Low-Power Smart Devices for the IoT Revolution

Matteo Nardello

Doctoral School in Materials, Mechatronics and Systems Engineering

Department of Industrial Engineering

University of Trento

A thesis submitted in partial fulfillment of the requirements for the
Degree of Doctor of Philosophy
in Materials, Mechatronics and Systems Engineering

Supervisor Prof. Davide Brunelli

September 2020

Acknowledgements

Looking back on my Ph.D. journey, one thing that has become clear is that even if it is possible to navigate this path alone, it is probably not worth it. A lot of people should be thanked for the work presented in this thesis, and for the help they provided me. I have learned an immense amount from the people I have had the opportunity of working with, including new technical skills and how to be a successful researcher.

First of all, I want to express my sincere and personal gratitude to my Advisor, Davide Brunelli, who has wisely guided me towards interesting ideas over the years while making sure that the results get written up too. Thank you for your help and trust.

Next are my former and new colleagues, Maurizio, Pietro, Alessandro, Luca G. and Luca S. that helped and supported me exploring and implementing new ideas, while also distracting me during hard times.

I also want to mention Brandon Lucia, Harsh Desai and all the Abstract Research Group at CMU for supporting and providing me invaluable guidance during my internship in Pittsburgh.

I am extremely grateful to my parents for their love, sacrifices and endlessly patience that allowed me to complete this strange – but incredible – journey.

I also want to thank my second family that I met in Trento – Elmar, Michele, Marcello, Daniel e Alessio – for the great time we spent during these years. Finally, I want to extend my sincere gratitude to all my other friends to be always there for me, also during hard times. You have made this journey less burdensome. I don't think I would have made it this far without you all.

Abstract

Internet of Things (IoT) is a revolutionary paradigm approaching both industries and consumers everyday life. It refers to a network of addressable physical objects that contain embedded sensing, communication and actuating technologies, to sense and interact with the environment where being deployed. It can be considered as a modern expression of Mark Weiser's vision of ubiquitous computing where tiny networked computers become part of everyday objects, fusing together the virtual world and the physical word.

Recent advances in hardware solutions have led to the emergence of powerful wireless IoT systems that are entirely energy-autonomous. These systems extract energy from their environment and operate intermittently, only as power is available. Battery-less sensors present an opportunity for the pervasive widespread of remote sensor deployments that require little maintenance and have low cost.

As the number of IoT endpoint grows – industry forecast trillions of connected smart devices in the next few years – new challenges to program, manage and maintain such a huge number of connected devices are emerging. Web technologies can significantly ease this process by providing well-known patterns and tools - like cloud computing - for developers and users. However, the existing solutions are often too heavyweight or unfeasible for highly resource-constrained IoT devices.

This dissertation presents a comprehensive analysis of two of the biggest problems that the IoT is currently facing: R1) How are we going to provide connectivity to all these devices? R2) How can we improve the quality of service provided by these tiny autonomous motes that rely only on limited energy scavenged from the environment?

The first contribution is the study and deployment of a Low-Power Wide-Area-Network as a feasible solution to provide connectivity to all the expected IoT devices to be deployed in the following years. The proposed technology offers a novel communication paradigm to address discrete IoT applications, like long-range (i.e., kilometers) at low-power (i.e., tens of mW). Moreover, results highlight the effectiveness of the technology also in the industrial environment thanks to the high immunity to external noises.

In the second contribution, we focus on smart metering presenting the design of three smart energy meters targeted to different scenarios. The first design presents an innovative, cost-effective smart meter with embedded non-intrusive load monitoring capabilities intended for the domestic sector. This system shows an innovative approach to provide useful feedback to reduce and optimize household energy consumption. We then present a battery-free non-intrusive power meter targeted for low-cost energy monitoring applications that lower both installation cost due to the non-intrusive approach and maintenance costs associated to battery replacement. Finally, we present an energy autonomous smart sensor with load recognition capability that dynamically adapts and reconfigures its processing pipeline to the sensed energy consumption. This enables the sensor to be energy neutral, while still providing power consumption information every 5 minutes.

In the third contribution, we focus on the study of low-power visual edge processing and edge machine learning for the IoT. Two different implementations are presented. The first one discusses an energy-neutral IoT device for precision agriculture, while the second one presents a battery-less long-range visual IoT system, both leveraging on deep learning algorithms to avoid unnecessary wireless data communication. We show that there is a clear benefit from implementing a first layer of data processing directly in-situ where the data is acquired, providing a higher quality of service to the implemented application.

Table of Contents

List of Figures

List of Tables

Introduction

The human factor is what historically limit measurement, as any data acquisition required a person to be involved. Peter Drucker, an Austrian-born American management consultant, is often quoted as saying: *"you can't manage what you can't measure."* Even though the quote refers to the economic context, the same concept can be applied to broader scenarios. Only after an in-deep evaluation of the environment we can excogitate improvements plans, and only after measuring the changes it is possible to evaluate the outcomes.

You can't know whether or not you are successful unless success is defined and tracked.

Today, however, we have the tools and opportunity to obviate to this problem. Thanks to information technology we can expand monitoring beyond what we could accomplish ourselves. As this computing aided monitoring has been proven effective for business efficiency, public safety, environmental sustainability and personal health, to name a few, there is an increasing interest in extending what we can measure and therefore manage.

Sustained by the recent advances in miniaturization and low-power sensing and computing technology, Internet of Things (IoT) devices are increasingly capable of interacting with the physical world. Systems that sense, compute and communicate are now frequently deployed into human environments to sense and process important signals for a range of different applications. Some examples includes security [1]–[3], environmental science [4]–[6], urban planning and optimization [5], [7], [8], precision agriculture [9]–[11], and, more recently, space economy [12]–[14].

The purpose of IoT is to transform the way we live today; converting passive objects around us into intelligent devices that can perform daily tasks and chores.

In the context of sustainable development, many governments worldwide have started to invest [15], [16] to transform Cities into Smart Cities, with the aim of providing a better quality of life to citizens. With over 50% of the world's population now in cities, significant strains are placed on city resources and infrastructures. In this revolutionary process, cities adopt new technologies (i.e., IoT) to their core systems to better manage and monitor the use of limited resources. Smartness can be seen as the ability to extract information from the environment, with the help of Information and Communications Technologies (ICT), and turn them into knowledge that can later be used to improve services for any citizen.

The use of IoT to modernize cities promises to create Smart Cities that mitigate the impacts of increased city populations while improving the quality of life for all inhabitants [17]. Smart Cities are large, complex, distributed and continuous systems containing and using mission-critical data. They are increasingly becoming IoT-enabled and IoT dependent [18].

While the growth in the number of deployed sensors will accelerate in the next decades, many challenges related to successfully deploying so many devices are still unaddressed. Current methods of managing, networking and powering devices will likely not scale to support the trillion-sensor vision [19]. Communication protocols will have to evolve as the number of devices increase, while device management, maintenance and replacement will be an amplified concern.

Due to the heterogeneity and the large number of deployed devices spread in any context, IoT faces many problems/challenges. Leaving out sociological challenges in making people aware and knowledgeable of the technology, we can identify some technological challenges in the system design, data usability, security and privacy that must be addressed to develop a future proof IoT device [20]–[27]:

Cost. A pervasively deployed sensing system must have a low cost. Devices must be manufactured at large scale (i.e., millions) requiring inexpensive devices to minimize the total cost .

Maintenance. A pervasively deployed sensing system must operate for a long period of time without maintenance. A device should require no component or battery replacements over its lifetime. A device should operate without costly centralized power or communications infrastructure; fully autonomous and wireless operation is ideal.

Communication. A pervasively deployed sensing system must make judicious use of bandwidth. Long-range wireless links typically have a low data rate [28], [29] and the system should ensure that any transmitted byte is valuable to the end-to-end application that the system implements. Moreover, to respect the cost, maintenance, and distribution requirements, the system must limit the number of required base station.

Environmental Impact. A pervasively deployed sensor must minimize its negative impact on the environment into which it is deployed. It must minimize its environmental impact in the short term by being small, unobtrusive, and by not interfering with existing (e.g., radio) infrastructure. It must minimize its environmental impact in the long term by minimizing the amount of hazardous chemical waste due to batteries and other toxic components.

Thesis statement

Pervasive and distributed sensing and monitoring will require the development of energy harvesting enabled sensors for the majority of the application. The energy intermittency and unpredictability nature that characterize energy harvesting operations, however, pose serious strain on current sensor networks.

A new distributed wireless sensor network architecture must be investigated. New algorithms for in-situ data analysis to lower the energy cost associated to wireless communication must be developed. Data processing strategies must be optimized by strategically and dynamically reconfigure processing pipelines to react to energy availability, deferring energy-intensive operations.

Meeting this requirement would enable to perpetually operate small, unobtrusive and inexpensive devices for future sensing problems.

Contributions of this Dissertation

This dissertation contributes to the study and implementation of novel IoT solutions, targeted to real-world sensing needs, inspired by the state of the art. The goal is to reduce power consumption while improving services and operations.

Low-Power Long-Rage network deployment. After an in-depth analysis of the technology, a LoRa network is deployed and analyzed. Exploiting the installation of a full channel LoRaWAN gateway on top of our university campus, we have evaluated LoRa's link performance, both in terms of energy requirements and achievable range. The final goal is to provide an open network for the municipality of Trento where low-power IoT devices can connect and stream acquired data to the Internet. Due to the unique characteristic of LoRa, we have also evaluated and tested the technology for industrial process monitoring showing the potential of this communication standard.

Smart energy meters for energy efficiency in smart cities. We present a new generation of smart meters running advanced algorithms for power consumption analysis. These devices are tailored for tracking and monitoring single appliances starting from the aggregate power consumption of a whole building. Load monitoring can foster the development of advanced strategies for the production, distribution and consumption of energy, and also to better exploit the availability of renewable energy. Different approaches are explored, based on the application scenario (i.e., Domestic or Industrial), with also the integration of energy harvesting power supply to ease the installation and to lower maintenance costs, thanks to the non-intrusive approach [30].

Battery-less visual sensing systems. Future systems must support video and image sensor data to directly, rather than indirectly, observe complex environmental phenomena. Some applications will also demand deployment into environments requiring wireless data backhaul at kilometer scale. The tandem requirements of high-data-rate sensing and long-range communication pose a key challenge for an energy-constrained system. We present two different IoT visual sensing systems with energy harvesting capabilities that harvest energy using small solar panels, storing energy in a small supercapacitor. These devices address the challenge of energy and bandwidth scarcity by spending energy to apply sophisticated at-sensor processing pipelines to sensed data.

Thesis outline

In **Chapter 1**, we present the IoT paradigm. After a short introduction about the origin of the IoT we discuss the system architecture and the common problems that are currently limiting the widespread of IoT solutions. Here we also give a first sneak peek on the importance of ubiquitous computing applied to the IoT.

Chapter 2 deals with a key technology for providing connectivity to IoT devices, LoRa. The chapter starts with an introduction to the technology, showing the key characteristics that make this protocol the perfect candidate for long-range low-power communications, as required by the IoT. We then present the deployment of a local LoRa network along with range test to validate the effectiveness of the proposed protocol. Closing this chapter, the study of LoRa performance in the industrial environment where the long-range capability can be traded for a higher immunity to external interference and superior sensitivity.

In **Chapter 3** we discuss the importance of a distributed smart energy monitoring infrastructure to foster energy efficiency in smart cities. Providing real-time feedback regarding energy consumption to both end-users and utility companies, can foster the development of advanced strategies for the production, distribution and consumption of energy. Also to provide useful insight to better exploit the availability of renewable energy. Here the key aspect that a smart meter should have are discussed and evaluated.

Chapter 4 presents the importance of edge computing applied to energy-constrained low-power IoT sensing devices. Here two example of low power smart cameras that leverage deep learning to avoid un-necessary data streaming are discussed. As shown, machine learning is now possible even in resource constrained devices that rely only on the energy harvested from the environment. This enables IoT nodes to efficiently use the limited energy scavenged from the environment to apply sophisticated processing on the acquired data, to avoid the wireless transmission of un-interesting signals.

Closing this dissertation, **Chapter 5** concludes this thesis with a summary and some final remarks.

Chapter 1

The Rise of the IoT

1.1 The first IoT device

Before there were Internet-connected thermostats, umbrellas, toaster and juicers – even before the modern Internet was developed – there was a common Coke selling machine at CMU, Pittsburgh, that could report its status through a network. Even though the type of data provided was quite limited and the implementation primitive for today's standards, it represents, as far as anyone knows, the world's first and unique IoT device.

In 1970, students and faculty studying and working in Wean Hall on the CMU campus were consuming a high number of Coca-Cola bottles each day. The number of sodas sold was so high that it had the highest sales volume of any Coke machine in the Pittsburgh area. Because of this high consumption rate and a rather erratic loading schedule, customers would frequently find themselves standing in front of an empty Coke machine or, even worse, to receive a recently loaded, still-warm Coke. This issue was even worsened during the mid-seventies expansion of the department, where people's offices where relocated further away from the main terminal room where the Coke machine stood.

To overcome this problem, one day, a couple of people got together to find a solution. They equipped the vending machine with micro-switches to sense how many bottles were available, and wired them to the main departmental computer. A server program was written to keep track of the Coke machine's state, so students and faculty could ping the computer for information about the availability of soft drinks. The program also tracked for how long the bottles had been in the machine after restocking, in order to know if the bottle was still warm. After three hours, the bottles simply registered as "cold." Finally, the group added the code to the main computer's finger[1] program, which allowed anyone on a computer connected to the ARPANET, or anyone connected to CMU's local Ethernet, to access information about the machine. With a simple finger request, they could find out if there were any Cokes in the machine and, if so, which ones were cold.

Considering the connected nature of the coke machine and the type of the service it provided, It was, as far as the author is aware, what today can be defined as the world's first rudimental example of an IoT device[2]. Figure 1.1 show the coke vending machine installed in Wean Hall.

Figure 1.1: CMU's connected Coke machine

1.2 The origin of the IoT

The idea of adding sensor and intelligence to basic objects was discussed for the first time throughout the 1980s and 1990s, but apart from some early projects – including the previously presented network-connected vending machine – progress was slow, mostly due to the fact that technology wasn't ready. Before it became cost-effective to install and connect-up billions of devices, microcontroller and

[1] https://tools.ietf.org/html/rfc1288
[2] https://www.cs.cmu.edu/~coke/history_long.txt

sensors cheap and power-efficient enough to be suitable for the development of a connected network of low-cost sensor nodes were required.

The first big step towards what nowadays is defined *The Internet of Things* was made by the introduction and adoption of RFID, Radio-Frequency Identification, tags; passive, low-power nodes integrating short-range wireless communication.

The term *IoT* was first used by Kevin Ashton[3], back to 1999, during a presentation to executives at Proctor & Gamble, the multinational corporation focused on cleaning agents and personal care products. At that time, the focus was on the potential benefits and improvements bring by the integration of the RFID technology in P&G's supply chain.

The IoT was initially most interesting to business and manufacturing, where its application is sometimes known as Machine-to-Machine (M2M). This mainly due to the costs derived by the technology available in those years. But, since then, the cost of adding sensors and an internet connectivity to objects has continued to fall, making possible to connect nearly everything to the internet; event "dumb" consumer products that are daily used by everyone around the world (i.e., smart shoes [31]).

Today, IoT refers to the billions of physical devices around the world that are now connected to the internet, collecting and sharing data. Thanks to cheap processors and wireless networks it's possible to turn anything, from a slipper to self-driving car to an autonomous UAVs, into part of the global internet network. This adds a level of digital intelligence to devices that would be otherwise passive, enabling them to communicate real-time data without a human being involved, effectively merging the digital and physical worlds.

1.3 IoT system architecture

In the literature we can find different IoT architectures proposed by researchers [32], [33]. While a reference model can probably be identified, several archi-

[3]https://www.rfidjournal.com/articles/view?4986

tectures actually coexist, as the definition of IoT covers a wide range of computer devices around us.

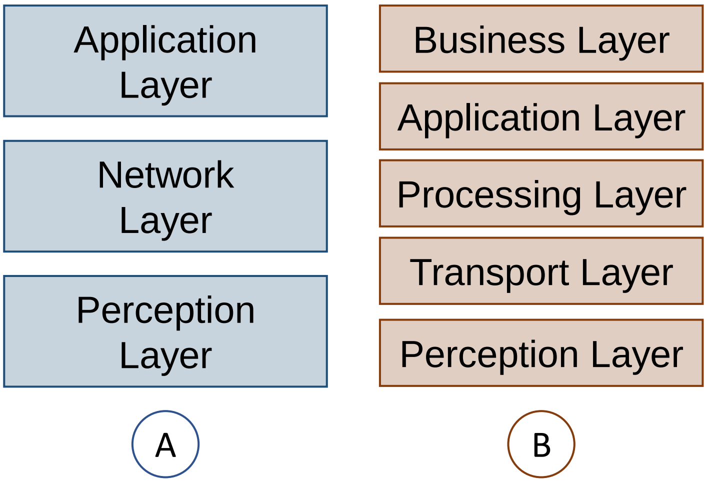


Figure 1.2: IoT Three (A) and Five (B) layers architecture model

1.3.1 Protocol based architectures

Basic and Standard architectures divide the IoT ecosystem into three layers [34]–[36] presented in Figure 1.2a. This division has been proposed in the early stages of IoT research. It has three main layers, namely, *perception*, *network*, and *application*.

Perception Layer The perception layer is the lowest level and encompasses the physical layer, which has sensors for acquiring useful information about the environment, or actuators for executing some tasks. It senses some physical parameters (i.e. temperature or air quality) or identifies other interesting elements (i.e. object and people detection) in the environment.

Network Layer Above the physical layer there is the network layer responsible for connecting to other "things", network concentrators, and cloud servers. It's in charge of transmitting, receiving and processing sensor data.

Application Layer On top we finally find the application layer responsible for implementing and delivering application specific services to the end user.

It defines various applications in which the IoT can be deployed. Some of them are, for example, smart buildings, smart cities and smart farming.

This simple three-layer architecture can be useful to define the main idea of the IoT ecosystem, but it is limiting as research often focuses on finer aspects. To overcome this limit, researchers have proposed to divide the IoT into more layers which additionally includes the *processing* and *business layers* [34]–[37]; a five-layer architecture, proposed in Figure 1.2b. Perception and application layers cover the same aspects as the architecture with three layers. The function of the remaining three layers is:

Transport Layer The transport layer is in charge of streaming the sensor data from the perception layer to the processing layer and vice versa using a wired or wireless medium.

Processing Layer The processing layer is also known as the middleware layer. It stores, analyzes, and processes all the data acquired by the physical layer. It can manage and provide a diverse set of services to lower layers. It employs many technologies such as databases, cloud computing and big data processing modules. It is worth to be noted that in this classification, the processing layer is not located in the physical devices.

Business Layer The top layer is represented by the business layer. It manages the whole IoT application, including, business and profit models and users' privacy.

1.3.2 System based architectures

The two architectures presented in the previous section are somehow outdated. Thanks to the technological progress, IoT nodes can now implement also a pre-processing layer that analysis the data coming from the physical layer or from the transport layer. It makes sense to move from an architecture based on protocol to one that describes the IoT as a complex system where edge devices - things but also gateways and concentrators - carry out a substantial amount of computation, storage and communication locally. This architecture takes its name after the paradigm that it implements: fog computing [33].

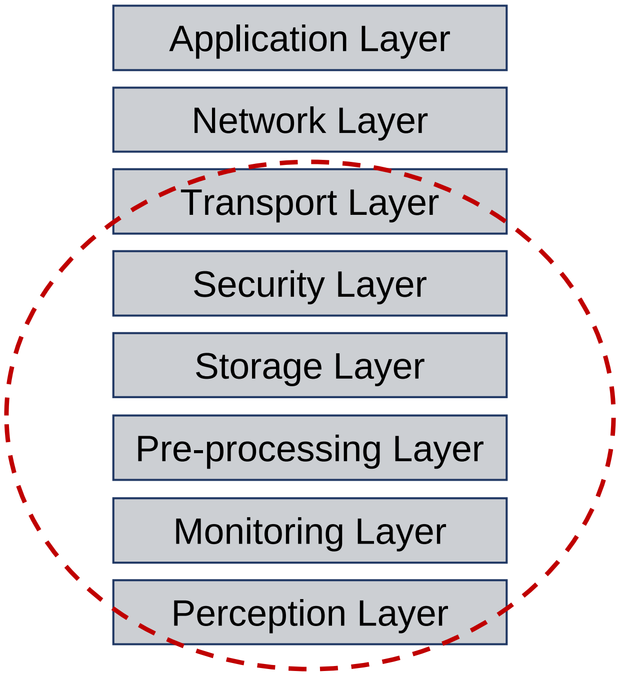


Figure 1.3: Fog Computing enabled IoT architecture. The red dash line highlights the layers that modern IoT device can implement

A fog architecture [38]–[40] presents a layered approach as the one depicted in Figure 1.3, which also includes monitoring, pre-processing, storage and security layers between the perception and transport layers.

Monitoring Layer The monitoring layer monitors power, resources, and responses of the physical device.

Pre-Processing Layer The pre-processing layer performs a first filtering, processing and analysis of the data coming from the physical layer.

Storage Layer We then have the storage layer that provides storage functionalities such as data retention, distribution, and storage.

Security Layer Finally, the security layer performs encryption/decryption and ensures data integrity and privacy.

This paradigm envisions adding smart data pre-processing capabilities to physical devices such as motors, pumps or lights with the aim to do as much data filtering as possible on the edge. The final goal is to minimize the bandwidth required for data streaming and to exploit the computational power of edge devices.

This is the reference architecture that we will consider in this dissertation, as it presents all the macro areas where our work focuses.

1.4 Toward ubiquitous computing: intelligence on the edge

The IoT notation does not refers only to "things" but to the whole end-to-end architecture, as presented in the previous section. It obviously requires compute and storage capacity for processing and storing data coming from the end-nodes. As a matter of fact, managing such resources poses non-trivial requirements to platform designers. Tens of billions of heterogeneous mobile and fixed endpoints distributed over vast geographical areas, and arranged for a variety of different use cases will require different computational resources. Many of these applications will have stringent requirements as very low latency, periods of high throughput and prompt reaction to real-time data acquired; others will have to deal with quite restrictive constraints as long-range links, energy harvesting intermittence and lossy communications.

Eased by the recent growth of remote computing services, the cloud may seem a solution to where IoT compute and storage resources should be placed. Unfortunately, the requirements and design space of IoT make cloud computing unfeasible in numerous scenarios, wasting all the computational power now made available even by ultra-low-power microcontrollers.

One solution is represented by what is defined as *Fog Computing* [39], [41] and its extension *Edge Computing* [42]–[45]. This creates *Ubiquitous Computing*, a model in which data can be analyzed and processed by applications running in devices at different levels within the network rather than in a centralized manner in the Cloud. Ubiquitous Computing is not devised as a competitor of the Cloud

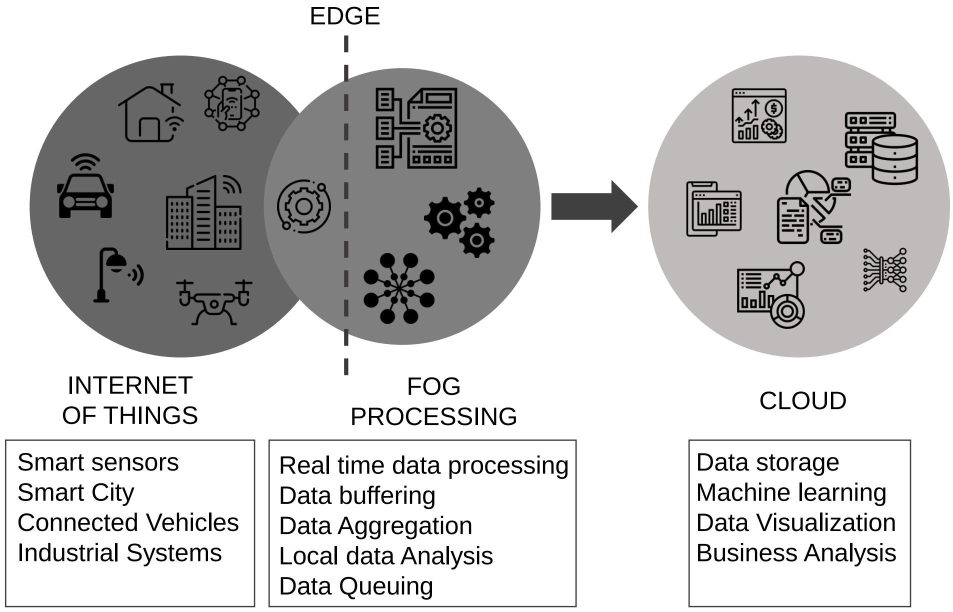


Figure 1.4: Ubiquitous Computing extend the fog model providing real time data analysis directly on the IoT device.

but as an extension for a broad spectrum of use cases and applications for which traditional Cloud Computing is not sufficient.

Ubiquitous computing enables:

- Data analysis on the edge as well as selectively transmitting and receiving data from the Cloud;
- Data aggregation at the edge;
- Adaptability and scalability upon network expansions;
- In-situ real-time decision making without query the Cloud;
- Optimization of computational resources and transmission bandwidth.

The essence of this computing model is schematically shown in Figure 1.4.

1.5 IoT challenges

Industry projections (figure 1.5) forecast that around 22 billion sensors will be deployed by 2025[4], that will be around 3 smart objects for every human being on Earth, taking a world population of around 8 billions[5]. This presents new opportunities for Smart Cities, Responsive Buildings, Instrumented Infrastructure, and many other applications where adding sensing capabilities could dramatically improve how those systems operate.

This soaring number of deployed sensors opens many challenges that can be identified in several areas:

Connectivity. Connecting billions of devices in virtually the same network is not an easy task. What make this task even more difficult is the heterogeneity that we can expect from the involved devices since the many different application scenarios. In some situations, certain communication approaches are totally unfeasible. For instance, the wide availability of the GSM network could not be exploited by ultra-low-power IoT node, due to the energy requirement of a GSM modem. A device feasible for the IoT paradigm must address these problems during the design phase.

Power consumption. All electronic devices require power to operate. We can have AC main powered devices, like smart lighting systems, that uses the power grid; we can have sensors deployed in remote location that either rely on batteries, or use some form of energy harvesting. In both cases, due to the number of expected IoT nodes, devices should be developed such that they are as energy efficient as possible, integrating some sort of sustainable energy harvesting technologies, as relaying only on battery is unfeasible. This must be taken into consideration during the design phase, since we move from the stability and predictability of batteries to the intermittent and unpredictable nature of the energy harvesting.

[4] https://iot-analytics.com/product/state-of-the-iot-2018-2019-winter-edition/
[5] https://population.un.org/wpp/

System Architecture. When designing a new device that have to be integrated into and existing infrastructure or can be pervasively deployed, the system architecture and design must be carefully evaluated. The heterogeneity of the IoT makes it difficult to create a single killer architecture. The development must be tailored to the specific application scenario, as solutions for a certain domain are inapplicable to others, either due to functional requirements, or due to hardware differences. The challenge in the IoT is to construct such architectures, with portability, auto-configuration, integration and connectivity in mind. Excogitating solutions that do not represent a barrier when retrofitting existing infrastructures, minimizing its negative impact on the environment into which it is deployed.

Security and user Privacy. As the role of the IoT is every day more important in our daily lives, security and privacy concerns are arising. It is very important to ensure that the data is secure and only available to authorized users, as the potentially collectable knowledge and its impact is enormous. Based on the heterogeneity and the scale of the IoT, such security problems are more complex compared to the security problems that we have faced till now. Any solutions have to be lightweight and portable to a wide set of devices, despite their intrinsic differences.

Computational complexity. Potentially, all the devices that shape the IoT can generate massive amounts of data, at an unpredictable rate; continuous or bursty, in structured or unstructured form. This might create enormous pressure on the backbone infrastructure and networking. The challenge is to fully exploit the computational power nowadays made available by the modern microcontrollers, optimizing the efficiency of what can be defined as a Distributed Computing System. This enables a first analysis of the collected data that can be discarded if not useful for the application, lowering wireless channel occupation and the amount of information that must be transmitted, analyzed and stored by the cloud.

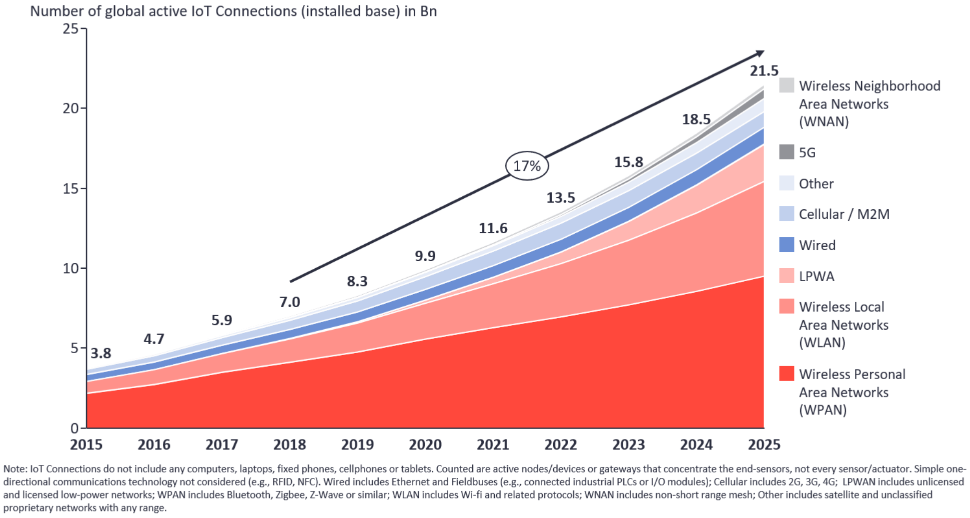


Figure 1.5: Global number of connected IoT devices forecast. Source [46]

Even if this contribution is mainly focused on the study of *Connectivity*, *Power Consumption* and *Ubiquitous Computing*, it is clear that IoT is a complex matter that depends on the progress in several fields, including but not limited to wireless sensor networks, communication standards, distributed system and computer security.

Chapter 2

Low-Power Wide Area Network

2.1 Introduction

Low Power Wide Area Networks (LPWANs) are increasingly seen as an attractive communication platform for the deployment of city-scale IoT networks. They offer the ability to interconnect devices and gateways over distances of many kilometers, and are specifically tailored for low-power, low-maintenance and low-throughput sensing applications.

Several radio technologies (Sigfox[1], LoRa[2], NB-IoT[3], LTE-M[4] and DASH7[5]) are currently begin investigated to better characterize network performances in different environment and working conditions. A comparison between range and data rate is presented in figure 2.1.

One of the most interesting technology in this context is *LoRa*, proposed by Semtech and further promoted by the LoRa Alliance[6]. At the heart of LoRa's success an adaptive data rate chirp modulation technology allowing long-range communication combined with low-power consumption and a low-cost design. This is achieved, as we will see later, via spread spectrum multiple access techniques accommodating multiple users in one channel.

---

[1] SigFox. https://www.sigfox.com/en/sigfox-iot-technology-overview

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

[3] https://www.gsma.com/iot/narrow-band-internet-of-things-nb-iot/

[4] https://www.gsma.com/iot/long-term-evolution-machine-type-communication-lte-mtc-cat-m1/

[5] https://dash7-alliance.org/

[6] LoRa. https://www.lora-alliance.org/what-is-lora/technology

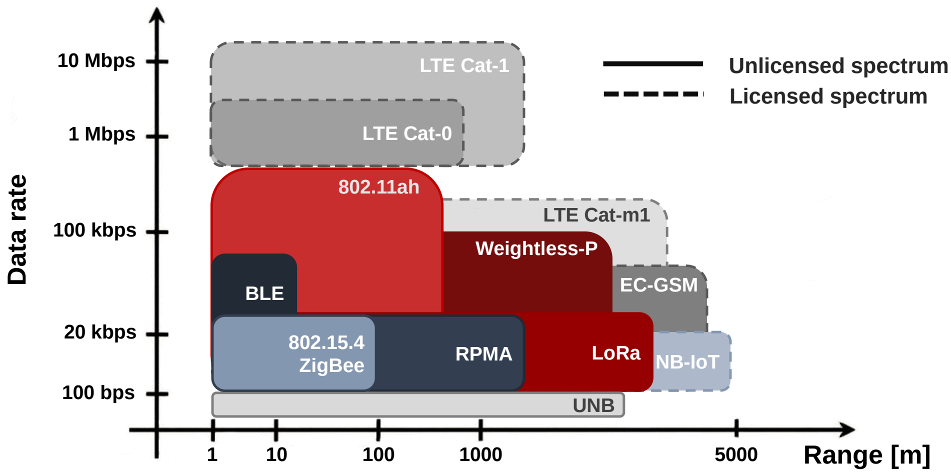


Figure 2.1: Range vs data rate comparison for different LPWAN communication protocols

Thanks to these unique characteristics we decided to focus on this technology by deploying an open LoRa network for the municipality of Trento and by using this communication standard to provide connectivity to the IoT devices presented in this thesis.

2.2 LoRa Modulation

LoRa is a sub-GHz modulation technology which has recently taken the interest of the scientific community for its very long-range (up to 15-20km) and high sensitivity. The physical layer was originally developed from Cycleo and then acquired by Semtech in 2012. It is based on Chirp Spread Spectrum (CSS) modulation technique that uses wideband linear frequency modulated chirp to encode information. Chirps are sinusoidal signals which frequency increase or decrease over time. The total power emitted by the radio is the same of a narrowband technology but on a much larger channel. The peak signal power is thus smaller, and it can be confused with noise at low powers.

Figure 2.2 show in a graphical way the difference between narrowband and spread spectrum signals. Thanks to these characteristics, the narrowband interference does not influence the transmission; narrowband technologies and LoRa can co-exist in the same portion of spectrum without compromises. Furthermore, with the spread spectrum transmission it is possible to decode also packets with negative SNR, thus it is possible to use LoRa also in very noisy environments. The excellent decoding capability allows LoRa to be used for very long-range transmission without exceeding the 25mW power limit for ISM band transmissions. Nonetheless, where a high floor noise level is expected, such as in an industrial environment, this technology is certainly appealing.

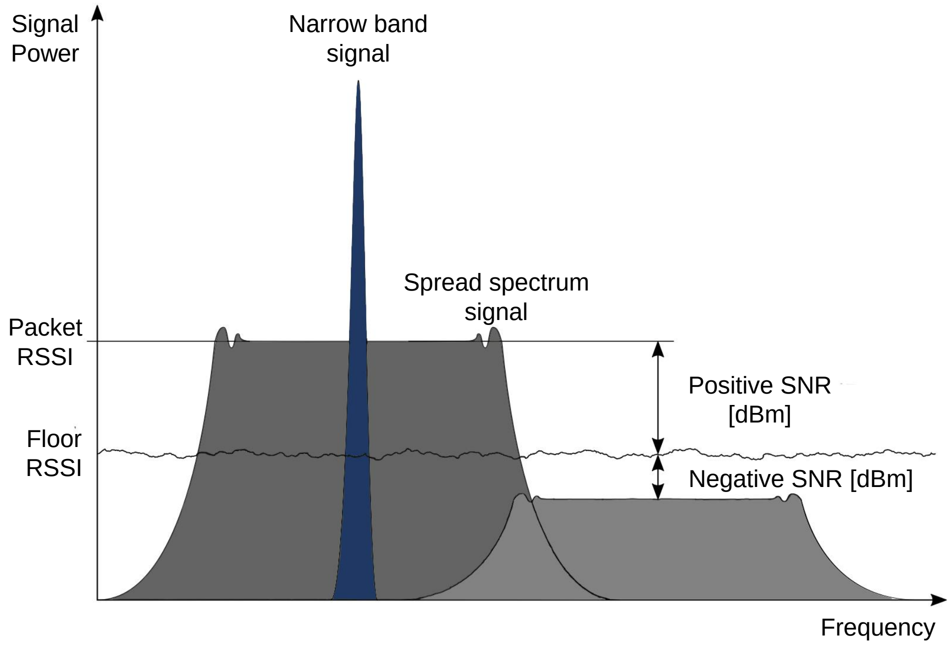


Figure 2.2: Difference between narrow band and spread spectrum signals.

As previously stated, the transmission is done by sending the payload symbols as chirps. By stretching the chirps in time or frequency it is possible to achieve different performance in term of throughput and achievable distance. Chirps can be adjusted, in time or frequency, by means of two parameters, respectively, Spreading Factor (SF) and Bandwidth (BW): the chirps stretching in time can be controlled with the SF while the channel width can be increased with the BW parameter. The SF can be set with values from 6 to 12 where 6 is the fastest configuration and 12 the slowest. The available bandwidths are 125 kHz, 250 kHz and 500 kHz. With slower chirps (in time), the message is less subject to interference. Increasing the BW makes the transmission of the message faster, but less robust to interference. It is worth to be noted that with respect to other technologies, where the transmission time is in the order of some ms, a single LoRa packet can require from 5 ms up to 12 s in the worst conditions. The graphical meaning of these two parameters – SF and BW – is presented in the spectrogram of figure 2.3.

An additional parameter in LoRa transmissions is the *Cyclic Redundancy Check* (CRC) which introduces the capability to understand if there have been errors in the message reception. Four levels are available and are indicated as the number of additional bits needed every 4 bits. With the addition of 1 or 2 bit it is possible to detect a single bit error; with 3 additional bits the error is also recoverable, while with 4 bits it is possible to recover an error and identify a second one.

By knowing the radio settings, it is possible to calculate the time needed to transmit one single symbol (equation (2.1)) which can be used to obtain the total transmission time after converting the packet length into symbols number with the equation (2.2).

