UPMC SORBONNE UNIVERSITÉS

SORBONNE UNIVERSITÉ

École doctorale 364 : Sciences Fondamentales et Appliquées

Doctorat ParisTech

THÈSE

pour obtenir le grade de docteur délivré par

ESIEE School/Paris-Est university

Spécialité doctorale "Informatique"

présentée et soutenue publiquement par

Aghiles DJOUDI

le 18 décembre 2014

LoRAWAN

Directeur de thèse : **xx XX**
Co-encadrant de thèse : **xx XX**

Jury

Mme xx XX,	Professeur	Examinateur
M. xx XX,	Professeur	Rapporteur
Mme xx XX,	Professeur	Examinateur
Mme xx XX,	Professeur	Examinateur

THESE

Abstract

Nowadays, Internet of things is witnessing a tremendous evolution due to the increasing growth in communication technologies, weather and environmental sensing, health care sensing. Indeed, sensors are being a kind of intelligent mobile agent able to perceive its environment and transmit information to the cloud for processing, This way of perception allow the development of several kinds of applications to enhance human capacity to understand their environment and make appropriate decision. However, developing such advanced applications relies heavily on the quality of the communication between sensors and between sensors and the infrastructure, therefore, such communication can be realized only with the help of a secure data collection and efficient data treatment and analysis.

Data collection in a vehicular network has been always a real challenge due to the specific characteristics of these highly dynamic networks (frequent changing topology, vehicles speed and frequent fragmentation), which lead to opportunistic and non long-lasting communications. Security, remains another weak aspect in these wireless networks since they are by nature vulnerable to various kinds of attacks aiming to falsify collected data and affect their integrity. Furthermore, collected data are not understandable by themselves and could not be interpreted and understood if directly shown to a driver or sent to other nodes in the network. They should be treated and analyzed to extract meaningful features and information to develop reliable applications. In addition, developed applications always have different requirements regarding quality of service (QoS). Several research investigations and projects have been conducted to overcome the aforementioned challenges. However, they still did not meet perfection and suffer from some weaknesses. For this reason, we focus our efforts during this thesis to develop a platform for a secure and efficient data collection and exploitation to provide vehicular network users with efficient applications to ease their travel with protected and available connectivity. Therefore, we first propose a solution to deploy an optimized number of data harvesters to collect data from an urban area. Then, we propose a new secure intersection based routing protocol to relay data to a destination in a secure manner based on a monitoring architecture able to detect and evict malicious vehicles. This protocol is after that enhanced with a new intrusion detection and prevention mechanism to decrease the vulnerability window and detect attackers before they persist their attacks using Kalman filter. In a second part of this thesis, we concentrate on the exploitation of collected data by developing an application able to calculate the most economic itinerary in a refined manner for drivers and fleet management companies. This solution is based on some information that may affect fuel consumption, which are provided by vehicles and other sources in Internet accessible via specific APIs, and targets to economize money and time. Finally, a spatio-temporal mechanism allowing to choose the best available communication medium is developed. This latter is based on fuzzy logic to assess a smooth and seamless handover, and considers collected information from the network, users and applications to preserve high quality of service.

Résumé

ᓱᕋᕕᕕᓄᐱ

ᓱᕋᕕᕕᓄᐱ ᐅ ᒪᑕᓄᒐ ᕋᐅᘐᘐᖏᓄᐅᑎ ᑕᓄᐱ ᔪᑦᐠᕋ ᙅᙆᘓᓄᐅᐠᕋᕕ ᕋᐅᙅᒪᓄᑎ ᓱ ᔪᕋᙅᙆ

ملخص

الأفكار الخضراء عديمة اللون تنام بغضب

Acknowledgements

Dedication

I dedicate this work to those who are denied to study by lack of money, health and means.

Contents

List of Tables

List of Figures

List of Abbreviations

CSS Chirp Spread Spectrum (Proprietary)

DSSS Direct Sequence Spread Spectrum

UNB Ultra narrow band

List of Nomenclatures

ADR Adaptive Data Rate

BER Bit Error Rate

BS Base Station

BW Bandwidth

CF Carrier Frequency

CR Coding Rate

DR Data Rate

FCM Fuzzy C-Means

FEC Forward error correction

FL Fuzzy Logic

IoT Internet of things

LoRa Long Range

LoRaWAN Long Range Wireless Access Network

LPWAN Low Power Wide Area Networks

LQI Link Quality Indicator

LS Link Symmetry

NB-IoT Narrow Band-Internet of Things

PDR Packet delivery ratio

PL Path loss

PS Payload size

QoS Quality of Service

RSSI Received Signal Strength Indication

SF Spreading Factor

SINR Signal-to-interference & noise ratio

SNR Signal Noise Rate

ToA Time on Air

TTN The Things Network

WSN Wireless Sensor Network

List of Publications

"There's no absolutely reliable way to achieve a great citation. However, hardworking could be fruitful" - Eraldo Banovac

International Conferences

National Conferences

Journals

Survey

1 | Introduction

"The secret of a good sermon is to have a good beginning and a good ending, then trying to have the two as close as possible." - George Burns

Contents

1 Context and motivation

2 Methodology and contributions

3 Organization of the thesis

2 | Academic Survey

"Given one hour to save the planet, I would spend 59 minutes understanding the problem and one minute resolving it." - Albert Einstein

[1]E7535MLG

In this section, we present the LoRa/ LoRaWAN technology. Even if the terms LoRa and LoRaWAN are used interchangeably but they refer to two different concepts in the network. In fact LoRa corresponds to the PHYSICAL layer and precisely to the modulation technique used and LoRaWAN defines the LoRa MAC layer. 2.1 LoRa Modulation: Physical Layer LoRa technology is a proprietary physical modulation designed and patented by Semtech Corporation. It is based on Chirp-Spread Spectrum (CSS) modulation [2] with Integrated Forward Error Correction. LoRa operates in the lower ISM bands (EU: 868 MHz and 433 MHz, USA: 915 MHz and 433 MHz). It offers different configurations (data rates, transmission range, energy consumption and resilience to noise) according to the selection of four parameters which are Carrier Frequency (CF), Bandwidth (BW), Coding Rate (CR) and Spreading Factor (SF). Each LoRa symbol is composed of 2 SF chirps [Project DecodingLoRa], where SF represents the corresponding spreading factor in the range of 6 to 12. SF6 means a shortest range, SF12 will be the longest. Each step up in spreading factor doubles the time on air to transmit the same amount of data. The use of a larger SF decreases the bit rate and increases the time on Air (ToA) which induces greater power consumption. In fact, in the case of a 125 kHz bandwidth and a coding rate 4/5, the bit rate is equal to 250 bps for SF12 and it is equal to 5470 bps for SF7 [LoRa Alliance Technical committee LoRawan regional parameters]. With LoRa, transmissions on the same carrier frequency but with different spreading factors are orthogonal, so there is no interference. 2.2 LoRaWan: LoRa Mac Layer Unlike the proprietary LoRa protocol, LoRaWAN is an open protocol defined by LoRa Alliance. A LoRaWAN network is based on star-of-stars topology and is composed of three elements. – End devices: nodes that send uplink (UL) traffic and receives Downlink (DL) traffic through LoRa gateways. The communication between end-devices and gateways is based on LoRa modulation. – LoRa gateways dispatch the LoRaWAN frames received from end devices via IP connections (using Ethernet, 3G, 4G or Wi-Fi, etc.) to a network server.

– A network server decodes the packets, analyzes information mined by end devices and generate the packets that should be sent to end devices. LoRaWAN end devices implement three classes: a basic LoRaWAN named Class A and optional features (class B, class C) [3]. LoRaWAN operates in ISM bands (863–870 MHz band in Europe) which are subject to regulations on radio emissions, thus radios are required to adopt either a Listen-Before-Talk (LBT) policy or a duty cycled transmission to limit the rate at which the end devices can actually generate messages. The current LoRaWAN specification exclusively uses duty-cycled limited transmissions to comply with the ETSI regulations [LoRa Alliance Technical committee LoRawan regional parameters]. In fact, each device is limited to an aggregated transmit duty cycle of 1% that means 36 s per hour.

LoRaWAN defines three MAC message types in [3] which are: the join message for connecting a device with a network server, the confirmed message which have to receive an ACK from a network server, and the unconfirmed message without ACK. A MAC payload length varies between 59 and 250 Bytes depending on the modulation rate [LoRa Alliance Technical committee LoRawan regional parameters]. 3 Related Work on LoRa Performance Enhancement In order to optimize the performance of a LoRa network and the quality of service, we identified three complementary approaches: (1) parameter selection, (2) data compression, (3) activity time sharing.

3.1 LoRa Parameter Selection As explained in previous section, for satisfying a desired performance level, one can choose his configuration by combining the various parameters CR (4/5, 4/6, 4/7 and 4/8), BW (125 kHz, 250 kHz and 500 kHz), SF (from 7 to 12) and TP (2 dBm to 17 dBm), resulting

in total 1152 combinations.

[4] QN8Y27W6

In HT-WLANs, dynamic link adaptation can be classified into two categories as follows. (i) Link adaptation in static environment: MiRA [5] is a dynamic data rate adaptation approach that selects spatial streams and rates. It is based on MIMO technology and the receiver's feedback. In poor channel condition, MiRA performs excessive rate selection. Further, RAMAS [6] is a credit-based scheme that also applies MIMO streams. So, this approach incurs overhead of assigning credit to select data rate.

[7] R2PBNFAQ

Selecting communication parameters of wireless transmitters to reduce energy consumption is a well researched area. In the Wireless Sensor Networks (WSN) research domain a large amount of research has been undertaken that investigates transmission power control to reduce transmission energy consumption (examples are [Transmission power control techniques for wireless sensor networks], [8], [9]). Typical transceivers used for WSNs only provide transmission power as means to influence energy consumption. Existing algorithms to adjust transmission power depend on probe transmissions; often data transmissions double as probe transmissions. Link quality is either determined by counting lost/erroneous packets over time and/or by estimation using RSSI or Link Quality Indicator (LQI). Depending on the current link quality, transmission power is adjusted. We follow in our work these established principles. However, LoRa transceivers as used in this work provide additional parameters to influence communication energy cost which we take into account. Previous work on WiFi and cellular networks has investigated either transmit power control (e.g. [10], [11], [8]), transmit rate control (e.g. [12], [sourceforge.net/p/madwifi/svn/HEAD/tree/madwifi/trunk/ ath rate/minstrel/minstrel.txt], [13]), or a combination of the two as 'joint transmit power and rate control' (e.g. [14], [**subramanian_joint_2005**], [15]. Most of the transmit power control is concerned with increasing the capacity, and not necessarily the energy consumption. The transmit rate control is often only concerned with maximising throughput. Compared to LoRa, WiFi data rates and packet rates are significantly higher, and the control algorithms run at a much higher rate then what is feasible with LoRa. For example, the most commonly used transmit rate control algorithm Minstrel [sourceforge.net/p/madwifi/svn/HEAD/tree/madwifi/trunk/ ath rate/minstrel/minstrel.txt] evaluates its links every 100 ms.

[16] HNKJMCMR

The current literature on LoRaWAN systems can be divided in three fields: i) works dealing with an overview of the current technology and proposing new solutions to optimize its performance [**bor_lora_nodate**] [17]; ii) papers aiming at analyzing the LoRa capabilities and studying their performance in specific scenarios [18][19][20][21]; iii) works defining channel models (through simulations in different environments and scenarios) and emphasizing how these infrastructures are sensitive to the environment in which they operate [22].

[23] I4JZZ98I

There is limited published work discussing scalability of LoRa.

Closest to this paper is the work by Petajajarvi et al. [22] and our own previous work reported in [**bor_lora_nodate**].

A vast number of generic wireless simulation tools such as ns-3 [Modeling and tools for network simulation] or OMNet++ [The OMNeT++ discrete event simulation system] exist. There are also simulators such as Cooja [Cross-level sensor network simulation with cooja] or TOSSIM [Tossim: accurate and scalable simulation of entire tinyos applications] designed for Wireless Sensor Networks (WSN) and IoT environments. These simulators can be extended by the components designed for our simulator LoRaSim to enable LoRa simulations. The Semtech LoRa modem calculator [www.semtech.com/apps/filedown/down.php? file=SX1272LoRaCalculatorSetup1%271.zip] helps with analysis of LoRa transmission features (airtime of packets, receiver sensitivity) but does not enable network planning. Siradel provides a simulation tool called S IOT [www.siradel.com/portfolio-item/alliance-lora]. S IOT relies on Volcano, a 3D-ray tracing propagation model and a portfolio of 2D and 3D geodata. The tool supports sink deployment decisions based on propagation models. This commercial tool considers the environment to a much greater detail than our simulator LoRaSim. However, it does not take into account actual traffic, collisions or details such as capture effect. Our models provided in Section 3 could be used to improve S IOT.

Moreover, [**bor_lora_nodate**] describes LoRaBlink, an IoT protocol for LoRa transceivers designed to support reliable and energy efficient low-latency bi-directional multi-hop communication.

However, the performance drastically decrease when the link load increases. Limits and potentialities of LoRaWAN are studied by Voigt et al. [18]. Through simulations based on real experimental

data, the paper shows that interference can drastically reduce the performance of a LoRa network. They also demonstrate that directional antennas and using multiple base stations can improve performance under interference. Scalability issues in the LoRA system are analyzed in several papers [23][19][20].

Finally, [21] and [24] derive throughput behavior and capacity limits under some ideal conditions (perfect orthogonality of the SFs).

[25] SG3YQG82

The LoRaWAN has attracted the IoT community as a promising platform for supporting smart city deployments. Thus, throughout the last years, different works have analyzed the technology limits and addressed open issues such as scalability. In this context, we can classify the related literature as follows: (i) Works analyzing the current capabilities and limitations of LoRaWAN [26][27][19][23], and studing its performance under specific settings [24][28][**varsier_capacity_2017**]. (ii) Papers proposing novel approaches and heuristics to optimize the network performance [**bor_lora_nodate**][21][29][30][16]. As for the first group, Adelantado et al. [26] the limits of LoRaWAN. An issue concerns the maximum duty cycle (DC) allowed within the ISM band. For instance, the 1% for the U E 868 M Hz band turns out into a maximum transmission time of 36 secs in an hour, for each device. This also limits the LoRa gateways in the down-link channel, which have to comply with the DC regulation. Another important analysis in [26] regards the use of ALOHA in a LoRaWAN deployment, which simplifies the network implementation, but at the expense of the throughput that is significantly limited by collisions.

Scalability issues also have been addressed in by Bor et al. [23] where it was identified a LoRa link model for the communication range and the collision behavior. They also provided the LoRa simulator (LoRaSim) implementing the link behavior model. In addition, it has been of interest the evaluation the LoRaWAN performance in smart city scenarios.

Other works dealt with application-tailored deployments such as in [28] where it was studied the support of LoRa for health care monitoring, or in [**varsier_capacity_2017**] for hosting smart metering devices. For optimizing the performance of LoRa, many works have addressed the scalability issue. To this aim, several heuristics have been focused on how to efficiently allocate the wireless resources.

Other efforts for optimizing the network have been done tackling other solutions such as the usage of new LoRa transceivers [**bor_lora_nodate**] or the development of multi-hop communication for choosing the minimal Time-on-Air path [30].

[31] RQLF94IS

This section presents the related works regarding SF assignment and analysis of LoRaWAN in both confirmed and unconfirmed mode. The performance of LoRaWAN network with the only unconfirmed mode in an urban environment is presented in [24].

Another work shows the performance of LoRaWAN network by performing a system-level simulation on NS-3 when heterogeneous traffics are transferred for smart metering communication [32]. The simulation was performed under a single GW located at the densely populated area in combination with multiple buildings of some random heights and sizes. EDs are distributed uniformly on each floor in the building within a coverage range of 2500 m. The lowest SFs are assigned according to SNR of ED packets, thus it reduces the ToA for each ED and minimizes the chances of interference. The SF control algorithm is presented by allocating SFs to interact with two types of collisions: (a) two packets with the same SF collide and (b) two packets with different SFs collide in [21]. Hoverer, it fails to provide a solution to the second type. The primary purpose of this research is to reduce PER, improve fairness, and the throughput between EDs. The algorithm sorts the EDs based on distance and path loss to form distinct groups, where each group uses a separate channel. EDs in each group get the same SF based on the distance. Then the sum of the received power and cumulative interference ratio (CIR) is computed. If CIR exceeds the highest received power then it passes the feasibility check. On the other hand, the lowest SF is assigned to each group if the CIR is lower than the threshold. The proposed scheme decreases the PER up to 42% overall. An SF allocation scheme for massive LoRaWAN network aims to enhance the success ratio by considering the interference among the same SFs and channel [33]. To identify the interference caused by the collision of two packets, it determines the collision overlap time between the packets of the same SFs over the same channel. Then the SIR and received power are computed. If it exceeds the threshold, then the packets survived from interference. Otherwise, the packets are lost due to interference. However, it ignores any interference occurred due to the different SFs over the same channel, because these SFs are not perfectly orthogonal.

[34] SGS7P626

Some authors deployed LoRa networks and experimentally studied its performance [35] [**petajajarvi__coverage__2015** [36] [37] [38]. The measurements were done in city centers, tactical troop tracking, and sailing monitoring systems. Nevertheless, experimental results in real life networks are not reproducible and MAC layer optimization is difficult.

[39] 4KSQ7ABK

There are numerous LPWAN technologies emerging. LoRa, in particular, has attracted both research and industry interest because of its long range and robust performance. Existing research mostly focuses on LoRas performance, especially its transmission range, capacity, and scalability and on interaction between LoRa transmissions.

They include [7] [17] [LoRa from the city to the mountains: Exploration of hardware and environmental factors], where the authors evaluate LoRa performance under various set of configurations and conditions.

LoRa scalability is investigated in [19][18][26]. The authors in [19] analyze a LoRa network using a single gateway. Their results show that with an increase in the number of enddevices, the coverage probability drops exponentially, due to their interfering signals.

[21] 2ILCWX9Z

As described above, CSS enables decoding multiple messages with different spreading factors simultaneously. To decode simultaneous transmissions, power control is important because a threshold SNR needs to be guaranteed which is only possible when the received powers of all simultaneously transmitting nodes are of the same magnitude. Code Division Multiple Access (CDMA) is also a spread spectrum technique in which power control is a well investigated topic towards 3G cellular networks. Different algorithms exist: BER-based [Transmitter power control with adaptive safety margins based on duration outage], SNR-based [Power control in wireless networks: A survey] and RSSI-based [Location based power control for mobile devices in a cellular network]. In our scenario however, we cannot use the SNRor BER-based solutions, as they require fast feedback. In 3G networks, the update rate is 800Hz, while in LoRaWAN only one downlink message is available for each uplink message.

Finally, interesting research has been done concerning random access. [**dhillon__fundamentals__2014**] has shown the limits for random access networks with respect to retransmission probabilities and optimization of throughput given some failure constraints. This paper is different in the sense that the goal of our optimization is not throughput but packet error rate fairness.

