layer consists of the LoRa gateway and the local server. The local server hosts the LoRaWAN network server for processing of LoRa packets and links end devices to the applications consuming the data. This layer is responsible for the processing of data to reduce the amount of raw data transmitted to the cloud. Typically the amount of raw data collected is big at the end device level while the intelligence created increases in the upper layers as data is processed to get meaningful information [83]. Therefore data processing is done in the local server to give meaningful information to the farmers and send notifications when necessary. The data is also stored at the local server as it is consumed locally and pushing of data to the cloud is not done in real-time to overcome challenges related to the bandwidth usage and cost.

Due to minimal infrastructure and limited information on smart farming, solutions designed for developing countries do not only include the collection of data and integration of information and communication to the farm management system, but it also requires the provision of and access to information on smart farming. To this end we have included WiFi access points or 'information spots' from Basic Internet Foundation' solution as access to information empowers and builds the capacity of local communities. In addition, their backhaul connectivity solution(Internet Lite) is suitable in rural area scenarios as discussed in section 3.2.4. Notification is an important part of smart systems. To this end, we have used EnvayaSMS gateway application that is installed on an Android phone. Figure 21 shows an overview of the whole system.

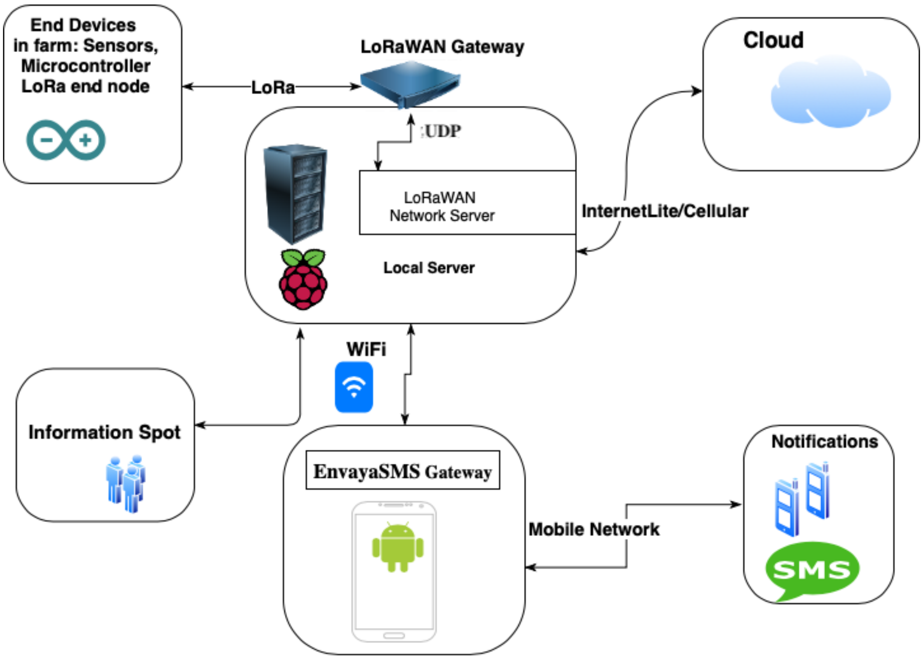
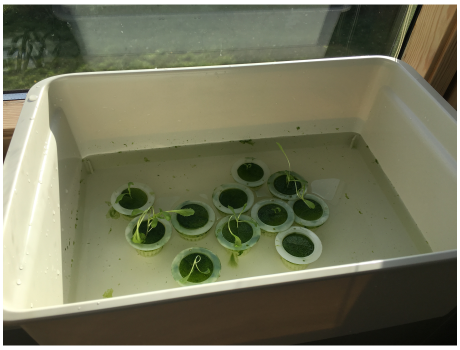


Figure 21: An overview of the system.

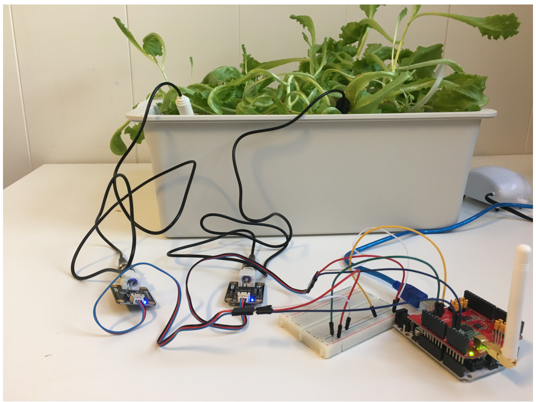
4.2 System implementation

Experiment set-up

We have used a simple hydroponic system and planted lettuce. This is a deep water culture system, and the nutrient does not flow in our set-up. We have included an air pump to provide aeration and prevent the roots from suffocating in the water.



(a) Lettuce



(b) MCU,Dragino shield and sensors

Figure 22: Hydroponic experiment set-up

4.2.1 End devices

The end devices comprise of:

- Sensors: pH and electrical conductivity (EC)
- Microcontroller unit(MCU)
- LoRa end nodes for transmission of data to gateway

4.2.1.1 Sensors

In hydroponic farming, monitoring the nutrient solution is crucial for plant health and necessary for efficient use of resources. pH and EC sensors are used to monitor the nutrients in this thesis. Electrical Conductivity (EC) is measured in siemens and it indicates the amount of dissolved material in a solution. Table 5 shows the details of the sensors.

Table 5: Sensors used in the experiment set-up

Sensor	Manufacturer	Model	Data Interface	Voltage Input
pH sensor	DFRobot	Version 2	Analog	3.0 - 5.0V
EC sensor	DFRobot	Version 2	Analog	3.0 - 5.0V

Sensor calibration

The pH, EC sensors and two standard calibration solutions shown in figure 23 are from DFRobot[9]. The EC sensor was calibrated using Arduino, two standard buffer solutions and manufacturer's software library [10] that uses a two-point calibration method. Two-point calibration is used when readings from the sensor are known to be fairly linear. The buffer solutions have different concentration levels: 12.88ms/cm is used to set the high end and 1413us/cm is used to set the low end of the measurement range. The software library automatically identifies the buffer solutions once the calibration procedure is initiated.

Figure 23: EC, pH probes, and standard buffer calibration solutions

4.2.1.2 Microcontroller Unit

The current trend of IoT end devices development is open source software and low-cost hardware providing a baseline architecture enabling users to develop

---

[9] `https://www.dfrobot.com/product-1123.html`

[10] `https://github.com/DFRobot/DFRobot_EC`

Table 6: Arduino UNO specifications

Arduino	
Specifications	Information
Microcontroller	ATmega328P
Flash Memory	32 KB
SRAM	2KB
EEPROM	1KB
Digital I/O Pins	14
Analog Input Pins	6
Operating Voltage	5V

their custom end devices [98]. However, this raises a compatibility problem as different sensors are developed by different vendors and might not be compatible with some boards. Arduino microcontroller development boards have an inbuilt analog to digital converter making it suitable for sensing analog signals. In addition, it is widely used in education and has a huge online community. There are a variety of sensors that are compatible with it and many well documented open-source programs. As such, we chose Arduino Uno MCU board [11]. The MCU will facilitate data acquisition and implement the LoRaWAN protocol stack as the LoRa chip provides modulation only. Table 6 gives specifications of Arduino Uno

4.2.1.3 LoRa Module

Since the smart farming solution is not automated and not time-critical, data can be sent from the end node hourly or can be configured according to the needs of the farm. From device categories offered by LoRaWAN, device A fits the needs of this system and is thus used in the end nodes. This also suits the power consumption requirements as the transmission is initiated by the end device and done asynchronously. For communication with the gateway, a Dragino shield that is compatible with Arduino is used [12]. It is based on Semtech SX1276 chip. The specifications of this LoRa module is shown in table 7. Initially, we wanted to use the RAK811 LoRa module but we have faced some problems and this is explained in appendix A.

---

[11] `https://store.arduino.cc/arduino-uno-rev3`

[12] `https://www.dragino.com/products/module/item/102-lora-shield.html`

Table 7: Dragino LoRa shield specifications

Dragino LoRa Shield for Arduino	
Specifications	Information
Chip	Semtech SX1276
Frequency	ISM 868(Pre-configured)
Bit rate	Programmable up to 300 kbps
Sensitivity	-148dBm
Compatibility	3.3V or 5.5v Arduino board

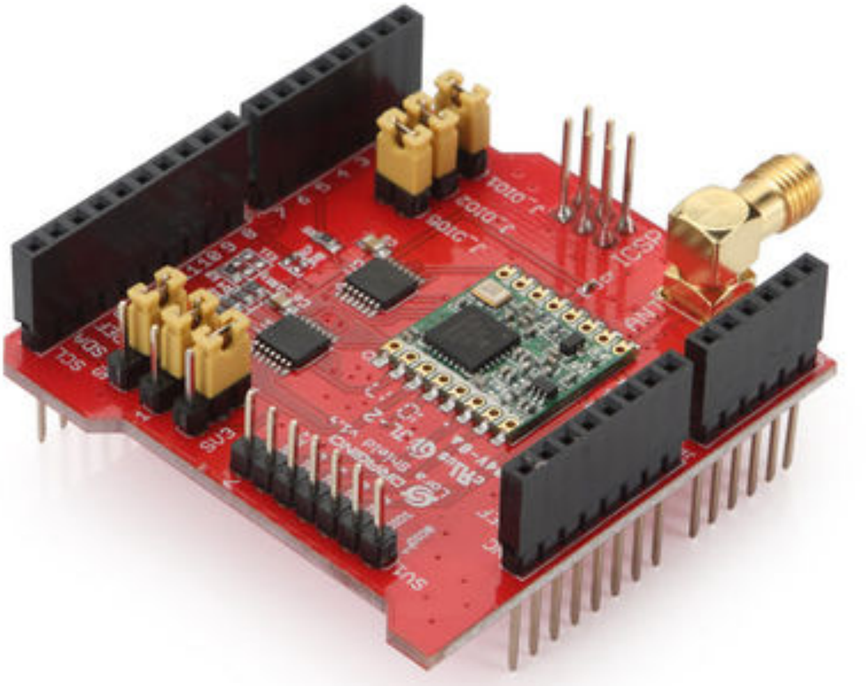


Figure 24: Dragino LoRa shield

Table 8: RAK7249 specifications

	RAK7249
Specifications[99]	Information
RAM	128MB DDR2
Flash	16MB
LoRa chip	SX1301 Mini PCIe card
Channels	8
TX Power:	27dBm (Max)
RX Sensitivity:	-142dBm (Min)
Cellular:	EG95: LTE CAT 4
GPS	L70 GPS Module
Wireless	2.4GHz 802.11b/g/n
Power consumption	12W
Power supply	Power over Ethernet(PoE)

4.2.2 LoRa gateway and Local Server

4.2.2.1 LoRa Gateway

The gateway forms the link between the end devices and the LoRaWAN network server. It receives packets and runs packet forwarder that sends packets to the network server through IP/UDP. The requirement of the proposed system is a gateway that can give a wide coverage and supports connection from many end devices. We used RAK7249 DIY outdoor gateway [13]. RAK7249 is based on SX1301 LoRa chip. This is an enterprise-grade outdoor gateway that comes with LoRa, two LTE and GPS antennas (see figure 25). The cellular connectivity option it offers is suitable where Internet Lite connectivity is not available. RAK7249 offers three configuration options. It can be configured as an integrated system that uses the inbuilt network server, act as LoRa gateway MQTT bridge and communicate with network server through MQTT or use Semtech UDP packet forwarder. This implementation uses an external network server that offers gateway MQTT bridge functionality as such we used Semtech Packet Forwarder. Semtech developed Gateway Message Protocol(GWMS) which is the first gateway protocol for LoRaWAN. This protocol uses the User Datagram Protocol (UDP) and JSON format for the frames transported. In the gateway web interface figure 26, we configured the packet forwarder to communicate with the server as shown in figure 27. Table 8 shows the specifications of the gateway.