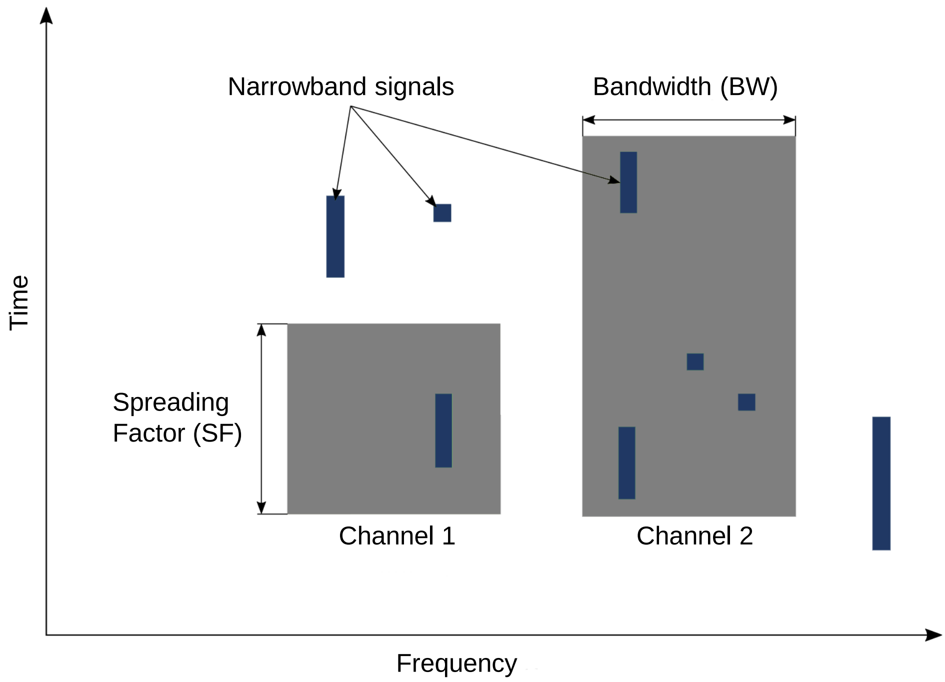


Figure 2.3: Schematic spectrogram that shows the meaning of bandwidth, spreading factor and channel.

$$t_s = \frac{2SF}{BW} \tag{2.1}$$

$$n_{pl} = max\left[ceil\left(\frac{8PL - 4SF + 28 + 16CR - 20IH}{4(SF - 2DE)}\right) * (CR + 4), 0\right] + 8 \tag{2.2}$$

where:

- n_{pl} is the number of symbols associated to the specified payload length with the specific radio configuration,
- PL is the number of bytes of the Payload (1 to 255),
- SF is the spreading factor (6 to 12),
- CR is the coding rate (1 corresponding to 4/5, 4 to 4/8),
- IH = 0 when the header is enabled, IH = 1 when no header is present (explicit or implicit),
- DE = 1 when LowDataRateOptimize is enable, DE = 0 otherwise. DE is an additional boolean setting that can be used or not to optimize the transmission when using high spreading factors (10 to 12).

Once calculated the symbol time and the number of symbols of the payload, we can calculate the total transmission time with equation (2.3).

$$t_{tx} = (n_{preamble} + n_{payload} + 4.5) * t_s \tag{2.3}$$

The packet length can be left implicit in the protocol or can be explicitly added inside the LoRa header for more flexibility. Notice that the header can have its own CRC. Moreover, with SF6 the header cannot be used (i.e. only implicit packet length can be used). Finally, in front of the header a preamble (typically 6 or 8 Bytes) is automatically added by the hardware before sending the message. The preamble helps the gateway to synchronize with the incoming wireless signal.

2.3 LoRaWAN MAC layer

LoRaWAN is the MAC layer that sit on top of LoRa physical layer previously presented. The specifications are intended for wireless battery-operated devices in regional, national or global networks. LoRaWAN target the key requirements of IoT such as: 1) Secure bi-directional communication; 2) Mobility and low-power capabilities.

2.3.1 End-Node Classes

LoRaWAN provides three different end-node classes to satisfy different needs: *Class A, B* and *C*.

In *Class A*, each up-link transmission is followed by two short down-link receiving windows in which the node waits for possible down streams from the gateway. Class A devices are characterized by the lowest power consumption. In this class, a down-link message sent to the end-node can be sent only after the network server receives an up-link transmission from the node. In this way, it can synchronize with the end-node action of opening the two receiving windows. In addition to the class A two receiving slots, *Class B* devices can schedule extra receiving windows at a periodic time slot. The gateway sends a time-synchronized beacon to the end-node device. This beacon allows the network server to sync with the end-node receiving windows. *Class-C* end-devices are almost continuously in receiving mode, waiting for down-link transmission from gateways. They only switch to transmission mode when the end-nodes need to transmit. This class of end-devices provide the lowest latency but also have the highest power consumption.

LoRaWAN also introduce a first layer of privacy and security by encrypting RF data frame using 3 different keys:

- **Network Session Key** (128-bit key) ensures security on network level
- **Application Session Key** (128-bit key) ensures end-to-end security on application level

- **Application Key** (128-bit key) ensures end-to-end security on application level (when in OTA mode)

Figure 2.4 presents the message encryption scheme

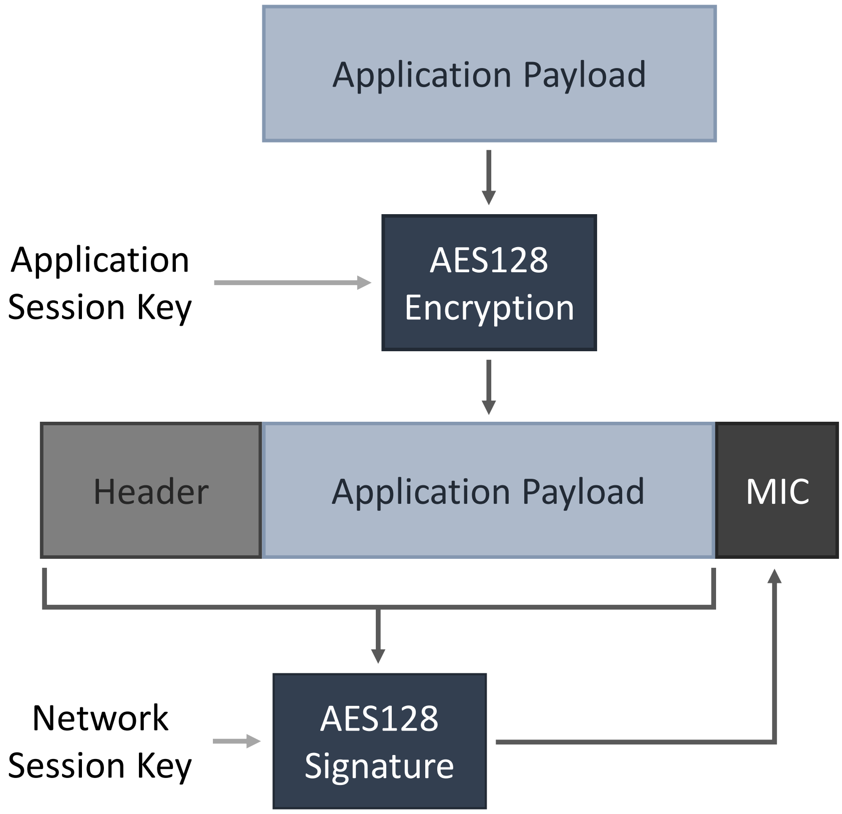


Figure 2.4: LoRaWAN data frame encryption scheme. The payload is first encrypted using the AppSKey. Then, the header plus the encrypted payload is signed using the NwkSKey. The signature is stored inside the *Message Integrity Code* (MIC) part of the data frame.

2.3.2 End-node device activation

In order to join a LoRaWAN network, an end-node must be activated either by Over-The-Air-Activation (OTAA) or Activation-by-Personalization (ABP). ABP end-nodes store directly the device address (DevAddr), Network Session key (NwkSKey) and Application Session Key (AppSKey). Therefore, ABP end-node devices skip the join procedure and assume the end-node devices have already joined the network. For OTA Activation, a join procedure must be followed before end nodes can exchange data with the network server. It is required by the end-nodes and the network server to keep the same following information: a globally Device Identifier (DevEUI), the Application Identifier (AppEUI), and the Application key (AppKey). These information are combined with the DevNonce from the end-node device and the AppNonce from the network server to derive the two session keys NwkSKey and AppSKey. The keys are then stored both by the end-node device and the network server. Those two keys will then be used to encrypt the messages interchanged between the end-node and the network server.

2.4 Related work

In [47] LoRa self-interference is studied by calculating the co-channel interference rejection for all combination of *Spreading Factors* (SF). Results show that a packet can successfully be received with the same SF and in the same channel if it is received with a margin 6 dB higher that its interferer, thanks to the so-called capture effect. To mitigate this problem, thus to enhance the battery life of end nodes (i.e. to avoid the retransmission of a packet due to external interferences), Adwait et. al. [48] have developed a custom LoRa gateway and a smart algorithm to coherently combine in the cloud received signals from multiple gateways to detect weak signals that are not decodable at any individual gateway demonstrating a gain of up to 3x in range and 4x in client battery-life. This improvement is also confirmed in [49] where two techniques to decrease the self-interference, namely directional antennas and multiple base station, are discussed. It is shown that by increasing the number of base stations, data extraction can be increased up to 56%, compared to 32% when directional antennas are used.

A study regarding LoRa physical layer and chirp modulation is presented in [50] and [51]. They number the advantages of using CSS modulation techniques and the possibility of adopting upper layer solution from other technologies. They argue that the range is shorter than other UNB (Ultra-NarrowBand) solutions but more robust against interference. LoRa technology is targeted to resource constrained devices enabling low-power long-range real-time communication of data and analytics that can be used to enhance efficiency and productivity.

LoRa outdoor coverage is studied by different research group. In [52] they reached a range transmission of 3 Km for SF12 and 2.3 Km for SF7. In [53] authors have shown that a 100 km2 city can be easy covered with 30 gateways, half the number of base stations that are used for cellular network for covering the same area. Another interesting study were conducted in the city of Oulu, Finland, by Petajajarvi et. al. [54]. Authors have tested the performance of a LoRa channel both on ground and on water. The node was configured to transmit using SF12 with the maximum transmission power for the 868 ISM band, 14 dBm (25 mW). Results on ground shows that 80% of the packets can be correctly received up to 5 Km away, while between 5 to 10 Km the successful rate goes down to 60%. On water, almost 30 Km communication range was reached, with about 70% of the packets delivered successfully at the distances below 15 km. The paper also presents a channel attenuation model derived from the measurement data that can be used during the infrastructure design.

LoRa indoor coverage is still being investigated, and the number of published works discussing this topic are limited. Petajajarvi et. al. [55] investigate LoRa indoor performance inside the main campus of the University of Oulu, Finland. The obtained results indicate that with the largest spreading factor, SF12, and 14 dBm transmit power, the whole campus area (182000 m2) can be covered by a single base station, with an average packet delivery ration equal to 96%. In [56] authors characterize the performance in terms of packet loss, indoor coverage, received signal strength, power consumption of end devices and delays due to duty cycle.

In recent years, academia started to evaluate the feasibility of using LPWAN, like LoRa, in short range industrial applications motivated by the advent of Industry 4.0 [1]. As highlighted by different studies [57]–[62] LoRa can represent a feasible solution to implement wireless communication in an industrial environment. Its intrinsic robustness and high sensitivity, designed to achieve very long outdoor transmissions, make it very precious also in short-range industrial environments characterized by high electromagnetic noise where robust transmissions and high radio sensitivity are required. In [57] LoRa and the upper MAC layer, LoRaWAN, have been evaluated to assess their applicability to industrial wireless networks. By implementing a Time Slotted Channel Hopping (TSCH) channel access method, authors showed that it is possible to implement an industrial WSN for soft real-time applications using LoRa links. Performance evaluation in noisy environment is discussed by Angrisani et. al. in [58]. An interesting case study is presented in [60]. Here a LoRaWAN network is implemented to monitor the movement of a large number of trolleys across a flower auction warehouse. Results show that with a single gateway it is possible to cover an area bigger than the warehouse (34000m2) with a number of nodes as high as 6000. Brunelli et. al. [59] presents the design of system to monitor the harsh environment of a Shallow Geothermal Systems (SGS) using a LoRa. This allow the transmission of data wirelessly in rural areas where conventional wireless connections (e.g. WiFi, GSM) are not guaranteed and energy availability is limited. In [62] is proposed a WSN for data center temperature monitoring. The network was able to operate for six months without battery replacement, showing a 300x better energy efficiency compared to the previously deployed WSN, ensuring also a coverage area 7 time wider.

2.5 Trento LPWAN network

Fueled by the Trento Smart City Initiative[7] and to provide connectivity for IoT devices, we have developed a management framework implementing LoRaWAN connectivity for the municipality of Trento. The overall architecture is presented in Figure 2.5. It allows us to manage the activation of multiple devices and provides data context, storage, visualization, and access control over the web. Thanks to the deployment of a full channel gateway on top of the building where our laboratory is located, we are able to provide LoRaWAN connectivity to large portions of the municipality of Trento.

LoRa support is granted by the excellent open source Chirpstack[8] LoRaWAN server, while data visualization and storage are managed by ThingsBoard[9] plat-

[7]https://smartcities.ieee.org/
[8]https://www.chirpstack.io/
[9]https://thingsboard.io/

form. The integration between these two main components is implemented by means of a secure MQTT[10] connection, that also allow easy integration with third parties application.

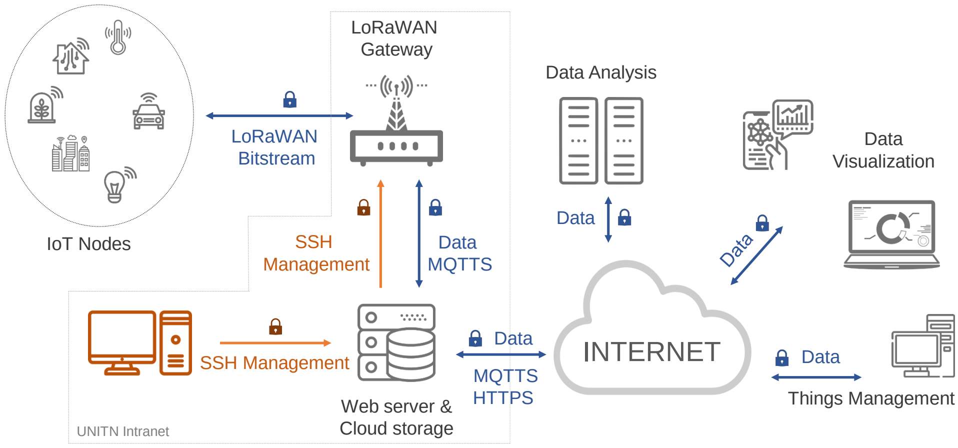


Figure 2.5: Implemented LoRaWAN network architecture

2.5.1 Coverage Experiment

The coverage experiments are conducted around the municipality of Trento at several specific locations, as shown in Figure 2.6. We have randomly chosen a set of points which are expected to cover the whole municipality. The distance from each location is summarized in Table 2.1.

The main parameters that can affect LoRaWAN range are: TX power, SF, CRC, and BW. The transmission power determines the strength of the transmitted signal. Therefore, it directly impacts the RSSI at the gateway side and the reliability of the communication link. Spreading factor is a key parameter that influence the robustness of the communication link to external noises, while higher coding rate can offer protection against bursts of interference. Finally, a higher bandwidth can integrate additional noise that decreases the protocol's reliability.

Since the goal of these experiment is to test LoRa range, we have decided to keep all the parameters fixed but the transmission power. Table 2.2 presents the four configuration tested. As we can notice in Figure 2.7, that presents the experimental results, the reliability generally increases with the transmission

[10] http://mqtt.org/

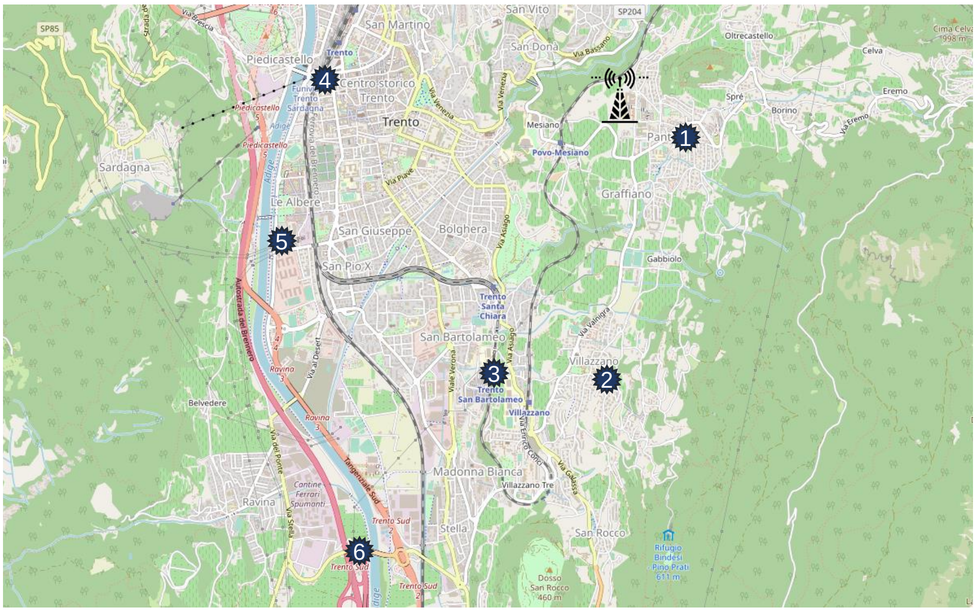


Figure 2.6: Map showing the selected test points for the coverage experiment.

Table 2.1: Coverage test points distance from gateway

Index	**Location**	**Distance from the Gateway [Km]**
1	Vittorio Manci Square	0.5
2	Villazzano	2.5
3	San Bartolameo campus	2.6
4	San Severino Parking	3.0
5	Monte Baldo Parking	3.1
6	Brennero Highway Exit	4.5

power. There are no packets received at Villazzano and a decrease in the number of packets received at San Bartolameo due to the fact that a hill lies in the line of sight.

From the results we can confirm the incredible long-range performances of a LoRa link, highlighting the fact that we can cover an extensive area with just a couple of gateways. In fact, to cover the whole municipality of Trento, 2 or 3 gateways would be enough.

This confirms the effectiveness of the technology for providing connectivity for the future IoT devices.

Table 2.2: LoRaWAN Testing Configurations

Config #	Frequency [MHz]	Bandwidth [kHz]	TransmissionPower [dBm]	**Spreading Factor**	**Payload [Bytes]**
1	868	125	5	12	49
2	868	125	8	12	49
3	868	125	11	12	49
4	868	125	14	12	49

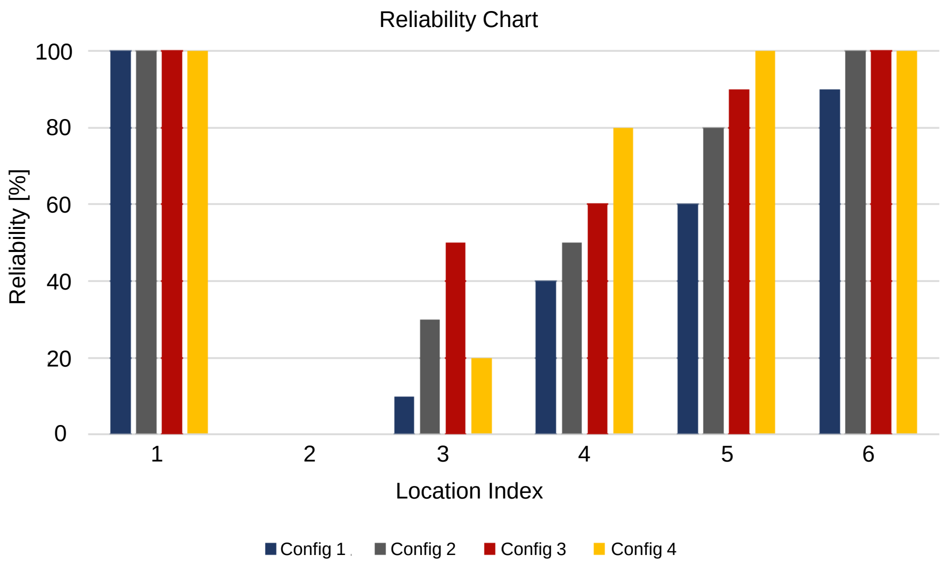


Figure 2.7: Coverage experiment result

2.6 LoRa Networks Enabling Industrial IoT Applications

In the vision of Industry 4.0 there is a growing interest in the study of *Industrial-IoT* (I-IoT), as industrial environments introduce additional challenges for wireless communication and wireless sensors. LPWAN are seen as promising technologies for implementing monitoring solutions in all those environments where it is required to cover large areas with low bandwidth requirements.

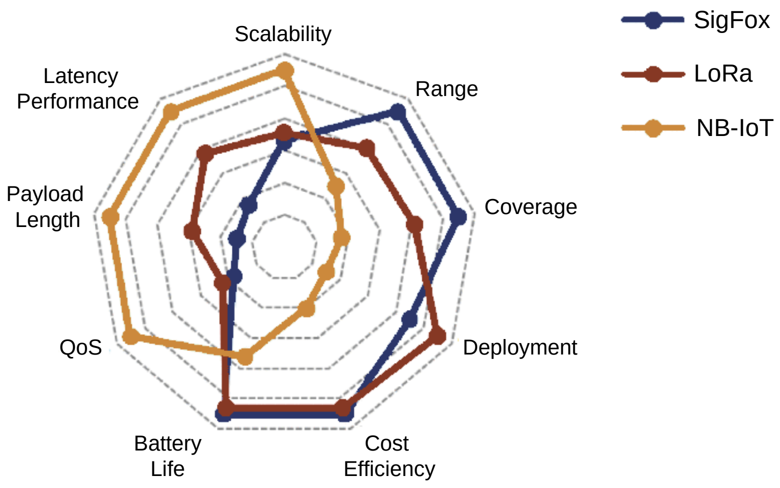


Figure 2.8: Main characteristic comparison of the three presented LPWAN technologies

2.6.1 LPWAN Technologies for I-IoT Deployment

To select a suitable technology for an I-IoT application many factors should be considered, including quality of service, coverage, latency, battery life, scalability and deployment cost. I-IoT applications have specific requirements such as long-range communication, very low-energy consumption, and cost effectiveness. Among all, the three leading LPWAN technologies are SigFox, LoRa, and NB-IoT. Figure 2.8 presents a comparison of the main characteristics of the different LPWANs.

QoS

SigFox and LoRa employ license-free sub-GHz bands and asynchronous communication. They can efficiently mitigate interference and fading/multipath. However, they do not provide a level of quality of service. NB-IoT employs licensed spectrum its time slotted synchronous protocol is optimal for QoS. However, this advantage of QoS is at the expense of cost [63]. Due to the tradeoff between cost and QoS, NB-IoT should be preferred for applications that need a guaranteed level of quality of service, while the applications that do not have this constraint should choose LoRa or SigFox.

Coverage

NB-IoT has the lowest range and coverage capability. The deployment of NB-IoT is limited to 4G/LTE base stations. Thus, it is not suitable for those areas that do not have 4G coverage (i.e. noisy industrial environment or deep-indoor scenarios). One significant advantage of LoRa and SigFox is that a whole city could be covered by one gateway or base station, lowering installation and maintenance costs.

Battery life & latency

LPWAN end-devices are in sleep mode the most of time in order to reduce as much as possible the amount of consumed energy. However, the NB-IoT end-nodes consume additional energy because of infrequent but regular synchronization, and OFDM or FDMA require more peak current for the linear transmitter [64]. On the other hand, these demands offer NB-IoT the advantage of low latency and high data rate. Therefore, for those applications that are insensitive to the latency and do not have large amounts data to send, LoRa is the best choice. For applications that require low latency and high data rate, NB-IoT is the better choice.

Scalability & Payload length

Supporting thousands of end-devices is one of the key features for all the 3 technologies compared. However, NB-IoT offers the advantage of very higher scalability than SigFox and LoRa (i.e. 100k devices for single base station compared to around 50k for SigFox and LoRa). NB-IoT offers also the advantage of maximum payload length, allowing up to 1600 bytes per packet. LoRa allows sending only a maximum of 243 bytes while SigFox stops at 12 bytes per packet. It is clear that for high data rate application, NB-IoT should be preferred.

Cost

There are different cost aspects that need to be taken into consideration, such as spectrum cost, network cost, device cost, and deployment cost. Table 2.3 presents

Table 2.3: Cost comparison of SigFox, LoRa, and NB-IoT

	Spectrum Cost	**Deployment Cost**	**End-Node Cost**
SigFox	Free	> 4000€/Base Station	< 5€
LoRa	Free	< 1500€/Base Station	< 5€
NB-IoT	> 500 M€/MHz	> 15000€/Base Station	> 20€

the comparison for the different technologies. It is clear that LoRa has a huge advantage in relation to cost.

In our case, LoRa was selected for its intrinsic robustness and high sensitivity – mainly designed to achieve very long outdoor distances – making it very appealing also for industrial environment where the long-range capability can be traded for a higher immunity to external interference and superior sensitivity. Moreover, LoRa is easily integrable with existing infrastructures, has a low deployment cost and suitable scalability.

2.6.2 LoRa Radio Parameters

The signal transmitted by the LoRa radio can be tuned by means of three main parameters: SF, BW and CRC. Table 2.4 summarize the reduced set of values for the different parameters used during the analysis for addressing the disturb introduced by external electromagnetic noise that characterize an industrial environment.

Table 2.4: Parameters value tested during the analysis of the noise floor level

Parameter	Value
Spreading Factor	6, 7, 9, 12
Bandwidth [kHz]	125, 250, 500
Code Rate (CRC)	4/5, 4/6, 4/7

2.6.3 Noise analysis

Before starting investigating the effect of the noise generated by an industrial machine to a wireless communication, a first analysis of the noise floor is conducted. These tests have been done inside a harsh industrial environment with a number of laser cutting machines. The main goal of this test is to find which points around and inside the machine can be considered critical and which ones instead are less sensitive to the environment characteristics. Indeed, high floor noise negatively impacts on the packet reception reducing the SNR, thus the points with high noise can be considered as the worst case. On the contrary, the lowest floor noise points are taken as the best case condition.

In order to find these points, a grid of equidistant points is deployed in the area of interest and used as spatial reference for the noise floor measurement. Inside the machine, as the points cannot follow the grid scheme, they were placed where they do not interfere with the machine operations. The full map of points is represented in figure 2.9. Some other points are added to the grid where there is interest to put a sensing node. The measures are taken, for practical reasons, at 1.2m from the floor level, with the exception of the internal points and the master position. The master is put on top of the machine chassis in order to have similar shielding effects for every network's node.

Measurements have been acquired using two nodes, one acting as master while the other as slave, exploiting the noise floor measurements capability of the radio used. The data sent from the slave node consist in the average and standard deviation of 10 successive noise floor measurements for each point of the grid. To this end, a randomized approach is used to choose the points order: this approach reduces the effect of uncontrollable time varying variables – like temperature or humidity – on the final data analysis. Tests have been performed once with the machine at work and once with the machine at rest. The rest condition cannot be tested during the working shift thus the test is performed before the beginning of every activity, which means that every machine in the area is powered off (not only the machine under testing). The results of the campaign are reported in

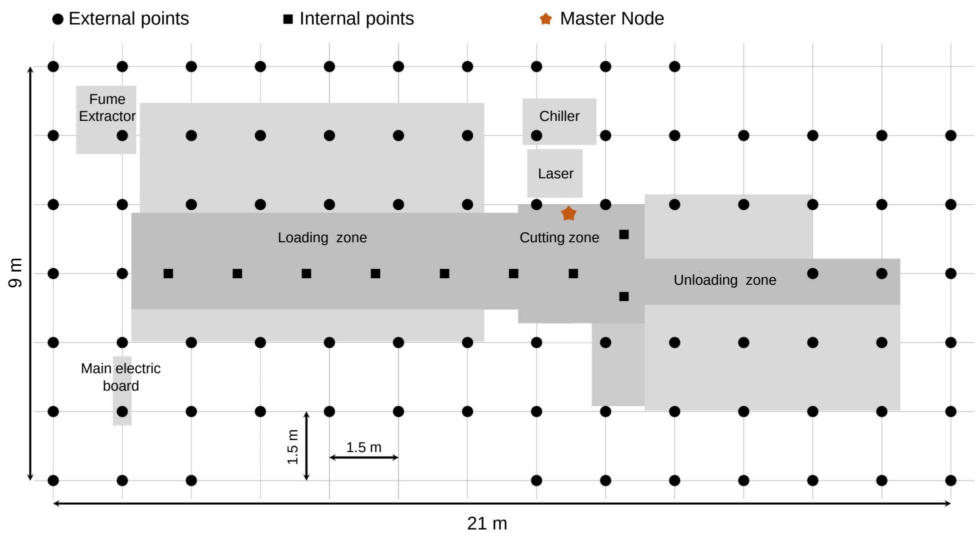


Figure 2.9: Schematic top view of the machine with the grid of points used for the floor noise measurements. The internal points are at floor level while the external points are at 1.2m from the floor. The star indicates the master's position on top of the machine.

Figure 2.11. It can be noted that the noise is very low (-90 dBm) with respect to the one typical of the 2.4 GHz band (-50/-60 dBm). Reasonably, the effect of other networks in this band is not significant and the floor noise is lower than the one measurable in the 2.4 GHz band.

From Figure 2.11, two unexpected peaks are visible on the bottom left of the map (at (3m,1.5m) and (6m,1.5m)) with no apparent motivation.

The map reports only the machine related components and does not consider the external equipment in the area, which depends on the specific application. As a matter of fact, in the area of the abnormalities there is a technical station, which may generate electromagnetic noise. To better investigate those noise spikes, additional tests are accomplished to characterize only that area. The two tests have been performed both with the machine at rest and only the technical station status is changed. The results of this additional tests confirm the influences of the hardware as visible in Figure 2.10. This phenomenon confirms that the network may still be subjected to external interference that are application specific and not related to the product itself.

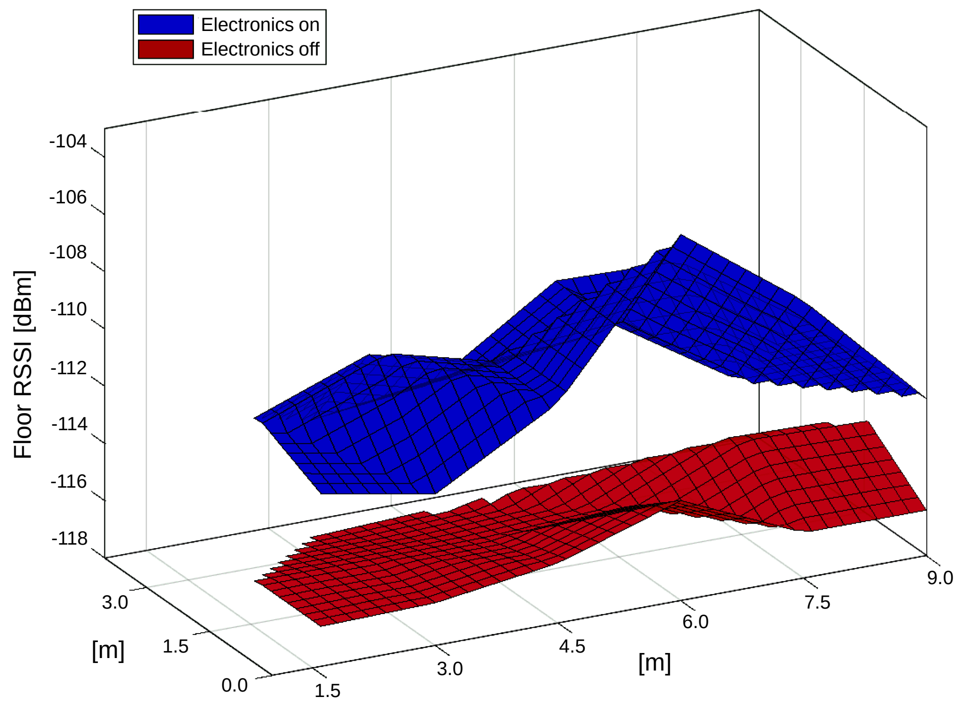
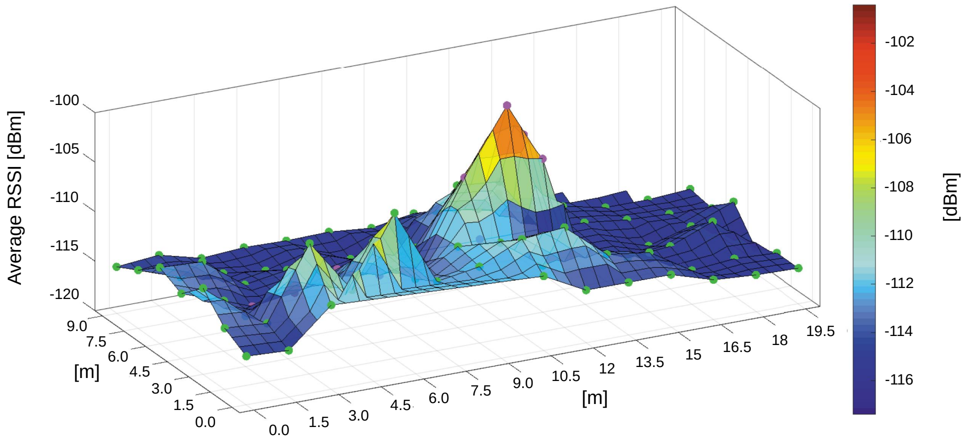


Figure 2.10: Noise floor around the technical station area. It is clearly visible the difference between the two cases: one performed with everything shut off and one with only the technical station on (the machine is still off).

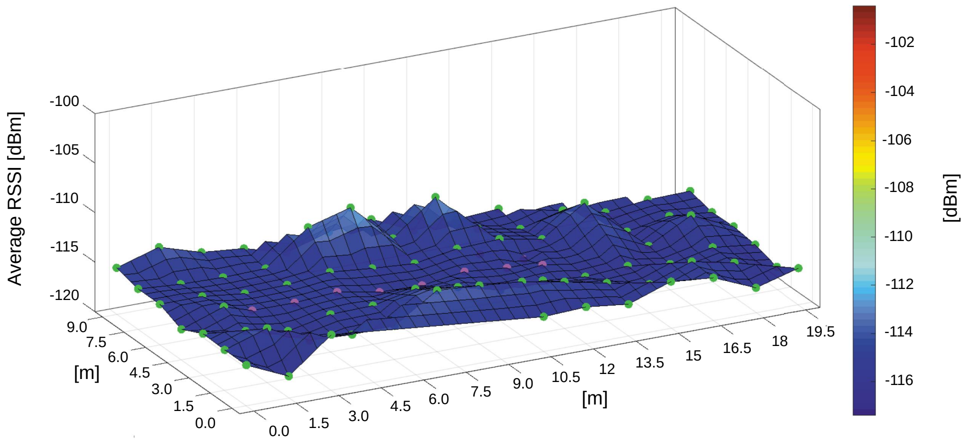
2.6.4 Radio configuration analysis

Starting from the noise floor analysis, presented in previous section, we can test the performance of the radio link. It can be argued that, for this kind of analysis, the Packet Error Rate (PER) is the most valuable information that must be estimated before designing the protocol. Along with the ISM band duty cycle limitation, the PER is fundamental to choose the right algorithms and message recovery policies in order to optimize the WSN performance. In addition to this analysis, we have analyzed the packet signal strength to better understand hardware independent factors such as the node location effect. This because the different positioning can influence the link quality due to multiple effects like shielding, refraction and reflection of the electromagnetic radiation.

The points chosen for the link analysis, shown in Figure 2.12, are the one in which a node will be positioned. Even if the analysis is not specifically intended for the machine under study but for a general evaluation of the performance in industrial



(a)



(b)

Figure 2.11: Surface plots of the floors noise with different machine conditions: machine cutting (2.11a) and machine off (2.11b). The cutting head is the worst point and elsewhere the noise is not changing significantly with respect to the machine state. For a clearer representation, the surfaces in the plots are the results of an interpolation.

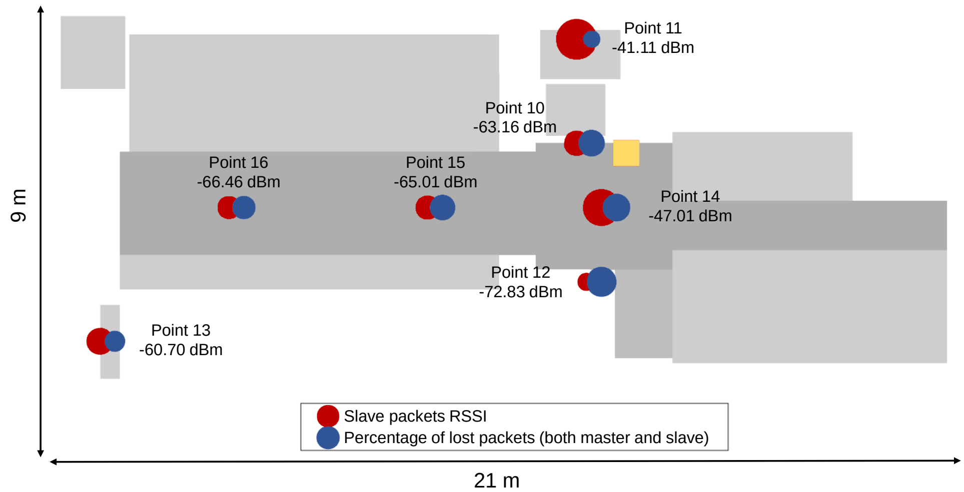


Figure 2.12: The image presents qualitatively which are the best and the worst points. Red circles represent the packet RSSI of the point while blue circles the percentage of lost packets. The yellow square represents the master's position.

environment, by using the actual nodes locations it is possible to evaluate also the specific problems that may arise in the specific product. Furthermore, the nodes may be affected by different factors (e.g. obstacles or noise floor) thus they can be used to assess the effect of different uncontrollable factors (e.g. chassis shielding effect and electromagnetic noise).

Taking into consideration the reduced set of parameters from Table 2.4 and the number of points (as per Figure 2.12), the number of combinations tested are:

$$N_{comb} = SF_{lvls} * BW_{lvls} * CRC_{lvls} * P_{lvls} = 252 \tag{2.4}$$

To measure the probability of losing a message, the master sends a burst of 20 numbered messages (200 ms apart one from the other) to the slave which counts the number of lost packets. Meanwhile the slave measures also the packet and floor RSSIs and averages them over the whole message burst. After the last message is sent, the master asks the slave to send back a new burst to evaluate the symmetry of the link.