[40] MWBMBZYE

Many work has been done in estimating the performance of LoRA networks, such as [26] [17] [20]. Although their conclusions are interesting, they only use a simpliied MAC protocol. This calls for a powerful network simulator that is useful to study the real network performance. Several simulation tools have been proposed for LoRaWAN. The most well-known LoRaWAN simulator is the LoRaSim built with python [23] [18]. It is open source and gives great insights in the LoRaWAN performance. However, LoRaSim does not implement acknowledgments. Thus, it cannot be used to study the network performance where nodes switch their spreading factor based on the feedback or absence of feedback from the gateway. Similarly, an Omnet++ implementation has been proposed in [Adaptive Coniguration of LoRa Networks for Dense IoT Deployments]. It implements an Automatic Data Rate (ADR) scheme where nodes can update their spreading factor and power at runtime. For ns-3, two diferent modules have been proposed in literature.

Independently from the previous implementation, the authors in [41] have proposed their solution. Their proposal also supports multiple gateways and overlapping networks. However, they did not include MAC commands. With their solution, interfering networks are possible as they accept interference from any network working on the same channel and frequency. Also in this implementation, the network is not connected to the IP layer, but directly to gateways. Compared to the above models, our implementation is totally compliant with the LoRaWAN v1.0 class A speciication. It is highly conigurable. Its lexible backbone architecture allows for easy integration of new protocols. Our model supports distributed gateways that are connected over an IP network to the network server that controls the whole network. We also provide base classes for the easy implementation of new applications on the network server and new MAC commands. With this model, we have investigated many aspects of LoRa networks, such as the efect of diferent spreading factors [21], the efect of interference [42], the reliability and scalability [43], etc. Our model can also be used to study the efect of downlink messages [44] and multiple gateways [41] [18].

[45] P3CSS7S2

LoRa and LoRaWAN technologies are relatively recent standards [17]. Most existing research based on LoRa and LoRaWAN has focused on features such as delay, range, throughput and network

capacity [7] [17][46][47]. Since the LoRa modulation is deployed for sensor applications, several papers evaluated this new technology with respect to its energy consumption. Driven by the challenges of energy consumption of wireless sensor applications, many recent works have focused on the power dissipation of communicating sensors.

To save power, Mare S. et al. have concluded that the communication module and the microcontroller must be in idle state as long as possible when they are not active. This work proposes interesting results, but LoRa and LoRaWAN technologies are not integrated into this study.

[48]Z38EPLZL

Understanding the limitations of LoRa technology is critical to the design and management of LoRa networks.

Elkhodr et al.(2016) reviewed various IoT communication technologies, including ZigBee, 6LoWPAN, Bluetooth Low Energy, LoRa and Wi-Fi. The capabilities and behaviours of these technologies were analyzed.

Khutsoane et al. (2017) surveyed a number of various LoRa technologies applications and contended that LoRa was ideal for low-power, long-range communications where low data rates were acceptable. However, there had not been a common, comprehensive or holistic strategy for IoT network design, development and management from Elkhordr et al. and Khutsoane et al.

Our study fills this gap by proposing [48], particularly, using practical measurements of LoRa network dependencies and performance metrics to support our proposal.

Memos et al. (2017) focused on the security and privacy issues on IoT network and proposed a security scheme to protect routing in IoT networks. Their study contribution is on a design of an algorithm for surveillance systems used in Smart Cities. Mesh networks may increase the coverage areas; however, forwarding traffic to other devices through multi-hop communications increases transmission latency and routing traffic, as stated by Filho et al. (2016). Our study focused on LoRa networks which are based on a star topology with single-hop communication and no routing complexity. In particular, security complication is also alleviated and it has better throughput compared to mesh networks.

[49] WQ72KWJM

Performance evaluation over LoRa networks has been intensively reviewed by many research studies in the literature [36] [38] [Experimental performance evaluation of lorawan: A case study in bangkok]. Other research studies focused on evaluating LoRa scalability [50] while considering co-SF interference that comes from collisions when using the same SF configuration on the same channel [19] whereas others assumed that SFs on a channel are perfectly orthogonal [23] [**bor_lora_nodate**]. SF represents the ratio between the chirp rate and the data symbol rate and affects directly the data rate and the range that a LoRa device can reach away from a LoRaWAN gateway.

Moreover, co-SF directly impact communication reliability, reduces the packet delivery ratio (PDR) successfully decoded at the gateway [51] and limits the scalability of a LoRa network when increasing the number of devices [Lora throughput analysis with imperfect spreading factor orthogonality]. Therefore, the latter should be considered in any upcoming study related to SF configuration strategies and network deployments. Some study examples focused on finding the optimal transmitter parameter settings that satisfy performance requirements using a developed link probing regime [7].

The work in [52] introduced a slicing infrastructure for 5G mobile networking and summarized research efforts to enable end-to-end network slicing between 5G use cases.

Furthermore, authors in [53] and [**rezende_adaptive_2018-1**] adopted network slicing in LTE mobile wireless networks. The former proposed a dynamic resource reservation for machine-to-machine (M2M) communications whereas the latter present a slice optimizer component with a common objective in both papers to improve QoS in terms of delay and link reliability. In a 5G wearable network, the authors took advantage of slicing technology to enhance the network resource sharing and energy-efficient utilization [54]. Moreover in [Joint application admission control and network slicing in virtual sensor networks], the authors perform slicing in virtual wireless sensor networks to improve lease management of physical resources with multiple concurrent application providers. In [Network slicing for ultra-reliable low latency communication in industry 4.0 scenarios], authors focused on URLLC and proposed several slicing methods for URLLC scenarios which require strong latency and reliability guarantees. Nowadays, guaranteeing service requirements in LoRa wireless access network (LoRaWAN) with traffic slicing remain as open research issues [26]. Therefore, unlike the previous work, in this article network slicing is investigated in LoRa technology which, to the best of our knowledge, has not been treated before by the research community.

[55] EUXH6LEM

The scientific literature on LoRa, and LPWANs in general, is slowly expanding but most of the

papers are still related to the link-level evaluation of the technology. Tests using Sigfox, LoRaWAN, and pre-standard NB-IoT solutions, have been made on the field by several network operators. Some field trials have been carried out, to determine LoRa ranging performance, in free space conditions [**aref_free_2014-1**] and in more complex scenarios [22].

Different studies have investigated the use of LoRa technology in specific fields of application, as for example, sailing monitoring systems [38], tactical troops tracking systems [37], smart cities [56], etc. In contrast with these works, we address a large set of applications, properly categorizing them. Many details regarding the LoRa modulation and physical layer have been recently published in [revspace.nl/DecodingLora] and [Decoding LoRa: Realizing a modern LPWAN with SDR], where the Authors studied the output signal generated by commercial transceivers to understand how information is encoded and embedded in the chirp waveforms.

The interference problem has been addressed in [**bor_lora_nodate**], where the Authors study packet collisions applying a time offset between each other; in [19] and [57] the orthogonality of transmissions performed with different Spreading Factors, an important issue discussed in more detail in Section V, is studied mathematically. More precisely, in these articles the Authors analyze the architecture of the LoRa (de)modulator and determine the conditions for a capture to happen in the presence of two signals with different SF. We performed a similar analysis, but using an experimental approach.

The first papers about the system-level LoRa network capacity have been published very recently, most of them addressing the problem through simple mathematical approaches [20], [26]. In these works the limitations imposed by regulation on the utilization of the channel are taken into consideration as a major limit for the network capacity; although this is true when few continuously transmitting devices are considered, if the traffic generated is more sporadic and the number of devices is higher, this does not represent a problem. It is possible to show that among the use cases that we consider in this paper the channel utilization for a single device is always below 0.55%.

Moreover, with respect to [20] and [26], we determine

the capacity of a network considering the full LoRa protocol stack, the presence of concurrent transmissions and consequent collisions or captures, and realistic information on the physical layer obtained through experiments.

[58] ZD2JYZZS

Although the appearance and widespread of LoRaWAN are recent, a number of works have been published aiming at analyzing or evaluating its performance in different scenarios or proposing enhancements to the off-the-shelf version of LoRaWAN [**herrera-tapia_evaluating_2017**]. From a theoretical perspective, works in [19][20][**bankov_limits_2016**] analyzed the capacity of LoRaWAN in terms of scalability and node-throughput. All these works concluded that LoRaWAN systems should be carefully configured and dimensioned with the aim of hosting a great number of end-devices. Concretely, in [19], the negative impact of interferences within highly-populated LoRaWAN cells was studied. Authors found some issues related to the co-spreading sequence interference, which notably harms the scalability of LoRaWAN systems.

However, as stated in [20], LoRaWAN networks can be generally utilized for fairly dense deployments with relaxed latency or reliability requirements. Thus, in order to ensure the network scalability when the end-device population prominently grows, two measures can be taken: (i) the number of delivered packets per node per day should be reduced; or (ii) the number of gateways should be increased [**bankov_limits_2016**]. As mentioned above, some works have proposed enhancements to the original LoRaWAN features [59][60].

In turn, authors of [22] investigated the coverage of LoRaWAN in different environments by placing the end-device onboard a car and a boat. Interesting coverage ranges over 10 km were reached with a not excessive Packet Loss Rate (PLR). A more elaborated work was developed in [27], in which the same authors extended their measurements and evaluated the performance of the system under mobility conditions.

From a simulation perspective, the work in [**herrera-tapia_evaluating_2017**] focused on vehicular scenarios and examined the performance of LoRaWAN in vehicular opportunistic networks, showing better results in comparison with WiFi technology. As observed in the reviewed works, the coverage range and the performance of different LoRaWAN configurations were evaluated. However, these studies did not characterize the sampling points depending on their adversity against wireless transmissions. In this work, we evaluate the performance of LoRaWAN under three well-defined conditions, namely, urban, suburban, and rural scenarios. By using this methodology, we identify the most proper configuration for LoRaWAN PHY layer parameters in order to reach the best performance in each type of scenario.

Besides, a comparison of the attained experimental results with a theoretical propagation model is also presented. A similar approximation was considered in [22] but, in this work, the attained coverage range was compared with the predictions given by the simple Free-Space model. This model is known to be inadequate to predict path loss in complex scenarios like those with the presence of obstacles. For that reason, in the present work, we make use of the predictions provided by a network-planning tool employing the widely used Okumura–Hata propagation model over realistic topographic maps.

[61] H6QDESI3

Most of the research done on LoRa/LoRaWAN has focused on features such as coverage, robustness, capacity, scalability, delay and throughput [20] [22] [26] [17] [47] [27] [50] [46] [28] However, a characteristic such as energy consumption, which is crucial considering that many LoRa/LoRaWAN devices will not be grid-powered, has received limited attention. We next review the literature on LoRaWAN energy consumption. We first focus on current consumption details of LoRa/LoRaWAN devices reported in published works, and secondly we discuss the few existing models of LoRa/LoRaWAN energy consumption, node lifetime or energy cost of data delivery. Several works provide current consumption data of LoRa/LoRaWAN devices, obtained from a datasheet or by empirical means [28] [62] [**mahmoud_study_2016**] [LAMBS: Light and Motion Based Safety] [7] [omparación de Soluciones Basadas en LPWAN e IEEE 802.15.4 Para Aplicaciones de Salud Móvil ("m-Health")] [63] [Design and implementation of the plug&play enabled flexible modular wireless sensor and actuator network platform] [64] [65] [66] Such details, which are summarized in Table 1, correspond to sleep, transmission and reception device states. As it can be seen, sleep current ranges from 7.66 A to 34 mA (or between 30.9 A and 3.4 mA excluding LoRa-only and custom devices). Sleep current for the considered hardware platforms is up to several orders of magnitude greater than that of their transceivers (see Table 2), which can be near or even lower than 1 A. One important conclusion is that current LoRa/LoRaWAN nodes are far from the degree of optimization exhibited by platforms that use other low power technologies. For example, IEEE 802.15.4 and Bluetooth Low Energy (BLE) commercial devices feature a sleep current near 1 A [AN079-Measuring Power Consumption of CC2530 with Z-Stack][**aguilar_opportunistic_2017**]. Therefore, in order to achieve attractive node lifetime figures (e.g., in the order of years), current LoRaWAN nodes need batteries with greater capacity than typical button cell batteries, e.g., of AA type, which however have bigger size and are more expensive. We attribute the sleep current in LoRaWAN devices to suboptimal hardware integration of device components, e.g., the microcontroller and the transceiver. Based on the characterization of sleep, transmit and receive states of a LoRa/LoRaWAN device, a few analytical models of LoRa/LoRaWAN energy consumption, node lifetime or energy cost of data delivery have been published [62] [66] [67] [68]. However, these models are too simple, since there exist several other states for a LoRa/LoRaWAN device involved in a communication that need to be considered (see Section 4). Next, we briefly present the main results and other limitations of these works. An accurate calculation of message transmission time is only provided in [67], however the study only focuses on LoRa, and therefore it does not model the MAC layer mechanisms defined in LoRaWAN, such as use of acknowledgments, receive windows, and retransmissions (see Section 3).

[69] LRLSSXZL

[70] QTLET7Q6

A real platform to test capture in LoRa networks is used in [23]. The conclusions drawn from the tests lead to a collision model close to that derived in [46] and to similar scalability evaluation results. We recall the collision rules established in [46] and [23] in Section 3.2 since we integrate them in our stochastic-geometric LoRa network model.

[51] D35WN7JM

Since LoRa is quite a recent technology, relatively few works have already been published on its performance.

Specifically, [71]focus on LoRa applications and PHY, while in [17] some test-bed and simulation results are presented but with a low number of devices.

[72] 9AZ7VKCG

Since LoRa has so many transmission parameters to confgure, the crucial task of fnding a parameter combination to balance packet delivery performance and energy consumption can be difcult. The pTunes framework is a general modelling framework for selecting optimal MAC parameters based on measurements [73]. We propose a similar approach for selecting LoRa parameters. Several

studies have studied the capability of LoRa technology measuring performance for diferent parameter settings in indoor

[7] [23] [**hutchison_data-aware_2013**] [74] [75] [LoRa from the city to the mountains: Exploration of hardware and environmental factors] [LoRaWANTM Specification] [76] [22] [28]

or outdoor [17][LoRa from the city to the mountains: Exploration of hardware and environmental factors][76] settings.

Bor and Roedig propose an algorithm for fnding the best transmission setting for a specifc transmission channel [7]. It performs a type of binary search of the parameter space, testing each setting for its packet reception rate until a good setting is found. The aim is to balance the cost of fnding good parameters against the packet delivery rate achieved. The ground truth for optimal settings are determined from a large look up table of receive probabilities based on in situ experiments with all combinations of the transmission parameters.

Cattani considered optimal parameter settings by measuring the packet reception rate and energy efciency for three types of channel (indoor, outdoor and underground) under diferent LoRa parameter settings [75]. An interesting fnding was that it was not worth tuning LoRa parameters to reduce the data rate in order to maximise the probability of successful reception. Instead lower energy settings which have a high data rate are preferred. They also considered the efect of environmental parameters on channel performance and found that high temperature at the node reduced packet delivery rate signifcantly. LoRaWAN is a mesh protocol for LoRa nodes. It specifes message scheduling and supports an Adaptive Data Rate (ADR) protocol for adjusting LoRa transmission parameters [LoRaWANTM Specification]. Our paper uses LoRa physical layer without the LoRaWAN protocols, but some results from LoRaWAN papers also apply in our case. LoRaWAN nodes start with a default parameter setting and then after reception of some messages the receiver can instruct a transmitter node to step up or down its spreading factor or transmission power. ADR uses 8 data rate settings and 6 transmission energy settings selected for a balance between packet delivery success and energy saving.

[77] ZC5I8BLQ

LoRa is a physical layer radio modulation technique based on chirp spread spectrum (CSS). The goal is to enable low throughput communication across long distances with low power consumption. LoRa features include long range, multipath resistance, robustness, low power consumption, forward error correction (FEC), and Doppler resistance. LoRa provides several physical layer parameters that can be customized. These parameters include spreading factor (SF), bandwidth (BW), transmission power (TP), and code rate (CR). These parameters affect the available bit rate, resilience against interference, and ease of decoding. LoRa uses seven different SFs, namely: [SF 6 , SF 7 , SF 8 , SF 9 , SF 10 , SF 11 , SF 12]. In LoRa a transceiver can select a BW in the range [-7.8, 500] kHz. However, LoRa transceivers typically operate at 125 KHz, 250 KHz, or 500 KHz. LoRa defines four different coding rates, 45 , 46 , 47 , and 48 . Higher CR implies higher protection against burst interference, and vice versa. LoRaWAN [LoRaWAN Specifications] is a MAC layer protocol and network architecture designed to be used with the LoRa physical layer. LoRaWAN uses pure Aloha

(PA) as a channel access protocol. A LoRaWAN gateway can decode eight simultaneous transmissions based on different combinations of SFs and BWs, however at any given time a node in a LoRaWAN network uses particular combination of SF, BW, TP, and CR.

LoRa throughput is analyzed in [26], [23], [46], and [20], which have primarily focused on Class A LoRaWAN devices. It has been shown that although LoRaWAN uses PA channel access control protocol, due to LoRa's robust modulation technique an increase of up to 1000 nodes per gateway results in only 32% more packet losses, whereas for the same scenario the losses are up to 90% in other PA-based networks [46]. As LoRa allows customization of transmission parameters, therefore recently some research efforts focused on devising algorithms for effective LoRa's transmission parameter selection by considering a specific goal.

[78] D6JC9B4S

As LoRa is considerably new technology, only limited work has been done in the area of battery optimization.

LoRa is used for obtaining battery status in moving Electric Vehicles in [An efficient electric vehicle charging architecture based on lora communication], tracking system for moving vehicles in [Design and implementation of object tracking system based on lora]–[Efficient, real-time tracking of public transport, using lorawan and rf transceivers], [Long-range wireless sensor networks for geolocation tracking: Design and evaluation], and human monitoring in [We-safe: A wearable iot sensor node for safety applications via lora] and [The experimental trial of lora system for tracking and monitoring patient with mental disorder]. However, none of the aforementioned works account for energy optimization in transmission settings for moving nodes. All of them use conventional method of using a fixed preconfigured setting for transmissions throughout the movement. This conventional mechanism is highly energy inefficient as it uses high power settings even when the node is close to a gateway. Besides, battery optimization techniques [Performance-aware energy optimization on mobile

devices in cellular network], [etime:Energy-efficient transmission between cloud and mobile devices] for moving nodes used in other wireless communications such as LTE, Wi-Fi, 5G, require frequent exchanges of information about channel conditions, previous communication history, user traffic, etc. However, such methods become inapplicable for LoRa due to its duty-cycle restriction, in which each LoRa node is allowed to only send a few messages a day to avoid interference.

[79] CNASAVPC

Wireless transmission parameters in the unlicensed spectra are regulated regionally with occasional additions by countrylevel regulators. Regulations typically include, but are not limited to, definition of maximal effective radiated power (ERP) and duty-cycle, and medium access methods.