---

[13] `https://store.rakwireless.com/products/rak7249-diy-outdoor-gateway`

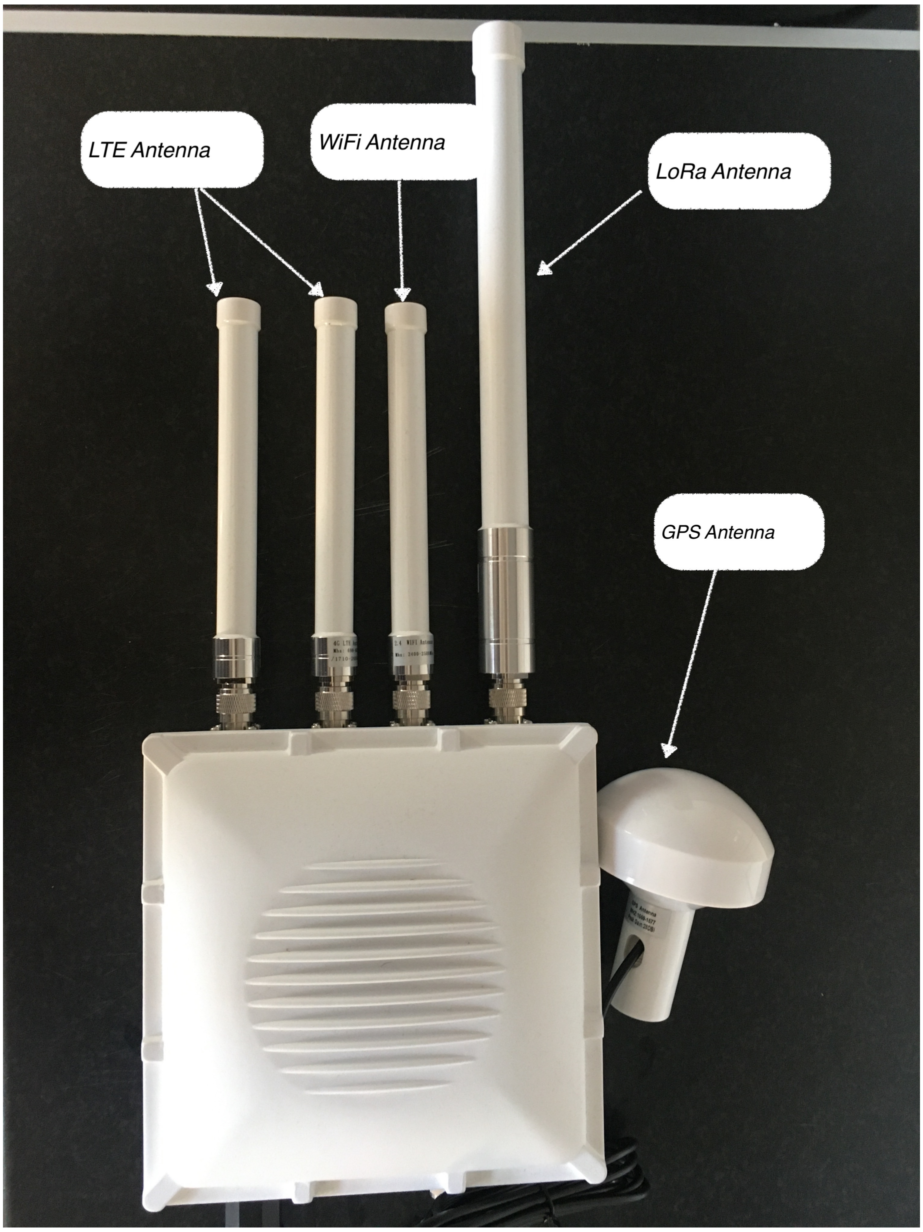


Figure 25: RAK7249 outdoor gateway

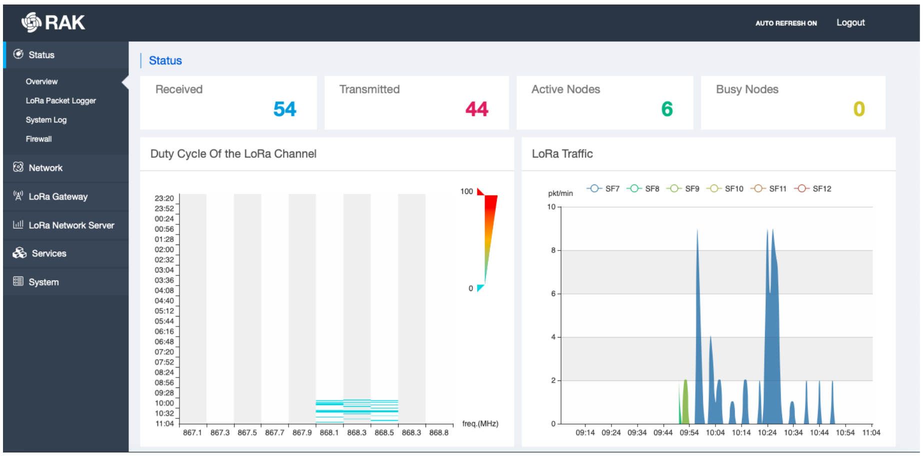
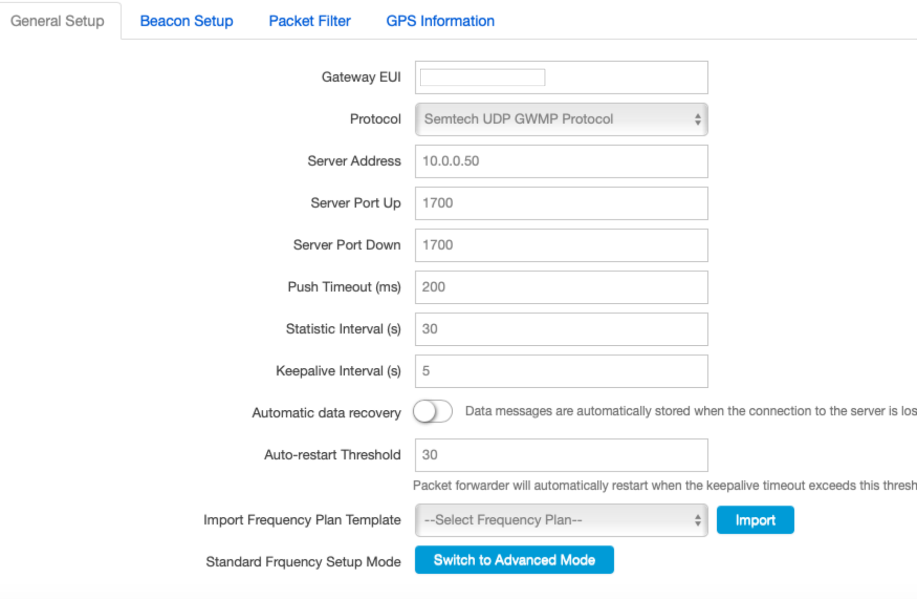


Figure 26: RAK7249 web interface



Gateway EUI:
Protocol: Semtech UDP GWMP Protocol
Server Address: 10.0.0.50
Server Port Up: 1700
Server Port Down: 1700
Push Timeout (ms): 200
Statistic Interval (s): 30
Keepalive Interval (s): 5
Automatic data recovery: ○ Data messages are automatically stored when the connection to the server is los
Auto-restart Threshold: 30
Packet forwarder will automatically restart when the keepalive timeout exceeds this thresh
Import Frequency Plan Template: --Select Frequency Plan--
Standard Frquency Setup Mode

Figure 27: Semtech UDP configuration on the gateway

4.2.2.2 Local server

Raspberry Pi 3+ [14] was used as the local server hosting the LoRaWAN network server. Raspberry Pi is a low cost and powerful single board computer. It has been used in several smart farming approaches e.g [95] in a low cost smart farming for monitoring animal health. In our implementation, the local server

---

[14] https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/

Table 9: Raspberry Pi 3B+ specifications

Specifications	Raspberry Pi Model 3B+ Information
RAM	1GB
CPU	Broadcom BCM2837B0 quad-core, 64-bit @1.4GHz
GPU	GPU: Broadcom Videocore-IV
Ethernet	Gigabit Ethernet
WiFi	2.4GHz and 5GHz 802.11b/g/n/ac Wi-Fi,
Bluetooth	Bluetooth 4.2, Bluetooth Low Energy (BLE)
Storage	MicroSD
Power consumption	5V

hosts the LoRaWAN network server and application server. It also hosts the gateway bridge. These two entities can physically be separated since they perform different functions, but in this case, there are both hosted on a Raspberry Pi. The local server also acts as an edge layer, an intermediary processing layer that performs the storage of sensor data, send notifications to farmer based on data analysis and also pushes data periodically to the cloud where further analysis and storage can be done. Since the suggested solution is integrated into the Basic Internet infrastructure, agriculture-related information and other local content can be stored in the local server and accessed by farmers at the information spots (Wi-Fi access points). Table 9 shows the specifications of Raspberry Pi Model 3B+ [15].

Configuration architectures : In the experimental set-up, the gateway and the Raspberry Pi are in the same local network. The LoRaWAN components used in this thesis are from an open-source LoRaServer project[16] that offers applications that can be implemented flexibly. While a common alternative is The Things Network(TTN), a crowdsourced community network, it does not offer the flexibility needed in the developing world scenario. Because TTN's network server is hosted in Cloud it would be expensive to transmit data. LoRaServer components include LoRa Gateway bridge, LoRa Server and LoRa App Server. All of these three components are installed on the same server. The LoRa Server project offers two main architecture as shown in figure 28. The difference between these two approaches is where the LoRa Gateway bridge is

---

[15] `https://www.raspberrypi.org/magpi/raspberry-pi-3bplus-specs-benchmarks/`

[16] `https://www.loraserver.io`

installed. It can either be installed on the gateway or on a separate server that may or may not host the other components.

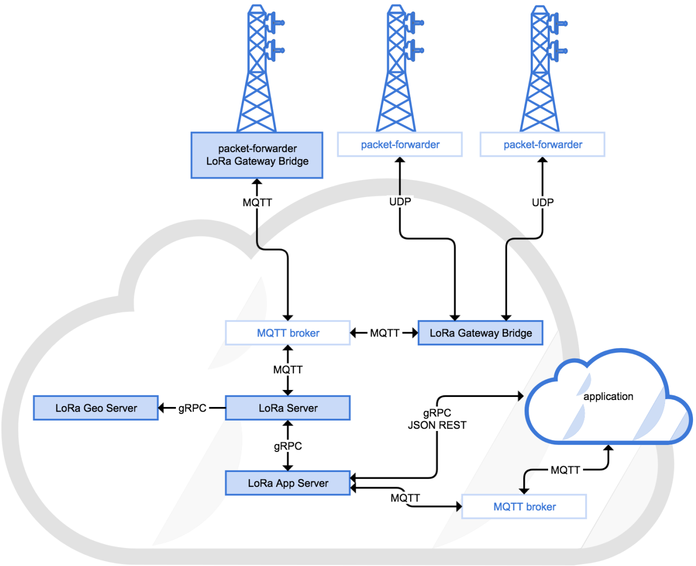


Figure 28: General LoRaWAN configuration architectures.

The configuration used in this thesis is shown in 29. The gateway bridge is installed on the same server together with other LoRaServer components. We chose this configuration because a single gateway bridge handles the conversion of packets from different gateways in case the system is scaled to increase coverage. As mentioned earlier RAK7249 also has an inbuilt LoRa gateway MQTT bridge, but we have not used it because the message format on the gateway is not compatible with the LoRaServer project message format at the time of this writing. RAK7249 uses JavaScript Object Notation (JSON) while LoRaServer uses protocol buffer (protobuf) [17] format.

---

[17] `https://developers.google.com/protocol-buffers`

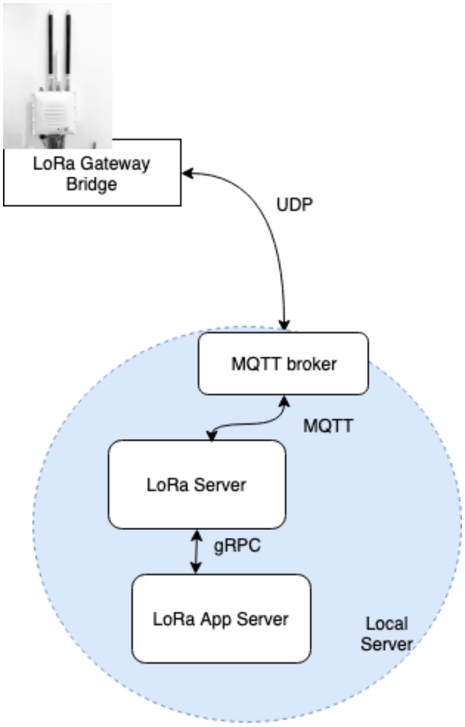
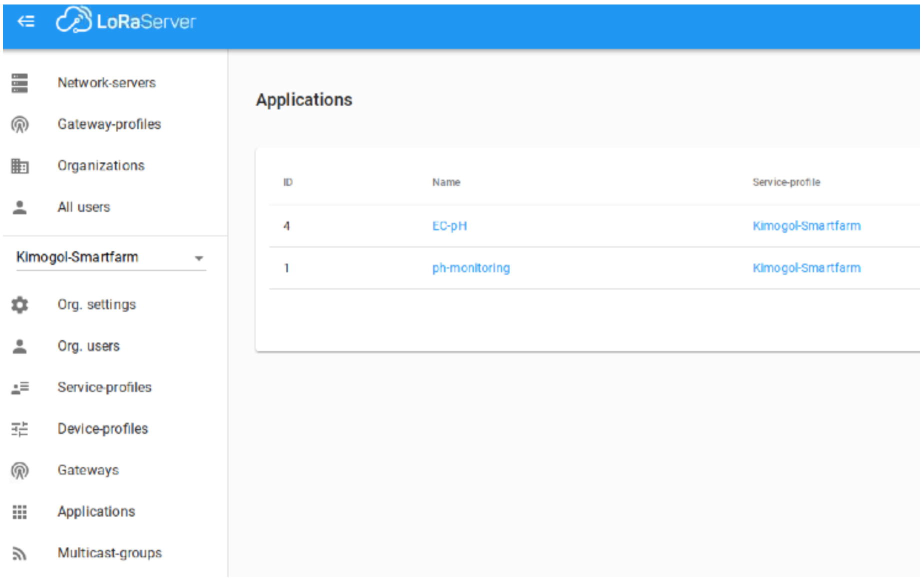


Figure 29: Gateway bridge, LoRa server and LoRa app server are installed in the same server instance.

LoRa Gateway bridge: LoRa gateway bridge abstracts LoRa packets to messages that can be sent over MQTT.

LoRa Server: LoRaWAN network server handles the general state of the network, processing of uplink and scheduling of downlink communication. It is also responsible for de-duplication of packets if the packets are sent from different gateways such that messages are sent to the applications once. It also serves the function of scheduling downlink transmissions.

LoRa App Server: It provides a web interface to enable the management of users and is also an inventory for applications and devices. Live LoRaWAN frames can also be inspected through this interface. It encrypts and decrypts application payloads thus network server can not access them. It also generates application keys and manages the join-request of network and end device activation. Moreover, it provides integration like HTTP, integration with databases e.g. InfluxDB and it also offers MQTT, gRPC[18] and RESTful API for integration with other applications.



Applications

ID	Name	Service-profile
4	EC-pH	Kimogol-Smartfarm
1	ph-monitoring	Kimogol-Smartfarm

Figure 30: LoRa App Server web-interface

LoRa Server components and their dependencies were installed in Raspberry Pi. Because the LoRa gateway bridge uses publish-subscribe communication, Mosquitto[19], a lightweight broker that implements MQTT protocols was also installed. The Semtech UDP packet forwarder that runs on the gateway forwards the data to the LoRa Gateway Bridge which converts the LoRa packets to MQTT and publishes to a topic that LoRa Server subscribes to.

4.2.3 Data collection, transmission, and processing

4.2.3.1 End Device Activation

LoRaWAN offers two methods for activating end nodes: Over the air activation(OTAA) and Activation By Personalization (ABP). Device activation is handled by the LoRa App Server and this can be done through the web-interface-see figure 30. When applications and devices are created, they are assigned 64 bit end-device identifier (DevEUI) and application identifier (AppEUI) (EUI - Extended Unique Identifier) [100]. The devices are dynamically also assigned

---

[18] `https://grpc.io`
[19] `https://mosquitto.org`

Software	
LoRa Server	LoRa Server
	LoRa App Server
	LoRa Gateway bridge
	PostgreSQL - to persist gateway data
Broker	Mosquitto
Node-red	Node-red server
Database	InfluxDb
Visualization	Grafana
Notification	EnvayaSMS server

Table 10: Software

a 32-bit address (DevAddr) and is used to identify the device after it joins the network[100]. LoRaWAN also has further three more security keys namely: network session key (NwkSKey), Application session key (AppSKey) and Application key (AppKey). End devices and the network use NwkSKey to calculate message integrity code (MIC) for data integrity while AppSKey is used to encrypt and decrypt payload[100]. In ABP, DevAddr, NwkSKey and AppSKey are preprogrammed in the end device and also stored in the network thus the device is only attached to a specific network. Therefore, the activation process does not go through the join request and accept procedure. On the other hand OTAA, uses DevEUI, AppEUI and AppKey which must be stored both in the network and the end device for the join procedure [100]. AppKey is used to generate the NwkSKey and AppSKey. The DevAddr is also dynamically assigned in the process. In our case, OTAA was used to connect the end device to the network. To facilitate this, we used Arduino LoRaWAN-MAC-in-C (LMIC) library [20] that was developed by International Business Machines(IBM). Dragino LoRa shield was connected to the Arduino and since the shield is based on Arduino form factor no jumper cables were required for connection. Then DevEUI, AppEUI and AppKey generated in the LoRa App Server were added to sketch. Once the sketch is uploaded to the Arduino, the end device activation process starts automatically.

