Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Electric Power Engineering

Trym Johannes Berg

Investigating a LoRaWAN Communication Infrastructure For Cloud-Based Home Energy Management

Master's thesis in Energy and Environmental Engineering
Supervisor: Ümit Cali
Co-supervisor: Murat Kuzlu

June 2021

NTNU
Norwegian University of
Science and Technology

Trym Johannes Berg

Investigating a LoRaWAN Communication Infrastructure For Cloud-Based Home Energy Management

Master’s thesis in Energy and Environmental Engineering
Supervisor: Ümit Cali
Co-supervisor: Murat Kuzlu
June 2021

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Electric Power Engineering

NTNU
Norwegian University of
Science and Technology

NORWEGIAN UNIVERSITY OF SCIENCE AND TECHNOLOGY

Abstract

Faculty of Information Technology and Electrical Engineering (IE)
Department of Electric Power Engineering

Master Thesis

by Trym Johannes Berg - 766077

Increased focus on sustainable energy consumption and production has seen the rise of Home Energy Management Systems (HEMS) in residential homes. A HEMS is capable of monitoring and controlling the energy consumption and production in a home, based on signals from either the homeowner or a utility operator. To facilitate the communication infrastructure in large clusters of HEMS, it is necessary to use a long-range, low-latency communication protocol. This thesis aims to determine if LoRaWAN is a suitable communication protocol for powering bi-directional communication in clusters of HEMS. The stated research question is addressed through the design, implementation, and validation of cloud-based LoRaWAN communication in a HEMS communication infrastructure. Extensive research on LoRa, LoRaWAN, and cloud computing was performed to design and implement a test bed solution. Coverage and latency capabilities were then investigated by performing both indoor and outdoor transmission experiments. Additional calculations regarding airtime and metering resolutions were also performed. Results from the theoretical and experimental analysis show that a LoRaWAN-powered HEMS can achieve a large coverage area while still maintaining fairly low latency. It was further observed that end-node configuration and gateway deployment played a key part in enabling low-latency communication. The findings in this thesis reveal that LoRaWAN is a suitable communication protocol for clusters of HEMS, however, it highly depends on the deployed solution. Further studies and experiments on latency and coverage in large-scale deployments are needed to better define the limitations of LoRaWAN.

NORGES TEKNISK-NATURVITENSKAPELIGE UNIVERSITET

Sammendrag

Fakultet for Informasjonsteknologi og Elektroteknikk (IE)
Institutt for Elkraftteknikk

Master Avhandling

av Trym Johannes Berg - 766077

Økt søkelys på bærekraftig energiforbruk og produksjon har fremmet veksten av Home Energy Management Systems (HEMS) i boliger. Et HEMS er i stand til å overvåke og kontrollere energiforbruket og produksjonen i et hjem, basert på signaler fra enten huseieren eller en elektrisk nettoperatør. Kommunikasjonsinfrastrukturen i store klynger av HEMS krever en kommunikasjonsprotokoll med lang rekkevidde og lav forsinkelse. Denne oppgaven undersøker om LoRaWAN er en passende kommunikasjonsprotokoll for å drive toveiskommunikasjon i klynger av HEMS. Det foreslåtte forskningsspørsmålet er besvart gjennom design, implementering og validering av en skybasert LoRaWAN-kommunikasjonsinfrastruktur. Omfattende forskning på LoRa, LoRaWAN og skytjenester ble utført for å kunne designe og gjennomføre en test-bed løsning. Dekning og forsinkelse ble deretter undersøkt ved å utføre både innendørs og utendørs overføringseksperimenter. Ytterligere beregninger på time on air og måleoppløsninger ble også utført. Resultatene fra den teoretiske og eksperimentelle analysen viser at et LoRaWAN-drevet HEMS kan oppnå stort dekningsområde samtidig som den fortsatt holder relativt lav ventetid. Det ble videre observert at konfigurasjon av endenoder og best mulig plassering av mottakspunkter var avgjørende for å muliggjøre kommunikasjon med lav forsinkelse. Funnene i denne oppgaven avslører at LoRaWAN er en passende kommunikasjonsprotokoll for klynger av HEMS, men det avhenger av den implementerte løsningen. Ytterligere studier om forsinkelse og dekning i store systemer er nødvendig for å bedre definere begrensningene til LoRaWAN.

Preface

The master thesis presented in this paper concludes the 5-year MCs program in Energy and Environmental Engineering at the Norwegian University of Science and Technology (NTNU), Department of Electric Power Engineering. The study was conducted during the fall of 2021, and it is loosely based on a specialization project written during the autumn of 2020. The main focus of that project was to build a home automation system that predicts the desired lighting output and changes it based on detected emotions from a camera. For the master thesis, it was decided to shift the focus towards communication in home automation systems.

Sincere gratitude goes to my supervisor Umit Cali for the knowledge and experience he has provided me with. His feedback have encouraged me to always try my best, even when the wind is not blowing in my direction. I would also like to thank Ugur Halden, a PhD researcher for NTNU, for helping me with programming and hardware related issues. Furthermore, thanks to my co-supervisor Dr Murat Kuzlu for providing me with relevant literature and feedback on the thesis. Thanks also to Anders Gytri at the Department of Electrical Power Engineering for ordering the hardware used in this thesis.

Finally I would like to give a huge thanks to my wife Johanne Dieserud and my family for supporting me and cheering me on during the last semester. Thanks for always making sure that i had enough food in the fridge.

Contents

List of Figures

List of Tables

Abbreviations

SG	Smart Grid
HAS	Home Automation Systems
QoL	Quality of Life
MC	Microcontroller
HCHAS	Human-Centric Home Automation Systems
AMI	Advanced Metering Infrastructure
LPWAN	Low-Power Wide-Area Network
IoT	Internet of Things
NB-IoT	Narrowband Internet of Things
CSS	Chirp Spread Spectrum
DSSS	Direct Sequence Spread Spectrum
SF	Spreading Factor
SNR	Signal-to-Noise Ratio
DR	Data Rate (LoRaWAN data rate index A.2)
LAN	Local Area Network
LoRaWAN	Long-Range Wide-Area Network
ADR	Adaptive Data Rate
RTO	Regional Transmission Organization
ISO	Independent System Operator
CPP	Critical Peak Pricing
VPP	Variable Peak Pricing
RPi	Raspberry Pi
SSH	Secure Shell
AWS	Amazon Web Services
TTCS	The Things Community Stack

TTS	The Things Stack
TTN	The Things Network
CCS	Cloud Computing Service
Tx power	Transmission Power
LTE	Long Term Evolution
MQTT	Message Queuing Telemetry Transport
HTTPS	Hypertext Transfer Protocol Secure
DBMS	Database Management System
DSM	Demand Side Management
ADSM	Active Demand Side Management
ToA	Time on Air
ToU	Time-of-Use

Chapter 1

Introduction

Presented below is the motivation and declaration of the chosen research objective for this thesis. Section 1.1 presents a context motivation for the study. Section 1.2 details the main research problem, research objectives, and the research topics derived from the objectives. Section 1.3 poses the thesis structure with an overview of each chapter

1.1 Motivation

The rapid increase in global energy demand for the last 70 years has mainly been covered by non-renewable energy resources. These resources will, at the current rate, be depleted in the next century (Smil, 2017). Renewable energy sources supply only a small percentage of global consumption, but that is inevitably going to change. To increase the global renewable energy consumption percentages, we can deploy two methods: Simply put, one can either reduce consumption or increase production. This is fairly simplified; however, it still represents two areas with potential for improvements.

Current electricity grids have not yet fully accommodated solutions for the increased renewable energy production causing irregularities on the grid. The residential sector is seeing more and more consumers installing on-premise renewable energy production (Askeland et al., 2018). Energy producing homeowners are called prosumers, who can sell, use or store the energy they produce. Smart Grid (SG) poses as a solution for modernizing the electricity grid with advanced control and monitoring capabilities, facilitated by a state-of-the-art communication network. Large amounts of prosumers clustered together in a network can potentially become a self-sufficient power plant while maintaining satisfactory grid conditions

To achieve self-sufficient status, households also have to reduce their consumption. HEMS is a solution that allows prosumers (and consumers) to monitor, control and shape their load profile in response to signals from a utility company (Mahapatra & Nayyar, 2019) (Pipattanasomporn, Kuzlu, & Rahman, 2012) (Dinh, Yun, Kim, Lee, & Kim, 2020). The utility uses the metering information from HEMS to stabilize the grid conditions based on current consumption/production. Their goal is to suggest optimal load profiles for each home (or even appliance) and perform Demand Side Management (DSM) to achieve this. Those actions can be a manual effort from household residents or automatic control from the utility side. A HEMS is highly dependent on a reliable communication system that facilitates the traffic of in and out-going information.

Clustering a large number of prosumers together requires a large-scale communication network with long-range capacities. It is important to select the proper communication protocol(s) for this network. LoRaWAN is an emerging protocol based on the LoRa technology designed by Semtech (*LoRaWAN 1.1 Specification*, 2017) ("SX1276/77/78/79", 2020). It is a low-power long-range communication protocol suitable for battery-powered devices, often situated in hard-to-reach places. The protocol can be used by end-nodes to transmit small-sized sensor readings or simple commands to actuators on devices (Bingöl, Kuzlu, & Pipattanasompom, 2019). One of the main benefits of LoRaWAN is the simple installation process, on both ends of the network. It operates on a license-free frequency band, which allows the community to contribute with coverage. A LoRaWAN solution may be applied in HEMS; however, it depends on the system's functionality and requirements.

The communication infrastructure is dependent on a low-latency operating center with software that stores and analyzes the incoming metering data. These applications are conducted by utility, and their operating center can either be on-premise or in the cloud. Cloud-based services are computing services hosted by decentralized data centers (Peter Mell, 2011). Instead of having to pay large investment costs for computing and network hardware, these can instead be bought over the internet from cloud service providers (Soliman, Abiodun, Hamouda, Zhou, & Lung, 2013). Cloud services are easily scalable, which makes them a good fit for a growing LoRaWAN infrastructure.

1.2 Research Objectives & Topics

The main objective of this thesis is to determine if LoRaWAN is capable of powering a bi-directional communication infrastructure for prosumer clusters. To answer this question, the thesis proposes the following research objectives: Design, implementation, and validation of a cloud-based LoRaWAN communication infrastructure, capable of facilitating both uplinks to and downlinks from HEMSs. The infrastructure is designed and implemented based on a literature review of relevant topics. Coverage, latency, and metering resolution assessments are performed to validate the proposed solution. The literature research topics are derived from the main objective and cover communication technology and cloud-based operating centers in HEMS. First presented is a look at HEMS infrastructure and requirements. Following is a small survey of communication protocols to show how LoRa bridge the range/power gap. Furthermore, a detailed investigation on LoRa and the LoRaWAN protocol is conducted, as with Cloud Computing Services (CCS). The final section introduces the concept of DSM, which details the possible applications for a HEMS. Figure 1.1 visualizes the research topics.

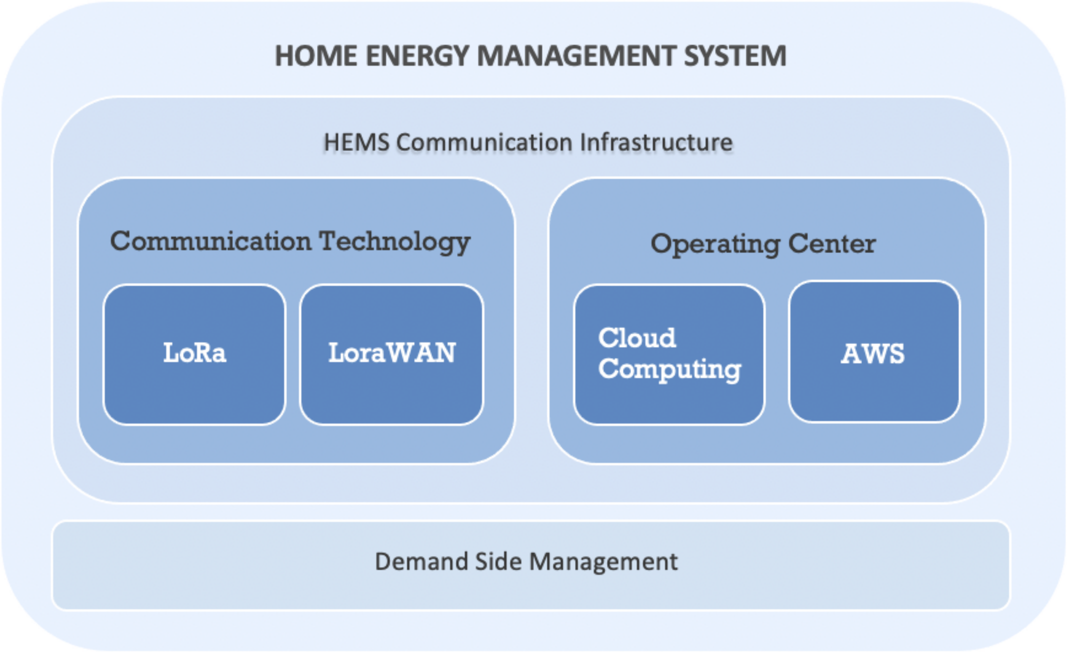


FIGURE 1.1: Structure of literature research

1.3 Thesis Structure

This thesis follows a generic chapter composition of literature review and background, methodology, results, discussion, conclusion, and further works. The literature/background chapter investigates the research topics stated above. It presents necessary knowledge about CCS and LoRaWAN, which is needed for realizing the communication infrastructure.

After the background chapter, the methodology is covered. This chapter focus on the design and implementation of the various components involved in the communication infrastructure. It details the hardware implementation, installation, programming, and operations required for realizing the cloud-based communication system. The methodology chapter also includes an experimental design section that presents testing procedures and parameters used to evaluate the test-bed solution. The experiments are complemented with a theoretical investigation on metering resolution based on the first principles of LoRa and LoRaWAN.

Results from the coverage and resolution calculations are presented with relevant explanations. Finally, the discussion section covers the analysis and evaluation of the results and the proposed solution. It is used to identify if LoRaWAN is a suitable communication protocol for a HEMS system. Improvements and issues with the proposed system are also presented. The conclusion then ties together the results and discussions to summarize the findings. Further work is presented last, and it contains research pointers to follow-up studies. Figure 1.2 visualize the structure of the thesis.

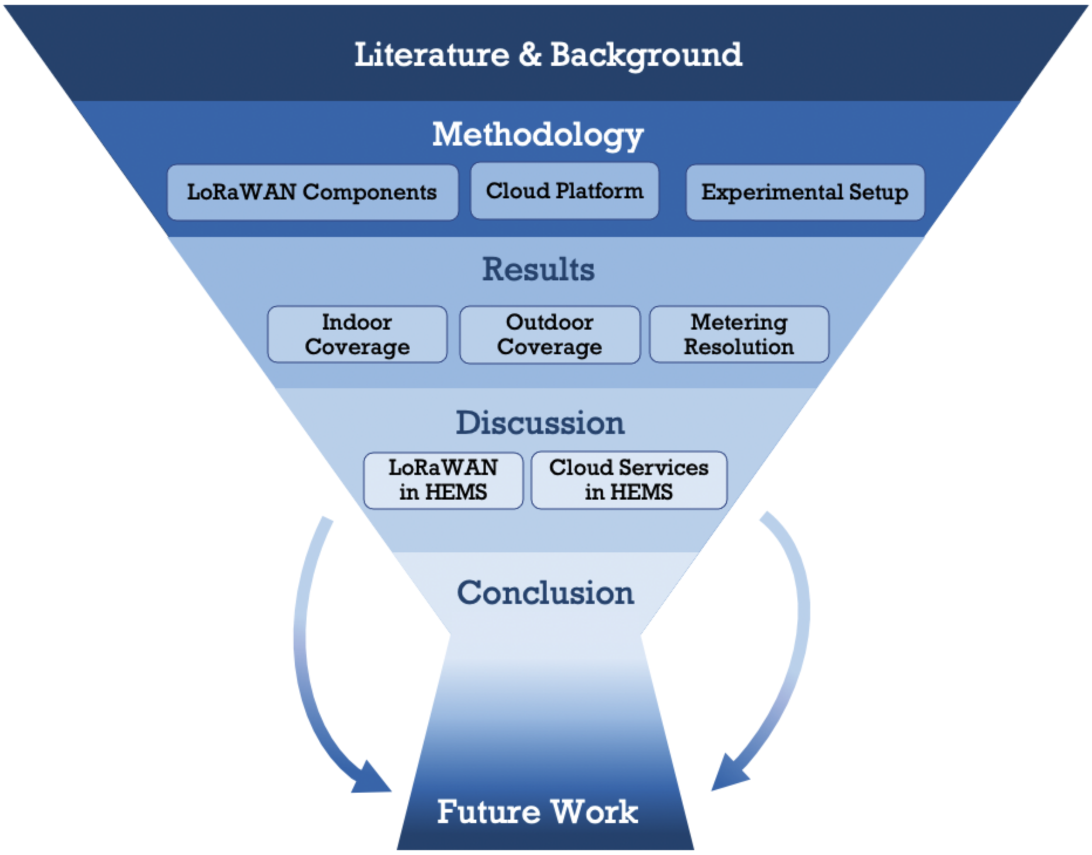


FIGURE 1.2: Thesis structure visualized

Chapter 2

Theory & Literature Review

In 1966, a computer systems engineer named Jim Sutherland designed what is known to be the first-ever home automation system (HAS) ("A Computer in the Basement?", 1968). It was built to manage digital clocks and calendars, track food items, and other basic household routines. Jim and his wife wanted to make everyday life easier with a computer programmed to automate simple tasks. Their design philosophy stands as a forever-lasting foundation for future home automation systems; They are built for serving human needs.

Home Automation Systems automate household tasks to reduce human efforts and increase Quality of Life (QoL) (Berg, 2020). Since QoL is a subjective parameter, it is not possible to make a one-size-fits-all system. What tasks to automate depends on a large number of factors: Desires, infrastructure, weather, price, interior design, legislation's, accessibility and many more. Since HAS designs are user-specific, it is hard to find a general consensus on a typical HAS. They range from automated calendar planer/optimizer (Tay, Botzheim, & Kubota, 2018) to emotion-based lighting automation (Berg, 2020). Although the functionalities of modern HAS vary, they mostly follow the same structure.

Modern HAS consists of a central gateway, "hub", and nodes connected to it (appliances, sensors, meters etc.). The participating devices (nodes) can connect to the gateway through a wired connection. However, they are usually wireless. Home gateways do not only facilitate the communication between devices but can also operate as a controller for the connected nodes. Much like how human brains receive, perceive, process, and act upon signals coming from different parts of the body, the home gateway acts in a similar manner. HAS replaced eyes, limbs, and brains with sensors, actuators, and processing units. These processing units can be computers or Microcontrollers (MC) installed on-premise; however, a cloud-based alternative for processing is also possible. With cloud-enabled systems, users access almost unlimited storing and processing power via a data center operated by the cloud provider. This is an alternative solution with reduced investment costs and decentralized operation.

The actions taken by HAS's to increase QoL for residents are mainly divided into two groups; Grid-centric actions specifically designed for grid and energy efficiency, and Human-Centric actions implemented to serve human needs and desires. HAS can be both human- and grid-centric; however, their operational principles are mostly focused on one of them. This paper will mainly focus on grid-centric systems; however, human-centric systems are briefly covered in the paragraph below. More information on this topic can be found in (Berg, 2020).

Human-Centric Home Automation Systems (HCHAS) manage Home Things based on QoL indicators drawn from each individual user (Berg, 2020). QoL indicators describe certain conditions a user values in their home. An example of this can be dim lighting at dawn or increased ventilation right after coming home from work. HCHAS tries to understand and learn user patterns and behaviors in order to adapt to their lifestyle. Such systems attempt to solve personal constraints by automating and scheduling appliance use. These personal constraints are mathematically formulated based on the QoL indicators.

In some cases, conflicting constraints can occur. This is the case when f.ex. one user prefers dim lights right before bed. However, the user is performing an activity coupled with a higher preferred lighting level. One solution to solving the conflicting constraints is by formulating a constraint satisfaction problem (CSP). However, solving a CSP might result in no solution due to conflicting constraints. (Tay et al., 2018) propose a weighted CSP to combat conflicting constraints by partially solving personal constraints. The benefit of such a solution is that conflicting constraints can be resolved to some degree, instead of having no solution at all. CSP can become quite complex and time-consuming, so solving them in a cloud environment could potentially reduce the computing time.

2.1 Home Energy Management Systems

In developed countries, most people have access to energy sources like heating and electricity. However, many end-users have little to no knowledge about peak energy demand in the electricity grid (Iglesias Vázquez, Kastner, Gaceo, & Reinisch, 2011). Users lack the knowledge and motivation for counteracting grid imbalances with energy conservation. It is, therefore, crucial to implement systems that encourage and help homeowners to manage their energy consumption. The solution is a Home Energy Management System (HEMS), which is built for control and monitoring of energy consumption and production in a home.

Energy management systems control and supervise energy consumption in residential, industrial, or commercial buildings (Khamphanchai et al., 2014). Residential buildings with or without energy production have the potential of implementing a HEMS (Elkazaz, Sumner, Davies, Pholboon, & Thomas, 2019). The main objective in HEMS is to manage a building consumption profile in order to satisfy grid operators and increase energy efficiency. If the home is equipped with solar panels or other forms of on-site energy production, then HEMS will delegate how this energy is used (or not used). Surplus electricity produced by homes can be exported back into the distribution grid, or stored in an energy storage solution installed on-premise. Optimal management of renewable on-site production is achieved by either storing excess electricity (Home Energy Storage System), or moving loads in time (Demand Side Management).

2.1.1 Operating Functionalities

(Mahapatra & Nayyar, 2019) suggests five main facilitating components of HEMS; Monitoring, logging, control, management, and alarm. To make it simpler, the author of this thesis concatenates them into the three M's; Measuring, Messaging, and Managing. These functionalities are vital for employing various operating strategies in HEMS. Each functionality is facilitated by one or more of the infrastructure components mentioned in section 2.2.2.

Measuring	**Sensing and monitoring of the real world**. This functionality is mostly carried out by the smart meters, or (if installed) other sensing devices like CO2- and temperature sensors. The system needs to be able to measure and store energy information like 24-hour time series of consumption and generation, grid conditions, electricity price, etc. This functionality is vital for any automation system, as data is the main source on which these systems act. Modern automation systems utilize and analyze historic data, which makes it important to have an available storage solution. The measuring part also includes gathering open-source data like weather information, which can be accessed over the internet.
Messaging	**Facilitate bidirectional communication between home Things and utility**. HEMS should have a robust communication system and provide users with alarms and notifications about faulty equipment, errors, and scheduled tasks. Users should be able to receive and send messages to home Things from anywhere in the world. Designing the communication infrastructure can be a tricky process, as there are many implementation possibilities (Mataloto et al., 2019). Latency, messaging rate, and reliability are some of the important aspects to consider. The messaging functionality is critical for realizing both management and measuring functionality.

Managing	**Control and management of Things in the home.** Users and utility can actively administer the home by sending commands to Things. The messages (commands) are dependent on device competence; some Things only respond to binary messages, while others allow for more complex control. The true power of HEMS comes from combining measuring, messaging, and control. HEMS uses applications to automatically operate and manage Things based on selected objectives.

TABLE 2.1: The three functionalities of Home Energy Management Systems

The three M's should always be viewed in context together, as their cooperation lays the foundation for energy management strategies in HEMS.

2.1.2 Energy Management Infrastructure

The general infrastructure of HEMS is parallel to the Advanced Metering Infrastructure (AMI), which embodies energy meters, operating center, home Things, and communication networks (Mahapatra & Nayyar, 2019) 2.1.

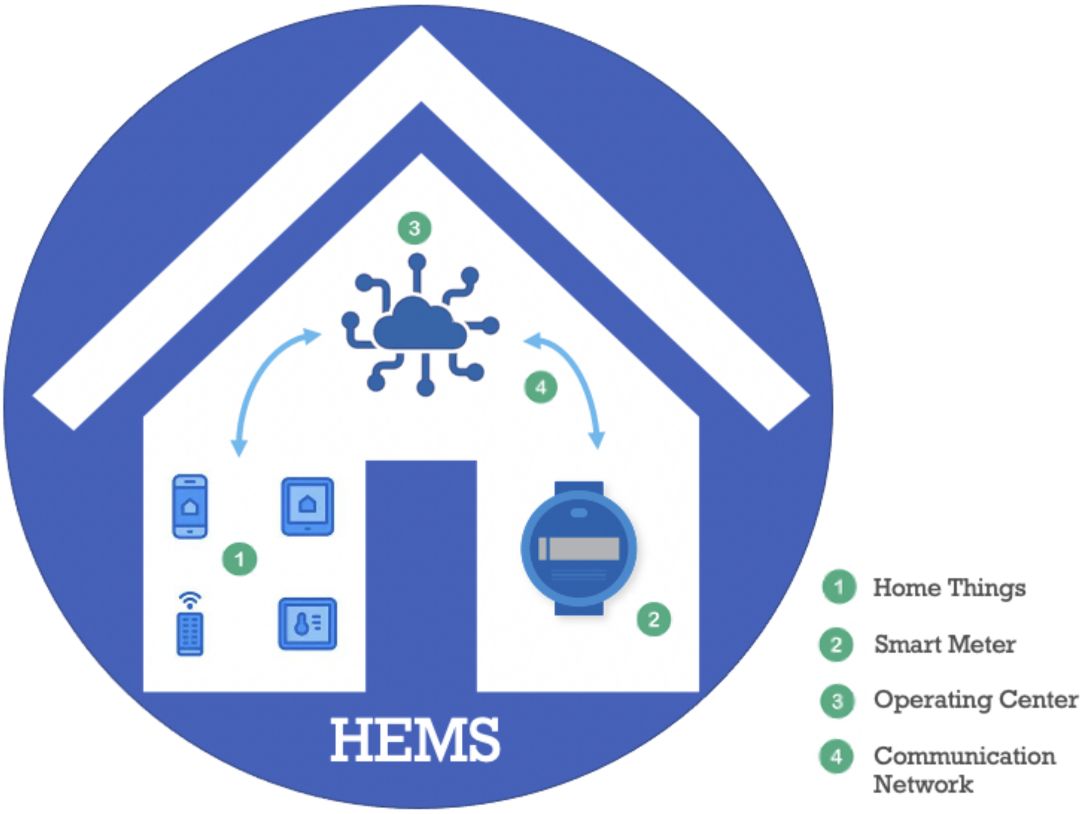


FIGURE 2.1: HEMS infrastructure

2.1.2.1 Advanced Metering Infrastructure

AMI operates with bi-directional messaging, meaning it permits communication between customer and utility (Grid, 2016). Smart meters are deployed with two-way communication technology to bypass the need for manual labor. The metering intervals range from 5 to 60 minutes depending on the use case. It is necessary to build a communication network that facilitates the continuous data stream coming from the smart meters. Such networks need to be robust in order to ensure the reliability of AMI. At the utility side, this information is used to generate a response for customers containing key information about their load profile. The concept of HEMS extends AMI as it includes the ability for the utility to directly control some home appliances through DSM.

HEMS encompasses other solutions like Energy Storage Systems (ESS) (Dinh et al., 2020), However, these are not always implemented by consumers. The following sub-sections cover the basic building blocks of the HEMS infrastructure.

2.1.2.2 Smart Meters

Smart meters allow for automatic metering of electricity consumption and production, delivered to Distribution System Operators (DSO) in either real-time or small incremental time steps (Sintef, n.d.). The ability to get real-time measurement benefits to consumers, distribution, and power supply companies. Voltage, current, and power factors are measured to provide useful information about outages, efficient grid operation, electricity prices, and stable voltage. Smart meters with bi-way communication can both receive and share messages using a wireless communication protocol (Daud, Thariq, Kaulika, Nugraha, & Adiono, 2020). Clusters of smart meters utilize bi-way communication to inform nearby meters about device errors or problems occurring on some part of the grid.