[80] RJ28KVZT

The Internet-of-Things (IoT) is pushing a paradigm shift in the design of connectivity solutions for smart devices. Nowadays, the community widely accepts that current connectivity solutions like WiFi, Bluetooth, and ZigBee alone cannot cope with the billions of devices expected to integrate the IoT in the forthcoming years [Long-range commun. in unlicensed bands: the rising stars in the IoT and smart city scenarios]. The IoT is emerging as a solution to integrate different communication technologies, each focusing on the requirements of specific applications. The so-called Massive IoT (mIoT) figures within this context as a network scenario with thousands of connected devices running noncritical, low-power, and low-cost applications tolerant to high latency and small data-rates [Cellular netw.for massive IoT enabling low power wide area appl]. New communication technology must address such peculiar scenario, and this is the case of Low-Power Wide-Area Network (LPWAN) technologies like LoRa WAN, S IG F OX , NB-I O T, and RPMA [Understanding the IoT connectivity landscape: a contemporary M2M radio technology roadmap]. This paper concerns the assessment of the uplink channel of the LoRa (Long-Range) technology, which forms the physical layer (PHY) of the LoRa WAN protocol stack [lora-alliance.org]. Although LoRa is in fast-paced adoption, reports on deployments with large numbers of stations are yet to come out, making their performance and capacity models still an open problem. Recent related work has sought to assess the performance of LoRa networks using both analytic modelings [Do LoRa low-power wide-area netw. scale?]–[19][44] and real measurements [Performance of a low-power wide-area netw. based on LoRa technol.: Doppler robustness, scalability, and coverage] [81] [63] [82] [83] [Long range commun. in urban and rural environ]. Analytic models have been proposed for a variety of scenarios and communication phenomena.

There are similar measurement reports in other environments, including a university campus [81], indoor applications [63], industry [82], dense cities downtown [83], and rural areas [84]. These measurements show interesting results, however, none of them used a large numbers of nodes and, thus, it is impossible to use the results to validate dense network models. As reports on the performance of LoRa and other LPWAN started to reveal the limitations of such networks, a few techniques were proposed to enhance the performance of LoRa [16] [7][85] and other LPWAN technologies [86][87][24].

Bor and Roedig [7] explore LoRa configuration

[88] 3B3UUVC9

Some of the LPWAN offerings are mainly proprietary but the LoRa Alliance develops LoRaWAN as an open standard. The physical layer (LoRa) was however developed by Semtech which remain the sole LoRa integrated circuit producer [**bankov_limits_2016**]. LoRaWAN is the communication protocol (ALOHA-based) [A technical overview of LoRa and LoRaWAN] and system architecture for a network using the LoRa physical layer [36]. LoRaWAN is not the only communication protocol that uses LoRa as the physical layer: Symphony Link TM and LoRaBlink are other examples. LoRaBlink [**bor_lora_nodate**] adds multi-hop support while Symphony Link offers guaranteed Acknowledgements (ACKs), over the air firmware updates and many other features. The DASH7 stack can also be configured to use LoRa as its physical layer and can potentially run side-by-side with a LoRaWAN stack [DASH7 Specification-DRAFT 16-An Advanced Communication System for Wide-Area Low Power Wireless Applications and Active RFID]. A. Physical layer (LoRa) LoRa is a derivative of Chirp Spread Spectrum (CSS) modulation with integrated Forward Error Correction (FEC) [89]. LoRa uses sub one GHz ISM bands in Europe and North America and its wide band nature allows LoRa to better compensate for a low Signal to Noise Ratio (SNR) [LoRa Modulation Basics,]. This allows LoRa to demodulate signals even when they are 19.5 dB below the noise floor [23]. CSS allows for a longer communication range than Frequency-Shift Keying (FSK) without an increase in power consumption [A technical overview of LoRa and LoRaWAN].

Transmitting at higher power levels will increase a LoRa node's range. Nodes can adjust their output power to meet regulatory requirements. In Europe +14 dBm is the maximum transmit power except in the G3 band (+27 dBm) [A technical overview of LoRa and LoRaWAN]. LoRaWANs deployed in Europe have channel bandwidths of either 125 kHz or 250 kHz and a single FSK transmission

channel providing a higher data rate is also available [36]. Data rates are region (regulatory restrictions) as well as Spreading Factor (SF) dependent. Increasing the spreading factor improves the SNR but results in longer transmission times [**bor__lora__nodate**]. Using a higher bandwidth shortens the transmission times but reduces the maximum receiver sensitivity [23]. Capacity calculations performed in [20] revealed that when a single gateway must serve many devices the majority of them should be close to the gateway (SF = 7) as only a few nodes with the maximum SF can be supported given their long transmission times. LoRa uses FEC to allow the recovery from transmission errors due to bursts of interference, but the use of FEC adds some encoding overhead [SX1272/3/6/7/8: LoRa Modem Designer's Guide,]. LoRa's coding rates are 4/(CR + 4) with CR 1, 2, 3, 4. A LoRa packet's header and its Cyclic Redundancy Check (CRC) will always be transmitted using a CR of 4/8 and the payload with its optional CRC at the chosen coding rate [SX1272/3/6/7/8: LoRa Modem Designer's Guide]. When LoRa is transmitting with a BW of 125 kHz and a SF of 11 or 12 a low data rate optimization can be enabled. This reduces the impact on transmission due to drift in the reference frequency of the oscillator, but does add additional data overhead [SX1272/3/6/7/8: LoRa Modem Designer's Guide,]. LoRa can detect channel activity using Carrier Activity Detection (CAD) [**bor__lora__nodate**]. This is faster than Received Signal Strength Indicator (RSSI) identification and can differentiate between noise or a desired LoRa signal [LoRa Modulation Basics]. B. LoRaWAN LoRaWANs in Europe are limited to 10 channels, has duty cycle restrictions but no channel dwell time limitations. LoRaWANs in North America have 64 channels, also have duty cycle restrictions but no channel dwell time limitations [A technical overview of LoRa and LoRaWAN]. LoRaWAN has 3 common 125 kHz channels for the 868 MHz band namely 868.10, 868.30 and 868.50 MHz that devices use to join the network [Indoor Deployment of LowPower Wide Area Networks (LPWAN): a LoRaWAN case study]. Once a node has joined the network, the network server can provide additional channels to the device. In Europe, the same channels are used for uplink and downlink. The network architecture is a star of stars topology in which end nodes connect directly communicate with gateways which in turn connect to a central network server [**bankov__limits__2016**]. Gateways are always on devices and have LoRa capabilities and potentially Ethernet or cellular capabilities to connect to the network server. In a LoRaWAN transmissions are received by any nearby gateway(s). This allows mobile nodes to transmit to any gateway without any handover. The network server drops any copies of a message and replies using the optimum gateway [A technical overview of LoRa and LoRaWAN], [36].

[90] H6DXYYB9

Even though several articles studied the scalability of LoRa networks [Long-range communications in unlicensed bands: The rising stars in the IoT and smart city scenarios][22][17], none of them has considered the impact of ADR on performance.

varsier__capacity__2017 analyzed the capacity limits of LoRaWAN networks for smart metering applications. The authors considered a distribution of spreading factors based on the median of the SNR values received at the gateway. However, the exact details on how to configure the transmission parameters were not provided in their work. In contrast, we have evaluated the impact of ADR on network performance and proposed modifications to the original ADR algorithms.

[91] 253585LX

Other studies in the literature analyzed the performance of the LoRa modulation—Goursaud and Gorce [57] considered other technologies (SigFox, Weightless, and RPMA by Ingenu) in addition to LoRa to highlight their pros and cons.

[92] D8LVPLDY

[29] S7MJTV2C

Here we provide an overview of the LoRaWAN protocol stack and highlight related work in the LoRaWAN domain. A. Long Range (LoRa) LoRa is a proprietary low-cost implementation of Chirp Spread Spectrum (CSS) modulation by Semtech that provides long range wireless communication with low power characteristics [An1200.22, lora modulation basics] and represents the physical layer of the LoRaWAN stack. CSS uses wideband linear frequency modulated pulses, called chirps to encode symbols. A LoRa symbol covers the entire bandwidth, making the modulation robust to channel noise and insensitive to frequency shifts. LoRa modulation

is defined by two main parameters: Spreading Factor sf SF s(7, ..., 12), which affects the number of bits encoded per symbol, and Bandwidth bw BW s(125, 250, 500)KHz, which is the spectrum occupied by a symbol. A LoRa symbol consists of 2 sf chirps in which chirp rate equals bandwidth. LoRa supports forward error correction code rates cr equal to 4/(4 + n) where n ranges from 1 to 4 to increase resilience. The theoretical bit rate R b of LoRa is shown in Eq. 1 [An1200.22, lora modulation basics]. bw ∗ cr bits/s (1) 2 sf Moreover, a LoRa transceiver allows adjusting the Transmission Power T P . Due to hardware limitations the adjustment range is limited from 2dBm to 14dBm in 1dB

steps. A LoRa packet can be transmitted using a constant combination of SF , BW , CR and T P , resulting in over 936 possible combinations. Tuning these parameters has a direct effect on the bit rate and hence the airtime, affecting reliability and energy consumption. Each increase in SF nearly halves the bit rate and doubles the airtime and energy consumption but enhances the link reliability as it slows the transmission. Whereas each increase in the BW doubles the bit rate and halves the airtime and energy consumption but reduces the link reliability as it adds more noise. The airtime of a LoRa packet can be precisely calculated by the LoRa airtime calculator [Lora modem design guide]. Fig. 1a shows the effect of SF s and BW s at code rate CR = 4/5 on the airtime to transmit an 80 bytes packet length. As shown, the fastest combination uses the lowest SF with the highest BW , whereas, the highest SF with the lowest BW achieves the slowest combination. Fig. 1b shows the energy consumption for combinations of SF s and T P s at CR = 4/5 and BW = 500KHz to transmit an 80 bytes packet. As shown, the SF has much higher impact than the T P on the energy consumption, e.g. increasing SF consumes more energy than increasing T P especially for large SF s. LoRa modulation can enable concurrent transmissions, exploiting the pseudo-orthogonality of SF s as long as none of the simultaneous transmissions is received with significantly higher power than the others [57]. Otherwise, the strongest transmission suppresses weaker transmissions if the power difference is higher than the Co-channel Interference Rejection (CIR) of weaker SF s. In case of the same SF , all simultaneous transmissions are lost, unless one of the transmissions is received with higher power than the CIR of the SF . This suppression of weaker signals by the strongest signal is called capture effect [23]. The CIR of all SF pairs has been calculated using simulations in [57] and validated by real LoRa link measurements in [51].

Recent research on LoRa/LoRaWAN has mainly focused on LoRa performance evaluation in terms of coverage, capacity, scalability and lifetime. The studies have been carried out using real deployments in [84] and [27], mathematical models in [Mathematical model of lorawan channel access] and [19], or computer simulations in [23] and [24]. Almost all these works have assumed perfectly orthogonal SF s although it has been shown in [50] and [51] that this is not a valid assumption. Furthermore, recent work has proposed transmission parameter allocation approaches for LoRaWAN with different objectives. For example, authors in [7] proposed a transmission parameter selection approach for LoRa to achieve low energy consumption at a specific link reliability. Here a LoRa node probes a link using a transmission parameter combination to determine the link reliability. It then chooses the next probe combination based on whether the new combination achieves lower energy consumption while maintaining at least the same link reliability. Finally, the approach terminates when reaching the optimal combination from an energy consumption perspective.

The two aforementioned works [7] and [16] assumed perfectly orthogonal SF s, which leads to a higher overall data rate than in reality. In the context of our work presented here, allocating data rates and T P s to achieve data rate fairness in LoRaWAN is not well investigated, with the exception of [21], where authors proposed a power and spreading factor control approach to achieve fairness within a LoRaWAN cell. We provide an overview of [21] and a detailed comparison with our proposal in Section IV. While in general data rate and power control approaches have been well studied for cellular systems and WiFi [A framework for uplink power control in cellular radio systems] [**subramanian_joint_2005**], we argue that these solutions are not suitable for constrained systems like LoRaWAN. The reason is that cellular based approaches require fast feedback and high data rates to work, which are not available in LoRaWAN. In the end, an interesting work was done to ensure an interoperability between LoRaWAN and the native IoT stack i.e. IPv6/UDP/CoAP at the device level. The interoperability was done by adopting legacy solution like 6LoWPAN over LoRaWAN [60] or by developing a new header compression technique to be more suitable for the constraints of LoRaWAN [Lschc: Layered static context header compression for lpwans].

[41] J22SN85R

A number of works have been published in literature that study the scalability of LoRa(WAN) LPWA networks.

[93] P8FP2R7R

In [Long-range communications in unlicensed bands: the rising stars in the iot and smart city scenarios], Centenaro et al. provide an overview of the LPWAN paradigm in the context of smart-city scenarios. The authors also test the coverage of a LoRaWAN gateway in a city in

Italy, by using a single base-station without antenna gain. The covered area had a diameter of 1.2 km.

[94] S33CL98I

Several recent related works have sought to evaluate the performance of LoRa networks using analytic modeling [19]– [23] [32] [44] [95] and real measurements [27] [28] [81] [63] [82] [83] [96] [97]

[84]. Additionally, a few techniques have been proposed to enhance the performance of LPWANs in general, with potential modifications to the current technologies, as for LoRa in [16] [7] [85] [18], UNB/S IG F OX in [86], and for others in [87], [24]. Analytic models have been proposed for a variety of scenarios and communication phenomena.

Concerning the modeling of communication fading, only Georgiou and Raza [19] and Pop et al. [44] take this impairment into account, to the best of our knowledge. Several works have used measurements to evaluate the performance of LoRa networks.

Petäjäjärvi et al. [27], [28] analyze Doppler robustness, scalability, and coverage of LoRa networks and report the experimental validation of such metrics in terrestrial and water environments for static and mobile nodes. Considering a delivery ratio of at least 60% and LoRa most conservative configurations, they were able to communicate to static nodes ranging up to 30 km over the water and up to 10 km on the ground. Regarding Doppler robustness, they observed that communication degrades significantly when the velocity of the node in relation to the gateway is above 40 km/h. There are similar reports of LoRa measurements done in different environments, including a university campus [81], indoor applications [Indoor deployment of lowpower wide area networks (LPWAN): A LoRaWAN case study], industry [82], dense cities downtown [83], [Usability of LoRaWAN technol], smart metering [97], and rural areas [84]. Albeit these measurements show interesting results, it is important to note that none of them used a large number of network nodes, thus making it difficult to validate models for dense networks. A few recently published work also propose some enhancements to LoRa .

In this paper, we model and validate the behavior of LoRa networks using message replication to exploit time diversity and using a single gateway with multiple receive antennas to exploit spatial diversity, striving to maximize network performance. To do that, we take the work of [19] as baseline model and extend it to incorporate the proposed techniques. Our work on message replication differs from [86] because that work considers UNB networks where each transmission uses a random central frequency – an assumption that changes the collision model. Moreover, our work takes fading into account, what [86] does not. Our approach using multiple receive antennas differs from [87] and [24] because they consider spatial diversity generated by multiple gateways. Our work examines the case where multiple receive antennas in a single gateway create signal diversity able to enhance signal quality, an approach that can be naturally extended to the case of multiple gateways in the future. To the best of our knowledge, no work has investigated the use of multiple receive antennas and message replications in LoRa networks.

[98] DFN3W8GF

Noreen, Bounceur, and Clavier [99] described LoRa PHY performance theoretically. They explained the transmission rate in terms of three basic parameters: BW, CR, and SF. Their results show that increasing the packet length results in a sharp increase in ToA. SK Telecom [sktiot.com], a leading telecommunication operator in Korea, presented its experimental results on LoRa transmission ranges at the IoT-LPWA working conference in 2016. LoRa nodes that are located outdoors transmit packets to a LoRa gateway with an output power of 14 dBm. In the experiment, the SF was set to 12, and retransmission was not conducted. The company announced that transmission ranges of 1.09 km, 1.54 km, and 3.03 km are achievable for dense urban, urban, and suburban areas, respectively. These specified values satisfy the requirement for a LoRa transmission success rate of above 90%. The researchers also deployed nodes inside the buildings on the first floor and performed a similar test. In this case, the communication coverage was measured to be about 2/3 of the performance measured outdoors.

[100] EKZHXVKL

[101] AAHXK9LU

Multiple works exist for the performance evaluation of LPWANs [20], [102], [26], [42]. Most of these works use Poisson processes to model the packet arrival, which we believe is not the most adapted for periodic packet sending scenarios in LPWANs. For example, a device reporting on a daily basis would not send more than one message per day. However, as Poisson models the intensity of packet arrival, an intensity of one message per day represents in fact the mean value, i.e., one message on average per day, which is not quite the case described.

[103] J8NXUG9T

In the last years, the LoRaWAN technology has been the subject of many studies, which analyzed its performance and features with empirical measurements, mathematical analysis and simulative tools.

Some seminal papers on LoRaWAN such as [22], [36] test the coverage range and packet loss ratio by means of empirical measurements, but without investigating the impact of the parameters setting

on the performance.

The authors in [79], [90] target the original ADR algorithm proposed by [thethingsnetwork.org], suggesting possible ameliorations. Generally, the modified algorithms yield an increase of network scalability, fairness among nodes, packet delivery ratio and robustness to variable channel conditions.

In [**zucchetto_uncoordinated_2017**], the authors investigate, via simulation, the impact of DC restrictions in LPWAN scenarios, showing that rate adaptation capabilities are indeed pivotal to maintain reasonable level of performance when the coverage range and the cell load increase. However, the effect of other parameters setting on the network performance is not considered. In this study we differ from the existing literature in that we target large networks with bidirectional traffic, a scenario that makes it possible to observe some unforeseen effects rising from the interaction of multiple nodes served by one single GW and NS. Furthermore, in our analysis we examine one by one the role played by the configurable network parameters, as detailed in Sec.IV-A, thus highlighting some pitfalls that can affect the network performance. We then propose possible counteractions that require some small changes at the MAC layer, and we evaluate their effectiveness in some representative scenarios. As a side result, we enriched the ns-3 lorawan module with new functionalities.

[104] AXRU35IH

Performance evaluations of the LoRaWAN protocol frequently consist of a network with a single gateway and one or two nodes with which measurements are taken at several identified points [88], [35], [22], [17]. These provide valuable insights but can produce results impacted by device specific characteristics. Experiments on nodes in motion showed that at speeds higher than 40 km/h, the communication performance worsens due to the Doppler effect [27], [58]. Extensive research regarding the ADR scheme has resulted in additions and modifications targeting network performance metrics such as scalability [105], throughput [16], PDR [21], and contention [106]. As an example, congestion estimation is achieved through evaluation of network throughput, RSSI and the number of connections at a gateway before nodes are sent LinkADRReq messages [105]. Fair Adaptive Data Rate (FADR) uses Received Signal Strength Indicator (RSSI) values in its calculations when determining SF and Transmit Power (Tx) assignments [Poster: A Fair Adaptive Data Rate Algorithm for LoRaWAN].