4.2.3.2 Reading sensor values and transmission

Figure 31 shows the state of the end device. After the activation, the end

---

[20] `https://github.com/matthijskooijman/arduino-lmic`

device starts to transmit data. The EC and pH codes were adapted from the DFRobot product libraries [21]. This code and LMIC code can be found in appendix B. The data mitigation techniques used here is data compression which involves encoding data at end nodes and decoding them at the application server. This technique reduces the size of information transmitted and reduces power consumption thus improving battery life [27]. Because the EC and pH do not change significantly within an hour, values are transmitted to the gateway once every hour. Before the transmission, the sensor readings are encoded as shown in the following code.

```
void do_send(osjob_t* j) {
  struct sensorValues ss = ecread();
  float structec = ss.ec;
  float structph = ss.ph;
  byte payload1[4];
  uint32_t ecValue = structec*100;
  uint32_t phValue = structph*100;
  payload1[0] =highByte(ecValue);
  payload1[1] =lowByte(ecValue);
  payload1[2] =highByte(phValue);
  payload1[3] =lowByte(phValue);

  if (LMIC.opmode & OP_TXRXPEND) {
    Serial.println(F(”OP_TXRXPEND,␣not␣sending”));
  } else {
     LMIC_setTxData2(1, payload1, sizeof(payload1), 0);
    Serial.println(F(”Packet␣queued”));
    Serial.println(LMIC.freq);
  }}
```

---

[21] https://wiki.dfrobot.com/Gravity__Analog_Electrical_Conductivity_Sensor___Meter_V2__K=1__SKU_DFR0300

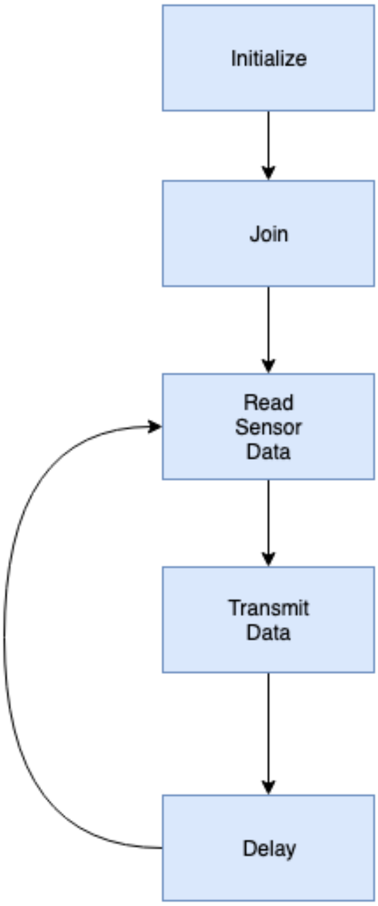


Figure 31: State machine end device

4.2.3.3 Data processing

The flow of the data from the end devices to the applications is depicted in figure 32. The UDP Semtec software running in gateway forwards the data to LoRa-Gateway- bridge. LoRa Gateway bridge publishes messages to a topic which LoRa server subscribes to. LoRa server sends the data to the LoRa App Server through gRPC API. Here the data is decoded and is published to Mosquitto broker for the application using MQTT to access them. In the LoRa App server, we used the following custom JavaScript code to decode the payload.

```
function Decode(fPort, bytes) {
    var ec = (bytes[0] << 8) | bytes[1];
```

```
    var ph = (bytes[2] << 8) | bytes[3];

    var dataout = {
        "sensorvalues": {
            'ec': ec / 100,
            'ph': ph / 100
        },
    };
    return dataout;
}
```

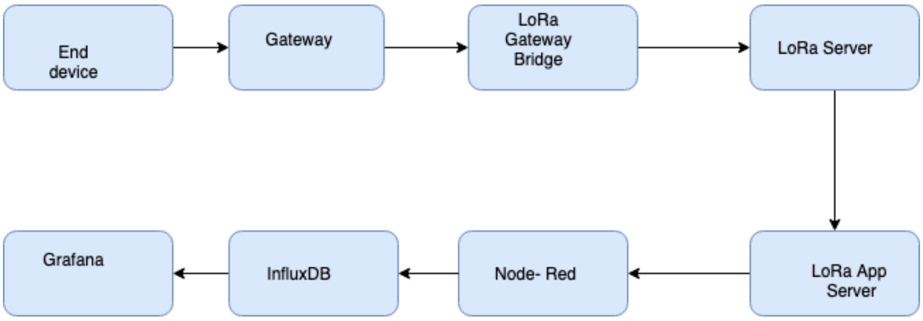


Figure 32: Data flow.

To integrate other functions into our system, we have used Node-Red[22], an open-source web-based programming tool. It is a flow-based program and can easily be used to connect things, applications and process the data they produce. It offers a broad collection of nodes in the palette that can be dragged and dropped into the flow canvas. In Node-red, we use MQTT client which subscribes to topics published by the LoRa App Server. The event in MQTT topics are in this format: **application/[applicationID]/device/[devEUI]/rx**. applicationID is automatically generated and can be found in the LoRa App Server web interface. In this case, the node named EC-PH in figure 33 is the MQTT client and it subscribes to this topic:
application/4/device/8a90dc387df11f42/rx

Following is an example of the data received after subscribing to the above

---

[22] `https://nodered.org`

topic in Node-Red. It shows the end device details, gateway details, Received Signal Strength Indicator(RSSI), SNR and it also indicates that ADR has been activated. It also contains the values of the sensor data.

```
”{”applicationID”:”4”,”applicationName”:”EC-pH”,
”deviceName”:”EC-pH_Hydroponic”,
”devEUI”:”8a90dc387df11f42”,”rxInfo”:
[{”gatewayID”:”XXXXXXXXXX”,”name”:”RAK7249”,
”time”:”2019-10-12T11:49:03.960173Z”,”rssi”:-55,
”loRaSNR”:10,”location”:{”latitude”:60.44765,
”longitude”:12.05757,”altitude”:349}}],
”txInfo”:{”frequency”:868300000,”dr”:5},”adr”:true
,”fCnt”:34,”fPort”:1,”data”:”AE4CmA==”
,”object”:{”sensorvalues”:{”ec”:0.78,”ph”:6.64}}}”
```

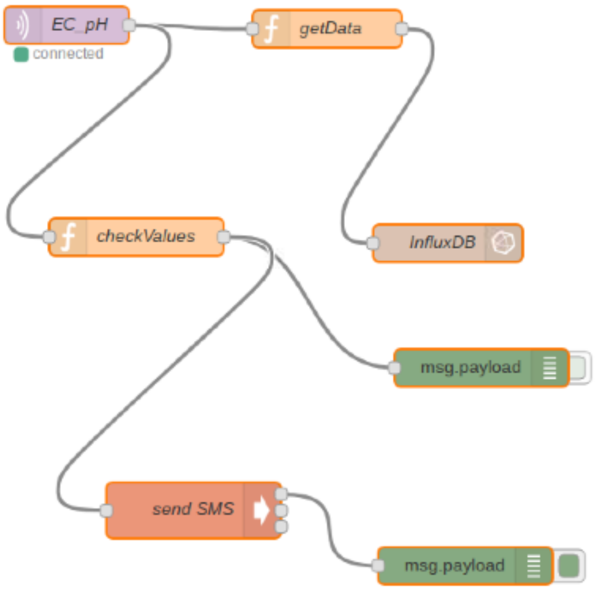


Figure 33: Node-red flow

The data received is in a string format and we used JSON.parse() and extracted the payloads as shown in figure 34. We used InfluxDB[23] to store the

---

[23] `https://www.influxdata.com`

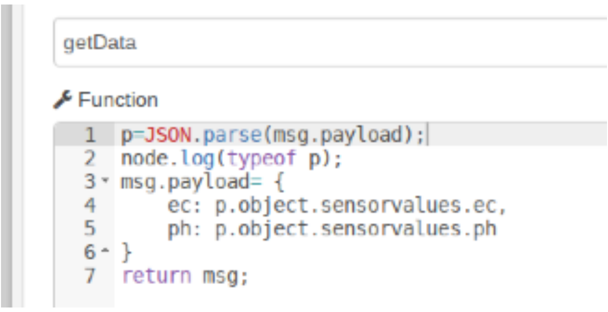
sensor data. InfluxDB is an open-source time series database that enables the storage of sensor data in equally spaced time intervals. This makes it suitable for IoT applications. It suits the needs of smart farming as data is stored with a specific time-stamp making data analysis easy. Furthermore, this enables analysis with a high level of granularity. Farmers or the experts helping farmers can get information on how plants absorb nutrients and they can make informed decisions on when and what amount of nutrient solution to use.

In Node-Red we used a function node, named getData (see figure 33 and 34) for data extraction and posting them to the InfluxDB. Since sensors will be deployed in different sections of the hydroponic farm, knowledge inference can be done using such functions to compare data and identify faulty sensors. We have, however, not implemented this since we only have single EC and pH sensors in the experimental set-up. For data visualization we used Grafana[24] - an open-source tool for visualization which comes with data source plug-in for InfluxDB. We created a dashboard on Grafana and visualized data with graphs to show the pH and EC levels. This can help system administrators to get insights from the data and valuable information which can help farmers make informed decisions. Interactive graphics help better understand underlying data and with time series representation of the database, farm conditions can be compared to crop performance.

Edge Analysis: Edge computation enables optimization of bandwidth usage and performance of data analytic. The edge layer defines the rules related to the storage of data and the sending of notifications. Alerts are sent to the farmers depending on the sensor readings. If the sensor readings fall below a set value then farmers are notified through SMS. For the growth of lettuce, a pH of 5.5 to 6.5 and EC of 0.8 to 1.2 ms/cm is considered suitable for the growth of the plant. We have implemented a function in Node-Red that checks whether sensor data is within the above-defined values (see figure 34). Node-red's exec node executes a command that sends notifications. We will discuss SMS gateway in 4.2.3.4. As such local server also manages the connection between the server and gateway application running on an Android phone.

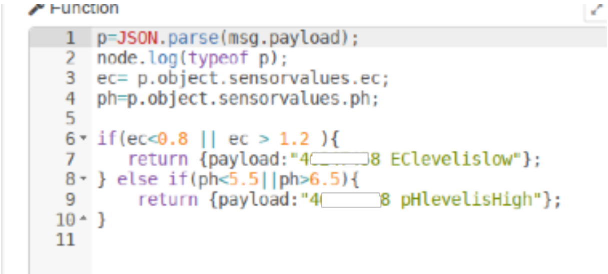
---

[24] `https://grafana.com`



```
p=JSON.parse(msg.payload);
node.log(typeof p);
msg.payload= {
    ec: p.object.sensorvalues.ec,
    ph: p.object.sensorvalues.ph
}
return msg;
```

(a) Get data



```
p=JSON.parse(msg.payload);
node.log(typeof p);
ec= p.object.sensorvalues.ec;
ph=p.object.sensorvalues.ph;

if(ec<0.8 || ec > 1.2 ){
    return {payload:"4[redacted]8 EClevelislow"};
} else if(ph<5.5||ph>6.5){
     return {payload:"4[redacted]8 pHlevelisHigh"};
}

```

(b) Checking pH and EC values

Figure 34: Extracting data and checking values

4.2.3.4 SMS gateway

Notifications are an important part of the IoT system as this informs users of the conditions of the things they are monitoring. The choice of notification system depends on the type of devices used by the clients. In most developing countries basic and feature phones remain the most commonly used devices and uptake of smart devices is influenced by battery life and access to fast internet [3]. Most farmers in rural areas use low-tech phones whose primary communication channels are SMS and voice. Even farmers with smartphones are restricted from using applications due to expensive data plans thus use of data connectivity orientated services is not suitable. In addition, phones have limited processing capabilities and might not support applications. The most suitable way to send notifications, in this case, is SMS. However, setting up a gateway with telecommunication operators and getting shortcodes that are accessible from local numbers is costly. Lightweight SMS gateway application that reside on Android phones like RapidSMS [25] and frontlineSMS [26] have already been used in health sector to send reminders to enhance postnatal care

---

[25] `https://www.rapidsms.org`

[26] `https://www.frontlinesms.com`

appointments [101] and SMS based alert system to monitor pregnancy, maternal and child deaths [102]. In [103], EnvayaSMS [27], an open-source SMS gateway was used to support immunization programs. We can leverage this technology by integrating into smart farming solutions suggested here. Since the phone will be using a local phone number, the cost is reduced as compared to using cloud-based SMS aggregators like Twilio[28]. In this thesis, we are using EnvayaSMS gateway application as it does not require subscription as frontlineSMS. It also offers expansion packs to increase messages sent per hour to 500 from the 100 per hour limit on Android phones. An example of EnvayaSMS configuration with webserver hosted in the cloud is shown in figure 35. However, in our implementation, the server that sends notifications is hosted locally.

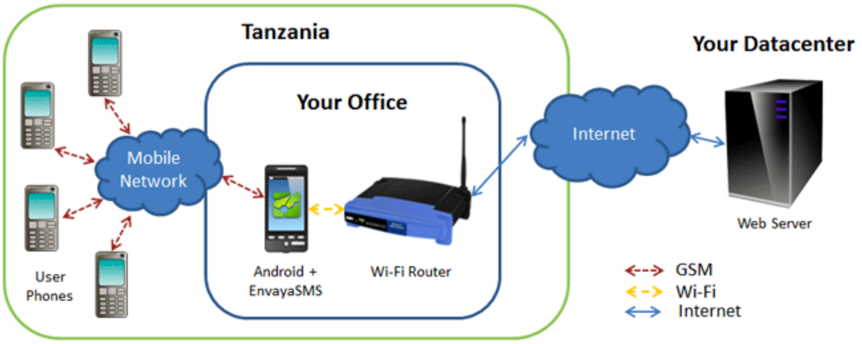


Figure 35: EnvayaSMS configuration

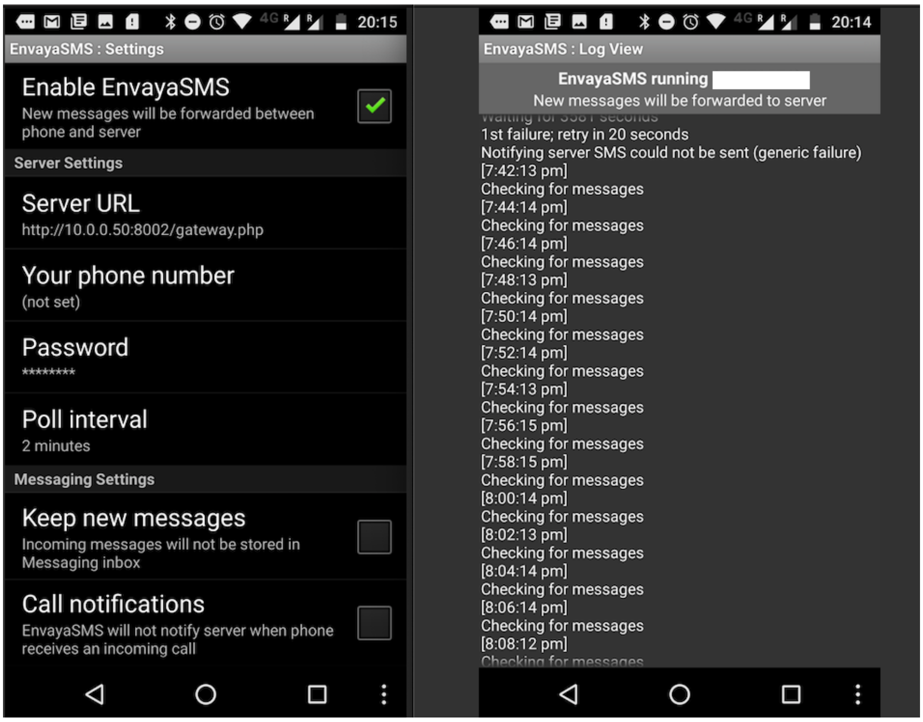
From EnvayaSMS webpage, it is stated that the app can run any Android phone with Android version 1.6 or higher. In this case we used MoTo G Plus [29] phone running on Android 8.1.0. The phone is connected to the same local WiFi as the Raspberry Pi. Scripts from the EnvayaSMS GitHub repository [30] was used to handle the server functions. In this repository, there are three main scripts that are important to mention. These are server.php, gateway.php and send_sms.php. The first scripts is a standalone HTTP server, the second script implements the EnvayaSMS API while the last one enables sending of messages from the command line. The server script was enabled to run at

---

[27] `http://sms.envaya.org`
[28] `https://www.twilio.com`
[29] `https://www.motorola.com/us/products/moto-g-plus`
[30] `http://github.com/youngj/EnvayaSMS`

the Raspberry Pi on start-up. In the application settings, the server Uniform Resource Locator (URL) was set to the path of the script implementing the EnvayaSMS API that is also on the Raspberry Pi as shown in Figure 36. The application was configured to poll for new messages every 2 minutes. Figure 36 shows the application configuration and application polling for messages.