2.1.2.3 Home Things

Home Things refer to objects capable of sending and/or receiving messages from the internet. Information transmitted by home Things provides useful data about consumption patterns and preferences for residents. Some Things can also receive downlink messages to act upon specific requests. (Mahapatra & Nayyar, 2019) categorize Things into schedulable (also called curtailable loads) and non-schedulable, where the latter can not be implemented in a time scheme. Curtailable Things like AC, EV charger, radiators, lights, etc., are capable of operating with fully automatic control. Not only is this a great feature for residents; it also allows grid operators to achieve more optimal grid conditions.

2.1.2.4 Operating Center

At the heart of every HAS is a central operating unit managing the whole infrastructure. The operating center facilitates measuring, messaging, and managing in order to automatically control the HEMS. Incoming data is processed, analyzed, stored, and used to schedule Things operation. Demand-side management is a crucial functionality of HEMS, and the operating center enables this process. Based on how sophisticated the system is, one gets different levels of automatic and manual control.

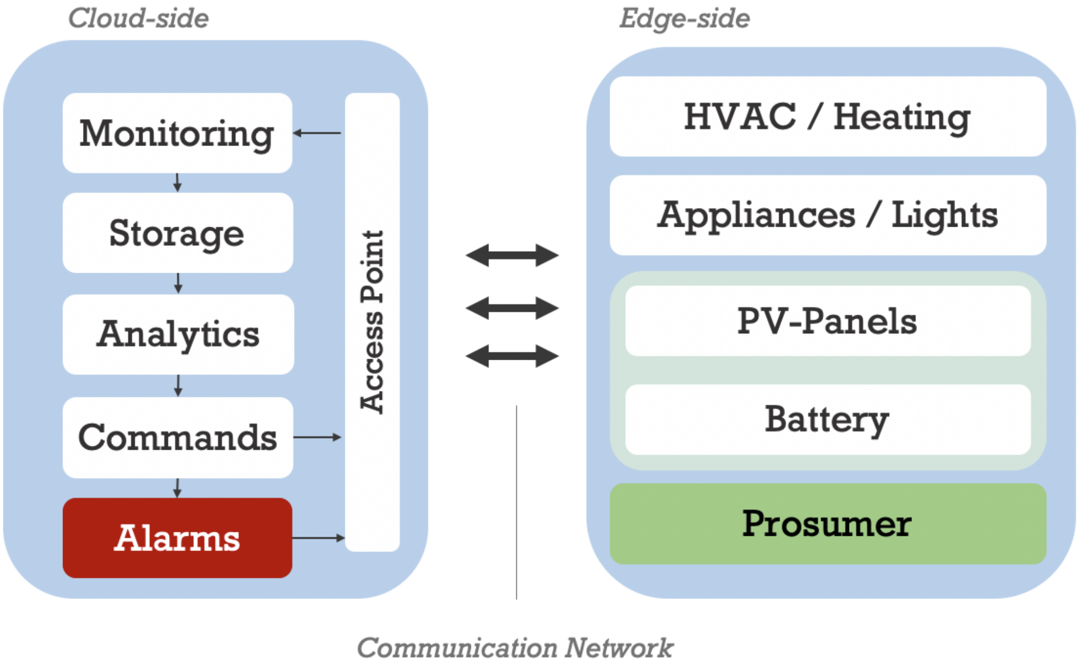


Figure 2.2: Cloud based HEMS infrastructure with operating center located on the cloud

Modern HEMS have transitioned from edge operating centers to one deployed on the cloud (Berg, 2020). The availability and accessibility of scalable computing power and storage make it simple to launch and further develop a HEMS. Data analysis, visualization, scheduling, tracking, and off-site control can be accessed by users from anywhere in the world with an internet connection. Cloud-based systems gather data through an IP-backhaul from the home gateway and transmit commands and state updates in return. The backhaul facilitates the traffic between edge and cloud networks. Common Web Services are AWS, Microsoft Azure, and Google Cloud.

2.1.2.5 Communication Network

Access to the internet in HEMS is enabled through various communication protocols based on available connectivity options inside the system. The communication network is the backbone of all HEMS and it is the most vital component in the infrastructure. The characteristics and architecture of communication networks vary. However, they are all built for the seamless transportation of large amounts of data. Popular communication protocols for smart homes include SigFox, ZigBee, LoRaWAN, Bluetooth, etc., and their deployment varies depending on the use case (Al-Sarawi et al., 2017) (Kuzlu, Pipattanasomporn, & Rahman, 2015). Wi-Fi is suitable for low-latency, high packet-size transfers, while LoRaWAN transfers smaller packets with less power consumption. The latter is an example of a Low-Power Wide Area Network (LPWAN) often used for applications with low bit-rate requirements. Sensors and other battery-powered devices can easily connect to the internet via a common gateway.

A home gateway is a central communication unit used as an access point for Things with various communication protocols. Implementing a robust communication network with more than one protocol requires a gateway to connect the home network to the internet. Most gateways have an IP-backhaul, which is a link between the outer edge and core networks. Simply put, it relays information from and to Things using different transmission mediums.

2.1.3 Prosumer Agent

An electrical prosumer produces and consumes energy. It barters electricity produced and stored on-site back to an electrical grid (Muzi, Calcara, Pompili, & Sangiovanni, 2018) 2.3. Electrical energy is produced on-site using renewable energy sources (solar, wind, hydro), which in turn can either be consumed or sold back to the power grid. Prosumers have an increased incentive for implementing a HEMS, as smart controls reduce energy costs (Han, Choi, Park, Lee, & Kim, 2014). Controllers are acting as prosumer agents, sharing information with other agents using a bi-directional communication network. The ability to exchange information and power between prosumer nodes can help stabilize the grid, reduce CO2 emissions and boost reliability. Prosumer agents are integrated into HEMS and operate within the three functionalities (three M's) mentioned above

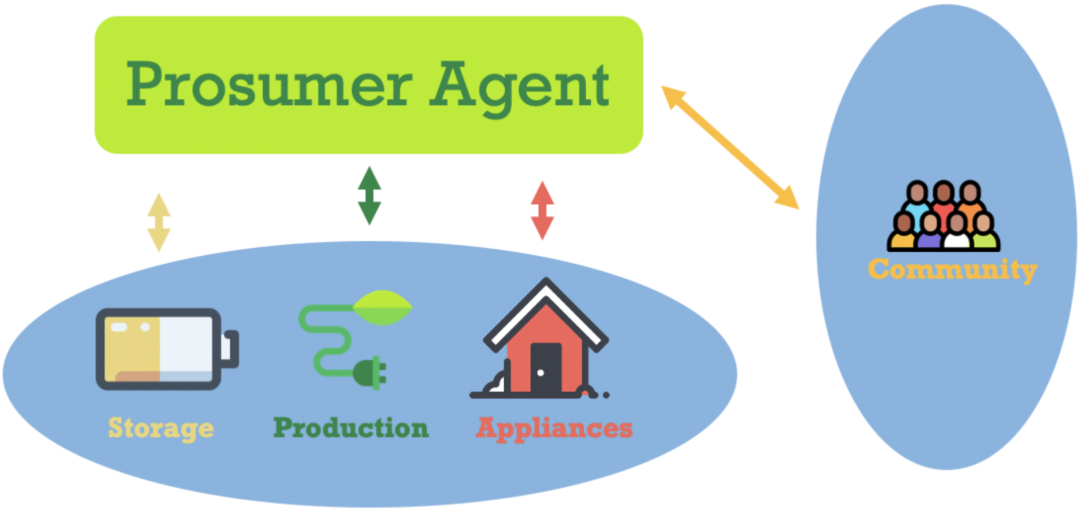


FIGURE 2.3: Prosumer agents control the HEMS in response to the connected power community

2.2 Smart Home Communication Networks

A smart home communication network is the infrastructure that enables Things in a home to communicate. It is essentially the backbone of every HAS. Things need a common language or protocol in order to receive and transmit messages and commands. One of the traditional protocols used in home automation is X10, which uses the mains power wiring to transmit signals (Withanage, Ashok, Yuen, & Otto, 2014). Although X10 is still widely used, smart homes tend to shift towards wireless communication between Things. The explosive popularity and widespread use of wireless communication contributed to the adaptation of the phrase Internet of Things (IoT).

IoT describes a system of interconnected, unique devices that can communicate over a network without any interaction from humans (Goyal, Sahoo, Sharma, & Singh, 2020). It is the main driver of the 4th industrial revolution (merging the physical and virtual world) and its applications range from landscape design (Kang, 2021) to authentication using biometrics (Shah & Bharadi, 2016). IoT is enabled in HEMS via home gateways or routers that creates a communication network for devices. The gateway acts as a bridge between the internet and Things located inside a home. Things may use different communication protocols within the same network, so installers need to ensure that the gateway conforms to the utilized protocols. It is also important to implement security measurements for the protocols used, as hackers and ill-intended intruders could potentially take control of appliances (Geneiatakis et al., 2017). Security breaches can be intentional or unintentional, and protection against both of them has to be implemented.

The IoT stack is comprised of devices, platforms, and protocols/connectivity that allows devices to communicate (*IoT Technologies and Protocols: Microsoft Azure*, n.d.). The protocol/connectivity component is vital as it details how devices connect and communicate. There seem to be some confusion as to what the definition of an IoT protocol really is. A simplified way of categorizing IoT protocols is by separating them into data protocols (application/presentation layer) and network protocols (datalink/physical layer). The IoT network protocols can be divided into long- and short range protocols. Figure 2.4 shows a range/data rate graph for the communication protocols mentioned below.

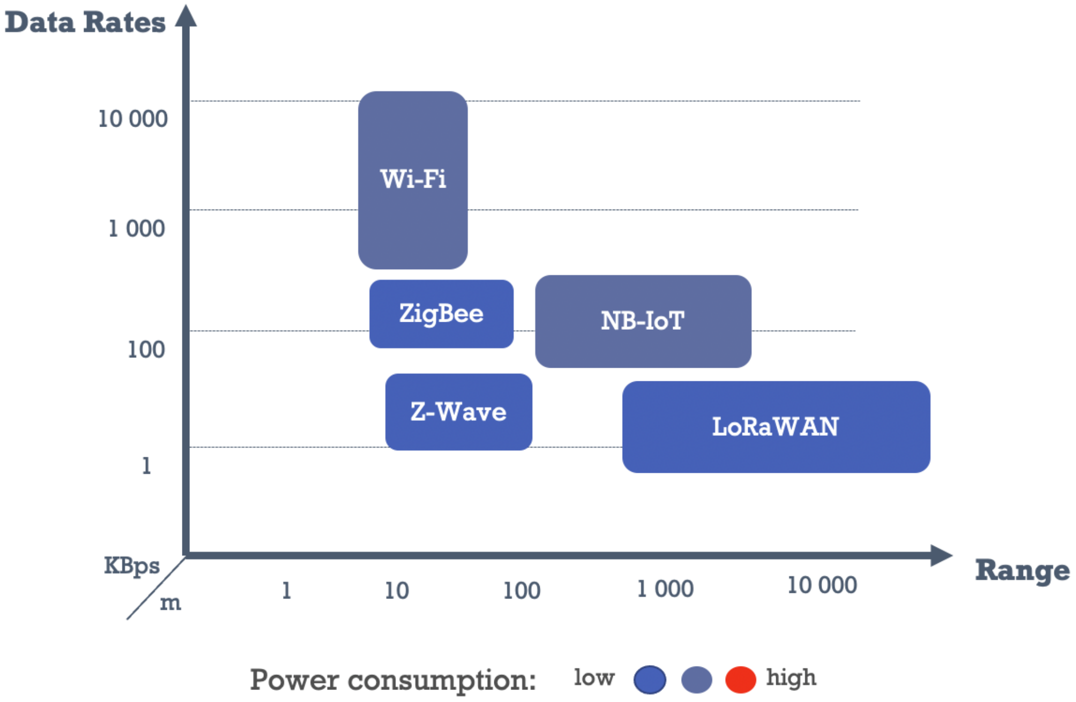


FIGURE 2.4: IoT communication protocols compared by range and data rate (Al-Sarawi et al., 2017)

2.2.1 Short/Medium Range IoT Network Protocols

This subsection includes both short and medium-range solutions.

ZigBee Alliance created the **ZigBee** protocol in 1998 and based on the IEEE802.15.4 network standard. It operates in a wireless mesh network, but can also support star topology. Its low data transmission rate makes it suitable for battery-powered devices. The range can reach up to above 200 meters (Al-Sarawi et al., 2017). However, this is almost never achievable. The average range highly depends on line-of-sight conditions, with indoor ranges rarely exceeding 100 meters.

Considered as one of the most common communication protocols, **Wi-Fi** is a group of network protocols based on the IEEE 802.11 standard supporting high data rates. The high bandwidth used in Wi-Fi makes it suitable for transmitting images, movies, and sounds. Wi-Fi is capable of reaching up to 100 meters; however, the typical router only reaches up to 50 meters. The main drawbacks of Wi-Fi, in relation to home automation, are high power consumption and its exposure to interference (Berg, 2020).

Developed by Zensys, the **Z-Wave** communication protocol is common in home automation systems and supports transmissions with up to 100 meters range. It is designed for transmitting small packets at low data rates (up to 100 kbps). The transmission radios have low power consumption, so Z-Wave devices can run on batteries. Z-Wave operates in a mesh network where slaves (nodes) are connected to the main controller (Al-Sarawi et al., 2017).

2.2.2 Long Range IoT Network Protocols

The **Narrowband IoT** (NB-IoT) communication standard is designed for IoT devices to use existing carrier networks. It is a low-cost solution based on Long Term Evolution (LTE). Devices using NB-IoT have much less power consumption due to the narrow bandwidth. 99% of the devices using NB-IoT have a latency of 10 seconds or less. One of the key selling points for NB-IoT compared to SigFox and LoraWAN is that it utilize existing infrastructure, rather than depending on new implementations (Ratasuk, Mangalvedhe, Zhang, Robert, & Koskinen, 2016).

LoRaWAN is an LPWAN protocol used in remote IoT applications. The MAC IoT protocol is designed for low power consumption with data rates ranging from 0.3 to 37.5 kbps (Berg, 2020). Since it operates on unlicensed frequency bands, anyone can set up a LoRaWAN network. A more detailed description of the networking protocol and the underlying technology follows.

2.3 LoRa: The Physical Layer

Long-**Ra**nge (LoRa) is a technology designed to fit in between Local Area Network LAN and high power-consuming LPWAN's like cellular (Sağır, Kaya, Şişman, Baltacı, & Ünal, 2019). What separates LoRa from other technologies is its low power consumption and extreme coverage area. Having such qualities makes it a well fit for Internet of Things applications where meters and sensors are placed at semi-remote locations. LoRa is not replacing existing communication technologies, but rather complementing them. It is designed to transfer simpler packets, like sensor or meter readings. This combined with the long-range makes it very useful in agriculture where many simple, yet important measurements needs to be taken.

2.3.1 Spread Spectrum Technique

LoRa is a proprietary spread-spectrum modulation technique based on Chirp Spread Spectrum (CSS) technology. This technique is what enables the LoRa. technology to reach such high ranges. Spread spectrum communications refer to methods where the bandwidth of a signal is increased, resulting in the signal being spread out over a frequency domain (*LoRa Modulation Basics*, 2015). The Shannon-Hartley theorem explains the relationship between channel capacity and channel bandwidth on a spread-spectrum signal when exposed to noise (2.1).

$$C = BW * log_2(1 + \frac{S}{N}) \tag{2.1}$$

The signal-to-noise ratio (SNR) is small (S/N <<1) due to low signal power. SNR is a minimum ratio for receivers to be able to demodulate the incoming signal (*LoRa Modulation Basics*, 2015). Combine this with changing log base 2 to natural log e results in 2.2

$$\frac{N}{S} \approx \frac{BW}{C} \tag{2.2}$$

It can be seen that with a fixed signal-to-noise ratio, the channel capacity is only dependent on channel bandwidth. This means that bandwidth can compensate for low SNR.

Direct Sequence Spread Spectrum (DSSS) is a common technique where the signal is multiplied with a chip sequence 2.5. The sequence has a higher frequency than the original signal, so the resulting signal is spread out over a frequency range. Incoming signals at the receiver side is multiplied by the same spreading sequence to retrieve the original signal Although it is widely used, it still requires expensive components and significant power. This technique might not be suitable for small battery-powered devices.

As mentioned previously, LoRa is a technique based on CSS. Instead of being modulated by a chip sequence, signals are now modulated using a sinusoidal signal pulse which increases or decreases in frequency over time (chirp pulse) 2.6. This low-power alternative to DSSS reduces the receiver complexity and increases the sensitivity.

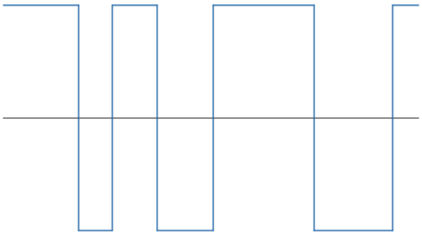


FIGURE 2.5: Chip sequence signal

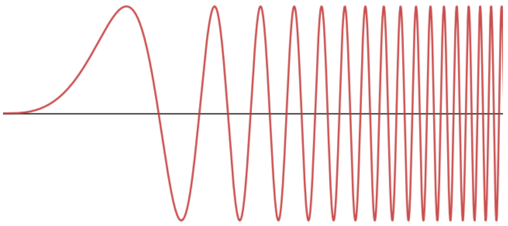


FIGURE 2.6: Up-chirp pulse

2.3.2 LoRa Modulation

The modulation bit rate for LoRa transmission is displayed in equation 2.3

$$R_b = \frac{SF * BW}{2^{SF}} \tag{2.3}$$

SF represents the **Spreading Factor** (7..12) and BW is the bandwidth. When transmitting using LoRa, signals are modulated onto a chirp signal. These signals include a collection of symbols which is further divided into chips (1 or 0). The amount of chips in each symbol is SF dependent and equals 2^{SF}, where SF is . The symbol rate is derived form SF and BW 2.4.

$$R_S = \frac{BW}{2^{SF}} \tag{2.4}$$

One chip is sent per second per Hz of bandwidth. We can use this, combined with 2.4 to get 2.5

$$R_C = R_S * 2^{SF} \rightarrow R_C = BW \tag{2.5}$$

LoRa also includes a forward error correction (FEC) scheme, which includes a coding rate from 1 to 4 representing redundancy bits. 4-bit data is encoded with redundancy bits to increase toleration against interference. Combining nominal bit rate with FEC results in equation 2.6

$$R_b = \frac{SF * 4 * BW}{2^{SF}(4 + CR)} \tag{2.6}$$

Having the possibility to easily configure many transmission variables makes LoRa suitable for a variety of applications.

2.3.3 Link Budget & Receiver Sensitivity

The link budget of a wireless transmission describes the sum of all system gains and losses (*LoRa Modulation Basics*, 2015). It is used to investigate the strength of a signal coming in at the receiver side. Th The main use of a radio link budget is to evaluate the proper transmission power (Tx power) in order to achieve satisfactory signal transmissions. The link budget can be calculated using equation 2.7

$$P_{RX}(dBm) = P_{TX}(dBm) + G_{system}(dB) - L_{system}(dB) - l_{chanel}(dB) - M(dB) \tag{2.7}$$

where:

P_{RX} = Expected power at receiver side
P_{TX} = Transmission power
G_{system} = System gains (antenna gain)
L_{system} = System losses
L_{channel} = Propagation channel losses (calculated or empirical value)
M = Fade margin (calculated or empirical)

The link margin explains the difference between Received Signal Strength Indicator (RSSI) and receiver sensitivity. RSSI is measured (in dBm) at the receiver side, and it represents the received signal strength of the link. One can simply calculate the link margin by taking RSSI minus receiver sensitivity. System operators should ensure a positive link margin with some dB's of headroom in order to have a reliable and efficient link.

It is important to mind the receiver sensitivity when planing a LoRa communication system. The sensitivity is calculated using BW, noise figure (NF), and SNR ("SX1276/77/78/79", 2020) 2.8. NF is a fixed value for the selected hardware.

$$S = -174 + 10log_{10}BW + NF + SNR \tag{2.8}$$

The minimum SNR level is directly affected by SF, and it drops 2.5dB for each increase in SF. Table 2.2 displays the SF (exponent and chips per symbol) and corresponding SNR levels for SX1276/77/78/79.

SF (exponential)	**SF (chip/symbol)**	**Minimum SNR**
6	64	-5 dB
7	128	-7.5 dB
8	256	-10 dB
9	512	-12.5 dB
10	1024	-15 dB
11	2048	-17.5 dB
12	4096	-20 dB

TABLE 2.2: SNR values for selected SF values

2.3.4 LoRa Signal Transmission

LoRa packets comprise a preamble, an optional header, and a payload ("SX1276/77/78/79", 2020) 2.7. The preamble is used by the receiver to detect the start of each packet. Following is the header, in which payload length, forward error correction rate, and configuration for an optional redundancy check on the payload. The payload is the actual message and includes the cyclic redundancy check bits. The presence of a header is only available in explicit mode while implicit mode uses fixed header values. This decreases the packet size, but also requires the receiving end to know the exact header information beforehand.

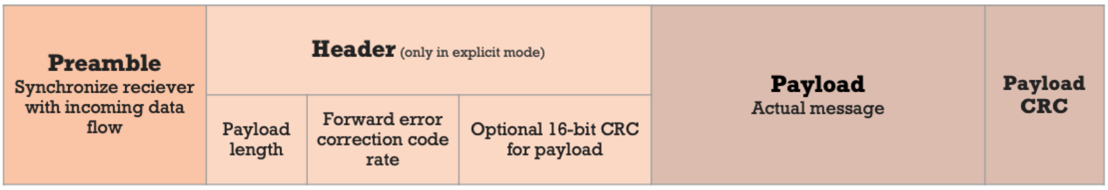


FIGURE 2.7: LoRa packet format

2.3.4.1 Time on Air

Time on air (ToA), also known as packet duration, is the total amount of time for the whole signal to reach the receiver. ("SX1276/77/78/79", 2020). $\mathrm{T}_{\mathrm{packet}}$ is the total sum of transmission time for preamble and payload. The preamble duration is calculated using 2.9 where symbol duration (T_{s}) is derived from 2.4. $\mathrm{n}_{\mathrm{preamble}}$ is predetermined (= 8 for EU868) and represents the number of preamble symbols.

$$T_{preamble} = T_s(n_{preamble} + 4.25) \tag{2.9}$$

Following is the payload time ($\mathrm{T}_{\mathrm{payload}}$), which is calculated using a fairly more complex formula 2.10. The expression inside the parenthesis amounts to the number of symbols occupied by the payload.

$$T_{payload} = T_s * (8 + max(ceil(\frac{PL - 4SF + 28 + 16CRC - 20H}{4 * (SF - 2DE)})(CR + 4), 0)) \tag{2.10}$$

where:

PL = Payload size (in bytes)
CRC = Cyclic Redundancy Check (0 = off, 1 = on (default))
H = Header (1 = disabled, 0 = enabled (default))
DE = LowDataRateOptimize (1 = enabled, 0 = disabled)
CR = Coding Rate (1-4, default = 1)

By summing the ToA for preamble and payload, we get the ToA for the whole packet

$$T_{packet} = T_{payload} + T_{preamble} \tag{2.11}$$

With 2.11 it is possible to calculate the total ToA for a single LoRa packet. This formula is used to investigate the optimal payload size and data rate configuration for end-node deployments.

2.3.4.2 Free-Space Loss

The signal propagating in the real world is affected by many external sources that can cause changes to the transmitted signal. Free space propagation is when signals travel in free space, only being affected by the path loss (Hogg, 1993). Equation 2.12 is the Friis transmission formula used to investigate free space propagation with isotropic antennas.

$$\frac{P_r}{P_t} = \frac{A_r}{4\pi d^2} \tag{2.12}$$

with

$$A_r = \frac{\lambda^2}{4\pi} \tag{2.13}$$

Combine and inverse 2.12 and 2.13, and replace wavelength with frequency, results in the free-space loss formula ($\mathrm{P_t}/\mathrm{P_r}$ in terms of dB)

$$FSPL = 20log(\frac{4\pi df}{c}) = 20log(d) + 20log(f) + 32.4 \tag{2.14}$$

Note that the equation above operates with km and GHz for distance and frequency. 2.14 represents the propagation losses (path loss) in 2.7. Rearranging this equation for the distance and inputting the receiver sensitivity for path loss results in the maximum theoretical distance for LoRa.

2.3.4.3 Fresnel Zone

The Fresnel Zone is defined as an elliptic cylinder zone stretching around the direct line-of-sight between end-nodes and gateways (Jebril, Sali, Ismail, & Rasid, 2018). It is important to consider this zone, as trees, structures or any obstacles inside the zone contribute to energy losses in the signal. signals transmitted simultaneously might arrive at the receiving side with some time variation. This is caused by interfering objects reflecting the signal via another path. A general rule of thumb is that the Fresnel zone should be equal or above 60% free from obstacles, so that the wireless link is strong enough. Equation 2.15 can be used to calculate the radius of the Fresnel zone. Note that this measures the max co-vertex value of an ellipse, which is at the halfway point 2.8.

$$r = 8.656 * \sqrt{D/F} \tag{2.15}$$

r = Fresnel zone minor radius

F = Signal frequency

D = Distance between transmitter and receiver

In order to calculate the percentage of clear Fresnel zone, one must first know the elevation profile of the line between transmitter and receiver. This is best done by using an elevation mapper to accurately get the height profile. One important thing to be aware of when utilizing Fresnel zone calculations is the curvature of the earth. Some elevation mappers consider this fact; however, it is not crucial for scenarios where the distance between Tx and Rx is below 4km.

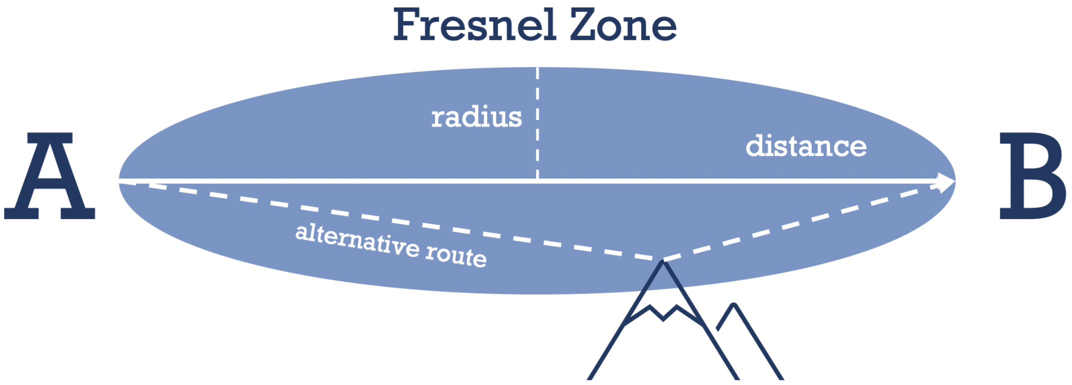


FIGURE 2.8: Signal transmission illustration with Fresnel zone

2.4 LoRaWAN: The MAC Protocol

The following subsection includes relevant information about the LoRaWAN protocol. Further information can be found in the LoRaWAN 1.1 Specification sheet (*LoRaWAN 1.1 Specification*, 2017). Information presented in this section is derived from this source, if nothing else is stated. Note that this sheet covers the 1.1 version of the network protocol, and some older end-nodes only conform to 1.0.x specifications.

LoRaWAN is sometimes used in parallel with LoRa, and they can be mistaken for the same concept. While LoRa encapsulates the physical layer with radio modulation technology, LoRaWAN is a network protocol using LoRa as the physical layer. It is a Low Power Wide Area Network (LPWAN) designed for transmitting small data packets from battery-powered devices. The LoRaWAN specification dictates three components required in a LoRaWAN network:

- **End-Devices** are the physical nodes representing the edge of the network. These are low-power sensors or actuators, communicating with a central gateway

- **Gateways** collect packets from nodes and forward them to a network server through an IP-backhaul. Large amounts of end-device data can be sent to the gateway in a star topology.
- **Network Servers** are responsible for handling the incoming packages and managing outgoing packets to end-devices.