With a 20 messages burst of 64 Byte in each direction the total number of messages for evaluating the performance is:

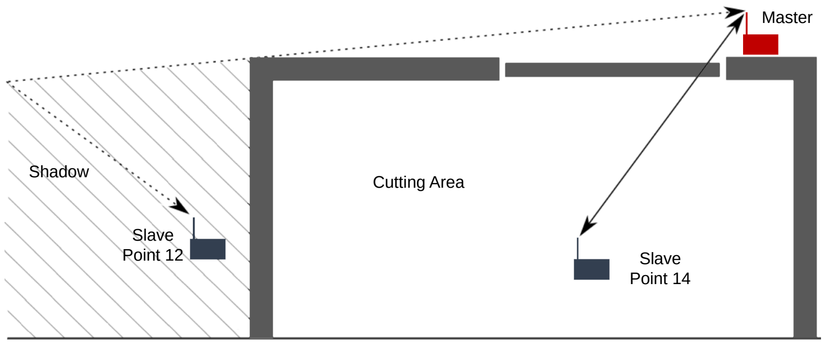


Figure 2.13: Schematic representation of the position of the nodes with the shadowing and shielding effects. P14 is affected by the highest level of noise, but P12 exhibit a worse behavior.

$$N_{messages} = 20 * 2 * N_{comb} = 10080 \tag{2.5}$$

Using this parameters, the time necessary for each combination is between the 10 seconds of the configuration *SF6 - BW500 - CRC4/5* and the 2 minutes of *SF12 - BW125 - CRC4/7*, being the SF factor the parameter that influence the most. Figure 2.12 qualitatively shows the result of this analysis while Figure 2.14 shows more in details both master and slave packet RSSI for each tested point. It is clearly visible that *P11* presents the best performance while *P12* is the worst even though *P14* is placed where we have the higher noise floor (as per figure 2.11a). This can be justified by the fact that *P11* has a clear line of sight with the master location, while *P14* is shaded by the shielding of the cutting area, as showed in Figure 2.13. With regards to the error probability, Figure 2.15 shows how the error probability is influenced by the different factors. It is clear that the different levels of CRC do not influence much the error probability. On the contrary, spreading factor and bandwidth have major influence on the process.

2.6.5 Packet Error Rate analysis

With the acquired data it is now possible to restrict the set of combinations to only a few interesting configurations and test them deeply. Firstly, the worst (SF6 BW500 CRC4/5) and the best (SF7 BW250 CRC4/6) configurations are selected.

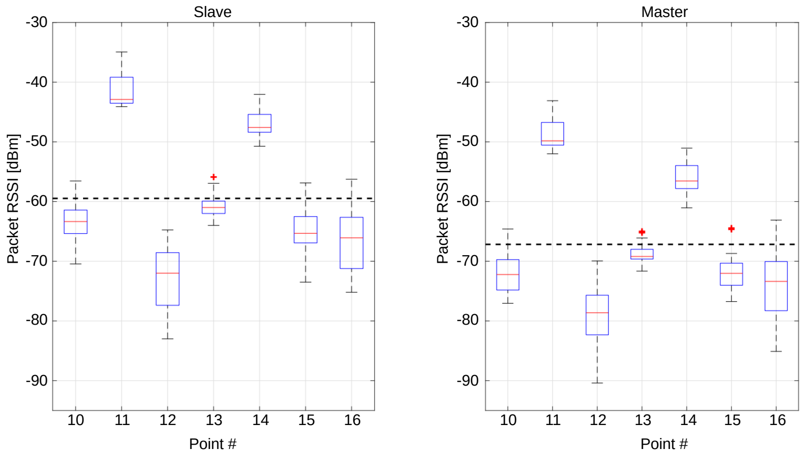


Figure 2.14: Boxplot of the packets RSSI for the slave on the left and for the master on the right.

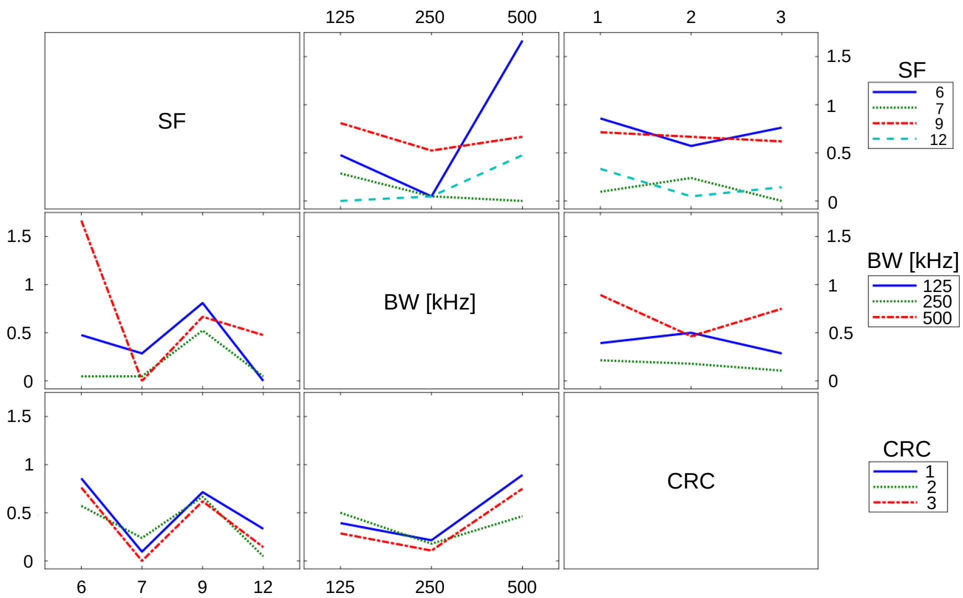


Figure 2.15: Lost messages percentage for each factor. Notice that the CRC level is not affecting significantly the performance.

In addition to those, two more combinations are chosen:

- **SF7 BW250 CRC4/5:** being the CRC not influencing much the performance, CRC4/5 could be a better choice thanks to a lower overhead. Furthermore, from the 2.15 plot, the level presents better performances in conjunction with SF7. Even if CRC4/7 further increase the performance, the additional overhead related makes this configuration less appealing.
- **SF7 BW500 CRC4/5:** this is the fastest configuration with SF7. Furthermore, from Figure 2.15 can be concluded that the optimal bandwidth for SF7 is BW500 (conversely to the other spreading factors).

The points for the link analysis have been chosen based on the previous experiments:

- **P11** - best condition;
- **P12** - worst condition;
- **P14** - worst floor noise condition;

As expected, *P14* presents better performance than *P12* even if it is inside the machine casing in a noisy area. Reasonably, the shadowing effect to which *P12* is subjected is more critical than the reduced shielding effect of *P14*. The results of PER analysis are resumed in Figure 2.16. For this reason the second combination, *SF7 – BW500 – CRC 4/5*, has been chosen as the best working mode.

2.7 Conclusion

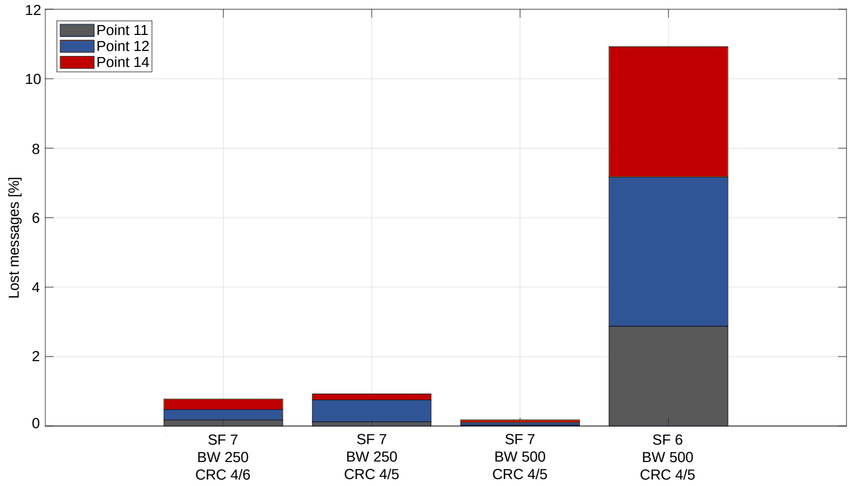
Several new options are emerging to communicate wirelessly over long distances. One promising option is chirp spread-spectrum long-range radio technologies, such as LoRa/LoRaWAN. LoRa has given rise to long-range sensor data backhaul options at kilometer scale such as OpenChirp [28]. LoRa ICs are commercially available, inexpensive, and offer long range (i.e., kilometers) at low power (i.e., tens of mW).

The reliability of the protocol is evaluated with a coverage experiment around

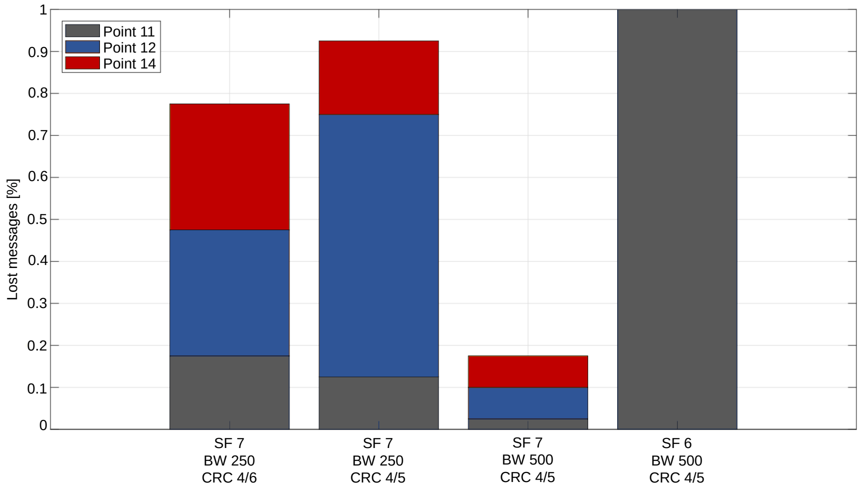
Trento City. It has been proven that a single LoRa base station can provide connectivity over large areas, lowering installation costs. This easy to deploy extensible receiver infrastructure can be expanded using simple, publish/subscribe data management (e.g., MQTT) making the data from IoT nodes easily available online.

The ability to communicate kilometers away at low power creates the opportunity for more devices to be deployed to more environments than is possible using legacy backhauls. For example, 4G/LTE is expensive per byte [65], Bluetooth/BLE is range-limited [66], [67], and WiFi requires deploying many base stations for wide-area coverage with an energy requirement unfeasible for energy-harvesting applications.

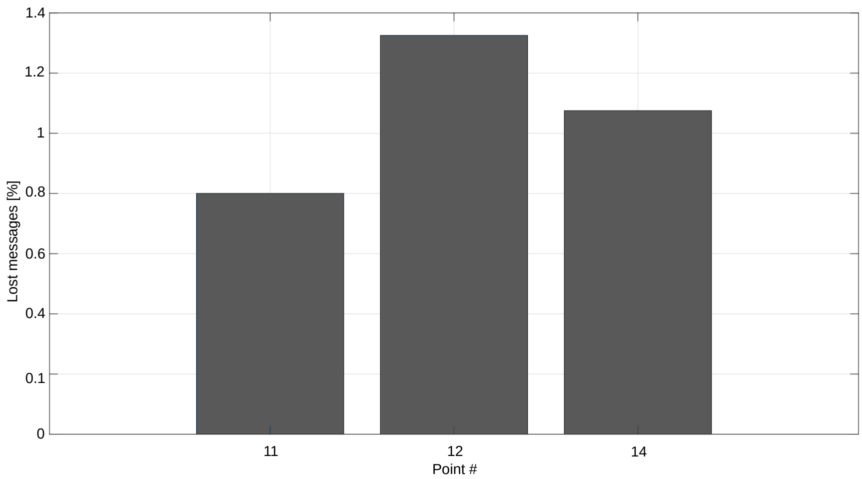
Even if LoRa was not intended for the industrial environment, its use has been proven successful even when applied in harsh industrial environment. Additional testing could be interesting to better characterize its performance for industrial applications in different ISM bands.



(a)



(b)



(c)

Figure 2.16: Main configurations error probability. Graph (2.16b) shows a zoomed view of (2.16a). In (2.16c) the global error probability in each point for the main configurations.

Chapter 3

Non-Intrusive Smart Metering

3.1 Introduction

One of the many challenges that smart cities are currently facing is the reduction of energy consumption [68]. Classical electrical distribution systems have been used for decades to transport electrical energy. Usually being generated at a central power plant and then delivered to end-users via long-established, uni-directional transmission and distribution systems. These systems have served us well, in many cases for more than a hundred years. Nowadays, along with the increase of the importance of renewable electricity generation, intelligent systems are needed in the electricity market.

The usual approach for sensing in buildings provide the use of wired sensors added when the building was constructed or renovated. Those sensors are typically cost effective and reliable, when incorporated into an existing construction project. However, adding new hardwired sensors at a later stage is often infeasible. One solution is battery powered sensors, but adds additional and unwanted maintenance requirements for battery charging or replacement.

The solution for providing better sensors, ready for deployment at scale, even in existing infrastructures, is changing from wired, expensive to install or, fixed-lifetime battery-based sensors, to energy-harvesting sensors with perpetual lifetime. This change, however, raises many technical challenges as the quantity of energy provided by harvesting solutions are often highly variable, both spatially and temporally. Two are the main rising challenges: *energy intermittency* (i.e.

energy is not always available) and *energy unpredictability* (i.e. we do not know when the energy will be available).

3.1.1 The Energy monitoring challenge

Electricity is one of the most flexible and complicated power sources that are available. Unlike most other energy sources, such as oil or coal, electricity cannot currently be stored in large scale volumes in a cost-effective way. This imposes some tight boundaries in managing and producing energy. The infrastructure must adapt in real-time to different load scenarios with the ability to face the dramatic increase in energy demand every time during the day. This means that power stations, also referred as power plants, must be kept ON all the time; also when energy demand is lower than the production capability leading leads to resource wastage and pollution generation.

The dysfunctionality of the current energy measurement system has been recognized for many years. More than thirty years ago, Kempton and Montgomery (1982) described the paradox of consumption without meaningful information as *"a store without prices on individual items, which presents only one total bill at the cash register, where shoppers have to estimate item prices by themselves"*.

To foster the development of advanced strategies for the production, distribution and consumption of energy, and also to better exploit the availability of renewable energy, in recent years, researcher effort has moved on the development of a new generation of smart meters. These devices integrate advanced algorithms for power analysis tailored for tracking and monitoring single appliances, starting from the aggregate power consumption of a whole building.

The ability to track single appliances contribution, starting from the aggregate power consumption acquired by a single power meter, takes the name of Non-Intrusive Load Monitoring, NILM. This analysis, introduced in the early 90s by George hart [69], can represent the turn key in the energy sector. It permits to monitor a high number of appliances with just one power meter.

By knowing the contribution of the single appliance, end-users can identify those more power hungry. This can motivate the replacement of those appliances with new and more efficient ones, or to adapt their consumption behavior accordingly.

The same data can also be exploited by the utility company that can implement a new range of services. *Demand Side Management* (DSM) to influence customer use of electricity, encouraging to consume less power during peak times, to shift energy use to off-peak hours, to flatten the demand curve or to follow the generation pattern of renewable sources; *load forecasting* to predict the energy demand in order to balance the grid to achieve an optimal utilization of electrical energy.

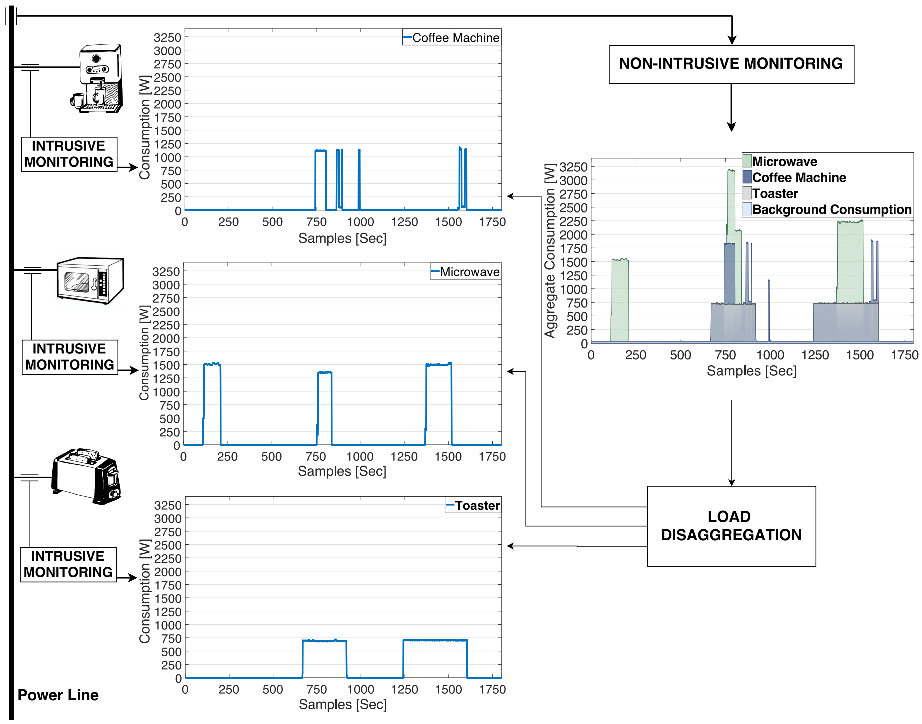


Figure 3.1: Intrusive Load Monitor requires a sensor for each appliance we want to monitor. Contrarily, Non-Intrusive Load Monitoring can achieve a comparable result, using just one sensor. Source [70]

3.2 Behind Non-Intrusive Load Monitoring

In the last decades, Load Monitoring has been an active research area, studied by researchers who investigated algorithms that try to discern what electrical loads (i.e., appliances) are running within a physical area.

Two different approaches can be exploited for monitoring energy consumption:

the Intrusive Load Monitoring (ILM) approach, which use one sensor for each appliance, and the Non-Intrusive Load Monitoring (NILM) approach which aims at estimating the load consumption from a unique overall current and voltage measurement. Figure 3.1 presents the difference between the two approaches.

The idea was born in the late 80's when George Hart [69], while at the Massachusetts Institute of Technology (MIT), proposed using changes in power consumption, as measured at the utility meter, to track the operation of individual appliances automatically. Although the proposed approach was appealing, during the subsequent 15 years, NILM studies were limited.

Recently, researchers renewed the interest as a result of decreasing hardware costs and expanding connectivity infrastructure. Stimulated also by the governments around the world that have started to put in place action to reduce the global electricity consumption[1].

3.2.1 How it works

NILM is the disaggregation of individual appliances consumption from the total consumption, using one, or few more, point of measurement. Electrical loads usually exhibit unique energy consumption patterns as presented in figure 3.2. This pattern is usually referred as *appliance signatures* and enables the disaggregation algorithms to recognize appliances operations from the aggregated load measurement. NILM process basically involve three stages [30], [71]: 1)Data Acquisition, 2)Features Extraction and 3)Appliance Classification.

Data acquisition is the process of acquiring the aggregated load consumption, sampled at appropriate rate for the purpose of appliance recognition. After data acquisition, the next step is to process the raw data previously acquired, in order to extract the features that characterize the single appliance contribution. Finally, the extracted features are used by pattern recognition algorithms to identify the specific states of the appliances from the aggregated measurement.

[1] https://eur-lex.europa.eu/eli/dir/2018/2002/oj

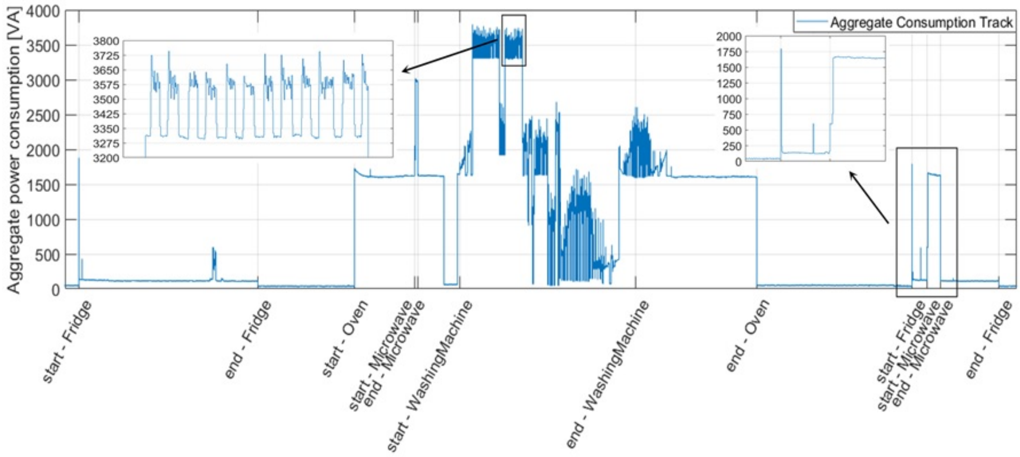


Figure 3.2: Different appliances pattern. Source [72]

3.3 Related work

To support recent advancements in the field of energy distribution and management, growing attention from both commercial and research group is begin focused on the development of new solutions for energy monitoring and tracking.

In response to both environmental concerns and financial incentives, prosumer are expressing a growing interest in smart devices to intelligently reduce the energy consumption, since, as highlighted by different studies [73]–[76], the first step for energy minimization is understanding where the energy goes. To meet this growing demand for greater visibility into energy consumption, researchers have designed a variety of energy metering solutions. From the most basic, plug-load meters, to circuit/panel meters capable of detecting different load starting from the aggregate consumption. This is also confirmed by utilities companies that are encouraging efficient behaviors, by means of incentive [77]–[82], to reduce or delay the need to build or operate high-cost peaker plants.

A range of metering solution is presented in Table 3.1, along with the 3 implementations proposed in thesis.

Table 3.1: Comparison of various power/smart meters. Last 3 design presented in this table are those discussed in the next sections. It worth to note that the last design, presented in section 3.6, is the only smart meter with integrated load detection and wireless connectivity for data streaming relying only on the harvested energy from the same load under monitoring

Metering device	Type	Current Sensing	Is wireless	Harvest Energy	On-Site load monitoring
WeMos Insight [83]	Plug Load	Shunt resistor	Yes	No	No
Stick-on [84]	Circuit Level	Piezo-electromagnetic	Yes	Yes	No
PowerBlade [85]	Plug Load	Magnetometer	Yes	No	No
TED [86]	Circuit Level	Split-Core CT	No	No	No
Monjolo [87]	Plug Load/Circuit Level	Split-Core CT	Yes	Yes	No
Gemini [88]	Plug Load/Circuit Level	Split-Core CT	Yes	Yes	No
Clamp-and-Forget [89]	Plug Load/Circuit Level	Split-CoreCT	Yes	Yes	No
NIWEM [90]	Circuit Level	Split-CoreCT	Yes	Yes	No
LoRa Meter [91]	Circuit Level	N/A	Yes	No	No
Panoramic Power PAN-10 [92]	Circuit Level	Split-Core CT	Yes	Yes	No
INTENS'O [93]	Circuit Level	Split-Core CT	Yes	No	No
Neurio [94]	Circuit Level	Split-Core CT	Yes	No	No
e-meter with µDisagg [75]	Circuit Level	Split-Core CT	No	No	Yes
Cost-Effective Smart Meter [72]	Circuit Level	Split-Core CT	Yes	No	Yes
Low-Cost autonomous Power Meter [95]	Plug Load/Circuit Level	Split-Core CT	Yes	Yes	No
Autonomous Smart Meter with load recognition	Circuit Level	Split-Core CT	Yes	Yes	Yes

3.3.1 Plug-Load Meters

Plug-load meters [83], [85], [96]–[98] are the first introduced solution for monitoring energy consumption. They can provide accurate power measurements of individual loads and appliances. The main drawback of these solution is that usually attaching a plug load meter is difficult or impossible, like in the case of loads that are hard to move (e.g. kitchen appliances) or build in (e.g. lighting). Another common issue with plug load meters is that they are typically active, also when the metered load is off, increasing metering energy overhead. Finally the most obvious drawback; the need for one sensor for each device we want to monitor.

3.3.2 Whole-Building and Circuit Level Meters

A different approach is the one followed by whole-building, or circuit level, metering solution [86], [99]–[101]. In this case, a single meter monitors the power draw from a set of loads or appliances. Panel and circuit level meters provide reasonable sensing insight, while not requiring one sensor at every load. However, they can be difficult to install and to retrofit into existing infrastructure. Hiring an electrician and the downtime cost associated to power disconnection during installation represent a barrier to most of the user.

3.3.3 Non-Contact Meters

To obviate high installation cost that can limit the deployment scenarios, many recent meters have explored various non-contact options. We can divide this group into two smaller families of devices. The first are battery operated (or can be) and attach outside of circuit breaker. They sense the electromagnetic field generated by the current flowing to the circuit or uses current transformer (CT) to measure the current. The second family is energy-harvesting enabled to eliminate the need for additional infrastructure or batteries during installation.

Non-Energy-Harvesting Enabled

A range of solution, both commercial and academia, fall inside this category. Nke-watteco [93] has recently started to offer a battery powered LoRaWAN power meter. The company claims a battery life time of 10 years on a single charge, with a current sample every one minute and a packet transmission every hour. Patel et al. [102] designed a stick-on power meter using magnetometers designed to be attached to the circuit breaker. A similar approach is the one presented in [103] where a giant magnetoresistive sensor is used to measure the current flowing through an individual circuit breaker. The main drawback of theses solution is the use of batteries as power source, that may represent a problem if deployed at scale.

Energy-Harvesting Enabled

To obviate the problem of battery recharge/replacement energy meter are now starting to integrate energy harvesting capabilities [84], [87]–[90], [92]. In [84] Xu et al. present a stick-on wireless current monitoring system. The system integrates a piezo-electromagnetic AC current sensor to both sense the magnetic fields produced by the current-carrying conductors and to harvest energy. In [87] a CT sensor is used for harvesting power from the AC line. The current consumption is then inferred by the node's activation frequency that increase monotonically with the primary load's draw. In [89] a single CT sensor is used for both energy harvesting and current sensing. The node is configured to sense current while in active mode and to harvest energy when in sleep mode.

The interest in energy harvesting current meters is also confirmed by the availability of commercial solution like [92], where a CT sensor is used to measure current and power the meter.

3.3.4 (Real) Smart Meter

Last innovation in the field of power meters are all those solutions that integrate a further layer of intelligence. We are moving from the classic approach – where the principal task of these devices is to provide only metering functionalities – to the integration of new "smartness" like load detection and monitoring. Makonin et al [75] discuss the development of an open source smart meter along with a disaggregation algorithm running locally, able to detect appliances from the aggregate power consumption. The solution, however, does not integrate a power supply and wireless connectivity for real time data streaming. Similar approach is also offered by a commercial solution [94]. However, in this case, the load detection algorithm is run in a cloud environment, and the device need to be connected to an AC source to be powered.

3.4 An Innovative Cost-Effective Smart Meter with embedded Non Intrusive Load Monitoring

Most of the time, domestic energy usage is invisible to the users. Households have only a vague idea about the amount of energy they are using for different purposes during the day. The electric bill is usually the only monthly/bimonthly feedback. Making the energy consumption promptly visible can make the difference and foster a really efficient usage of the energy in a city.

In this section a Home Intelligence device is presented. The proposed solution enables customers to monitor and control their home energy consumption using a Non-Intrusive Load Monitoring system. Power consumption trends can then be extracted from customer data and used to give energy saving recommendations to customers.

The meter follows the Non-Intrusive approach, lowering installation hassle thanks to the use of CT sensors. The development of the proposed smart meter was driven by the idea to provide a cost-effective solution that could be attractive in the consumer marker, easy to install and maintain.

The result is a 35x90 mm footprint device that integrates the MCU, 3 ADC input, Wi-Fi capabilities and also the power supply. The final cost is below 50 € including the three CT sensors; a competitive trade-off between innovative metering service such as NILM, and the cost.

3.4.1 Hardware Implementation

The design and development of the proposed sensing device is driven by the idea to develop a solution to provide NILM functionalities for the residential sector. We chose to base the system on a top of a CC3200[2] System-on-Chip (SoC) from Texas Instruments. This SoC allows the development of low-cost Wi-Fi enabled devices also supporting common functionalities such as high speed ADC interfaces. The acquired data is then used by the NILM engine that adds information about appliance detection and sends a comprehensive report to a cloud service.

Figure 3.3 shows the prototype of the developed smart meter.

Acquisition Front End

To acquire the aggregated current consumption we chose to use non-invasive current sensors (also called split-coil Current Transformer, CT). CTs are cheap and effective sensors for measuring aggregated current consumption since they can be clipped onto either the hot or neutral wire, without any contact to the underneath conductor.

[2]http://www.ti.com/product/CC3200

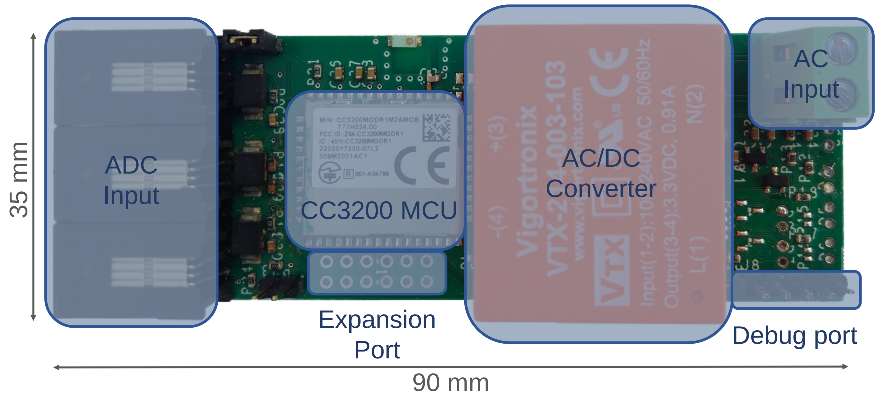


Figure 3.3: Prototype of the developed smart meter

Before connecting the CT sensor to the meter, the output signal from the CT sensors needs to be conditioned so to meet the input requirements[3] of the CC3200 ADC. Specifically it requires a positive voltage with a maximum below 1.4 V. The signal is first converted through an I-V filter and then biased to fit the conversion range of the 12-bit ADC. The schematic of the ADC input is depicted in Figure 3.4. We used a VITEC 57PR1673[4] split-core current sensor with 1:3000 ratio. By using a proper burden resistor we can monitor loads up to 5kW without ADC saturation with a sensitivity below 3W/bit.

The smart meter provides three measuring channels making possible to monitor multi-phase systems or houses where the main power is split in multiple lines. Regulations in some countries impose to split the residential electrical system into sub-circuits, for example one for lights and the others for small and high loads. A multi-line meter such as the one proposed can be applied also in the US where lines are separated and based on the type of the nominal voltage.

Control Unit

The SoC used is a CC3200 MCU from Texas Instruments, developed for the Internet of Things (IoT). The SimpleLink CC3200 device is a Single-Chip Wireless MCU that integrates a high-performance ARM Cortex-M4 core running at 80 MHz, an ARM Cortex-M0 Network Processor Subsystem implementing the

[3]http://processors.wiki.ti.com/index.php/CC32xx_ADC_Appnote
[4]http://www.viteccorp.com/data/CatalogSensing.pdf

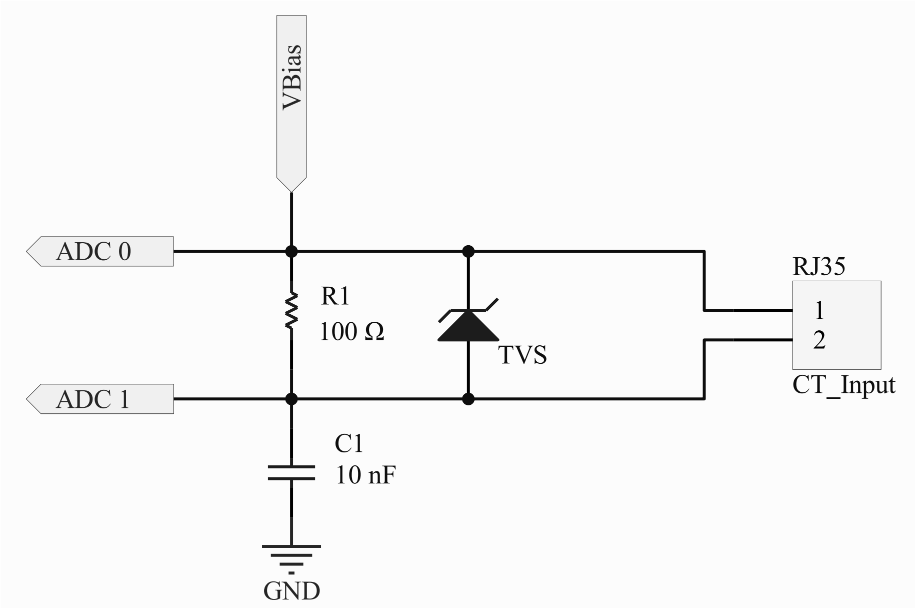


Figure 3.4: Conditioning circuit for current acquisition

Wi-Fi stack, four-channel 12-bit ADC and a wide variety of peripherals. The M4 core is supported by 256 kB of RAM and by 4 MB external serial flash connected through a dedicated SPI interface. The network subsystem includes 802.11 b/g/n radio, baseband and MAC, and supports Station, Access Point, and Wi-Fi Direct modes. This makes possible to firstly start the smart meter in Access Point mode for the initial configuration, and once configured to switch to Station mode to connect to the HAN Wi-Fi. The NILM algorithm implemented in the firmware of the smart meter has been designed to have a small memory footprint, but still high accuracy when measuring.

3.4.2 Firmware Implementation

The firmware running on the smart meter can be divided in multiple stages. We have a preliminary phase called *Initialization* where the smart meter looks for a configuration file stored in the local flash memory. If no configuration is present (i.e. first time initialization) the node operates in *Access Point* mode. In this way the user can directly connect to the smart meter for the first configuration. The device provides a web page where the user can fill with the local HAN Wi-Fi credentials details. Once configured and connected to the local Wi-Fi network, the internal RTC clock is updated from a NTP server.

In the second phase the meter samples the AC signals and calculates the RMS values that are saved into circular buffers. Two buffers with a different length are used. One for event detection and another for event disaggregation. The node repeats the analysis of the buffers every new value, with the goal to identify those appliances currently ON and those that have switched ON or OFF.

For the identification, the node uses the *Appliance Database* saved in the flash memory. This database is previously created during the supervised learning phase and uploaded together with the firmware inside the memory.

Last phase is called *Data Posting*. Here the node sends both the acquired aggregated current consumption and the information about appliances' operating state. System architecture is presented in Figure 3.5. Please note that, in this figure, the preliminary initialization phase has been omitted.

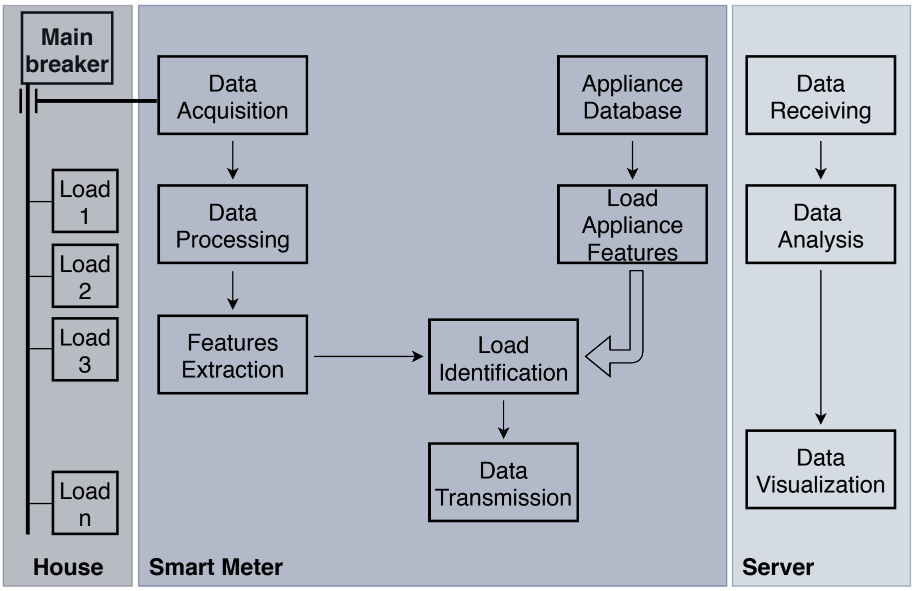


Figure 3.5: Smart metering architecture

Data Acquisition

In this phase, the meter samples the AC signal coming from the CT sensors at a rate of 10 kHz. We chose to sample 5 periods of the AC signal, thus 200 samples per period for a total of 1000 samples for each cycle. The meter calculates RMS values at a rate of 2Hz, storing the samples inside a circular buffer. The workflow of the acquiring phase is presented in Figure 3.6.

For the sake of completeness, the ADC inside the CC3200 MCU reaches Nyquist rate around 31 kHz, hence by modifying the firmware there is enough computing capability to increase the sampling rate to include features for higher order harmonic analysis.

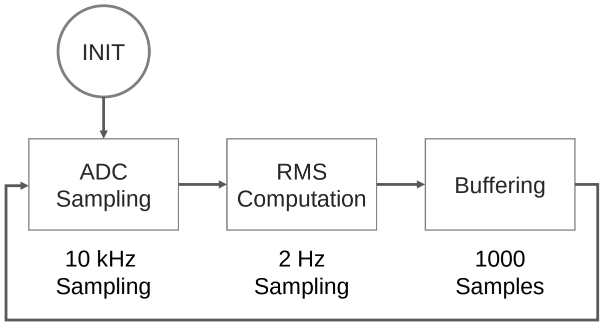


Figure 3.6: Acquisition workflow

Supervised learning

Before the algorithm is able to recognize different appliances from the aggregate consumption, a preliminary learning phase is necessary. During this phase a database of known appliances is created. It will later be used by the identification algorithm that try to match detected appliance during runtime to one already present in the database.

Two different strategies can be followed during the learning phase differentiated by the type of interaction and on the way to engage the user. The First method exploits MQTT connectivity for sending data regarding an unknown appliance detected by the smart meter. This data is then processed for features extraction and inserted in the appliance database. Once completed, the only effort required by the user, is to insert a label that describes the detected appliance. The other method requires more user interaction. In fact, the customer can manually trigger the so called *recording mode* for adding a new appliance power trace. Once the data is acquired by the smart meter, features extraction algorithm is executed and the new appliance added in the database.

Features extraction algorithm is not computational intensive and can run both directly inside the smart meter or in a cloud environment. In this way we can better adapt to different working scenarios.

Data Processing

Data processing uses the combination of two different methods.

The first one uses a circular buffer with a depth of 40 RMS samples. In this phase, the load identification engine looks for changes in the consumption trace, and labels them as *events*.

The second methods uses at least 1000 samples to confirm the previously detected event. In fact, it calculates a Power Spectral Density estimation that needs a large set of data to detect some periodic components typical of some electric appliances.

The number of samples to be considered by the load identification engine is therefore a trade-off between time delay and accuracy in the identification and has been tuned by experimental tests.

Event Detection

Starting from the RMS sample acquired, event detection routine tries to identify the instant in which a transition from OFF to ON state, and vice versa, occurs. Event diction is based on a sliding window that moves forward checking sample by sample for a power change. In first place, the algorithm calculates the average before (AVG_B) and after (AVG_A) the sample we are analyzing, by means of a Simple Moving Average (SMA). The windows size, used for both averages, is adapted dynamically during run time, to better fit different appliances and to ensure to check one state change per time.