As mentioned earlier SMS alerts are triggered after the sensor values fall or go beyond a certain range. To trigger the sending of messages, we analyzed the sensor data in Node-red, checkValues function as shown in 33. This function analyzes the data and the payload it returns contains the message and phone number of the recipient. The messages can be customized to the local language in this function. Node-red offers an execute node (exec) that can be used to run scripts and programs. This node runs php (send_sms.php) script that queues the message to the local file system. This message is sent to the EnvayaSMS gateway app once it sends a request to the server for outgoing SMS. EnvayaSMS gateway uses an HTTP POST request to poll for outgoing messages and send the status of the sent messages to the server.



Enable EnvayaSMS ☑
New messages will be forwarded between phone and server

Server Settings

Server URL
http://10.0.0.50:8002/gateway.php

Your phone number
(not set)

Password
****

Poll interval
2 minutes

Messaging Settings

Keep new messages ☐
Incoming messages will not be stored in Messaging inbox

Call notifications ☐
EnvayaSMS will not notify server when phone receives an incoming call

EnvayaSMS running
New messages will be forwarded to server

```
1st failure; retry in 20 seconds
Notifying server SMS could not be sent (generic failure)
[7:42:13 pm]
Checking for messages
[7:44:14 pm]
Checking for messages
[7:46:14 pm]
Checking for messages
[7:48:13 pm]
Checking for messages
[7:50:14 pm]
Checking for messages
[7:52:14 pm]
Checking for messages
[7:54:13 pm]
Checking for messages
[7:56:15 pm]
Checking for messages
[7:58:15 pm]
Checking for messages
[8:00:14 pm]
Checking for messages
[8:02:13 pm]
Checking for messages
[8:04:14 pm]
Checking for messages
[8:06:14 pm]
Checking for messages
[8:08:12 pm]
Checking for messages
```

Figure 36: EnvayaSMS configuration and log view.

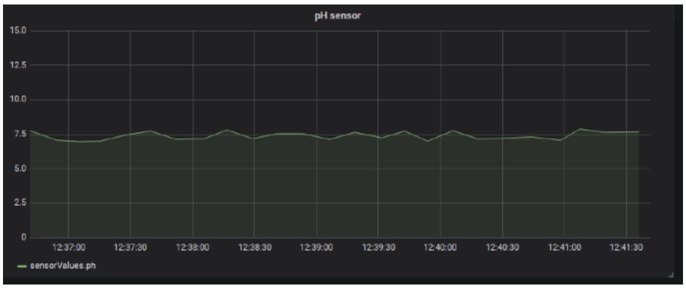
4.2.4 Cloud

As discussed in section 3.3, Cloud offers ubiquitous and on-demand access to computing resources making it suitable for storage and processing of huge data produced by IoT. It allows data consolidation, long term data analysis and effective way to share data with stakeholders and agricultural extension officers. However, lack of or limited internet connectivity and cost is hindering the uptake of this computing paradigm in developing countries. In the proposed solution we have incorporated Basic Internet's InternetLite as a backhaul option to cellular network as depicted in figure 21. Because the InfoInternet standard allows text and pictures, sensor data can be categorized as text and be transmitted for free. With this solution, sensor data can be shared with other stakeholders. Since data is consumed locally and data analysis that trigger alerts are also done locally, the transmission of data to the cloud needs not to be done in real-time. Moreover, smart farming in this scenario is latency tolerant. Consequently, data can be pushed to the cloud at a pre-defined time. Batch transfers to the cloud can also be enabled in the local server subject to data ownership framework guidelines. As much as data sharing is important for smart farming, there is a need for regulated transparency and a framework for sharing of farmers' data with the government (agricultural extension officers) and other stakeholders [31]. In this thesis, the focus is mainly on local data processing and access but data ownership is an area that needs to be considered when this system and other smart farms are deployed in the real world. As shown in 37, shared data from the farm raises privacy and ownership issues that need to considered.

Figure 37: Types of farming data and challenges of shared data (source [4]).

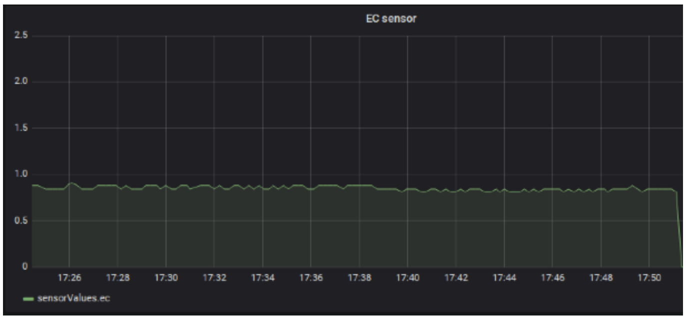
4.3 Results and Discussion

In this thesis, we suggest smart farming based on IoT and LoRaWAN approach to monitor the conditions of the hydroponic system. The sensor data are collected and transmitted periodically to the network server, analyzed and stored in InfluxDB. We use an open-source visualization tool to visualize data simplifying the understanding of the underlying data. Figure 38 shows the EC and pH levels. From the Grafana dashboard, data can be checked as frequently as seconds to a year, making it simple to identify crop performance compared to the quantity of nutrients used. Over time with the help of this data, farmers will identify which plants to grow in the hydroponic system depending on the season to complement traditional farming. Agricultural extension officers can help smallholder farmers with the analysis of data and give advice based on the results.



(a) pH

(b) pH hike



(c) EC

Figure 38: pH, EC visualization on Grafana

We have also observed that the EC probe affects the pH values. Once the EC is inserted into the nutrient solution, the pH values almost doubles, as shown in figure 38b. This has not been investigated further and is left as part of future work.

Using Node-Red, we have analyzed data and sent notifications to an SMS-gateway that is running on Android phone. The SMSgateway can manage to send 500 messages per hour if the expansion packets are used. Since this system is not time-critical, delayed notifications are tolerated.

As mentioned in section 3.2.6, the network server handles the data rate using adaptive data rates. In this implementation, LoRa Server is responsible for the data rate. Adaptive data rate was activated and a spreading factor of 7 use as shown by the graph from RAK7249 web interface in figure 39. This is because the gateway and the end device are just a few meters from each other.

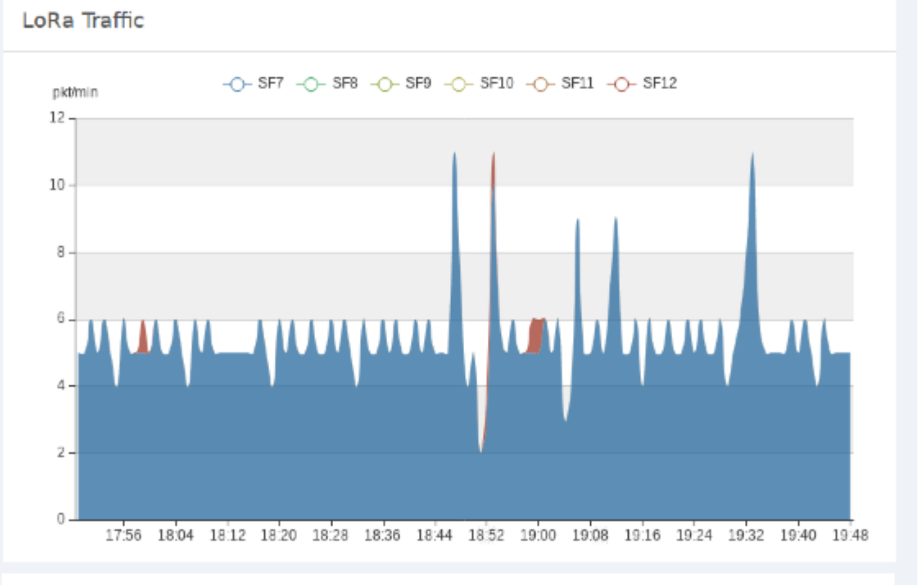


Figure 39: LoRa Traffic per minute and the spreading factor.

4.3.1 From data collection to empowerment

The term empowerment has many definitions and meanings in different contexts. In this section, we focus on how smart farming and access to information empowers local communities. According to Mukherjee, empowerment, in the context of ICT and development, revolves around building capacities of individuals and communities, giving choices and enhancing participation. More important to this thesis is improving the individual's ability to do instrumental roles (a person's everyday task)[104]. In the smart farming solution proposed, the goal is to enable smallholder farmers to do their farming(instrumental role) efficiently and diversify their crop farming (having a choice) by using a smart hydroponic system. A hydroponic system can complement the 'normal' farming (arable) when the weather conditions are unfavourable. By connecting farm to the digital world through IoT, smallholder farmers get another dimension on interacting with their farms. With this technology, information-driven agriculture is encouraged, increase efficient food production and reducing over-dependence on rain-fed agriculture and thus empowering farmers to transition to sustainable farming.

To fully realize the potential of smart farming and make it sustainable, capacity building must be done. To this end, we have included Informations spots from the Basic Internet solution in the proposed approach. Information spots will be a reference point where farmers can access data from their farms and other local content that is relevant to them. In addition, information related to building LoRaWAN network, sensor calibration and maintenance can be stored in local servers and access through WiFi.

Basic Internet's Internet Lite is aimed at bringing connectivity to the rural areas and bridging the digital divide by providing access to information. The intention of inclusion of this approach into the solution proposed in this thesis is to provide information on smart farming and empower local communities by reducing information asymmetry. Since text and pictures are free in the Internet Lite, farmers can access the internet without incurring additional cost and they can also access local content through the WiFi in the information spots. Similarly, farmers can access information related to sustainable farming practices and technologies that facilitate smart farming.

Information spots will also be a point to access value-added services: Here local meteorological departments can share weather information and agricultural departments can provide advisory services and information related to smart farming, disease outbreaks, information on fertilizers and nutrients for hydroponic farming.

5 Evaluation and future work

5.1 Evaluation

In this section, we will evaluate the proposed solution against the requirements of smart farming described in section 2.4.

5.1.1 Low cost devices

We will consider the capital expenditure (CAPEX) and operational expenditure (OPEX). In this cost evaluation, we will only consider the cost related to devices that are part of the proposed solution. As such the cost related to setting up a hydroponic farm has not been analyzed. However, there is a wide range of literature that covers this topic e.g Naik and Singh suggested cheap materials that can be used to set up a greenhouse for hydroponic farm for the cultivation of fodder [36].

To meet the cost requirement, we have proposed a shared infrastructure where the community shares the initial and maintenance cost. The gateway and local server are core components of the shared infrastructure. This is the reason for choosing a gateway that can give wide coverage. The proposed solution uses RAK7249 outdoor gateway with a range of at least 15KM line-of-sight according to the documentation[99]. Thus the farmers within the range can share the cost. The cost of RAK7249 is $599 at the time of this writing. For the local server, we have used Raspberry Pi 3+ which costs $35. The total CAPEX for the shared infrastructure is $634. For each specific hydroponic farm, LoRa node and MCU are needed and these cost $21 and $16 respectively. LoRa is operating on ISM bands that are free, no initial and operational charges are incurred in the use of those frequencies (868MHz in Europe).

The cost of pH and EC sensor was $110. Currently, the common pH and EC probes in the market are mostly hand-held and are not suitable for IoT applications for monitoring smart farms. The cost of these devices varies but these are some of them: Bluelab handy EC-pen[31], cost $109 and ADWA pH-pen[32] which cost $70. The cost of the sensors used in the proposed solution is relatively lower as compared to other alternatives and they are also suitable and

---

[31] `https://www.gartnerbutikken.no/products/bluelab-handy-ec-penn2`

[32] `https://www.gartnerbutikken.no/products/adwa-ph-penn2`

convenient for IoT applications in smart farming. However, they are laboratory grade and can not be immersed in the nutrient solution for long. Atlas Scientific has industry grade EC [33] and pH[34] sensors that cost \$162 and \$40 respectively. These are suitable for smart farming and as they can be submerged to nutrient solutions indefinitely. LoRa technology and IoT are still in their nascent form and with the continuous decrease in the cost of electronics, cheaper sensors and MCU designed for these kinds of applications will be available in the near future.

Even though farmers can share the cost of the shared infrastructure, the cost of this system i.e. sensors in the farms is still beyond the reach of smallholder farmers and alternative approaches are needed. Recall section 2.2.2, a holistic approach for smart farming in developing countries is needed to realize the potential of technology to make food production efficient. It is, therefore, necessary to include other actors e.g. local governments, non-governmental organizations(NGOs), academia and industry to help rural farmers set-up these systems. With significant amounts of aid going into food support especially in sub-Saharan Africa, there is a need to invest such funding into farming systems that leverage recent technological advancement (IoT and LPWAN). Such cooperation with development agents can make smart farming affordable to smallholder farmers. Besides, development agents can cooperate with local and international institutions to develop customized solutions that meet the requirement of the smart farms for resource-constrained settings. This has already been done before and District Health Information System (DHIS)[35] - a health management information system that was developed at the University of Oslo, is an example of academia and other development agents helping in addressing the issues related to health. Similarly, smart farms solutions suggested here can be implemented in the same way. This reduces the financial burden of setting up from the smallholder farmers and at the same time address the problems related to the food crisis in the face of climate change.