It is common for LoRaWAN server providers to offer an additional application server for handling decoding and encoding, and downlink queuing. Some providers also host an MQTT broker, which is used to further delegate the uplinks. The LoRaWAN architecture (with application server) can be viewed at 2.9.

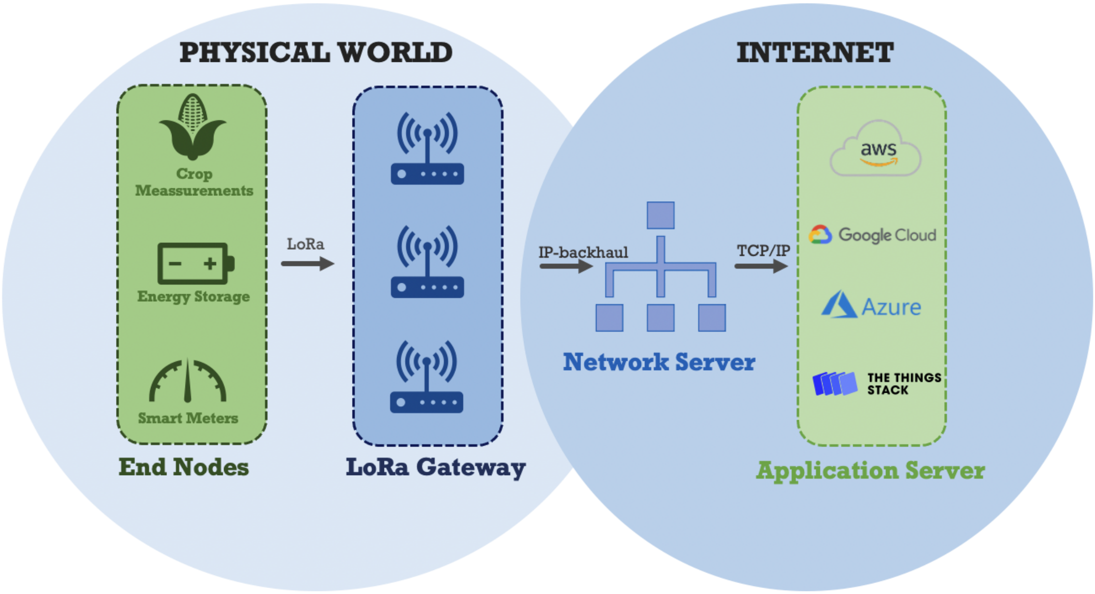


FIGURE 2.9: Architecture of LoRaWAN network from nodes to application server

2.4.1 LoRa End Nodes & Gateways

LoRa end-nodes are small MCs mounted with a radio module and an antenna. Sensors or actuators connect to the microprocessor which in turn delegate incoming and outgoing messages. There are three different endpoint device classes used in LoRaWAN. They are all bi-directional and each class (except A) extends the functionality of Class A (Lavric & Popa, 2017). Deciding on which class to use highly depend on the intended use.

2.4.1.1 End-Node Classes

Class A devices uplink data to gateways at any time needed. Following the uplink is two short time slots where the device can receive messages from the gateway 2.10. End nodes specify the receiving delay (Rd.) for both downlink windows. The gateway can therefore not initiate communication before the device uplink. This device type uses the least power of all of the classes. **Class B** devices behave in the same way as Class A devices, but they also have extra time slots for receiving. The gateway initiates communication with the end nodes using beacon frames, that way they receive data at specified times. These devices use more power than Class A, but in turn, have a deterministic downlink latency. Note that Class B devices are less common than A & C.

Lastly we have **Class C** devices, extending all of the functionalities of Class A but with continuous downlink when not transmitting. Since end nodes can receive data whenever they are not transmitting, the resulting power consumption is higher while the latency is quite low. Class A devices are able to switch to Class C type, but not to Class B.

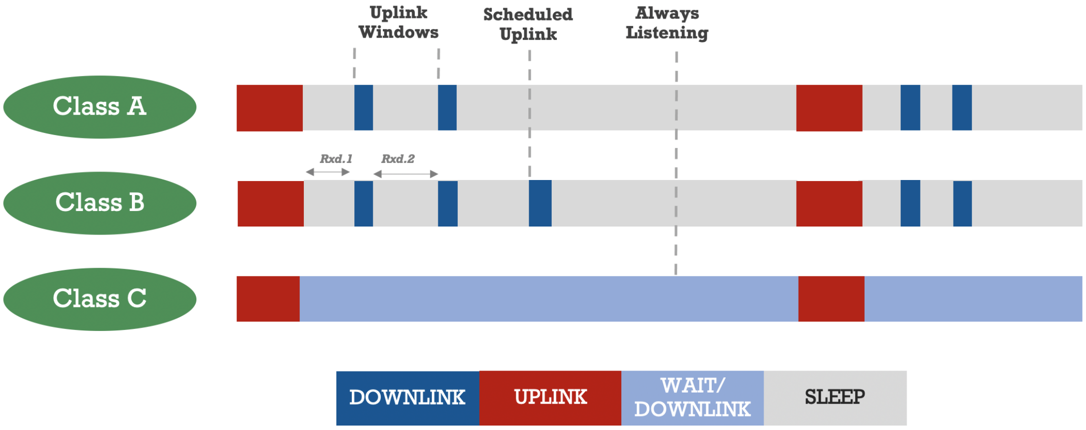


FIGURE 2.10: Characteristics of LoRaWAN classes A, B and C

2.4.1.2 Gateways

Similar to the end-nodes, LoRa gateways also require a radio module with an antenna connected to a microprocessor. Furthermore, it is equipped with a backhaul option (IP/Cellular) linking end-node networks to the internet. They are mains powered due to the power requirements for backhaul communication methods. Simultaneous incoming messages are handled by multiple channels on the gateway, as long as the gateway is not single channeled. In areas with overlapping gateway coverage, some end-node uplinks might be received by more than one gateway. If multiple gateways receive the same message, and they are connected to the same network server, then the network selects the gateway with best reception.

2.4.2 MAC Message Formats

When a LoRa chip demodulates an incoming signal, it passes along the PHY Payload. If the signal comes from an uplink performed by an end-device, then the PHY Payload contains a MAC Header, MAC Payload, and MIC 2.11.

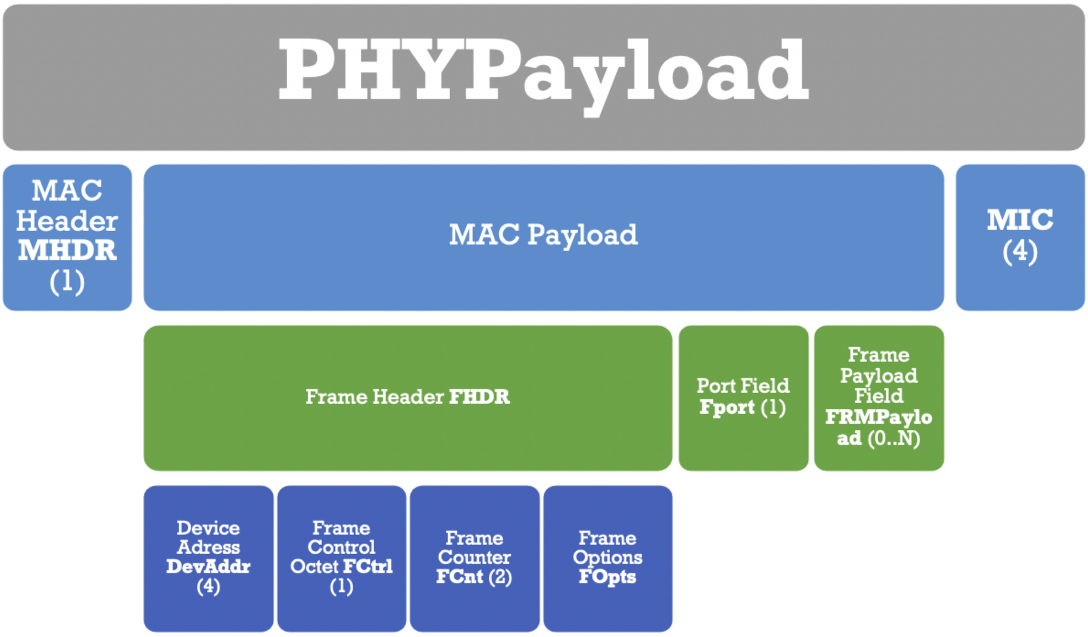


FIGURE 2.11: Physical payload format

The **MHDR** field specifies the message type and which major (LoRaWAN version) used for encoding. The **MAC Payload** contains a frame header (FHDR) with an optional port field (FPort) and frame payload field (FRMPayload), where the last one contains the actual payload message. Enclosed in FHDR is the end-device address (DevAddr), which contains seven bits for identifying the network and 25 bits for the end-device network address. Furthermore FHDR contains a frame control octet (FCtrl), a frame counter and frame options containing optional MAC commands. The last part of the physical payload is **MIC**, which is the message integrity code. Note that the LoRaWAN header contains a minimum of 13 bytes and a maximum of 28 bytes. Those bytes are added to PL in the equation 2.10, which is then multiplied by the symbol duration. This has to be accounted for in ToA calculations.

2.4.3 End Device Activation

In order for end devices to communicate with the LoRaWAN network server, an activation process needs to be performed. There are two methods end-devices can be personalized and activated: via Over-The-Air-Activation (OTAA) or via Activation By Personalization (ABP)

2.4.3.1 Over-the-Air Activation

Devices employing OTAA needs to follow a join procedure every time it wants to establish a new session. Before initiating the join procedure, end devices needs the following information:

- **DevEUI** is the unique 64-bit global identifier (EUI-64) assigned by IEEE. In OTAA, end devices must store this if locally in order to join the network server.
- **JoinEUI** is a unique EUI-64 that identifies the join server, which in turn assists the join process for end-devices.
- **NwkKey & AppKey** are root keys specific to each end-device used to derive session keys. The provisioner of end-devices has to ensure secure usage and storage of these keys, in order to improve overall security

The join procedure consists of a join request and a join-accept exchange. When end devices initiate a join request, they send a join-request message containing the JoinEUI and the DevEUI. When the network server receives the join-request (or rejoin-request), it responds with a join-accept message only if the end device is authorized. The accept request is received by end-devices as downlinks coming in join-accept windows similar to the receive windows detailed in 2.10. If the end device receives the join-accept downlink, it can start the payload uplink to the network server.

2.4.3.2 Activation by Personalization

ABP removes the need for a join procedure, as end devices store all of the necessary information and keys. Instead of using the JoinEUI, DevEUI, AppKey, and NwkKey to derive the session keys and device address, it is all located on the end device. What's important with this activation method is that no public information can be used to derive the session keys.

2.4.4 Security Properties

Security is a key component in LoRaWAN networks, and it has been integrated with the protocol from day one. Fundamental security properties in LoRaWAN are mutual authentication, integrity protection, and confidentially. Although security is integrated into the protocol, it is important to mind implementation when managing security in a LoRaWAN network (Gemalto & Semtech, 2017).

Mutual authentication or two-way authentication is a security measure that authenticates two devices before they begin to talk. This process is initiated in parallel with the join procedure.

Integrity protection is integrated in LoRaWAN to ensure that the messages have not been tampered with. Confidentiality is security measurements integrated to ensure protection from eavesdropping. LoRaWAN comes with end-to-end encryption, which reduces the need for additional over-the-top security layers.

2.4.5 Spreading factors & ADR

(Seye, Ngom, Gueye, & Diallo, 2018) used a Spreading Factor (SF) of 12 to measure a maximum of 10km range with a satisfactory RSSI. Some have even been able to successfully transfer packets over 702km (Network, n.d.). Selecting the right spreading factor for a LoRaWAN highly depends on the intended use. The key is to balance battery lifetime with communication range. The spreading factor reflects the number of chirps being transferred per second. A low SF translates to more chirps sent per second. Higher SF requires more transmission time, which results in better receiver sensitivity. On the other hand, increased transmission time also results in higher power consumption. Often the SF is decided by the network using ADR, which analyses environmental conditions to select the best value.

Adaptive Data Rate (ADR) is an end-node functionality where data rates are optimized for energy consumption and stability (*Adaptive Data Rate*, n.d.). It is an optional functionality, and should only be active when the location (and surrounding conditions) are stable. ADR is usually determined from the last 20 uplinks (ADR bit is set before the uplinks are measured). When the measurements are collected, we select the gateway with the best SNR and compute the difference between measured and required SNR. This difference is called a link margin, and indicates how much we can increase the data rate or reduce the transmission power. If the measured SNR is 0 and the data rate is SF8BW125, we have a 10dB margin (see table A.3. Hence we could increase the data rate or reduce transmission power to close the margin gap.

2.4.6 Tx Power and Max EIRP

Tx power is the same as end-device EIRP, which is Equivalent Isotropically Radiated Power. EIRP is the radiated power output of an antenna in reference to an isotropic antenna. It essentially is a measurement of the signal strength leaving the antenna. Max EIRP is generally +16dBm for end-nodes; however, it is important to notify the network server if the end-node can not achieve this power level. For programming with AT-commands, users can set the EIRP configurations according to TX power provided in (“LoRaWAN Regional Parameters”, 2020) A.5.

2.4.7 Regional Parameters

One of the key benefits of LoRa communication is that it is operating in an unlicensed radio spectrum (“LoRaWAN Regional Parameters”, 2020). There is no need for any license or permit to use the frequency bands, as they are open to the public. LoRaWAN uses lower radio frequencies (470, 868, 915 MHz) compared to WiFi (2.4, 5 GHz), which is why it is able to operate over much longer distances. Countries around the world have different restrictions on lower radio frequencies, which reduce the consistency of LoRaWAN around the globe. Therefore, one should always control that the operating frequency spectrum is matching the regional specification. The table 2.3 contains detailed information about LoRaWAN parameters specific to EU868.

Parameter	**Value**	**Description**
Default Frequency Bands	863 - 870 MHz	All LoRa end-nodes shall be able to operate within the default frequency band range. Gateways must at least be able to use the three listen in parenthesis.
Bit Rate	0.3-5 kbps	The physical bit rate depends on the bandwidth and SF. Increased bandwidth (125-500 kHz) results in bit rates up to 11 000 bit/s.
EIRP	max = +16 dB	The EIRP stands for Equivalent Isotropically Radiated Power and refers to a perfect antenna radiating equally in all directions. It consists of Tx radio gain, cable loss, and antenna gain.
Duty Cycle	<1 %	Represents the maximum time-on-air percentage during a day. If the total ToA for one uplink is 500ms, we would have to wait for 99*0.5s for the next uplink.

TABLE 2.3: LoRaWAN Regional Parameters

Duty Cycle is a key concept in LoRaWAN stability. It is implemented to prevent over-usage of the unlicensed frequency band. Duty cycle = 1% for EU makes 864 seconds of allowed ToA during one day. Dividing the daily maximum allowed ToA by the ToA of a single uplink gives the maximum allowed uplinks during one day. End-nodes should be programmed to never uplink more messages than what is allowed for the given duty cycle.

2.5 Cloud-Based Operating Center

Cloud computing is the deployment of on-demand IT resources from public or private clouds, accessible by users via an internet connection (Peter Mell, 2011). The National Institute of Standards and Technology (NIST) has defined CC as pools of decentralized services like storage and network. Furthermore, they provide a list of the three most common service models; Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS) 2.4.

2.5.1 Cloud Computing

Accessibility and scalability are possibly the key selling factors for integrating CC in HAS (Soliman et al., 2013). Computing resources are deployed when needed and scale proportional to system expansion. CC uses a pay-per-use transaction model which decreases redundant costs. Additionally, it is more user-friendly than the on-site deployment of resources since it is maintained by IT professionals. Cloud computing services are provisioned in private, public, and community clouds (Peter Mell, 2011) 2.5.

SaaS	**PaaS**	**IaaS**
Softwares are hosted on cloud providers' infrastructure, which is maintained by their personnel. Consumers can access this software through client devices like web-browsers. They are not responsible for maintaining the infrastructure, as the service is strictly focused on providing the software.	Instead of hosting software, cloud providers also offer platforms for consumers to operate their own applications. Providers support a variety of tools, programming languages, and libraries for consumers. Users do not have access to the underlying infrastructure, but can however manage the deployed applications.	IaaS allows users access to cloud infrastructure services like processing, storage, and networks. Users can deploy their own applications and operating system. The user has control over OS and applications but does not have access to the underlying cloud infrastructure.

Table 2.4: Cloud computing service models

Private Cloud	Community Cloud	Public Cloud
Distributed only to consumers with access keys. Often distributed by a single corporation or organization. This solution can either be on-site or operated by a third party off-site.	Cloud infrastructure managed by multiple corporations or organizations. It is deployed for consumers within a certain interest group, across establishments. The infrastructure is managed by one or more parties including third parties.	Provisioned to the public by governments, businesses etc. It is available to all customers with internet access, and consumers pay only for what they use. Businesses using public clouds can deploy their solution faster compared to using private clouds.

TABLE 2.5: Cloud deployment models

2.5.2 Cloud Platform Infrastructure

CC infrastructure is comprised of front- and back-end deployment (see 2.12). Each cloud deployment needs a **front end**, so end-users can view and interact with the service (Berg, 2020). Crucial for the front end is an interface, software (f.ex web browsers), and client device. The front end is a connection medium between consumers and the **back end**, which is where physical hardware operates provisioned cloud services. Cloud storage solutions like Dropbox utilize the storage component of the back-end infrastructure to store files located in data centers. The application component also uses back-end storage, in addition to computing power managed by the cloud provider's back end. This layer needs to be sturdy, as it forms the foundation of cloud services.

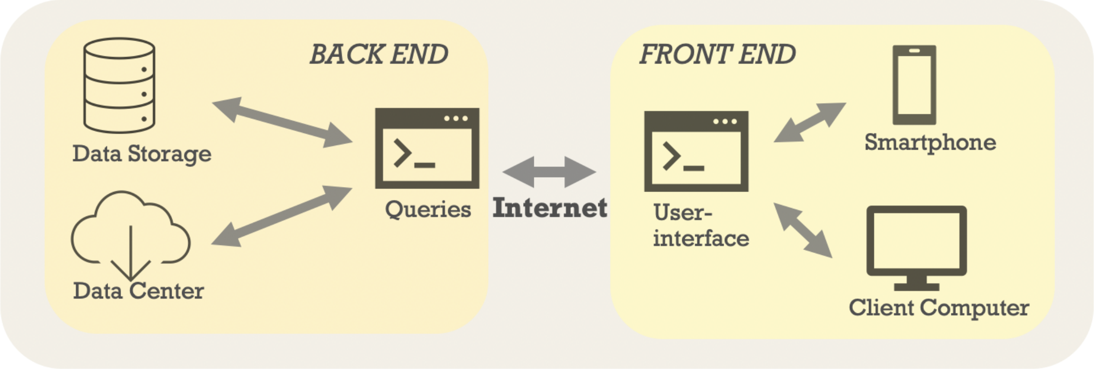


FIGURE 2.12: Cloud infrastructure comprised of front- and back end (Berg, 2020)

2.5.3 Cloud Service Providers

Several companies offer cloud services to the public, including Amazon Web Services, Microsoft Azure, and Google Cloud. In Q4 2020, the market share of those mentioned above amounted to 61%, with AWS accounting for 32 % in total (Richter, 2021). The top three providers have by far the biggest portfolio of services provisioned, offered in many regions. One research paper presents a criteria guide for evaluating the quality-of-service for various CSPs (Lang, Wiesche, & Krcmar, 2018). Their results support the hypothesis that customers are concerned about constant changes in cloud environments. Reliability and functionality are therefore a crucial part when selecting cloud providers.

Before implementing a cloud-based system, we first need to decide upon which CSP to use. Note that this paper does not perform extensive research on cost/performance between CSPs. The selection process of CSP was only based on LoRaWAN compatibilities and free-user services. Access to educational material for new users was also considered an important factor. If such a system is to be implemented commercially, the cloud platform would have to be user-friendly and ”simple” to learn. Prosumers might not want to spend too much time studying and learning the platform. Considering the above qualities, AWS was selected as CSP.

Customers of AWS have access to a large pool of services for use in their projects. This includes computes, databases, blockchain, IoT services, and many more. Designing the cloud architecture always starts with dissecting the intended application in order to identify what services are needed. A solution architect is, in fact, a certification within AWS with the sole purpose of identifying services and designing reliable cloud architectures. Although there are hundreds of services available, some are more used than others. IoT solutions with MQTT messaging and incoming data handling require both IoT- and storage services. It is important to identify the vital services in a prosumer cluster operating center; services that are required in order for the system to work. Other services can also be implemented, although they are not crucial for the systems functionality.

2.5.4 AWS IoT Services

IoT services fully manage and operate IoT devices. IoT Core is a central operating service for handling incoming and outgoing messages to devices. It includes many sub-services for managing device fleets, security, and rules. AWS provision and manage servers dedicated to run these services, allowing customers to focus on implementation and application. Things communicate with AWS IoT through four supported communication protocols:

- **MQTT** (Message Queuing Telemetry Transport)

- MQTT over **Websockets Secure**
- **HTTPS** (Hypertext Transfer Protocol Secure)
- **LoRaWAN** (Long-range Wide Area Network)

MQTT is a publish and subscribe messaging protocol ideally used for remote, low-power devices. Devices publish messages to a topic, or receive messages by subscribing to a topic. It is common to utilize MQTT for communication between LoRaWAN gateway and network server. Most often (the case in this paper) the network server acts as an MQTT broker, managing incoming and outgoing messages. Published messages are forwarded to the IoT rules engine, which is used to act upon payload content.

Security measurements like authentication and authorization are deployed by AWS to secure the IoT Core. Devices connecting to AWS IoT core receive an X.509 certificate used to authenticate the server. The certificate helps AWS to authenticate both client and device connections. After successfully authenticating devices, AWS authorizes device permission based on a pre-defined policies. AWS policies are JSON documents used to define the authority level of authenticated devices. Identities accessing IoT Core must be assigned policies before they can execute on the platform.

The rules service within IoT Core performs actions based on the MQTT topic stream. AWS uses rules to make automatic operations based on incoming messages from the MQTT broker. The rules are written with SQL-like syntax. Adding rules require a rule query statement and a predefined action from supported AWS rule actions. Some of the possible actions are storing message in DynamoDB, invoking lambda function and sending a message to IoT Events. Although Amazon only supports a limited amount of actions, it is possible to work around it by using lambda functions to connect more services. Furthermore, it is possible to create topic rule destinations to route messages across services not integrated with AWS.

2.5.5 AWS Storage/Database Services

The main reason for connecting so many devices on the internet is collecting data about the real world which needs to be stored somewhere. All computerized applications or projects use some form of repository to store information and access it whenever it needs. Implementing a solid storage solution, whether the system is cloud-based or not, is a requirement for IoT solutions.

The most common storage solution is on-site storing, which utilizes computer hardware located at the site of the application. It is a good solution for projects that do not require near-future-up scaling and/or need a high level of security for protecting the data. Moving storage solutions to the cloud seems to have several benefits compared to traditional on-site storage; Storage is much more scalable and available, also it won't require investment costs for hardware. Users only pay for the storage they actually provision. Although cloud storage is the simpler choice, beginners find that it requires a bit more know-how than drag-drop that the common person is used to. AWS storage solutions are grouped into two categories: databases and storage.

AWS storage services like S3 and Elastic Block Storage (EBS) are mostly used for storing various files, like images, videos, and other unstructured data types. EBS provisions a block of storage which the user can choose to do whatever on, even install a database. S3 is a Simple Storage Service using object-based storage for storing both the data and metadata to quickly find the desired file. The service is not optimized for frequent file updates, which makes it more suited for storing images or sound files.

AWS database is actually a form of storage since it could be installed on block storage like EBS. Databases are mostly used for storing plain text data in a structured manner. The word database not only encapsulate the actual data, but also a database management system (DBMS) and the applications built on top of it. Data is structured in tables with rows and columns, accessible with a key that identifies the specific row. AWS provides many different database types, including relational and key-value. AWS DynamoDB is a document-based database utilizing the key-value principle. The low-latency database is suitable for IoT solutions with more complex device information. An alternative for IoT is AWS Timestream, which is a serverless database type used for storing and analyzing IoT events. This paper proceeds with DynamoDB as a storage solution due to the large number of parameters linked with one event (up-link). It is also the only database available in AWS free tier.

2.5.6 AWS Compute Services

Data applications need computing power to operate and run code. AWS offers computing services like virtual machines and serverless computing to fit the various needs for customer applications. This paper utilize a serverless computing service called Lambda to run code in the cloud. Server capacity is automatically managed by infrastructure operators on the AWS side, and users only choose a programming language. The code is uploaded to AWS Lambda and saved as a Lambda function. It is then set up to be triggered from other services or HTTP endpoints.

As mentioned previously, IoT rules can invoke Lambda functions to call AWS- or third-party services. When rules trigger Lambda functions, they are invoked asynchronously. Functions are queued for processing and Lambda immediately returns a response, instead of waiting for the function code response (synchronously). One of the many use cases for Lambda functions in relation to IoT is formatting MQTT messages and forwarding them. This use case is later deployed for storing LoRa uplinks in DynamoDB.

2.6 Demand Response

Demand Side Management is a collection of actions taken by utility to shape the load curve of consumers (*Demand-Side Management Glossary*, 1992). Actions are performed on-site and may or may not be human-controlled. DSM includes two functionalities; load shifting and load reduction (Anjana & Angel, 2017). Shifting loads from peak hours to off-peak hours will not only financially benefit the residents, but it will also reduce stress on the electricity grid. The foundation for load shifting and reduction is established with good, two-way communication between grid companies and consumers. Data flowing through the communication network is then analyzed and used to perform DSM in the building. Some actions may be automated, while others require human effort. This means that the system has an x-factor (humans), which must not be forgotten.

Demand Response is a promising DSM technique, which involves prompting homeowners to change their current electricity consumption in response to electrical price and incentives(*National Action Plan on Demand Response*, 2010). Like with DSM, it helps to categorize the concept of DR to better understand it. DR can be dispatchable or non-dispatchable (Incentive-based or price-based), depending on whether the customer has agreed to a planned change in consumption.

Employing dispatchable DR actions like direct load control of appliances results in consumers receiving lower rates (this arrangement is called curtailable rates) or other types of compensations from RTOs/ISOs. Consumers can ”sell” their consumption in an organized market, and when bought receive directions on how to manage the consumption sold. Consumers who value the freedom to choose when or what appliances to manage can sign up for a non-dispatchable program. which offers dynamic pricing programs. The fluctuation in price during the day (also during the year) is meant as an incentive for consumers to reduce consumption at peak-demand hours without signing up for a plan. To sum it up, DR is any on-site actions (automatic or not) performed over a short time in response to shaping the total load profile of an electricity grid.

2.6.1 Incentivized DR

Incentivized DR refers to utility programs where users receive compensation for residential load control (automatic or human) (Paterakis, Erdinç, & Catalão, 2017). Utility issues load control calls based on reliability events and peak-demand. Residential loads like AC and lighting can be managed by utility through **direct load control**. End-users receive discounts and benefits for signing up, and in turn agree to some amount of unnotified load control by utility. Programs for curtailable load services binds consumers to respond to calls from utility agents, prompting curtailable loads to be turned off. **Demand side bidding** is another incentivized DR solution closely related to DSM; rather than encouraging load reduction and habit changes, DSB encourages consumer flexibility by involving them in the process of electricity trading. The market-driven process of short-term electricity trading opens the possibility for consumers to earn clean and easy money, much like the stock market.

2.6.2 Time-Of-Use Tariffs

Price-based DR programs, also called ToU tariffs are time-varying price signals available for customers to voluntarily act on by managing power consumption (*Time-Of-Use Tariffs: Innovation Landscape Brief*, 2019). ToU tariff structure consists of **static, dynamic** and **locational pricing**. The latter tackles electrical clogging with node pricing at a connection point, providing further incentives to prosumers for injecting electricity if congestion occurs. Static pricing involves blocks in time with different pricing levels. Block length and price is decided in advance, and usually stretch across many hours (day/night f.ex.).