To address the inaccuracy of the measuring device, and to neglect small power consumption changes more prone to false detection, we consider only power changes over a fixed threshold equal to 45 VA. This value, experimentally chosen, is a trade-off between resolution and false positives rejection.

Event Disaggregation

Once an event has been detected, the algorithm tries to understand which appliance has changed its state. The process is based on pattern recognition techniques. The same features that characterize the event under analysis are compared with those already present in the database that characterize the various appliances. The appliance that exhibits a behavior like the one analyzed in the aggregate consumption is chosen as the one that has changed his working state. First, we compute the periodicity analysis looking for relevant frequencies (or periods) in the observed signal. We then move to transient analysis to extract those features characterizing the transitional behavior associate to a working state change. Finally, the algorithm extract some more features (i.e. RMS steady current), decides what are more meaningful for disaggregating the event under analysis and then computes the probability for each appliance to be selected. The appliance with higher probability is associated to the detected event.

The probability is a measure that defines the degree of similarity between the features extracted in previous phases and those coming from the appliance database.

Features for load characterization will be presented in the following section.

Data Posting

The smart meter is configured to post periodical messages about the current state of appliances (i.e. which are currently ON) and about the electricity consumption level. The smart meter creates every 60 seconds a MQTT message with a time stamp, the last RMS current samples for each line under measurement and the report about appliances state change. In this way the results from the load detection analysis are available in real-time to both households and utility companies that can use those information to achieve a better energy efficiency.

Memory requirement and used features

During run time, only 150 kB of RAM are available to the user, as part of the RAM is reserved for the underlying operating system. The main challenge was to fully exploit the serial flash and dynamically load data only when needed. The node is also configured to connect to the Wi-Fi network only when it needs to stream the acquired data to decrease the RAM impact of the network subroutine.

Memory wise, the firmware running inside the smart meter can be divided into two separate parts; 1) Data acquisition and 2) Data analysis. 16 kB is the memory requirement for the data acquisition part, that includes the buffers needed for ADC sampling, RMS calculation and the circular buffers used for storing the RMS samples used for the load identification engine. (48 kB if we want to monitor three lines).

Regarding the NILM algorithm, we ported our initial [72] MATLAB script in C with changes to the code necessary for operating on the embedded platform. Firstly, we reduced the number of features used to identify an appliance; then we have optimized of the moving average function, used also for calculating rise and settling time of an event. Finally, the implementation of a simple FFT analysis for extracting potential cyclical behavior from the aggregated current consumption.

After a deep evaluation, we kept only 9 main features from the original MATLAB algorithm:

- **Transient state consumption [VA]**, average consumption in transient state of an appliance when it switches ON.
- **Steady state consumption [VA]**, average consumption in steady state of an appliance.
- **Peak consumption value [VA]**, the maximum consumption value of an appliance.
- **Minimum consumption value [VA]**, the minimum consumption value of an appliance.

- **Settling time [Samples]**, expressed as number of samples, the time required to reach and stay within a range of about 5% of the average power in steady state.
- **Rise time [Samples]**, expressed as number of samples, the time required to rise from 10% to 90% of its steady state value.
- **Periodicity [Samples]**, if an appliance exhibits a cyclical behavior, like, for example, the spinning motor inside a washing machine, this feature states the frequency of the cycle.
- **Minimum ON time [Samples]**, the average minimum number of samples that an appliance stays in ON state.
- **Maximum ON time [Samples]**, the average maximum number of samples that an appliance stays in ON state.

The database of features for the whole set of appliances, that in our case consists of 6 loads, once loaded from the serial flash, accounts for less than 256 Bytes. With these improvements more than 100 kB of free RAM and more than 3 MB in serial flash are available on the meter for running the features extraction algorithm followed by the load identification (NILM).

3.4.3 Results

We evaluated the load identification performance by installing the smart meter inside real houses. The smart meter was connected on the main electrical panel of each residential unit, monitoring the global current consumption. During this period, we have recorded, among others, events associated to 6 different appliances; Coffee Machine, Electric Heater, Fridge, Hair Dryer, Microwave Oven and Washing Machine.

Table 3.2 shows testing results using the proposed smart meter.

True detect ratio expresses the number of times the algorithm has correctly detected an appliance. This ratio is calculated by comparing the number of real appliances state changes – manually extracted from the aggregated consumption

Table 3.2: Load identification algorithm recognition performance

Appliance	TrueDetected	FalseDetected	Falsepositive	FalseNegative
Coffee Machine	97.71%	1.17%	2.34%	3.51%
ElectricHeater	100%	0%	8.33%	8.33%
Fridge	92.73%	0.71%	23.75%	4.04%
HairDryer	83.33%	26.67%	26.67%	0%
Microwave	79.27%	0%	3.08%	1.54%
Washing Machine	100%	8.33%	0%	75.00%
Total	92.17%	6.15%	10.69%	15.40%

– and the number of appliances state changes provided by the load identification algorithm.

False detection ratio expresses the number of times the algorithm has associated a known appliance (i.e. that is in our database) to a wrong one. *False Positive* and *False Negative* express, respectively, the number of times the algorithm has detected that an appliance has switched ON after the right moment and the number of times that the algorithm has detected that an appliance has switched OFF before the right moment.

As can be noted from Table 3.2, the average correct detection ration is over 90%, which is in line with more complex and off-line load identification algorithms. Performance in detecting when the Washing Machine switches OFF, however, needs to be improved.

Despite all progress made concerning disaggregation techniques, performance evaluation and comparability remains an open research question. The lack of standardisation and consensus on evaluation procedures makes reproducibility and comparability extremely difficult [104]. Moreover, most of the time, the proposed NILM algorithms does not run real-time on the sensing device in contrast to the proposed approach. It is thus difficult to directly compare the result obtained with the proposed device to other implementations.

Similar approaches are proposed in the literature [75], [105]. The proposed smart meter exhibits comparable accuracy with the advantage of having the whole appliance recognition application running real-time inside the smart meter. Also integrating wi-fi connectivity for data streaming. Comparing the results with more complex implementation running off-site [106], the developed smart meter still presents an accuracy comparable with the state of the art.

3.4.4 Summary

Residential load monitoring and identification are particularly important tasks for power demand side management and user energy efficiency management.

In this section we have presented the development of a complete and cost effective metering system capable of detecting and classifying the power consumption of a set of residential appliances. We have presented the main parts of the developed smart meter, with a focus on the data acquisition part, that provides RMS values at 2 Hz.

Using this low-cost hardware platform, we have implemented a load disaggregation algorithm that executes real-time using only RMS current samples.

From the experimental results it was demonstrated that good accuracy in detecting individual loads running inside a household is now possible without complex off-line systems nor cloud support.

3.5 A battery-free non-intrusive power meter for low-cost energy monitoring

In this section we discuss the design, development and characterization of a power-proportional, energy-harvesting, non-intrusive energy meter. Thanks to a clamp-on Current Transformer, it is able to harvest the energy needed for its operation directly from the same load under monitoring. The proposed solution is equipped with an ultra-low-power STM32L0 microcontroller from STMicroelectronics, that is in charge of configuring the radio, a LoRa modem, that permit us to meet the energy budget for sending a packet also at low primary load, while still ensuring a transmission in the range of kilometers. We chose to use a Murata Connectivity

Module that integrates both the MCU and the LoRa radio in a compact package allowing to reduce the device final footprint.

The working principle is simple: once the sensor is clamped around one phase of the main line it starts to harvest energy from the load under monitoring. This energy is then stored inside an energy reservoir, in our case a supercapacitor; when sufficient energy has been harvested the sensor node, powered by the supercapacitor, turns on and sends a data packet to a remote server.

Under steady-state conditions, we may assume that the power-on plus the packet transmission task consume a fixed amount of energy from the supercapacitor. Therefore, because the rate at which the supercapacitor is charged depends on the current drawn by the load, the node activation rate will also depend on the load power. After characterizing this dependency, the server can therefore estimate the load power simply from the rate at which it receives the packets.

This design offers several advantages over more traditional solutions. First, installation costs are lowered due to the non-intrusive approach that does not require to deal with primary main voltage. No physical connection with the monitored circuit is needed. Secondly, maintenance costs associated to battery replacement are avoided and the energy overhead introduced by the metering infrastructure negligible and proportional to the load under monitoring. This ensures zero-power under zero-load condition.

3.5.1 System overview

Our design choices have been driven primarily by three properties that characterize the system: *non-intrusiveness, battery-free operation and low power consumption.*

In accordance with these properties, we have employed a clamp-on inductive current sensor which functions as the transformer. This type of transformer allows the device to be non-intrusive since it can be deployed easily without modifying the circuit and the device under measurement. In addition, the transformer is the

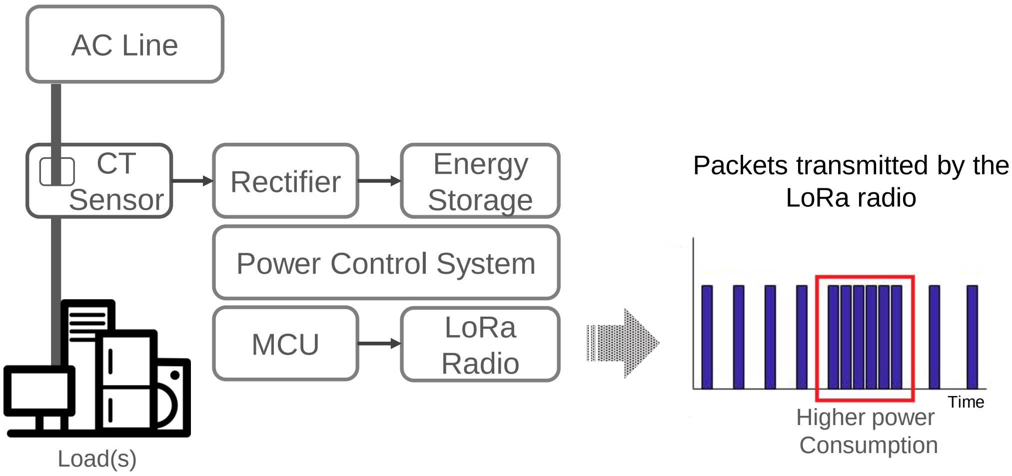


Figure 3.7: Block diagram of the proposed metering architecture.

sole energy supplier of the node, which operates without the use of any battery. The meter can therefore operate completely unattended.

Because there is no actual sensing layer with its associated sampling and conversion circuit, the node architecture is extremely simple. The direct consequence is the implementation of an extremely low power transmission system, an essential feature that increases the effectiveness of the harvesting solution.

According to the implemented choices, we can monitor the load consumption by analyzing the rate of the transmitted packets. This is possible thanks to the correlation between the charging speed of the capacitor and the current absorbed by the primary load.

This technique for monitoring electricity consumption has many advantages, among which:

1. No use of peripheral devices is required by the MCU to perform the measurements. This includes the current transformer, which is used as a harvester and only acts indirectly as a current sensor. This approach drastically decreases the overall power consumed by the node, which is limited to setting up the radio and sending a packet. No Data processing is required, simplifying the energy harvesting task.

2. The MCU remains deactivated for the entire capacitor charging period, and

is activated only to carry out a transmission. Nevertheless, because the harvester continuously integrates the generated current into the capacitor, the node is effectively measuring the power consumption even during the sleep periods of the MCU. In other words, we do not have to deal with problems related to infrequent sampling or lost data, nor do we need to worry about changes in the load power level. These are automatically accounted for. The method is therefore suitable to monitor individual appliances as well as the main power distribution lines.

3. The packet does not need to contain any measured value, since power consumption is inferred by the frequency of the transmissions. It is therefore possible to decrease the number of bits sent for every transmission, thus reducing transmission time and consumption. This is especially relevant in the context of long range transmissions, which are unable to carry large amounts of data. Conversely, data packets could be delivered more infrequently by collecting the wake-up times in the local memory, and then transmitting the information only once every n activations. By doing so we can strike the best trade-off between node power consumption and data availability, without sacrificing accuracy.

Data loss mitigation

Because we estimate the energy consumption from the packet transmission rate, we also have to correctly receive all the packets. If packets get lost during transmission, the energy consumption will be underestimated, leading to an error. To mitigate this problem, the meter inserts in the payload of the packet a monotonically increasing counter along with the timestamp of the last 3 activations. The server can therefore recover lost packets with very high probability.

3.5.2 Energy meter hardware design

The block diagram in Figure 3.7 shows the overall architecture and working principle of the proposed system.

A current transformer is clamped around one phase of the supply line going from the mains to the load that we intend to monitor. The supply line acts as the primary winding of the CT. As the load draws power from the mains, a current proportional to the load consumption is induced in the secondary winding of the CT and can be harvested by storing it in an energy reservoir, in our case, a super-capacitor. To harvest this energy, the current must be rectified through a diode bridge before it is stored in the super-capacitor. A microcontroller (MCU) and a radio transmitter are connected to the capacitor through a power supply control system and are normally switched off, to eliminate energy waste when the super-capacitor is recharging. When the voltage across the capacitor reaches a predetermined threshold, the control system activates the microcontroller, that configures the LoRa radio and send a Packet that will be received by a remote server. At the end of the transmission, the microcontroller and the radio are deactivated. The operation draws energy from the capacitor, whose voltage decreases back below the threshold. At the same time, the harvester starts to recharge the capacitor, at a rate that is proportional to the power drawn by the load from the main line. Thus, once the voltage crosses the threshold again, a new transmission operation is started by the MCU.

This cycle repeats at a rate proportional to the energy harvested from the load: the higher the load power level, the more frequent the packet transmission, as shown in the inset of Figure 3.7. Hence, the packet transmission rate depends on the load power, which can therefore be estimated by the remote server by simply counting the number of received packets in a specific time frame.

In the rest of this section we provide more details on the individual components that constitute our architecture.

Microcontroller and radio

The device use a Murata ABZ-078[5] module integrating the microcontroller, a low-power STM32L0[6] from STM and the radio used for transmission based on

[5]https://wireless.murata.com/type-abz-078.html
[6]https://www.st.com/en/microcontrollers-microprocessors/stm32l0-series.html

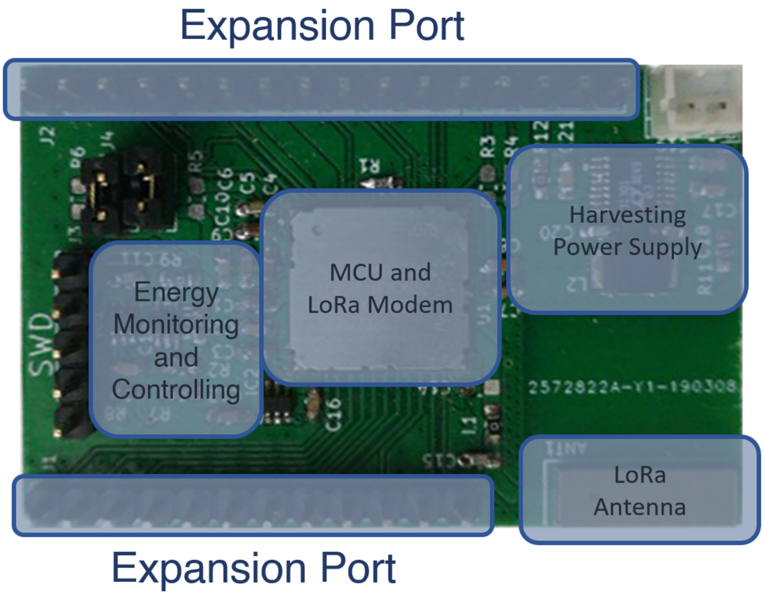


Figure 3.8: Prototype of the proposed meter. The core is a Murata module integrating a low power STM32 MCU and the LoRa radio.

a LoRa SX1276[7] modem. The prototype, assembled with all its components, is shown in Figure 3.8.

Current Sensor

The current sensor consists of a clamp-on inductive sensor, applied to the load. For harvesting the energy needed to power the board, as previously stated, we have used a Current Transformer (CT), that is clamped around one of the phases that goes to the load that we want to monitor. Different tests were conducted to determine the most suitable sensor for this kind of application.

Two different sensors have been compared. A LEM TOP90-S10/SP2[8] with a 1:1000 ratio and a VITEC 57PR1673[9] with a 1:3000 ratio. Contrary to what we would have expected, the sensor with 1:3000 ratio, that theoretically should provide one third of the current compared to the former, performs much better. Thus, experimental tests were conducted with the latter.

[7]https://www.semtech.com/products/wireless-rf/lora-transceivers/sx1276
[8]https://docs-emea.rs-online.com/webdocs/14ce/0900766b814cedb3.pdf
[9]http://www.viteccorp.com/data/CatalogSensing.pdf

Rectifier circuit

The current rectifier is of fundamental importance, since it affects the overall efficiency of the energy harvester. Our circuit consists of a Graetz bridge with Schottky diodes, featuring a low forward active voltage. To choose the right diode for this type of application, our research has started by searching for the diode with the lowest forward voltage and leakage current.

Once a range of diodes was selected, tests were performed to determine the one with the best efficiency at lower loads. After some experiments, we have chosen to use four STM BAT30[10] diodes, as they proved to be the most efficient at lower load level, even though not the most efficient at higher loads. The motivation behind this choice is that we are more interested in higher efficiency when the energy that can be scavenged is limited, in order to increase the node activation rate at lower primary loads. Conversely, at higher primary loads we have plenty of energy for powering the meter, so the lower efficiency of the selected diode can be neglected.

Capacitor

The capacitor is the heart of the system as it is in charge of storing the energy harvested and needs to be properly sized. It must guarantee sufficient energy for the microcontroller operation and for the radio to transmit a packet, but also not represent a barrier for the activation at lower loads. After some experimental tests and measurements, the AVX BestCap[11] 22 mF / 4.5 V supercapacitor was chosen as the best compromise between charge times and stored energy.

Power supply control system

The power supply control system activates/deactivates the MCU, allowing or preventing the flow of current from the capacitor. At the current state of work, this component is still a prototype, as its operation must be carefully calibrated according to the harvester and the node characterization. In the tests that we

[10] https://www.st.com/en/diodes-and-rectifiers/bat30.html
[11] http://www.avx.com/products/supercapacitors/bestcapreg-bz-series/

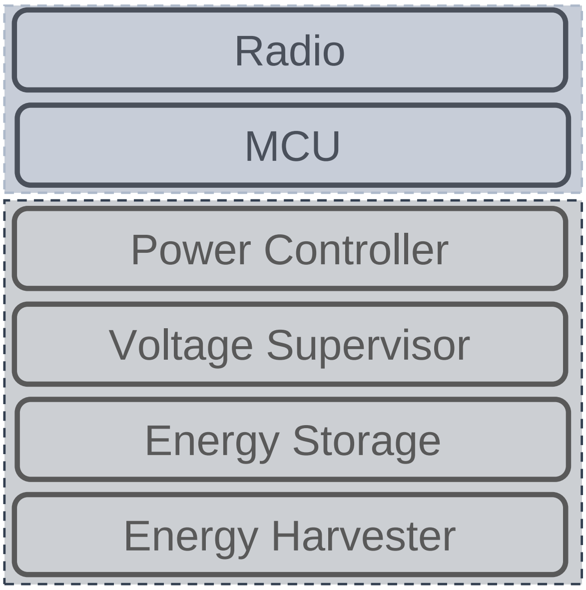


Figure 3.9: Power supply control system block schematics. The voltage supervisor monitors the energy storage status comparing it to the activation threshold. The power controller is in charge of both, 1) Wake the MCU when the activation threshold is crossed; 2) Deactivate the MCU after the transmission of a LoRa packet.

have carried out, a completely autonomous power supply system was used, so that no external factor would influence our measurements. It is mainly composed by an ultra-low-power voltage supervisor, that activates its output only when the input reaches a fixed threshold. Moreover, it provides a "done" input pin that can be used to disable the output. This pin is connected to the MCU that is configured to send a signal when the packet has been sent. In this way, we keep the MCU and radio totally disconnected while the super capacitor is recharging.

The power supply is shown schematically in Figure 3.9.

3.5.3 Experimental results

As our main goal was the full characterization of the relation between the transmission rate and the load power level, our experiments have evaluated different device solutions for the node architecture and the characterization of both the energy required for data transmission. Finally the rate at which energy can be harvested and stored in the super capacitor.

The experimental data are essential in order to correctly calibrate the measurement system. This to quantify the energy quanta needed for transmitting one packet and then link this parameter to the load power consumption to express a relation between the load level and the activation rate. A first set of experiments was intended to characterize the performance of two current transformers, having respectively a 1:1000 and 1:3000 ratio.

Because in a CT there is an inverse relationship between the number of turns in the secondary windings and the induced current, we expect the transformer with the lower number of turns to perform better, i.e., to charge the super-capacitor more quickly. However, other parameters, such as the impedance mismatch with the non-linear rectifying circuit – which ultimately determines the amount of energy that can effectively be transferred – and the geometric shape of the sensor, must also be taken into account. Various measurements were performed, showing how these secondary effects have a significant impact on the overall performance, leading to the use of the sensor with a higher ratio.

The result of this analysis is presented in Figure 3.10. While the 1:1000 sensor behaves better at the very beginning of the charging cycle, the overall charging time at 100 W with the 1:3000 sensor is considerably lower, denoting a better efficiency compared with the other one.

Besides the current sensor, a set of experiments has also been conducted to test different diode bridges to select the one that is more suitable for this kind of application. We have decided to use the one with a higher efficiency at the lower load levels, even if not the best at higher load levels. This is because we are more interested in the best measurement granularity at lower loads, as the energy that can be scavenged at higher loads is always sufficient to meet the energy budget of the platform.

This analysis has lead us to use a BAT30 diode. Figure 3.11 shows the result of these experiments, where the charge level of the super capacitor over time is compared among the different diode bridges while harvesting energy from a 130 W load. The differences are due to the different forward resistance of the

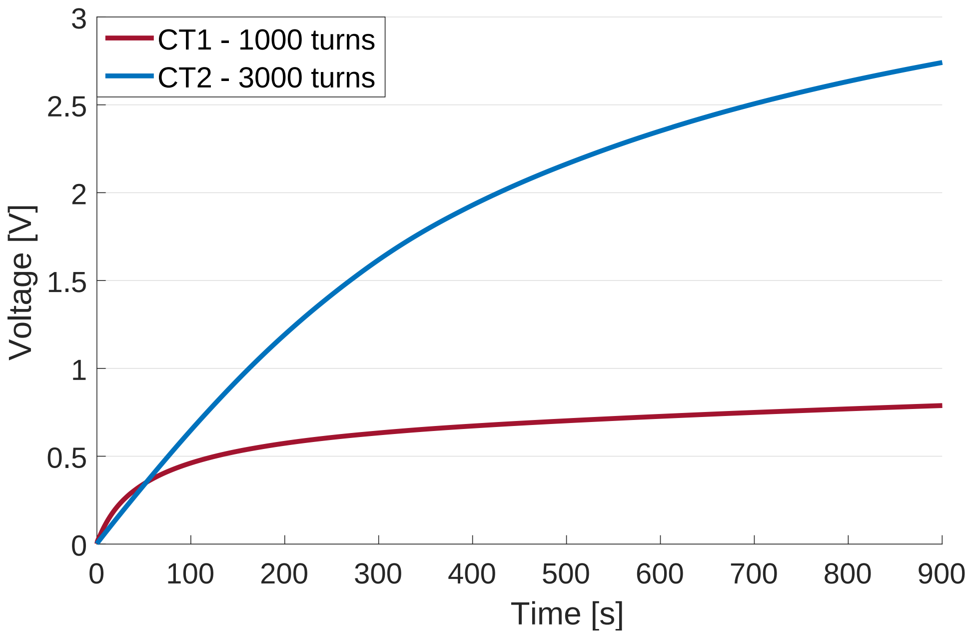


Figure 3.10: Supercapacitor voltage trend using different transformer ratio. Contrary to the expected behavior, the CT sensor with a higher number of turns better match the energy harvesting system, leading to a higher efficiency in charging the super-capacitor.

devices as a function of the current level. To simulate the various main loads, a variable load made by a cluster of incandescent light bulbs has been used. This cluster permits us to vary the load from 50 to 1600 watt in 50 W steps.

Once the characterization of the current sensor and the most efficient bridge rectifier had been conducted, it was necessary to size the capacitor. For this purpose, we have collected the current consumption trend of the node during one operation cycle (i.e., Power on, configuration and packet transmission), and then calculated the energy required for the task. As the energy is drawn from the capacitor, the node supply voltage decreases. For correct operation, the voltage should not decrease below the radio operating limit, which we fix at 3 V. We can therefore compute the value of the capacitor, by considering the energy stored at the desired voltage levels. We obtain a capacitance of approximately 11 mF. Given the device tolerance, and to provide sufficient operating margins, we have employed a 22 mF capacitor, because it stores enough energy to supply both

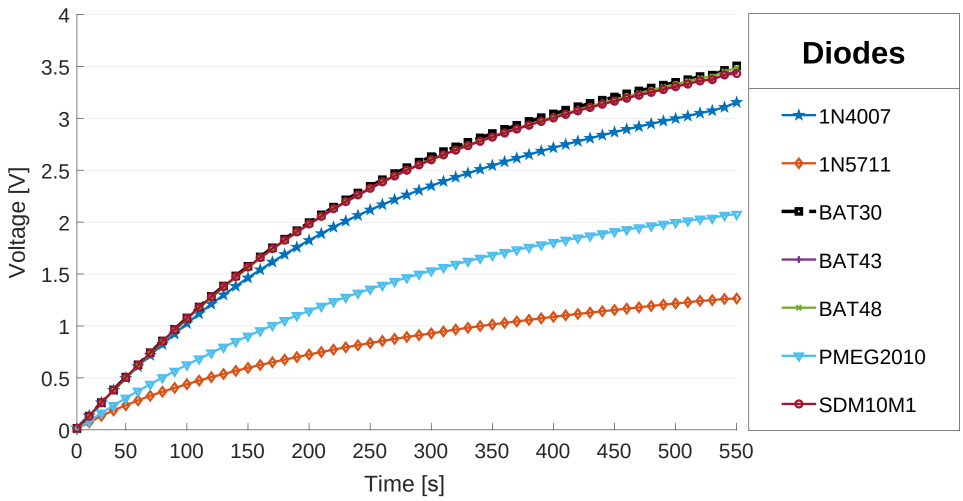


Figure 3.11: Supercapacitor charge profile for a 130 W load using different rectifier bridges. The black line (BAT30), represent the chosen solution. We chose this diode as more efficient at lower load, in order to extend the functionalities of the meter also at lower primary loads.

the microcontroller and the radio during the transmission, guaranteeing however reasonable charge times.

Figure 3.12 shows the charging profile of the capacitor over time as the load varies from 200 W to 1465 W. As the energy needed to ensure a fail-safe LoRa transmission is met at 3.39 V, the graphs show the charging time to reach that threshold. Naturally, the lower the load the longer the time required to reach the threshold. It can be noted that the platform exhibits a reasonable charging time starting from 200 W already.

Figure 3.13 shows the curve obtained by interpolating the measured full charging time at different load power levels, starting from a fully discharged capacitor. In other words, it shows the minimum load power required to reach the voltage threshold given a required time. The estimated curve represents a branch of an equilateral hyperbola. In contrast to the fitting curve, the measurements (the dots in the figure) show that the lower power loads need slightly more time to reach the threshold voltage compared to the estimated time. This is due to the non-linear behavior of the circuit, with the non-ideal parameters having a more pronounced effect at lower loads and currents.

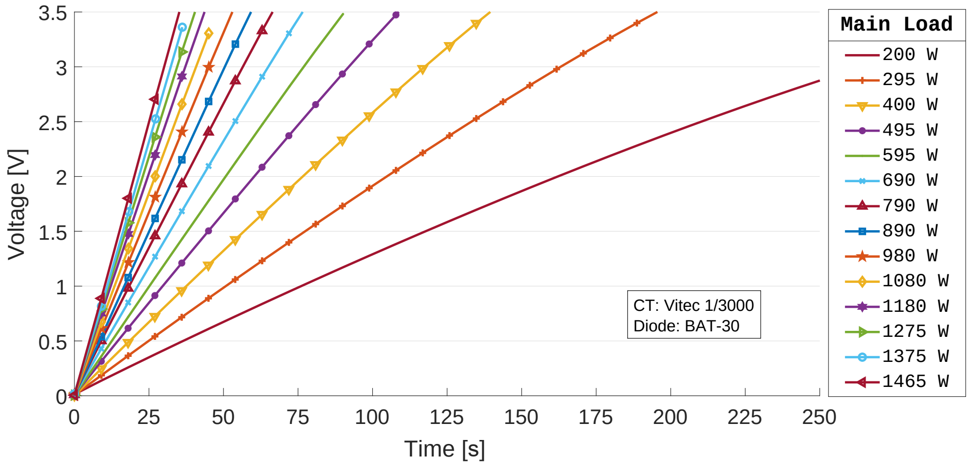


Figure 3.12: Voltage trend of the 22 mf super-capacitor while harvesting energy from different primary loads.

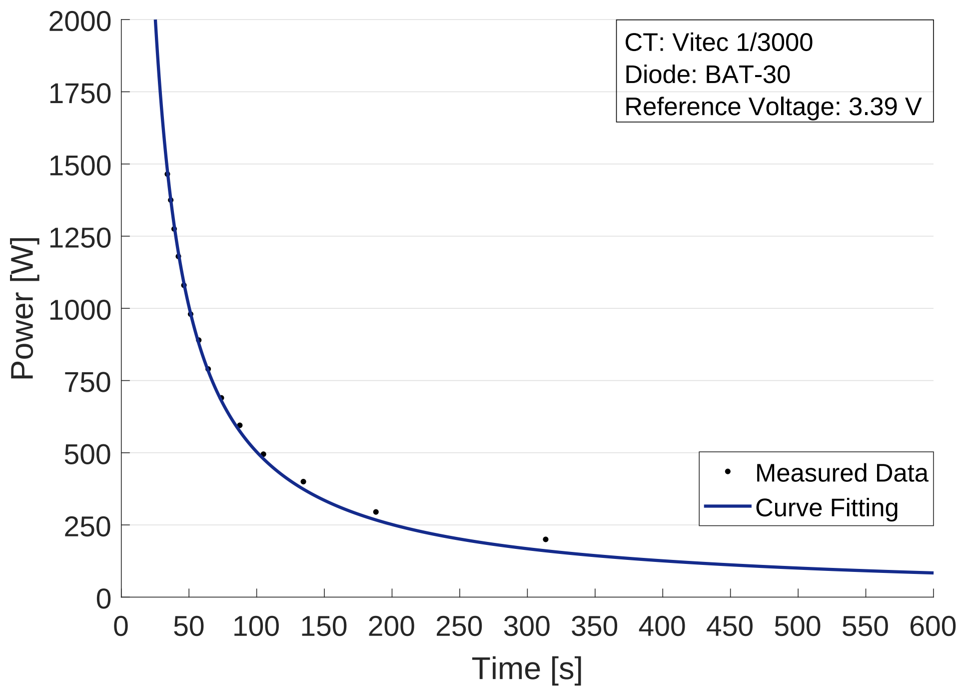


Figure 3.13: Packet transmission rate at different primary loads (starting from a fully discharged supercapacitor).

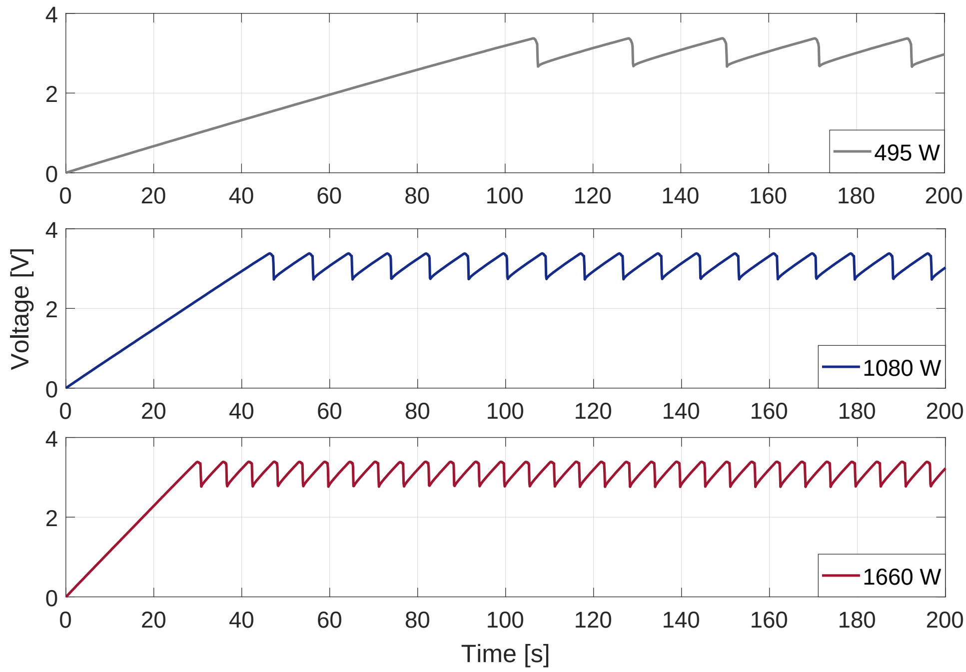


Figure 3.14: Super-capacitor voltage drop due to a packet transmission at different main loads. The amount of energy for transmitting a packet remains constant through different loads, while the transmission rate increases as the primary load increases.

In the next step of the experimentation, we have analyzed the behavior of the capacitor during the transmission period. The radio has been programmed to transmit a packet with a reduced number of bits. In these measurements, the packet is transmitted with a maximum gain, equal to +18 dBm and a spread factor of 12, to simulate the worst working condition. Of course, in case we do not require the maximum transmission range, these parameters can be relaxed, leading to a lower energy requirement, meaning a higher activation rate and accuracy. At the end of each transmission, the microcontroller sends a signal to the power supply control system, which inhibits the current flow, so that the capacitor can start to charge again.

Figure 3.14 shows the trend of the voltage across the capacitor during the initial and the full operation period, for three different primary load power levels. As expected, the packet transmission rate depends on the load and remains constant if there are no power variations.

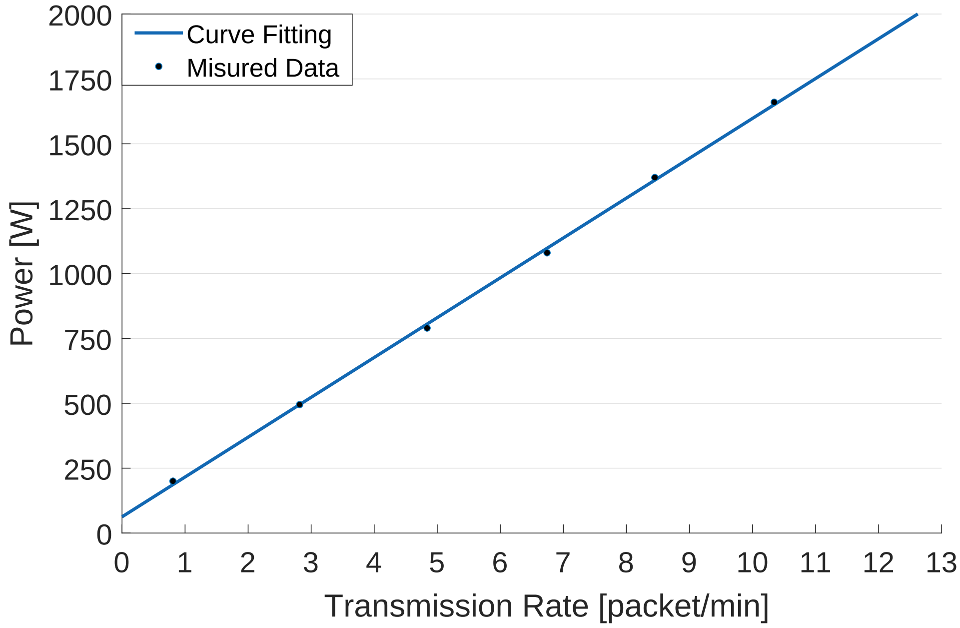


Figure 3.15: Relation between main load and Packet rate [packet/min].

On the basis of the data shown above, it is easy to relate the load power level to the steady state packet transmission rate. This relation is shown, together with its linear interpolation, in Figure 3.15.

In contrast to the initial full charge time, the relation is close to linear, since the capacitor is only partially discharged and the energy required to bring its voltage back to the threshold is much smaller than that required during start up. We observe that even at low power levels, we are still able to send one packet per minute. This confirms that the system can operate on a wide range of loads.

3.5.4 Summary

In this section we discussed a battery-free, non-intrusive power meter for low-cost energy monitoring. We show that for a certain class of energy sensing, an effective sensor can avoid a common acquisition front end by deriving the energy consumption from the energy-harvested itself. This enables very simple sensor hardware that drives down the cost of energy sensing and enables simple and straightforward deployments. Each meter uses a broadcast packet to reach a nearby gateway which can then calculate the packet rate. The characterization shows that the optimal operation is obtained with medium-high power loads, with rates of several packets per minute. At the same time, lower power levels can still be monitored and duly reported to the server, despite the lower packet transmission rate, as the node continuously integrates the current from the supply line.

Future work could include optimizing the system when it works with a low power consumption load. This will guarantee a complete range of measurement on commonly used devices, including more recent lower power lighting and battery chargers. In particular, it is possible to change the capacitor in order to obtain devices with different measuring times and various resolutions, by adopting different data transmission strategies. This would decrease significantly the energy that needs to be stored in the capacitor, leading to much faster charging times.

3.6 An energy autonomous smart sensor with load recognition capability

In this section a novel smart device able to adapt its working modality based on the sensed current consumption, in order to ensure energy neutrality working condition, is presented. By exploiting a LoRa radio, and an energy adaptive firmware, the device can provide basic metering functionalities (i.e. apparent power) when the measured current is below the sustainable threshold and advanced report, like loads detection and recognition, when the consumption of the load under monitoring is higher.

Experimental results have shown that the meter is able to provide power consumption report every one minute starting from a measured power consumption of 50 W, while advanced functionalities are activated starting from 700 W.

3.6.1 Hardware Overview

The system architecture of the proposed smart meter is presented in the block diagram in Figure 3.16. It is mainly composed of 4 parts:

1. The control unit, a STM32L4 MCU from STMicroelectronics, in charge of

calculating RMS current, compute load detection and to adapt duty cycle and the CPU frequency based on the sensed current

2. The acquisition front-end, that filter the signal coming from the first CT sensor to be suitable for the ADC, used for current sensing
3. The energy harvesting power supply, that uses the second CT sensor for harvesting energy from the load under measurement and stores this energy inside a super capacitor
4. The LoRa radio, in charge to send periodic reports to a remote server. We have chosen to use a LoRa radio due to its unique low power and long-range proprieties and also to guarantee the energy budget provided by the harvesting circuit

This modular design make the device flexible and adaptable to different scenarios: 1) *Residential*, the utility company can provide the smart meter and deploy just a few gateways around the town, to develop a distributed monitoring system [107]; 2) *Commercial*, by deploying a distributed metering infrastructure that needs near-zero maintenance costs, to monitor the energy consumption of commercial building [108]; 3) *Industrial*, to supervise and control equipment to ensure their operation stays productive and to avoid failures [109].