---

[33] `https://www.atlas-scientific.com/product_pages/probes/ec_k1-0-mini.html`

[34] `https://www.atlas-scientific.com/product_pages/probes/c-ph-probe.html`

[35] `https://www.dhis2.org`

5.1.2 Power consumption

Most of the rural areas in developing countries are not connected to the power grid. All components used in this thesis can be powered by a battery. The gateway can operate on 12V/10AH batteries according to the documentation[99]. The MCU, LoRa node and sensors can be powered by rechargeable batteries. Also, solar power which offers a cheap and renewable source of energy can be used to power the system. However, solar panels will increase initial cost, but a worthy investment in the long term thus recurring costs will be reduced as batteries won't be changed often. The end devices, Dragino shield, is configured as class A devices that are the most energy-efficient as they initiate transmission. ADR has been initiated which streamlines transmission power, payload length, and SF. To further reduce energy consumption, transmissions have been set to once every hour as conditions of the farm does not change rapidly.

5.1.3 Cost-efficient communication

As mentioned earlier, the communication between sensors and gateway is through LoRa technology that uses ISM bands and no cost is incurred in using those frequencies. Realtime transfer of sensor data to the cloud is costly due to bandwidth usage or might not be possible due to a lack of connectivity. The proposed solution has a local server that offers local storage and computation to facilitate edge computation. Further, compression and batch transfers can be done at the edge layer to reduce bandwidth consumption. The Wi-Fi available in the information spots will help farmers access the information related to their farms e.g Grafana dashboards to get an insight into their farms. Since content is hosted locally, farmers will not incur additional costs compared to the cloud-based system.

Notifications are sent from an Android-based SMS gateway application that uses a local number. EnvayaSMS is compatible with old versions of Android OS (Version 1.6) and can be installed on cheap and widely available Android phones. This reduces the cost of sending SMS notifications as local sim cards are used thus local SMS rates applied. This is cheaper compared to other solutions like Twilio that require a monthly subscription and limitation on the number of messages sent per data. There are other SMS gateway apps in the market e.g FrontlineSMS and Telerivet [36] - but they also require a subscription. Using EnvayaSMS is cheaper and satisfies the requirements of the proposed solution.

5.1.4 Software

One limiting factor to the adoption of technology in developing countries is the lack of access to new technology and ownership rights of technology. Fortunately, there has been an increasing focus on the need to democratize technology and knowledge and open source software has revolutionized this. In the implementation of the proposed solution, we have used open-source software, open and widely used standards. This software has a large community and free of licenses that restrict its usage. Considering the computing capacities of the end node and local server (Raspberry Pi 3+), we have used lightweight protocols e.g MQTT that are efficient in bandwidth and power consumption. For network server, which is an important part of the LoRaWAN, we used components of the LoRaServer Project. For storage and visualization, InfluxDB and Grafana were used. These are also open-source software tools. Node-red which is also open source gives the platform that is suitable for IoT implementations and easy to program system functionalities. As mentioned in the previous sections, EnvayaSMS gateway was used to send notifications to farmers.

To build the capacity of the local community, platforms that contain free information on farming can be hosted in the local server and accessed from the information spots. There are platforms that provide free information for different sectors e.g in education Khan Academy, in health (yeboo.com - part of Basic Internet services) offers free teaching videos for health workers in a low resource setting. Similarly, content that is related to farming can be hosted in the local server and accessed at the information spots.

5.1.5 Computation and storage

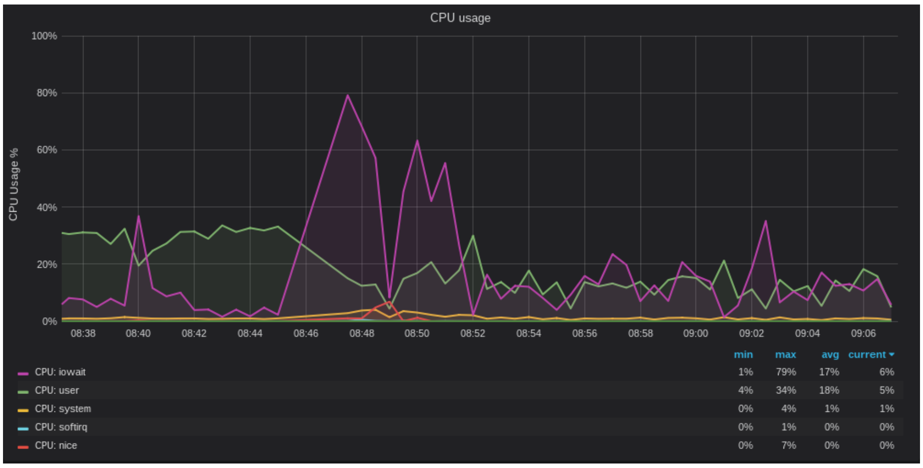
In the proposed solution all computation is done at the local server. In our implementation we used a Raspberry Pi 3+. Figure 40 shows percentage of CPU used by CPU: iowait (time system waits for input and output processing), user, system (kernel usage), softirq (time used on interrupts processing) and

---

[36] `https://telerivet.com`

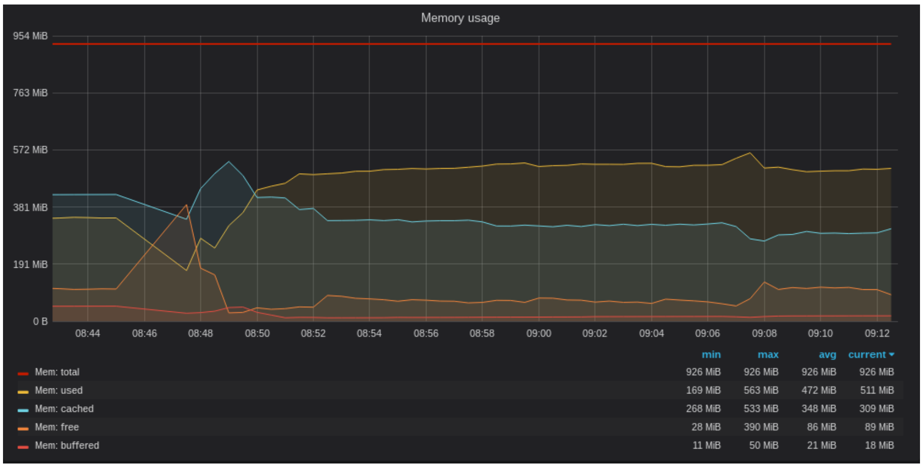
nice (CPU utilization on user level) and 41 shows memory usage (total memory, cached, free and buffered). The components of the LoRaServer project (LoRa-Gateway Bridge, LoRaServer, LoRa-App-Server) are in the local server. Data analysis is also done locally and activation of notification is also processed in the local server. EnvayaSMS server is also hosted on the same machine. This reduces the cost of sending data to the cloud for processing. We have, however, suggested the inclusion of the cloud layer in the system for long-term storage, analysis and sharing information with other stakeholders. To minimize the cost, Basic Internet Foundation's Internet Lite is adopted into the solution to offer internet connectivity.

We used a 16 GB SD card for the experiment but the storage can be expanded if there is a need. In the current set up the sensors transmit data every hour. This is 4 bytes per hour, 96 bytes per day, 2880 bytes per month and 34 560 bytes per year. The current set-up can easily handle this amount of data.



	min	max	avg	current ▾
CPU: iowait	1%	79%	17%	6%
CPU: user	4%	34%	18%	5%
CPU: system	0%	4%	1%	1%
CPU: softirq	0%	1%	0%	0%
CPU: nice	0%	7%	0%	0%

Figure 40: CPU usage.



	min	max	avg	current ▾
Mem: total	926 MiB	926 MiB	926 MiB	926 MiB
Mem: used	169 MiB	563 MiB	472 MiB	511 MiB
Mem: cached	268 MiB	533 MiB	348 MiB	309 MiB
Mem: free	28 MiB	390 MiB	86 MiB	89 MiB
Mem: buffered	11 MiB	50 MiB	21 MiB	18 MiB

Figure 41: Memory usage.

5.1.6 Scalability

The Network capacity of the LoRaWAN network is determined by the frequency of data transmission, data rate, the number of channels in the gateway and duty cycle as discussed in section [59]. The regulation on the duty cycle depends on the region. In Europe, the duty cycle for the 868 ISM band is 1 %. This equates to 36 sec/hour transmission per end device [37]. In the experimental set-up, the data is transmitted once every hour which is below 36 sec/hour thus complying with duty cycle requirements. RAK7249 gateway has 8 channels and ADR has been enabled to optimize the performance and capacity of the network. Also, it had a 15KM line-of-sight offering sufficient coverage in rural areas.

The scale of data produced (amount of data: number of sensors, frequency of transmission per hydroponic farm) is small as compared to big data in other IoT sectors, but in countries with limited internet and high cost, even small data transmission is costly. The proposed solution mainly focuses on storing data at the local servers. However, the use of InternetLite for connectivity with cloud enables long term storage and sharing of data with other stakeholders.

5.1.7 Ease of Use and sustainability

The ease of use affects the dynamics of adoption and scaling up of the new technologies in any environment. It also affects the perceived benefits of smart

---

[37] (Number of seconds in a day 86400 *1/100 = 864 seconds per day)

farming. Farmers are generally not early adopters of technology and for farmers in developing countries where knowledge of even existing technology is limited, new technologies should be easy to use so to encourage their adoption. In the proposed solution, the farmers share the infrastructure. Farmers only need to install end devices (sensors, MCU and LoRa transceiver) in their hydroponic farms while the gateway and the network server are hosted in a single place.

The use of smartphones is increasing gradually, but basic and feature phones remain the dominant mobile devices used in developing countries. As discussed in section 4.2.3.4 the high cost of mobile and battery life affects the uptake of smart devices in emerging economies. Application-based notifications require data-based connection to the internet making it not suitable for smallholder farmers who can not afford to have daily data bundle subscriptions. Based on this, the suitable notification service is SMS and in text format. Text messages are a format that can easily be understood and is a service that is available even in feature phones. EnvayaSMS offers a cost-efficient SMS notification compared to app-based notifications or other SMS aggregators like Twilio.

It is important to involve farmers in the control loop and in the decision-making process. The suggested solution gives the farmers the necessary information for them to make changes related to their hydroponic system. For farmers, especially in rural areas where the use of technology is not common, the sense of being in control of the system gives them the confidence to use and sustain the system. Since the notification system is running locally, messages are written in the local language for local farmers to understand. This cost-efficient approach makes this system sustainable.

The inclusion of technology can also encourage the youth into farming. With a smart hydroponic farm, youth can be motivated to do farming and this creates job opportunities for them. In most emerging economies, youths migrate to cities and the practice of agriculture has been left to the older generation. With the digitalization of farming, like the system proposed here, the profile of farming is improved increasing the chances of youth adopting this as a source of employment. The ease of use evaluated here is not only on how simple this system is for farmers to use, but also this system simplifies and modernizes farming for the younger generation to practice it.

In table 11 we have given a summary of the evaluation. The following symbol legends are used in evaluation table 11.

Better: ++, Good:+,

Reasonable: o,

Bad: $-$, Worse $- -$,

Not applicable: x

As mentioned in device cost evaluation, the cost of sensors for this use case is high (evaluated as $- -$ in the table) while the initial cost for gateway and local server are shared making it reasonable (o). LoRa is an efficient IoT connectivity (++) while InternetLite is suitable (++) for backhaul connectivity compared to cellular due to cost. Cloud offers computation, storage and is scalable thus performs better than the solution suggested here as such cloud is evaluated as better suited for data storage and computation (+ +). Also the gateway can scale with increased number of end devices and users (+ +).

Table 11: Evaluation of the proposed solution

Evaluation Criteria	Specification	Smart farming		
		Device layer MCU, LoRA sensor	Edge layer Gateway, Raspberry Pi	Cloud layer
Cost	Capex: sensors	− −	x	x
	Capex: MCU/Transceiver	++	++	x
	Capex: Gateway/Pi	x	o	x
	Opex	o	++	x
Power consumption	Efficiency	++	++	x
Communication	IoT connectivity LoRa	++	++	x
	Backhaul Cellular	x	x	+
	Backhaul InternetLite	x	x	++
Software	Open source	++	++	x
Computation and storage	Computation	+	+	++
	Storage (sensor data)	x	+	++
Scalability	Increase number of end devices	x	++	x
	Increased volume of sensor data	x	+	++
	Increased number of users	x	++	x
	Coverage area	x	++	++
Ease of Use	Installation	+	++	x
	Maintenance	+	++	x

5.2 Future Works

The following are some of the areas future works can pursue. In future research, the implementation should be done in a real rural setting and the performance of the battery-powered end devices and the overall performance of the system evaluated. As mentioned earlier the EC sensor interferes with pH sensor readings and this is something that can be investigated further.

Future works can also look into integrating LoRa based message system suggested by Pinto et al. for notifications into the smart farm solution proposed in this thesis [75] .

Another pertinent issue future works can examine is the end-user involvement in the development of smart farming. A living lab is an approach that can be used. This approach allows the involvement of farmers and other stakeholders in the process and their needs are taken into consideration. Together with other stakeholders and local communities crops that give high returns and can do well in hydroponic farming can be identified.

The sharing of farm data with other stakeholders is important. However, there are no well-defined regulations on ownership of data in most emerging economies. How this data is used and how it can best benefit smallholders is an interesting topic future works can examine.

6 Conclusion

Agriculture and climate change are interlinked. Agricultural activities contributes to greenhouses emission and change in weather patterns affects food production. However, food production has to increase because of rapid population growth. This growth in population is projected to occur mostly in developing countries. Therefore, there is a need to facilitate smallholder farmers in these countries to efficiently produce sufficient food.

The use of IoT in agriculture will increase over the coming years even in developing countries, albeit slowly compared to their developed counterparts. Currently, mobile phone-based services are the most common service available to smallholder farmers. There is a need to leverage advanced technologies and use low-cost hard and open source software to monitor and enhance a conservative use of resources to make food production efficient.

In this thesis, we have presented a smart hydroponic farming solution for smallholder farmers in emerging economies. The suggested solution takes a holistic view of smart farming that involves the use of technology to monitor farms, diversify farming practices and reduce the overreliance of rain-fed agriculture. Further, the approach gives another dimension for interaction between farmers, agricultural extension officers, and other stakeholders by the use of sensor data to make informed decisions. We based the proposed solution on IoT and LoRaWAN including the edge layer that performs data processing and sends notifications. We have suggested a shared infrastructure where farmers share the cost of the gateway and local server. The proposed solution draws upon availability of open-source software and low-cost hardware for MCU and the LoRa transceivers. However, the cost of sensors for the hydroponic system is still high and we have suggested cooperation with local governments and development agents to reduce the financial burden from farmers.

One of the bottlenecks to the adoption of technology is the lack of Internet connectivity. To this end, we have adopted Basic Internet's Internet Lite for connectivity between edge and cloud layer. Further, we have integrated information spots for access farm data and other value-added services like weather information and advisory information. Access to such information will build capacity and empower the local communities.

References

[1] FAO. *Agriculture and climate change – Challenges and opportunities at the global and local Level – Collaboration on Climate-Smart Agriculture. Rome.* 2019. URL: `http://www.fao.org/3/CA3204EN/ca3204en.pdf` (visited on 17/10/2019).