In [58], the influence of variation in payload length was tested and a definite PDR improvement was observed, which was however not consistent over the range of data rates evaluated. The payload length experiments conducted in [35] found similar inconsistencies, with similar PDRs for 10 and 100 bytes but a decrease for 50 bytes. Performance evaluations in urban, suburban and rural environments resulted in coverage of around 6 km in urban and suburban areas with over 18 km in the rural scenario [58]. The urban evaluation, which enabled ACKs, showed a PDR of 100 % for DR0 to DR5 for distances below 3 km, although over how many packets this was calculated was not specified. Even at distances between 5 km and 6 km, a 100 % PDR achieved when DR0 was used, however, other data rates resulted in lower PDRs of between 30 % and 50 %. Tests on ACK requests by nodes in an evaluation of a three gateway LoRaWAN, found that in 2.5 % of cases the data arrived but the device did not receive an ACK which could result in unnecessary retries [36]. To investigate the impact of downlink traffic in which ACKs materially influence performance, the popular LoRaSim simulator was extended into LoRaWANSim [44]. Evaluation of the effects of increased network size showed that a gateway will reach its duty cycle limits when attempting to transmit all of the required ACKs. The use of ACKs has a major impact on performance in large networks and greatly reduce their capacity [44]. Tests on a single gateway network found that ACKs only improved the PDR for a low number of devices (100, 500 and 1000) and only when data was sent every sixty thousand seconds [41].

[44] TVTNCP95

As we present a simulator for LoRaWAN LPWA networks, we briefly discuss related tools and studies on LoRaWAN. A. LoRaWAN Analytical Models and Simulators Multiple analytical models [Analysis of the delay of confirmed downlink frames in Class B of LoRaWAN], [19], [68] and simulators [23], [Massive Access for Machine-to-Machine Communication in Cellular Networks], [github.com/maartenweyn/lpwansimulation] have been proposed to understand the performance of LoRaWAN. None of these models is provide any insights on the interplay between downlink traffic and the gateway's duty cycle limit or effect of MAC parameter settings on the reliability of LoRaWAN. To bridge this gap, we design LoRaWANSim, which extends the functionalities of LoRaSim [23], an existing discrete event simulator. Other simulators [Massive Access for Machine-to-Machine Communication in Cellular Networks], [github.com/maartenweyn/lpwansimulation] including LoRaSim focus more on LoRa physical layer aspects, including modulation, channel effects, and path loss. Unfortunately, their MAC layer capabilities are very much limited to an implementation of the ALOHA protocol. With LoRaWANSim, we take an important step forward by incorporating multiple MAC layer features that are part of the LoRaWAN standard. These features include the possibility to send downlink traffic, special control

messages, confirmed messages, acknowledgments, and retransmissions. By doing so, LoRaWANSim enables users to evaluate the performance of the LoRaWAN MAC layer, derive useful insights about the effect of several MAC layer parameters, and evaluate possible enhancements to the LoRaWAN standard.

[107] PVJDK4XI

Long range technologies such as LoRa [lora-alliance.org] and Sigfox [sigfox.com] have started to draw significant attention from the academic and industrial communities. Some of the published works in this field devote their efforts to analyzing the performance of real LPWAN deployments under different conditions: IoT devices monitoring civil infrastructures such as bridges [Vibrations powered LoRa sensor: An electromechanical energy harvester working on a real bridge], LoRa-based video surveillance systems [Deploying a pool of long-range wireless image sensor with shared activity time], health monitoring motes [28], etc. On the other hand, some other studies are focused on analyzing the advantages, disadvantages, capabilities, and limits of the current implementations of these technologies from a technological point of view. For example, the real scalability of current LoRa networks [Do LoRa Low-Power Wide-Area Networks Scale], [19], the performance of their different configurations [7], and how these types of networks tolerate download traffic [44], amongst other things are being studied. Although they are very practical and illustrating, none of these works optimizes or analyzes the performance of LPWAN in a generic and theoretic fashion, which would allow their extrapolation to different technologies (LoRa, Sigfox, etc.) or their future implementations, beyond current transceivers. As a notable exception, [Analysis of Latency and MAC-layer Performance for Class A LoRaWAN,] studied the impact of sub-band selection on LoRa motes by modeling nodes as an infinite, jockeying M/M/c queue (i.e. c servers, arrivals determined by a Poisson process, and exponentially distributed job services). Although the work is very well detailed, mathematically neat and applicable to future deployments, it does not capture the true, complex nature of real Long-Range networks, where resources are very scarce (i.e. infinite queues are impossible to implement) and traffic cannot always be assumed to follow a certain distribution. Regarding the TDC-limitation problem, two works [QoS for Long-Range Wireless Sensors Under Duty-Cycle Regulations with Shared Activity Time Usage], [Deploying a pool of long-range wireless image sensor with shared activity time] have recently highlighted the importance of TDC-aware networks by illustrating the problem of transmitting real-time video in Long-Range deployments. Although practical, the solution proposed focuses on deliberately breaking the 36s/hour TDC limitation by complying with it in a network-aggregated fashion (i.e. the average network TDC is kept below 36s/hour, not the per-node TDC). In fact, [26] highlighted that the effects of TDC limitations jeopardize the actual capacity of largescale deployments, and the only de-facto proposal to manage it, a fixed limit on the number of permitted messages per day, fails to provide the network with enough flexibility. With the interest of contributing to fill the notable gap in research, we propose an approach to derive MDP-based transmission policies that fully comply with the TDC regulations while maximizing the number of high-priority reported events.

[108] U5RX6JLY

LoRaWAN is a wireless communication protocol and has got more and more researches in recent years. Its application domains include smart agriculture, security city, river bank monitoring and street lighting control. However, LoRaWAN employs simple random access schemes that may suffer from important latency and low data delivery, which are key performance indicators in smart elderly care networks. Some researches focus on the features and performance of LoRaWAN.

Analytical models of energy consumption and the performance of LoRaWAN are analyzed by Casals et al. [61]. The impact of LoRa transmission parameter selection on communication performance is presented by Bor and Roedig [7], a link probing regime to save energy consumption and improve communication reliability is also developed.

Considering energy constraints of wireless sensor networks in IoT, Al-Turjman et al. [109] propose a novel traffic model to investigate the effects of multi-hop communication of IoT. Tunc and Akar [110] present a Markov fluid queue model for energy management to prolong the lifetime of IoT. For Markov model itself, a classical single server vacation model is generalized by **servi__m/m/1__2002** to consider a server in a WDM optical access network, they exploit the equilibrium joining strategies for customers in an M/M/1 queue with working vacations and vacation interruptions [**li__equilibrium__2016**]. Kempa and Kobielnik [111] consider a single-channel finite-buffer queuing model with a general independent input stream of customers. Besides the characteristics of LoRaWAN network researches, the applications in industry and IoT are also attracted some researchers' attention. A solution is proposed in article [112] aims to integrate LoRaWAN with 4G/5G mobile networks, which will allow the current infrastructures of mobile network operators are reutilized.

cruz__algorithm__2019 address the IoT utilization of the public transportation system in smart city.

The combined use of IoT with industrial sensors in structural health monitoring is given by **alonso__middleware__2018**.

[113] ZRKCKJY9

Transmission parameter configuration mechanisms such as ADR scheme need to be executed on both LoRa node and LoRa network server. Taking into account low power consumption, the mechanism running on LoRa node should be as simple as possible and has been detailed in LoRaWAN. However, LoRa network server is responsible for the complex management mechanism, which can be carefully designed to improve network performance. Therefore, the discussed related works focus on server-side mechanism. In addition, the mechanism running on LoRa node is in accordance with the definition of LoRaWAN 1.1 specification [LoRaWAN Specification]. The basic ADR scheme provided by LoRaWAN estimates channel conditions using the maximum value of the received signal-to-noise ratio (SNR) in several recent packets [LoRaWAN Specification]. When the variance of the channel is low, using ADR scheme significantly reduces the interference and increases the system capacity compared with using the static data rate [23], [90], [**zheng__smdp-based__2015**]. However, the scheme may also have potential drawbacks. First, SNR measurements are determined by different models of LoRa Gateway. Therefore, the value of SNR is inaccurate as a result of hardware calibration and interfering transmissions. Second, selecting the maximum SNR in the last 20 packets is not an desirable method. Because there may be a long time interval between consecutive packets for some IoT applications. The antiquated SNR information is not able to accurately estimate the channel condition for the next uplink packet. Third, the scheme only considers the link of single node to decide whether to adjust transmission parameters. If massive LoRa nodes are densely distributed near LoRa Gateway, it will cause most of nodes using the fastest data rate. With the number of LoRa nodes using the same data rate increases, the possibility of collisions also increases dramatically. Moreover, a lot of researchers propose various approaches to allocate transmission parameters with different objectives. Most of the approaches utilize SNR or RSSI information to control transmission power and spreading factor. The authors in [90] slightly modify the basic ADR scheme. The maximum operation in the SNR of several recent packets is replaced with the average function.

[114] IWAC4Y9B

LoRa, a proprietary wireless communication standard promoted by the LoRa alliance, enables long-range communications. Even though the typical topology in LoRa is a single-hop network named LoRaWAN [LoRaWAN specification], [44], the SF allocation is an important issue because it improves the network efficiency in both single-hop and multi-hop LoRa networks. A. SF ALLOCATION IN A SINGLE-HOP NETWORK LoRaWAN, which is a single-hop network, implements an ALOHA or a slotted ALOHA mechanism on the Medium Access Control (MAC) layer with the physical design of LoRa technology [115]. LoRaWAN ensures connectivity by standardizing the Adaptive Date Rate (ADR) mechanism to allow the node to step down its data rate. However, the ADR, which is based on the number of received acknowledgment (ACK) messages from gateways, is a basic method. These methods are inaccurate for assessing the highlyvarying wireless environment, and render data transmission inefficient [105]. Subsequent research [26] [21] [43] [90] considered an SF distribution scheme based only on the distance from the node to gateways.

3 | Industrial Survey

"Given one hour to save the planet, I would spend 59 minutes understanding the problem and one minute resolving it." - Albert Einstein

Contents

1 IoT applications requirements [bregell_hardware_2015]

1.1 Summary and discussion

Use cases	Re-sources	Mobil-ity	Hetero-geneity	Scalabil-ity	QoS	Data	standardiza-tion	Safety	Secu-rity
Logistics									
Health-care									
Smart environment									
Personal and social									
Futuristic									
Smart Mobility									
Smart semaphores control									
Smart Red Swarm									
Smart panels									
Smart bus scheduling									
Smart EV management									
Smart surface parking									
Smart signs									
Smart energy systems									
Smart lighting									
Smart water jet systems									
Smart gathering									
Smart building									
Smart tourism									
Smart QRinfo									
Smart monitoring									
Smart hawk-eye									
Health Monitoring									
Water Distribution									
Electricity Distribution									
Smart Buildings									
Intelligent Transportation									
Surveillance									
Environmental Monitoring									

Table 3.1. Main IoT challenges[**kouicem__internet__2018**] + [116] [**hancke__role__2012**] [**alba__intelligent__2016**]

2 IoT Wireless Networks (Norms & Standards)

Several different wireless communication protocols, such as Wireless LAN (WLAN), BLE, 6LoWPAN, and ZigBee may be suitable for IoT applications. They all operate in the 2.4GHz frequency band and this, together with the limited output power in this band, means that they all have a similar range. The main differences are located in the MAC, PHY, and network layer. WLAN is much too power hungry as seen in table 2.6 and is only listed as a reference for the comparisons.

2.1 SigFox

2.2 IETF

2.2.1 6LoWPAN

is a relatively new protocol that is maintained by the Internet Engineering Task Force (IETF) [7, 6]. The purpose of the protocol is to enable IPv6 traffic over a IEEE 802.15.4 network with as low overhead as possible; this is achieved by compressing the IPv6 and UDP header. A full size IPv6 + UDP header is 40+8 bytes which is tild 38% of a IEEE 802.15.4 frame, but with the header compression this overhead can be reduced to 7 bytes, thus reducing the overhead to tild 5%, as seen in figures 2.3 and 2.4.

2.3 3GPP

2.3.1 NB-IoT

2.3.2 EC-GSM

2.3.3 e-MTC

2.4 IEEE

2.4.1 IEEE 802.11

2.4.2 IEEE 802.15.4

At present days, there are several technology standards. Each one is designed for a specific need in the market. For the Wireless Sensor Networks, the aim is to transmit little information, in a small range, with a small power consumption and low cost. The IEEE 802.15.4 standard offers physical and media access control layers for low-cost, low-speed, low-power Wireless Personal Area Networks (WPANs)

A) Physical Layer The standard operates in 3 different frequency bands: - 16 channels in the 2.4GHz ISM band - 10 channels in the 915MHz ISM band - 1 channel in the European 868MHz band

B) Definitions Coordinator: A device that provides synchronization services through the transmission of beacons. PAN Coordinator: The central coordinator of the PAN. This device identifies its own network as well as its configurations. There is only one PAN Coordinator for each network. Full Function Device (FFD): A device that implements the complete protocol set, PAN coordinator capable , talks to any other device. This type of device is suitable for any topology. Reduced Function Device (RFD): A device with a reduced implementation of the protocol, cannot become a PAN Coordinator. This device is limited to leafs in some topologies.

C) Topologies Star topology: All nodes communicate via the central PAN coordinator , the leafs may be any combination of FFD and RFD devices. The PAN coordinator usually uses main power.

Peer to peer topology: Nodes can communicate via the central PAN coordinator and via additional point-to-point links . All devices are FFD to be able to communicate with each other.

Combined Topology: Star topology combined with peer-to-peer topology. Leafs connect to a network via coordinators (FFDs) . One of the coordinators serves as the PAN coordinator .

IEEE 802.15.4

The IEEE 802.15.4 standard defines the PHY and MAC layers for wireless communication [6]. It is designed to use as little transmission time as possible but still have a decent payload, while consuming as little power as possible. Each frame starts with a preamble and a start frame delimiter; it then continues with the MAC frame length and the MAC frame itself as seen in figure 2.2. The overhead for each PHY packet is only 4+1+1 133 tild 4.5%; when using the maximum transmission speed of 250kbit/s, each frame can be sent 133byte in 250kbit/s = 4.265ms. Furthermore, it can also operate in the 868MHz and 915MHz bands, maintaining the 250kbit/s transmission rate by using Offset quadrature phase-shift keying (O-QPSK).

Several network layer protocols are implemented on top of IEEE 802.15.4. The two that will be examined are 6LoWPAN and ZigBEE.

2.4.3 ZigBee

is a communication standard initially developed for home automation networks; it has several different protocols designed for specific areas such as lighting, remote control, or health care [27, 6]. Each of these protocols uses their own addressing with different overhead; however, there is also the possibility of direct IPv6 addressing. Then, the overhead is the same as for uncompressed 6LoWPAN, as seen in figure 2.5.

A new standard called ZigBee 3.0 aims to bring all these standards together under one roof to simplify the integration into IoT. The release date of this standard is set to Q4 2015.

2.5 LoRaWAN

LoRa (Long Range) is a proprietary spread spectrum modulation technique by Semtech. It is a derivative of Chirp Spread Spectrum (CSS). The LoRa physical layer may be used with any MAC layer; however, LoRaWAN is the currently proposed MAC which operates a network in a simple star topology.

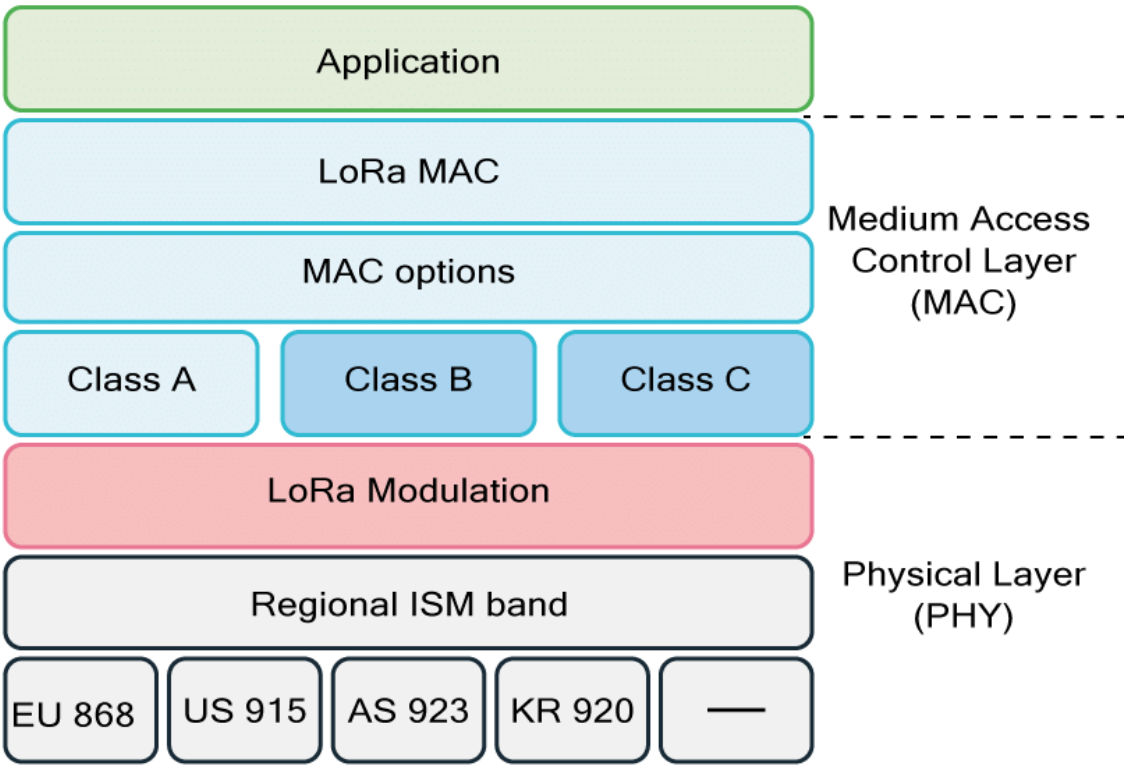

Figure 1. uhuhuh.

As LoRa is capable to transmit over very long distances it was decided that LoRaWAN only needs to support a star topology. Nodes transmit directly to a gateway which is powered and connected to a backbone infrastructure. Gateways are powerful devices with powerful radios capable to receive and decode multiple concurrent transmissions (up to 50). Three classes of node devices are defined: (1) Class A enddevices: The node transmits to the gateway when needed. After transmission the node opens a receive window to obtain queued messages from the gateway. (2) Class B enddevices with scheduled receive slots: The node behaves like a Class A node with additional receive windows at scheduled times. Gateway beacons are used for time synchronisation of end-devices. (3) Class C end-devices with maximal receive slots: these nodes are continuous listening which makes them unsuitable for battery powered operations.

2.5.1 ALIANCE

A) Class-A

A.1) Uplink LoRa Server supports Class-A devices. In Class-A a device is always in sleep mode, unless it has something to transmit. Only after an uplink transmission by the device, LoRa Server is able to schedule a downlink transmission. Received frames are de-duplicated (in case it has been received by multiple gateways), after which the mac-layer is handled by LoRa Server and the encrypted application-playload is forwarded to the application server.