In this section the architecture of the proposed smart meter is discussed. We will show in detail both front ends for energy harvesting and current sensing and then we will illustrate the adaptive firmware running on the smart meter with also a focus on the NILM algorithm.

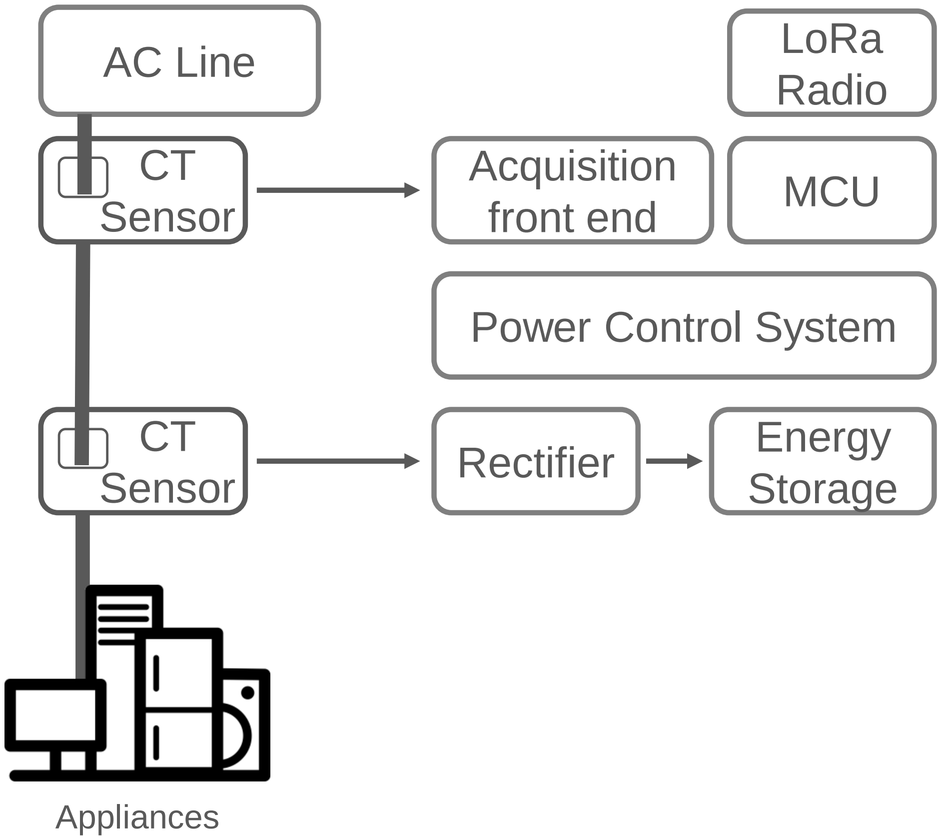


Figure 3.16: System block diagram. From top to bottom: 1) LoRa radio for data streaming; 2) STM32L4 MCU, in charge of acquiring the current readings and adapting the firmware accordingly; 3) Acquisition front end connected to the first CT sensor for current sensing; 4) The power control system monitoring the status of the energy storage; 5) The energy harvesting power supply mainly compose by 2 storage capacitor and the second CT sensor for energy harvesting.

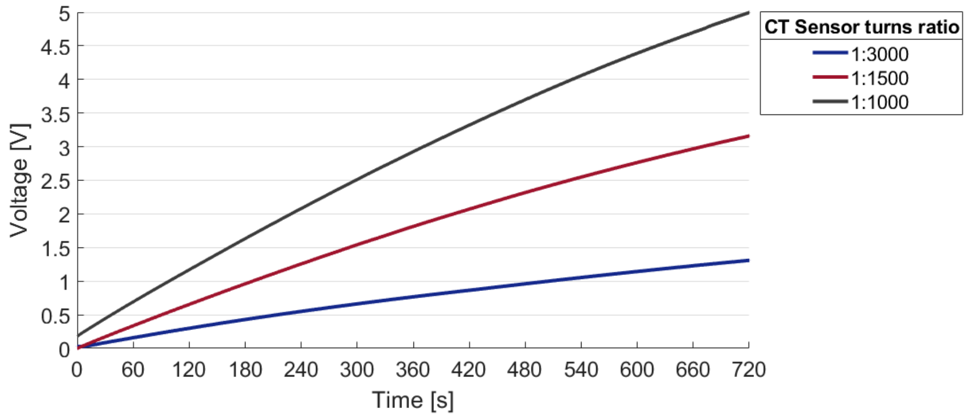


Figure 3.17: Voltage trend of the super capacitor using the Vitec CT sensor with different turns ratio.

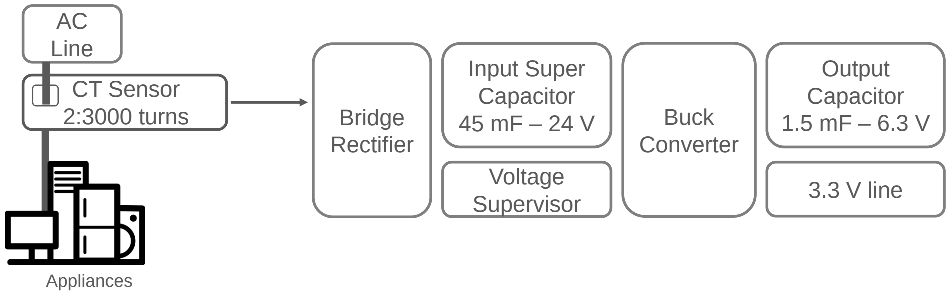


Figure 3.18: Smart meter energy harvesting power supply. Current coming from the second CT sensor is rectified and stored inside the input capacitor (C_{store}). When reached 5.05 V the buck converter is activated, providing an output voltage of 3.3 V.

Energy Harvesting

The energy harvesting power supply is in charge of providing power to the IoT sensor. Harvesting is based on rectifying the output of a current transformer that is clamped around the phase of the AC main line running to a breaker in a circuit panel. When the circuit is drawing current, the CT produces an output current proportional to the current flowing in the main circuit. This current is then used for charging a supercapacitor.

This approach provides different advantages over classic methods. It does not require any battery, which impose a lifetime constraint and maintenance cost; it removes the need for an external power supply plugged into a nearby outlet, which may not be available. The block scheme of the circuit is presented in Figure 3.18.

As CT sensor, we employed a 57P Vitec split core current sensor[12] with 1:3000 current ratio, with 2 turns of the primary coil for a turns ratio of 1:1500. Experimental test has shown that with this configuration we can harvest energy starting from a 50 W load with a cold start (i.e., the time needed to charge the super capacitor from 0 V) shorter than a hour. At the same time, no problems are encountered with higher loads that could provide more energy that the one used by the smart meter. A brief comparison of the performance with different turn ratios is presented in Figure 3.17.

[12] http://www.viteccorp.com/data/CatalogSensing.pdf

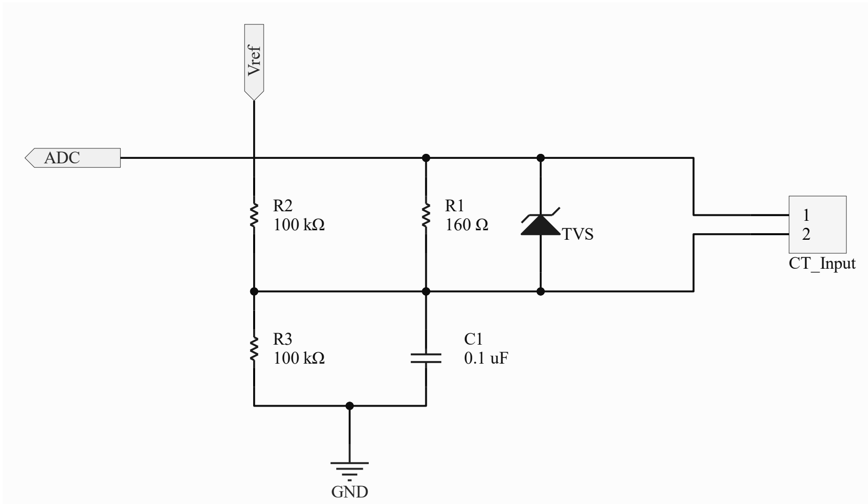


Figure 3.19: Acquisition front end. Current signal is I-V filtered with the 160 Ω resistor and then centered to 1.65 V. The TVS diode protect the ADC input from spikes, while the small 0.1 μf capacitor is used to filter the signal.

The output of the transducer is then wired to a Linear Technology LTC3588 Nano-power Energy Harvesting Power Supply[13]. This harvester integrates a low-loss full-wave bridge rectifier that rectifies the AC current coming from the CT sensor, and store this energy inside a supercapacitor (C_{store}). We used two AVX bestcap capacitor of 90 mF 12 V connected in series in order to obtain an energy reservoir tailored to the input specification of the energy harvester. The harvester also integrates a high efficiency buck converter to form a complete power supply solution. We configured the power supply to provide a 3.3 V output and to activate when the input capacitor reaches the threshold voltage of 5.05 V. The output is then disabled when C_{store} voltage drops down to 3.67 V.

Acquisition Front End

For acquiring the aggregate current consumption, a second current transformer is used. The output of the sensor is connected across a burden resistor in order to transform the current signal into a voltage signal and then biased to meet the input requirement of the 12-bit ADC.

We have designed the acquisition front-end to be suitable for the residential sector and to fully exploit the dynamics of the ADC requiring a positive signal within

[13] http://cds.linear.com/docs/en/datasheet/35881fc.pdf

3.3 V. The value of the burden resistor can be calculated using (I), once fixed the maximum current we want to monitor. In our prototype we chose a 160 Ω resistor. This allows to monitor loads up to 22A with a resolution of 3,5 W per bit.

The signal coming from the CT sensor is also biased and centered around 1.65 V using a voltage divider connected to a voltage reference generator. Two 100 $k\Omega$ resistors are used to minimize the drained current. Finally, to protect the ADC from spikes, a TVS diode is used to clip the signal at a maximum of 6V.

The whole acquisition front-end is presented in figure 3.19.

$$R_{burden} = \frac{V_{ref} \ * \ CT_{turns}}{2\sqrt{2} \ * \ I_{primary-max}} \tag{I}$$

3.6.2 Adaptive Firmware

The firmware running on the smart meter has been developed to dynamically adapt based on the consumption level measured. The final goal is to achieve energy neutrality in different work load scenario. This strategy is implemented mainly with two different techniques: 1) *Dynamic Frequency Scaling* (DFS) and 2) *Dynamic Duty Cycle* (DDC). These two techniques allows to modulate the energy requirement of the platform and to adapt to the energy intermittency that characterize harvesting solutions.

Once enough energy has been harvested and stored inside the supercapacitor, the node activates and checks the flash storage to determine last report transmission or other configurations. In the case of the first start, the device samples the ADC, stores the acquired value inside the flash, reconfigures the clock and duty cycle and then goes to sleep. Once every 5 minutes, if the harvested energy is enough, the node sends a LoRa packet with all the data stored.

Advanced functionalities are activated only if the acquired consumption level is over the sustainability threshold, discussed in the following sections.

The node operation is resumed in the state machine shown in Figure 3.20.

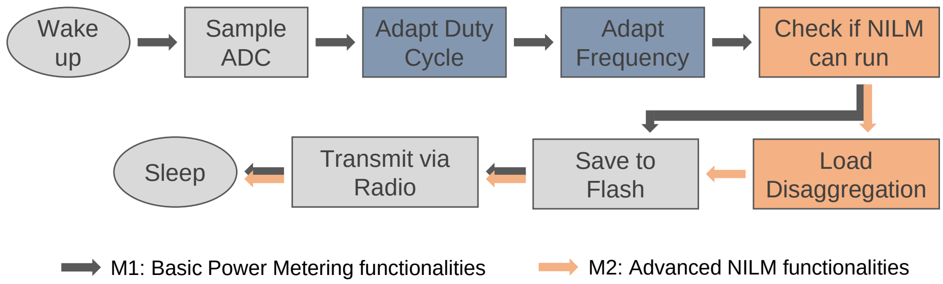


Figure 3.20: State machine of the Adaptive Firmware. Once the meter boots, after have checked if previously data is stored inside the flash, it samples the ADC and tune the firmware based on the sensed current. Data is streamed once every 5 minutes

Dynamic Frequency Scaling

To minimize the power consumption, when the primary load is small, but also to maximize the performances when we have to disaggregates loads, the frequency of the platform is dynamically adjusted during run time. The node always starts at the lowest frequency to minimize the energy expenditure and then reconfigure based on the acquired data. Frequency is scaled up and down based on the measured primary load. When we have a power increase, the frequency is slowly increased starting from 1 MHz up to the maximum frequency of 80 MHz; contrarily when we have a power decrease the platform is slowed down to preserve the previously harvested energy.

Frequency adjustment follows the transient computing idea [110]. The frequency increase is slower than the decrease, due to the intermittent nature of an energy harvesting solution.

When dealing with home appliances we can observe two behavior:

1. An appliance that has just switched ON will stay ON for some amount of time. An example is the washing machine with fixed length programs, or the coffee machine that needs a specific amount of time to make a coffee. The same happens for other appliances, like the fridge or an air conditioner-
2. The probability that an appliance that has just switched OFF will be shortly

switched ON again is low. Like the case above, after the coffee is ready, the coffee machine will not turned on again for some time.

For these reasons, when the nodes detect a power increase the frequency is ramped up slowly, only after a number of sample from when the power increase was detected. On the other side, when the measured primary load decreases, the frequency is quickly decreased to preserve the harvested energy.

Both frequency increase and decrease follow a prefixed number of step, experimentally chosen during the development of the smart meter. These steps are then saved as lookup table inside the smart meter.

Dynamic Duty Cycle

The policy that manage the node duty cycle has been developed for trying to ensure a minimum of one LoRa packet transmission every 5 minutes. To this end, first we characterized the amount of energy that can be scavenged with different primary loads. Secondly, we measured the energy needed for acquiring and computing a single RMS current sample and the energy needed for a LoRa packet transmission. With this data we then developed an adaptive duty cycle policy that decreases the sleep time of the meter proportionally to the increase of the measured primary load.

During run time, both techniques (i.e. DFS and DDC)cooperate for ensuring energy neutrality. However, DFS is activated only when the duty cycle of the node is near to zero. Only load detection task needs more computational power thus requiring to run the CPU faster. Sampling the ADC and calculating the RMS can be done with a low CPU clock without any noticeable time overhead, compared to executing the same task at a higher clock.

NILM algorithm

When the load under monitoring is over the sustainable threshold, the node configures its self to provide NILM functionalities. This process in mainly divided into three different phases. In the first phase the algorithm tries to detect if during the time interval under analysis an appliance state change (i.e. ON/OFF) has

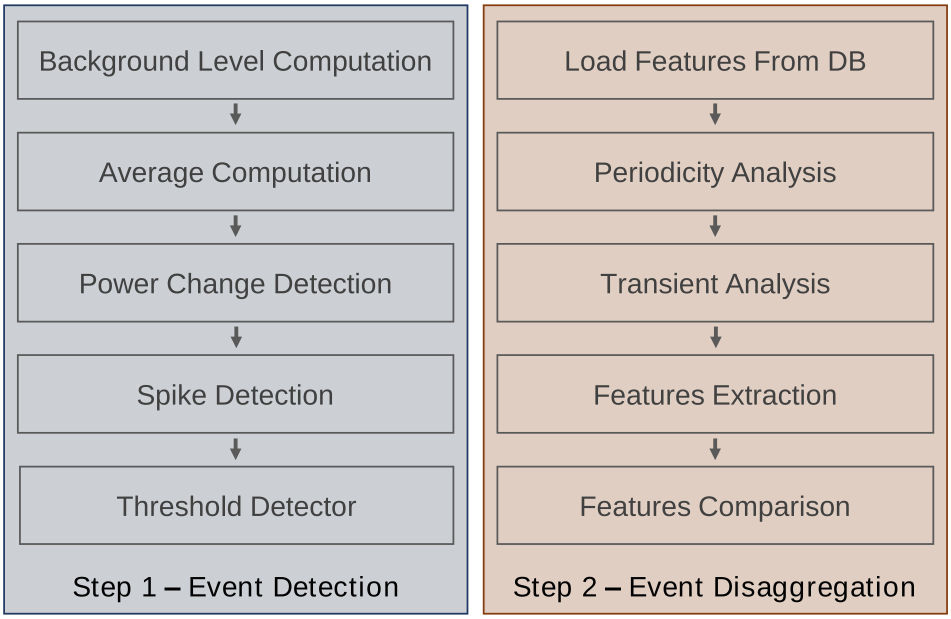


Figure 3.21: Non-Intrusive Load Monitoring algorithm implementation

occurred. This state change is defined as Event. In the second stage, the event is analyzed and some features that characterize it are extracted. In the last phase, the algorithm tries to associate that event to a known appliance, by comparing the extracted features with those already present in the appliance database. The NILM analysis implemented in the proposed device is a lightweight version of the algorithm presented in [72]. The various steps of this process are summarized in Figure 3.21. More details are discussed in section 3.4.

3.6.3 Results and discussion

In this section we evaluate different aspect of the proposed smart meter. First, we evaluate the energy requirements for the three main tasks: ADC sampling and RMS current calculation; NILM analysis; LoRa packet transmission. Once addressed the energy requirements, we focus on the energy harvesting power supply tailoring the storage capacity. Closing this section, the analysis about how the execution time and energy consumption changes by varying the CPU frequency.

Energy requirements

As stated in the previous section, a critical aspect of this type of solution is dimensioning the two storage capacitors, namely C_{store} and C_{out}. The first goal is to ensure that enough energy is available for sending at least one LoRa packet. On the other side, to avoid choosing a storage capacitor arbitrarily big that may represent a barrier for the activation of the smart meter.

In order to choose the correct value, we have measured the energy requirement for the three main tasks: *RMS current calculation*, *NILM analysis* and *LoRa packet transmission*. The result of this analysis is presented in Table 3.3.

Figure 3.23a shows the current consumption for acquiring 1000 ADC samples and calculating one RMS current sample. The node samples the ADC at a frequency of 10 kHz, acquires 1000 samples, equal to 5 periods of the AC main signal, and calculate one RMS current sample. For decreasing the energy requirements, the node uses DMA to offload the CPU, that is kept in sleep mode during the ADC acquisition. When all the samples are acquired, the CPU wakes up and calculates the RMS. Working at 1 MHz, the time required for the whole first task is around 158 mS with an energy requirement of about 123 μJ.

Current consumption for sending one LoRa packet is presented in Figure 3.23b. As can be noted, this is the most demanding task as it requires an high amount of current with an execution time in the range of seconds. Overall, the requirements for a complete cycle (RMS sampling and Packet transmission) of 5 minutes is 1.9 J.

It can be argued that this amount of energy varies based on the sleep time between the acquisition and the transmission of the data. For this purpose, we have calculated how the energy demand varies by decreasing the sleep time between two successive RMS samples (i.e. duty cycle). The graph depicted in Figure 3.22 summarize this analysis.

Last task being evaluated is load detection. This task is activated only when the main load is over the sustainability threshold, equal to 700 W. The current con-

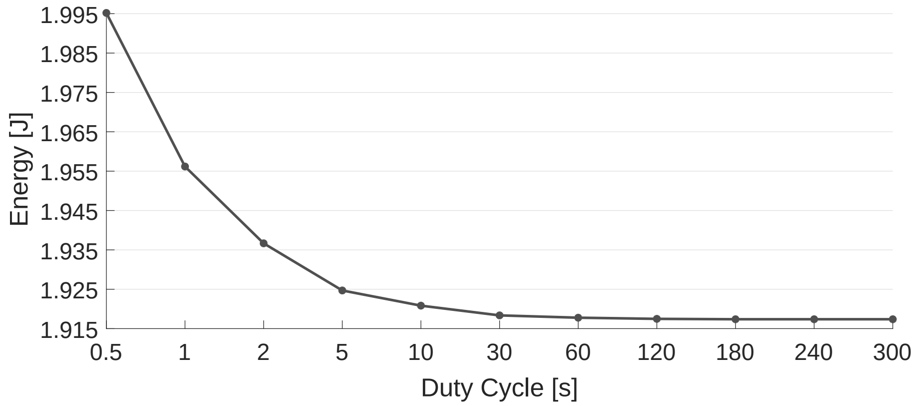


Figure 3.22: Relation between duty cycle and energy requirement for a complete cycle of 5 minutes. As the duty cycle increases, the energy for acquiring and sending the data decreases. This curve has been acquired at 3.3 V with the CPU clocked at 1 MHz.

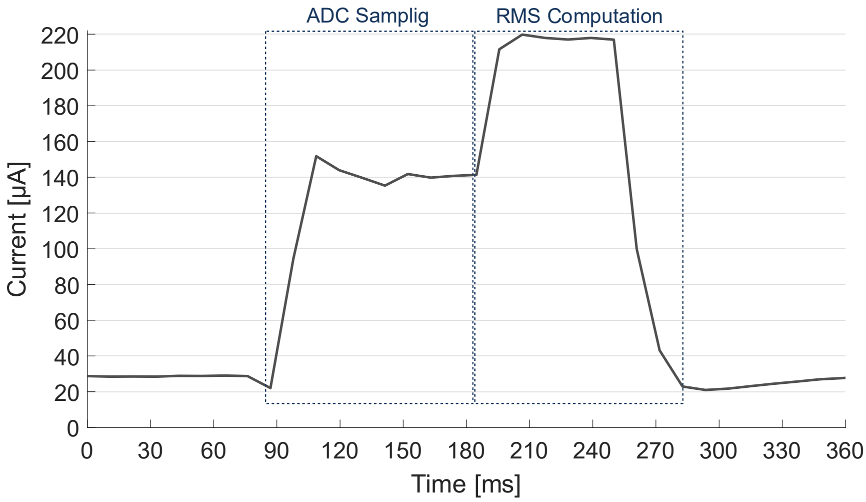
Table 3.3: Characterization of tasks execution time and energy requirement analysis

Task	Execution Time	Average Current	Energy Requirement
Sleep	N/A	23.12 μA	N/A
ADC Sampling	157.7 ms	234.92 μA	122.25 μJ
LoRa Transmission	5.43 s	105,38 mA	1,89 J
NILM Analysis	25.41 ms	2.19 mA	183.24 μJ

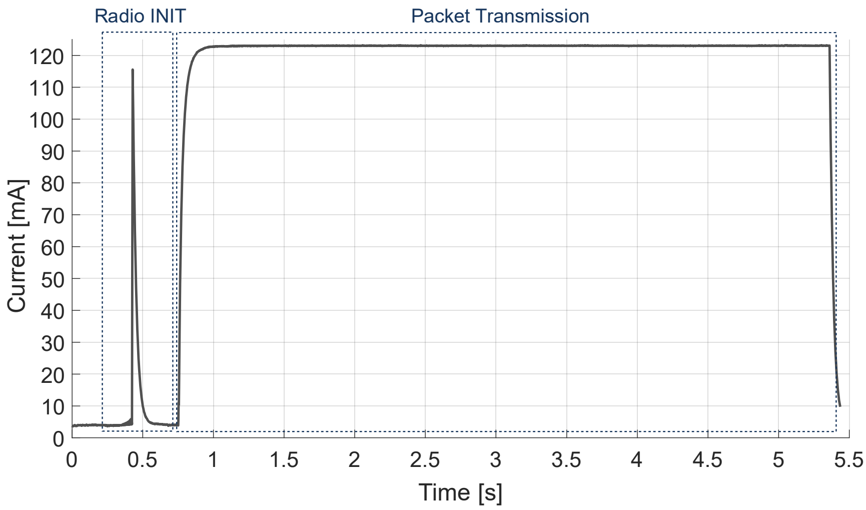
sumption for one NILM analysis is presented in Figure 3.23c. It is worth noting that this energy requirement is related to the single execution of the analysis, equal to 20 samples. This means that, over a windows of 5 minutes, this analysis is computed 30 times.

Energy harvesting power supply

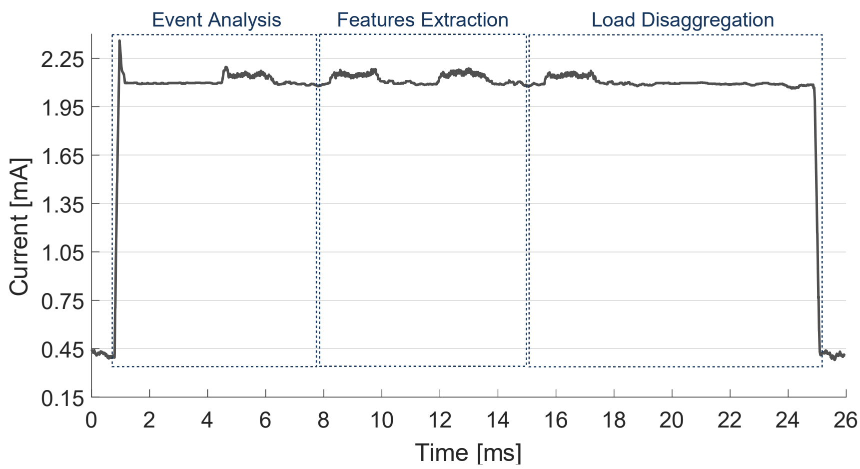
Before discussing the harvester performance, we focus on the energy reservoir selection. To size the input capacitor, C_{store}, different test has been conducted, for addressing the voltage drop due to the radio sending a packet. As highlighted, the most demanding task. Our goal was to keep the C_{store} voltage over the UVLO (Under Voltage Lock Out) threshold of the LTC3588 energy harvester, in order to keep active its output after the transmission of at least one packet. After various



(a)



(b)



(c)

Figure 3.23: Current curves for the three main tasks: a) ADC sampling and RMS calculation; b) LoRa packet transmission; c) NILM analysis

tests, we have decided to use two 12 V 90 mF capacitor connected in series in order to have a 45 mF 24 V energy storage. This permit us to:

- Have enough energy for sending one LoRa packet starting from C_{store} charge at 5.1 V. This means that, once the node boots up, we have harvested enough energy to keep the node operational for several minutes, also in the event that the main load is too small for recharging the storage capacitor;
- Exploit the full dynamics of the harvester that accepts an input voltage up to 20 V;
- Have a cold start (the time for charging the input capacitor from 0 to 5.05 V, the voltage threshold for switching ON the LTC3588 output) of less than 35 minutes with a main load as low as 100 W.

The output capacitance, C_{out}, is less critical and has been chosen to withstand the peak current of the radio, to avoid VDD line from dropping below the brownout level when the radio switches ON. In this case, a 1.5 mF capacitor is used.

To evaluate the harvester performance and the amount of energy that can be scavenged, we have emulated different loads by means of a variable power resistor and collected different charging curves of the 45 mF 24 V super capacitor used as energy storage.

The results are summarized in Figure 3.24.

Sustainability

After the evaluation of the energy requirements of the various tasks and the energy provided by the harvester at different main loads, we have derived the sustainability curve. This curve expresses the relation between the main primary load and the duty cycle between two successive RMS current samples, in order to achieve a LoRa packet transmission every 5 minutes. Result shows that the node is able to start providing a current measure every half second starting from a main load as low as 400W. With regards to NILM functionalities, experimental result has shown that they can be enabled starting from 700 W.

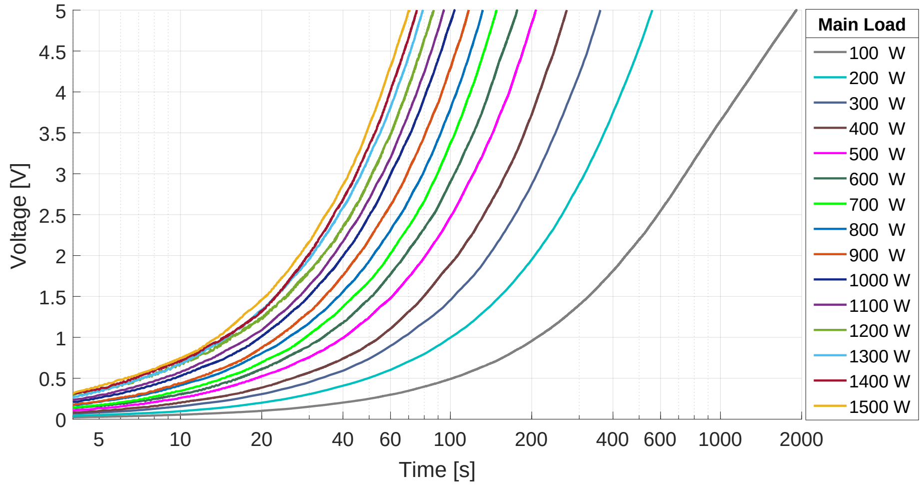


Figure 3.24: C_{Store} charging time with different primary main loads. Curves are plotted using semi-logarithmic scale.

Figure 3.25 the voltage trend of the supercapacitor while harvesting energy from a 400 W primary load. We can see that the cold start is reasonable and under 3 minutes.

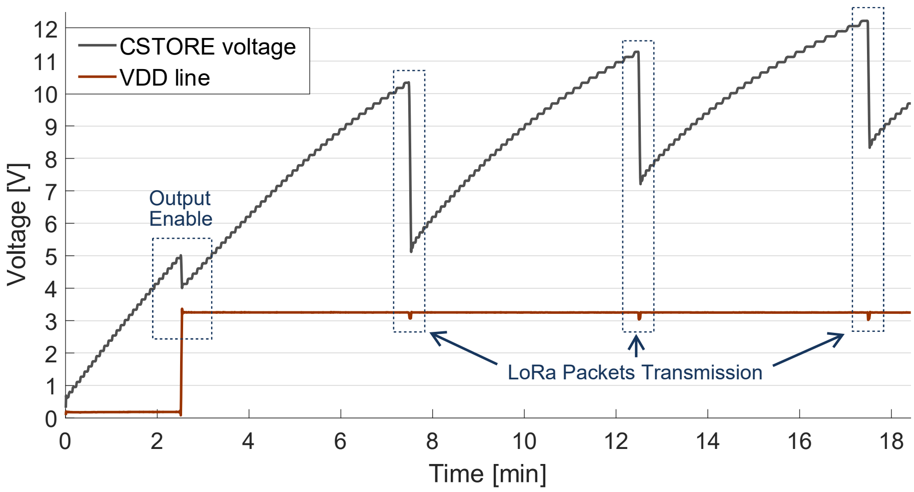


Figure 3.25: Voltage trend of C_{Store}. Measure are made with a packet transmission every 5 minutes, sample rate of 2 Hz with the CPU clocked at 1 MHz.

Frequency scaling analysis

To complete the sustainability analysis, we have investigate the relationship between execution time, CPU frequency and current consumption. We have evaluated only ADC sampling and NILM Analysis tasks, as the energy requirement for the third task, LoRa transmission, is independent from the CPU frequency. This analysis, however, is out of the scope of this section and will be presented in section 4.2.1. During our tests, we configured the radio to achieve the maximum range with the maximum payload length. We used SF 12, BW 125 kHz, CRC 4/5 with a payload of 51 bytes and a transmission power of +18 dBm.

The board is configured for using two different clock sources, respectively LSE (Low Speed External oscillator) and MSI (Multispeed internal RC oscillator), one for low-power operation (i.e. stop and sleep mode) and the other one at run time, with a maximum frequency of 48 MHz. The node is configured to work at 80 MHz only for ensuring that the voltage inside C_{store} stays within the operational range of the energy harvester. In this case, the node also activates the PLL for reaching STM32L4 maximum frequency of 80 MHz, increasing the overall energy consumption, thus discharging the energy reservoir.

We have chosen to use this two clock sources because they perfectly cope with this application scenario. LSE permit to time the operations also when the MCU is in stop mode. The MSI current consumption is proportional to the frequency generated, perfectly suited for all those solutions based on an energy harvesting power supply with tight energy requirements.

Results of this analysis are presented in Table 3.4 and in Figure 3.26. As can be noted, we have a different behavior for the two tasks. For the first one, RMS current calculation, the time required for executing the task proportionally decreases as we increase the working frequency while the energy requirement increase. This behavior is mainly related to the consumption overhead introduced by the CPU running at higher frequencies, as we have a constant time of 100 mS in which we sample the ADC, where the CPU is actually doing nothing but still powered. This means that increasing the frequency does not introduce any improvement in the execution of the task, but only an additional energy overhead. In the case of the second task, we have a different result than expected. As the frequency increase, both time and energy requirement decrease, till we reach the frequency of 16 MHz, where the energy requirement start rising again. In this case, increasing the frequency lead to a higher computational efficiency, till we

Table 3.4: Relation between frequency, execution time and Energy Requirement for the two main tasks

Frequency [MHz]	RMS current calculation		NILM analysis	
	Execution Time	Energy Requirement	Execution Time	Energy Requiremen
1	157.7 mS	122.25 μJ	364.67 mS	420.69 μJ
2	131.3 mS	138.74 μJ	179.49 ms	291.46 μJ
4	116.39 mS	189.25 μJ	90.26 ms	226.98 μJ
8	110.04 mS	308.02 μJ	46.6 ms	195.59 μJ
16	106.12 mS	550.51 μJ	25.41 ms	183.24 μJ
24	106.69 mS	790.55 μJ	18.55 ms	214.76 μJ
32	107.02 mS	1017.83 μJ	14.64 ms	209.05 μJ
48	105.88 mS	1483.74 μJ	11.63 mS	208.08 μJ

reach the optimal frequency of 16 MHz, where the energy requirement starts rising again with the frequency.

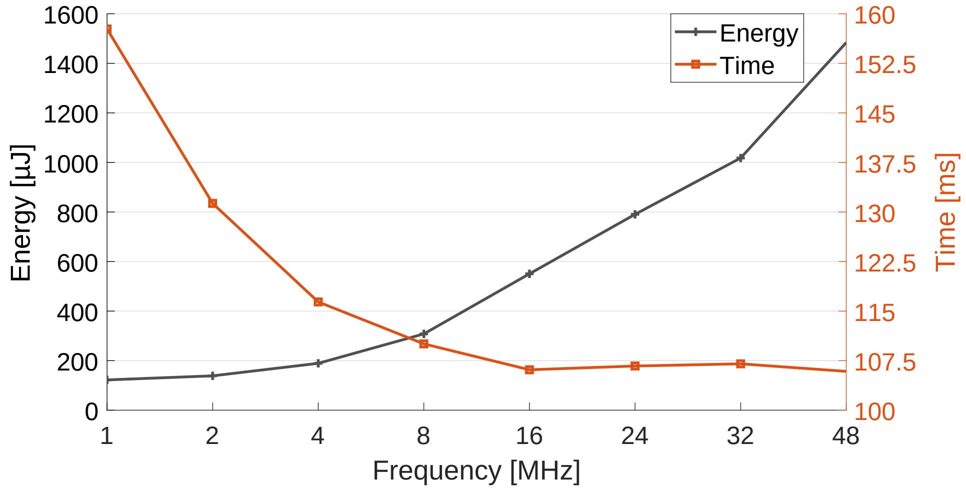
We have decided to analyze only frequencies up to 48 MHz, as a higher frequency is activated only when we need a high consumption to ensure C_{store} voltage stay within the operational range of the energy harvester (i.e. lower than 20 V). Running the node at higher frequencies does not bring any benefit in the execution time. Contrarily it introduces additional energy overhead worsening the energy budget.

3.6.4 Summary

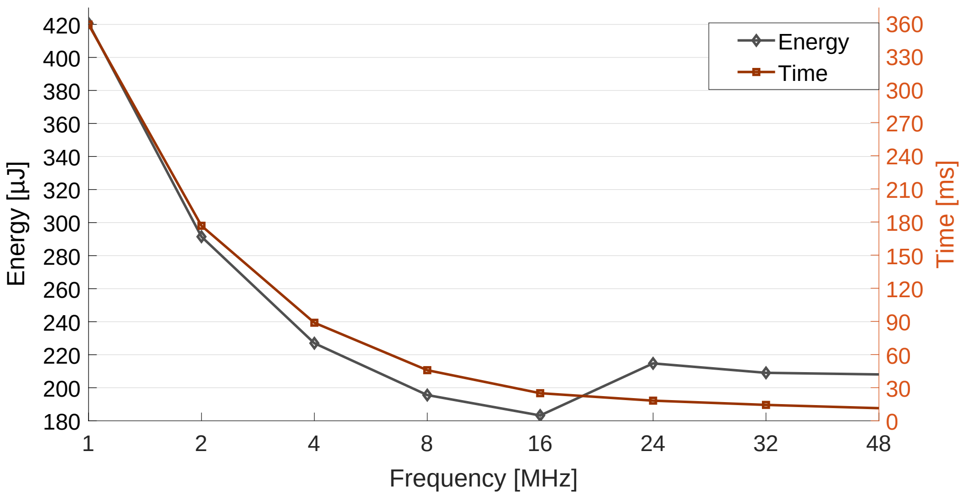
To address some challenges of the existing power meters, in this section, an easy-to-install self-powered smart meter, has been presented. It exploits two current sensors, one for metering and one for harvesting. It is able to provide basic metering functionalities and also Load Monitoring analysis relaying just on the scavenged energy from the same load under measurement. The result of the analysis is then streamed up to kilometers away, thanks to the use of a low power LoRa radio.

Experimental results have shown that the proposed solution is able to perpetually self-sustain its operations starting from a main load as low as 50 W while still providing one RMS current reading every one minute. This removes maintenance cost for battery replacement and also unwanted standby consumption, as the energy drawn by the meter is proportional to the load under monitoring.

This could represent a key element in lowering the barriers for the widespread of a distributed energy metering infrastructure.



(a)



(b)

Figure 3.26: Comparison of energy/time vs CPU frequency for both the RMS computation (3.26a) and the NILM analysis (3.26b). As can be noted, for the RMS computation, the energy overhead of running the CPU at higher frequency does increase the energy requirements, event if the task takes less time to complete. Downclocking the CPU allows to lower the energy requirement up to 8 times while still ensuring a sample in less then 160 ms. On the contrary, for the NILM analysis, both time and energy requirement decrease by incrementing the CPU clock, up to the optimal frequency of 16 MHz. After this point the energy efficiency drops.

3.7 Conclusion

Improving the strategies to generate, distribute and use the energy produced is one of the main challenges that smart cities are currently trying to address. A sustainable energy transition means substantial changes in technology, but also a change in human behavior and policies. Before this can happen, a solid smart metering infrastructure must be built in order to monitor and give real-time feedback about energy usage.