2.6.2.1 Dynamic Pricing DR

In order to realize a dynamic pricing DR solution, it is crucial to establish AMI (Grid, 2016). HEMS actually require AMI deployment to access all DSM services, since AMI requires smart meters, a communication network, and an operating center. Real-Time Pricing (RTP) is hourly pricing (or even more fragmented) and updates continuously over the day based on most recent measurements (Paterakis et al., 2017). Combining static and real-time pricing results in Variable Peak Pricing (VPP), which only deploys real-time market price on scheduled periods. Another hybrid ToU pricing method, Critical Peak Pricing (CPP), is synonymous to long term variable peak pricing. This pricing scheme raises the electricity price substantially for some short periods (extreme days) during the year (*Time-Of-Use Tariffs: Innovation Landscape Brief*, 2019).

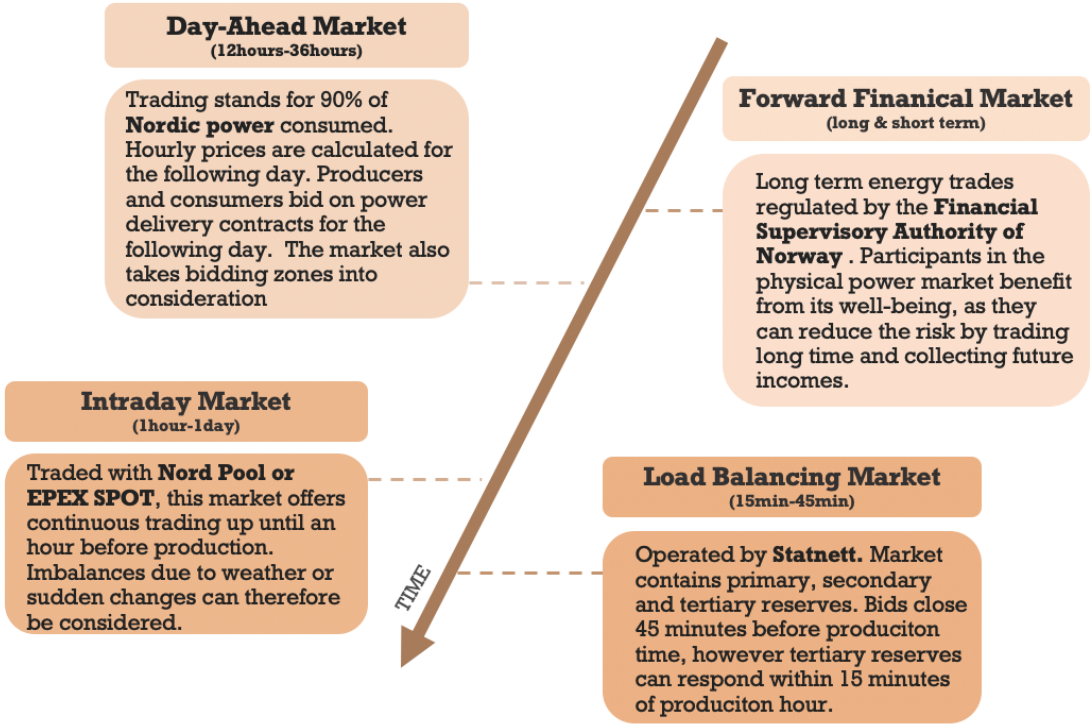
2.6.3 Prosumer-Based DR

Prosumers operate under three business models(Askeland et al., 2018); Peer-to-peer, Prosumer-to-grid and Organized Prosumer Group models. Currently, Norway does not have the necessary regulations to support P2P. P2G is the most common solution, and involves transaction between the prosumer and the grid operator. Here, the main goal is maintaining energy efficiency. The OPG models include prosumer clusters pooled together, like a micro-grid. Power trading happens through an **aggregator**, which handles the input/output between internal and external net.

Price-based models for DR is a popular choice for prosumers. (Liu et al., 2017) suggests an energy-sharing model for P2P prosumers with more economical operation than traditional, independent operation. They implement an Energy Sharing Provider (an aggregator, mentioned above), which is responsible for facilitating the energy sharing between prosumers and utility grid. Internal prices are decided by all prosumers inside the cluster, and operate on the day-ahead and intraday market. Results show that enabling price-based energy-sharing inside prosumer clusters can achieve cost savings compared to independent, P2G trading.

2.6.4 Norwegian Wholesale market

The electricity wholesale market in Norway allows producer, consumer, supplier etc, to trade power with market participants over varying time frames (*Wholesale market*, n.d.). Participants choose what time frames they want to trade in, from hourly and day-ahead markets to long term financial markets. Prosumer agents who want to implement DR solutions could participate in LB, ID or DA markets, depending on their production/storage infrastructure 2.13.



Day-Ahead Market
(12hours-36hours)

Trading stands for 90% of **Nordic power** consumed. Hourly prices are calculated for the following day. Producers and consumers bid on power delivery contracts for the following day. The market also takes bidding zones into consideration

Forward Finanical Market
(long & short term)

Long term energy trades regulated by the **Financial Supervisory Authority of Norway** . Participants in the physical power market benefit from its well-being, as they can reduce the risk by trading long time and collecting future incomes.

Intraday Market
(1hour-1day)

Traded with **Nord Pool or EPEX SPOT**, this market offers continuous trading up until an hour before production. Imbalances due to weather or sudden changes can therefore be considered.

Load Balancing Market
(15min-45min)

Operated by **Statnett.** Market contains primary, secondary and tertiary reserves. Bids close 45 minutes before produciton time, however tertiary reserves can respond within 15 minutes of produciton hour.

FIGURE 2.13: Nordic wholesale markets

The **Norwegian prosumer market** allows customers a maximum of 100 kW surplus power production at any time, without paying construction contribution. Prosumers require contracts with local grid companies regarding grid rent. Additional requirements may come due to legislation's. Norwegian grid tariffs might see a change with the introduction of capacity-based grid tariff(Askeland et al., 2018). The energy tariff implemented today consists of a fixed part and an energy part. Capacity based grid tariffs include a capacity part, which is a proposed solution where customers pay a fixed capacity subscription and only extra if limit is exceeded.

Chapter 3

Methodology

As previously stated, realizing a HEMS requires meters, communication, and management. The diversity and complexity of HEMS can be very high, which makes it hard to define a standard for them. One of the crucial components in such systems is the robust communication system, which needs to be selected based on latency, reliability, and life expectancy. Communication is not limited to the grid but includes other HEMS systems. Realizing seamless communication between power distributors, prosumers and consumers contributes to optimizing grid conditions. Selecting the right communication system for user applications is therefore highly important. Furthermore, we need a management system that processes the information conveyed by the communication system. Cloud services help with managing the communication system and handle the data transmissions. Designing a communication system for HEMS depends on the intended cloud (if any) deployment and how the utility can access it. The most simple solution would be to use only Wi-Fi components to transmit data between utility and home meters/appliances. As we will see later, this is not the only solution.

3.1 Overview of Cloud-based LoRaWAN Communication Infrastructure

This thesis proposes a design and implementation of a LoRaWAN communication infrastructure for cloud-based prosumer clusters. It is a small community of prosumer agents with semi-proximity, connected to a common LoRa gateway 3.1. Each cluster represents a neighborhood or group of residential buildings. A test bed solution shall be constructed and used to evaluate LoRaWAN capabilities in clustered HEMS. Houses/buildings act as Prosumer agents by using LoRaWAN energy meters to share time-series of consumption and production. The gateway relays the time series to a network server, which sends the data to an application server using MQTT. Here, all of the data is processed and stored for DR and DSM purposes. The application server is functioning as an operating center, where information is delegated and acted upon. Utility agents and prosumer agents have different roles in the operating center, which means they have restrictions on what they can view/manipulate. Access level is determined before the prosumer/utility relationship begins, and is drawn in a contract before implementation. Some users might restrict details about certain appliances etc., so utility won't have access to that information.

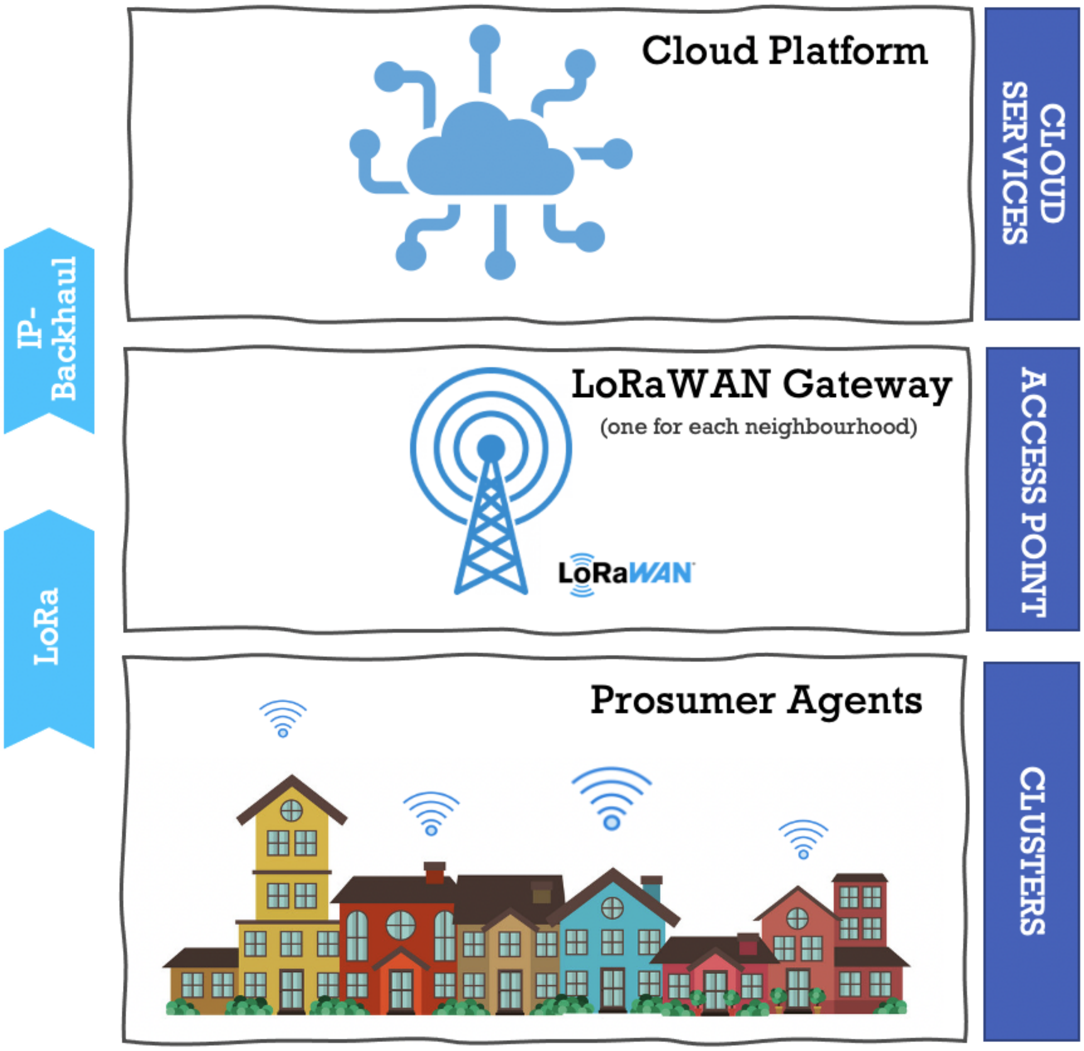


FIGURE 3.1: Prosumer cluster architecture

This paper uses the basics of prosumers clusters to investigate communication and management solutions, as well as some use-cases for systems already implemented. Various communication schemes with multiple end-node configurations and gateway scenarios are evaluated based on coverage and latency. Note that there are many more possibilities for expanding upon the communication network. Other protocols like Z-Wave and ZigBee are favorable in IoT solutions at home; however, they are not covered more extensively here. Only LoRaWAN and TCP/IP will be discussed and used in the HEMS communication system.

3.1.1 Communication Architecture

The system architecture for cloud-based prosumer clusters splits up into three responsibility areas. The first one is prosumer side components, which include LoRa nodes and a connection to the cloud platform. This covers the parts operated and managed by the prosumer. Users access the cloud platform through a smartphone, or any device supporting web-browsers. The next area is the utility agent with associated gateways. Utility access the cloud platform to push information to residents. They are also responsible for installing the allocated gateways. If one gateway is already covering a prosumer cluster, then the utility can use it as long as the capacity is sufficient and the owner allows it. In that case, the utility won't actually manage the gateway, only utilize it for increased coverage. Finally, we have the cloud platform, which is the connection hub between utility and prosumer. As mentioned in previous sections, the cloud deployment can be private or public depending on the policies of the utility.

3.2 display the complete architecture of the system. Later sections go into further details about the design and functionality of each component; however, utility agent will not be covered more extensively.

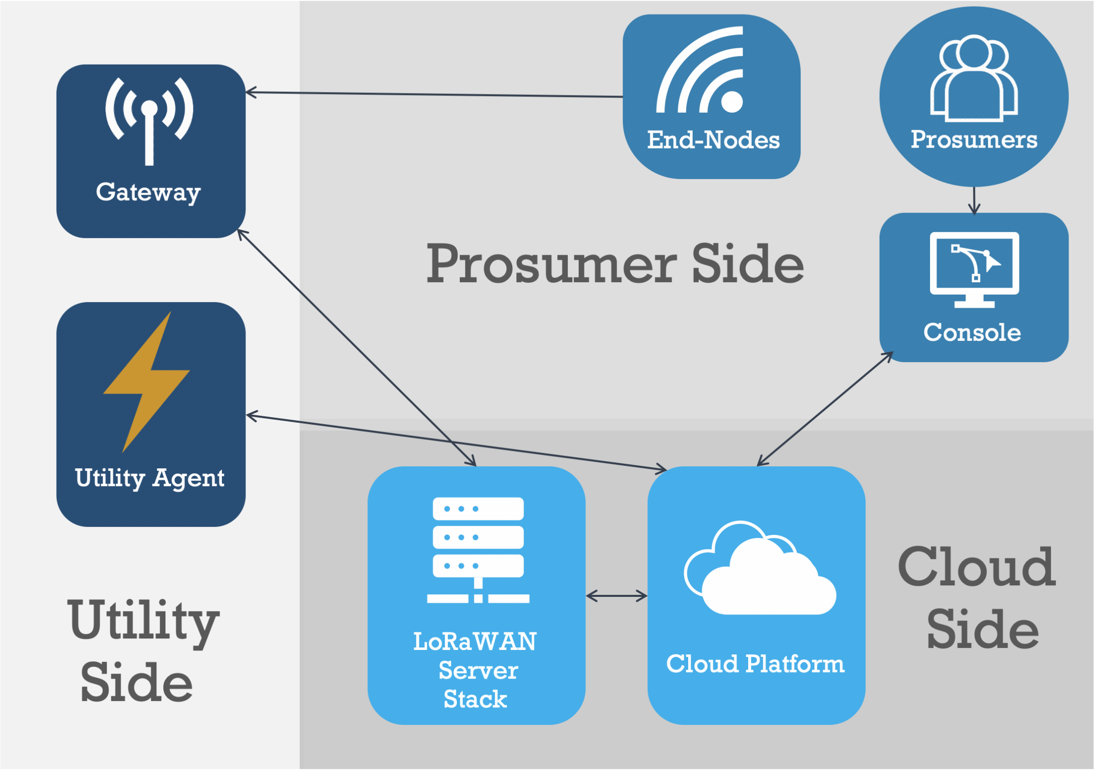


FIGURE 3.2: System architecture of cloud-based prosumer clusters

In cases where utility won't use public network servers, they have the option to install a network/application server compatible gateway on-premise. This solution allows the utility to operate the servers on-site, which can remove the need for cloud providers. Operating a private network server on the utility side does not directly imply that cloud services are redundant since the network server can easily be integrated with a cloud platform. Multitech LoRa gateways have the ability to run a Linux environment so the utility can deploy applications locally. Computing data and storing it on edge makes it less vulnerable to data leaks and server downtime. The issue with on-edge computing and data storage is that prosumers could have more restricted access to the data. Although this paper presents a cloud platform solution, it is important to know all the deployment options before implementing the system. Edge deployment can in some cases be more beneficial, depending on costs and application.

3.1.2 Prosumer Agent Operating Modes

The bi-directional communication of LoRa nodes enables utility agents to not only receive time series from prosumer agents but also transmit control signals to residents. To simulate the bi-directional communication between the utility and household users, two situations are analyzed:

- **24-hour time series** uplink of on-site generation and consumption from prosumers to utility. Households with internet-connected appliances can also transmit individual appliance consumption or device state, which provides the utility with more detailed information. Battery information like state of charge is also delivered if installed.
- **DSM control** by utility agent based on power grid conditions. LoRa nodes receive downlinks with updated load profiles. This case allows utility to dynamically shift load profiles by performing Active DSM (ADSM). The signal can also be used to updated device shadows in the IoT Core.

3.1.3 LoRa Gateway Options

Clusters of prosumers can vary in size and range. The distance between prosumers located in rural areas can get quite large, which makes it important for installers to evaluate gateway coverage. Gateway location and selection are important to ensure reliability and effectiveness of deployment. This paper proposes two gateway scenarios for clustering prosumer agents:

1. Multiple gateways forming clusters of prosumer agents. They are installed in specific locations to provide coverage for a village, neighborhoods, or other semi-large areas. Installers should investigate the number of prosumers connecting to each gateway to make sure the capacity is sufficient

2. Centralized utility agent gateway covering a whole city or large chunks of land. This scenario requires powerful gateways with a capacity to handle large clusters of prosumer agents. The utility can choose to install the gateway at their side; however, this might be sub-optimal in terms of optimal location.

A third option exists, which is a hybrid of the two. Large gateways can sometimes have blind spots where signals won't reach with sufficient power. This could be nodes located deep inside a building or behind a mountain/hill. Installing lower capacity gateways at these blind spots ensure total coverage.

3.1.4 Communication Scenarios

Prosumer agents can communicate with the utility using numerous communication protocols. Smart meters and appliances might be connected to a smart hub, which in turn is connected to the internet. In that situation, the utility can receive and send messages using the smart hub as a link. Security cameras, f.ex, often need heavyweight communication protocols like WiFi in order to transmit real-time pictures. Whatever the use-case is, installers have to evaluate the most suited protocol for each device. This paper considers two entities communicating: a prosumer agent representing all of the connected devices in a home, and a utility agent responsible for analyzing prosumer data and issuing DR commands. The communication between agents is evaluated for both LoRaWAN and MQTT over TCP/IP.

3.1.4.1 LAN: Ethernet

The first communication scenario utilize a LAN, combined with an Ethernet cable connected to the end-node, to upload and download messages from AWS cloud platform 3.3. With this solution, prosumers can facilitate the communication between utility. with home routers, which many homes already have installed. This solution would also be possible with a cellular connection to the internet. Prosumer agents, in this case, the Raspberry Pi-based (RPi) smart hub, communicate with the cloud provider using MQTT commands. Publishing messages with HTTPS is also possible; however, MQTT was selected for this scenario. AWS IoT Core message broker is a server that handles all of the incoming messages and directs them to the correct destination. The communication between a prosumer agent and an MQTT broker is facilitated by a Software Development Kit (SDK), which is installed on the end device. SDK's help developers build messaging applications for devices connected to AWS IoT Core.

Figure 3.3: Prosumer agent connect to utility agent via cloud platform by using MQTT over TCP/IP

One of the main features of this scenario is the simple setup and management of device communication. Most users are familiar with establishing an internet connection through Wi-Fi, and the SDKs can easily be downloaded from AWS's website. It will require some know-how when it comes to building the messaging application since the SDKs are language-specific. Furthermore, it might be a better solution for users located in areas where access to a LoRaWAN gateway is non-existent or the geographical topography hinders cost-effective use of LoRaWAN. As long as the prosumer agent is within Wi-Fi (or Ethernet) range, then outside topography won't affect the connection since it goes through a cable-connected modem.

3.1.4.2 LPWAN: LoRaWAN

Instead of connecting directly to the cloud platform through Wi-Fi or Ethernet, this solution utilize LoRaWAN nodes and gateways to connect prosumer agents to the internet. In this scenario, prosumer agents use a LoRa module to enable communication with a gateway. The gateway receives signals from prosumer agents, then backhaul the data onto the internet. Nodes do not require an internet connection via Wi-Fi or Ethernet, since they use LoRaWAN to communicate. Since the gateways job is to serve as the link between nodes and the internet, it needs to support both Wi-Fi or Ethernet, and LoRa. It is possible to use cellular networks as a backhaul to the internet; however, not all gateways support this solution.

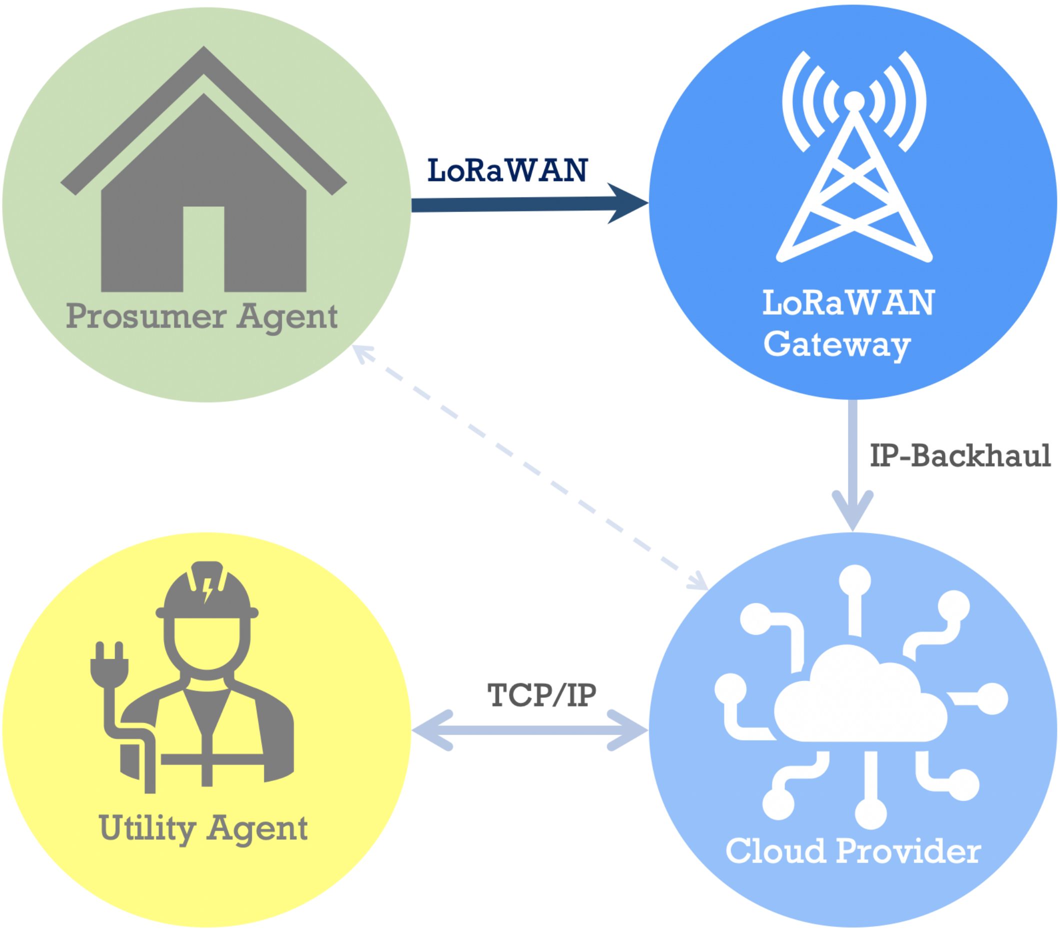


FIGURE 3.4: Prosumer agent connect to utility agent via cloud platform using LoRaWAN and an IP-backhaul

At first, this scenario might seem redundant or unnecessarily complicated compared to the first one. However, it is to be shown that the inclusion of LoRa technology in communication infrastructures allows remote internet access to battery-powered devices. In situations where prosumers have installed a renewable energy system (PV-panels with storage) outside the Wi-Fi coverage of a home, LoRa can be deployed to extend the internet connection to smart meters and battery-bank controllers.

3.2 Prosumer Node Implementation

This section contains the implementation of hardware used for realizing LoRa-based prosumer nodes. It presents a deployment solution for 24-hour time series and DSM control in gateway scenarios 1 & 2 (presented in chapter 3.1.3).

A LoRa node corresponds to an MC powered with a LoRa transceiver. There are two types of nodes used in a HEMS: smart meter nodes (smart energy meter) and appliance nodes. The meter node is simulated using an MC with pre-recorded energy production and consumption time series. Every hour, the MC transmits time-series data to nearby LoRa gateways. Appliance nodes are responsible for controlling residential appliances. These nodes are simulated using LEDs to demonstrate active DSM. One LED is controlled with a PWM signal to simulate lighting inside a prosumer home. Each node consists of one RPi (Raspberry Pi MC) and a LoRa transceiver module 3.5. The implementation in this paper combines the meter and appliance node in a single LoRaWAN node, however this was only during downlink testings

Figure 3.5: Raspberry Pi based prosumer node

3.2.1 Microcontroller

Raspberry Pi was selected as MC for the LoRa nodes. It supports micro SD cards to provide storage and an operating system. 40-pin GPIO access makes it possible to connect several modules in order to expand the range of communication protocols. Keeping track of the GPIO pins is important when using several pins at once. Many Raspberry Pis supported modules are sold as a ”hat”, placed over the GPIO pins. Often, as is the case in this paper, some of the GPIO pins are not used, since the module only uses a couple of pins. Although the Pi is connected to the internet via Wi-Fi, it is not necessary to set up this when only using LoRaWAN. Connecting it however makes it easier to troubleshoot errors by accessing the terminal using Secure Shell (SSH). The SD card is flashed with Linux OS and additional software for the LoRa module is installed.

3.2.1.1 Remote Connection

To manage the RPi, a remote desktop connection through **VNC Viewer** was deployed. RPi is commonly used as a regular computer, connected to a screen with HDMI and controlled with a mouse/keyboard. Quite often the RPi is not located in nearness to a screen, or it is situated somewhere inaccessible, so we need another way to view and control it. VNC (Virtual Network Computing) is a protocol that allows one computer to access another computer's graphical interface using a keyboard/mouse. This solution allows for a more user-friendly experience when managing the node. SSH connections were also utilized when only the command line was needed. The protocol employs decryption to secure a connection running on an unsecured network.

3.2.1.2 Power Management

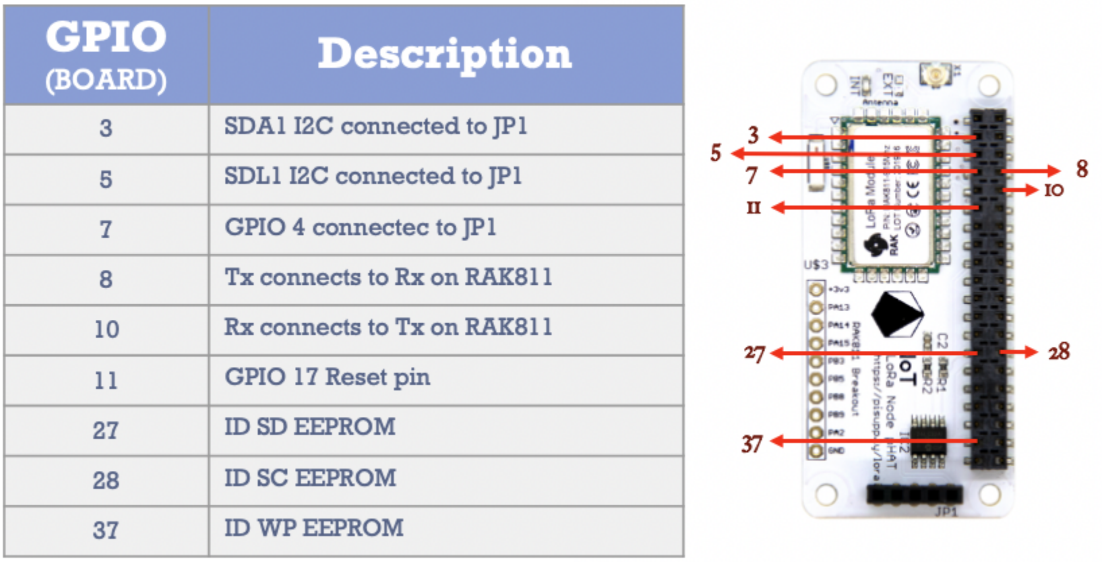
It is often useful to have battery-powered nodes, as they can be placed almost anywhere. RPi's require 5.5V through a micro-USB or USB-C connector. Bare-board active current consumption ranges from 100-600mA, with model B+ drawing 500mA (*Power Supply*, n.d.). The battery lifetime of a 10 000mAh battery pack powering a model B+ amounts to around 58 days, and that is without the transceiver module connected. Still, there are multiple ways of reducing the power consumption of RPi's, including turning off HDMI and USB connectors and disabling Bluetooth and Wi-Fi. This paper uses a wall outlet for powering the node when inside, and a battery pack for outside experiments. The RPi is mostly used for simulation and testing purposes, as it is not suitable for battery-powered applications.