A.2) Downlink LoRa Server persists a downlink device-queue for to which the application-server can enqueue downlink payloads. Once a receive window occurs, LoRa Server will transmit the first downlink payload to the device.

A.3) Confirmed data LoRa Server sends an acknowledgement to the application-server as soon one is received from the device. When the next uplink transmission does not contain an acknowledgement, a nACK is sent to the application-server.

Note: After a device (re)activation the device-queue is flushed.

B) Class-B LoRa Server supports Class-B devices. A Class-B device synchronizes its internal clock using Class-B beacons emitted by the gateway, this process is also called a “beacon lock”. Once in the state of a beacon lock, the device negotioates its ping-interval. LoRa Server is then able to schedule downlink transmissions on each occuring ping-interval.

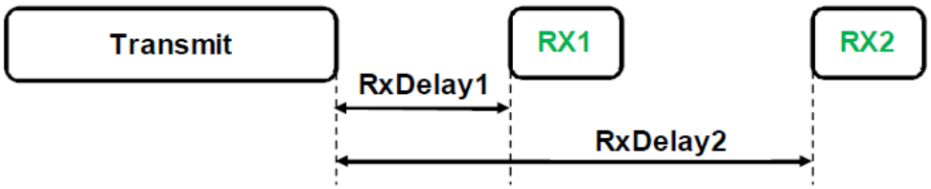

Figure 2. Class A.

B.1) Downlink LoRa Server persists all downlink payloads in its device-queue. When the device has acquired a beacon lock, it will schedule the payload for the next free ping-slot in the queue. When adding payloads to the queue when a beacon lock has not yet been acquired, LoRa Server will update all device-queue to be scheduled on the next free ping-slot once the device has acquired the beacon lock.

B.2) Confirmed data LoRa Server sends an acknowledgement to the application-server as soon one is received from the device. Until the frame has timed out, LoRa Server will wait with the transmission of the next downlink Class-B payload.

Note: The timeout of a confirmed Class-B downlink can be configured through the device-profile. This should be set to a value less than the interval between two ping-slots.

B.3) Requirements

Device The device must be able to operate in Class-B mode. This feature has been tested against the develop branch of the Semtech LoRaMac-node source.

Gateway The gateway must have a GNSS based time-source and must use at least the Semtech packet-forwarder version 4.0.1 or higher. It also requires LoRa Gateway Bridge 2.2.0 or higher.

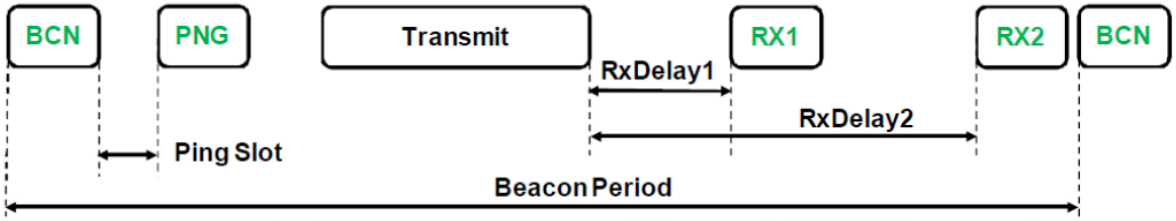

Figure 3. Class B.

C) Class-C

C.1) Downlink LoRa Server supports Class-C devices and uses the same Class-A downlink device-queue for Class-C downlink transmissions. The application-server can enqueue one or multiple downlink payloads and LoRa Server will transmit these (semi) immediately to the device, making sure no overlap exists in case of multiple Class-C transmissions.

C.2) Confirmed data LoRa Server sends an acknowledgement to the application-server as soon one is received from the device. Until the frame has timed out, LoRa Server will wait with the transmission of the next downlink Class-C payload.

Note: The timeout of a confirmed Class-C downlink can be configured through the device-profile.

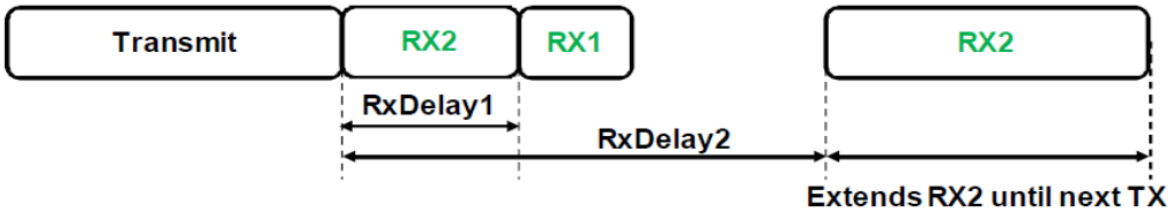

Figure 4. Class C.

2.5.2 SEMTECH

LoRa has four configurable parameters:

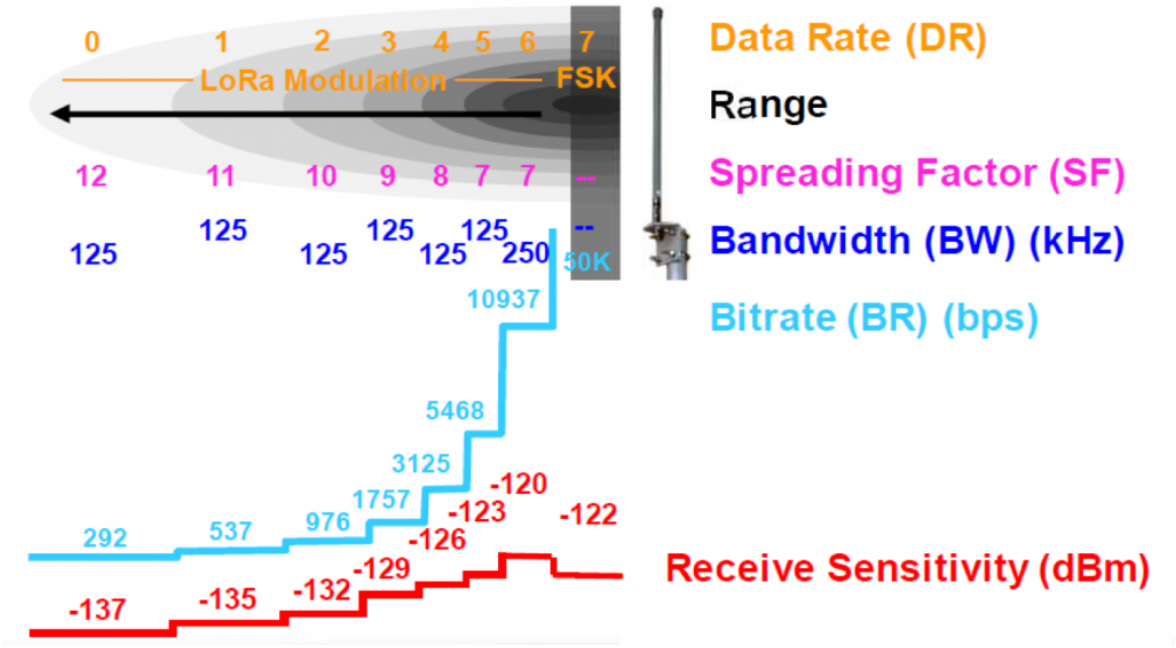

Figure 5. LoraWan Parameters.

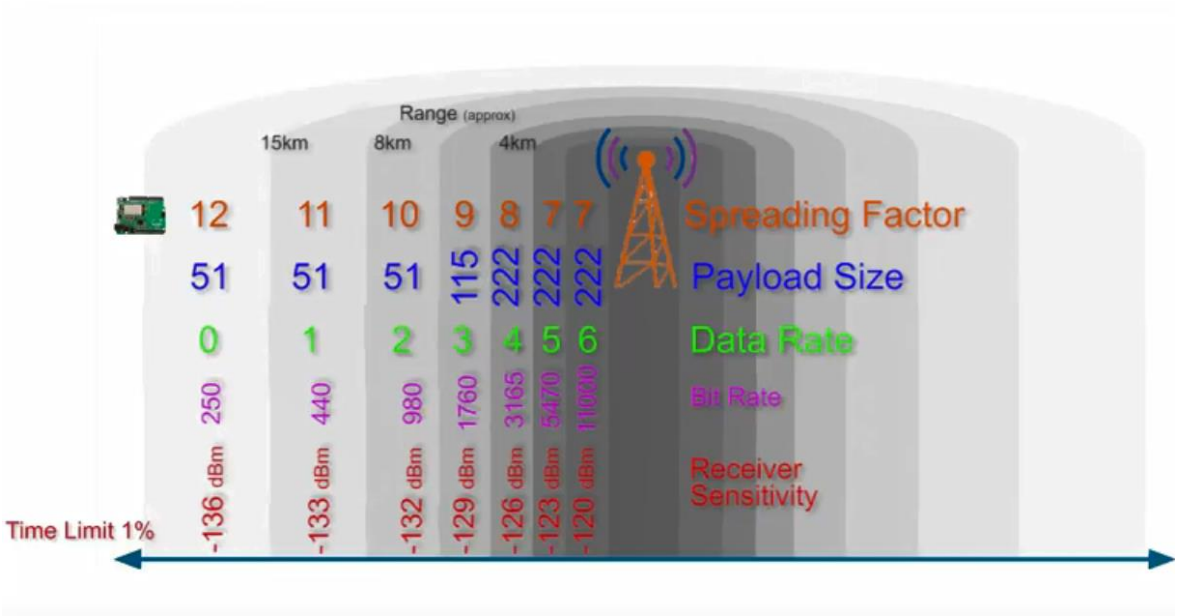

Figure 6. .

BW **Bandwidth:** Bandwidth (BW) is the range of frequencies in the transmission band. Higher BW gives a higher data rate (thus shorter time on air), but a lower sensitivity (due to integration of additional noise). A lower BW gives a higher sensitivity, but a lower data rate. Lower BW also requires more accurate crystals (less ppm). Data is send out at a chip rate equal to the bandwidth. So, a bandwidth of 125 kHz corresponds to a chip rate of 125 kcps. The SX1272 has three programmable bandwidth settings: 500 kHz, 250 kHz and 125 kHz. The Semtech SX1272 can be programmed in the range of 7.8 kHz to 500 kHz, though bandwidths lower than 62.5 kHz requires a temperature compensated crystal oscillator (TCXO).

CF **Carrier Frequency:** Carrier Frequency (CF) is the centre frequency used for the transmission band. For the SX1272 it is in the range of 860 MHz to 1020 MHz, programmable in steps of 61 Hz. The alternative radio chip Semtech SX1276 allows adjustment from 137 MHz to 1020 MHz.

CR **Coding Rate:** Coding Rate (CR) is the FEC rate used by the LoRa modem and offers protection against bursts of interference. A higher CR offers more protection, but increases time on air. Radios with different CR (and same CF/SF/BW), can still communicate with each other.

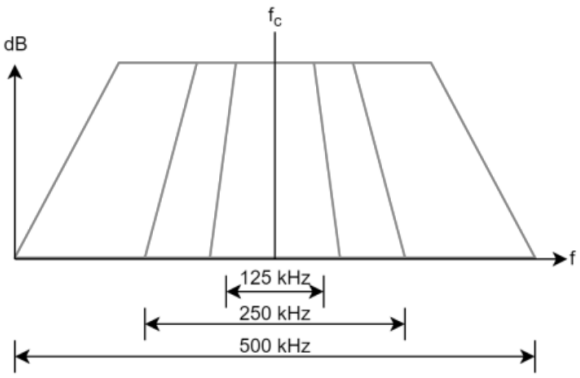

Figure 7. .

CR of the payload is stored in the header of the packet, which is always encoded at 4/8.

SF **Spreading Factor:** SF is the ratio between the symbol rate and chip rate. A higher spreading factor increases the Signal to Noise Ratio (SNR), and thus sensitivity and range, but also increases the air time of the packet. The number of chips per symbol is calculated as 2 sf . For example, with an SF of 12 (SF12) 4096 chips/symbol are used. Each increase in SF halves the transmission rate and, hence, doubles transmission duration and ultimately energy consumption. Spreading factor can be selected from 6 to 12. SF6, with the highest rate transmission, is a special case and requires special operations. For example, implicit headers are required. Radio communications with different SF are orthogonal to each other and network separation using different SF is possible.

Tx **Transmition power:**

Payload **Payload legth:**

SF		07	08	09	10	11	12	07	08	09	10	11	12	07	08	09	10	11	12
	BW	125						250						500					
07	125	x								x								x	
08			x								x								x
09				x								x							
10					x								x						
11						x													
12							x												
07	250							x								x			
08									x								x		
09		x								x								x	
10			x								x								x
11				x								x							
12					x								x						
07	500													x					
08															x				
09								x								x			
10									x								x		
11		x								x								x	
12			x								x								x

Table 3.2. uyuyuy

	SF_7	SF_B	SF_9	SF_{10}	SF_{11}	SF_{12}
SF_7	- 6	16	18	19	19	20
SF_8	24	- 6	20	22	22	22
SF_9	27	27	- 6	23	25	25
SF_{10}	30	30	30	- 6	26	28
SF_{11}	33	33	33	33	- 6	29
SF_{12}	36	36	36	36	36	- 6

Module	SX1261/62/68	SX1272/73	SX1276/77/78/79
Modem	LoRa & FSK	LoRa	LoRa
Link budget	170dB	157dB	168dB
Power amplifier	/61: +15dBm 62/68:+22dBm	+14 dBm	+14dBm
Rx current	4.6 mA	10 mA	10 mA
Bit rate	62.5kbps-LoRa 300kbps-FSK	300 kbps	300 kbps
Sensitivity	-148 dBm	-137 dBm	-148 dBm
Blocking immunity	88 dB	89 dB	Excellent
Frequency	150-960 MHz	860-1000 MHz	137-1020 MHz
RSSI		127 dB	127 dB

Table 3.3. [**gaddam_comparative_2018**]

2.6 Divers

2.6.1 IPLC

2.6.2 BACnet

2.6.3 Z-WAze

2.6.4 Bluetooth LE

BLE is developed to be backwards compatible with Bluetooth, but with lower data rate and power consumption [28]. Featuring a data rate of 1Mbit/s with a peak current consumption less than 15mA, it is a very efficient protocol for small amounts of data. Each frame can be transmitted 47bytes in 1Mbit/s = 376Mus; thanks to the short transmission time, the transceivers consumes less power as the transceiver can be in receive mode or completely off most of the time. BLE uses its own addressing methods and as the MAC frame size (figure 2.6) is only 39bytes, thus IPv6 addressing is not possible.

Starting from Bluetooth version 4.2, there is support for IPv6 addressing with the Internet Protocol Support Profile; the new version allows the BLE frame to be variable between 2 257 bytes. The network set-up is controlled by the standard Bluetooth methods, whereas IPv6 addressing is handled by 6LoWPAN as specified in IPv6 over Bluetooth Low Energy [29].

2.7 Summary and discussion

3 IoT Protocols

Application protocol	DDS	CoAP	AMQP	MQTT	MQTT-SN	XMPP	HTTP
Service discovery	mDNS		DNS-SD				
Transport	UDP/TCP						
Network	IPv6 RPL		IPv4/IPv6				
	6LowPan		RFC 2464				RFC 5072
MAC	IEEE 802.15.4		IEEE 802.11 (Wi-Fi)		IEEE 802.3 (Ethernet)		2G, 3G, LTE
	2.4GHz, 915, 868MHz		2.4, 5GHz				
	DSS, FSK, OFDM		CSMA/CA		CUTP, FO		

Table 3.4. Standardization efforts that support the IoT

3.1 Application

3.1.1 LwM2M

3.1.2 CBOR

3.1.3 DTLS

3.1.4 OSCOAP

3.1.5 CoAP

- **Ver:** is the version of CoAP
- **T:** is the type of Transaction
- **TKL:** Token length
- **Code:** represents the request method (1-10) or response code (40-255).
  - Ex: the code for GET, POST, PUT, and DELETE is 1, 2, 3, and 4, respectively.
- **Message ID:** is a unique identifier for matching the response.
- **Token:** Optional response matching token.

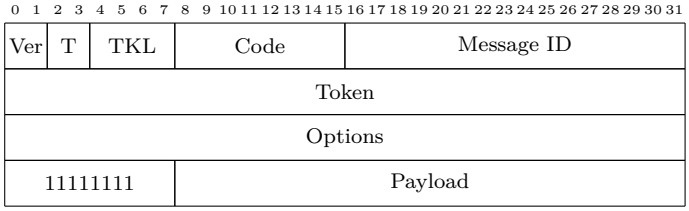

Figure 8. CoAP Header.

3.1.6 MQTT

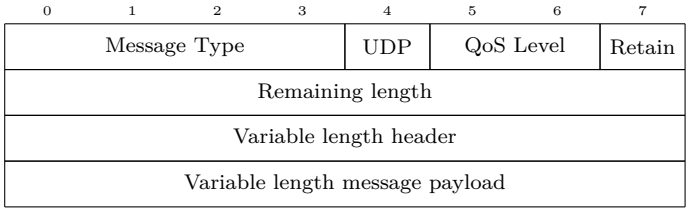

Figure 9. CoAP Header.

- **Message type:** CONNECT (1), CONNACK (2), PUBLISH (3), SUBSCRIBE (8) and so on
- **DUP flag:** indicates that the massage is duplicated
- **QoS Level:** identify the three levels of QoS for delivery assurance of Publish messages
- **Retain field:** retain the last received Publish message and submit it to new subscribers as a first message

3.1.7 XMPP

- Extensible Messaging and Presence Protocol
- Developed by the Jabber open source community
- An IETF instant messaging standard used for:
  - multi-party chatting, voice and telepresence
- Connects a client to a server using a XML stanzas
- An XML stanza is divided into 3 components:
  - message: fills the subject and body fields
  - presence: notifies customers of status updates
  - iq (info/query): pairs message senders and receivers
- Message stanzas identify:
  - the source (from) and destination (to) addresses
  - types, and IDs of XMPP entities

3.1.8 AMQP

- **Size:** the frame size.
- **DOFF:** the position of the body inside the frame.
- **Type:** the format and purpose of the frame.
  - Ex: 0x00 show that the frame is an AMQP frame
  - Ex: 0x01 represents a SASL frame.