Providing appliance-specific electricity consumption insight enables end-users to identify their saving potential and helps utilities company to better manage the production and distribution. Starting from the information provided by non-intrusive load monitoring approaches, power consumption patterns can be studied and analyzed. Innovative ideas for optimizing the grid load – like demand side management – and to provide real-time feedback to households about how they can optimize energy consumption can be studied and implemented.

In this section a range of solution to improve the way energy consumption is being monitored is presented.

The first example is presented in 3.4 showing that is now possible to run NILM analysis directly on low-power smart meters in order to provide real-time feedback regarding per-appliance energy consumption. This approach brings many improvements to the classical technique that reckon on cloud services to provide NILM analysis. By developing and running simpler algorithms inside a power meter we can fully exploit the meter resources, making the whole metering infrastructure more efficient and also more robust to system failures (i.e., if the cloud server facility encounters a shortage, the metering infrastructure of a whole town could be taken down). At the same time, we lower network pressure, as we are not required to transmit all the acquired data, but just the result of the analysis. Finally, privacy is ensured by the ability to decide if we want to share consumption patterns with the utility company or other parties.

In section 3.5 the focus has been moved to energy harvesting solutions. Here, the complete study of a battery-free, non-intrusive power meter for low-cost energy monitoring is presented. With this approach we can provide a device that lowers maintenance cost related to installation and battery replacement; does not need to deal with high main voltage; and does not introduce any additional energy consumption overhead as it draws zero-power under zero-load condition. Moreover, thanks to the use of a LoRa radio it offers also the ability to supervise remote devices, without battery support nor human supervision.

Combining the strategies previously presented (i.e., local NILM analysis and energy-harvesting) have led to the development of the solution presented in 3.6. The proposed power meter exploit two current transformer sensors to respectively monitor energy consumption and to harvest energy. The development has followed a hardware-software co-design approach allowing a flexible device that can be future adapted to different scenarios. The key aspect is represented by the modularity of the system, that abstract the software layer from the hardware layer. Both components can be interchanged and adapted to the deployment scenario. The software provides both basic meter capabilities and load monitoring, based on the sensed energy consumption. The processing pipeline is able to reconfigure during run time in order to guarantee energy neutrality. In the same way, the energy harvesting power supply is agnostic to the platform and can be used with other sensor nodes. For example, the same device can be used to monitor the fingerprint of a single appliance in a predictive maintenance scenario, by just changing the processing algorithm [109], [111].

Concluding it was demonstrated that the proposed design is able to meet all the design requirement that a pervasive sensing device should have. It is low-cost and easy to install to lower installation barriers. It is fully autonomous and does not require any supervision after deployment, lowering maintenance costs. It flexible and modular to be adaptable to the environment where being installed.

Chapter 4

Edge Machine Learning for the IoT

4.1 Introduction

Deep Neural Networks (DNNs) have become predominant for image recognition and for other pattern and task detection like speech and natural language processing. More recently, also to build strong analytic tool for huge volumes of data [112]–[114].

DNNs, or deep learning, refers to a specific class of algorithms called neural networks in which learning tasks are broken down and distributed onto machine learning algorithms that are organized in consecutive layers. In other words, a deep neural network is a graph with a series of fully connected layers in which every node in a particular layer has an edge to every node in the next layer. Together the layers constitute an artificial neural network that mimics the distributed approach to problem solving carried out by neurons in a human brain. Figure 4.1 presents a graphical representational of both a neural network, which usually has only one hidden layer, and a deep neural network with multiple hidden layers.

DNNs have drastically increased the recognition accuracy with respect to traditional methods [115]. This is associated with an exponential growth of network size which comes with significant computational complexity and memory con-

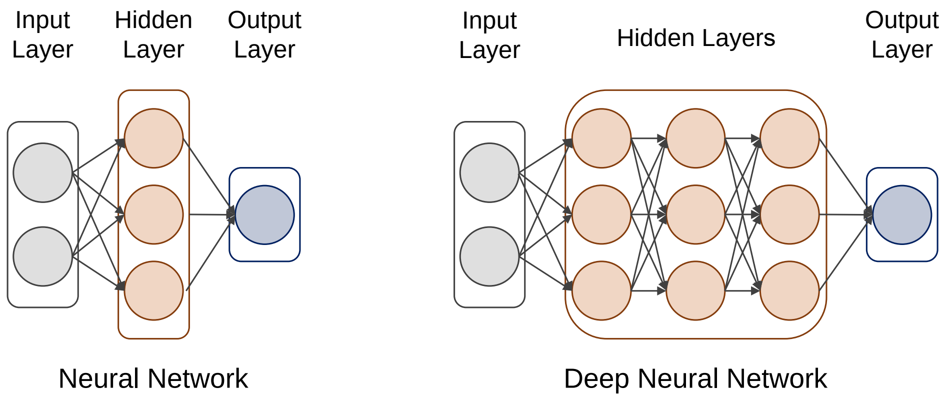


Figure 4.1: Construction of a neural network model.

sumption [116] making it, until recently, feasible only on power-hungry server platforms in the cloud.

Deep learning can enable IoT devices to interpret unstructured multimedia data and intelligently react to both user and environmental events. Unfortunately it has demanding performance and power requirements. Optimized implementation targeted to IoT platforms that have limited area and power budget are being investigated [117]–[122]. However, the performance and energy budget of edge nodes in which the inference (the application of a trained network to new data) is executed locally, still makes it a challenging task [123].

4.2 The need for inference on the edge

Deep learning is a promising approach for extracting accurate information from raw sensor data from IoT devices deployed in complex environments. In conventional cloud computing, all data must be uploaded to centralized servers and, after computation, the results need to be sent back to sensors and devices. This process creates high pressure on the network, specifically in the data transmission costs of bandwidth and energy resources [124].

A more critical situation arises for IoT applications that are time-sensitive and that poses hard constraint in the communication delay (i.e., smart transportation [125], smart grid [107], [108] or smart city [126], [127]). It can be argued that conventional cloud computing-based service definitively cannot satisfy the demand [128]. This because the data to be analyzed first needs to be uploaded to the cloud and then sent back to the end-node. The result will be a large latency in the end-to-end application, unacceptable for time-sensitive IoT architectures. Furthermore, most IoT devices have limited power or rely on energy harvesting. Avoiding un-necessary wireless communication would extend their lifetime.

4.2.1 The communication burden for energy harvesting enabled IoT nodes

Many applications today offload most computation to the cloud by sending acquired data to a remote server and waiting for a response.

Unfortunately, communication is not energy free.

While LoRa technology is a key enabler of pervasively deployed IoT devices, due to the long range and low power characteristics, the energy required still dominates the device's total operating energy; even at low communication duty cycle. To this end, we have characterized the behavior of a common LoRa module[1] for sending a single 255-byte packet, using different spreading factor and bandwidth. Figure 4.2 summarizes the results of this experiment. The graphs present the relation between the achievable sensitivity - the lower, the better - in regard to the energy cost of transmitting. The data shows that sending the longest distances possible (higher sensitivity) requires an order of magnitude increase in energy cost compared to the shortest distances (lower sensitivity).

When dealing with energy harvesting, collecting the energy to execute the different tasks, i.e. sensing, processing and transmitting, takes time. Assuming an intermittent operating model that charges it capacitor fully before executing tasks, this time might be not negligible. Consequently, the energy cost also became a latency cost. This problem is well investigated by Gobieski et al. [29]. Results highlights how sending a single 28x28 image would take over an hour while performing inference locally just 10 seconds when operating by harvested RF energy - an improvement of more than 360x.

[1]https://www.hoperf.com/modules/lora/RFM95.html

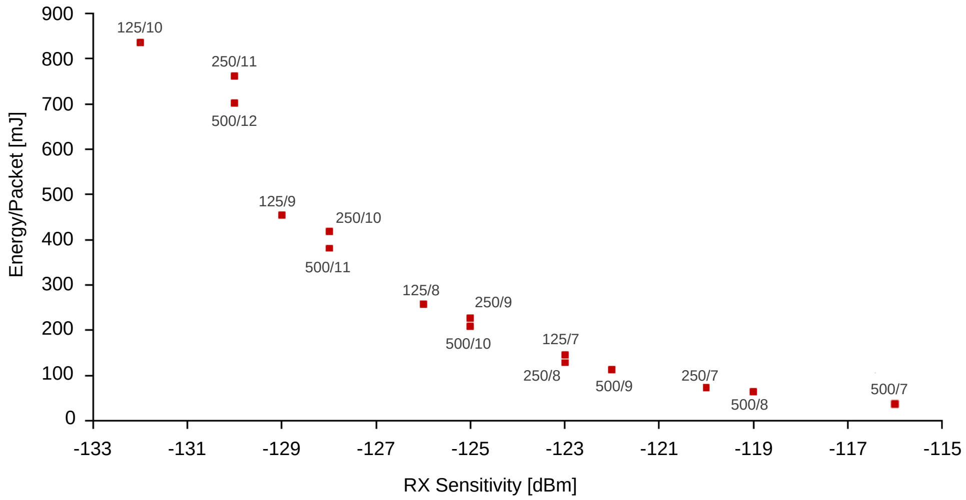


Figure 4.2: Relationship between LoRa radio sensitivity and energy requirement for transmitting a packet of 255 bytes. Each point is presented along with the transmission parameter, [BW/SF].

Depending on the application, even larger end-to-end improvements are possible by sending only the result of inference rather than the full sensor reading. For instance, in an object detection scenario, the IoT device could just send a single small packet when an interesting object is detected, rather than the full image.

Edge Processing has become increasingly popular due to its power savings capabilities. Efficient use of data compression and local data analysis can reduce the required transmission bandwidth and thus the time for harvesting the required energy. IMote2 authors [129] have achieved a 48x reduction in energy cost by performing FFT analysis on edge sensor node instead of offloading all the data to the cloud. This is confirmed by several works conducted in the last few years [130]–[135]. A theoretical framework for applying machine learning to the IoT can be found in [136]

4.3 Energy neutral IoT device for precision agriculture

Monitoring the number of insect pests is crucial in pheromone-based pest management systems as pest infestation is one of the main factors that affect harvest losses [137], [138]. Currently, the most common method of measuring insect infestations is to manually identify and count the insects; captured digital images are analyzed by human experts, or farmers, for recognizing and counting pests. However visual inspection is labor intensive and inefficient; therefore, subjective factors can affect the accuracy of population counts.

With the development of information technology, the researchers have proposed to use computer vision techniques for automatically identify and count agricultural pests. The same digital images used for manually identify pest infestation are being used to train machine learning algorithms for automatic disease detection [139]–[144]. Once trained, intelligent visual IoT devices can be deployed directly in orchards for autonomously monitoring dangerous parasites.

Apples are among the topmost fruit crops of the world, and apple orchards are widely expanding in many regions and countries. The most common problem for these crops is the attack of the codling moth, which is a dangerous parasite for apples.

In this section an Energy Neutral smart IoT device for pest detection in precision agriculture is presented and discussed. Near sensor ML algorithms automatically detect the codling moth and eventually sends a notification to the farmer if any is detected. The application is developed on a low-energy platform powered by a solar panel of a few hundred square centimeters, realizing an energy autonomous system capable of operating unattended continuously over low power wide area networks.

4.3.1 System overview

The prototype system is presented in Figure 4.3. It is based on a Raspberry PI 3[2] that provides the environment for acquiring and processing the captured pictures. To speed up the DNN evaluation, the node uses an Intel Movidius Neural Compute Stick (NCS)[3] providing an Intel Myriad X neural accelerator as a Vision Processing Unit (VPU). For streaming the result of the analysis, a LoRa modem provide connectivity to the platform

[2]https://www.raspberrypi.org/products/raspberry-pi-3-model-b/
[3]https://software.intel.com/en-us/neural-compute-stick

Figure 4.3: Prototype overview. Starting from the left we can see the camera module along with the Intel NCS below. On the right the 3D printed case enclosing the Raspberry Pi 3 and LoRa module.

The system has been designed to bring IoT technologies in agriculture where there is the need for collecting the output over vast areas requires long-range communication and, possibly, from multiple devices. Thanks to the onboard intelligence, the output of the smart trap is limited to the few bytes. Only the result of the DNN evaluation have to be streamed, making it suitable even for low bit rates communications. If the farmer needs a visual confirmation from the captured picture, a few images per day can be transmitted as well. This justify the use of a low-power wide area network (LPWAN) technologies, more specifically the long-range WAN (LoRaWAN) protocol.

4.3.2 Images collection and dataset creation

To create the dataset, and to test the effectiveness of the proposed solution, the smart camera was installed inside a normal pheromone-based trap to capture pictures of the insect glued on the trap. An example of the smart trap is presented in figure 4.4b, along with a common trap showed in 4.4a. An example of a photo taken inside the trap is presented in figure 4.5a. Once captured, images are processed in-situ to separate each insect, for creating sub-tiles from the original picture. This step is essential since it filters the raw pictures and produces tiles that contain only one insect as shown in Figures 4.5b and 4.5c.

(a)

(b)

Figure 4.4: Codling moth traps: (a) commercial trap; (b) prototype of the IoT neural network codling moth smart trap. Source [139].

(a)

(b)

(c)

Figure 4.5: (a) Raw photo capture by the proposed system. (b) and (c) An example of the tile created after the processing of the raw photo. Source [139].

The dataset initially used to train the DNN contains approximately 1500 pictures and was lately combined with more insects' images acquired during the preliminary on field experiments. After the acquisition, images have been labeled, creating two different classes. The first one containing only codling moth pictures; the second one representing generic insects. Once labeled, the figure are used to train a TensorFlow model[4] based DNN. We used the VGG16 model developed by Oxford University [145], later converted to a graph model suitable to be analyzed with the Myriad VPU.

During the training phase, the algorithm efficiently exploits features such as color (dark subjects on a white background) and the shape of the insects with a Blob Extraction Algorithm. This process involves 4 main steps:

1. Conversion of the frame from RGB to gray scale
2. Smoothing (or blurring) of the frame with a Gaussian filter
3. Edge extraction through a Canny operator
4. Some dilation and erosion of the picture

After these steps, the blobs are detected through the OpenCV[5] blob detector.

4.3.3 Training

For the training stage we used a custom TensorFlow library[6] that allows the rapid development of a neural networks for image classification. This training phase consists of an offline process that adjusts the neural network weights using the dataset of labeled images previously created. In this way, the system can learn to classify the images. The dataset is composed by 1500 images; 1200 images were used for the training while the remaining 300 for the validation.

When dealing with a training phase, the main parameters that define the dimension of a network are the number of neuron (or node) that multiplies by weight values the input signals, and the number of hidden layers through which

[4] https://www.tensorflow.org/
[5] https://opencv.org/
[6] https://github.com/frank1789/NeuralNetworks

the image is filtered. The training phase adjusts the weight values, while some parameters, such as the number of epochs and the image size, are imposed by the user.

Epochs represent the number of times all of the training vectors are used once to update the weights. Each epoch finishes with a validation step that evaluates the ongoing training process. A good trade-off between the number of epochs and image size is necessary for a correct training stage, as the overall training process and final network dimension is highly dependent from those parameters. It also helps to avoid over-training problems [146] and to meet the hardware constraints.

To this end, 3 different training phases has been assessed, with the following configuration:

1. 75 epochs with an image size 224 x 224 pixels
2. 10 epochs with an image size 112 x 112 pixels
3. 10 epochs with an image size 52 x 52 pixels

The results of the training tests are presented in Figure 4.6

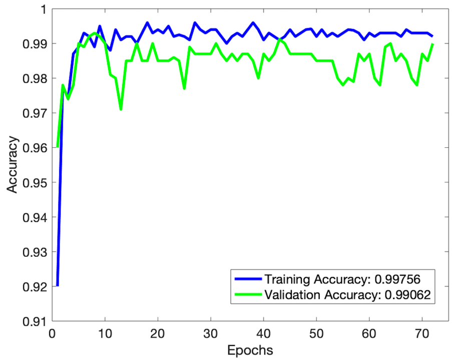
As can be noted, training and validation accuracy using 75 epochs saturates. This suggests that the number of epochs can be decreased without sacrificing the network accuracy. As shown, 10 epochs are enough for the target accuracy.

Moreover, to avoid possible overflow in the Movidius NCS and to save memory on the Raspberry PI 3, the image size can be decreased, providing a simpler model that can meet the hardware constraints.

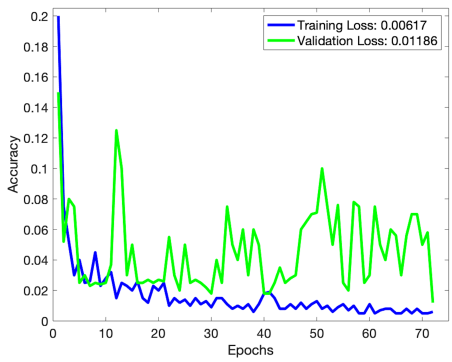
Small images clearly show worse performance with respect to bigger tiles. Nevertheless, about 98 percent accuracy has been achieved, that satisfies the requirements expected by farmers for an IoT application for parasites monitoring.

4.3.4 Validation and test

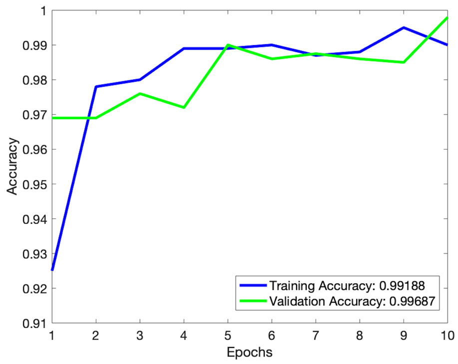
The DNN provides a confidence measure that indicates how close the detected object is to a general insect or the target Codling Moth. Figure 6 shows an example of the output from the classification.



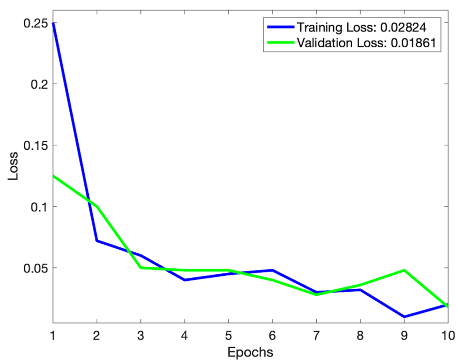
(a) 75 epochs, image size 224x224.



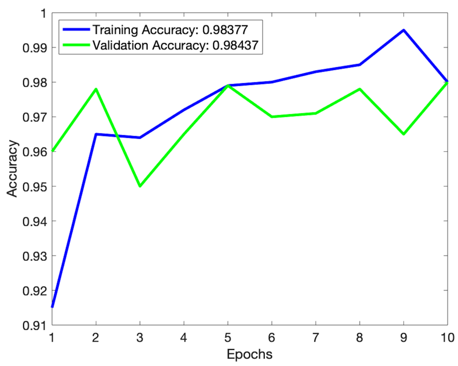
(d) 75 epochs, image size 224x224.



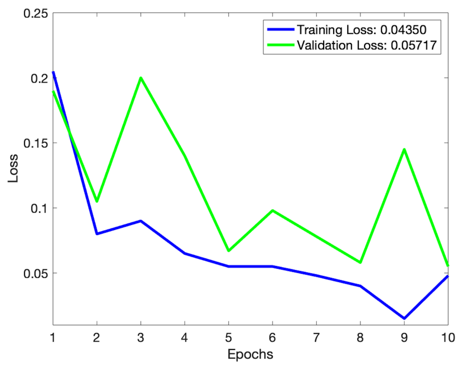
(b) 10 epochs, image size 112x112..



(e) 10 epochs, image size 112x112..



(c) 10 epochs, image size 52x52.



(f) 10 epochs, image size 52x52.

Figure 4.6: Training results. On the left the validation accuracy. On the right loss function accuracy

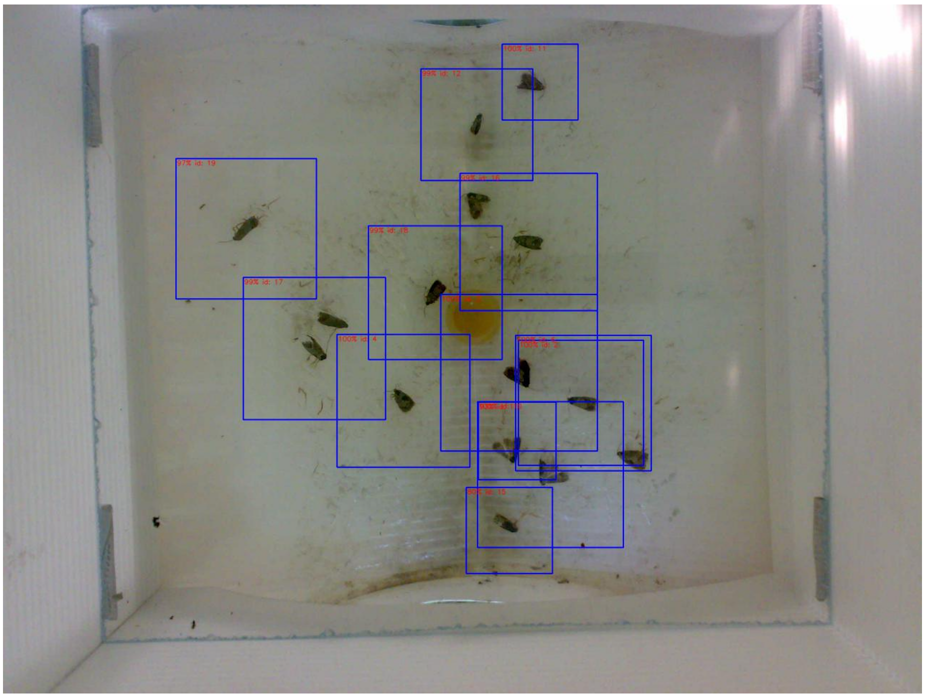


Figure 4.7: Example of the results produced by the recognition algorithm. Detected codling moths are highlighted with a blue box along with the confidence value. Source [139]

The tests of the DNN model were carried out during a 12 weeks window in an apple orchard with the insect glue trap shown in Figure 4.7. The result of the experiment is reported in table 4.1

Table 4.1: Recognition results during the 12 weeks on field experiment.

Number of detected insects	True Positive	False Positive	False Negative	Unclassified	Precision	Recall
262	80.6%	4.8%	6.4%	8.2%	**94.38%**	**92.6%**

As can be noted, out of 262 insects, the network has scored a 94.38% accuracy.

4.3.5 Power consumption analysis

In order to assess the power requirements of the proposed system, we have evaluated the power consumption of the overall system's classification. The result is showed in Table 4.2 that presents the single task breakdown, while figure 4.8 shows the power trend during the execution of the different tasks. As can be noted, the last two tasks – T3 and T4 – are the most power-hungry. Task 3 combines the usage of the Raspberry and the Intel Movidius, while task 4 also encompass the consumption associated to the radio module.

Table 4.2: Energy consumption and execution time breakdown

Task #	**Description**	**Execution Time [s]**	**Current [mA]**	**Energy [J]**
0	Boot	43.68	345	75.35
1	Image Capture	3.45	394	6.8
2	Preprocessing	4.07	501	10.19
3	Classification	10.19	525	26.75
4	Report transmission	0.34	525	0.89

In apple orchards the presence of codling moth is usually checked twice every day, by the farmer. To automate this process, the system also integrates a nano-watt Real-Rime Clock (RTC) that wakes up the raspberry when planned. After the report has been generated and transmitted, the system shuts down, zeroing its power consumption, waiting for the next cycle.

The total energy requirement for a single application cycle is 119.98 J. Thus, a 9000 mAh battery is sufficient to sustain the system for more than one year, with two wakes-up per day. Moreover, when combining the system with a 0.5 W solar panel of a few hundred square centimeters, as presented in [147], the energy intake will be enough to permit the smart camera to operate unattended indefinitely.

It worth to be noted that the overall energy requirement is dominated by the boot phase, that takes several seconds to complete. By optimizing the boot phase or by using a faster storage media (i.e. higher micro-SD class) it would be possible to boost the life time of the device when operated only by battery.

This is a key aspect that all those devices expected to be spread over large areas should have, representing a breakthrough for agricultural activities. The farmer could use smart IoT insect traps, forget about their maintenance, and wait for only automatic alerts if a codling moth is captured.

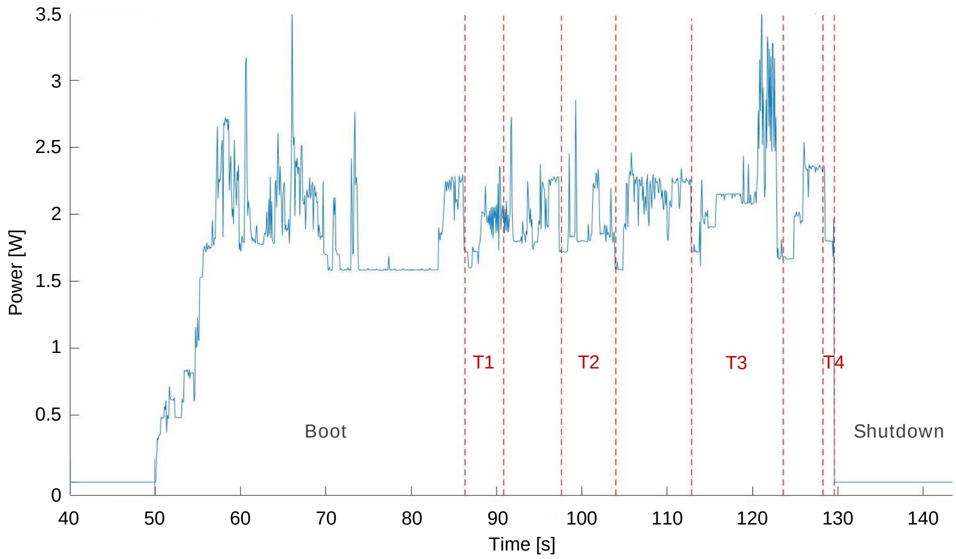


Figure 4.8: System power consumption trend.

4.3.6 Summary

Computer vision systems are already widely employed in different segments of precision agricultural and industrial food production. In this section an effective solution to automate the detection of codling moth, a dangerous parasite for apple orchards, is presented and discussed. Even though the proposed system does not use ultra-low-power microprocessors or microcontrollers, it was demonstrated that the system can still be suitable to be powered by harvesting needed energy from environment.

Due to the low cost of the hardware, this type of system can scale to several installations in an apple orchard. In this way, it is possible to use treatments for codling moth only when the system detects threats for crops, optimizing the use of chemicals and mitigating their impact on the environment. Also saving time and money for human intervention that are no longer required to check insects traps every day.

4.4 Battery-less Long-Range Visual IoT System

Future applications such as enhanced situational awareness, wildlife and smart city monitoring require more sophisticated sensing, computing, and communication capabilities. The ability of acquiring video and image data to directly, rather than indirectly, observe complex environmental phenomena must be supported. Furthermore, some application scenario might demand deployment into environments requiring wireless data backhaul at kilometer scale.

These requirements (i.e., high-data-rate sensing and long-range communication) pose a key challenge for an energy-constrained system. New strategies to avoid un-necessary data transmission to optimize both scarce bandwidth and energy must to be investigated.

As highlighted in section 4.2, emerging IoT devices powered by energy harvesting can nowadays apply sophisticated deep-learning algorithms to understand their environments and to avoid unwanted wireless communications. However, enabling deep learning on the device side is still very challenging. Indeed, the main characteristic of these IoT devices is low power, which usually means limited computing power and small memory size.

In this section, Camaroptera [148], an energy-neutral Long-Range IoT Visual Sensing System is presented and discussed. Exploiting an ultra-low-power image sensor and a LoRa radio, the device is able to acquire visual data from the environment where it is installed and to communicate it over long distances, even in the presence of urban signal occlusion[28].

Camaroptera is battery-less and harvests its operating energy using small solar panels, storing energy in a small supercapacitor. Energy neutrally is endowed by a flexible software pipeline architecture that allow deploying signal processing (e.g., compression, filtering), and machine inference using CNN or DNNs in situ.

Camaroptera's dynamic pipeline adapts to environmental signals (power, base station availability) and can dynamically vary software and radio operating parameters to vary end-to-end latency, power, transmit distance, and received data quality. This enables the device to substantially reduces energy costs while improving the quality of service of the application.

4.4.1 System Overview

Camaroptera is a battery-less sensing, computing, and communication system composed of a custom hardware platform, application-level software components, and software control components. Figure 4.10 present the schematic block overview of the proposed IoT device.

Each Camaroptera device is built on a custom hardware platform that includes: a small, low-power image sensor to collect images; a microcontroller with and embedded non-volatile memory to process images; a long-range radio chip for communication; and a solar energy-harvesting power system for collecting and storing energy from the environment. Figure 4.9 shows the prototype, composed of several printed circuit boards (PCBs) that assemble into a three-dimensional device.

A simple operating system and device driver layer manage the sensor main functionalities: data collection; re-configurable at-sensor processing pipelines to process collected images; and data transmission. The processing pipeline can include arbitrary application-specific operations (e.g., CNN/DNN-based image classifiers [29]]), and built-in operations, which include video differ detection, and image compression.

As prior work has observed [29], [149] the key determinant of end-to-end latency in an energy harvesting system is the time to collect the energy required to perform a task. With low harvested input power, energy collection time dominates and a system design must optimize for end-to-end energy-efficiency.

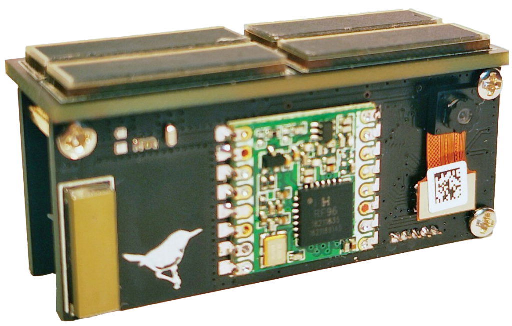


Figure 4.9: Camaroptera hardware prototype

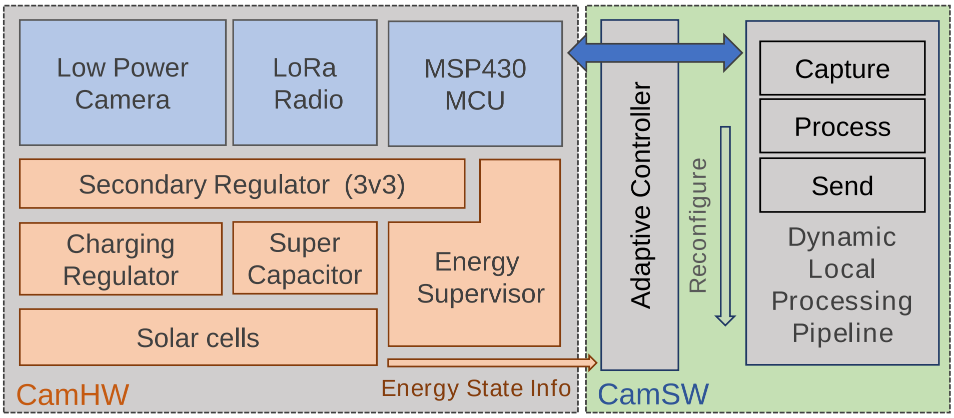


Figure 4.10: System overview of Camaroptera

To this end the camaroptera's operating system monitors the device's input power using dedicated hardware. Based on the input power and a measurement-based model of the cost of different processing and transmission operations, the OS decides how to process and send collected images.

4.4.2 Hardware implementation

The platform is designed for sensing, computing, and long-range communication. It is composed entirely of COTS components, to limit the per-device cost. The platform is composed by 3 small 2-layer PCBs, connected to one another in a three-dimensional package, namely sensor board, power board, and solar board. Figure 4.11 shows a photograph of the populated sides of the three boards

Sensor board

The sensor board incorporates the main active components of the sensing platform: 1) An MCU which manages the whole device, the camera sensor that provide visual data and finally a LoRa Transceiver with a ceramic antenna for data streaming. This board can be operated standalone, by just providing a stable 3 V supply line. This makes the platform completely agnostic to its power system, that can be customized based on different requirements (i.e. constant power supply or a battery).

MCU. At the hearth of the device, a Texas Instruments MSP430FRx Ultra

Low-Power FRAM Microcontroller. More precisely, the board is powered by an MSP430FR5994[7] capable of running at 16 MHz integrating 8K SRAM and 256 Kb of embedded FRAM non-volatile memory. Thanks to the low-power capabilities and non-volatile memory, it is a perfect candidate for battery-less nodes, where the amount of available energy is limited and intermittent.

Image Sensor. Camaroptera uses a Himax HM01B0[8] image sensor. The HM01B0 is an ultra-low-power CMOS image sensor that enables the integration of computer vision applications into resource constrained devices. The sensor utilizes 3.6 μm pixel technology that offers sensitivity of below 1 lux and only need three external passive components to work. It integrates black level calibration circuit, automatic exposure and gain control loop, to reduce host computation and commands to the sensor to optimize the system power consumption. HM01B0 contains an active area of 320 x 320 pixels and support QVGA and QQVGA window mode. Camaroptera configures the camera to operate in QQVGA (160x120) mode, capturing 8-bit grayscale images.

Transceiver. Camaroptera is equipped with a LoRa module. The connectivity is provided by a RFM95W[9] transceivers featuring a LoRa low-power modem and a +20 dBm power amplifier that can achieve a sensitivity of over -148 dBm. The LoRa IC is then connected to a ceramic chip antenna with a maximum gain of 3.42 dBi.

Power board

The power board implements the energy harvesting power system. Camaroptera power supply is based on a double stage boosting circuit with hardware voltage comparators used to keep the second booster in its most efficient operating region and to provide a stable voltage to the sensor board. On the back of this PCB we find the surface mounted supercapacitor-based energy storage.

Boosting stage. The first boosting stage connects the solar panels to the super-

[7] http://www.ti.com/product/MSP430FR5994
[8] https://www.himax.com.tw/products/cmos-image-sensor/image-sensors/hm01b0/
[9] https:www.hoperf.com/data/upload/portal/20190801/RFM95W-V2.0.pdf

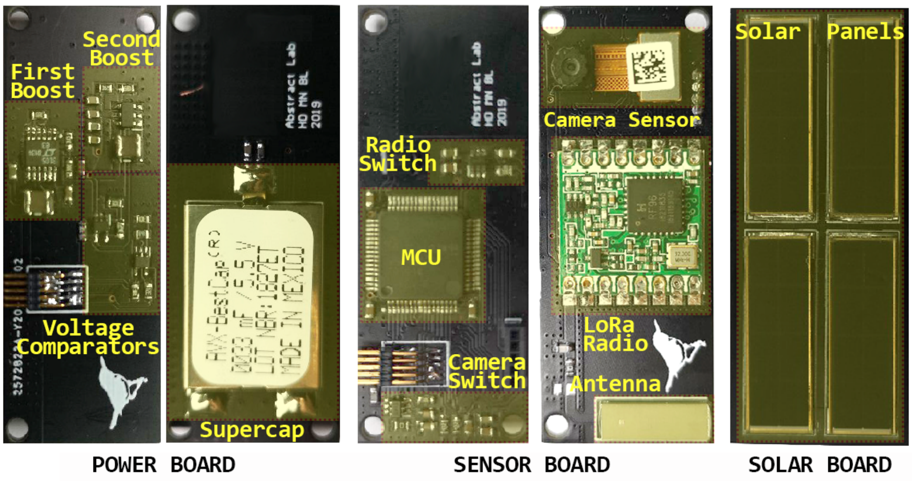


Figure 4.11: Camaroptera prototype PCBs

capacitor using an LTC3105[10]. It is a high efficiency step-up DC/DC converter that can operate from an input voltage as low as 225mV and integrates a maximum power point controller (MPPC). This enable operation directly from low voltage, like the case of solar panels in low light condition. The second boosting stage uses the energy stored inside the supercapacitor to provide a stable 3 V line to the sensor board. In this case a TPS61070[11] Synchronous Boost Converter is used. It provides an efficiency over 85% starting from an input voltage as low as 1.2 V for and output of 3 V. Moreover, thanks to the enable input we can keep the booster shutdown when the supercapacitor voltage is outside the maximum efficiency region.

Voltage thresholding. Two MIC841[12] precision voltage comparators with external adjustable hysteresis are respectively used to drive the enable input of the second booster and the reset line of the MCU. The first one ensures to operate the second boosting stage only in its maximum efficiency region between 1.24 V and 3V. This means that the second booster switch on only when the supercapacitor reaches 3 V and then it works till the lower threshold equal to 1.24V. The second comparator is used to keep the MCU in reset while the VCC line, the output of

[10]https://www.analog.com/en/products/ltc3105.html
[11]http://www.ti.com/product/TPS61070
[12]https://www.microchip.com/wwwproducts/en/MIC841

the second boosting stage, stabilizes. In this case the lower threshold is set to 2.2 V, the minimum voltage to operate the LoRa modem, while the upper to 3V.

Supercapacitor. One of the key parts of a power supply based on energy harvesting is the energy storage, in our case a high-density supercapacitor, as it must be tailored for the specific use case. We want to store enough energy to ensure that the longest atomic task completes without exhausting the device's stored energy. On the other side, we want to avoid using arbitrarily big supercapacitor that would take too long for charge up.

The second parameter that must be evaluated when using a supercapacitor is the Equivalent Series Resistance (ESR), to minimize the instantaneous voltage drop when a load is connected. This parameter is even more relevant for wireless application where the transmission requires a high current short pulse. After testing different solution, Camaroptera uses a BestCap Ultra-low ESR supercapacitor[13] with a capacitance of 33 mF. This is the largest capacitance available in the small-formfactor size that fits on Camaroptera's power board; a larger capacity would require a larger supercapacitor volume and footprint size, thus a bigger node.

Solar Board

The last board fit perpendicular on top of the other two, like an hat, and is entirely covered by solar panels. The solar panels are an array of four IXYS[14] high-efficiency monocrystalline panels, measuring 1cm x 2cm each. The solar board provides structure and power for the assembled device with mechanical and electrical connections to the other boards

4.4.3 Software implementation

As previously anticipated, powering a visual sensing system by relying only on the energy harvested from the environment poses tight constraints on the way the device can operate. The amount of energy that can be used is limited and not

[13] http://catalogs.avx.com/BestCap.pdf
[14] http://ixapps.ixys.com/DataSheet/IXYS%20Solar%20Product%20Brief.pdf

fixed during time. This is particularly true in the case of solar cells, that converts the light into energy. We can have periods of high brightness, associated with more energy, but also dark periods where the amount of energy is very limited.

However, visual sensing system must ensure a minimum quality of service in order to provide data useful for the end-to-end application that the system implements, making clear the importance of using the limited energy available in the most efficient way.