3.2.2 LoRa Transceiver

A transceiver is a chip that enables the transmitting and receiving of signals through a medium. LoRa transceivers operate with radio signals where frequencies range from 100 to 1100 MHz. The intellectual property of LoRa belongs to Semtech, however other companies have received licenses to use it for manufacturing. Various LoRa chips exist and their usage is mostly restricted to either gateways or end-nodes. This setup includes the SX1276 LoRa chip from Semtech. It has a max link budget of 168 dBm and a max RX sensitivity of -148 dBm. Furthermore, the chip contains a temperature sensor and a battery indicator.

3.2.2.1 RAK811 Module

Powering the LoRa communication in each node is the RAK811. It is a transceiver module integrating both the SX1276 chip and the STM32. The latter is a small 16kb RAM, 32-bit MC integrated circuit. One of the key benefits of RAK811 is the low power consumption in both active and sleep mode. A PCB board is used to connect the RPi and the module. Pi-supply provides pre-built LoRa hats for RPi's, including one with RAK811. Their documentation gives a detailed description of GPIO pin usage. The communication between RPi and module happens over UART with 3 GPIO pins. Additionally, 8 GPIO pins are available for control via UART. These are breakout points located down to the left in 3.6



GPIO (BOARD)	Description
3	SDA1 I2C connected to JP1
5	SDL1 I2C connected to JP1
7	GPIO 4 connectec to JP1
8	Tx connects to Rx on RAK811
10	Rx connects to Tx on RAK811
11	GPIO 17 Reset pin
27	ID SD EEPROM
28	ID SC EEPROM
37	ID WP EEPROM

Figure 3.6: GPIO connections on Raspberry Pi LoRa Hat

The RAK811v3 python library includes access to AT Commands performed by the module. Early 21 editions of the RPi hat come with updated firmware which as of mid-April is supported in rak811_v3. The python library contains all necessary commands for managing RAK811. Before the node can join and send a message to the network server, it needs to configure the operating region, app eui, and app key. After joining the network, further configurations like setting data rate follows. The library is designed for users who want to control the SX1276 by using the python language. It allows users to issue AT commands directly from a python script programmed on the RPi

The antenna is connected to a u.FL connector on top of the LoRa hat. By default, the LoRa node is configured to use an internal antenna. Next to the RAK811 module is two inductor pads, indicating what antenna to be used. The INT pad is soldered by default, so INT pads were de-soldered and EXT pads were soldered. Included with the LoRa hat is a small 868MHz antenna. One thing to keep in mind when transmitting with antennas is the near-field region. Electromagnetic fields produced by gateways and nodes can result in unexpected behavior when situated too close to one another. No further investigation on internal antenna use was conducted.

3.2.3 LED's for Simulation of Home Appliances

Three LEDs are connected to the RPi for simulating home appliances 3.7 3.8. The LEDs represent the lighting, the washing machine, and the ventilation. The on and off state of each energy-consuming group is simulated with the light on/off. Each LED is connected to a breadboard, which in turn is controlled by the GPIO pins on RPi. Figure 3.6 displays the GPIO pins occupied by the transceiver, which makes it easy to identify which pin is available for use. 330-ohm resistors are connected in series with each LED to limit the current flowing in the circuit. Since RPi GPIO's only supply a small current, connecting LEDs without a resistor could potentially damage it as LEDs want to draw more current than what is supplied.

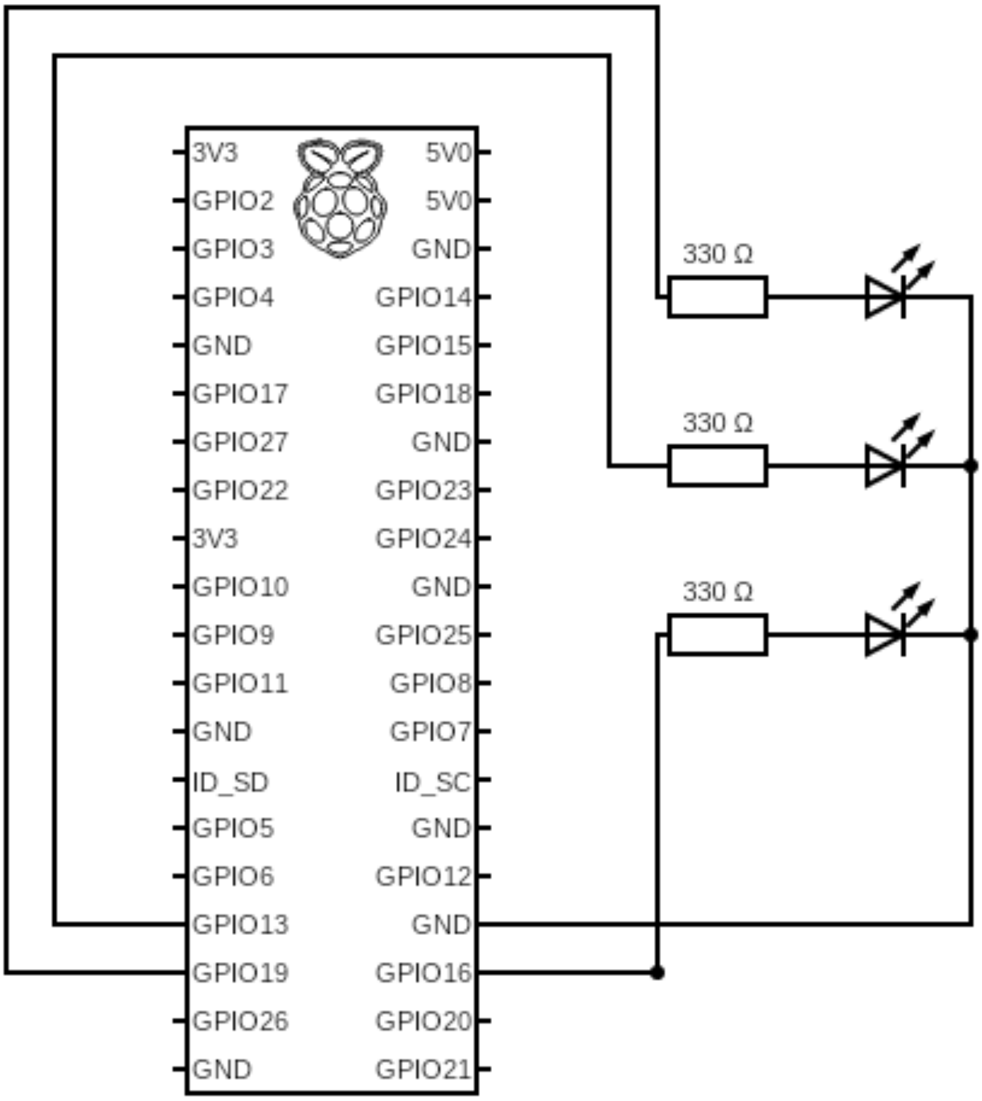


Figure 3.7: Circuit diagram for connected LEDs

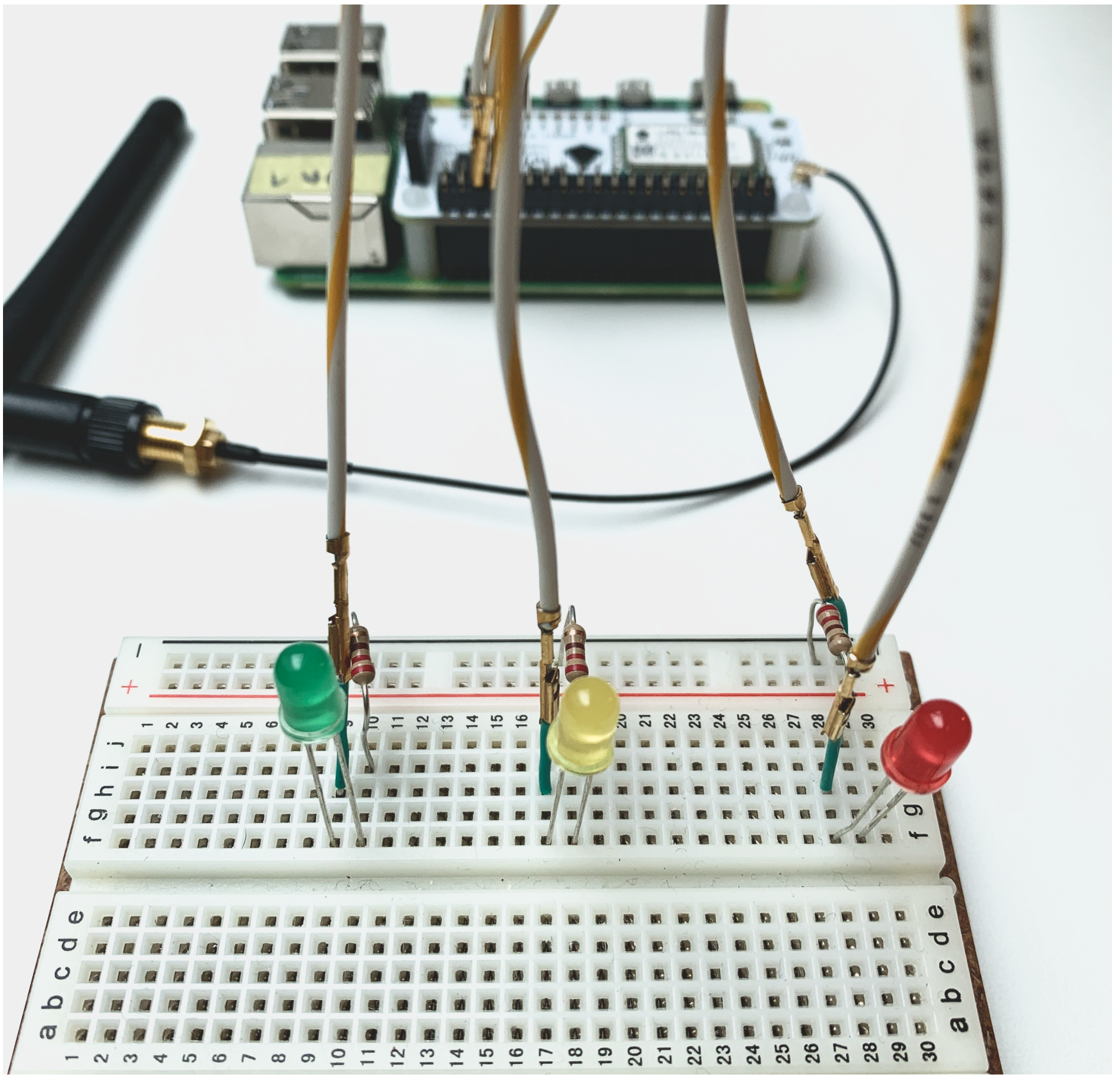


FIGURE 3.8: External LED circuit connected to GPIO's on the RPi

3.3 Utility Gateway Implementation

Looking back on the gateway scenarios stated in 4.1.1, we have two situations. The first one requires smaller gateways to cover a broad area, while the second includes a single more powerful gateway. The author combines the two in a hybrid model, where the large gateway is used as the main connection point, while the smaller gateways is supposed to cover up blind spots. Gateway installation is heavily location-dependent, which makes each deployment different.

3.3.1 RAK2247 RPi LoRaWAN Gateway

Building a DIY LoRa gateway is now easier than ever, thanks to a large amount of MCs and LoRa modules available. Prosumers can easily build a home gateway with the necessary coverage available. This option is great for homes or neighborhoods barely outside the coverage area, where some nodes do not reach the gateway with sufficient signal strength.

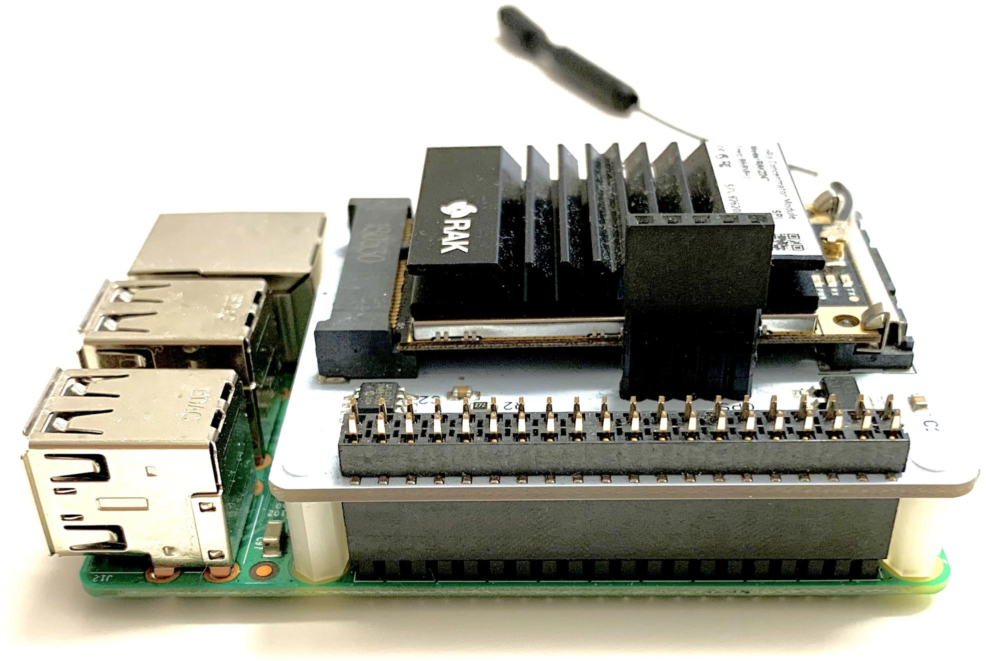


Figure 3.9: RAK2247 LoRa Gateway HAT for Raspberry Pi

This paper deploys an RPi-based LoRaWAN gateway HAT that utilizes the RAK2247 LoRa gateway concentrator module 3.9. It listens on 8 channels at the same time; however, it can only transmit one downlink at a time. The module is based on SX1301, which is a digital signal processing engine designed by Semtech and used in LoRa gateways. Many pre-built gateways for MCs use the RAK2247, including the IoT Gateway HAT for Raspberry Pi used in this paper. The RAK2247 module comes fully soldered and assembled on a PCB so users won't need to perform any soldering. Users only need to connect the HAT to an RPi with 40 GPIO pins, and then connect an antenna to the HAT. It also comes with a header for GPS connectivity; however, this is not used.

The installation process of the RAK2247 gateway starts with flashing an SD card image with pre-installed software. Realizing the IP-backhaul is done through either an Ethernet cable or a Wi-Fi connection. An Ethernet cable was connected to the RPi, since this provides the most stable internet access. Once the RPi is up and running with an internet connection, it is then registered on a public LoRaWAN network server. The registration process requires a gateway EUI, which is drawn from the MAC address of the RPi. A gateway server address is also required; however, this is already provided by the network server. It is possible to use a local gateway server (depending on the gateway properties); however, that is not implemented in this thesis.

Gateway configurations are performed using the a web-based GUI. The GUI is designed to manage the packet forwarder, which is responsible for interactions with the LoRa chip. It is capable of rebooting the RPi, restarting the packet forwarder, and configuring the local_conf.json file. The local_conf.json file provides information about the network server address, gateway ID, and port number. Due to the recent changes in the public network server (migrating from V2 to V3), the TTN mode on the gateway HAT is not functioning. Therefore, the Semtech/Legacy mode was selected, which allows for connections with any LoRa Network provider by inserting IP/Domain name and ports.

3.3.2 Laird Connectivity RG186 LoRaWAN Gateway

The second gateway option is a pre-built gateway from Laird Connectivity. The Sentrius RG1xx series for LoRaWAN-enabled gateways allows users to fully manage a private LoRaWAN network. This is a highly scalable solution with both dual-band Wi-Fi and wired Ethernet options. It uses the SX1301/SX1257 chipset from Semtech, which allows for transmission ranges above 15 kilometers. Also, this gateway supports 8 channels, with a transmission power of +27 dBm. One of the advantages of RG1xx series gateways is their Bluetooth and Wi-Fi connectivity options. Prosumer deploying the gateway at home can also use it for connecting Bluetooth devices, making it a viable option as a home smart hub. Like the RPi based gateway mentioned above, this one also comes with a gateway web interface for quick and simple configurations.

Users log in to the gateway web interface by first identifying the Ethernet MAC address. The last three bytes of the MAC address in combination with rg1xx[MAC address].local allows users access to the gateway. Once users log in to the web interface, they can configure the LoRa setup by either typing information manually, or using a preset. The gateway EUI is a unique value used to identify the gateway, and this value needs to be typed into the interface (it is also required later in the network configuration process). The interface also allows users to select what packet forwarder the gateway is going to use. A Packet Forwarder is essentially a program that connects the packets received/transmitted from the LoRa chip with a LoRa network server. For this thesis, the gateway is going to use the Semtech UDP Packet forwarder.

3.4 Application/Network Server: The Things Stack Community Edition

With edge components installed, it is time to implement the cloud section of the system. This section details the implementation of a LoRaWAN network server, managed by The Things Industries. The Things Community Stack (TTCS) is an open-source LoRaWAN network server deployed by The Things Industries. An enterprise version of TTCS called The Things Stack (TTS) exists, however TTCS was used in this thesis. TTCS integrates with AWS IoT, which means messages are routed from the TTCS network server to the AWS IoT application server.

One of the most popular solutions for a LoRaWAN network server is The Things Network. Like many other cloud platforms, TTN can be accessed through a Command-Line Interface (CLI) or via a web-console. In the console window, users can register gateways and nodes, make applications and monitor traffic on the network. TTN runs the open, decentralized LoRaWAN network TTS Community Edition (aka TTCS). In 2021 TTN changed from server stack V2 to TTS Community Edition (V3), which prompted users to migrate their devices onto the new stack. This process was somewhat problematic, as the procedures and requirements for adding applications, nodes, and gateways are slightly different in TTCS. Due to complications regarding the migration, the RPi gateway is not able to successfully merge with the V3 stack. The network server deployment used in this thesis is therefore planned to be a combination of V2 and V3, however it is later shown that this was not achievable. The resulting solution will therefore only use the V3 stack. New users and/or deployments should operate on TTCS.

The following setup presented below is performed in the web console; however, it can also be achieved through the TTCS CLI. An in-depth explanation of TTN gateway and node registration will not be covered, as the TTN cluster shuts down at the end of 2021.

3.4.1 Application & End Node Registration

The application section lets users create applications where end nodes are registered. TTCS offers the possibility to register nodes by either manually typing device information, or selecting a device preset from the LoRaWAN Device Repository. Each node is assigned a network server address, an application server address, and a join server address. Live data and location (if the node has a GPS module connected) can be viewed in the console; however, the data is not saved as a user exits the page. Therefore it is crucial to integrate a storage solution to keep track of end node data.

3.4.1.1 Payload Format Function

The payload format function is a programming function used to decode the uplink payload coming from each end node. Incoming payloads can therefore be simplified before transmission, since the function knows the structure of incoming signals. Encrypted signals coming from the end nodes are decrypted by TTCS, and the decrypted binary data is represented as hexadecimal (two symbols are one byte). These bytes can represent strings, numbers, or any other encoding solution. This is an important fact to consider as users want to minimize the payload size as much as possible. A common mistake to avoid is an unnecessary use of plain ASCII text, as one text character corresponds to one byte.

The payload transmitted from prosumer nodes consists of consumption and production data. It has a fixed length of 4 bytes where the first two bytes represent hourly prosumer demand and the last two represent hourly prosumer production. Before passing the data along to the application server, it needs to be organized. Listing 3.1 shows the JavaScript payload decoder function implemented in the TTCS console.

```
function Decoder(bytes, port) {
    var demand = bytes[0] | bytes[1]
    var production = bytes[2] | bytes[3]

    return {
        //results given in kW
        demand: demand/1000
        production: production/1000
    }
}
```

Listing 3.1: Uplink payload formatter

3.4.2 Gateway Registration

The gateway registration is a fairly simple process in TTCS. However, due to the recent upgrades from TTN to TTCS, the small RPi based gateway had to be merged with TTCS. This gateway is marked as another cluster, meaning it still operates on the old Things Network. Data received by the smaller gateway is routed from the old network server to the new one linked with TTCS. When registering a gateway to The Things Network (the old network server), users didn't need to provide a gateway server address. This is because the architecture of TTCS is a bit different compared to TTN. TTCS employs a gateway server that maintains connections with gateways. Its main job is to forward uplinks to the network server and schedule downlink traffic.

Although the RPi-based gateway was functional during the first months of 2021, it was unable to operate consistently on the new server stack after moving from V2. It was still fully operational on TTN; however, data was not successfully routed to the new stack. For this reason, the RPi gateway was not used for experiments. The issues are most likely resolved with an updated RPi image programmed to support TTCS.

3.4.3 TTN Fair Use Policy

The Fair Use Policy is uplink and downlink limitations for devices operating on TTN or TTCS. It is similar to the LoRaWAN duty cycle, however it is much more restricted. End-nodes are only allowed 30 seconds of ToA during one day. The total amount of allowed downlinks is restricted to only 10 during 24h hours, including ACK downlinks to confirm uplink.

3.5 Cloud Server: Amazon Web Services

Although TTCS employs an application server for handling uplink data decryption and decoding, this paper proposes an AWS cloud platform integrated with TTCS to serve as the operating center of the system. The reason for using an external application server comes from the large amount of IoT and computing services offered by AWS.

The AWS account used in this paper is restricted to free tier operations, meaning the services offered are limited compared to what AWS is offering. The IoT service is free of charge, as long as the monthly message count doesn't exceed 250 000 messages. The storage solution with DynamoDB offers 25GB of storage to free tier users. Finally, AWS also offers 1 million free lambda requests per month. The amount mentioned above is more than enough for the system deployed in this thesis; however, larger deployments will likely exceed the free tier usage.

3.5.1 Integrating AWS IoT with The Things Network

The integration of The Things Community Stack with AWS IoT is covered before the AWS architecture, as it plays an important role in handling the uplink and downlink messages. The main component of the integration is the cross-account role for connecting TTCS with AWS IoT Core MQTT endpoint. With this integration it is possible to subscribe and publish, using the MQTT client, to various LoRaWAN end-devices topics. The integration process is facilitated by AWS Cloud Formation. This service deploys a specified cloud infrastructure, based on a prewritten text file. TTCS integration comes with a link between AWS IoT Things and registered TTCS nodes, meaning it is possible to manage end-devices directly from the IoT Core.

3.5.1.1 Messaging using MQTT

The main functionality of this integration is the messaging option. When end-devices uplink messages to TTCS, they are forwarded and published to AWS IoT Core MQTT. An IoT rule created during the integration is activated and triggers a lambda function, which processes the message. The message processing is for decrypting the incoming payload and update the things shadow state. If the incoming message is from an undiscovered device, lambda automatically registers the. thing in AWS. A reversed process is also implemented for downlinks. Messages published to the downlink topic activate the downlink rule, which in turn triggers a lambda function that passes the encrypted payload to TTCS.

3.5.2 AWS Cloud Architecture

Figure 3.10 displays the full cloud architecture with some of the TTCS integrations. The empty box represents the additional cloud section for utility applications. A scenario might occur where the utility would like to receive messages directly, instead of collecting them from a Dynamo DB table. In that case, proper changes to the cloud architecture need to be performed.

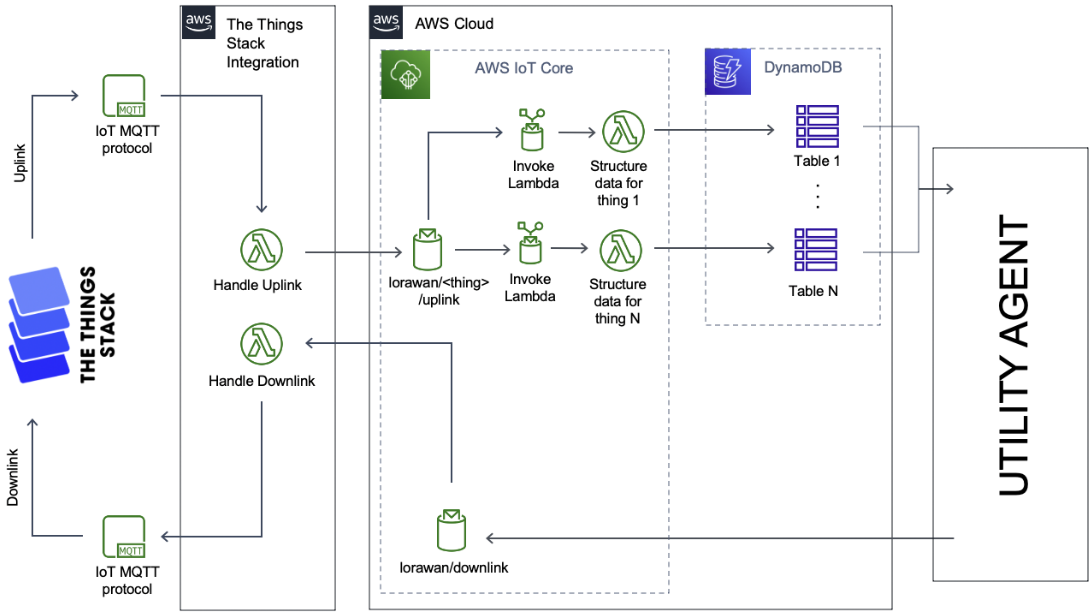


FIGURE 3.10: AWS cloud platform architecture with TTS integration

This architecture represents a basic storage and payload structuring solution. It is to serve as a foundation for the operating center in cloud-based prosumer clusters. The TTCS integration pictured on the left side is only applicable for applications using the integration. Incoming data handled by the TTCS integration is published to the IoT topic *lorawan/<thing>/uplink*. <*thing*> represents the end device ID, which is the name of the thing in AWS IoT Core. When the integration publishes a message to the respective thing topic, an IoT Core rule is invoked which sends the data to the lambda function. Here, the data is structured and passed to a DynamoDB table corresponding to that specific end device. Contributions from this paper include the AWS cloud section with data structuring and DynamoDB storage.

3.5.3 Database Storage: DynamoDB

DynamoDB is used to store the energy measurements transmitted by the prosumer nodes. Creating a storage solution for simple energy measurements can be a simple process; however, if the amount of nodes is in the thousands it can get a bit complex. The solution deployed in this thesis use one table for every prosumer node. Since database structure design is outside the scope of this thesis, no other solution will be evaluated extensively. With that being said, there are multiple, application-dependent solutions for storing the measurements. One possible solution is to create a single table for each gateway, as this can represent a geographical cluster of nodes.

3.5.3.1 Data Structure

Decrypted node measurements are collected from the relevant MQTT topic. From here on, they pass through another lambda function responsible for structuring the data before inserting it into the DynamoDB table. It is important to structure the incoming MQTT messages, as they include a some redundant information (frequency, modulation, repeated values, etc.). Depending on the application, some of this data (RSSI, SNR, data rate f.ex.) is still useful to store for evaluation of deployment solution. The data structure in this thesis follows the common JSON structure with three attributes: message payload, gateway metadata, and node metadata 3.11.