3.1.9 DDS

- Data Distribution Service
- Developed by Object Management Group (OMG)
- Supports 23 QoS policies:
  - like security, urgency, priority, durability, reliability, etc
- Relies on a broker-less architecture
  - uses multicasting to bring excellent Quality of Service
  - real-time constraints
- DDS architecture defines two layers:
  - **DLRL:** Data-Local Reconstruction Layer
    - serves as the interface to the DCPS functionalities

- **DCPS:** Data-Centric Publish/Subscribe
  - delivering the information to the subscribers
- 5 entities are involved with the data flow in the DCPS layer:
  - Publisher:disseminates data
  - DataWriter: used by app to interact with the publisher
  - Subscriber: receives published data and delivers them to app
  - DataReader: employed by Subscriber to access received data
  - Topic: relate DataWriters to DataReaders
- No need for manual reconfiguration or extra administration
- It is able to run without infrastructure
- It is able to continue working if failure happens.
- It inquires names by sending an IP multicast message to all the nodes in the local domain
  - Clients asks devices that have the given name to reply back
  - the target machine receives its name and multicasts its IP @
  - Devices update their cache with the given name and IP @

3.1.10 mDNS

- Requires zero configuration aids to connect machine
- It uses mDNS to send DNS packets to specific multicast addresses through UDP
- There are two main steps to process Service Discovery:
  - finding host names of required services such as printers
  - pairing IP addresses with their host names using mDNS
- Advantages
  - IoT needs an architecture without dependency on a configuration mechanism
  - smart devices can join the platform or leave it without affecting the behavior of the whole system
- Drawbacks
  - Need for caching DNS entries

3.1.11 COAP (COnstrained Application Protocol)

The Constrained Application Protocol (CoAP) is a specialized web transfer protocol for use with constrained nodes and constrained networks in the Internet of Things. More detailed information about the protocol is given in the Contiki OS CoAP section.

A) Overview Like HTTP, CoAP is a document transfer protocol. Unlike HTTP, CoAP is designed for the needs of constrained devices. The packets are much smaller than HTTP TCP flows. Packets are simple to generate and can be parsed in place without consuming extra RAM in constrained devices. CoAP runs over UDP, not TCP. Clients and servers communicate through connectionless datagrams. Retries and reordering are implemented in the application stack. It follows a client/server model. Clients make requests to servers, servers send back responses. Clients may GET, PUT, POST and DELETE resources. CoAP implements the REST model from HTTP, with the primitives GET, POST, PUT and DELETE.

B) Coap Methods CoAP extends the HTTP request model with the ability to observe a resource. When the observe flag is set on a CoAP GET request, the server may continue to reply after the initial document has been transferred. This allows servers to stream state changes to clients as they occur. Either end may cancel the observation. CoAP defines a standard mechanism for resource discovery. Servers provide a list of their resources (along with metadata about them) at /.well-known/core. These links are in the application/link-format media type and allow a client to discover what resources are provided and what media types they are.

C) Coap Transactions

D) Coap Messages The CoAP message structure is designed to be simpler than HTTP, for reduced transmission data. Each field responds to a specific purpose.

- Constrained Application Protocol
- The IETF Constrained RESTful Environments
- CoAP is bound to UDP
- CoAP can be divided into two sub-layers
  - messaging sub-layer
  - request/response sub-layer
    a) Confirmable.

b) Non-confirmable.
c) Piggybacked responses.
d) Separate response

- CoAP, as in HTTP, uses methods such as:
  - GET, PUT, POST and DELETE to
  - Achieve, Create, Retrieve, Update and Delete
  - Ex: the GET method can be used by a server to inquire the client's temperature

3.1.12 MQTT

- Message Queue Telemetry Transport
- Andy Stanford-Clark of IBM and Arlen Nipper of Arcom
  - Standardized in 2013 at OASIS
- MQTT uses the publish/subscribe pattern to provide transition flexibility and simplicity of implementation
- MQTT is built on top of the TCP protocol
- MQTT delivers messages through three levels of QoS
- Specifications
  - MQTT v3.1 and MQTT-SN (MQTT-S or V1.2)
  - MQTT v3.1 adds broker support for indexing topic names
- The publisher acts as a generator of interesting data.

3.1.13 XMPP

- Extensible Messaging and Presence Protocol
- Developed by the Jabber open source community
- An IETF instant messaging standard used for:
  - multi-party chatting, voice and telepresence
- Connects a client to a server using a XML stanzas
- An XML stanza is divided into 3 components:
  - message: fills the subject and body fields
  - presence: notifies customers of status updates
  - iq (info/query): pairs message senders and receivers
- Message stanzas identify:
  - the source (from) and destination (to) addresses
  - types, and IDs of XMPP entities

3.1.14 AMQP

- Advanced Message Queuing Protocol
- Communications are handled by two main components
  - exchanges: route the messages to appropriate queues.
  - message queues: Messages can be stored in message queues and then be sent to receivers
- It also supports the publish/subscribe communications.
- It defines a layer of messaging on top of its transport layer.
- AMQP defines two types of messages
  - bare massages: supplied by the sender
  - annotated messages: seen at the receiver
- The header in this format conveys the delivery parameters:
  - durability, priority, time to live, first acquirer & delivery count.
- AMQP frame format

Size the frame size.
DOFF the position of the body inside the frame.
Type the format and purpose of the frame.
  - Ex: 0x00 show that the frame is an AMQP frame
  - Ex: 0x01 represents a SASL frame.

3.1.15 DDS

- Data Distribution Service
- Developed by Object Management Group (OMG)
- Supports 23 QoS policies:
  - like security, urgency, priority, durability, reliability, etc
- Relies on a broker-less architecture

Application protocol	Rest-Full	Transport	Publish/Subscribe	Request/Response	Security	QoS	Header size (Byte)
COAP	✓	UDP	✓	✓	DTLS	✓	4
MQTT	✗	TCP	✓	✗	SSL	✓	2
MQTT-SN	✗	TCP	✓	✗	SSL	✓	2
XMPP	✗	TCP	✓	✓	SSL	✗	-
AMQP	✗	TCP	✓	✗	SSL	✓	8
DDS	✗	UDP TCP	✓	✗	SSL DTLS	✓	-
HTTP	✓	TCP	✗	✓	SSL	✗	-

Table 3.5. Application protocols comparison

- uses multicasting to bring excellent Quality of Service
- real-time constraints

- DDS architecture defines two layers:

DLRL Data-Local Reconstruction Layer
- ✳ serves as the interface to the DCPS functionalities

DCPS Data-Centric Publish/Subscribe
- ✳ delivering the information to the subscribers

- 5 entities are involved with the data flow in the DCPS layer:
  - Publisher:disseminates data
  - DataWriter: used by app to interact with the publisher
  - Subscriber: receives published data and delivers them to app
  - DataReader: employed by Subscriber to access received data
  - Topic: relate DataWriters to DataReaders
- No need for manual reconfiguration or extra administration
- It is able to run without infrastructure
- It is able to continue working if failure happens.
- It inquires names by sending an IP multicast message to all the nodes in the local domain
  - Clients asks devices that have the given name to reply back
  - the target machine receives its name and multicasts its IP @
  - Devices update their cache with the given name and IP @

3.1.16 mDNS

- Requires zero configuration aids to connect machine
- It uses mDNS to send DNS packets to specific multicast addresses through UDP
- There are two main steps to process Service Discovery:
  - finding host names of required services such as printers
  - pairing IP addresses with their host names using mDNS
- Advantages
  - IoT needs an architecture without dependency on a configuration mechanism
  - smart devices can join the platform or leave it without affecting the behavior of the whole system
- Drawbacks
  - Need for caching DNS entries

3.2 Network

3.2.1 6TiSCH

3.2.2 OLSRv2

3.2.3 AODVv2

3.2.4 LoRaWAN

3.2.5 ROHC

3.2.6 IPHC

3.2.7 SCHC

3.2.8 NHC

3.2.9 ROLL

3.2.10 RPL

RPL is a Distance Vector IPv6 routing protocol for LLNs that specifies how to build a Destination Oriented Directed Acyclic Graph (DODAG) using an objective function and a set of metrics/constraints. The objective function operates on a combination of metrics and constraints to compute the 'best' path.

An RPL Instance consists of multiple Destination Oriented Directed Acyclic Graphs (DODAGs). Traffic moves either up towards the DODAG root or down towards the DODAG leafs. The graph building process starts at the root or LBR (LowPAN Border Router). There could be multiple roots configured in the system. The RPL routing protocol specifies a set of ICMPv6 control messages to exchange graph related information. These messages are called DIS (DODAG Information Solicitation), DIO (DODAG Information Object) and DAO (DODAG Destination Advertisement Object). The root starts advertising the information about the graph using the DIO message. The nodes in the listening vicinity (neighbouring nodes) of the root will receive and process DIO messages potentially from multiple nodes and makes a decision based on certain rules (according to the objective function, DAG characteristics, advertised path cost and potentially local policy) whether to join the graph or not. Once the node has joined a graph it has a route toward the graph (DODAG) root. The graph root is termed as the 'parent' of the node. The node computes the 'rank' of itself within the graph, which indicates the "coordinates" of the node in the graph hierarchy. If configured to act as a router, it starts advertising the graph information with the new information to its neighbouring peers. If the node is a "leaf node", it simply joins the graph and does not send any DIO message. The neighbouring peers will repeat this process and do parent selection, route addition and graph information advertisement using DIO messages. This rippling effect builds the graph edges out from the root to the leaf nodes where the process terminates. In this formation each node of the graph has a routing entry towards its parent (or multiple parents depending on the objective function) in a hop-by-hop fashion and the leaf nodes can send a data packet all the way to root of the graph by just forwarding the packet to its immediate parent. This model represents a MP2P (Multipoint-to-point) forwarding model where each node of the graph has reach-ability toward the graph root. This is also referred to as UPWARD routing. Each node in the graph has a 'rank' that is relative and represents an increasing coordinate of the relative position of the node with respect to the root in graph topology. The notion of "rank" is used by RPL for various purposes including loop avoidance. The MP2P flow of traffic is called the 'up' direction in the DODAG.

The DIS message is used by the nodes to proactively solicit graph information (via DIO) from the neighbouring nodes should it become active in a stable graph environment using the 'poll' or 'pull' model of retrieving graph information or in other conditions. Similar to MP2P or 'up' direction of traffic, which flows from the leaf towards the root there is a need for traffic to flow in the opposite or 'down' direction. This traffic may originate from outside the LLN network, at the root or at any intermediate nodes and destined to a (leaf) node. This requires a routing state to be built at every node and a mechanism to populate these routes. This is accomplished by the DAO (Destination Advertisement Object) message. DAO messages are used to advertise prefix reachability towards the leaf nodes in support of the 'down' traffic. These messages carry prefix information, valid lifetime and other information about the distance of the prefix. As each node joins the graph it will send DAO message to its parent set. Alternately, a node or root can poll the sub-dag for DAO message through an indication in the DIO message. As each node receives the DAO message, it processes the prefix information and adds a routing entry in the routing table. It optionally aggregates the prefix

information received from various nodes in the subdag and sends a DAO message to its parent set. This process continues until the prefix information reaches the root and a complete path to the prefix is setup. Note that this mode is called the “storing” mode of operation where intermediate nodes have available memory to store routing tables. RPL also supports another mode called “non-storing” mode where intermediate node do not store any routes.

3.2.11 6LowPAN

6LoWPAN is a networking technology or adaptation layer that allows IPv6 packets to be carried efficiently within a small link layer frame, over IEEE 802.15.4 based networks. As the full name implies, “IPv6 over Low-Power Wireless Personal Area Networks”, it is a protocol for connecting wireless low power networks using IPv6.

As the full name implies, “IPv6 over Low-Power Wireless Personal Area Networks”, it is a protocol for connecting wireless low power networks using IPv6.

A) Characteristics

- Compression of IPv6 and UDP/ICMP headers
- Fragmentation / reassembly of IPv6 packets
- Mesh addressing
- Stateless auto configuration
-

B) Encapsulation Header format All LowPAN encapsulated datagrams are prefixed by an encapsulation header stack. Each header in the stack starts with a header type field followed by zero or more header fields.

C) Fragment Header The fragment header is used when the payload is too large to fit in a single IEEE 802.15.4 frame. The Fragment header is analogous to the IEEE 1394 Fragment header and includes three fields: Datagram Size, Datagram Tag, and Datagram Offset. Datagram Size identifies the total size of the unfragmented payload and is included with every fragment to simplify buffer allocation at the receiver when fragments arrive out-oforder. Datagram Tag identifies the set of fragments that correspond to a given payload and is used to match up fragments of the same payload. Datagram Offset identifies the fragment’s offset within the unfragmented payload and is in units of 8-byte chunks.

D) Mesh addressing header The Mesh Addressing header is used to forward 6LoWPAN payloads over multiple radio hops and support layer-two forwarding. The mesh addressing header includes three fields: Hop Limit, Source Address, and Destination Address. The Hop Limit field is analogous to the IPv6 Hop Limit and limits the number of hops for forwarding. The Hop Limit field is decremented by each forwarding node, and if decremented to zero the frame is dropped. The source and destination addresses indicate the end-points of an IP hop. Both addresses are IEEE 802.15.4 link addresses and may carry either a short or extended address.

E) Header compression (RFC4944) RFC 4944 defines HC1, a stateless compression scheme optimized for link-local IPv6 communication. HC1 is identified by an encoding byte following the Compressed IPv6 dispatch header, and it operates on fields in the upper-layer headers. 6LoWPAN elides some fields by assuming commonly used values. For example, it compresses the 64-bit network prefix for both source and destination addresses to a single bit each when they carry the well-known link-local prefix. 6LoWPAN compresses the Next Header field to two bits whenever the packet uses UDP, TCP, or ICMPv6. Furthermore, 6LoWPAN compresses Traffic Class and Flow Label to a single bit when their values are both zero. Each compressed form has reserved values that indicate that the fields are carried inline for use when they don’t match the elided case. 6LoWPAN elides other fields by exploiting cross-layer redundancy. It can derive Payload Length – which is always elided – from the 802.15.4 frame or 6LoWPAN fragmentation header. The 64-bit interface identifier (IID) for both source and destination addresses are elided if the destination can derive them from the corresponding link-layer address in the 802.15.4 or mesh addressing header. Finally, 6LoWPAN always elides Version by communicating via IPv6.

The HC1 encoding is shown in Figure 11. The first byte is the dispatch byte and indicates the use of HC1. Following the dispatch byte are 8 bits that identify how the IPv6 fields are compressed. For each address, one bit is used to indicate if the IPv6 prefix is linklocal and elided and one bit is used to indicate if the IID can be derived from the IEEE 802.15.4 link address. The TF bit indicates whether Traffic Class and Flow Label are both zero and elided. The two Next Header bits indicate if the IPv6 Next Header value is 7UDP, TCP, or ICMP and compressed or carried inline. The HC2 bit indicates

if the next header is compressed using HC2. Fully compressed, the HC1 encoding reduces the IPv6 header to three bytes, including the dispatch header. Hops Left is the only field always carried inline.

RFC 4944 uses stateless compression techniques to reduce the overhead of UDP headers. When the HC2 bit is set in the HC1 encoding, an additional 8-bits is included immediately following the HC1 encoding bits that specify how the UDP header is compressed. To effectively compress UDP ports, 6LoWPAN introduces a range of wellknown ports (61616 – 61631). When ports fall in the well-known range, the upper 12 bits may be elided. If both ports fall within range, both Source and Destination ports are compressed down to a single byte. HC2 also allows elision of the UDP Length, as it can be derived from the IPv6 Payload Length field.

The best-case compression efficiency occurs with link-local unicast communication, where HC1 and HC2 can compress a UDP/IPv6 header down to 7 bytes. The Version, Traffic Class, Flow Label, Payload Length, Next Header, and linklocal prefixes for the IPv6 Source and Destination addresses are all elided. The suffix for both IPv6 source and destination addresses are derived from the IEEE 802.15.4 header.

However, RFC 4944 does not efficiently compress headers when communicating outside of link-local scope or when using multicast. Any prefix other than the linklocal prefix must be carried inline. Any suffix must be at least 64 bits when carried inline even if derived from a short 802.15.4 address. As shown in Figure 8, HC1/HC2 can compress a link-local multicast UDP/IPv6 header down to 23 bytes in the best case. When communicating with nodes outside the LoWPAN, the IPv6 Source Address prefix and full IPv6 Destination Address must be carried inline.

F) Header compression Improved (draft-hui-6lowpan-hc-01) To provide better compression over a broader range of scenarios, the 6LoWPAN working group is standardizing an improved header compression encoding format, called HC. The format defines a new encoding for compressing IPv6 header, called IPHC. The new format allows Traffic Class and Flow Label to be individually compressed, Hop Limit compression when common values (E.g., 1 or 255) are used, makes use of shared-context to elide the prefix from IPv6 addresses, and supports multicast addresses most often used for IPv6 ND and SLAAC. Contexts act as shared state for all nodes within the LoWPAN. A single context holds a single prefix. IPHC identifies the context using a 4-bit index, allowing IPHC to support up to 16 contexts simultaneously within the LoWPAN. When an IPv6 address matches a context's stored prefix, IPHC compresses the prefix to the context's 4-bit identifier. Note that contexts are not limited to prefixes assigned to the LoWPAN but can contain any arbitrary prefix. As a result, share contexts can be configured such that LoWPAN nodes can compress the prefix in both Source and Destination addresses even when communicating with nodes outside the LoWPAN.

The improved header compression encoding is shown in Figure 8. The first three bits (011) form the header type and indicate the use of IPHC. The TF bits indicate whether the Traffic Class and/or Flow Label fields are compressed. The HLIM bits indicate whether the Hop Limit takes the value 1 or 255 and compressed, or carried inline.

Bits 8-15 of the IPHC encoding indicate the compression methods used for the IPv6 Source and Destination Addresses. When the Context Identifier (CID) bit is zero, the default context may be used to compress Source and/or Destination Addresses. This mode is typically when both Source and Destination Addresses are assigned to nodes in the same LoWPAN. When the CID bit is one, two additional 4-bit fields follow the IPHC encoding to indicate which one of 16 contexts is in use for the source and destination addresses. The Source Address Compression (SAC) indicates whether stateless compression is used (typically for link-local communication) or stateful context-based compression is used (typically for global communication). The Source Address Mode (SAM) indicates whether the full Source Address is carried inline, upper 16 or 64-bits are elided, or the full Source Address is elided. When SAC is set and the Source Addresses' prefix is elided, the identified context is used to restore those bits. The Multicast (M) field indicates whether the Destination Address is a unicast or multicast address. When the Destination Address is a unicast address, the DAC and DAM bits are analogous to the SAC and SAM bits. When the Destination Address is a multicast address, the DAM bits indicate different forms of multicast compression. HC also defines a new framework for compressing arbitrary next headers, called NHC. HC2 in RFC 4944 is only capable of compressing UDP, TCP, and ICMPv6 headers, the latter two are not yet defined. Instead, the NHC header defines a new variable length Next Header identifier, allowing for future definition of arbitrary next header compression encodings. HC initially defines a compression encoding for UDP headers, similar to that defined in RFC 4944. Like RFC 4944, HC utilizes the same well-known port range (61616-61631) to effectively compress UDP ports down to 4-bits each in the best case. However, HC no longer provides an option to carry the Payload Length in line, as it can always be derived from the IPv6 header. Finally, HC allows elision of the UDP Checksum whenever an 10upper layer message integrity check

covers the same information and has at least the same strength. Such a scenario is typical when transportor application-layer security is used. As a result, the UDP header can be compressed down to two bytes in the best case.