Processing Pipelines

A pipeline is a set of processing elements that describes the flow of data from the origin system to destination systems and defines how to transform the data along the way. The specific sub-computations that make up a pipeline are specific to the target application.

Camaroptera implements a multi-stage pipeline that process an image after its capture in order to identify interesting images, in our case are those containing peoples. The goal is to optimize the ratio between energy and transmitted data, in order to maximize the useful information acquired from the environment. Also, to avoid transmitting uninteresting images, which has a high time and energy cost contrarily to process the captured images locally, which has a low time and energy cost.

Camaroptera processing pipeline presents four stages:

- **Stage 1: Startup.** After the startup, the devices capture a photo and then checks if a previous configuration is already stored inside the FRAM. If past data is stored, the reconfiguration routine is executed, and the pipeline moves to stage 2. If not, we skip stage 2 and moves directly to stage 3. Reconfiguration routine will be discussed in Subsection 4.4.3
- **Stage 2: Difference Filtering.** This stage represents the first filtering phase do discard not interesting images. After capturing an image, Camaroptera compares it to the most recently captured image to determine if the new image differs. If the image differs, it may be of interest to the

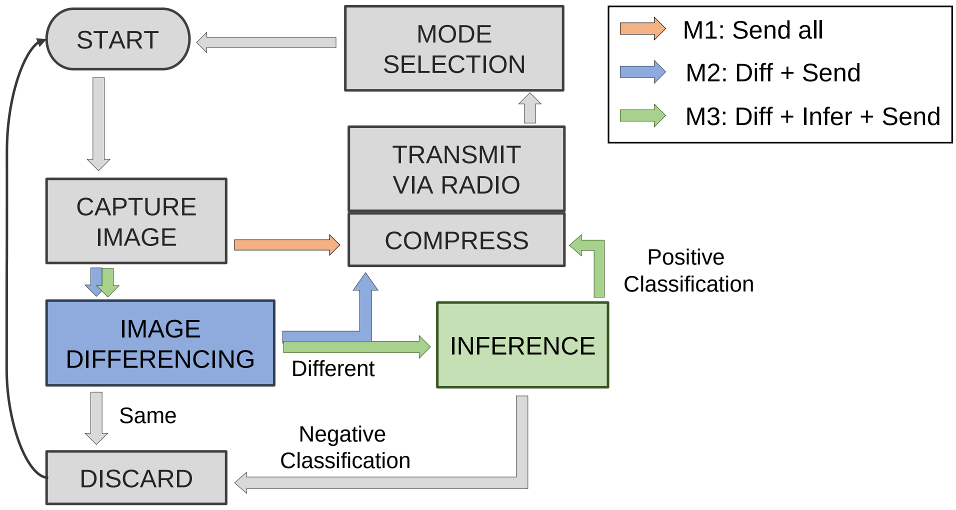


Figure 4.12: Camaroptera software flow chart

application and should continue through the pipeline. If not, Camaroptera can safely discard the image. After comparison to a previous frame, Camaroptera takes the new frame and saves it as a reference for future differences.

- **Stage 3: Inference.** Once we detect something new in the captured frame, Camaroptera tries to infer if it is something we are interested in or not. In our case, Camaroptera runs a deep neural network (described in section 4.4.3) trained to detect images containing people. If no people are detected, we go back to stage 1. In the event of people, the pipeline moves to stage 4.

- **Stage 4: Compression and Transmission.** In the last stage, the captured frame is compressed, packetized and transmitted in sequence using the LoRa radio. In this stage useful information regarding the time between the transmission of the different packets is collected. This data will be used to infer the luminosity level and exploited by the reconfiguration routine.

Adaptive reconfiguration

The performances of a battery-less long-range remote visual sensing system can be measured in many different ways. Two of them can be expressed in terms of energy efficiency in executing a particular task, or in terms of end-to-end latency form the moment we capture a picture to the moment we receive it remotely. To this end, it is important to consider that even data processing can be expensive in terms of computational power and may introduces latency that reduce the effective frame rate.

Camaroptera's software adapts to changing lighting conditions by varying the operating parameters of its radio and the operations that it performs in its pipeline. The goal of this reconfiguration is to maximize system availability, by minimizing the end-to-end latency required for capturing, processing and transmitting an image. The reconfiguration routine runs periodically, assesses the current lighting conditions, and sets the operating mode of the pipeline and radio. In our prototype, the reconfiguration routine runs after each successfully transmitted image.

The reconfiguration routine uses a measure of available energy to choose the appropriate operating mode. It does this by recording the supercapacitor charging rate after sending every individual radio packet. After all the packets for an image are sent, the individual charging rates are averaged and the average charging rate is used to select the best operating mode for processing the next image. The measured charging rate corresponds to the power available in the environment, which in turn corresponds to the light level.

The reconfiguration routine is pre-loaded with a table that indicates the mode with minimum end-to-end latency for a range of charging rate values. This table is based on a measured pre-characterization of Camaroptera's end-to-end latencies in different modes across varying light levels. At runtime, the reconfiguration routine chooses the operating mode that minimizes latency for its measured charging rate value, thereby making the system available for recording and processing maximum interesting events.

We consider three possible configurations of the processing pipeline. These are *send all*, *diff+send*, and *diff+infer+send*, which include different subsets of the pipeline stages described above. Here *diff* corresponds to Difference Filtering and *infer* corresponds to Inference. The chart presented in figure 4.12 shows the pipeline flow for the 3 different working configurations.

Image differencing

We implemented a simple image differencing algorithm that compare the captured frame, $F_{(t)}$, with the previous frame, $F_{(t-1)}$. We deem images different from one another if the number of different pixels exceeds a heuristically-defined threshold. We set the threshold empirically to 400 pixels by observing that human figures in our images tend to be around 20x20 pixels in size.

Camaroptera can supports more sophisticated methods, but it would introduce additional computation complexity, thus increasing the pipeline latency. This method instead is very light-weight, simple and fast (requires only a few subtractions, one addition and one comparison operation) and needs the storage of a single historical image, which works well for our space constrained device.

Inference algorithm

As second filtering stage, Camaroptera uses a DNN for image classification. The DNN's structure is derived from the LeNet [150] digit classification CNN. We trained a LeNet-structured DNN using a set of images collected using our prototype. We collected 4000 images around our university campus in 5 different locations in a wide variety of lighting conditions. We used Amazon Mechanical Turk[15] workers to label them as containing a person, not containing a person, or not being a valid image. The dataset included 60% negative images and 40% positive images. After labeling the dataset, we trained the network using a subset of 3600 of the images, holding 10% aside for testing and validation.

As trained, even LeNet, which is a small CNN, does not fit in the 250 kB of available memory on our MSP430 MCU.

To fit the small FRAM provided by the MSP430, network's structure has been altered and the hyperparameter optimized. To reduce the size of the input layer, we pass our 160x120 input image through a 4x4 average pooling layer before

[15] https://www.mturk.com/

passing it to LeNet's first convolutional layer, in order to shrinks the input down to 40x30 pixels.

Next, we used Genesis [29] network minimization tool to perform hyperparameter (i.e., structural) optimization on our trained, modified network. Genesis applies aggressive near-zero-pruning and layer separation techniques to reduce a network's weight storage requirements, the size of its intermediate activations, the expected inference latency, and the accuracy and precision of the network. Pruning leads to zeroing of weights that have little influence over the final outcome.

Taking the fully optimized network that Genesis produced, we encoded the weights in Compressed Sparse Row (CSR) representation for space efficiency. After applying these optimizations, the trained and optimized network weights consume 20kB of memory, which is a significant reduction compared to the initial 3.6MB of network weights, before optimization.

The final network occupies 20 kB space for the weights, 80kB for the intermediate activations and gives 78% accuracy on a test set, with 40% False Positives and 1% False Negatives. We chose a Genesis output with higher false positives because image differencing is likely to filter out many negative images, avoiding the risk of false positives. The low false negative rate ensures that Camaroptera does not miss valuable data.

Compression

Camaroptera implements an optimized version of baseline JPEG compression, derived from Moodstocks[16] JPEG encoder. We modified the implementation to use fixed point arithmetic instead of floating point since our MCU does not natively support floating point operations (software emulation is extremely slow). As result, the total execution time for compressing a 160 x 120 image is about 3.5 time faster, going from 25 seconds to 7 seconds.

Additionally, transmission of JPEG headers has been optimized. The first 500

[16] https://github.com/Moodstocks/jpec

(a) (b) (c)

Figure 4.13: Comparison between the original and compressed version of a frame captured by Camaroptera. (a) Original image. (b) Floating point JPEG. (c) Fixed point JPEG.

bytes of the compressed bit stream is always the same, using the same quality factor and frame resolution. We store the header on the receiver and avoid sending the 500 bytes of header data, which amounts to the transmission of two LoRa packets and seconds or minutes of device operation.

Fixed point JPEG degrades image quality, but not excessively. Figure 4.13 presents a comparison between the original image and the compressed version using floating- and fixed-point arithmetic.

Transmission

If the captured image is deemed interesting for the implemented application, Camaroptera packetizes the image and transmits it using the LoRa modem. We operate the LoRa radio in the following configuration: Frequency = 915MHz, Bandwidth = 500kHz, Spreading Factor = 7, Coding Rate = 4/5, Preamble Length = 8 bytes, Output TX Power = 17 dBm, Packet Size = 255 bytes.

4.4.4 Evaluation and Results

To evaluate our system, we performed several experiments regarding power consumption and execution times in both a controlled laboratory environment and outside real-word environment. We evaluated Camaroptera's power consumption and end-to-end latency across a range of operating conditions showing that Camaroptera processes frames with low latency and it is robust across a range of software configurations and environments.

The evaluation centers on a representative application that detects people outdoors. For this application, an interesting image is one containing at least one person.

Power Consumption breakdown

The whole energy harvesting system, that also integrates the hardware circuit for monitoring the two power lines, namely $V_{(cap)}$ and $V_{(cc)}$, consumes 197 μW when disconnected form the sensor board, with the supercapacitor fully charged. The sensor board, while capturing a QQVGA picture consumes 5.1 mW where 3 mW is consumed by the microcontroller and 2.1 mW by the camera sensor. This first task takes a time that can vary from a minimum of 661 ms to a maximum of 1.62 seconds, as in low light condition the camera needs more time for exposure calibration. Image differencing is computed in 46.14 ms, and the MCU requires 5.77 mW of power. During JPEG compression, that takes 7.23 seconds for a QQVGA, the MCU consumes 5.17 mW on average. The CNN inference algorithm takes 11.9 s to process a QQVGA images, with a power consumption equal to 4.85 mW. Lastly, sending a LoRa packets with a transmission power of 17 dBm consumes 363.83 mW. Transmission times varies based on the selected Spreading Factor and Bandwidth parameters. Our 33 mF supercapacitor can store enough energy for any of three different operating modes – (BW/SF) 500/7, 500/8, 250/7. Transmission times are, respectively, 117 ms, 193 ms and 217 ms.

Tasks latency characterization

In a controlled laboratory environment, we characterized the latency to collect energy for different sub-operations that make up Camaroptera's capture, processing, and transmission pipeline. Tests have been conducted with an illuminance ranging from 1.5 klx to 95 klx for simulating different light condition. Results are presented in Table 4.3 that shows direct measurements for capture, difference, inference, compression, and transmission of a single packet in three different radio configurations. Note that a compressed image requires more than a single radio packet in most cases – around 7 to 10 packets on average in our case. The latencies include time to collect energy for each operation.

Table 4.3: Recharge time for Camaroptera operations, expressed in seconds. '-' indicates that the power provided by the solar panels is enough to sustain the specific task. As can be noted, transmission is power limited.

Task	**Illuminance [klx]**										
	1.5	**5**	**15**	**25**	**35**	**45**	**55**	**65**	**75**	**85**	**95**
Capture	23.2	1.7	-	-	-	-	-	-	-	-	-
Differencing	1.6	-	-	-	-	-	-	-	-	-	-
Inference	216.2	25.3	-	-	-	-	-	-	-	-	-
Compression	245	14.97	6.5	-	-	-	-	-	-	-	-
TX - 500/7	307	17.3	6.6	4.24	3.4	2.6	1.9	1.7	1.3	1.1	0.7
TX - 500/8	803	44.6	16.7	9.4	6.3	4.5	4	3.1	2.5	2.2	1.8
TX 250/7	1096	64.7	18.5	10.3	7.5	5.3	4.1	3.5	2.9	2.3	2

The data shows that above 15 klx all operations, except radio transmission, can run continuously without depleting the capacitor as the power provided by the solar panels is sufficient to power the MCU and camera. The radio is power limited and always draws more power than the panels provide.

As can be noted, inference at low light level (e.g., 5 klx) has high latency, but still significantly less than the combined latency of compression and transmission, providing a key advantage. It is clear the degree to which the radio's latency and energy cost dominates. The benefit of difference detection is similarly clear, with its very low energy cost and outsize benefit in avoiding costly communication.

Adaptive pipeline analysis

As the final goal of the adaptive routine is to maximize the number of interesting images sent to a base station by Camaroptera, we have characterized the fraction of interesting images captured. The application defines what makes an image interesting. For our person detection application, an image is interesting if it contains a person.

To characterize Camaroptera's fraction of interesting images sent, we ran Camaroptera's three different operating modes separately, as well as its adaptive configuration, which selects the best mode for a given recharge rate. We compare Camaroptera to a continuously powered system variant (i.e., not pervasively deployable). We also compare Camaroptera to an "ideal" harvesting configuration that uses a bench power supply providing the datasheet-maximum power of our solar panel array. The ideal configuration represents an idealized harvested-energy deployment.

We ran Camaroptera's full capture, processing, and send pipeline, but to ensure experimental repeatability, we emulated the arrival of interesting images using a separate MSP430 board. The MSP430 generated Poisson ($\lambda = 10$ s) inter-arrival times and Gaussian ($\mu = 3$ s, $\sigma = 1$ s) durations for interesting image events.

Figure 4.14 shows the results. As can be noted, the mode that sends the largest fraction of interesting images varies with the light level of the environment. At low light, *diff+infer+send* sends the most interesting images because inference saves the time and energy that other modes spend transmitting packets. As light level increases, the benefit of processing instead of sending diminishes because the time cost of recharging is relatively much lower at high light levels. At a high light level, inference latency becomes the bottleneck, not communication and recharge latency. At such high light levels, *diff+send* is superior.

The line shows the behavior of Camaroptera's adaptive mode, which selects the best operating mode based on the light level. Comparing Camaroptera's actual solar-powered configurations to the ideal and continuously powered configurations shows that Camaroptera's behavior at high light levels nears ideal operation.

Both Camaroptera and the ideal harvesting configuration fall short of continuously powered operation because continuous power avoids recharge delays. However, continuously powered operation is inapplicable to pervasive deployment and this configuration makes clear the primary cost of recharging paid by an energy-harvesting remote image sensor.

End-to-End Per-Image Latency

Latency is an indirect metric of application quality because a high frame latency impedes capturing and processing future frames. Camaroptera should operate

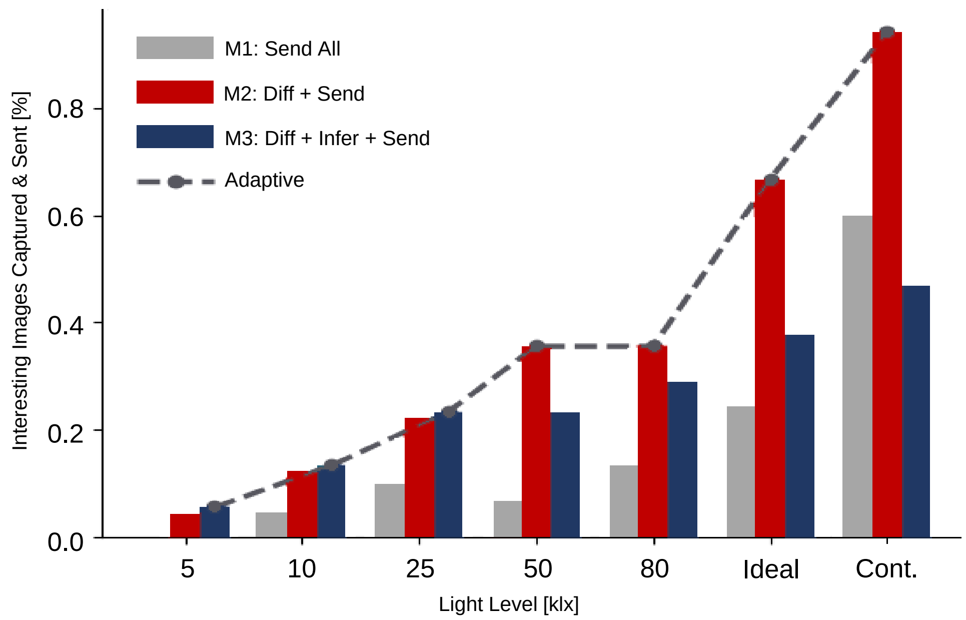


Figure 4.14: Comparison of the fraction of interesting images captured and sent in every mode at different light levels. The adaptive mode selects the best mode at every light level to maximize the fraction of interesting images captured and sent.

with minimum latency per image captured to return to capturing after processing as quickly as possible, and to make judicious use of energy. Camaroptera's highest latency operation is sending an image as a multi-packet message; Figure 4.15 shows the transmit latency including recharge time at different light levels.

Following the approach discussed in the previous subsection, to measure average end-to-end event latency, we ran the system's three operating modes across the same five light levels as above. We again emulated the arrival of interesting events, sampling a binomial distribution at capture time to determine whether a captured image is interesting or not.

We measured Camaroptera's average end-to-end per-image latency across its operating modes and for different light levels. Result of the average of 40 trials per configuration is presented in Figure 4.16. The data shows that across light levels, average latency is highest when the system sends all images. The high latency derives from the long latency cost of recharging after each packet sent. Note that the higher the average latency for a single frame, the less available the system is

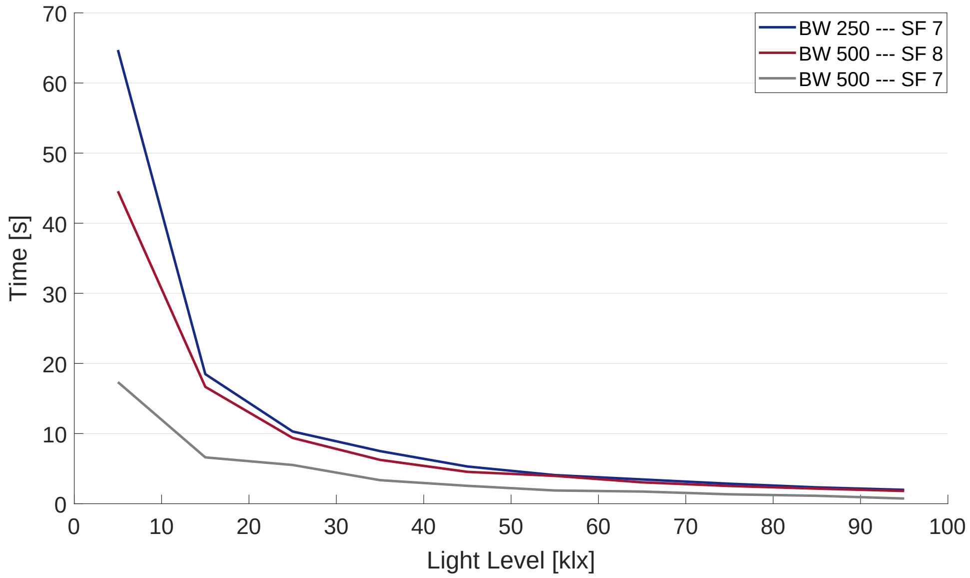


Figure 4.15: Time taken to recharge the 33mF capacitor for sending a 255 byte packet in the feasible operating modes of the radio across varied harvesting conditions. For clarity purposes, the graph show data from 5 klx to 95 klx

to capture and process new events. Consequently, the mode with lowest latency sends the most interesting events.

With very low light, Camaroptera benefits from running its relatively low-latency inference algorithm to avoid sending images with differences unnecessarily. The more energy, the less benefit from invoking inference. The lowest latency points in Figure 4.16 are the best points in Figure 4.14.

4.4.5 Summary

Battery-less image sensors present an opportunity for pervasive wide-spread remote sensor deployments that require little maintenance and have low cost. However, these devices present tight constraint in the quantity of energy that can be stored and used, as the only solution to power the devices is relying on energy harvesting systems. Developing smart devices that can autonomously adapt and reconfigure based on the environment condition where they are installed is the key solution to maximize the ratio between acquired useful data and the available energy.

In this section, Camaroptera, an IoT battery-less image sensor with the ability

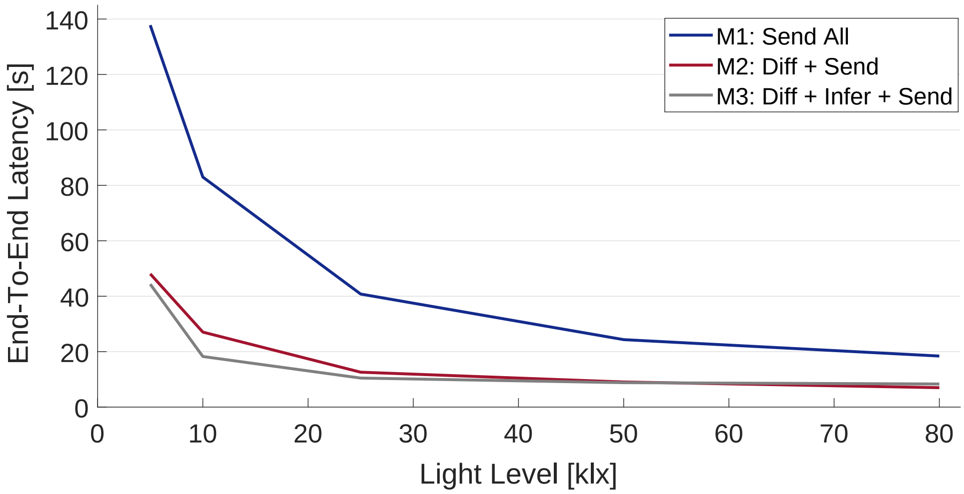


Figure 4.16: Average End-to-End Latency Comparison across different operating modes with 20% true positives and inference with 40% false positives and 1% false negatives.

to communicate over extremely long distances using an active LoRa radio, has been discussed.

Camaroptera avoids the high latency and energy cost of communication using near-sensor processing pipelines that process captured images to identify interesting images, discarding uninteresting images, and transmitting interesting images to a far-away base station.

The ability to dynamic reconfigures the processing pipelines allows Camaroptera to adapts its behavior depending on the energy availability in the environment where it is deployed, to minimize latency per captured image. Evaluation have highlighted how using machine inference on sensed data substantially reduces energy costs leading to captures and sends 2-5X more images of interest to an application.

4.5 Conclusion

Cameras have been used widely in multiple domains, ranging from security, entertainment, interactive environments and robotics. The emergence of ultra-low-power sensing devices along with low-power connectivity and cloud services has led to the rise of visual Internet of Things. Visual IoT poses significant challenges due to the need for sensing and processing of the acquired visual data. The richness of visual data provides many opportunities for analytics, while at the same time requiring high computational capabilities and therefore potentially high-bandwidth data transfer to a more powerful node. These challenges are then amplified when dealing with autonomous IoT devices, that rely on energy harvesting. Sensing node available energy, interaction latency, bandwidth costs and data privacy must be balanced between the IoT node and the remaining of the application.

In this section two low-power autonomous smart cameras with unique characteristics have been presented and discussed. The use of such systems provides a simple and objective analysis of the acquired information, producing accurate descriptive knowledge. Through these systems, it is possible to automate complex tasks, in a non-destructive way, providing adequate data for future analysis.

Section 4.3 presents a smart IoT camera for pest detection in precision agriculture. A computer vision solution combined with artificial intelligence algorithms that achieve important results in the early detection of orchards diseases, as required to effectively protect crops. It is based on a low-cost platform to be easily deployable over large areas. Taking advantage of a LoRa radio long range communication is ensured, lowering also the cost associated to the network architecture. Finally, it is completely autonomous to avoid maintenance costs.

Starting from the results and knowledge acquired during the development of the device presented in 4.3, the feasibility study to implement a computer vision solution based on an ultra-low-power MCU, like an MSP430, has been conducted. This has led to Camaroptera, presented in 4.4.

Results demonstrate the viability of a battery-less remote sensing platform in a small package that collects, processes and transmits over long distances only useful data. The whole prototype hardware platform is a compact 2cm x 3cm x 5cm device that integrates a complete energy-harvesting power supply for perpetual operation; a low-power camera sensor; a long-range radio modem and the MCU with non-volatile memory to ensure data retention.

Compared to a system that transmits all image data, Camaroptera's processing pipelines and adaptive processing scheme captures and sends 2-5X more images of interest to an application by avoiding recharging time associated with the transmission of useless data.

Chapter 5

Conclusions

The increasing popularity of wearables, ubiquitous computing, smart environment and industrial monitoring and controlling, are going to continue the exponential growth in connected IoT devices and generated data. Consumer products, durable goods, cars and trucks, industrial and utility components, sensors, and other everyday objects are being combined with Internet connectivity and powerful data analytic capabilities that promise to transform the way we live and work. As a result, using these devices, and extracting the value they promise, is going to require overcoming many challenges to meet the requirements of connectivity, scalability, and interoperability, as well as to efficiently support the transmission and processing of large volumes of data. Hence, there is a need for new enabling technologies, end-to-end architectures, and data management platforms.

As low-power IoT devices continue to significantly spread, two key challenges that are going to continue to grow in importance are how they are going to be powered and how connectivity can be provided.

Focusing on the powering problem, using AC mains power for powering the different devices might be effective for some application, but does not scale due to the limited location of outlets, physical size of AC-DC adapters, and retrofitting costs. The next solution is to rely on batteries, that can provide much more flexibility, but require periodic maintenance also creating unwanted environmental pollution due to chemicals. Moreover, as the number of devices grows, overhead maintenance cost will be untenable. These problems have led to energy-harvesting powered devices. With energy-harvesting, devices scavenge for their own energy after they are deployed. This can enable perpetual sensing where the device operates for as long as it is installed.

Energy-harvesting, however, also comes with many trade-offs and challenges. Scavengable energy is often unpredictable, intermittent and limited to only small quantities. This makes existing sensors and protocols difficult to use, as they typically assume a reliable source of energy.

The second key challenge that brings all Thing's together to form the Internet of Things is connectivity. The main requirements for IoT wireless communication protocols, are simplicity and low power, as the devices that implement these protocols should be cheap and be able to operate perpetually on battery and/or energy harvesting power. Various LPWAN technologies are currently contending to gain an edge over the competition to provide the massive connectivity that will be required by the world in which everyday objects are expected to be connected through wireless network in order to communicate with each other. A solution is represented by LoRa, a chirp spread-spectrum modulation, that promise long-range connectivity, in the range of km, with low-power requirements. LoRa also minimize the initial infrastructure cost associated to gateway deployments, as a single concentrator can cover wide areas and provide connectivity to a huge number of end-nodes.

Even if the energy required by a LoRa modem is minimal, the limited achievable bit rate imposes tight trade-offs on the amount of data that can be transmitted and received. It is important to make a judicious use of the available bandwidth and avoid unnecessary communications.

This even more important when relying on harvested energy, as the time to obtain enough energy for sending a data packet might degrade the quality of service provided by the IoT device.

To overcome these challenges, while enabling pervasive and perpetual sensing, this dissertation presents and discusses a series of designs for IoT devices based on energy-harvesting systems that enables meaningful sensing while operating only on the energy harvested from the environment.

A wide overview and general background about the IoT and its main system building blocks and challenges have been presented in **Chapter 1**. Here we also highlight the key role of ubiquitous computing. It is a fusion between fog and edge computing; a model in which data can be analyzed and processed by applications running in devices at different levels, within the network rather than in a centralized manner in the Cloud. In our case we have focused our research on providing innovative smart functionalities to IoT end-nodes, enabling the deployment of flexible devices that can adapt to the deployment scenario.

Chapter 2 tackles the connectivity problem, suggesting as a solution to provide connectivity to low-power devices the use of LoRa links. After a brief background regarding the technology and its unique peculiarity, we have presented the deployment of a local LoRa network for the municipality of Trento. Results highlight how LoRa is a feasible solution to provide connectivity for IoT devices. A single gateway can cover wide areas, lowering installation and maintenance cost. LoRa concentrators can then easily be connected to the legacy internet, making data coming from end-nodes easily available online. This chapter concludes with the promising results of LoRa links in industrial environments, showing how the technology's long-range capabilities can be traded off with a high noise immunity in short range communications.

Following the discussion of wireless connectivity, **Chapter 3** presents a suite of effective solutions to monitor the energy consumption. Here we have highlighted how the normal metering infrastructure is currently lacking some key functionalities to foster energy efficiency. The first design is targeted to the domestic sector. It is designed to be easily deployable, non-intrusive and low cost, to remove installation barriers. Results point out how it is now possible to run NILM algorithms directly inside smart meters, making the metering infrastructure more efficient and secure. The second design monitor energy consumption by observing how rapidly it is able to harvest energy and presents the complete study of a power supply based on energy harvesting using current sensors. Combining the NILM functionalities with the study of energy-harvesting using CT sensors has led to the last device presented in this chapter. The key functionality of this device is the ability to adapt to the load under monitoring, which, in turns, dictates the quantity of energy that can be harvested. A reconfiguration routine running in real-time along with the metering functionalities is able to adapt the level of processing based on the sensed energy consumption. This ensures perpetual functionalities without the need for battery nor maintenance, while still providing NILM functionalities over 700 W.

This sensor suite can then be used to instrument buildings to better understand how energy is being used, and, more importantly, where it is wasted.

Chapter 4 move to presents how it is now possible running machine learning algorithms on the edge and how this can increase the quality of service provided by the IoT application implemented. It can be argued that wireless communication is what takes more energy in low-power sensor nodes. This translates into a latency cost when relying on energy harvesting.

By leveraging on in-situ processing we can just transmit meaningful data, avoiding unnecessary communications. The first design presents an effective solution to automate the detection of the codling moths. The system is fully autonomous and can operate unattended. Detection reports can then be used for optimizing the use of chemicals only when the system detects threats for crops. Following this approach, Camaroptera, leverages a DNN to discard meaningless captured images. The key is a multi-stage processing pipeline that adapts to the sensed energy availability. As local processing functionalities can be both time and energy consuming, the reconfiguration routine ensure to operate Camaroptera in its best working mode, based on the incoming energy. Thanks to this approach, the smart camera is able to capture more images compared to a static processing pipeline, leading to a better quality of service for the implemented application.

Concluding, this dissertation demonstrates how modern microcontrollers provide enough computational power to implement sophisticated data processing on the edge and how a clever hardware-software co-design approach can lead to an overall optimization of the IoT architecture. Edge computational power is not wasted. Wireless data streaming is reduced to only meaningful data, lowering networking pressure and possible latencies. Energy harvesting solutions can be exploited to power IoT nodes, avoid batteries and AC-DC adapter.Finally, thanks to adaptive processing pipelines we can provide more useful data to the implemented application, all relying only on the energy harvested from the deployment scenario.

References

[1] M. Lacinák and J. Ristvej, 'Smart city, safety and security,' *Procedia Engineering*, vol. 192, pp. 522–527, 2017, 12th international scientific conference of young scientists on sustainable, modern and safe transport, ISSN: 1877-7058. DOI: https://doi.org/10.1016/j.proeng.2017.06.090. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S187770581732636X.

[2] P. Fraga-Lamas, T. M. Fernández-Caramés, M. Suárez-Albela, L. Castedo and M. González-López, 'A review on internet of things for defense and public safety,' *Sensors*, vol. 16, no. 10, 2016, ISSN: 1424-8220. DOI: 10.3390/s16101644. [Online]. Available: https://www.mdpi.com/1424-8220/16/10/1644.

[3] K. Biswas and V. Muthukkumarasamy, 'Securing smart cities using blockchain technology,' in *2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, Dec. 2016, pp. 1392–1393. DOI: 10.1109/HPCC-SmartCity-DSS.2016.0198.

[4] L. Khoukhi, 'Iot technologies for smart cities,' English, *IET Networks*, vol. 7, 1–13(12), 1 Jan. 2018, ISSN: 2047-4954. [Online]. Available: https://digital-library.theiet.org/content/journals/10.1049/iet-net.2017.0163.

[5] H. Arasteh, V. Hosseinnezhad, V. Loia, A. Tommasetti, O. Troisi, M. Shafie-khah and P. Siano, 'Iot-based smart cities: A survey,' in *2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC)*, Jun. 2016, pp. 1–6. DOI: 10.1109/EEEIC.2016.7555867.

[6] N. Neyestani, M. Y. Damavandi, M. Shafie-khah and J. P. S. Catalão, 'Modeling the pev traffic pattern in an urban environment with parking lots and charging stations,' in *2015 IEEE Eindhoven PowerTech*, Jun. 2015, pp. 1–6. DOI: 10.1109/PTC.2015.7232637.

[7] M. M. Rathore, A. Ahmad, A. Paul and S. Rho, 'Urban planning and building smart cities based on the internet of things using big data analytics,' *Computer Networks*, vol. 101, pp. 63–80, 2016, Industrial Technologies and Applications for the Internet of Things, ISSN: 1389-1286. DOI: https://doi.org/10.1016/j.comnet.2015.12.023. [Online].

Available: http://www.sciencedirect.com/science/article/pii/S1389128616000086.

[8] D. Kyriazis, T. Varvarigou, D. White, A. Rossi and J. Cooper, 'Sustainable smart city iot applications: Heat and electricity management eco-conscious cruise control for public transportation,' in *2013 IEEE 14th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, Jun. 2013, pp. 1–5. DOI: 10.1109/WoWMoM.2013.6583500.

[9] X. Shi, X. An, Q. Zhao, H. Liu, L. Xia, X. Sun and Y. Guo, 'State-of-the-art internet of things in protected agriculture,' *Sensors*, vol. 19, no. 8, 2019, ISSN: 1424-8220. DOI: 10.3390/s19081833. [Online]. Available: https://www.mdpi.com/1424-8220/19/8/1833.

[10] T. Popović, N. Latinović, A. Pešić, Ž. Zečević, B. Krstajić and S. Djukanović, 'Architecting an iot-enabled platform for precision agriculture and ecological monitoring: A case study,' *Computers and Electronics in Agriculture*, vol. 140, pp. 255–265, 2017, ISSN: 0168-1699. DOI: https://doi.org/10.1016/j.compag.2017.06.008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0168169916307566.

[11] A. Khattab, A. Abdelgawad and K. Yelmarthi, 'Design and implementation of a cloud-based iot scheme for precision agriculture,' in *2016 28th International Conference on Microelectronics (ICM)*, Dec. 2016, pp. 201–204. DOI: 10.1109/ICM.2016.7847850.

[12] Z. Zhao, V. Tsoutsouras, D. Soudris and A. Gerstlauer, 'Network/system co-simulation for design space exploration of iot applications,' in *2017 International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS)*, Jul. 2017, pp. 46–53. DOI: 10.1109/SAMOS.2017.8344610.

[13] B. Denby and B. Lucia, 'Orbital edge computing: Machine inference in space,' *IEEE Computer Architecture Letters*, vol. 18, no. 1, pp. 59–62, Jan. 2019, ISSN: 2473-2575. DOI: 10.1109/LCA.2019.2907539.

[14] I. I. Lysogor, L. S. Voskov and S. G. Efremov, 'Survey of data exchange formats for heterogeneous lpwan-satellite iot networks,' in *2018 Moscow Workshop on Electronic and Networking Technologies (MWENT)*, Mar. 2018, pp. 1–5. DOI: 10.1109/MWENT.2018.8337257.

[15] R. P. Dameri and C. Rosenthal-Sabroux, *Smart city: How to create public and economic value with high technology in urban space*. Springer, 2014.

[16] T. Nam and T. A. Pardo, 'Conceptualizing smart city with dimensions of technology, people, and institutions,' in *Proceedings of the 12th Annual International Digital Government Research Conference: Digital Government Innovation in Challenging Times*, ser. dg.o '11, College Park, Maryland, USA: Association for Computing Machinery, 2011, pp. 282–

291, ISBN: 9781450307628. DOI: 10.1145/2037556.2037602. [Online]. Available: https://doi.org/10.1145/2037556.2037602.

[17] R. Hall, B. Bowerman, J. Braverman, J. Taylor, H. Todosow and U. Von Wimmersperg, 'The vision of a smart city,' Sep. 2000.

[18] A. Zanella, N. Bui, A. Castellani, L. Vangelista and M. Zorzi, 'Internet of things for smart cities,' *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, Feb. 2014, ISSN: 2372-2541. DOI: 10.1109/JIOT.2014.2306328.

[19] B. Robert, 'Towards the trillion sensors market,' vol. 34, no. 2, pp. 137–142, Jan. 2014, ISSN: 0260-2288. DOI: 10.1108/SR-12-2013-755. [Online]. Available: https://doi.org/10.1108/SR-12-2013-755.

[20] M. Stolikj, 'Building blocks for the internet of things,' English, Ph.D. dissertation, Department of Mathematics and Computer Science, 2015, ISBN: 978-90-386-3928-4.

[21] R. Mahmoud, T. Yousuf, F. Aloul and I. Zualkernan, 'Internet of things (iot) security: Current status, challenges and prospective measures,' in *2015 10th International Conference for Internet Technology and Secured Transactions (ICITST)*, Dec. 2015, pp. 336–341. DOI: 10.1109/ICITST.2015.7412116.

[22] K. Maenaka, 'Future sensor systems in internet of things or trillion sensor universe,' English, *Sensors in Network*, vol. 28, no. 11, pp. 1247–1254, Nov. 2016. DOI: https://doi.org/10.18494/SAM.2016.1490.

[23] B. Campbell, M. Clark, S. DeBruin, B. Ghena, N. Jackson, Y.-S. Kuo and P. Dutta, 'Perpetual sensing for the built environment,' *IEEE Pervasive Computing*, vol. 15, no. 4, pp. 45–55, Oct. 2016, ISSN: 1536-1268. DOI: 10.1109/MPRV.2016.66. [Online]. Available: https://doi.org/10.1109/MPRV.2016.66.

[24] D. Minoli, K. Sohraby and B. Occhiogrosso, 'Iot considerations, requirements, and architectures for smart buildings—energy optimization and next-generation building management systems,' *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 269–283, Feb. 2017, ISSN: 2372-2541. DOI: 10.1109/JIOT.2017.2647881.

[25] H. C. Pöhls, V. Angelakis, S. Suppan, K. Fischer, G. Oikonomou, E. Z. Tragos, Rodrigo Diaz Rodriguez and T. Mouroutis, 'Rerum: Building a reliable iot upon privacy- and security- enabled smart objects,' in *2014 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, Apr. 2014, pp. 122–127. DOI: 10.1109/WCNCW.2014.6934872.