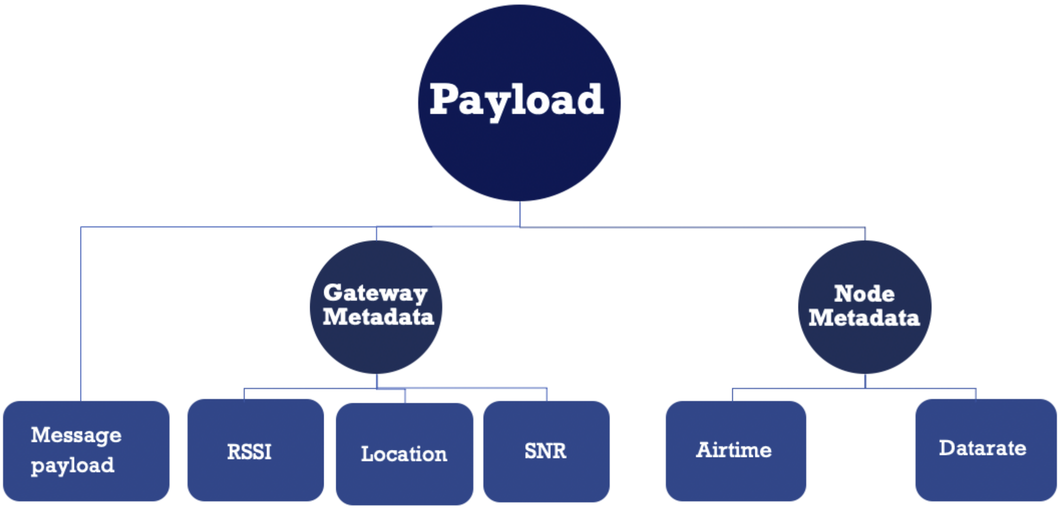


FIGURE 3.11: Data structure for uplink payloads stored in DynamoDB

Data in DynamoDB tables are stored in key/value pairs. Incoming data is stored as a hash table where keys are unrepeatable. The value connected to each key in this storage solution is a JSON object following the structure presented above 3.11. Selecting the proper key for a single table solution is more difficult compared to multi-table deployment. The timestamp value was selected to represent the key, since single nodes can not uplink multiple messages at the exact same time.

3.6 Evaluating the LoRaWAN Communication Infrastructure

The proposed LoRaWAN communication system is designed to evaluate the possibility of using LoRa technology in HEMS. The main question to answer is whether or not LoRaWAN is a suitable communication protocol for HEMS. In order to evaluate the communication network proposed in this thesis, we first need to determine the communication requirements in HEMS.

3.6.1 HEMS Communication Requirements

Communication requirements depends on the intended HEMS functionalities in the home. In its simplest form, HEMSs collect residential production and consumption measurements, analyze the data and use it to perform DSM. These measurements can be collected in multiple ways. First, it is the use of general smart meters, which measure electrical energy parameters while looking at the house as a single unit. Furthermore, it is possible to communicate with individual smart appliances using built-in transceivers. This solution often provides little to no freedom when it comes to selecting communication protocol, which in turn leads to interoperability problems. Lastly, it is possible to connect measuring/controlling devices in between appliances and power outlets (smart plugs). The third option use one LoRa node per appliance. To simplify the options presented above, this thesis only test uplink/downlink in a single node HEMS infrastructures (represents meter and appliances).

A HEMS requires a bi-directional communication network in order to both send and receive messages. Appliance management can be automated (utility performs active demands side management) or human-controlled based on signals coming from the utility. Residential buildings vary a lot in terms of appliance installations, existing communication infrastructure, and storage/production capabilities, which can pose different requirements for specific buildings. With that being said, it is still required to have low-latency operation and good coverage to ensure reliable communication. Evaluating the performance of the proposed communication infrastructure is based on the use-case of this system, which is the implementation of prosumer clusters. **Coverage** and **metering resolution/latency** are the two main focus points of this investigation.

In order to evaluate and test the performance metrics of a cloud-based LoRaWAN system, this paper deploys two main methods. The first one deploys a theoretical ToA calculator built as a python script. It is based on equations (2.10, 2.9) drawn from the background section. The second method is to deploy the test bed designed at the start of this chapter. Evaluation of stated performance metrics can not be isolated from each other, as they must be evaluated together. In the discussion section, these metrics are assessed and evaluated in a holistic view.

3.6.2 Gateway Location

Indoor transmission tests are performed at NTNU Gløshaugen campus, inside the Electrical Engineering building. Restricted access to the roof prohibited the installment of gateways outside, so the gateway was placed inside on the 4th floor 3.12. It is situated next to an indoor window facing towards the city. Gateway installment should preferably (if it is not meant to cover deep indoor areas) be installed high up on a roof, or at least with the antenna located far above the ground. One of the reasons for this is to combat the Fresnel zone phenomena, mentioned in section 2.4.4.2.

FIGURE 3.12: Indoor gateway located at NTNU Gløshaugen campus in the Electrical Engineering building, block E/F, at the fourth floor (floor plan taken from MazeMap)

The Laird RG186 gateway was utilized for this experiment, as it provides more coverage than the RPi based gateway.

3.6.3 Indoor Coverage Experiments

Evaluating the indoor coverage of LoRaWAN solutions is important to ensure reliable communication with end-nodes. This section details the indoor experiments performed with the designed test bed solution. Measurements collected from the experiments will be stored in AWS DynamoDB and later used in the result section to visualize the findings.

3.6.3.1 Testing Situations & Variables

The goal of the indoor experiments is to investigate the optimal data rates for end nodes situated at various distances from the gateway. Furthermore, a downlink latency test is also performed. The indoor coverage tests consist of two testing situations:

- Uplink from end-node with various data rates
- Round-trip latency test with optimal data rate found from the uplink test

RSSI and SNR are used as performance metrics for evaluating the LoRaWAN communication network. One of the main focus points when designing a LoRaWAN network is to not use redundant transmission power, in order to preserve battery life. The signal should therefore be strong enough to be heard, but not unnecessarily strong. In order to investigate the proper node configurations, this thesis proposes the following independent and dependent variables for the experiments 3.1 and 3.2

INDEPENDENT VARIABLES	
name	**unit**
Distance	meter
Data rate	index A.2
Wall density	index A.1

Table 3.1: Independent test variables

DEPENDENT VARIABLES	
name	**unit**
Signal-to-noise Ratio	dB
Received Signal Strength Indicator	dBm

Table 3.2: Dependent test variables

3.6.3.2 Procedure

For each indoor location, the same testing procedure follows. First, the node performs a join to TTS by issuing a join request and receiving a join accept downlink. After a successful join attempt, the node proceeds to select the highest data rate possible (index = 5) and transmits a message using this rate. The same procedure is repeated for all data rates down to index = 0. With each step, the corresponding SNR and RSSI values are stored in the DynamoDB table associated with that node. The full data rate test is then repeated five times to ensure correct signal measurements. An average value across the five measurements for each data rate at each location is calculated.

After the best data rate is found for each node location, a downlink latency test is then performed. This is to evaluate the possibility for active load control by the utility. The nodes used in this thesis only operate as class A devices, which means that a downlink can only be scheduled after an uplink. This additional latency is addressed later in the discussion section. The chosen method for measuring latency is therefore decided by measuring the round-trip latency, rather than only downlink. Total round-trip latency equals the time between issued AT+send command on the transceiver and LED changes on the breadboard.

The python script A.2 presented in the appendix section details the whole end-node transmission process from setup to join connection closed. The payload transmitted is based on synthetic, 24-hour prosumer time-series, which has to be prepared for transmission. There is also implemented a somewhat abnormal channel on/off function that handles the join issue present in RAK811 firmware 3.0.0.14.H, where the transceiver transmit join requests on other channels than those specified by the LoRaWAN specification document (*LoRaWAN 1.1 Specification*, 2017) (Trymjb, 2021).

3.6.3.3 Testing Locations

Figure 3.13 shows the four different end-node locations utilized to perform the coverage tests. Values for distance, wall density, and floor level corresponding to end-node location are provided in table 3.3. Distance values between end-nodes and the gateway were calculated using the following procedure. First, the horizontal distance is captured using MazeMap and the reference scale presented in figure 3.12. Then the vertical distances are calculated using 3.5 m per floor. The gateway and end nodes are all located at working height (0.85m) on each floor. Finally, the total distance is calculated using Pythagoras' Theorem.

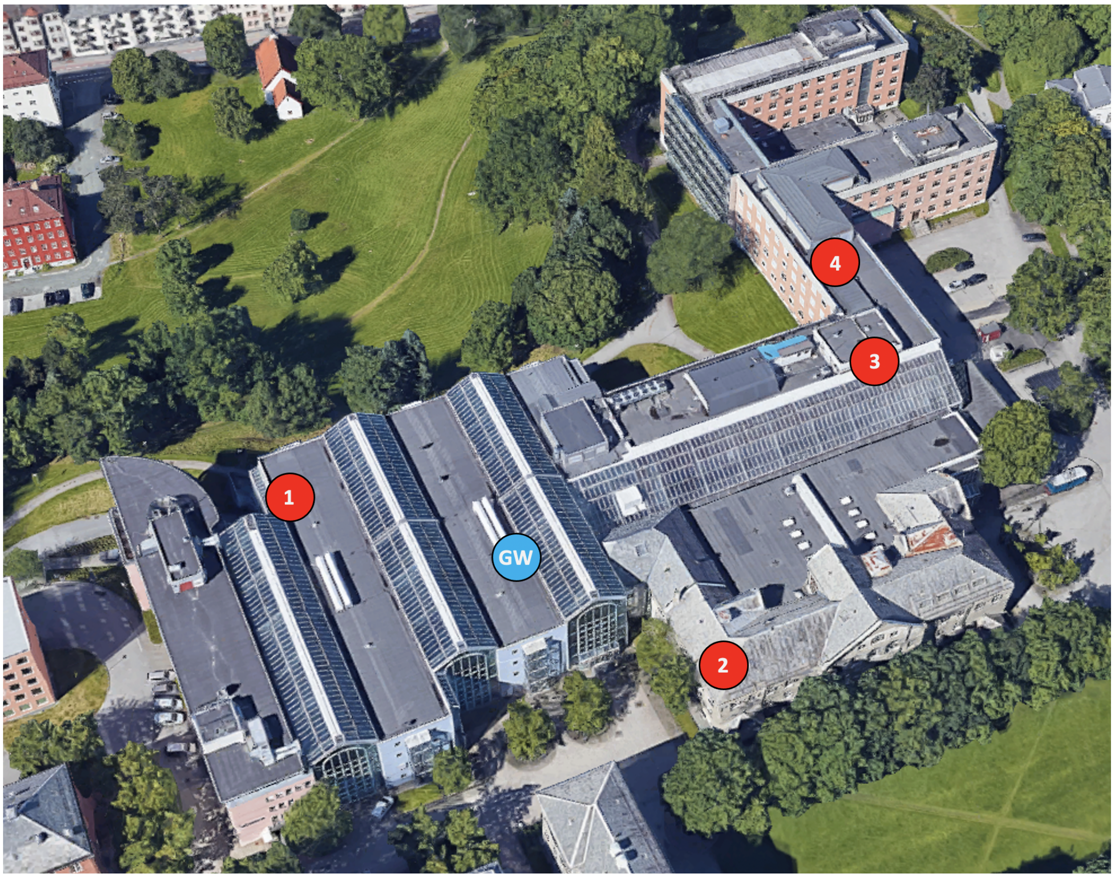


FIGURE 3.13: Locations of end-devices (red) and gateway (blue)

End node	Distance (m)	Wall thickness	Floor
1	43.7 m	MEDIUM/LARGE	4
2	42.1 m	MEDIUM	2
3	79.8 m	SMALL	1
4	99.0 m	LARGE	3

TABLE 3.3: End-node location parameters relative to gateway location

3.6.4 Outdoor Range Experiments

It is important to investigate the maximum signal transmission range achievable by end nodes. Such tests are vital in order to map the coverage zones provided by each gateway. The gateway used in this thesis is located inside, which makes for a good opportunity to investigate the maximum range provided by indoor gateways. In some cases, it is not feasible to install a gateway outside, which prompts installers to settle for an indoor location. This experiment evaluates the maximum range for such a gateway by transmitting signals from various distances to the gateway, and see which signals are received. Location number 1 and 2 has optimal line-of-sight (to the roof/wall of the building housing the gateway). Although the elevation profile for location 3 is promising, the optimal line-of-sight is broken by several buildings separating the node and the gateway. Also here the DRs are incremented to test long ranges with different SFs. Figure 3.14 display the testing locations and table 3.4 shows the distance from end-node locations to the gateway

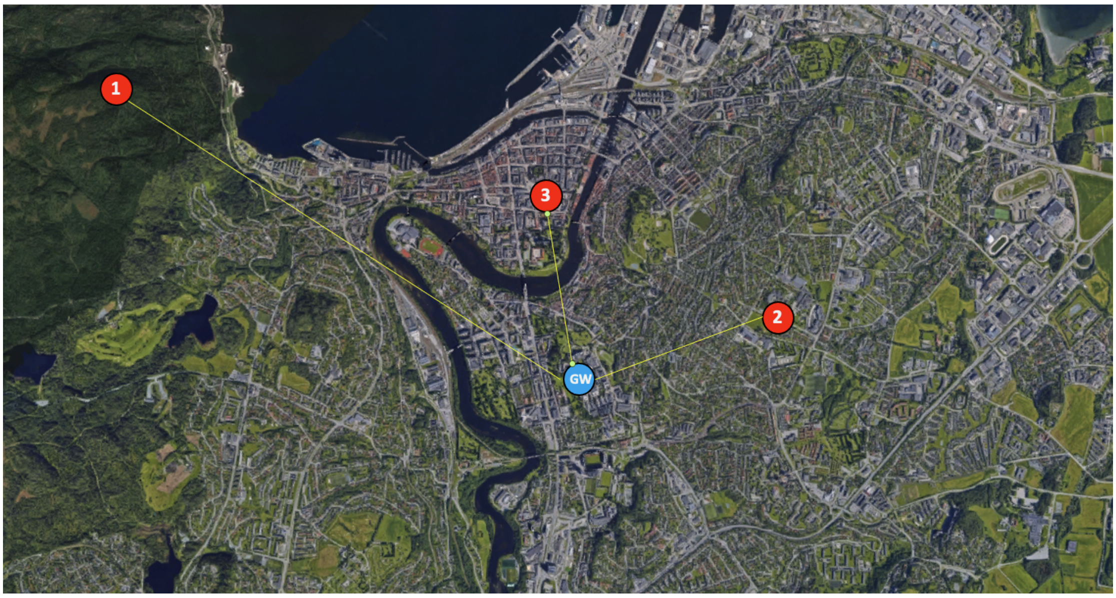


FIGURE 3.14: Outdoot test locations for end-nodes

End-node	Distance (km)
1	3.95
2	1.57
3	1.34

TABLE 3.4: Distance from end-nodes to gateway

An elevation mapper from Geocontext was used to map the elevation profile between the three end-node locations and the gateway (Pietruszka, 2010). It also displays the corresponding Fresnel zone for each scenario, indicating if the zone is clear or not. Figure 3.15, 3.16 and 3.17 displays the elevation profile for the three end-node locations relative to the gateway. The figures also display the height at which the nodes/gateway is located. The figures always show end-nodes to the left and gateways to the right.

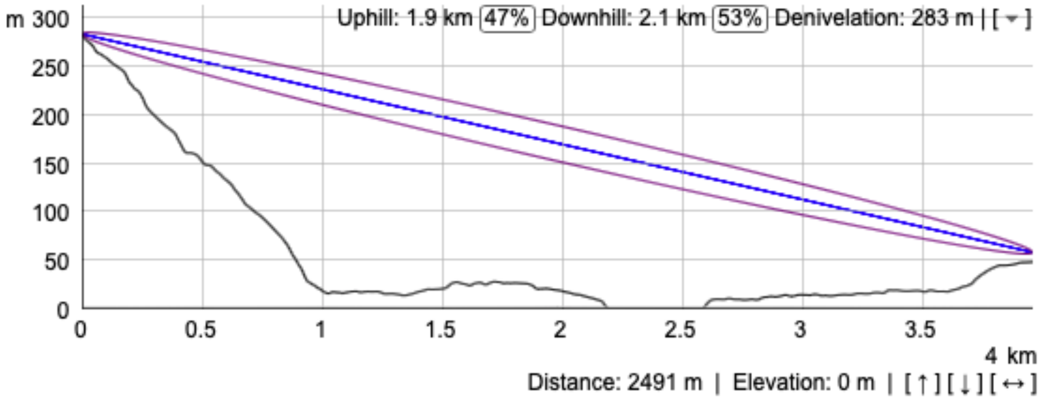


FIGURE 3.15: Elevation profile for Location 1

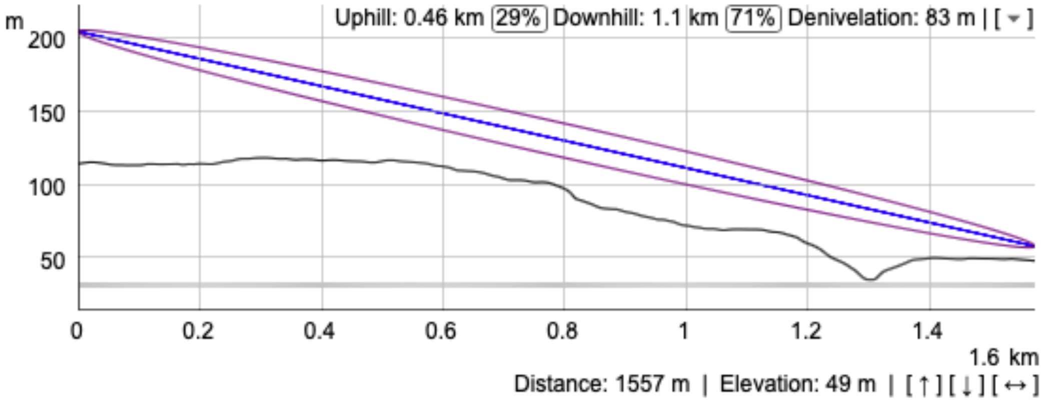


FIGURE 3.16: Elevation profile for Location 2

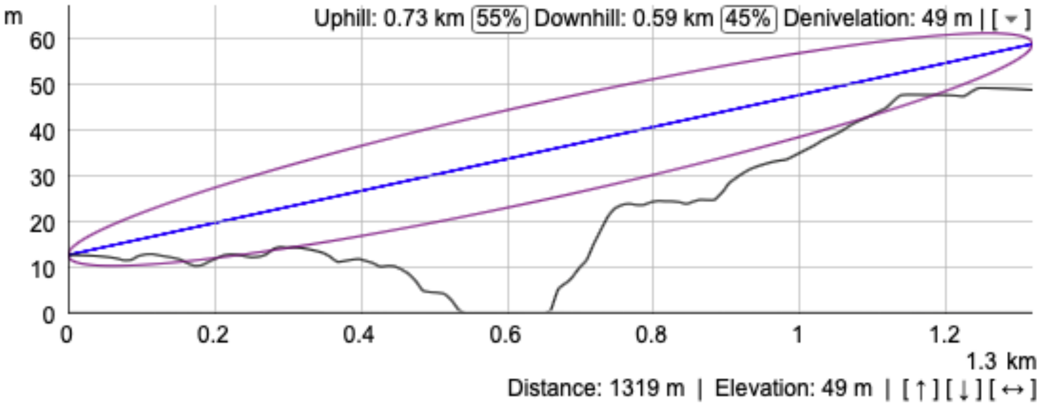


FIGURE 3.17: Elevation profile for Location 3

3.6.5 Metering Resolution

Before implementing a LoRaWAN smart metering and DSM system, it is necessary to evaluate latency and metering resolution in order to reveal the system's capabilities. LoRa signals vary quite much in terms of transmission time, ranging from tens of milliseconds to beyond a second. LoRaWAN certified end-nodes have to meet the national regulations, which dictates allowed transmission time during 24 hours (duty cycle). End-node configurations and payload size dictate the time-on-air, which means that it also dictates the number of uplinks allowed in one day. 2.10 and 2.9 is used to investigate the ToA for signals with varying DR. These equations combined with Europe's duty cycle regulations and the TTCS Fair Use Policy form the basis for evaluating end-node metering resolutions.

Although there are many options for choosing independent variables in the equations mentioned above, this investigation only evaluates the ToA based on data rate and payload size. Keep in mind that the data rate is still just an index describing different SF and BW combinations. All other variables are kept as default LoRaWAN values, with LowDataRateOptimize disabled for SF7-10. Finally, it is worth mentioning that small changes in payload size do not necessarily imply a change in ToA, as it occurs in block intervals for payload sizes. This is due to the ceiling function in the payload duration equation 2.10. The ceiling function comes from the fact that LoRa signals are transmitted in symbols, which have a fixed amount of chirps (2^{SF}). Even though the payload won't use the entire symbol, it still needs to be transmitted.

Chapter 4

Results

4.1 Indoor Coverage Experiments

This section covers the results from indoor experiments. Dependent variables detailed in 3.2 are investigated based on various combinations of the independent variables showed in 3.1. The most important findings from indoor measurements are showcased in this section.

4.1.1 RSSI Assessment

Fig 4.1 represents the average measured RSSI values for each indoor location plotted for DR0-5 (for SF to data rate conversion, see A.2). It is encouraged to look at these results while minding the wall thickness given for each case (see table 3.3).

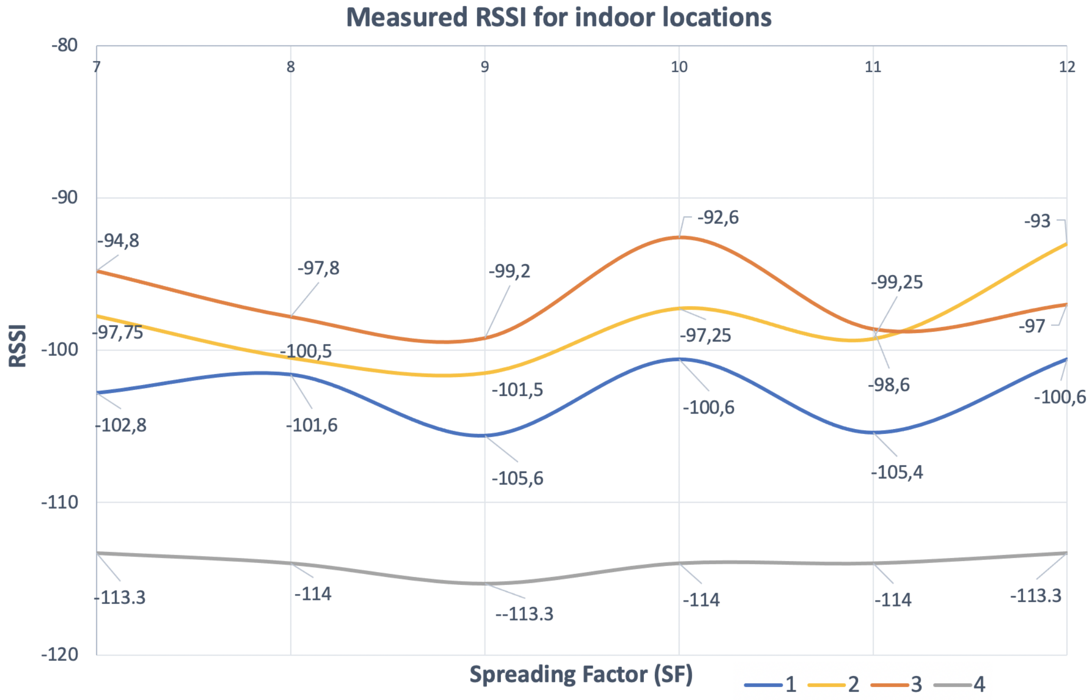


FIGURE 4.1: Average RSSI for each location plotted for each data rate

Furthermore, the same average measured RSSI values are plotted with the corresponding location distance from the gateway.

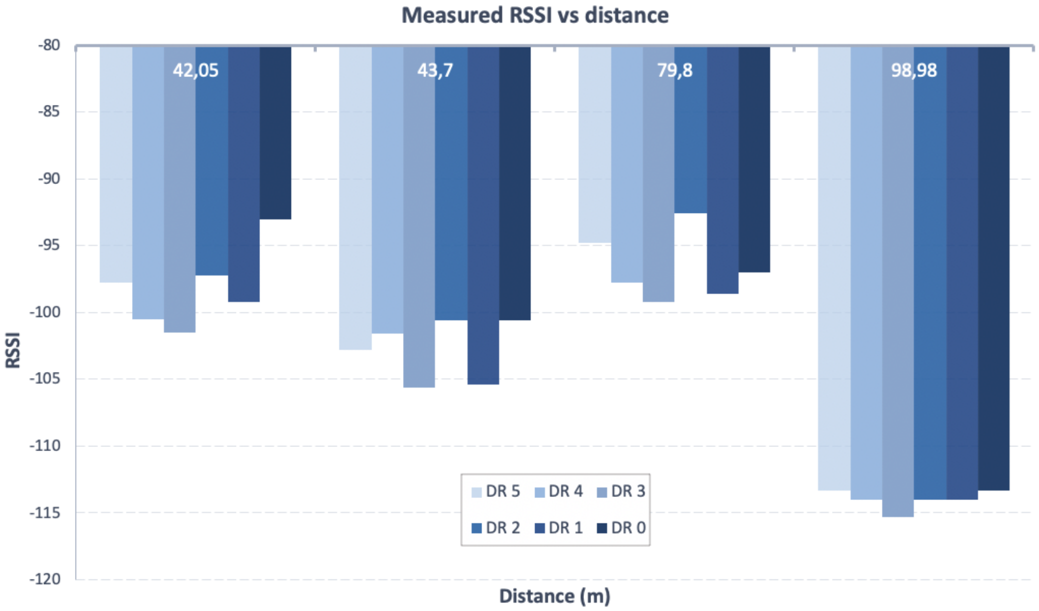


FIGURE 4.2: Average RSSI values plotted for each location with corresponding distances

4.1.2 SNR Assessment

The SNR measurements are also represented in a SF-dependent plot 4.3. Note that the SNR values corresponding to location 4 are negative. This means that the received signal operates below the noise floor.

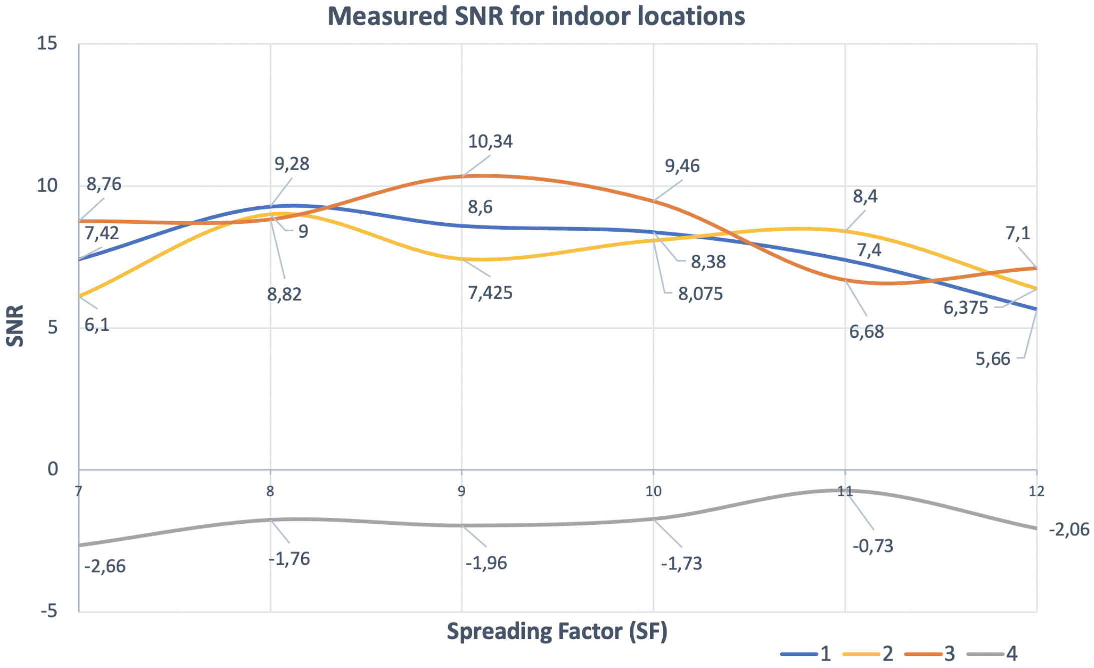


Figure 4.3: Average SNR for each location plotted for each data rate

Finally, the effects of wall thickness on SNR is showcased by plotting values for location 3 & 4. Both of the nodes are located approximately the same distance away from the gateway; however, the wall thickness is different. The resulting plot is used to evaluate the penetration effects of LoRa signal through walls.