Routing protocol	Control Cost	Link Cost	Node Cost
OSPF/IS-IS	✗	✓	✗
OLSRv2	?	✓	✓
RIP	✓	?	✗
DSR	✓	✗	✗
RPL	✓	✓	✓

Table 3.6. Routing protocols comparison [__rpl2__]

- Routing over low-power and lossy links (ROLL)
- Support minimal routing requirements.
  - like multipoint-to-point, point-to-multipoint and point-to-point.
- A Destination Oriented Directed Acyclic Graph (DODAG)
  - Directed acyclic graph with a single root.
  - Each node is aware of ts parents
  - but not about related children
- RPL uses four types of control messages
  - DODAG Information Object (DIO)
  - Destination Advertisement Object (DAO)
  - DODAG Information Solicitation (DIS)
  - DAO Acknowledgment (DAO-ACk)
- Standard topologies to form IEEE 802.15.4e networks are

  Star contains at least one FFD and some RFDs

  Mesh contains a PAN coordinator and other nodes communicate with each other

  Cluster consists of a PAN coordinator, a cluster head and normal nodes.
- The IEEE 802.15.4e standard supports 2 types of network nodes

  FFD Full function device: serve as a coordinator
  - It is responsible for creation, control and maintenance of the net
  - It store a routing table in their memory and implement a full MAC

  RFD Reduced function devices: simple nodes with restricted resources
  - They can only communicate with a coordinator
  - They are limited to a star topology

Routing protocol	Control Cost	Link Cost	Node Cost
OSPF/IS-IS	✗	✓	✗
OLSRv2	?	✓	✓
RIP	✓	?	✗
DSR	✓	✗	✗
RPL	✓	✓	✓

Table 3.7. Routing protocols comparison [__rpl2__]

3.3 MAC

Channel based	FDMA	OFDMA WDMA SC-FDMA		
	TDMA	MF-TDMA STDMA		
	CDMA	W-CDMA TD-CDMA TD-SCDMA DS-CDMA FH-CDMA MC-CDMA		
	SDMA	HC-SDMA		
Packet-based	Collision recovery	ALOHA Slotted ALOHA R-ALOHA AX.25 CSMA/CD		
	Collision avoidance	MACA MACAW CSMA CSMA/CA DCF PCF HCF CSMA/CARP		
	Collision-free	Token ring Token bus MS-ALOHA		
Duplexing methods	Delay and disruption tolerant	MANET VANET DTN Dynamic Source Routing		

Table 3.8

3.3.1 Sharing the channel

A) TDMA, FDMA, CDMA, TSMA

3.3.2 Transmitting information

A) TFDM, TDSSS, TFHSS

3.4 Radio

3.4.1 Digital modulation

A) ASK, APSK, CPM, FSK, MFSK, MSK, OOK, PPM, PSK, QAM, SC-FDE, TCM WDM

3.4.2 Hierarchical modulation

A) QAM, WDM

3.4.3 Spread spectrum

A) SS, DSSS, FHSS, THSS

3.4.4 Radio performance

A) Power Level (dB) The dB measures the power of a signal as a function of its ratio to another standardized value. The abbreviation dB is often combined with other abbreviations to represent the values that are compared. Here are two examples:

- dBm—The dB value is compared to 1 mW.
- dBw—The dB value is compared to 1 W.

$$Power(indB) = 10 * log(10)(Signal/Reference) \quad (1)$$

Where:

- log(10) is logarithm base 10.
- Signal is the power of the signal.
- Reference is the reference power.

For example, if you want to calculate the power in dB of 50 mW:
Power (in dB) = 10 * log(10) (50/1) = 10 * 1.7 = 17 dBm

B) Receive Signal Strength Indicator RSSI Receiver sensitivity is defined as the minimum signal power level with an acceptable Bit Error Rate (in dBm or mW) that is necessary for the receiver to accurately decode a given signal. This is usually expressed in a negative number depending on the data rate. For example an Access Point may require an RSSI of at least negative -91 dBm at 1 MB and an even higher strength RF power -79 dBm to decode 54 MB.

C) Signal to Noise Ratio SNR Noise is any signal that interferes with your signal. Noise can be due to other wireless devices such as cordless phones, microwave devices, etc. This value is measured in decibels from 0 (zero) to -120 (minus 120). Noise level is the amount of interference in your wireless signal, so lower is generally good for WLAN deployments. Typical environments range between -90dBm and -98dBm with little ambient noise. This value may be even higher if there is a lot of RF interference coming in from other non-802.11 devices on the same spectrum Signal to Noise Ratio or SNR is defined as the ratio of the transmitted power from the AP to the ambient (noise floor) energy present. To calculate the SNR value, we add the Signal Value to the Noise Value to get the SNR ratio. A positive value of the SNR ratio is always better. For example, say your Signal value is -55dBm and your Noise value is -95dBm. The difference of Signal (-55dBm) + Noise (-95dBm) = 40db—This means you have an SNR of 40. Note that in the above equation you are not merely adding two numbers, but are interested in the "difference" between the Signal and Noise values, which is usually a positive number. The lower the number, the lower the difference between noise and transmitted power, which in turn means lower quality of signal. The higher the difference between Signal and Noise means that the transmitted signal is of much higher power than the noise floor, thereby making it easier for a WLAN client to decode the signal.

D) Signal Attenuation Signal attenuation or signal loss occurs even as the signal passes through air. The loss of signal strength is more pronounced as the signal passes through different objects. A transmit power of 20 mW is equivalent to 13 dBm. Therefore if the transmitted power at the entry point of a plasterboard wall is at 13 dBm, the signal strength will be reduced to 10 dBm when exiting that wall. Some common examples are shown in Table 10-5.

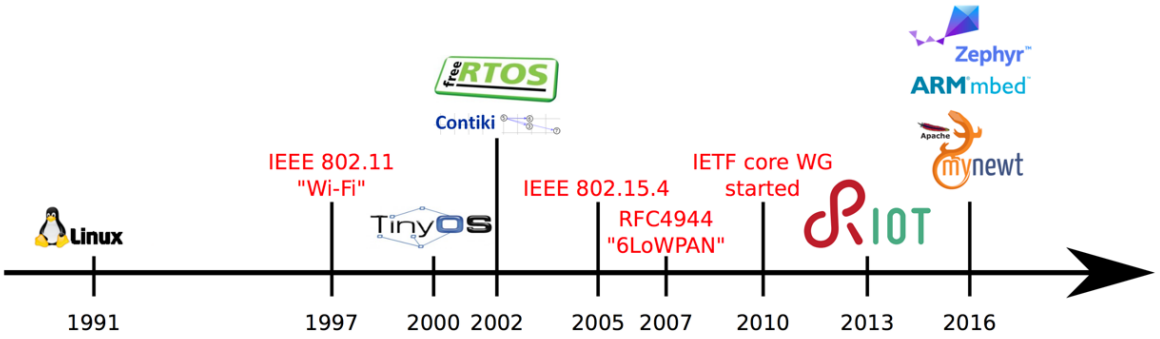

Figure 10. .

3.5 Summary and discussion

4 IoT end devices

4.1 Software platform

The operating system is the foundation of the IoT technology as it provides the functions for the connectivity between the nodes. However, different types of nodes need different levels of OS complexity; a passive node generally only needs the communication stack and is not in need of any threading capabilities, as the program can handle all logic in one function. Active nodes and border routers need to have a much more complex OS, as they need to be able to handle several running threads or processes, e.g. routing, data collection and interrupts. To qualify as an OS suitable for the IoT, it needs to meet the basic requirements: • Low Random-access memory (RAM) footprint • Low Read-only memory (ROM) footprint • Multi-tasking • Power management (PM) • Soft real-time These requirements are directly bound to the type of hardware designed for the IoT. As this type of hardware in general needs to have a small form factor and a long battery life, the on-board memory is usually limited to keep down size and energy consumption. Also, because of the limited amount of memory, the implementation of threads is usually a challenging task, as context switching needs to store thread or process variables to memory. The size of the memory also directly affects the energy consumption, as memory in general is very power hungry during accesses. To be able reduce the energy consumption, the OS needs some kind of power management. The power management does not only let the OS turn on and off peripherals such as flash memory, I/O, and sensors, but also puts the MCU itself in different power modes. As the nodes can be used to control and monitor consumer devices, either a hard or soft real-time OS is required. Otherwise, actions requiring a close to instantaneous reaction might be indefinitely delayed. Hard real-time means that the OS scheduler can guarantee latency and execution time, whereas Soft real-time means that latency and execution time is seen as real-time but can not be guaranteed by the scheduler. Operating systems that meet the above requirements are compared in table 2.1 and 2.2.

4.1.1 Contiki

Contiki is a embedded operating system developed for IoT written in C [12]. It supports a broad range of MCUs and has drivers for various transceivers. The OS does not only support TCP/IPv4 and IPv6 with the uIP stack [9], but also has support for the 6LoWPAN stack and its own stack called RIME. It supports threading with a thread system called Protothreads [13]. The threads are stack-less and thus use only two bytes of memory per thread; however, each thread is bound to one function and it only has permission to control its own execution. Included in Contiki, there is a range of applications such as a HTTP, Constrained Application Protocol (CoAP), FTP, and DHCP servers, as well as other useful programs and tools. These applications can be included in a project and can run simultaneously with the help of Protothreads. The limitations to what applications can be run is the amount of RAM and ROM the target MCU provides. A standard system with IPv6 networking needs about 10 kB RAM and 30 kB ROM but as applications are added the requirements tend to grow.

Contiki is an open source operating system for the Internet of Things. Contiki connects tiny low-cost, low-power micro-controllers to the Internet.

2k RAM, 60k ROM; 10k RAM, 48K ROM Portable to tiny low-power micro-controllers I386 based, ARM, AVR, MSP430, ... Implements uIP stack IPv6 protocol for Wireless Sensor Networks (WSN)

Uses the protothreads abstraction to run multiple process in an event based kernel. “Emulates” concurrency Contiki has an event based kernel (1 stack) Calls a process when an event happens

Contiki size One of the main aspect of the system, is the modularity of the code. Besides the system core, each program builds only the necessary modules to be able to run, not the entire system image. This way, the memory used from the system, can be reduced to the strictly necessary. This methodology makes more practical any change in any module, if it is needed. The code size of Contiki is larger than that of TinyOS, but smaller than that of the Mantis system. Contiki’s event kernel is significantly larger than that of TinyOS because of the different services provided. While the TinyOS event kernel only provides a FIFO event queue scheduler , the Contiki kernel supports both FIFO events and poll handlers with priorities. Furthermore, the flexibility in Contiki requires more run-time code than for a system like TinyOS, where compile time optimization can be done to a larger extent.

The documentation in the doc folder can be compiled, in order to get the html wiki of all the code. It needs doxygen installed, and to run the command “make html”. This will create a new folder, “doc/hmtl”, and in the index.html file, the wiki can be opened.

Contiki Hardware Contiki can be run in a number of platforms, each one with a different CPU. Tab.7 shows the hardware platforms currently defined in the Contiki code tree. All these platforms are in the “platform” folder of the code.

Kernel structure

4.1.2 RIOT

RIOT is a open source embedded operating system supported by Freie Universität Berlin, INIRA, and Hamburg University of Applied Sciences [14]. The kernel is written in C but the upper layers support C++ as well. As the project originates from a project with real-time and reliability requirements, the kernel supports hard real-time multi-tasking scheduling. One of the goals of the project is to make the OS completely POSIX compliant. Overhead for multi-threading is minimal with less than 25 bytes per thread. Both IPv6 and 6LoWPAN is supported together with UDP, TCP, and IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL); and CoAP and Concise Binary Object Representation (CBOR) are available as application level communication protocols.

4.1.3 TinyOS

TinyOS is written in Network Embedded Systems C (nesC) which is a variant of C [15]. nesC does not have any dynamic memory allocation and all program paths are available at compile-time. This is manageable thanks to the structure of the language; it uses modules and interfaces instead of functions [16]. The modules use and provide interfaces and are interconnected with configurations; this procedure makes up the structure of the program. Multitasking is implemented in two ways: trough tasks and events. Tasks, which focus on computation, are non-preemptive, and run until completion. In contrast, events which focus on external events i.e. interrupts, are preemptive, and have separate start and stop functions. The OS has full support for both 6LoWPAN and RPL, and also have libraries for CoAP.

4.1.4 freeRTOS

One of the more popular and widely known operating systems is freeRTOS [17]. Written in C with only a few source files, it is a simple but powerful OS, easy to overview and extend. It features two modes of scheduling, pre-emptive and co-operative, which may be selected according to the requirements of the application. Two types of multitasking are featured: one is a lightweight Co-routine type, which has a shared stack for lower RAM usage and is thus aimed to be used on very small devices; the other is simply called Task, has its own stack and can therefore be fully pre-empted. Tasks also support priorities which are used together with the pre-emptive scheduler. The communication methods supported out-of-the-box are TCP and UDP.

4.1.5 Summary and conclusion

4.2 Hardware platform

4.2.1 Processing Unit

Even though the hardware is in one sense the tool that the OS uses to make IoT possible, it is still very important to select a platform that is future-proof and extensible. To be regarded as an extensible platform, the hardware needs to have I/O connections that can be used by external peripherals. Amongst the candidate interfaces are Serial Peripheral Interface (SPI), Inter-Integrated Circuit (I 2 C), and Controller Area Network (CAN). These interfaces allow developers to attach

	LiteOS	Nano-RK	MANTIS	Contiki
Architecture	Monolithic	Layered	Modular	Modular
Scheduling Memory	Round Robin	Monotonic harmonized	Priority classes	Interrupts execute w.r.t.
Network	File	Socket abstraction	At Kernel COMM layer	uIP, Rime
Virtualization and Completion	Synchronization primitives	Serialized access semaphores	Semaphores	Serialized, Access
Multi threading	✓	✓	✗	✓
Dynamic protection	✓	✗	✓	✓
Memory Stack	✓	✗	✗	✗

Table 3.9. Common operating systems used in IoT environment [**al-fuqaha_internet_24**]

OS	**Contiki**	**MANTIS**	**Nano-RK**	**LiteOS**
Architecture	Modular	Modular	Layered	Monolithic
Multi threading	✓	✗	✓	✓
Scheduling	Interrupts	Priority	Monotonic	Round Robin
	execute w.r.t.	classes	harmonized	
Dynamic Memory	✓	✓	✗	✓
Memory protection	✗	✗	✗	✓
Network Stack	uIP/Rime	At Kernel/COMM layer	Socket/abstraction	file
Virtualization and Completion	Serialized/Access	Semaphores	Serialized/semaphores	Synchronization/primitives

Table 3.10. Common operating systems used in IoT environment [**al-fuqaha_internet_24**]

custom-made PCBs with sensors for monitoring or actuators for controlling the environment. The best practice is to implement an extension socket with a well-known form factor. A future-proof device is specified as a device that will be as attractive in the future as it is today. For hardware, this is very hard to achieve as there is constant development that follows Moore's Law [4]; however, the most important aspects are: the age of the chip, its expected remaining lifetime, and number of current implementations i.e. its popularity. If a device is widely used by consumers, the lifetime of the product is likely to be extended. One last thing to take into consideration is the product family; if the chip belongs to a family with several members the transition to a newer chip is usually easier.

A) OpenMote OpenMote is based on the Ti CC2538 System on Chip (SoC), which combines an ARM Cortex-M3 with a IEEE 802.15.4 transceiver in one chip [18, 19]. The board follows the XBee form factor for easier extensibility, which is used to connect the core board to either the OpenBattery or OpenBase extension boards [20, 21]. It originates from the CC2538DK which was used by Thingsquare to demo their Mist IoT solution [22]. Hence, the board has full support for Contiki, which is the foundation of Thingsquare. It can run both as a battery-powered sensor board and as a border router, depending on what extension board it is attached to, e.g OpenBattery or OpenBase. Furthermore, the board has limited support but ongoing development for RIOT and also full support for freeRTOS.

B) MSB430-H The Modular Sensor Board 430-H from Freie Universität Berlin was designed for their ScatterWeb project [23]. As the university also hosts the RIOT project, the decision to support RIOT was natural. The main board has a Ti MSP430F1612 MCU [24], a **Ti CC1100 transceiver**, and a battery slot for dual AA batteries; it also includes a SHT11 temperature and humidity sensor and a MMA7260Q accelerometer to speed up early development. All GPIO pins and buses are connected to external pins for extensibility. Other modules with new peripherals can then be added by making a PCB that matches the external pin layout.

C) Zolertia As many other Wireless Sensor Network (WSN) products, the Zolertia Z1 builds upon the MSP430 MCU [25, 26]. The communication is managed by the Ti CC2420 which operates in the 2.4 GHz band. The platform includes two sensors: the SHT11 temperature and humidity sensor and the MMA7600Q accelerometer. Extensibility is ensured with: two connections designed especially for external sensors, an external connector with USB, Universal asynchronous **receiver/transmitter (UART)**, SPI, and I 2 C.

4.2.2 Radio Unit

A) Lora Tranceiver To limit the complexity of the radio unit:

- limiting message size: maximum application payload size between 51 and 222 bytes, depending on the spreading factor
- using simple channel codes: Hamming code

- supporting only half-duplex operation
- using one transmit-and-receive antenna
- on-chip integrating power amplifier (since transmit power is limited)

limiting message size: maximum application payload size between 51 and 222 bytes, depending on the spreading factor using simple channel codes: Hamming code supporting only half-duplex operation using one transmit-and-receive antenna on-chip integrating power amplifier (since transmit power is limited)

Ref	Module	Frequency MHz	Tx power	Rx power	Sensitivity	Channels	Distance
[libelium_waspmote_2015]	Semtech SX1272	863-870 (EU) 902-928 (US)	14 dBm	dBm	-134 dBm	8 13	22+ km
[libelium_waspmote_2017]	2.4	2483					

Table 3.11

4.2.3 Sensing Unit

A) GPS

B) Humidity

C) Temperature

4.3 Summary and discussion

5 SDN platforms

Plan de controle	Plan de gestion	Plan de doonées
Controle d'admission	Controle et supervision de QoS	Controle du trafic
Réservation de ressources	Gestion de contrats	Façonnage du trafic
Routage	QoS mapping	Controle de congestion
Signalisation	Politique de QoS	Classification de paquets
		Marquage de paquets
		Ordonnancements des paquets
		Gestion de files d'attente

Table 3.12. An example table.

n_software_2014] Many studies have identified SDN as a potential solution to the WSN challenges, as well as a model for heterogeneous integration.

n_software_2014] This shortfall can be resolved by using the SDN approach.

obo_survey_2017] SDN also enhances better control of heterogeneous network infrastructures.

obo_survey_2017] Anadiotis et al. define a SDN operating system for IoT that integrates SDN based WSN **(SDN-WISE)**. This experiment shows how heterogeneity between different kinds of SDN networks can be achieved.

obo_survey_2017] In cellular networks, OpenRoads presents an approach of introducing SDN based heterogeneity in wireless networks for operators.

e_software_2017] There has been a plethora of (industrial) studies synergising SDN in IoT. The major characteristics of IoT are low latency,wireless access, mobility and heterogeneity.

e_software_2017] Thus a bottom-up approach application of SDN to the realisation of heterogeneous IoT is suggested.

e_software_2017] Perhaps a more complete IoT architecture is proposed, where the authors apply SDN principles in IoT heterogeneous networks.

waredefined_2017] it provides the SDWSN with a proper model of network management, especially considering the potential of heterogeneity in SDWSN.

waredefined_2017] We conjecture that the SDN paradigm is a good candidate to solve the heterogeneity in IoT.