[26] T. Abera, N. Asokan, L. Davi, F. Koushanfar, A. Paverd, A.-R. Sadeghi and G. Tsudik, 'Invited - things, trouble, trust: On building trust in iot systems,' in *Proceedings of the 53rd Annual Design Automation Conference*, ser. DAC '16, Austin, Texas: Association for Computing Machinery,

2016, ISBN: 9781450342360. DOI: 10.1145/2897937.2905020. [Online]. Available: https://doi.org/10.1145/2897937.2905020.

[27] M. Abdel-Basset, G. Manogaran and M. Mohamed, 'Internet of things (iot) and its impact on supply chain: A framework for building smart, secure and efficient systems,' *Future Generation Computer Systems*, vol. 86, pp. 614–628, 2018, ISSN: 0167-739X. DOI: https://doi.org/10.1016/j.future.2018.04.051. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X1830400X.

[28] A. Dongare, C. Hesling, K. Bhatia, A. Balanuta, R. L. Pereira, B. Iannucci and A. Rowe, 'Openchirp: A low-power wide-area networking architecture,' in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, Mar. 2017, pp. 569–574. DOI: 10.1109/PERCOMW.2017.7917625.

[29] G. Gobieski, B. Lucia and N. Beckmann, 'Intelligence beyond the edge: Inference on intermittent embedded systems,' in *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, ser. ASPLOS '19, Providence, RI, USA: Association for Computing Machinery, 2019, pp. 199–213, ISBN: 9781450362405. DOI: 10.1145/3297858.3304011. [Online]. Available: https://doi.org/10.1145/3297858.3304011.

[30] A. Zoha, A. Gluhak, M. A. Imran and S. Rajasegarar, 'Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey,' *Sensors*, vol. 12, no. 12, pp. 16 838–16 866, 2012, ISSN: 1424-8220. DOI: 10.3390/s121216838. [Online]. Available: https://www.mdpi.com/1424-8220/12/12/16838.

[31] D. Chen, H. Cao, H. Chen, Z. Zhu, X. Qian, W. Xu and M. Huang, 'Smart insole-based indoor localization system for internet of things applications,' *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 7253–7265, Aug. 2019, ISSN: 2372-2541. DOI: 10.1109/JIOT.2019.2915791.

[32] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari and M. Ayyash, 'Internet of things: A survey on enabling technologies, protocols, and applications,' *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 2347–2376, Fourthquarter 2015, ISSN: 2373-745X. DOI: 10.1109/COMST.2015.2444095.

[33] M. Weyrich and C. Ebert, 'Reference architectures for the internet of things,' *IEEE Software*, vol. 33, no. 1, pp. 112–116, Jan. 2016, ISSN: 1937-4194. DOI: 10.1109/MS.2016.20.

[34] I. Mashal, O. Alsaryrah, T.-Y. Chung, C.-Z. Yang, W.-H. Kuo and D. P. Agrawal, 'Choices for interaction with things on internet and underlying issues,' *Ad Hoc Networks*, vol. 28, pp. 68–90, 2015, ISSN: 1570-8705. DOI: https://doi.org/10.1016/j.adhoc.2014.12.006. [Online].

Available: `http://www.sciencedirect.com/science/article/pii/S1570870514003138`.

[35] O. Said and M. Masud, 'Towards internet of things: Survey and future vision,' *International Journal of Computer Networks*, vol. 5, no. 1, pp. 1–17, 2013.

[36] Miao Wu, Ting-Jie Lu, Fei-Yang Ling, Jing Sun and Hui-Ying Du, 'Research on the architecture of internet of things,' in *2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE)*, vol. 5, Aug. 2010, pp. V5-484-V5–487. DOI: `10.1109/ICACTE.2010.5579493`.

[37] R. Khan, S. U. Khan, R. Zaheer and S. Khan, 'Future internet: The internet of things architecture, possible applications and key challenges,' in *2012 10th International Conference on Frontiers of Information Technology*, Dec. 2012, pp. 257–260. DOI: `10.1109/FIT.2012.53`.

[38] F. Bonomi, R. Milito, P. Natarajan and J. Zhu, 'Fog computing: A platform for internet of things and analytics,' in *Big data and internet of things: A roadmap for smart environments*, Springer, 2014, pp. 169–186.

[39] F. Bonomi, R. Milito, J. Zhu and S. Addepalli, 'Fog computing and its role in the internet of things,' in *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, ser. MCC '12, Helsinki, Finland: Association for Computing Machinery, 2012, pp. 13–16, ISBN: 9781450315197. DOI: `10.1145/2342509.2342513`. [Online]. Available: `https://doi.org/10.1145/2342509.2342513`.

[40] M. Aazam and E. Huh, 'Fog computing and smart gateway based communication for cloud of things,' in *2014 International Conference on Future Internet of Things and Cloud*, Aug. 2014, pp. 464–470. DOI: `10.1109/FiCloud.2014.83`.

[41] N. Bessis and C. Dobre, *Big data and internet of things: a roadmap for smart environments*. Springer, 2014, vol. 546.

[42] J. Ren, H. Guo, C. Xu and Y. Zhang, 'Serving at the edge: A scalable iot architecture based on transparent computing,' *IEEE Network*, vol. 31, no. 5, pp. 96–105, 2017, ISSN: 1558-156X. DOI: `10.1109/MNET.2017.1700030`.

[43] H. Li, K. Ota and M. Dong, 'Learning iot in edge: Deep learning for the internet of things with edge computing,' *IEEE Network*, vol. 32, no. 1, pp. 96–101, Jan. 2018, ISSN: 1558-156X. DOI: `10.1109/MNET.2018.1700202`.

[44] M. Satyanarayanan, 'The emergence of edge computing,' *Computer*, vol. 50, no. 1, pp. 30–39, Jan. 2017, ISSN: 1558-0814. DOI: `10.1109/MC.2017.9`.

[45] W. Shi and S. Dustdar, 'The promise of edge computing,' *Computer*, vol. 49, no. 5, pp. 78–81, May 2016, ISSN: 1558-0814. DOI: 10.1109/MC.2016.145.

[46] I. Analytics. (2018). 'Lpwan market report (2018-2023),' [Online]. Available: https://iot-analytics.com/lpwan-market-report-2018-2023-new-report/.

[47] C. Goursaud and J. M. Gorce, 'Dedicated networks for iot: Phy / mac state of the art and challenges,' *EAI Endorsed Transactions on Internet of Things*, vol. 1, no. 1, Oct. 2015. DOI: 10.4108/eai.26-10-2015.150597.

[48] A. Dongare, R. Narayanan, A. Gadre, A. Luong, A. Balanuta, S. Kumar, B. Iannucci and A. Rowe, 'Charm: Exploiting geographical diversity through coherent combining in low-power wide-area networks,' in *2018 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, Apr. 2018, pp. 60–71. DOI: 10.1109/IPSN.2018.00013.

[49] T. Voigt, M. Bor, U. Roedig and J. Alonso, *Mitigating inter-network interference in lora networks*, 2016. arXiv: 1611.00688 [cs.NI].

[50] L. Vangelista, A. Zanella and M. Zorzi, 'Long-range iot technologies: The dawn of lora™,' in *Future Access Enablers for Ubiquitous and Intelligent Infrastructures*, V. Atanasovski and A. Leon-Garcia, Eds., Cham: Springer International Publishing, 2015, pp. 51–58, ISBN: 978-3-319-27072-2.

[51] B. Reynders and S. Pollin, 'Chirp spread spectrum as a modulation technique for long range communication,' in *2016 Symposium on Communications and Vehicular Technologies (SCVT)*, Nov. 2016, pp. 1–5. DOI: 10.1109/SCVT.2016.7797659.

[52] A. Augustin, J. Yi, T. Clausen and W. M. Townsley, 'A study of lora: Long range and low power networks for the internet of things,' *Sensors*, vol. 16, no. 9, 2016, ISSN: 1424-8220. DOI: 10.3390/s16091466. [Online]. Available: https://www.mdpi.com/1424-8220/16/9/1466.

[53] M. Centenaro, L. Vangelista, A. Zanella and M. Zorzi, 'Long-range communications in unlicensed bands: The rising stars in the iot and smart city scenarios,' *IEEE Wireless Communications*, vol. 23, no. 5, pp. 60–67, Oct. 2016, ISSN: 1558-0687. DOI: 10.1109/MWC.2016.7721743.

[54] J. Petajajarvi, K. Mikhaylov, A. Roivainen, T. Hanninen and M. Pettissalo, 'On the coverage of lpwans: Range evaluation and channel attenuation model for lora technology,' in *2015 14th International Conference on ITS Telecommunications (ITST)*, Dec. 2015, pp. 55–59. DOI: 10.1109/ITST.2015.7377400.

[55] J. Petäjäjärvi, K. Mikhaylov, R. Yasmin, M. Hämäläinen and J. Iinatti, 'Evaluation of lora lpwan technology for indoor remote health and well-

being monitoring,' *International Journal of Wireless Information Networks*, vol. 24, no. 2, pp. 153–165, Jun. 2017, ISSN: 1572-8129. DOI: 10.1007/s10776-017-0341-8. [Online]. Available: https://doi.org/10.1007/s10776-017-0341-8.

[56] P. Neumann, J. Montavont and T. Noël, 'Indoor deployment of low-power wide area networks (lpwan): A lorawan case study,' in *2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Oct. 2016, pp. 1–8. DOI: 10.1109/WiMOB.2016.7763213.

[57] M. Rizzi, P. Ferrari, A. Flammini, E. Sisinni and M. Gidlund, 'Using lora for industrial wireless networks,' in *2017 IEEE 13th International Workshop on Factory Communication Systems (WFCS)*, May 2017, pp. 1–4. DOI: 10.1109/WFCS.2017.7991972.

[58] L. Angrisani, P. Arpaia, F. Bonavolontà, M. Conti and A. Liccardo, 'Lora protocol performance assessment in critical noise conditions,' in *2017 IEEE 3rd International Forum on Research and Technologies for Society and Industry (RTSI)*, Sep. 2017, pp. 1–5. DOI: 10.1109/RTSI.2017.8065952.

[59] D. Brunelli, E. Bedeschi, M. Ferrari, F. Tinti, A. Barbaresi and L. Benini, 'Long-range radio for underground sensors in geothermal energy systems,' in *Applications in Electronics Pervading Industry, Environment and Society*, A. De Gloria, Ed., Cham: Springer International Publishing, 2018, pp. 3–11, ISBN: 978-3-319-55071-8.

[60] J. Haxhibeqiri, A. Karaagac, F. Van den Abeele, W. Joseph, I. Moerman and J. Hoebeke, 'Lora indoor coverage and performance in an industrial environment: Case study,' in *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Sep. 2017, pp. 1–8. DOI: 10.1109/ETFA.2017.8247601.

[61] S. Penkov, A. Taneva, V. Kalkov and S. Ahmed, 'Industrial network design using low-power wide-area network,' in *2017 4th International Conference on Systems and Informatics (ICSAI)*, Nov. 2017, pp. 40–44. DOI: 10.1109/ICSAI.2017.8248260.

[62] T. Polonelli, D. Brunelli, A. Bartolini and L. Benini, 'A lorawan wireless sensor network for data center temperature monitoring,' in *Applications in Electronics Pervading Industry, Environment and Society*, S. Saponara and A. De Gloria, Eds., Cham: Springer International Publishing, 2019, pp. 169–177, ISBN: 978-3-030-11973-7.

[63] R. S. Sinha, Y. Wei and S.-H. Hwang, 'A survey on lpwa technology: Lora and nb-iot,' *ICT Express*, vol. 3, no. 1, pp. 14–21, 2017, ISSN: 2405-9595. DOI: https://doi.org/10.1016/j.icte.2017.03.004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2405959517300061.

[64] S. Oh and J. Shin, 'An efficient small data transmission scheme in the 3gpp nb-iot system,' *IEEE Communications Letters*, vol. 21, no. 3, pp. 660–663, 2017.

[65] K. Mekki, E. Bajic, F. Chaxel and F. Meyer, 'A comparative study of lpwan technologies for large-scale iot deployment,' *ICT Express*, vol. 5, no. 1, pp. 1–7, 2019, ISSN: 2405-9595. DOI: https://doi.org/10.1016/j.icte.2017.12.005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2405959517302953.

[66] B. Campbell, J. Adkins and P. Dutta, 'Cinamin: A perpetual and nearly invisible ble beacon,' in *Proceedings of the 2016 International Conference on Embedded Wireless Systems and Networks*, ser. EWSN '16, Graz, Austria: Junction Publishing, 2016, pp. 331–332, ISBN: 9780994988607.

[67] F. Fraternali, B. Balaji, Y. Agarwal, L. Benini and R. Gupta, 'Pible: Battery-free mote for perpetual indoor ble applications,' in *Proceedings of the 5th Conference on Systems for Built Environments*, ser. BuildSys '18, Shenzen, China: Association for Computing Machinery, 2018, pp. 168–171, ISBN: 9781450359511. DOI: 10.1145/3276774.3282822. [Online]. Available: https://doi.org/10.1145/3276774.3282822.

[68] I. E. Agency. (2019). 'World energy outlook,' [Online]. Available: https://www.iea.org/reports/world-energy-outlook-2019.

[69] G. W. Hart, 'Nonintrusive appliance load monitoring,' *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, Dec. 1992, ISSN: 1558-2256. DOI: 10.1109/5.192069.

[70] M. T. Tesfaye, M. Nardello and D. Brunelli, 'Residential electrical consumption disaggregation on a single low-cost meter,' in *2017 IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems (EESMS)*, Jul. 2017, pp. 1–6. DOI: 10.1109/EESMS.2017.8052678.

[71] Y. F. Wong, Y. Ahmet Şekercioğlu, T. Drummond and V. S. Wong, 'Recent approaches to non-intrusive load monitoring techniques in residential settings,' in *2013 IEEE Computational Intelligence Applications in Smart Grid (CIASG)*, Apr. 2013, pp. 73–79. DOI: 10.1109/CIASG.2013.6611501.

[72] M. Nardello, M. Rossi and D. Brunelli, 'A low-cost smart sensor for non intrusive load monitoring applications,' in *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)*, Jun. 2017, pp. 1362–1368. DOI: 10.1109/ISIE.2017.8001444.

[73] D. Brunelli and L. Tamburini, 'Residential load scheduling for energy cost minimization,' in *2014 IEEE International Energy Conference (ENERGYCON)*, May 2014, pp. 675–682. DOI: 10.1109/ENERGYCON.2014.6850499.

[74] K. Ehrhardt-Martinez, K. A. Donnelly, S. Laitner *et al.*, 'Advanced metering initiatives and residential feedback programs: A meta-review for household electricity-saving opportunities,' American Council for an Energy-Efficient Economy Washington, DC, 2010.

[75] S. Makonin, W. Sung, R. Dela Cruz, B. Yarrow, B. Gill, F. Popowich and I. V. Bajić, 'Inspiring energy conservation through open source metering hardware and embedded real-time load disaggregation,' in *2013 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)*, Dec. 2013, pp. 1–6. DOI: 10.1109/APPEEC.2013.6837120.

[76] D. Brunelli, I. Minakov, R. Passerone and M. Rossi, 'Smart monitoring for sustainable and energy-efficient buildings: A case study,' in *2015 IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems (EESMS) Proceedings*, Jul. 2015, pp. 186–191. DOI: 10.1109/EESMS.2015.7175875.

[77] X. Fang, 'Demand response under real-time pricing for domestic households with renewable dgs and storage,' English, *IET Generation, Transmission and Distribution*, vol. 11, 1910–1918(8), 8 Jun. 2017, ISSN: 1751-8687. [Online]. Available: https://digital-library.theiet.org/content/journals/10.1049/iet-gtd.2016.1066.

[78] P. Jacquot, O. Beaude, S. Gaubert and N. Oudjane, 'Demand side management in the smart grid: An efficiency and fairness tradeoff,' in *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, Sep. 2017, pp. 1–6. DOI: 10.1109/ISGTEurope.2017.8260265.

[79] X. Yang, Y. Zhang, B. Zhao, F. Huang, Y. Chen and S. Ren, 'Optimal energy flow control strategy for a residential energy local network combined with demand-side management and real-time pricing,' *Energy and Buildings*, vol. 150, pp. 177–188, 2017, ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2017.05.076. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378778817301275.

[80] J. Zhao, 'Non-intrusive energy saving appliance recommender system for smart grid residential users,' English, *IET Generation, Transmission and Distribution*, vol. 11, 1786–1793(7), 7 May 2017, ISSN: 1751-8687. [Online]. Available: https://digital-library.theiet.org/content/journals/10.1049/iet-gtd.2016.1615.

[81] S. Karnouskos, 'Demand side management via prosumer interactions in a smart city energy marketplace,' in *2011 2nd IEEE PES International Conference and Exhibition on Innovative Smart Grid Technologies*, Dec. 2011, pp. 1–7. DOI: 10.1109/ISGTEurope.2011.6162818.

[82] M. Levorato, N. Ahmed and Y. A. Zhang, 'Consumer in-the-loop: Consumers as part of residential smart energy systems,' in *2014 IEEE Inter-*

national Conference on Smart Grid Communications (SmartGridComm), Nov. 2014, pp. 758–763. DOI: 10.1109/SmartGridComm.2014.7007739.

[83] Belkin. (). 'Wemo® insight smart plug,' [Online]. Available: http://www.belkin.com/us/p/P-F7C029/.

[84] Q. R. Xu, I. Paprotny, M. Seidel, R. M. White and P. K. Wright, 'Stick-on piezoelectromagnetic ac current monitoring of circuit breaker panels,' *IEEE Sensors Journal*, vol. 13, no. 3, pp. 1055–1064, Mar. 2013, ISSN: 2379-9153. DOI: 10.1109/JSEN.2012.2234738.

[85] S. DeBruin, B. Ghena, Y.-S. Kuo and P. Dutta, 'Powerblade: A low-profile, true-power, plug-through energy meter,' in *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '15, Seoul, South Korea: Association for Computing Machinery, 2015, pp. 17–29, ISBN: 9781450336314. DOI: 10.1145/2809695.2809716. [Online]. Available: https://doi.org/10.1145/2809695.2809716.

[86] TED. (). 'Ted: The energy detective.,' [Online]. Available: http://www.theenergydetective.com/.

[87] S. DeBruin, B. Campbell and P. Dutta, 'Monjolo: An energy-harvesting energy meter architecture,' in *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '13, Roma, Italy: Association for Computing Machinery, 2013, ISBN: 9781450320276. DOI: 10.1145/2517351.2517363. [Online]. Available: https://doi.org/10.1145/2517351.2517363.

[88] B. Campbell and P. Dutta, 'Gemini: A non-invasive, energy-harvesting true power meter,' in *2014 IEEE Real-Time Systems Symposium*, Dec. 2014, pp. 324–333. DOI: 10.1109/RTSS.2014.36.

[89] D. Porcarelli, D. Brunelli and L. Benini, 'Clamp-and-forget: A self-sustainable non-invasive wireless sensor node for smart metering applications,' *Microelectronics Journal*, vol. 45, no. 12, pp. 1671–1678, 2014, ISSN: 0026-2692. DOI: https://doi.org/10.1016/j.mejo.2014.05.019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0026269214001785.

[90] D. Balsamo, G. Gallo, D. Brunelli and L. Benini, 'Non-intrusive zigbee power meter for load monitoring in smart buildings,' in *2015 IEEE Sensors Applications Symposium (SAS)*, Apr. 2015, pp. 1–6. DOI: 10.1109/SAS.2015.7133611.

[91] Digimondo. (). 'Digimondo lora meter.,' [Online]. Available: https://lpwanmarket.com/shop/things/digimondo-lora-meter/.

[92] C. Businesss. (). 'Panoramic power pan-10 sensor.,' [Online]. Available: http://www.panpwr.com/technology.

[93] Nke-Watteco. (). 'Lorawan capteur intens'o.,' [Online]. Available: http://www.nke-watteco.com/product/intensity-sensor-lorawan/.

[94] Neur.io. (). 'Neurio home energy monitor,' [Online]. Available: https://www.neur.io/energy-monitor/.

[95] G. Dalpiaz, A. Longo, M. Nardello, R. Passerone and D. Brunello, 'A battery-free non-intrusive power meter for low-cost energy monitoring,' in *2018 IEEE Industrial Cyber-Physical Systems (ICPS)*, May 2018, pp. 653–658. DOI: 10.1109/ICPHYS.2018.8390784.

[96] P3international. (). 'Kill a watt ez.,' [Online]. Available: http://www.p3international.com/products/p4460.html.

[97] T. Weng, B. Balaji, S. Dutta, R. Gupta and Y. Agarwal, 'Managing plug-loads for demand response within buildings,' in *Proceedings of the Third ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, ser. BuildSys '11, Seattle, Washington: Association for Computing Machinery, 2011, pp. 13–18, ISBN: 9781450307499. DOI: 10.1145/2434020.2434024. [Online]. Available: https://doi.org/10.1145/2434020.2434024.

[98] Smarthome. (). 'Smarthome imeter solo.,' [Online]. Available: http://www.smarthome.com/2423A1/iMeterSolo-INSTEON-Power-Meter-Plug-In/p.aspx.

[99] N. Klingensmith, D. Willis and S. Banerjee, 'A distributed energy monitoring and analytics platform and its use cases,' in *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings*, ser. BuildSys'13, Roma, Italy: Association for Computing Machinery, 2013, pp. 1–8, ISBN: 9781450324311. DOI: 10.1145/2528282.2528283. [Online]. Available: https://doi.org/10.1145/2528282.2528283.

[100] A. Marchiori and Q. Han, 'Using circuit-level power measurements in household energy management systems,' in *Proceedings of the First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings*, ser. BuildSys '09, Berkeley, California: Association for Computing Machinery, 2009, pp. 7–12, ISBN: 9781605588247. DOI: 10.1145/1810279.1810282. [Online]. Available: https://doi.org/10.1145/1810279.1810282.

[101] eGauge. (). 'Egauge pro meter.,' [Online]. Available: http://www.egauge.net/eos/energy-meters/EG4130.

[102] S. N. Patel, S. Gupta and M. S. Reynolds, 'The design and evaluation of an end-user-deployable, whole house, contactless power consumption sensor,' in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '10, Atlanta, Georgia, USA: Association for Computing Machinery, 2010, pp. 2471–2480, ISBN: 9781605589299. DOI:

10.1145/1753326.1753700. [Online]. Available: https://doi.org/10.1145/1753326.1753700.

[103] R. Send, Q. R. Xu, I. Paprotny, R. M. White and P. K. Wright, 'Granular radio energy-sensing node (green): A 0.56 cm3 wireless stick-on node for non-intrusive energy monitoring,' in *SENSORS, 2013 IEEE*, Nov. 2013, pp. 1–4. DOI: 10.1109/ICSENS.2013.6688133.

[104] L. Pereira and N. Nunes, 'Performance evaluation in non-intrusive load monitoring: Datasets, metrics, and tools—a review,' *WIREs Data Mining and Knowledge Discovery*, vol. 8, no. 6, e1265, 2018. DOI: 10.1002/widm.1265. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/widm.1265. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/widm.1265.

[105] S. Welikala, C. Dinesh, M. P. B. Ekanayake, R. I. Godaliyadda and J. Ekanayake, 'A real-time non-intrusive load monitoring system,' in *2016 11th International Conference on Industrial and Information Systems (ICIIS)*, Dec. 2016, pp. 850–855. DOI: 10.1109/ICIINFS.2016.8263057.

[106] R. Bonfigli, S. Squartini, M. Fagiani and F. Piazza, 'Unsupervised algorithms for non-intrusive load monitoring: An up-to-date overview,' in *2015 IEEE 15th International Conference on Environment and Electrical Engineering (EEEIC)*, Jun. 2015, pp. 1175–1180. DOI: 10.1109/EEEIC.2015.7165334.

[107] A. S. M. A. Mahmud and P. Sant, 'Real-time price savings through price suggestions for the smart grid demand response model,' in *2017 5th International Istanbul Smart Grid and Cities Congress and Fair (ICSG)*, Apr. 2017, pp. 65–69. DOI: 10.1109/SGCF.2017.7947603.

[108] A. T. Eseye, M. Lehtonen, T. Tukia, S. Uimonen and R. J. Millar, 'Exploiting flexibility of renewable energy integrated buildings for optimal day-ahead and real-time power bidding considering batteries and evs as demand response resources,' in *2019 16th International Conference on the European Energy Market (EEM)*, Sep. 2019, pp. 1–6. DOI: 10.1109/EEM.2019.8916277.

[109] T. K. Nguyen, I. Azarkh, B. Nicolle, G. Jacquemod and E. Dekneuvel, 'Applying nialm technology to predictive maintenance for industrial machines,' in *2018 IEEE International Conference on Industrial Technology (ICIT)*, Feb. 2018, pp. 341–345. DOI: 10.1109/ICIT.2018.8352201.

[110] G. V. Merrett, 'Transient and power-neutral computing: A paradigm shift for embedded systems?' In *"Hilariously Low-Power Computing" Workshop (ASPLOS 2016)*, Apr. 2016. [Online]. Available: https://eprints.soton.ac.uk/390669/.

[111] F. Civerchia, S. Bocchino, C. Salvadori, E. Rossi, L. Maggiani and M. Petracca, 'Industrial internet of things monitoring solution for advanced

predictive maintenance applications,' *Journal of Industrial Information Integration*, vol. 7, pp. 4–12, 2017, Enterprise modelling and system integration for smart manufacturing, ISSN: 2452-414X. DOI: `https://doi.org/10.1016/j.jii.2017.02.003`. [Online]. Available: `http://www.sciencedirect.com/science/article/pii/S2452414X16300954`.

[112] Z. M. Fadlullah, F. Tang, B. Mao, N. Kato, O. Akashi, T. Inoue and K. Mizutani, 'State-of-the-art deep learning: Evolving machine intelligence toward tomorrow's intelligent network traffic control systems,' *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2432–2455, Fourthquarter 2017, ISSN: 2373-745X. DOI: `10.1109/COMST.2017.2707140`.

[113] N. Kato, Z. M. Fadlullah, B. Mao, F. Tang, O. Akashi, T. Inoue and K. Mizutani, 'The deep learning vision for heterogeneous network traffic control: Proposal, challenges, and future perspective,' *IEEE Wireless Communications*, vol. 24, no. 3, pp. 146–153, Jun. 2017, ISSN: 1558-0687. DOI: `10.1109/MWC.2016.1600317WC`.

[114] S. Verma, Y. Kawamoto, Z. M. Fadlullah, H. Nishiyama and N. Kato, 'A survey on network methodologies for real-time analytics of massive iot data and open research issues,' *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1457–1477, thirdquarter 2017, ISSN: 2373-745X. DOI: `10.1109/COMST.2017.2694469`.

[115] S. Pouyanfar, S. Sadiq, Y. Yan, H. Tian, Y. Tao, M. P. Reyes, M.-L. Shyu, S.-C. Chen and S. S. Iyengar, 'A survey on deep learning: Algorithms, techniques, and applications,' *ACM Comput. Surv.*, vol. 51, no. 5, Sep. 2018, ISSN: 0360-0300. DOI: `10.1145/3234150`. [Online]. Available: `https://doi.org/10.1145/3234150`.

[116] S. Bianco, R. Cadene, L. Celona and P. Napoletano, 'Benchmark analysis of representative deep neural network architectures,' *IEEE Access*, vol. 6, pp. 64 270–64 277, 2018, ISSN: 2169-3536. DOI: `10.1109/ACCESS.2018.2877890`.

[117] R. Andri, L. Cavigelli, D. Rossi and L. Benini, 'Yodann: An architecture for ultralow power binary-weight cnn acceleration,' *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 1, pp. 48–60, Jan. 2018, ISSN: 1937-4151. DOI: `10.1109/TCAD.2017.2682138`.

[118] J. Tang, D. Sun, S. Liu and J. Gaudiot, 'Enabling deep learning on iot devices,' *Computer*, vol. 50, no. 10, pp. 92–96, 2017, ISSN: 1558-0814. DOI: `10.1109/MC.2017.3641648`.

[119] V. S. Chua, J. Z. Esquivel, A. S. Paul, T. Techathamnukool, C. F. Fajardo, N. Jain, O. Tickoo and R. Iyer, 'Visual iot: Ultra-low-power processing architectures and implications,' *IEEE Micro*, vol. 37, no. 6, pp. 52–61, Nov. 2017, ISSN: 1937-4143. DOI: `10.1109/MM.2017.4241343`.

[120] R. Andri, L. Cavigelli, D. Rossi and L. Benini, 'Hyperdrive: A systolically scalable binary-weight cnn inference engine for mw iot end-nodes,' in *2018 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, Jul. 2018, pp. 509–515. DOI: 10.1109/ISVLSI.2018.00099.

[121] M. Motamedi, D. Fong and S. Ghiasi, 'Fast and energy-efficient CNN inference on iot devices,' *CoRR*, vol. abs/1611.07151, 2016. arXiv: 1611.07151. [Online]. Available: http://arxiv.org/abs/1611.07151.

[122] G. Gobieski, N. Beckmann and B. Lucia, 'Intermittent deep neural network inference,' in *SysML Conference'18*, Stanford, CA, Feb. 2018. [Online]. Available: https://arxiv.org/pdf/1810.07751.pdf.

[123] S. Venkataramani, K. Roy and A. Raghunathan, 'Efficient embedded learning for iot devices,' in *2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC)*, Jan. 2016, pp. 308–311. DOI: 10.1109/ASPDAC.2016.7428029.

[124] P. Patel, M. Intizar Ali and A. Sheth, 'On using the intelligent edge for iot analytics,' *IEEE Intelligent Systems*, vol. 32, no. 5, pp. 64–69, Sep. 2017, ISSN: 1941-1294. DOI: 10.1109/MIS.2017.3711653.

[125] J. Lin, W. Yu, X. Yang, Q. Yang, X. Fu and W. Zhao, 'A real-time en-route route guidance decision scheme for transportation-based cyberphysical systems,' *IEEE Transactions on Vehicular Technology*, vol. 66, no. 3, pp. 2551–2566, Mar. 2017, ISSN: 1939-9359. DOI: 10.1109/TVT.2016.2572123.

[126] A. Sikander, A. Sikander, A. Arif, H. Sikander and U. Anwar, 'A modified approach for real-time taxi sharing based on city scale,' in *2019 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, Jul. 2019, pp. 1–7. DOI: 10.1109/ICECCE47252.2019.8940639.

[127] N. Mohamed, J. Al-Jaroodi, I. Jawhar, S. Lazarova-Molnar and S. Mahmoud, 'Smartcityware: A service-oriented middleware for cloud and fog enabled smart city services,' *IEEE Access*, vol. 5, pp. 17 576–17 588, 2017, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2017.2731382.

[128] W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, J. Lin and X. Yang, 'A survey on the edge computing for the internet of things,' *IEEE Access*, vol. 6, pp. 6900–6919, 2018, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2017.2778504.

[129] L. Nachman, J. Huang, J. Shahabdeen, R. Adler and R. Kling, 'Imote2: Serious computation at the edge,' in *2008 International Wireless Communications and Mobile Computing Conference*, Aug. 2008, pp. 1118–1123. DOI: 10.1109/IWCMC.2008.194.

[130] M. Magno, M. Pritz, P. Mayer and L. Benini, 'Deepemote: Towards multi-layer neural networks in a low power wearable multi-sensors bracelet,' in *2017 7th IEEE International Workshop on Advances in Sensors and Interfaces (IWASI)*, Jun. 2017, pp. 32–37. DOI: 10.1109/IWASI.2017.7974208.

[131] P. Ravi, U. Syam and N. Kapre, 'Preventive detection of mosquito populations using embedded machine learning on low power iot platforms,' in *Proceedings of the 7th Annual Symposium on Computing for Development*, ser. ACM DEV '16, Nairobi, Kenya: Association for Computing Machinery, 2016, ISBN: 9781450346498. DOI: 10.1145/3001913.3001917. [Online]. Available: https://doi.org/10.1145/3001913.3001917.

[132] J. Lee, M. Stanley, A. Spanias and C. Tepedelenlioglu, 'Integrating machine learning in embedded sensor systems for internet-of-things applications,' in *2016 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, Dec. 2016, pp. 290–294. DOI: 10.1109/ISSPIT.2016.7886051.

[133] R. P. Adler, J. Huang, R. Kong, P. Muse, L. Nachman, R. C. Shah, C. Wan and M. Yarvis, 'Edge processing and enterprise integration: Closing the gap on deployable industrial sensor networks,' in *2007 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, Jun. 2007, pp. 620–630. DOI: 10.1109/SAHCN.2007.4292874.

[134] H. Chang, A. Hari, S. Mukherjee and T. V. Lakshman, 'Bringing the cloud to the edge,' in *2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Apr. 2014, pp. 346–351. DOI: 10.1109/INFCOMW.2014.6849256.

[135] F. Samie, V. Tsoutsouras, L. Bauer, S. Xydis, D. Soudris and J. Henkel, 'Computation offloading and resource allocation for low-power iot edge devices,' in *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, Dec. 2016, pp. 7–12. DOI: 10.1109/WF-IoT.2016.7845499.

[136] J. A. C. Soto, M. Jentsch, D. Preuveneers and E. Ilie-Zudor, 'Ceml: Mixing and moving complex event processing and machine learning to the edge of the network for iot applications,' in *Proceedings of the 6th International Conference on the Internet of Things*, ser. IoT'16, Stuttgart, Germany: Association for Computing Machinery, 2016, pp. 103–110, ISBN: 9781450348140. DOI: 10.1145/2991561.2991575. [Online]. Available: https://doi.org/10.1145/2991561.2991575.

[137] S. S. Patil and S. A. Thorat, 'Early detection of grapes diseases using machine learning and iot,' in *2016 Second International Conference on Cognitive Computing and Information Processing (CCIP)*, Aug. 2016, pp. 1–5. DOI: 10.1109/CCIP.2016.7802887.

[138] J. Behmann, A.-K. Mahlein, T. Rumpf, C. Römer and L. Plümer, 'A review of advanced machine learning methods for the detection of biotic stress in precision crop protection,' *Precision Agriculture*, vol. 16, no. 3, pp. 239–260, 2015, ISSN: 1573-1618. DOI: 10.1007/s11119-014-9372-7. [Online]. Available: https://doi.org/10.1007/s11119-014-9372-7.

[139] D. Brunelli, A. Albanese, D. d'Acunto and M. Nardello, 'Energy neutral machine learning based iot device for pest detection in precision agriculture,' *IEEE Internet of Things Magazine*, vol. 2, no. 4, pp. 10–13, Dec. 2019, ISSN: 2576-3199. DOI: 10.1109/IOTM.0001.1900037.

[140] D. Sartori and D. Brunelli, 'A smart sensor for precision agriculture powered by microbial fuel cells,' in *2016 IEEE Sensors Applications Symposium (SAS)*, Apr. 2016, pp. 1–6. DOI: 10.1109/SAS.2016.7479815.

[141] T. Liu, W. Chen, W. Wu, C. Sun, W. Guo and X. Zhu, 'Detection of aphids in wheat fields using a computer vision technique,' *Biosystems Engineering*, vol. 141, pp. 82–93, 2016, ISSN: 1537-5110. DOI: https://doi.org/10.1016/j.biosystemseng.2015.11.005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1537511015300866.

[142] Q. YAO, J. LV, Q.-j. LIU, G.-q. DIAO, B.-j. YANG, H.-m. CHEN and J. TANG, 'An insect imaging system to automate rice light-trap pest identification,' *Journal of Integrative Agriculture*, vol. 11, no. 6, pp. 978–985, 2012, ISSN: 2095-3119. DOI: https://doi.org/10.1016/S2095-3119(12)60089-6. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2095311912600896.

[143] W. Ding and G. Taylor, 'Automatic moth detection from trap images for pest management,' *Computers and Electronics in Agriculture*, vol. 123, pp. 17–28, 2016, ISSN: 0168-1699. DOI: https://doi.org/10.1016/j.compag.2016.02.003. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0168169916300266.

[144] C. Xie, R. Wang, J. Zhang, P. Chen, W. Dong, R. Li, T. Chen and H. Chen, 'Multi-level learning features for automatic classification of field crop pests,' *Computers and Electronics in Agriculture*, vol. 152, pp. 233–241, 2018, ISSN: 0168-1699. DOI: https://doi.org/10.1016/j.compag.2018.07.014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0168169916308833.

[145] S. Liu and W. Deng, 'Very deep convolutional neural network based image classification using small training sample size,' in *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, Nov. 2015, pp. 730–734. DOI: 10.1109/ACPR.2015.7486599.

[146] I. V. Tetko, D. J. Livingstone and A. I. Luik, 'Neural network studies. 1. comparison of overfitting and overtraining,' *Journal of Chemical Information and Computer Sciences*, vol. 35, no. 5, pp. 826–833, Sep. 1995,

ISSN: 0095-2338. DOI: 10.1021/ci00027a006. [Online]. Available: https://pubs.acs.org/doi/abs/10.1021/ci00027a006.

[147] D. Brunelli, D. Dondi, A. Bertacchini, L. Larcher, P. Pavan and L. Benini, 'Photovoltaic scavenging systems: Modeling and optimization,' *Microelectronics Journal*, vol. 40, no. 9, pp. 1337–1344, 2009, Quality in Electronic Design 2nd IEEE International Workshop on Advances in Sensors and Interfaces Thermal Investigations of ICs and Systems, ISSN: 0026-2692. DOI: https://doi.org/10.1016/j.mejo.2008.08.013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0026269208004631.

[148] M. Nardello, H. Desai, D. Brunelli and B. Lucia, 'Camaroptera: A batteryless long-range remote visual sensing system,' in *Proceedings of the 7th International Workshop on Energy Harvesting and Energy-Neutral Sensing Systems*, ser. ENSsys'19, New York, NY, USA: Association for Computing Machinery, 2019, pp. 8–14, ISBN: 9781450370103. DOI: 10.1145/3362053.3363491. [Online]. Available: https://doi.org/10.1145/3362053.3363491.

[149] A. Colin, E. Ruppel and B. Lucia, 'A reconfigurable energy storage architecture for energy-harvesting devices,' in *Proceedings of the Twenty-Third International Conference on Architectural Support for Programming Languages and Operating Systems*, ser. ASPLOS '18, Williamsburg, VA, USA: Association for Computing Machinery, 2018, pp. 767–781, ISBN: 9781450349116. DOI: 10.1145/3173162.3173210. [Online]. Available: https://doi.org/10.1145/3173162.3173210.

[150] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, 'Gradient-based learning applied to document recognition,' *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998, ISSN: 1558-2256. DOI: 10.1109/5.726791.