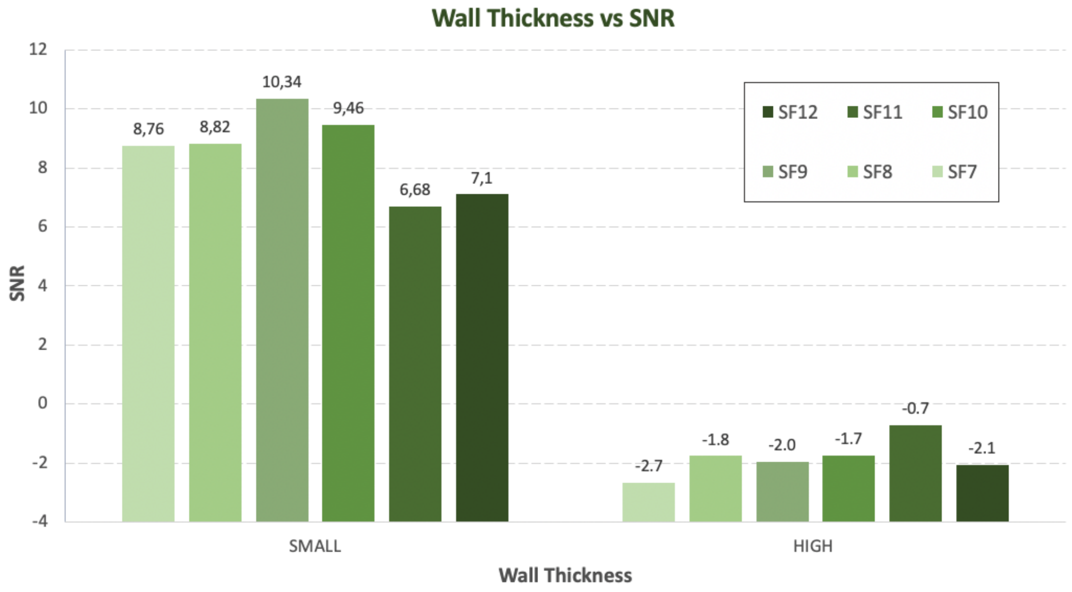


FIGURE 4.4: SNR values for SMALL and HIGH wall thickness at location 3 & 4

4.1.3 Latency Assessment

Table 4.1 shows the average latency for both LoRaWAN and Ethernet communication scenario, detailed in section 3.1.4. DR5 was used for the experiment since it produces the lowest ToA, and thus the lowest latency.

AVERAGE ROUND-TRIP LATENCY	
LAN: Ethernet	LPWAN: LoRaWAN
93 ms	1284 ms

TABLE 4.1: Measured average round-trip latency for LoRaWAN and Ethernet connection

Note that the round-trip latency expresses the total time from end-node uplink to received downlink triggering GPIO command (LED ON/OFF). It can be seen that the LoRaWAN solution contributes with over a second of additional latency.

4.2 Outdoor Coverage Experiments

Results from the outdoor coverage experiments show that only signals from location 1 were received by the indoor gateway. These signals were also received by an outdoor, public gateway, which was not expected before conducting the experiments. The results from these connections further indicate the effects of wall thickness, and the importance of line-of-sight.

4.2.1 Location 1

Measurements from location 1 provide some interesting details. The indoor gateway was only able to receive one uplink out of 15 tries. With DR1, the node was heard by the gateway with a measured RSSI = -115 and SNR = -15. No tests were performed with DR0, due to ADR rejections from the network, which could have resulted in more successful uplinks. At most 6 TTN-powered gateways received signals from the end node at this location. Since most of these gateways were private, these results were discarded. Surprisingly, a public gateway located over 13 km away from location 1 received the signal for all DR configurations. Gateway location is displayed in figure 4.5.

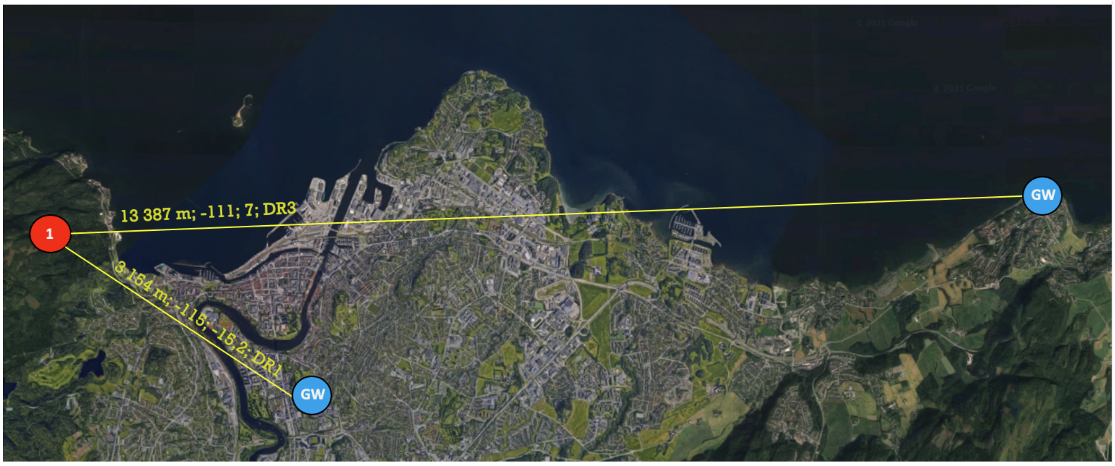


Figure 4.5: Successful uplinks from location 1 (distance;RSSI;SNR;DR)

The signals transmitted by end-nodes at location 1 were received and measured by the public gateway (located to the left in 4.5. These results show that signals received by the public gateway have much better signal qualities (RSSI and SNR values) than those received by the indoor gateway. Note that the public gateway is located more than four times further away.

4.6 shows the measured RSSI and SNR values for signals transmitted from location 1 to the public gateway. The values are plotted for the available DR configurations (DR1-5).

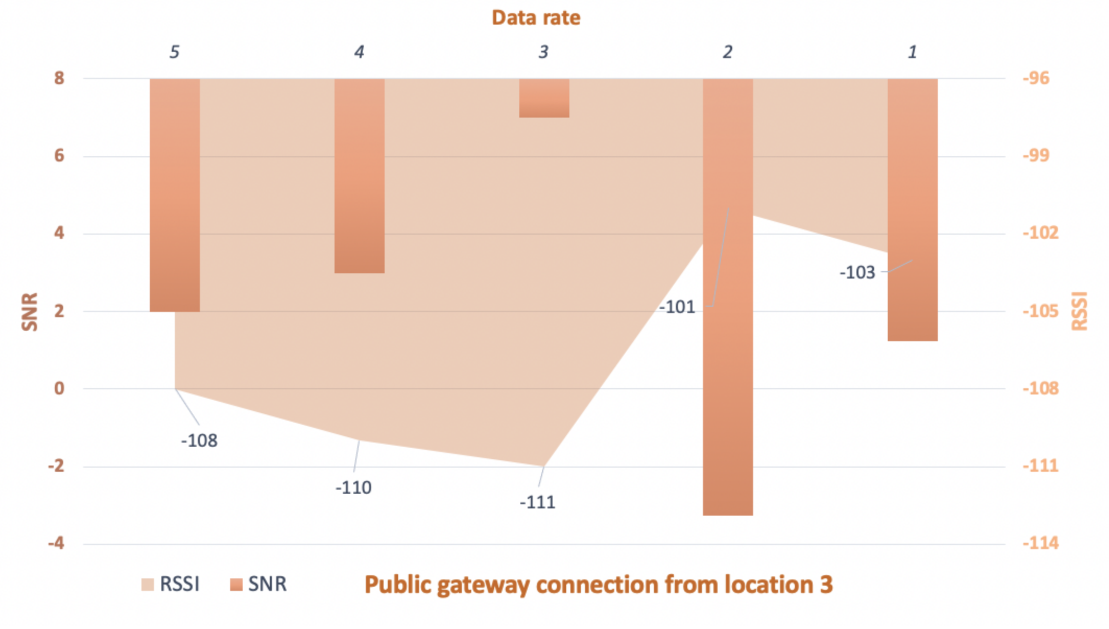


FIGURE 4.6: RSSI and SNR values for connection to public gateway from location 1

4.2.2 Location 2

The end-node at location 2, on top of ”Tyholt Tårnet”, was able to successfully communicate with several gateways, however not the indoor gateway. The other gateways were private TTCS-deployed gateways, which means they only provide information about RSSI and SNR. These gateways do not provide any location or ID information, since this is anonymized by the packet broker on the TTCS network. TTN provides a gateway map that effectively plots the locations of public LoRaWAN gateways connected to their community server. Private gateways however won’t show up on this map, which makes it even harder to pinpoint the exact location. Furthermore, no information about indoor/outdoor situations is detailed for these gateways. Due to the lack of gateway information, these signals were discarded.

4.2.3 Location 3

Signals transmitted by the end-node from location 3 did not reach the indoor gateway located at NTNU. Three attempts were performed with DR configurations 0-5; however, none of the signals were captured by the gateway. The join signal transmitted by the end node did however reach a gateway in proximity. A join-accept message was transmitted by the receiving gateway to the node, but the node did not receive the signal. The indoor gateway did not receive any of the join-request messages.

4.3 Time on Air Calculations & Metering Resolution

This section visualizes ToA calculations for LoRaWAN packets with respect to regional duty cycle, and TTN Fair Use Policy. Figure 4.7 details the staircase increase in ToA based on payload and data rate. Notice the payload limitations for DR0-2 and DR3 (see table A.4). This plot provides a simple and fast way to evaluate payload changes and plan optimal operation for specific applications.

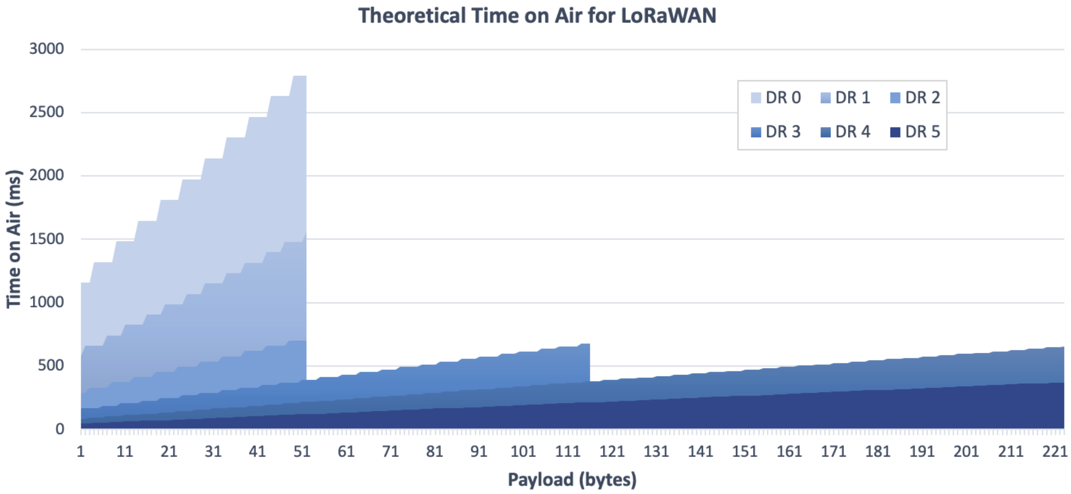


FIGURE 4.7: Visualization of ToA with respect to payload and SF

The resulting ToA's plotted above is used to investigate the metering resolution for devices with varying DR configurations. Using the regional LoRaWAN duty cycle and TTN's Fair Use Policy, it is possible to find out the maximum allowed transmission rate during 24 hours. Figure 4.8 displays uplink limitations for SF7-12 based on the two regulations.

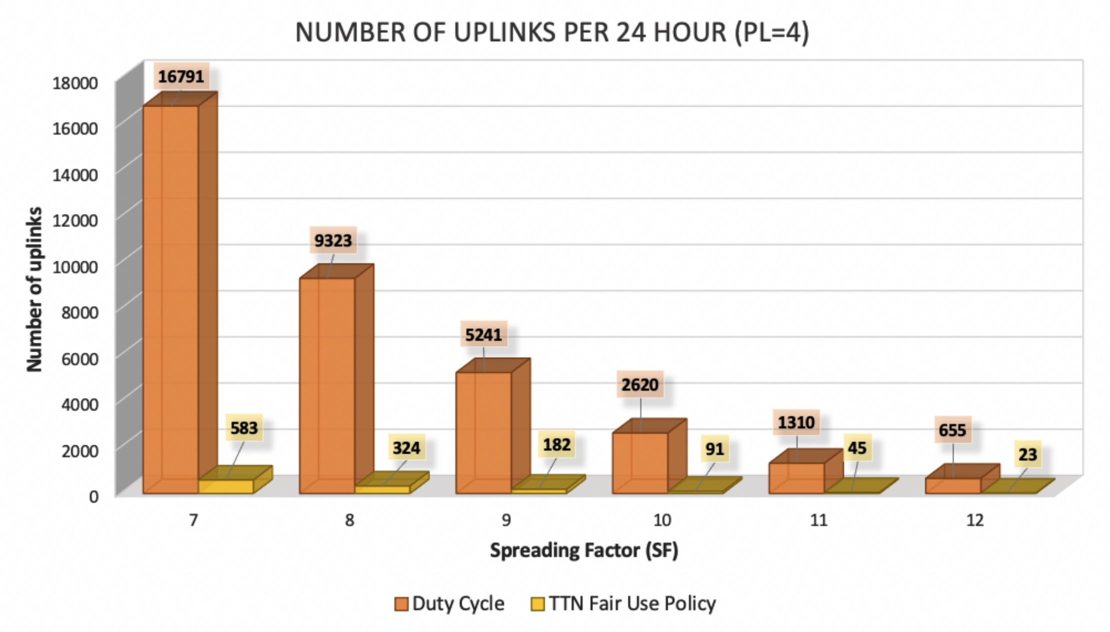


FIGURE 4.8: Uplink limitations for LoRaWAN specific duty cycle and TTN Fair Use Policy

The results from section 4.1 and 4.2 show that using a spreading factor of 7 is highly effective, even across long distances. Smart meters using DR5 and non-TTN LoRaWAN networks could in theory transmit readings every 6 seconds. The same meters would only be able to uplink every 2.5 minutes if operating on TTN/TTCS. Although battery-powered meters would not necessarily need such high metering resolution, it still shows the possibilities for LoRaWAN.

Chapter 5

Discussion

The discussion begins with assessing the experimental results and the main findings. This chapter then use the findings from the results to evaluate coverage, DR configurations and Active DSM (ADSM) possibilities. Then the proposed cloud-based operating center is discussed. Finally, a section covering the system improvements presents some information about possible refinements in the proposed system.

5.1 Coverage Experiments

One of the main findings from the indoor coverage test was the penetration capabilities of signals through walls with different thicknesses. First, figure 4.1 shows that the RSSI values for location 3 lie at the top of the performance range compared to the other three locations. Although this location is second to last in distance from the gateway, it is the only one with wall thickness SMALL. The high-performance values are somewhat linked with shorter distances, however, the amount and types of walls separating the location and the gateway have a large impact. This can be viewed in figure 4.2 where locations one and two have approx. the same distance to the gateway; however, the resulting RSSI is smaller for the location with wall thickness LARGE.

Furthermore, location four is only 10 meters farther away compared to location 3; however, the drop in RSSI is huge. The greatest difference in RSSI between locations three and four is above 20 dBm (DR2). Note that the real RSSI is even smaller when SNR values dip below 0 since the received power level is below the noise floor power level. Translating the logarithmic scaling of RSSI to linear scaling shows that the signal strength indicator is above 100 times stronger for location three compared with four. Again, this also indicates that the wall thickness has a large impact on LoRa transmissions inside a building. Another observation is the fact that RSSI values vary a smaller amount for location 4. A reason for this could be that the larger distance and increased wall thickness reduce the fluctuation in RSSI for all SF's.

Figure 4.4 also underlines the effect of wall thickness, as signals from location 4 have much lower SNR (below noise floor since SNR <0) compared to the other locations. It is interesting that the SNR is more tightly dispersed for locations 1-3, compared to RSSI (see 4.3). It could indicate that close connections have a good signal quality as they are affected by less noise. The SNR for location 4 is quite low compared to the other three, so it can be assumed that this signal was prone to interfering noise. The combination of interfering noise and HIGH wall thickness can reduce the SNR substantially.

Outdoor measurements reveal details about the importance of line-of-sight. Since location 2 has a high altitude and shorter distance to the gateway, one would expect it to perform the best out of the three locations. As it turns out, signals coming from this location were not captured by the gateway. Direct path to the gateway led through 2 buildings, which might be the cause of no connection. Furthermore, the measurements were performed on top of a radio tower, so unwanted noise could also be a contributing factor. Location 1, situated on top of a hill in the woods, managed to achieve a connection on DR configuration 1. The resulting SNR was below -15dB, which is almost below the minimum limit. Signals traveling from this location go through more open space and only one building before reaching the gateway.

The most interesting part of the outdoor experiments came when the end-node transmitting from location 1 made a connection with a public gateway located more than 13km away. Figure 5.1 shows the elevation profile, and map investigations reveal that there is a direct line-of-sight between the end-node and the public gateway. This is the only direct line-of-sight situation observed in the whole thesis. Although the resulting SNR values vary quite much, they are still way above the minimum limit for receivers. The received signal power is also within a good range. These findings show that LoRa's real power comes from direct line-of-sight situations. The problem here is that direct line-of-sight is often not achievable.

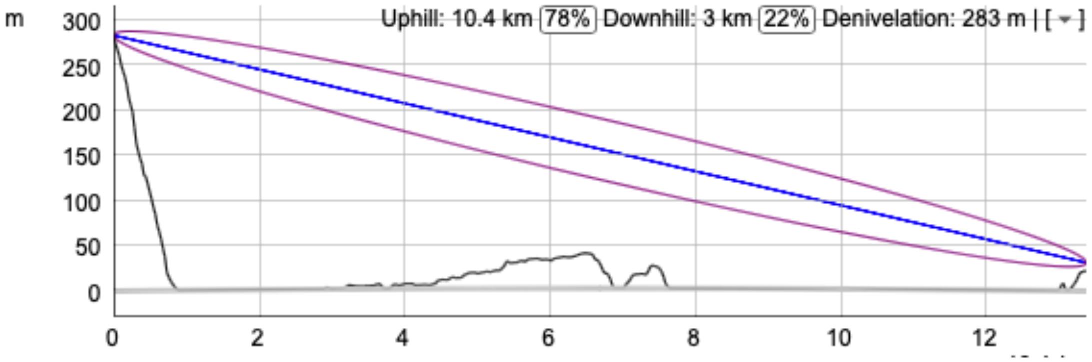


FIGURE 5.1: Elevation profile from location 1 to public gateway

To sum it up, LoRa technology shows promising results for indoor and outdoor network coverage capabilities. Indoor signals transmitted up to 50 meters away from a gateway show small effects on signal quality with respect to wall thickness. However, as this distance increase, wall thickness becomes a more dominating factor in terms of achievable range. With direct line-of-sight, signal qualities can still be high even over long distances.

5.2 Optimal Data Rate Configurations

The data rate is one of the most crucial parameters to control when installing battery-powered nodes. It is necessary to find the optimal SF configuration that provides a reliable signal quality, while still preserving battery power. Nodes located far away might need higher SF's to reach the gateway; however, this can also cause packets to be lost due to collisions. It is observed that data rate configurations have a small impact on indoor signal transmissions 4.3, which prompts installers to use DR5-4 for indoor end-nodes. In the case of large prosumer clusters with thousands of nodes, longer ToA can cause higher collision rates as there are limited amounts of operating channels. The results from outdoor test location 1 reveal that signals from end-nodes using DR5 (SF=7) can achieve sufficient signal strength and quality with good margins, even when transmission distances reach above 10km. As previously discussed this is highly due to direct line-of-sight conditions; however, it goes to show that low SF's can still be used for long-range transmissions.

The basic idea of prosumer clusters is to uplink time-series data about production and consumption while receiving load control signals from the utility as downlink. The rate at which these measurements and control signals occur must be accounted for to ensure the node is not exceeding the duty cycle limitations. This could be a problem for prosumer nodes located at the edge of a cluster, where nodes are required to use a higher SF due to increased distance. One increase in SF results in a number of uplinks being halved, which limits the applications for end-nodes located in those areas. With that being said, the results show that data-rate has almost no effect on wether the signal is received or not. This observation indicates that the DR should always be kept at 5, and that the gateway deployment should be responsible for providing DR5 coverage to all end-nodes.

5.3 Prosumer Cluster Coverage Solutions

The small Raspberry Pi LoRaWAN gateway was inoperable during coverage testing, so the Laird gateway replaced it as an indoor gateway. Since the system is built on top of TTCS, messages received by community gateways are routed to the specific application. That results in the community gateway acting as an outdoor gateway, while the laird gateway acts as an indoor gateway. This solution can be implemented for prosumer clusters with both residential and apartment buildings. Single homes could connect to local district gateways for coverage, while prosumers in large apartment buildings connect to an indoor gateway to ensure coverage for all units. It is showed that this gateway infrastructure allows for deep indoor penetration, as well as outside coverage. With that being said, results indicate that indoor gateways should merely be used for deep indoor coverage, especially if the building is surrounded by other buildings. Installing it on top of the building (outside) could increase the outdoor coverage; however, it can also reduce the indoor coverage.

There is no general method to ensure both indoor and outdoor coverage. When implementing the communication infrastructure for prosumer clusters, extensive testing is needed before rolling out multiple gateways to ensure full coverage. First, the cluster area needs to be mapped. This should be conducted while minding the possibility of further developments (more neighboring clusters), as other districts might use some of the existing infrastructures. Note that these clusters can contain both prosumers and regular consumers. After the area is mapped, installers can start to investigate the optimal gateway solution. This research paper could potentially ease the investigation for installers by giving pointers on what to focus on during the coverage tests. The results underline the long-range capabilities of LoRa; however, it also reveals the importance of line-of-sight. Finding the optimal location should therefore be the main concern when installing gateways in large clusters. Installers shall look for solutions without indoor gateways, if possible since they cover much smaller areas. If indoor gateways are necessary, then optimal outdoor gateway placement should be reevaluated based on the new addition.

The most ”simple” solution to coverage and reliability in LoRaWAN clusters would be to install more gateways. Implementing more gateways results in better coverage both inside and outside, which in turn enables more application opportunities for battery-powered end-nodes. The set-up process for a LoRaWAN gateway is fairly simple, so people from the community can install their own and share coverage. Compared to ZigBee and Z-wave, LoRaWAN end-nodes are not restricted to one gateway. LoRaWAN gateways demodulate and forwards all incoming signals to the network server, which then handles duplicates. This functionality allows prosumer clusters to scale easily with the introduction of new prosumers. Initial clusters with few units can utilize existing coverage and scale as they go. Nodes operating in clusters with many gateways should also consider reducing the transmission power, as some of it might be redundant for reaching the nearest gateway. This increases the battery lifetime and could cause clusters of nodes to split up or shrink. Increasing the density of nodes creates more traffic, which in turn could cause collisions as there are only a few operating channels.

This thesis did not conduct any scalability tests in order to investigate packet losses when scaling up the system. Performing simulations of large clusters are needed in order to ensure reliable packet delivery for systems deploying hundreds of nodes. Elevation mappers could work in parallel with delivery simulations to form software for optimal gateway placement and scaling solutions. Investigating scalability requires more extensive research and dedicated focus, due to the many factors influencing the whole system.

5.4 Metering and ADSM Applications

By now, it is quite obvious that lower SF increases the possibilities for near real-time applications. Maximizing uplink rates allows for uplinks every 5.2s, for payload size 4 bytes (additional to the LoRaWAN header) and SF=7. This is near real-time metering, however, it is still not below one second. On the other hand, high transmission frequency results in decreased battery life and increased traffic. It is up to utility companies to determine the rate at which measurements are transmitted, which in turn is based on their application requirements. Instead of having the same metering resolution across 24-hours, end-nodes can be programmed to increase the metering rate for peak hours. If needed, the utility could transmit metering resolution as downlinks to change node operation. A pre-determined schedule can also be programmed, where end-nodes decrease metering resolution during low-consumption hours and increase during peak hours. The issue of having high-resolution hours for many end-nodes is that it can create more package losses. The utility should only deploy high metering resolutions when needed and stick to the lowest resolution applicable for their application. If the metering resolution is vital for a certain application, then the main object should be to ensure optimal coverage (so end-nodes can use SF = 7).

One of the main requirements for ADSM is low latency communication. Dynamical shedding of household consumption requires a fast communication network that can act upon commands within a short period of time. Total round-trip latency for the LoRaWAN solution was measured to an average of 1284ms, while the LAN (Ethernet) solution averaged 93 ms. The LoRaWAN solution has more than ten times the latency compared to an Ethernet solution; however, it is quite small compared to the maximal metering resolution for class A devices. Downlink latency for class A devices is almost exclusively dependent on the metering solution, as they can only receive commands right after an uplink. As discussed above, maximum metering resolution might not be feasible for many nodes.

Implementing class C devices could potentially shed the total latency, as these devices are always listening. The downlink latency in such cases is independent of uplink rates, and thus reduced to approximately half of the measure round-trip latency. It would also be possible to install class B devices, which could open more receive windows during peak hours. Instead of inserting more uplinks (and thus open more receive windows), class B devices can pragmatically open more receive windows without transmitting before. This could allow for a scheme where the end-nodes receive a downlink with a schedule for receive windows for the next day. These windows are defined on utility side, based on predicted grid conditions.

It is important to note that LoRaWAN is not optimized for real-time operations, especially not for large amounts of downlinks. Gateways can receive signals on many channels, but can only transmit one downlink at a time (it can not receive signals during this time). This can be solved by installing more gateways; however, that is not always the optimal solution. Instead, installers could extend the communication network with other, short-range technologies like Bluetooth and ZigBee. One solution could be to combine LoRaWAN nodes with a Bluetooth module. One single downlink can then provide updated load profiles for all appliances, which the Bluetooth module forwards to the specific device. Again, all of these solutions are highly dependent on the intended use-case and requirements utility.

The paragraphs above do not consider the TTN Fair Use Policy, only EU duty cycle limitations. Although TTCS was used as a network server in this thesis, it is not designed for commercial deployments. The 30 seconds uplink and 10 downlinks a day limitations massively reduce the capabilities of energy metering and ADSM. It is still possible to transmit measurements every 2.5 minutes at maximum, however that means a longer latency for ADSM commands. The best solution for commercial deployment is either TTN's commercial LoRaWAN network stack (TTS) or implementing a private LoRaWAN network on-premise.

5.5 Cloud-Based Operating Center

Realizing a LoRaWAN communication infrastructure reduces the need for edge-side processing because all incoming traffic is handled by TTCS and AWS. The end nodes only need to worry about when to wake up, and what the message payload is. This solution enables a centralized control unit for prosumer clusters, where all parties can access and view the data. The proposed cloud architecture serves as a foundation for storage and message handling in HEMS. It is designed to easily allow expansion in the architecture with other types of services. AWS offers analytic services like Redshift and SageMaker, where the latter is deployed for energy forecasting and optimization. The utility can choose to perform analytics on AWS or in another cloud environment.