[2] FAO. *How to Feed the World in 2050.* URL: `http://www.fao.org/fileadmin/templates/wsfs/docs/expert_paper/How_to_Feed_the_World_in_2050.pdf` (visited on 11/02/2019).

[3] Heike Baumüller. "Towards smart farming? Mobile technology trends and their potential for developing country agriculture". In: *Handbook on ICT in Developing Countries* (2017).

[4] Ajit Maru et al. "Digital and data-driven agriculture: Harnessing the power of data for smallholders". In: Global Forum on Agricultural Research and Innovation. 2018.

[5] A Min Tjoa and Simon Tjoa. "The role of ICT to achieve the UN Sustainable Development Goals (SDG)". In: *IFIP World Information Technology Forum.* Springer. 2016, pp. 3–13.

[6] W Al, GR ORKING and OUPON CLIMA. *Climate change and food security: a framework document.* 2008.

[7] World Economic Forum. *The Global Risks Report 2016 11th Edition.* URL: `http://www3.weforum.org/docs/GRR/WEF_GRR16.pdf` (visited on 16/03/2019).

[8] Population Division United Nations D. o. E. a. SA. "World Population Prospects: The 2017 Revision, Key Findings and Advance Tables"". In: (2017).

[9] World Health Organization et al. *The State of Food Security and Nutrition in the World 2018: Building climate resilience for food security and nutrition.* Food & Agriculture Org., 2018.

[10] Mary Nyasimi et al. *Evidence of impact: climate-smart agriculture in Africa.* 2014.

[11] Harald Sundmaeker et al. “Internet of food and farm 2020”. In: *Digitising the Industry-Internet of Things connecting physical, digital and virtual worlds. Ed: Vermesan, O., & Friess, P* (2016), pp. 129–151.

[12] John Beddington. “Food security: contributions from science to a new and greener revolution”. In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 365.1537 (2010), pp. 61–71.

[13] World Bank. *World development report 2008: Agriculture for development.* World Bank, 2007.

[14] Muhammad Ayaz et al. “Internet-of-Things (IoT)-Based Smart Agriculture: Toward Making the Fields Talk”. In: *IEEE Access* 7 (2019), pp. 129551–129583.

[15] Emanuele Pierpaoli et al. “Drivers of precision agriculture technologies adoption: a literature review”. In: *Procedia Technology* 8 (2013), pp. 61–69.

[16] B Melesse. “A Review on Factors Affecting Adoption of Agricultural New Technologies in Ethiopia”. In: *Journal of Agricultural Science and Food Research* (2018).

[17] Uwe Deichmann, Aparajita Goyal and Deepak Mishra. *Will digital technologies transform agriculture in developing countries?* The World Bank, 2016.

[18] Food and Agriculture Organization of the United Nations. *2010b. Climate-Smart agriculture: policies, practice and financing for food security, adaptation and migration.* 2010.

[19] FAO. *The future of food and agriculture–Trends and challenges.* 2017.

[20] Brahima Sanou. “ICT facts and figures 2016”. In: *International Telecommunication Union* (2016).

[21] Pape Abdoulaye Barro et al. “A Smart Cities LoRaWAN Network Based on Autonomous Base Stations (BS) for Some Countries with Limited Internet Access”. In: *Future Internet* 11.4 (2019), p. 93.

[22] Beecham Research. *Towards smart farming Agriculture Embracing The IoT Vision.* URL: `http://www.beechamresearch.com/files/BRL%20Smart%20Farming%20Executive%20Summary.pdf` (visited on 16/03/2019).

[23] Remco Schrijver, K Poppe and C Daheim. "Precision agriculture and the future of farming in Europe: Scientific Foresight Study". In: *European Parliament Research Service: Brussels, Belgium* (2016).

[24] Roberto Fresco and Gianluigi Ferrari. "Enhancing precision agriculture by internet of things and cyber physical systems". In: *Atti Soc. Tosc. Sci. Nat. Mem. Supplemento* 125 (2018), pp. 53–60.

[25] Claudia Bahr et al. *EIP-AGRI Focus Group: Precision farming.* 2015.

[26] Sjaak Wolfert, Daan Goense and Claus Aage Grøn Sørensen. "A future internet collaboration platform for safe and healthy food from farm to fork". In: *2014 Annual SRII Global Conference.* IEEE. 2014, pp. 266–273.

[27] Haider Jawad et al. "Energy-efficient wireless sensor networks for precision agriculture: A review". In: *Sensors* 17.8 (2017), p. 1781.

[28] Dieisson Pivoto et al. "Scientific development of smart farming technologies and their application in Brazil". In: *Information processing in agriculture* 5.1 (2018), pp. 21–32.

[29] Sjaak Wolfert et al. "Big data in smart farming–a review". In: *Agricultural Systems* 153 (2017), pp. 69–80.

[30] Partha Pratim Ray. "Internet of things for smart agriculture: Technologies, practices and future direction". In: *Journal of Ambient Intelligence and Smart Environments* 9.4 (2017), pp. 395–420.

[31] Achim Walter et al. "Opinion: Smart farming is key to developing sustainable agriculture". In: *Proceedings of the National Academy of Sciences* 114.24 (2017), pp. 6148–6150.

[32] Howard M Resh. *Hydroponic food production: a definitive guidebook for the advanced home gardener and the commercial hydroponic grower.* CRC Press, 2016.

[33] Edoardo Pantanella. "Aquaponics Production, Practices and Opportunities". In: *Sustainable Aquaculture*. Springer, 2018, pp. 191–248.

[34] Merle H Jensen. "Hydroponics". In: *HortScience* 32.6 (1997), pp. 1018–1021.

[35] Arjina Shrestha, Bruce Dunn et al. *Hydroponics*. 2010.

[36] PK Naik and NP Singh. "Hydroponics fodder production: an alternative technology for sustainable livestock production against impeding climate change". In: *Model Training Course on Management Strategies for Sustainable Livestock Production against Impending Climate Change* (2013), pp. 70–75.

[37] Pallavi Sethi and Smruti R Sarangi. "Internet of things: architectures, protocols, and applications". In: *Journal of Electrical and Computer Engineering* 2017 (2017).

[38] Charith Perera et al. "Fog computing for sustainable smart cities: A survey". In: *ACM Computing Surveys (CSUR)* 50.3 (2017), p. 32.

[39] Dave Evans. "The internet of things: How the next evolution of the internet is changing everything". In: *CISCO white paper* 1.2011 (2011), pp. 1–11.

[40] James Manyika. *The Internet of Things: Mapping the value beyond the hype*. McKinsey Global Institute, 2015.

[41] Jayavardhana Gubbi et al. "Internet of Things (IoT): A vision, architectural elements, and future directions". In: *Future generation computer systems* 29.7 (2013), pp. 1645–1660.

[42] Ian G Smith. *The Internet of things 2012: new horizons*. CASAGRAS2, 2012.

[43] Parvaneh Asghari, Amir Masoud Rahmani and Hamid Haj Seyyed Javadi. "Internet of Things applications: A systematic review". In: *Computer Networks* 148 (2019), pp. 241–261.

[44] Hua Li et al. "Development of a remote monitoring system for henhouse environment based on IoT technology". In: *Future Internet* 7.3 (2015), pp. 329–341.

[45] Miguel A Zamora-Izquierdo et al. "Smart farming IoT platform based on edge and cloud computing". In: *Biosystems Engineering* 177 (2019), pp. 4–17.

[46] Ray Mulenga et al. "Applying Internet of Things in Monitoring and Control of an Irrigation System for Sustainable Agriculture for Small-Scale Farmers in Rural Communities". In: *2018 IEEE PES/IAS PowerAfrica*. IEEE. 2018, pp. 1–9.

[47] M Safdar Munir et al. "Design and Implementation of an IoT System for Smart Energy Consumption and Smart Irrigation in Tunnel Farming". In: *Energies* 11.12 (2018), p. 3427.

[48] Muhammad Hunain Memon et al. "Internet of Things (IoT) enabled smart animal farm". In: *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*. IEEE. 2016, pp. 2067–2072.

[49] Dimitrios G Kogias et al. "Realizing the Wireless Technology in Internet of Things (IoT)". In: *Emerging Wireless Communication and Network Technologies*. Springer, 2018, pp. 173–192.

[50] Vasileios Karagiannis et al. "A survey on application layer protocols for the internet of things". In: *Transaction on IoT and Cloud computing* 3.1 (2015), pp. 11–17.

[51] Nitin Naik. "Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP". In: *2017 IEEE international systems engineering symposium (ISSE)*. IEEE. 2017, pp. 1–7.

[52] Dinesh Thangavel et al. "Performance evaluation of MQTT and CoAP via a common middleware". In: *2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*. IEEE. 2014, pp. 1–6.

[53] Marco Centenaro et al. "Long-range communications in unlicensed bands: The rising stars in the IoT and smart city scenarios". In: *IEEE Wireless Communications* 23.5 (2016), pp. 60–67.

[54] Gil Reiter. "Wireless connectivity for the Internet of Things". In: *Europe* 433 (2014), 868MHz.

[55] Tamoghna Ojha, Sudip Misra and Narendra Singh Raghuwanshi. "Wireless sensor networks for agriculture: The state-of-the-art in practice and future challenges". In: *Computers and Electronics in Agriculture* 118 (2015), pp. 66–84.

[56] Nick Lethaby. "Wireless connectivity for the Internet of Things: One size does not fit all". In: *Texas Instruments* (2017), pp. 2–10.

[57] Godfrey Anuga Akpakwu et al. "A survey on 5G networks for the Internet of Things: Communication technologies and challenges". In: *IEEE Access* 6 (2018), pp. 3619–3647.

[58] Aloÿs Augustin et al. "A study of LoRa: Long range & low power networks for the internet of things". In: *Sensors* 16.9 (2016), p. 1466.

[59] LoRa Alliance. *What is LoRaWAN?* 2015. URL: `https://lora-alliance.org/sites/default/files/2018-04/what-is-lorawan.pdf`.

[60] Northstream. *Connectivity technologies for IoT.* URL: `https://www.telenor.no/binaries/Northstream%20-%20Connectivity%20Technologies%20for%20IoT%20-%20Full%20Report%202018_tcm95-353610.pdf` (visited on 10/05/2019).

[61] Andrea Biral et al. "The challenges of M2M massive access in wireless cellular networks". In: *Digital Communications and Networks* 1.1 (2015), pp. 1–19.

[62] Razvan-Andrei Stoica and Giuseppe Thadeu Freitas de Abreu. "6G: the Wireless Communications Network for Collaborative and AI Applications". In: *CoRR* abs/1904.03413 (2019). arXiv: `1904.03413`. URL: `http://arxiv.org/abs/1904.03413`.

[63] Jose Saldana et al. "Alternative Networks: Toward Global Access to the Internet for All". In: *IEEE Communications Magazine* 55.9 (2017), pp. 187–193.

[64] Josef Noll et al. "Internet Lite for Sustainable Development". In: *Nordic and Baltic Journal of Information and Communications Technologies* 2018.1 (2018), pp. 223–238.

[65] Johanna Johansen, Christian Johansen and Josef Noll. "InfoInternet for Education in the Global South: A Study of Applications Enabled by Free Information-only Internet Access in Technologically Disadvantaged Areas (authors' version)". In: *arXiv preprint arXiv:1808.09496* (2018).

[66] Dixit Sudhir and Josef Noll. *Free Access to Information for All (A Vision of the Basic Internet Foundation).* URL: `https://its-wiki.no/images/e/e9/Basic_Internet_White_Paper.pdf` (visited on 10/05/2019).

[67] Kais Mekki et al. "A comparative study of LPWAN technologies for large-scale IoT deployment". In: *ICT Express* 5.1 (2019), pp. 1–7.

[68] Joseph Finnegan and Stephen Brown. "A comparative survey of LPWA networking". In: *arXiv preprint arXiv:1802.04222* (2018).

[69] Usman Raza, Parag Kulkarni and Mahesh Sooriyabandara. "Low power wide area networks: An overview". In: *IEEE Communications Surveys & Tutorials* 19.2 (2017), pp. 855–873.

[70] Congduc Pham, Abdur Rahim and Philippe Cousin. "Low-cost, Long-range open IoT for smarter rural African villages". In: *2016 IEEE International Smart Cities Conference (ISC2).* IEEE. 2016, pp. 1–6.

[71] Stephen Fox. "Third Wave Do-It-Yourself (DIY): Potential for prosumption, innovation, and entrepreneurship by local populations in regions without industrial manufacturing infrastructure". In: *Technology in Society* 39 (2014), pp. 18–30.

[72] Marco Cattani, Carlo Boano and Kay Römer. "An experimental evaluation of the reliability of lora long-range low-power wireless communication". In: *Journal of Sensor and Actuator Networks* 6.2 (2017), p. 7.

[73] Martin Bor and Utz Roedig. "LoRa transmission parameter selection". In: *2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS).* IEEE. 2017, pp. 27–34.

[74] Jansen C Liando et al. "Known and unknown facts of LoRa: Experiences from a large-scale measurement study". In: *ACM Transactions on Sensor Networks (TOSN)* 15.2 (2019), p. 16.

[75] Miguel Kiyoshy Nakamura Pinto et al. "A LoRa enabled sustainable messaging system for isolated communities". In: *Proceedings of the 4th EAI International Conference on Smart Objects and Technologies for Social Good*. ACM. 2018, pp. 118–123.

[76] Albert Pötsch and Florian Haslhofer. "Practical limitations for deployment of LoRa gateways". In: *2017 IEEE International Workshop on Measurement and Networking (M&N)*. IEEE. 2017, pp. 1–6.

[77] Rajkumar Buyya et al. "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility". In: *Future Generation computer systems* 25.6 (2009), pp. 599–616.

[78] Luis M Vaquero et al. "A break in the clouds: towards a cloud definition". In: *ACM SIGCOMM Computer Communication Review* 39.1 (2008), pp. 50–55.

[79] Peter Mell, Tim Grance et al. *The NIST definition of cloud computing*. 2011.

[80] Pedro Garcia Lopez et al. "Edge-centric computing: Vision and challenges". In: *ACM SIGCOMM Computer Communication Review* 45.5 (2015), pp. 37–42.

[81] Yusuke Ito, Hiroyuki Koga and Katsuyoshi Iida. "A bandwidth allocation scheme based on residual bandwidth information in mobile edge computing". In: *Proceedings of the 2nd Workshop on Middleware for Edge Clouds & Cloudlets*. ACM. 2017, p. 3.

[82] Weisong Shi et al. "Edge computing: Vision and challenges". In: *IEEE Internet of Things Journal* 3.5 (2016), pp. 637–646.

[83] Yuan Ai, Mugen Peng and Kecheng Zhang. "Edge computing technologies for Internet of Things: a primer". In: *Digital Communications and Networks* 4.2 (2018), pp. 77–86.

[84] Ashkan Yousefpour et al. "All one needs to know about fog computing and related edge computing paradigms: a complete survey". In: *Journal of Systems Architecture* (2019).