One of the key benefits of cloud-based operating centers for prosumer clusters is the ability to scale up services. The platform scale proportional to cluster sizes, without utility or prosumers having to manage the underlying infrastructure. This is vital for utility, as it allows them to only focus on grid-related operations. DynamoDB allows for auto-scaling which dynamically provisions throughput based on traffic patterns. Additions to existing prosumer clusters are automatically accounted for, and thus the system remains reliable to both prosumers and utility. Although auto-scaling ensures reliability, it also comes with some negative impacts regarding cost-effectiveness. Developers need to carefully select partition keys to prevent a few partitions from being over-used. Throughput is evenly distributed for all partitions, so an over-used partition results in unnecessary provisioning of all other partitions. It is also important to mind the number of partitions used in a single table, as partition throughput decreases with the number of partitions. DynamoDB is a good solution for IoT storage; however, it requires proper monitoring to ensure cost-effective operation.

The simplicity of rules in IoT Core made it easy to handle LoRaWAN payloads. It seems like the IoT Core service was created with application development in mind, as it easily integrates with a lot of other AWS services. Working with AWS IoT Core was intricate at first, due to the author's lack of experience with cloud services. Some concepts can take time to fathom, but once understood it is fairly straightforward. IoT rules triggering a lambda function turned out to be the most versatile solution for acting upon incoming traffic. Lambda made it possible to easily structure LoRaWAN payload data, so it could be stored in DynamoDB. The existing cloud architecture has potential for improvements, as one IoT rule, and Lambda function is triggered for each device. In the case of prosumer clusters, one rule and Lambda function might be suitable for a whole cluster of devices. This would of course depend on the number of messages transported during short periods of time.

AWS also offers users the possibility of setting up their own private LoRaWAN network, without having to maintain a network server. The service is integrated into the IoT Core, so the whole operation is gathered in one place. Using AWS LoRaWAN network servers removes the ability to use public gateways registered on TTCS. Furthermore, TTCS also comes with an application server (accessed through a web console) that aids in coding and decoding messages. These functionalities are also available in AWS; however, it is much easier to implement in TTCS.

Using a cloud platform to serve as an operating center for LoRaWAN networks benefits the utility with more freedom to tailor operations and applications. The power and utility industry is getting more and more digitalized, which then encourages companies to implement cloud-based solutions. The benefits of switching from an on-premise solution to a cloud-based one highly dependent on the existing infrastructure within utility companies. The costs of moving operations to the cloud might be too high for some companies to make them profitable. End-users can also benefit from a cloud environment by gaining access through IAM roles. With the right architecture, one could see a platform that encourages more end-to-end interactions between end-users and utility. SaaS's like SNS and QuickSight can be deployed to enhance end-user knowledge about their production and load profile.

5.6 Improvements

As mentioned earlier, scalability analysis is key for understanding system operations in a real-world use case. Although the thesis employs both indoor and outdoor coverage tests, it does not show how the system would react to large amounts of nodes transmitting in parallel. This is especially important when some nodes have longer ToA, as these can have larger impacts on the interference and take up more time on a single channel. It would be necessary to perform simulations and experiments with large groups of nodes in order to find both optimal size and operating functionality. This would give a more detailed picture of the possibilities within HEMS applications. It is necessary to find out how the latency and metering resolution is affected by additional nodes operating at the same time. Such analysis and experiments could make it possible to define a general deployment model for prosumer clusters, where cluster size is determined by input parameters for a specific application. It is quite expensive to buy and deploy hundreds of LoRaWAN end-nodes, so performing satisfactory scalability analysis requires either robust simulation software or large investment costs for end-nodes.

One of the drawbacks of the proposed system in terms of power consumption is the fact that Raspberry Pi's do not come with a sleep mode integrated. Without sleep mode, none of the end-nodes are able to achieve sufficient lifetime with only batteries. Implementing MCs with sleep mode capabilities allows for battery lifetime analysis, which in turn provides further deployment details for prosumer clusters. Node configurations have a huge impact on battery lifetime, so it is necessary to implement this component into the total evaluation. Extensive power consumption experiments can help with identifying the end-nodes in need for class C configurations, based on requirements from the HEMS application. Large scale experiments (and deployments) of prosumer clusters should therefore use end-nodes with carefully selected MC's, in order to best replicate a real-life use case.

Tests regarding Tx power should also be performed, as it is another independent transmission variable. End nodes can configure their own EIRP with a simple AT command. This allows end-nodes to reduce the transmission power of need. It could be of importance to experiment further with various Tx power configurations, in order to see the effects on both signal quality and power consumption. Dynamic configurations could be implemented, where software analyzes and calculates the optimal Tx and DR configurations. ADR (from section 2.5.1) is an existing solution for end-node optimization integrated into LoRaWAN; however, it is possible to create alternative configuration schemes. The DR configuration results presented in this paper with results from similar tests with Tx power could aid in the making of an improved ADR software. Research and experiments into automatic optimal end-node configurations should be of high priority, as it could increase the energy efficiency of a LoRaWAN network.

The thesis does not compare an on-site infrastructure solution with the implemented cloud version. Since the network server is hosted by TTCS, the clear choice was to implement a cloud-based operating center. However, the option of deploying a private LoRaWAN network on-site is still a viable option. Comparing in-house with a cloud solution is complicated because it relies on company need. In the case of on-site, the utility would be responsible for managing their servers on-premise, including application servers, if any. The option of deploying an on-site infrastructure comes with additional costs of managing and maintaining the servers, including the cost of needed hardware, software licenses, etc. Investigating latency for on-site solutions is also an important topic that is not covered. Further research on comparisons between cloud and edge deployment for utility companies should be conducted.

The focus of this thesis was not in the direction of security; however, it has to be mentioned that this is a vital component for LoRaWAN networks. It is especially important in large networks with nodes situated in thousands of households. If ill-minded intruders manage to intercept and read consumption values for a home, they might be able to deduct sensitive information about occupancy (in the case of robberies f.ex.). Furthermore, if devices are controllable over LoRaWAN, then invaders could potentially take control of them and cause harm to residents. It is vital for large LoRaWAN infrastructures like prosumer clusters to carefully design and implement a robust security scheme with no weak spots. In order to realize a commercial prosumer cluster with HEMS, further research into security measurements must be conducted.

Chapter 6

Conclusion & Future Work

6.1 Conclusion

In order to regulate the electricity grid and accommodate the changes in residential energy production and consumption, utility needs to be able to monitor and control energy usage in residential homes. This process requires a large-scale, bi-directional communication network supporting low-latency communication. The main research objective in this thesis was to determine if LoRaWAN was suitable for powering the communication infrastructure of HEMSs. A test bed solution was designed and implemented in order to validate the communication capabilities. The validation objective was investigated through coverage and latency analyses, where theoretical calculations and test bed experiments were conducted. Results from the analysis showed that LoRaWAN can be a suitable communication protocol for HEMS clusters with low-latency requirements, however the deployment matters.

One of the main findings in this thesis details that end-node configuration is a key factor for realizing a low-latency LoRaWAN communication scheme. Results from the latency analysis show that managing device classes and DR configurations allows for near real-time control of the LoRaWAN end-node, but this requires class B or C end-nodes. If the end-node is not capable of operating in device class B or C, then the resulting control latency is limited by the maximum uplink rate for the current DR configuration. Metering applications and DR actions with a minimum 10s latency requirement can still be deployed in systems utilizing class A devices. It is important to mind that the results will likely change in large-scale deployments due to collisions and packet losses.

Results from the coverage experiments show that gateway deployment is one of the most crucial components in realizing HEMS capabilities. Carefully selecting the optimal gateway locations allows for better coverage, as it allows more end-nodes to have better line-of-sight conditions. Results from the coverage experiments showed that wall thickness and distance to end-node were the main contributing factors to whether the gateway received the signal or not. This observation adds to the total findings by showing that coverage issues should not necessarily be addressed with a reduced data rate, but rather with increased gateway coverage. Existing literature tends to address the end-node configurations, instead of discussing the results in terms of gateway placement.

This paper explored the possibility of using cloud-hosted servers to facilitate the LoRaWAN communication network, instead of using on-premise hardware deployments. Implementing a cloud-based operating center allowed the end-nodes to store and retrieve information with relatively low latency. The cloud deployment is an alternative solution for utility companies currently deploying on-premise servers. The proposed cloud architecture serves as a foundation for managing prosumer clusters in AWS and should be further developed to include analytic and visualization services. The cloud deployment used in this thesis might not be suitable for all utility companies, as some prefer to have their data stored on-site. Nevertheless, they still have the option of installing on-site servers to host both the network and application server, or either one of them.

Lack of scalability simulations and delayed hardware delivery were the two main limitations affecting the conclusion in this thesis. Limited time resulted in the exclusion of scalability simulations and/or experiments, which could have impacted the latency and metering resolution findings. Although scalability was touched upon several times in this thesis, it was not sufficiently analyzed for making any conclusions. Future scalability studies should therefore be performed, including both simulations and real-world experiments. Furthermore, delayed hardware delivery of LoRaWAN equipment resulted in increased time limitations. Reduced time for implementation and testing could have had some impact on the thoroughness of the experiments. The coverage experiments only evaluated long and short ranges, so medium ranges were not investigated. With more time it would be possible to perform a broader set of range experiments, which in turn leads to more detailed results.

This thesis has provided a framework for implementing and operating a cloud-based LoRaWAN communication system. It is to serve as an aiding tool in the development of a general LoRaWAN energy management framework. Based on the findings in this paper, it is recommended to implement and further develop schemes for sophisticated gateway deployment and end-node configurations.

6.2 Future Work

Further research and work is drawn from the main findings and limitations of this thesis. It shall be used as a guide for those who intend to perform further research on large-scale LoRaWAN energy management systems. Future studies should mainly revolve around scalability analysis and experiments, as these are needed in order to build a general framework from LoRaWAN energy management systems. The following list contains bullet points with suggestions for further research topics, with focus on scalability. Note that these points should eventually be assessed in a holistic manner, as one depends on the other.

- **Latency simulations/experiments in large-scale deployments.** Although latency is covered extensively in this thesis, further research should be conducted on how large scale deployments effect the latency. This would further define the limitations of large LoRaWAN solutions, and thus identify the feasible energy management applications deployed by utility.
- **Designing a gateway deployment scheme for large-scale deployments.** A dedicated study on the effects of gateway placements should be performed. Having a general gateway deployment procedure is crucial for realizing a common LoRaWAN energy management framework.
- **Analysis of class A end-node lifetime in large-scale deployments.** Lifetime analysis and power consumption measurements is another topic for further research. This research would investigate the possibility of deploying only battery-powered class A devices in an energy management system.
- **Implementing a dynamic end-node configuration algorithm for large-scale deployments.** It is also necessary to develop a dynamic end-node configuration algorithm, in order to optimize traffic and stability on the network. This topic is quite broad, and could require some of the studies presented above to be conducted first. Either way, such algorithms could be a key component in the general framework
- **Evaluating the security in large-scale LoRaWAN deployments.** Finally, it is also necessary to evaluate the security of large-scale LoRaWAN deployments. This study should investigate the possible security breaches, and methods for combating these. Security studies are vital to ensure the protection against sensitive information leaks and unwanted control of household appliances.

Appendix A

Appendix

Wall Thickness	**Description**
SMALL	Direct line-of-sight or separated by less than two thin walls
MEDIUM	Separated by one or more thick walls
LARGE	Separated by several walls

TABLE A.1: Description of Wall Thickness

Data Rate	**Spreading Factor / Bandwidth**	**Bit Rate (bit/s)**
0	12 / 125 kHz	250
1	11 / 125 kHz	440
2	10 / 125 kHz	980
3	9 / 125 kHz	1760
4	8 / 125 kHz	3125
5	7 / 125 kHz	5470

TABLE A.2: Data rates for the RAK811 transeiver module ("LoRaWAN Regional Parameters", 2020)

Spreading Factor (SF)	**SNR required for demodulator**	**Sensitivity**
7	-7.5 dB	-123 dBm
8	-10 dB	-126 dBm
9	-12.5 dB	-129 dBm
10	-15 dB	-132 dBm
11	-17.5 dB	-133 dBm
12	-20 dB	-136 dBm

TABLE A.3: Minimum SNR value required by demodulator at receiver side ("SX1276/77/78/79", 2020)

Data Rate	Payload Size (bytes)
0	51
1	51
2	51
3	115
4	222
5	222
6	222
7	222
8	50
9	115
10	50
11	115

TABLE A.4: Maximum payload size for EU863-870 ("LoRaWAN Regional Parameters", 2020)

Tx power	Configured EIRP
0	Max EIRP
1	Max EIRP - 2dBm
2	Max EIRP - 4dBm
3	Max EIRP - 6dBm
4	Max EIRP - 8dBm
5	Max EIRP - 10dBm
6	Max EIRP - 12dBm
7	Max EIRP - 14dBm
8.. 14	RFU

TABLE A.5: Tx power configurations for EU868 LoRaWAN ("LoRaWAN Regional Parameters", 2020)

```python
def time_on_air(pl, sf, bw = 125000, crc = 1, h=0, cr = 1, de = 0, LoRaWAN_hdr_size = 13):

    '''
    pl = payload in bytes
    sf = spreading factor (7-12)
    crc = cyclic redundancy check (0 = off, 1 = on (default))
    h = header (1 = disabled, 0 = enabled (default))
    de = LowDataRateOptimize (1= enabled, 0 = disabled)
    cr = codingrate (1-4) default = 1
    '''

    t_symbol = 2**(sf)/bw
    t_preamble = t_symbol*12.25
    t_payload = t_symbol*(LoRaWAN_hdr_size+8+max(math.ceil((8*pl-4*sf+28+16*crc-20*h)/(4*(sf-2*de)))*(cr+4),0))

    return t_preamble+t_payload
```

LISTING A.1: Time-on-air Calculator

```python
import os
import time
import pandas as pd
from sys import exit
from rak811.rak811_v3 import Rak811

#Used to import excel file into dataframe
demand = pd.read_excel('ProsumernetloadkWh.xls', sheet_name=0, header = None)
pv = pd.read_excel('ProsumernetloadkWh.xls', sheet_name=1, header = None)

#Lora module hard reset
def reset(lora):
    lora.hard_reset()
    return lora

def gen_uplink_message():
    time_now = time.localtime()
    dem_now = 1000*round(demand[0][time_now[7]*24+time_now[3]], 7)
    pv_now = 1000*round(pv[0][time_now[7]*24+time_now[3]], 7)

    '''Format: demand, production'''
    return int(dem_now).to_bytes(2, 'big') + int(pv_now).to_bytes(2, 'big')

#This function ensures that the module use the correct join channels
def set_join_channels(lora, join = 0):
    for i in [*range(3,14)]:
```

```
        command = 'lora:ch_mask:{}:{}'.format(i,join)
        lora.set_config(command)
    return lora

#Set LoRa datarate
def set_dr(lora, dr):
    lora.set_config('lora:dr:{}'.format(dr))
    return lora

def main():
    #Initialize RAK811 object and configurations
    lora = Rak811()
    lora = reset(lora)
    lora.set_config('lora:work_mode:0')
    lora.set_config('lora:join_mode:0')
    lora.set_config('lora:region:EU868')
    lora.set_config('lora:app_eui:xxxxxxxxxxxxxxxx')
    lora.set_config('lora:app_key:xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx')

    #Turn of confirmation downlink
    lora.set_config('lora:confirm:0')

    #Join process
    lora = set_join_channels(lora)
    lora = set_dr(lora, 0)
    counter = 0
    print('Trying to join')
    while True:
        try:
            lora.join()
            time.sleep(5)
            break
        except:
            #Terminates loop after join-requests
            if counter >= 5:
                os.system("sudo shutdown -h now")
            else:
                counter += 1

    #Opens the other channels back up
    lora = set_join_channels(lora, 1)

    #Uplink code
    for times in range(1):
        for j in [5,4,3,2,1]:
            try:
                print('Sending packet using dr = {}'.format(j))
                lora = set_dr(lora, j)
```

```
                time.sleep(5)
                lora.send(gen_uplink_message())
                print('sleeping')
            except:
                print('failed to uplink')
            time.sleep(20)

    #Close the connection and shut the RPi off.
    lora.close()
    os.system("sudo shutdown -h now")
```

Listing A.2: Join and uplink script for RPi with RAK811

References

Adaptive data rate. (n.d.). Retrieved from https://www.thethingsnetwork.org/docs/lorawan/adaptive-data-rate/index.html

Al-Sarawi, S., Anbar, M., Alieyan, K., & Alzubaidi, M. (2017). Internet of things (iot) communication protocols: Review. In *2017 8th international conference on information technology (icit)* (p. 685-690). doi: 10.1109/ICITECH.2017.8079928

Anjana, S. P., & Angel, T. S. (2017). Intelligent demand side management for residential users in a smart micro-grid. In *2017 international conference on technological advancements in power and energy (tap energy)* (p. 1-5). doi: 10.1109/TAPENERGY.2017.8397265

Askeland, M., Backe, S., Fagerstrøm, J., Crespo del Granado, P., Hofmann, M., Jaehnert, S., … Throndsen, W. (2018). *Prosumers' role in the future energy system* (Tech. Rep.). Retrieved from http://hdl.handle.net/11250/2600288

Berg, T. J. (2020). *Smart lighting with ai in human-centric home energy automation systems.*

Bingöl, E., Kuzlu, M., & Pipattanasompom, M. (2019). A lora-based smart streetlighting system for smart cities. In *2019 7th international istanbul smart grids and cities congress and fair (icsg)* (p. 66-70). doi: 10.1109/SGCF.2019.8782413

A computer in the basement? (1968, Apr). *Popular Mechanics*, *129*(4), 77–79.

Daud, A. R., Thariq, M. N., Kaulika, V. O., Nugraha, A. B., & Adiono, T. (2020). Digital electricity meter with two way internet communication. In *2020 ieee international conference on communication, networks and satellite (comnetsat)* (p. 194-197). doi: 10.1109/Comnetsat50391.2020.9328998

Demand-side management glossary (Technical Report No. TR-101158). (1992). 3420 Hillview Avenue, Palo Alto, California 94304: Electric Power Research Institute.

Dinh, H. T., Yun, J., Kim, D. M., Lee, K.-H., & Kim, D. (2020). A home energy management system with renewable energy and energy storage utilizing main grid and electricity selling. *IEEE Access*, *8*, 49436-49450. doi: 10.1109/ACCESS.2020.2979189

Elkazaz, M., Sumner, M., Davies, R., Pholboon, S., & Thomas, D. (2019). Optimization based real-time home energy management in the presence of renewable energy and battery energy storage. In *2019 international conference on smart energy systems and technologies (sest)* (p. 1-6). doi: 10.1109/SEST.2019.8849105

Gemalto, A., & Semtech. (2017, Feb). *Lorawan security.* LoRa-Alliance. Retrieved from https://lora-alliance.org/wp-content/uploads/2020/11/lorawan_security_whitepaper.pdf

Geneiatakis, D., Kounelis, I., Neisse, R., Nai-Fovino, I., Steri, G., & Baldini, G. (2017). Security and privacy issues for an iot based smart home. In *2017 40th international convention on information and communication technology, electronics and microelectronics (mipro)* (p. 1292-1297). doi: 10.23919/MIPRO.2017.7973622

Goyal, P., Sahoo, A., Sharma, D. T., & Singh, P. (2020, 05). Internet of things: Applications, security and privacy: A survey. *Materials Today: Proceedings*, *34*. doi: 10.1016/j.matpr.2020.04.737

Grid, S. (2016). Department of Energy. Retrieved from https://www.energy.gov/sites/prod/files/2016/12/f34/AMI%20Summary%20Report_09-26-16.pdf

Han, J., Choi, C.-s., Park, W.-k., Lee, I., & Kim, S.-h. (2014). Smart home energy management system including renewable energy based on zigbee and plc. *IEEE Transactions on Consumer Electronics*, *60*(2), 198-202. doi: 10.1109/TCE.2014.6851994

Hogg, D. (1993). Fun with the friis free-space transmission formula. *IEEE Antennas and Propagation Magazine*, *35*(4), 33-35. doi: 10.1109/74.229847

Iglesias Vázquez, F., Kastner, W., Gaceo, S., & Reinisch, C. (2011, 01). Electricity load management in smart home control..

Iot technologies and protocols: Microsoft azure. (n.d.). Retrieved from https://azure.microsoft.com/en-us/overview/internet-of-things-iot/iot-technology-protocols/

Jebril, A. H., Sali, A., Ismail, A., & Rasid, M. F. A. (2018). Overcoming limitations of lora physical layer in image transmission. *Sensors*, *18*(10). Retrieved from https://www.mdpi.com/1424-8220/18/10/3257

Kang, L. (2021). Street architecture landscape design based on wireless internet of things and gis system. *Microprocessors and Microsystems*, *80*, 103362. Retrieved from https://www.sciencedirect.com/science/article/pii/S0141933120305214 doi: https://doi.org/10.1016/j.micpro.2020.103362

Khamphanchai, W., Saha, A., Rathinavel, K., Kuzlu, M., Pipattanasomporn, M., Rahman, S., … Haack, J. (2014). Conceptual architecture of building energy management open source software (bemoss). In *Ieee pes innovative smart grid technologies, europe* (p. 1-6). doi: 10.1109/ISGTEurope.2014.7028784

Kuzlu, M., Pipattanasomporn, M., & Rahman, S. (2015). Review of communication technologies for smart homes/building applications. In *2015 ieee innovative smart grid technologies - asia (isgt asia)* (p. 1-6). doi: 10.1109/ISGT-Asia.2015.7437036

Lang, M., Wiesche, M., & Krcmar, H. (2018, 03). Criteria for selecting cloud service providers: A delphi study of quality-of-service attributes. *Information Management*, *55*. doi: 10.1016/j.im.2018.03.004

Lavric, A., & Popa, V. (2017). Internet of things and lora™ low-power wide-area networks: A survey. In *2017 international symposium on signals, circuits and systems (isscs)* (p. 1-5).

doi: 10.1109/ISSCS.2017.8034915

Liu, N., Yu, X., Wang, C., Li, C., Ma, L., & Lei, J. (2017). Energy-sharing model with price-based demand response for microgrids of peer-to-peer prosumers. *IEEE Transactions on Power Systems*, *32*(5), 3569-3583. doi: 10.1109/TPWRS.2017.2649558

Lora modulation basics (Semtech Application Note No. AN1200.22). (2015). 200 Flynn Road, Camarillo, CA 93012: Semtech.

Lorawan 1.1 specification (Documentation). (2017). 3855 SW 153rd Drive: LoRa Alliance.

Lorawan regional parameters (Computer software manual No. 1.0.2). (2020, 10). (Rev. B)

Mahapatra, B., & Nayyar, A. (2019, 11). Home energy management system (hems): concept, architecture, infrastructure, challenges and energy management schemes. *Energy Systems*. doi: 10.1007/s12667-019-00364-w

Mataloto, B., Ferreira, J. C., & Cruz, N. (2019). Lobems — iot for building and energy management systems. *Electronics*, *8*(7). Retrieved from https://www.mdpi.com/2079-9292/8/7/763

Muzi, F., Calcara, L., Pompili, M., & Sangiovanni, S. (2018). The new prosumer tasks in the energy management of buildings. In *2018 ieee international conference on environment and electrical engineering and 2018 ieee industrial and commercial power systems europe (eeeic / i cps europe)* (p. 1-4). doi: 10.1109/EEEIC.2018.8494537

National action plan on demand response (Technical Report No. AD09-10). (2010). Washington, DC 20426: Federal Energy Regulatory Commission.

Network, T. (n.d.). *Ground breaking world record! lorawan packet received at 702 km (436 miles) distance.* Retrieved from https://www.thethingsnetwork.org/article/ground-breaking-world-record-lorawan-packet-received-at-702-km-436-miles-distance

Paterakis, N. G., Erdinç, O., & Catalão, J. P. (2017). An overview of demand response: Key-elements and international experience. *Renewable and Sustainable Energy Reviews*, *69*, 871-891. Retrieved from https://www.sciencedirect.com/science/article/pii/S1364032116308966 doi: https://doi.org/10.1016/j.rser.2016.11.167

Peter Mell, T. G. (2011, 9). The nist definition of cloud computing (Computer software manual No. 800-145).

Pietruszka, K. (2010, Apr). *Geocontext profile.* Retrieved from http://www.geocontext.org/publ/2010/04/profiler/en/

Pipattanasomporn, M., Kuzlu, M., & Rahman, S. (2012). An algorithm for intelligent home energy management and demand response analysis. *IEEE Transactions on Smart Grid*, *3*(4), 2166-2173. doi: 10.1109/TSG.2012.2201182

Power supply. (n.d.). Retrieved from https://www.raspberrypi.org/documentation/hardware/raspberrypi/power/README.md

Ratasuk, R., Mangalvedhe, N., Zhang, Y., Robert, M., & Koskinen, J.-P. (2016). Overview of narrowband iot in lte rel-13. In *2016 ieee conference on standards for communications and networking (cscn)* (p. 1-7). doi: 10.1109/CSCN.2016.7785170

Richter, F. (2021, Feb). *Infographic: Amazon leads $130-billion cloud market.* Retrieved from https://www.statista.com/chart/18819/worldwide-market-share-of-leading-cloud-infrastructure-service-providers/

Sağır, S., Kaya, , Şişman, C., Baltacı, Y., & Ünal, S. (2019). Evaluation of low-power long distance radio communication in urban areas: Lora and impact of spreading factor. In *2019 seventh international conference on digital information processing and communications (icdipc)* (p. 68-71). doi: 10.1109/ICDIPC.2019.8723666

Seye, M. R., Ngom, B., Gueye, B., & Diallo, M. (2018). A study of lora coverage: Range evaluation and channel attenuation model. In *2018 1st international conference on smart cities and communities (sccic)* (p. 1-4). doi: 10.1109/SCCIC.2018.8584548

Shah, D., & Bharadi, V. (2016, 12). Iot based biometrics implementation on raspberry pi. *Procedia Computer Science*, *79*, 328-336. doi: 10.1016/j.procs.2016.03.043

Sintef. (n.d.). *Smart meters.* Retrieved from https://www.sintef.no/en/expertise/sintef-energy-research/advanced-metering-and-control-systems/

Smil, V. (2017). *Energy transitions: global and national perspectives.* Praeger, an imprint of ABC-CLIO, LLC.

Soliman, M., Abiodun, T., Hamouda, T., Zhou, J., & Lung, C.-H. (2013). Smart home: Integrating internet of things with web services and cloud computing. In *2013 ieee 5th international conference on cloud computing technology and science* (Vol. 2, p. 317-320). doi: 10.1109/CloudCom.2013.155

Sx1276/77/78/79 (Computer software manual No. SX1276-7-8-9). (2020, 5). (Rev. 7)

Tay, N. N. W., Botzheim, J., & Kubota, N. (2018). Human-centric automation and optimization for smart homes. *IEEE Transactions on Automation Science and Engineering*, *15*(4), 1759-1771. doi: 10.1109/TASE.2018.2789658

Time-of-use tariffs: Innovation landscape brief (Technical Report). (2019). Abu Dhabi: International Renewable Energy Agency.

Trymjb. (2021, May). *Forward join-accept message, but rak811 says lora join failed.* Retrieved from https://www.thethingsnetwork.org/forum/t/forward-join-accept-message-but-rak811-says-lora-join-failed/47741

Wholesale market. (n.d.). Retrieved from https://www.nve.no/norwegian-energy-regulatory-authority/wholesale-market/

Withanage, C., Ashok, R., Yuen, C., & Otto, K. (2014). A comparison of the popular home automation technologies. In *2014 ieee innovative smart grid technologies - asia (isgt asia)* (p. 600-605). doi: 10.1109/ISGT-Asia.2014.6873860

Trym Johannes Berg

Investigating a LoRaWAN Communication Infrastructure For Cloud-Based Home Energy Management

NTNU
Norwegian University of
Science and Technology