[85] Smith David and Anderson Ed. *Hype Cycle for Cloud Computing*. 2018. URL: `https://irp-cdn.multiscreensite.com/33cdee1b/files/uploaded/hype_cycle_for_cloud_computi_340420.pdf` (visited on 04/05/2019).

[86] Ovidiu Vermesan et al. "The Next Generation Internet of Things– Hyperconnectivity and Embedded Intelligence at the Edge". In: *Next Generation Internet of Things. Distributed Intelligence at the Edge and Human Machine-to-Machine Cooperation* (2018).

[87] Su Jingtao et al. "Steiner tree based optimal resource caching scheme in fog computing". In: *China Communications* 12.8 (2015), pp. 161–168.

[88] Frank Alexander Kraemer et al. "Fog computing in healthcare–a review and discussion". In: *IEEE Access* 5 (2017), pp. 9206–9222.

[89] Charith Perera et al. "Context-aware dynamic discovery and configuration of 'things' in smart environments". In: *Big Data and Internet of Things: A Roadmap for Smart Environments*. Springer, 2014, pp. 215–241.

[90] Michael J O'Grady and Gregory MP O'Hare. "Modelling the smart farm". In: *Information processing in agriculture* 4.3 (2017), pp. 179–187.

[91] AW Murage et al. "Gender specific perceptions and adoption of the climate-smart push–pull technology in eastern Africa". In: *Crop Protection* 76 (2015), pp. 83–91.

[92] Carlos Cambra et al. "Smart system for bicarbonate control in irrigation for hydroponic precision farming". In: *Sensors* 18.5 (2018), p. 1333.

[93] Padma Nyoman Crisnapati et al. "Hommons: Hydroponic management and monitoring system for an IOT based NFT farm using web technology". In: *2017 5th International Conference on Cyber and IT Service Management (CITSM)*. IEEE. 2017, pp. 1–6.

[94] Hirofumi Ibayashi et al. "A reliable wireless control system for tomato hydroponics". In: *Sensors* 16.5 (2016), p. 644.

[95] Marcel Caria et al. “Smart farm computing systems for animal welfare monitoring”. In: *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE. 2017, pp. 152–157.

[96] Francisco Ferrández-Pastor et al. “Developing ubiquitous sensor network platform using internet of things: Application in precision agriculture”. In: *Sensors* 16.7 (2016), p. 1141.

[97] Hong-Linh Truong. “Enabling Edge Analytics of IoT Data: the Case of LoRaWAN”. In: *2018 Global Internet of Things Summit (GIoTS)*. IEEE. 2018, pp. 1–6.

[98] Suk Lee, Mungyu Bae and Hwangnam Kim. “Future of IoT networks: A survey”. In: *Applied Sciences* 7.10 (2017), p. 1072.

[99] RAKwireless Technology. *DIY Enterprise Gateway WisDevice Series RAK7249*. URL: `https://downloads.rakwireless.com/en/LoRa/DIY-Gateway-RAK7249/Hardware-Specification/DIY_Outdoor_Gateway_RAK7249_Product_Brief_V1.2.pdf` (visited on 01/07/2019).

[100] LoRa Alliance. *LoRaWAN 1.0.3 Specification*. 2018. URL: `https://lora-alliance.org/sites/default/files/2018-07/lorawan1.0.3.pdf`.

[101] Abraham Sahilemichael Kebede, IkeOluwapo O Ajayi and Ayodele O Arowojolu. “Effect of enhanced reminders on postnatal clinic attendance in Addis Ababa, Ethiopia: a cluster randomized controlled trial”. In: *Global health action* 12.1 (2019), p. 1609297.

[102] Fidele Ngabo et al. “Designing and Implementing an Innovative SMS-based alert system (RapidSMS-MCH) to monitor pregnancy and reduce maternal and child deaths in Rwanda”. In: *The Pan African Medical Journal* 13 (2012).

[103] Richard Anderson et al. “Supporting immunization programs with improved vaccine cold chain information systems”. In: *IEEE global humanitarian technology conference (GHTC 2014)*. IEEE. 2014, pp. 215–222.

[104] Arunima S Mukherjee. "Empowerment: The invisible element in ICT4D projects? The case of public health information systems in India and Kenya". PhD thesis. University of Oslo, 2017.

A RAK811 Trials

Initially, we wanted to use RAK811[38] lora node issues and action taken trying to fix it WisNode (V1.2) in the end node. It is based on Semtech SX1276 chip and integrated with stm32L. Specifications [39] of RAK811 module is shown in table 12. At the time of this writing, the cost of this module is $22.50. However, we have encountered some problems. The RAK811 V1.2 we had could not communicate with the Arduino Uno. From RAK production description RAK811 is compatible with Arduino UNO.

Table 12: RAK811 WisNode LoRa Module specifications

RAK811 WisNode Specifications	 Information
Chip	Semtech SX1276
Frequency	ISM 868
RX Sensitivity	RSSI -130dBm SNR -15dB
Transmit TX Power	(MAX) 20dBm (TYP) 14dBm
Current consumption	TX mode(14dBm) 30mA RX mode 5.5mA sleep mode 7.2uA

---

[38] `https://store.rakwireless.com/products/rak811-wisnode-lora-module`

[39] `https://doc.rakwireless.com/rak811-wisnode-lora-module/rak811-wisnode-lora-module`

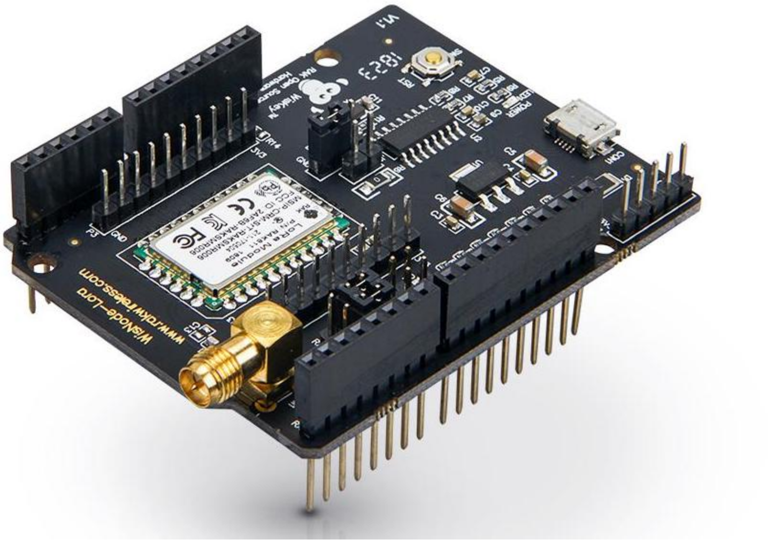
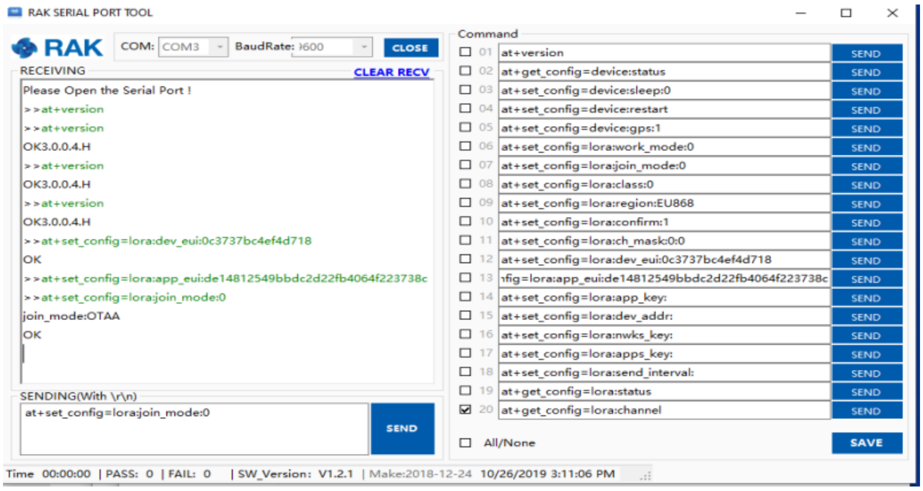


Figure 42: RAK811 WisNode

RAK wireless provides a serial port tool (supports windows only) that is used to configure RAK811 Wisnode. We used this tool to configure the Wisnode. Following the documentation provided by RAK and after an upgrade of firmware, we configured Wisnode and it could connect to the network. The device was registered on LoRaServer and OTAA activated. As shown in figure43 the device joined the network and the status is shown in figure 44.



COM: COM3
BaudRate: 9600

RECEIVING

```
Please Open the Serial Port !
>>at+version
>>at+version
OK3.0.0.4.H
>>at+version
OK3.0.0.4.H
>>at+version
OK3.0.0.4.H
>>at+set_config=lora:dev_eui:0c3737bc4ef4d718
OK
>>at+set_config=lora:app_eui:de14812549bbdc2d22fb4064f223738c
>>at+set_config=lora:join_mode:0
join_mode:OTAA
OK
```

SENDING(With \r\n)

```
at+set_config=lora:join_mode:0
```

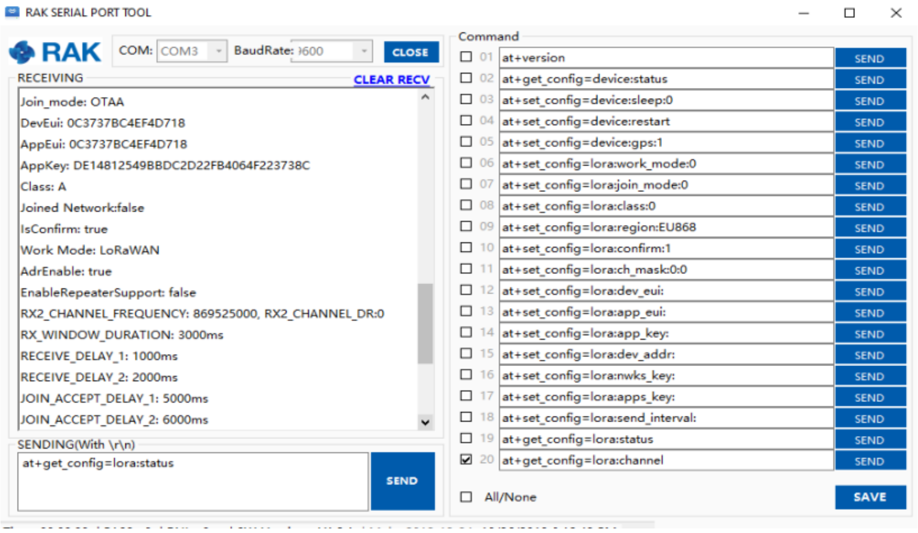
Command

☐ 01 at+version
☐ 02 at+get_config=device:status
☐ 03 at+set_config=device:sleep:0
☐ 04 at+set_config=device:restart
☐ 05 at+set_config=device:gps:1
☐ 06 at+set_config=lora:work_mode:0
☐ 07 at+set_config=lora:join_mode:0
☐ 08 at+set_config=lora:class:0
☐ 09 at+set_config=lora:region:EU868
☐ 10 at+set_config=lora:confirm:1
☐ 11 at+set_config=lora:ch_mask:0:0
☐ 12 at+set_config=lora:dev_eui:0c3737bc4ef4d718
☐ 13 ıfig=lora:app_eui:de14812549bbdc2d22fb4064f223738c
☐ 14 at+set_config=lora:app_key:
☐ 15 at+set_config=lora:dev_addr:
☐ 16 at+set_config=lora:nwks_key:
☐ 17 at+set_config=lora:apps_key:
☐ 18 at+set_config=lora:send_interval:
☐ 19 at+get_config=lora:status
☑ 20 at+get_config=lora:channel

☐ All/None

Time 00:00:00 | PASS: 0 | FAIL: 0 | SW_Version: V1.2.1 | Make:2018-12-24 10/26/2019 3:11:06 PM

Figure 43: RAK811 WisNode network join successful



```
Join_mode: OTAA
DevEui: 0C3737BC4EF4D718
AppEui: 0C3737BC4EF4D718
AppKey: DE14812549BBDC2D22FB4064F223738C
Class: A
Joined Network:false
IsConfirm: true
Work Mode: LoRaWAN
AdrEnable: true
EnableRepeaterSupport: false
RX2_CHANNEL_FREQUENCY: 869525000, RX2_CHANNEL_DR:0
RX_WINDOW_DURATION: 3000ms
RECEIVE_DELAY_1: 1000ms
RECEIVE_DELAY_2: 2000ms
JOIN_ACCEPT_DELAY_1: 5000ms
JOIN_ACCEPT_DELAY_2: 6000ms
```

```
at+get_config=lora:status
```

☐ 01 at+version
☐ 02 at+get_config=device:status
☐ 03 at+set_config=device:sleep:0
☐ 04 at+set_config=device:restart
☐ 05 at+set_config=device:gps:1
☐ 06 at+set_config=lora:work_mode:0
☐ 07 at+set_config=lora:join_mode:0
☐ 08 at+set_config=lora:class:0
☐ 09 at+set_config=lora:region:EU868
☐ 10 at+set_config=lora:confirm:1
☐ 11 at+set_config=lora:ch_mask:0:0
☐ 12 at+set_config=lora:dev_eui:
☐ 13 at+set_config=lora:app_eui:
☐ 14 at+set_config=lora:app_key:
☐ 15 at+set_config=lora:dev_addr:
☐ 16 at+set_config=lora:nwks_key:
☐ 17 at+set_config=lora:apps_key:
☐ 18 at+set_config=lora:send_interval:
☐ 19 at+get_config=lora:status
☑ 20 at+get_config=lora:channel

Figure 44: RAK811 WisNode status

The pH and EC probes that were used in this thesis are compatible with Arduino UNO. The Wisnode is essentially designed to connect with the Arduino directly as it comes in the Arduino Uno form factor. We connected it as shown in figure 45. Using the RAK811 Arduino library[40] and the necessary keys from LoRaServer, we tried to connect the device to the network but this failed. Since the Wisnode could communicate with the gateway when the serial port tool was used, the communication between the Arduino and Wisnode seems to be the problem. We tried connecting the RX of Arduino to TX of RAK811 using jumper wires but this didn't help either. From the product forum, there are discussions on this topic as others are also experiencing the same problem. We did not find a solution to this as such we decided to use the Dragino LoRa shield which offers the same functionality.

---

[40] `https://github.com/RAKWireless/RAK811_LoRaNode`

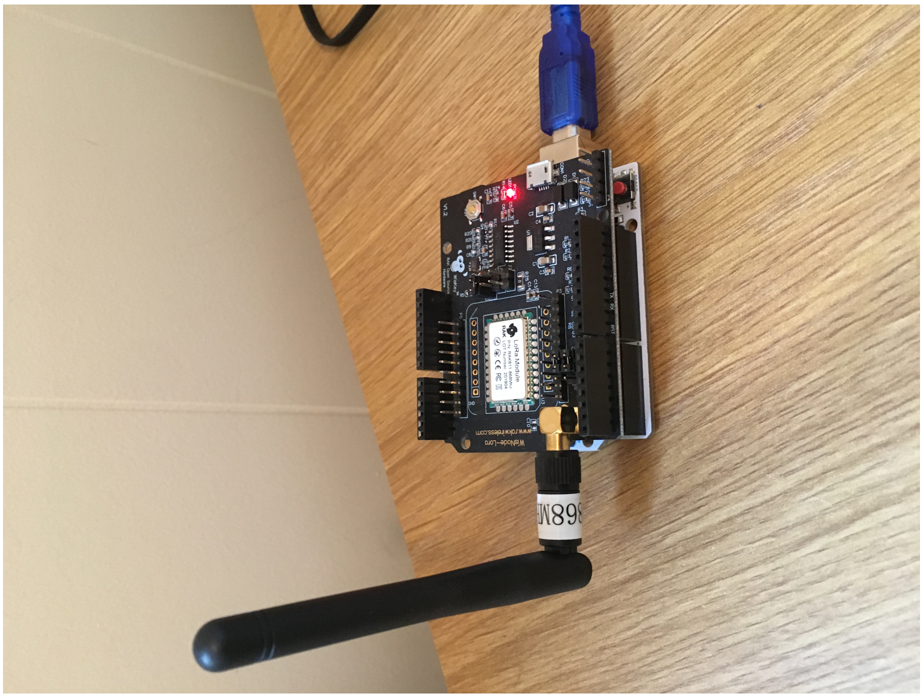


Figure 45: RAK811 WisNode and Arduino connection

B Program code

B.1 End device activation, sensor reading and transmission code

The code below is adapted from the DFRobot library[41] and LMIC library from this repository[42].

```
#include <lmic.h>
#include <hal/hal.h>
#include "DFRobot_EC.h"
#include "DFRobot_PH.h"
#include <EEPROM.h>
```

---

[41] https://wiki.dfrobot.com/Gravity__Analog_Electrical_Conductivity_Sensor___Meter_V2__K=1__SKU_DFR0300

[42] https://github.com/matthijskooijman/arduino-lmic

```
#define PH_PIN A1
#define EC_PIN A2
float voltageEC,voltagePH, phValue, ecValue,temperature = 25;
DFRobot_PH ph;
DFRobot_EC ec;
struct sensorValues {
 float ec;
 float ph;
};

// EUI little-endian format
static const u1_t PROGMEM APPEUI[8] = {  };
void os_getArtEui (u1_t* buf) {
  memcpy_P(buf, APPEUI, 8);
}

// little endian format
static const u1_t PROGMEM DEVEUI[8] = {  };
void os_getDevEui (u1_t* buf) {
  memcpy_P(buf, DEVEUI, 8);
}

static const u1_t PROGMEM APPKEY[16] = {  };

void os_getDevKey (u1_t* buf) {
  memcpy_P(buf, APPKEY, 16);
}

static osjob_t sendjob;

// Schedule TX every this many seconds (might become longer due to duty
// cycle limitations).
const unsigned TX_INTERVAL = 10;
```

```
// Pin mapping
const lmic_pinmap lmic_pins = {
  .nss = 10,
  .rxtx = LMIC_UNUSED_PIN,
  .rst = 9,
  .dio = {2, 6, 7},
};

void onEvent (ev_t ev) {
  Serial.print(os_getTime());
  Serial.print(":␣");
  switch (ev) {
    case EV_SCAN_TIMEOUT:
      Serial.println(F("EV_SCAN_TIMEOUT"));
      break;
    case EV_BEACON_FOUND:
      Serial.println(F("EV_BEACON_FOUND"));
      break;
    case EV_BEACON_MISSED:
      Serial.println(F("EV_BEACON_MISSED"));
      break;
    case EV_BEACON_TRACKED:
      Serial.println(F("EV_BEACON_TRACKED"));
      break;
    case EV_JOINING:
      Serial.println(F("EV_JOINING"));
      break;
    case EV_JOINED:
      Serial.println(F("EV_JOINED"));



      LMIC_setLinkCheckMode(0);
```

```
  break;
case EV_RFU1:
  Serial.println(F("EV_RFU1"));
  break;
case EV_JOIN_FAILED:
  Serial.println(F("EV_JOIN_FAILED"));
  break;
case EV_REJOIN_FAILED:
  Serial.println(F("EV_REJOIN_FAILED"));
  break;
  break;
case EV_TXCOMPLETE:
  Serial.println(F("EV_TXCOMPLETE (includes waiting for RX windows)"));
  if (LMIC.txrxFlags & TXRX_ACK)
    Serial.println(F("Received ack"));
  if (LMIC.dataLen) {
    Serial.println(F("Received "));
    Serial.println(LMIC.dataLen);
    Serial.println(F(" bytes of payload"));
  }
  // Scheduling next transmission
  os_setTimedCallback(&sendjob, os_getTime() + sec2osticks(TX_INTERVAL), d
  break;
case EV_LOST_TSYNC:
  Serial.println(F("EV_LOST_TSYNC"));
  break;
case EV_RESET:
  Serial.println(F("EV_RESET"));
  break;
case EV_RXCOMPLETE:
  // data received in ping slot
  Serial.println(F("EV_RXCOMPLETE"));
  break;
```

```
    case EV_LINK_DEAD:
      Serial.println(F("EV_LINK_DEAD"));
      break;
    case EV_LINK_ALIVE:
      Serial.println(F("EV_LINK_ALIVE"));
      break;
    default:
      Serial.println(F("Unknown␣event"));
      break;
  }
}

void do_send(osjob_t* j) {

  struct sensorValues ss = ecread();
  float structec = ss.ec;
  float structph = ss.ph;
  byte payload1[4];
  uint32_t ecValue = structec*100;
  uint32_t phValue = structph*100;
  payload1[0] =highByte(ecValue);
  payload1[1] =lowByte(ecValue);
  payload1[2] =highByte(phValue);
  payload1[3] =lowByte(phValue);



  if (LMIC.opmode & OP_TXRXPEND) {
    Serial.println(F("OP_TXRXPEND,␣not␣sending"));
  } else {
     LMIC_setTxData2(1, payload1, sizeof(payload1), 0);

    Serial.println(F("Packet␣queued"));
    Serial.println(LMIC.freq);
```

```
  }
  //Next transmission after TX_COMPLETE
}

void setup() {
  Serial.begin(115200);
  Serial.println(F("Starting"));
  ph.begin();
  ec.begin();

  // LMIC init
  os_init();
  // Reset the MAC state. Session and pending
  //data transfers will be discarded.
  LMIC_reset();
  LMIC_setClockError(MAX_CLOCK_ERROR * 1 / 100);
  LMIC_disableChannel(1);
  LMIC_disableChannel(2);
  printotaainformation();

  // Start job (sending automatically starts OTAA too)
  do_send(&sendjob);
}
//print OTAA info
void printotaainformation(void)
{
  unsigned char i;
  unsigned char chartemp;
  unsigned char messagelength;

  Serial.println(F("OTAA␣mode␣to␣join␣network"));
  Serial.print("DevEui:␣");
  for (i = 0; i <= 7; i++)
```

```
  {
    chartemp = pgm_read_word_near(DEVEUI+7-i);
    covertandprint((chartemp >> 4) & 0xf);
    covertandprint(chartemp & 0xf);
  }
  Serial.println(””);
  Serial.print(”AppEui:␣”);
  for (i = 0; i <=7; i++)
  {
    chartemp = pgm_read_word_near(APPEUI+7-i);
    covertandprint((chartemp >> 4) & 0xf);
    covertandprint(chartemp & 0xf);
  }

  Serial.println(””);
  Serial.print(”AppKey:␣”);
  //memcpy_P(buftemp, APPKEY, 16);
  for (i = 0; i <= 15; i++)
  {
    chartemp = pgm_read_word_near(APPKEY+i);
    //Serial.print(buftemp[i],HEX);
    covertandprint((chartemp >> 4) & 0xf);
    covertandprint(chartemp & 0xf);
  }
  Serial.println(””);

}

void covertandprint(unsigned char value)
{
  switch (value)
  {
    case 0  : Serial.print(”0”); break;
```

```
    case 1  : Serial.print(”1”); break;
    case 2  : Serial.print(”2”); break;
    case 3  : Serial.print(”3”); break;
    case 4  : Serial.print(”4”); break;
    case 5  : Serial.print(”5”); break;
    case 6  : Serial.print(”6”); break;
    case 7  : Serial.print(”7”); break;
    case 8  : Serial.print(”8”); break;
    case 9  : Serial.print(”9”); break;
    case 10  : Serial.print(”A”); break;
    case 11  : Serial.print(”B”); break;
    case 12  : Serial.print(”C”); break;
    case 13  : Serial.print(”D”); break;
    case 14  : Serial.print(”E”); break;
    case 15 :  Serial.print(”F”); break;
    default :
      Serial.print(”?”);   break;
  }
}

//Read sensor Values
struct sensorValues ecread(){

  float ecValueRead;
  struct sensorValues val;
    char cmd[10];
   static unsigned long timepoint = millis();
    if(millis()-timepoint>1000U)  //time interval: 1s
    {
      timepoint = millis();
      // read the ph voltage
       voltagePH = analogRead(PH_PIN)/1024.0*5000;
       // convert voltage to pH with temperature compensation
```

```
      phValue     = ph.readPH(voltagePH,temperature);


     voltageEC = analogRead(EC_PIN)/1024.0*5000;
     // convert voltage to EC with temperature compensation
     ecValueRead =  ec.readEC(voltageEC,temperature);



    val.ec = ecValueRead;
    val.ph = phValue;

     delay(60*60*1000);

     return val;

     //return ecValueRead;
   }

   if(readSerial(cmd)){
      strupr(cmd);
      if(strstr(cmd,"PH")){
       //PH calibration process by Serail CMD
        ph.calibration(voltagePH,temperature,cmd);
       }
       if(strstr(cmd,"EC")){
         //EC calibration process by Serail CMD
        ec.calibration(voltageEC,temperature,cmd);
       }
   }

}

int i = 0;
```

```
bool readSerial(char result[]){
    while(Serial.available() > 0){
        char inChar = Serial.read();
        if(inChar == '\n'){
             result[i] = '\0';
             Serial.flush();
             i=0;
             return true;
        }
        if(inChar != '\r'){
             result[i] = inChar;
             i++;
        }
        delay(1);
    }
    return false;
}


void loop() {
  os_runloop_once();

}
```

B.2 Custom decode function in LoRa App Server

Custom JavaScript decode function was used in the LoRa App Server

```
function Decode(fPort, bytes) {
    var ec = (bytes[0] << 8) | bytes[1];
    var ph = (bytes[2] << 8) | bytes[3];

    var dataout = {
        "sensorvalues": {
```

```
            'ec': ec / 100,
            'ph': ph / 100
        },
    };
    return dataout;
}
```

B.3 Node-RED Flow

```
[
    {
        "id": "6e4a0c5b.0bd304",
        "type": "tab",
        "label": "Smart Farm",
        "disabled": false,
        "info": ""
    },
    {
        "id": "4bc69f45.b1789",
        "type": "mqtt in",
        "z": "6e4a0c5b.0bd304",
        "name": "EC_pH",
        "topic": "application/4/device/8a90dc387df11f42/rx",
        "qos": "1",
        "broker": "e414bdfe.fd972",
        "x": 398,
        "y": 93,
        "wires": [
            [
                "5e559f37.01cdb",
                "729125f5.895e3c"
            ]
        ]
    },
```

```
{
    "id": "5e559f37.01cdb",
    "type": "function",
    "z": "6e4a0c5b.0bd304",
    "name": "getData",
    "func": "p=JSON.parse(msg.payload);\nnode.log(typeof p);
    \nmsg.payload= {\n    ec: p.object.sensorvalues.ec,\n
     ph: p.object.sensorvalues.ph\n}\nreturn msg;",
    "outputs": 1,
    "noerr": 0,
    "x": 617,
    "y": 134,
    "wires": [
        [
            "40c5596a.763538"
        ]
    ]
},
{
    "id": "40c5596a.763538",
    "type": "influxdb out",
    "z": "6e4a0c5b.0bd304",
    "influxdb": "8b7cdb11.e1c4d8",
    "name": "InfluxDB",
    "measurement": "sensorValues",
    "precision": "",
    "retentionPolicy": "",
    "x": 738,
    "y": 246,
    "wires": []
},
{
    "id": "729125f5.895e3c",
```

```
    "type": "function",
    "z": "6e4a0c5b.0bd304",
    "name": "checkValues",
    "func": "p=JSON.parse(msg.payload);
    \nnode.log(typeof p);\nec= p.object.sensorvalues.ec;
    \nph=p.object.sensorvalues.ph;\n\nif(ec<0.8 || ec > 1.2 )
    {\n   return {payload:\"46247458 EClevelislow\"};\n}
    else if(ph<5.5||ph>6.5){\n    return {payload:\"46247458 pHlevelisHigh\"};\n}\n",
    "outputs": 1,
    "noerr": 0,
    "x": 488,
    "y": 241,
    "wires": [
        [
            "446a71bd.7526e",
            "c81b2822.549468"
        ]
    ]
},
{
    "id": "446a71bd.7526e",
    "type": "debug",
    "z": "6e4a0c5b.0bd304",
    "name": "",
    "active": true,
    "tosidebar": true,
    "console": false,
    "tostatus": false,
    "complete": "false",
    "x": 765,
    "y": 346,
    "wires": []
},
```

```
{
    "id": "c81b2822.549468",
    "type": "exec",
    "z": "6e4a0c5b.0bd304",
    "command":
    "php ~/EnvayaSMS-master/server/php/example/send_sms.php ",
    "addpay": true,
    "append": "",
    "useSpawn": "false",
    "timer": "",
    "oldrc": false,
    "name": "send SMS",
    "x": 534,
    "y": 460.5,
    "wires": [
        [
            "32de7d8d.e766c2"
        ],
        [],
        []
    ]
},
{
    "id": "32de7d8d.e766c2",
    "type": "debug",
    "z": "6e4a0c5b.0bd304",
    "name": "",
    "active": true,
    "tosidebar": true,
    "console": false,
    "tostatus": false,
    "complete": "false",
    "x": 752,
```

```
        "y": 505,
        "wires": []
    },
    {
        "id": "e414bdfe.fd972",
        "type": "mqtt-broker",
        "z": "",
        "name": "",
        "broker": "localhost",
        "port": "1883",
        "clientid": "client3",
        "usetls": false,
        "compatmode": true,
        "keepalive": "60",
        "cleansession": true,
        "birthTopic": "",
        "birthQos": "0",
        "birthPayload": "",
        "closeTopic": "",
        "closePayload": "",
        "willTopic": "",
        "willQos": "0",
        "willPayload": ""
    },
    {
        "id": "8b7cdb11.e1c4d8",
        "type": "influxdb",
        "z": "",
        "hostname": "127.0.0.1",
        "port": "8086",
        "protocol": "http",
        "database": "SMARTFARM",
        "name": "",
```

```
        "usetls": false,
        "tls": ""
    }
]
```