Management architecture	Management feature	Controller configuration	Traffic Control	Configuration and monitoring	Scapability and localization	Communication management
[luo_sensor_2012] Sensor Open Flow	SDN support protocol	Distributed	in/out-band	✓	✓	✓
[costanzo_software_2012] SDWN	Recycling, aggregation, routing	Centralized	in-band	✓		
[galluccio_sdnwise_2015] SDN-WISE	Programming simplicity and aggregation	Distributed	in-band		✓	
[degante_smart_2014] Smart	Efficiency in resource allocation	Distributed	in-band		✓	
SDCSN	Network reliability and QoS	Distributed	in-band		✓	
TinySDN	In-band-traffic control	Distributed	in-band		✓	
Virtual Overlay	Network flexibility	Distributed	in-band		✓	
Context based	Network scalability and performance	Distributed	in-band		✓	
CRLB	Node localization	Centralized	in-band			
Multi-hope	Traffic and energy control	Centralized	in-band			✓
Tiny-SDN	Network task measurement	-	in-band			

Table 3.13. SDN-based network and topology management architectures. [**ndiaye_software_2017**]

6 Blockchain

6.1 Application

Blockchain Layers

- Transaction & contract layer
- Validation layer (forward validation request)
- Block Generation Layer (PoW,PoC, PoA PoS, PBFT)
- Distribution Layer

Consensus algorithms
- Proof of Work (PoW)
- Proof of Capacity (PoC)
- Proof of Authority (PoA)
- Proof of Stake (PoS)
- Proof of Bizantine Fault Tolerant (PBFT)

6.2 Summary and discussion

4 | Fuzzy C-Means Clustering

"la patience fait fondre le marbre" – Amer

Contents

Abstract

Long Range Wireless Access Network (LoRaWAN) emerged as one of the promising Low Power Wide Area Networks (LPWAN) for IoT applications. It allows end-devices to reach the gateway and then a core network through a star topology in a wide area. LoRa Transceivers send data packets according to a configuration or set of parameter's values: Spreading Factor (SF), Bandwidth (BW) and Coding Rate (CR). These parameters shall be fixed or adapted to application's requirements. Adaptive Data Rate (ADR) control system of LoRaWAN has been proposed to adapt modulation parameters dynamically based on the recent received packets. However, ADR doesn't adjust parameters considering the evolution of network's Quality of Service (QoS) metrics. In this paper, we propose to adjust parameters' values referring to measured metrics for each transmitted packet such as Bit Error Rate (BER), ToA and Received Signal Strength Indication (RSSI). We consider the set of configurations as a cloud of dots while measured metrics are dots' coordinates. We assume that a dots' cluster characterizes a suitable configuration of particular class of application. Furthermore, cluster heads are the representative settings of a given cluster. We applied Fuzzy C-Means (FCM) clustering algorithm using network's parameters and metrics to identify cluster heads and then extract suitable configurations. Simulations have been performed showing the impact of selected cluster heads.

1 Introduction

Knowing the heterogeneity of services and applications that need to be loaded in the IoT, and knowing the heterogeneity of wireless network configurations, the task to adapt at each time the

wireless network to applications running on each end-device became challenging. IoT applications need more and more wireless technologies that can offer low-cost and low-complexity to end devices to be able to communicate in wide areas IoT end devices are generally powered by battery to allow mobility. For this reason, the power consumption profile should be carefully studied in order to extend the battery lifetime. The communication range needs to achieve several kilometers, as end-devices are distributed in a large area like building and agricultural fields. Many LPWAN technologies are already present in the market like SigFox, Narrow Band-Internet of Things (NB-IoT) or LoRaWAN. SigFox plans to offer global coverage in 45 countries and regions by a single operator network [117]. NB-IoT is built by telecommunication companies as an alternative to sub-GHz LPWAN technologies. As NB-IoT uses licensed spectrum, it offers better traffic reliability compared to other sub-GHz technologies.

Unlike SigFox and NB-IoT, LoRaWAN could be deployed as a private network and integrated easily with many network platforms (e.g., The Things Network (TTN)). In addition, LoRaWAN specification is open access. Due to all of these advantages, since the first appearance of LoRaWAN in the market, the research community started working on it in 2015. Since then, many research papers [118] [119] have been submitted in different journals and presented in conferences all over the world. For that purpose, we use in our work LoRaWAN network and propose a new framework to make the ADR control more flexible by taking into consideration applications' requirements. In this work, we start by generating the required performance metrics of each LoRa transmission configuration, next we apply the FCM algorithm on this metrics to get the membership of each configuration to the three type of applications, in the end, we compute the rand index of the clustering process to get the accuracy membership values.

To find a set on LoRa transition settings that best fit each application requirement, our approach is to use FCM algorithm. The advantage of using this algorithm to solve our problem is the ability to know at which level LoRaWAN transmissions setting is suitable for different type of IoT applications, Rather than using traditional hard clustering process that can only generate labels to know at which cluster an object belongs to. The idea of using a fuzzy clustering algorithm is to know the extent to which an object belongs to his clusters based on their membership values. Knowing these values is important for the network server to rank them and attribute the corespondent setting with high a membership value to end-devices.

This paper is organized as follows. Section 2 elucidates summary of related works. Section 3 introduce the FCM algorithm. In Section 4, we describe how FCM algorithm could attribute membership values to LoRa transmission settings. We explain how this approach could easily be used by LoRa network servers. Our simulation settings is presented in Section 5. Our findings are presented in Section 6. Section 7 concludes this paper.

2 Related work

Selecting transmission parameters of wireless transceivers to reduce energy consumption and increase the QoS is a challenging research area actually.

A large amount of research in Wireless Sensor Network (WSN) investigate transmission power control to reduce transmission energy consumption (examples are [8], [9]). Transceivers used for WSN only provide transmission power as means to adapt the energy consumption. Existing solutions to adjust transmission power depend on data transmissions. Link quality is either determined by computing the BER over time and/or by estimation using RSSI or Link Quality Indicator (LQI). Depending on the link quality time t, transmission power is adjusted for t+1. We follow in our work the same idea regarding this approach. However, LoRa transceivers provide additional parameters to adapt communication energy cost which we take into account. Previous work on WiFi and cellular networks has investigated either i) transmit power control (e.g. [10], [11], [8]), ii) transmit rate control (e.g. [12], [13]), or iii) a combination of the two as 'transmit power and rate control' (e.g. [14], [15].

Most of the transmission power control are concerned with increasing the capacity, and not only decreasing the energy consumption. The transmission rate control is often only concerned with maximizing throughput. Compared to LoRa, WiFi packet rates are significantly higher, and the ADR control algorithms run at a much higher rate then in LoRa. For example, the most commonly used transmit rate control algorithm Minstrel [120] evaluates its links every 100 ms.

Reynders et al. [42] evaluated Chirp Spread Spectrum (Proprietary) (CSS) and Ultra narrow band (UNB) networks. They proposed a heuristic equation that gives BER for a CSS modulation as a function of SF and Signal Noise Rate (SNR). Cattani et al. [75] evaluated the impact of the LoRa physical layer settings on the energy efficiency and data rate. They evaluated the impact of environmental factors such as temperature on the LoRa network performance, they showed that high temperatures decrease the Packet delivery ratio (PDR) and RSSI. Goursaud et al. [57] studied

the performance of the CSS modulation. They showed the interference between different SFs and evaluated co-channel rejection for all combinations of SFs. Feltrin et al. [55] discussed the role of LoRaWAN for IoT and showed its application to many use cases. They considered the effect of non perfect orthogonality of SF for a link level analysis.

All previous work didn't take into consideration the application requirement in their data rate control. The closest study to our approach is that presented in [121]. However, the FCM algorithm was applied to get membership values of citizens to political parties in America. In this paper, we study the membership value of each LoRa transmission setting to 3 different applications, we assume in this paper that the 3 applications have 3 different QoS requirements.

3 Use case example

To adapt LoRa modulation settings dynamically, LoRaWAN network server like TTN change the modulation settings of end-devices based on the 20 recent packets received. Actual LoRaWAN network servers available on the market could adapt the required settings based on different QoS metrics: RSSI, BER and ToA. As the need of high or less QoS strongly depends on the requirements of applications running on end devices, LoRaWAN network servers should be able to rank each LoRa modulation setting to each kind of application, for example, if the application running on end-devices should send segment of sounds to measure noise in a construction site, the network server should select the required settings from a pool or cluster of settings with high RSSI, low ToA and low BER. LoRaWAN network servers should be able to rank transmission settings based on their membership to different kind of applications. In our we assume that we need to run 3 type of application, each of them has different QoS requirements,

Applications	Packet rate [pkt/day]	Min success rate [Ps,min]	Payload Size [Byte]
Wearables	10	90	10-20
Smoke Detectors	2	90	10-20
Smart Grid	10	80	10-20
Waste Management	24	60	10-20
Smart Bicycle	192	80	50-100
Animal Tracking	100	70	50-100
Environmental Monitoring	5	90	50-100
Asset Tracking	100	90	50-100
Water/Gas Metering	8	70	100-200
Environmental Data Collection	24	80	100-200
Medical Assisted Living	8	90	100-200
Safety Monitoring	2	95	100-200

Table 4.1. Application requirements in IoT [55] [116] [97]

Based on different studies [55], [116] and [97], we built a table of different requirements of IoT applications. The QoS metrics discussed in this paper are RSSI, ToA and BER. We describe in detail these metrics in the following subsections.

3.1 Reception Sensitivity

This metric considers the receiver sensitivity criteria. It describes the received signal sensitivity from LoRa end-devices,

$$\mathbf{RSSI_{[dBm]}} = -174 + 10\log_{10} BW + NF + SNR \tag{1}$$

3.2 Bit error rate

This metric considers the reliability of communication criteria. It describes the extent to which the transmitted data is fair at the reception side.

$$\mathbf{BER} = \frac{8}{15}.\frac{1}{16}.\sum k = 216{-}1^{k}(\frac{16}{k})e^{20.SNR(\frac{1}{k}-1)} \tag{2}$$

3.3 ToA

This metric considers the transmission delay criteria. It describes the time in takes for one packet to reach the destination.

$$\mathbf{ToA_{[s]}} = \frac{2^{SF}}{BW}\left((NP + 4.25) + (SW + G)\right) \tag{3}$$

where:

$$G = \max\left(\left\lceil \frac{8PS - 4SF + 28 + 16CRC - 20IH}{4(SF - 2DE)} \right\rceil (CR + 4), 0\right)$$

These three metrics have a high influence on applications. To study th

4 Approach

To describe the FCM algorithm, let us begin with some notations. Let n be the number of all LoRa transmission settings. Let p be the number of QoS metrics. $X = [x_{11}, ..., x_{np}]$ is a set of p measured QoS metrics of n settings with $x_{ik} \in \mathbb{R}, 1 \leq k \leq p, 1 \leq i \leq n$. The FCM algorithm takes as input X and generate tow sets: U and V. $U = [u_{11}, ..., u_{nc}]$ is a set of membership values of n settings to c clusters with $u_{ij} in \mathbb{R}$. $V = [v_{11}, ..., v_{cp}]$, is a set of cluster centers of p metrics to c applications.

$$[\mathbf{U}] = \begin{array}{c} \\ \textbf{setting 1} \\ \textbf{setting 2} \\ \vdots \\ \textbf{setting n} \end{array} \begin{array}{c} \begin{array}{ccc} \textbf{Cluster 1} & \textbf{...} & \textbf{Cluster C} \end{array} \\ \begin{bmatrix} u_{11} & \dots & u_{1c} \\ u_{21} & \dots & u_{2c} \\ \vdots & \ddots & \vdots \\ u_{n1} & \dots & u_{nc} \end{bmatrix} \end{array}$$

To get the membership value of each setting to different kind of applications, we use the Equation 4 to update at each iteration the membership values.

$$\mathbf{u}_{ij} = \left[\sum_{j'=1}^{c} \left(\frac{d_{ij}}{d_{ij'}} \right)^{\frac{2}{m-1}} \right]^{-1}, \forall j, i \sim \mathbf{U}_t = F_\partial \left(\mathbf{V}_{t-1} \right) \tag{4}$$

Cluster centers are the optimal measured metrics that are close to all the measured metrics of each cluster and are computed using Equation 5.

$$\mathbf{v}_j = \left[\frac{\sum_{i=1}^{n} u_{ij}^m \mathbf{x}_i}{\sum_{i=1}^{n} u_{ij}^m} \right], \forall j \sim \mathbf{V}_t = G_\partial \left(\mathbf{U}_{t-1} \right) \tag{5}$$

4.1 Objective function:

The objective of the FCM algorithm is to find a set of membership values U and a set of cluster centers V that minimize the objective function in

$$\min_{(\mathbf{U},\mathbf{V})} \left\{ M_m(\mathbf{U}, \mathbf{V}) = \sum_{j=1}^{c} \sum_{i=1}^{n} u_{ij}^m d_{ij}^2 \right\} \tag{6}$$

such that:

$$sum_{j=1}^{c} u_{ij} = 1, \forall i \tag{7}$$

$$d_{ij}^2 = \left\| \mathbf{x}_i - \mathbf{v}_j \right\|^2 \tag{8}$$

$$m \geq 1 \tag{9}$$

4.2 Performance Index:

$$\min_{(c)} \left\{ P(c) = \sum_{j=1}^{c} \sum_{i=1}^{n} u_{ij}^m \left(\left\| \mathbf{x}_i - \mathbf{v}_j \right\|^2 - \left\| \mathbf{v}_j - \overline{\mathbf{x}} \right\|^2 \right) \right\} \tag{10}$$

where:

$$\overline{\mathbf{x}} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i \tag{11}$$

Our simulation has 2 components. The first component is a LoRa transmiter and receiver tools winch represent the end-devices. To generate the QoS metrics offred by each setting, we transmit packets with different parameters presented in Table 4.2, it converts the quantitative values of LoRa transmission performance to qualitative values that represent level of membership to each Fuzzy Logic (FL) subsystem.

The second component is the FCM algorithm. It aims to update the values of the membership matrix. Membership matrix is a set of n setting over k applications, their values represent the membership of each setting to each application. When the value is equal to one, this means that the corespondent setting fits well the application that belongs to.

Algorithm 1: FCM

Input: $V = [v_{11}, ..., v_{cp}]$
Output: $(\mathbf{U}, \mathbf{V})$

```
t = 0
while ||v_i − x̄|| ≥ e or t ≤ T do
    t = t +1
    U_t = F_∂ (V_{t−1})
    V_t = G_∂ (U_{t−1})
(U, V) = (U_t, V_t)
```

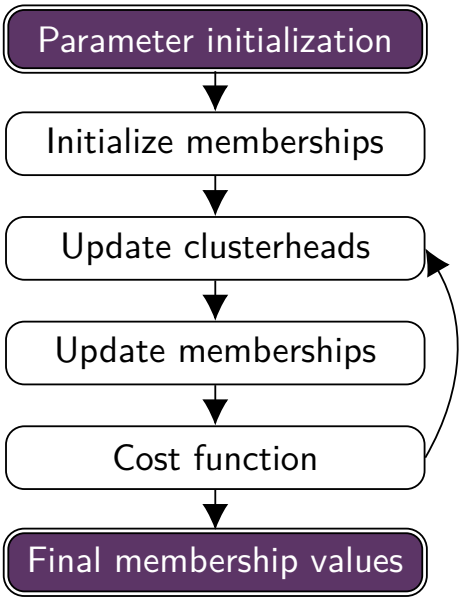

Figure 1. Clustering process.

5 Simulation settings

In order to generate all the required metrics to cluster LoRa transmission settings, we use all the combination of SF, BW,Payload size (PS) an SNR presented in Table 4.2

Setting	Values
$BW_{[kHz]}$	[125,250,500]
$SF_{[\#]}$	[7,8,9,10,11,12]
$PS_{[B]}$	[59, 230]
$SNR_{[dbm]}$	[-40,-30,-20,-10,0]

Table 4.2. LoRa transmission parameters

The FCM algorithm is coded in python programming language. The dataset is generated with matlab programming language and used to test the feasibility and effectiveness of FCM. Our experiments were conducted on a PC with i7 Intel CPU with 8G RAM. For our clustering process, we used a degree of fuzzification of 1.2.

6 Results

From the simulation results of the FCM algorithm, we observed different interesting facts, most of them are presented in this section.

As the aim of the fuzzy clustering process is to measure the membership values of each settings to the three clusters, we run the algorithm until the objective function returns values less than a threshold. A sample of membership values of the FCM algorithm is presented Table 4.3. From this table we can see that settings with high RSSI, low BER and ToA are clustered in the same cluster (C0), We can see also that settings with high SF and low SNR are mapped to the cluster, which proves

BW	SF	PS	SNR	BER	RSSI	ToA	C_0	C_1	C_2
125	11	30	-20	0	137	0.39	**0.91**	0.045	0.045
125	7	10	-10	0.005	127	0.02	0.015	**0.492**	**0.492**
125	11	70	0	0	117	0.46	**0.492**	**0.492**	0.015
250	12	70	-20	0.003	137	0.92	**0.734**	0.153	0.113
250	11	10	-10	0	127	0.33	0.104	**0.791**	0.104
250	12	90	-20	0	134	0.46	**0.965**	0.004	0.030
500	7	50	-20	0.050	131	0.00	0.003	0.030	**0.965**
500	12	10	-20	0	131	0.16	0.003	**0.965**	0.030
250	12	110	-20	0.009	134	0.52	**0.469**	0.061	**0.469**
500	12	110	-20	0.001	131	0.26	0.113	**0.734**	0.153

Table 4.3. Samples of membership values of LoRa transmission settings to 3 different applications

that the LoRa modulation is resilient against noise. Settings with low RSSI, high BER and ToA have a high membership values to cluster (C2). Based of these values, the LoRaWAN network server could easily rank settings that belong to the same cluster assign the best one to end-devices.

One can see from Table 4.3 that most of the settings have a heavy membership for exact one application and light memberships for other applications. For example, the setting BW125, SF11, PS30, SNR-30, has a membership of 0.91 for cluster 0 and 0.045 for others, which implies that it is the best choice of applications of cluster 0. The third setting in this table has two equal light memberships for two different clusters, so it can be assigned to either of the two clusters.

Cluster	BER	RSSI	ToA
0	1	75	7
1	49	47	58
2	67	24	76

Table 4.4. Cluster centers

Table 4.4 represents the final cluster centers of the FCM algorithm. The cluster centers of each cluster represent the normalized values of the three measured QoS metrics. The cluster center of (C0) has the lowest BER and ToA and highest RSSI which represent well the cluster with hight QoS requirements. In the other side, the cluster center of (C2) has the lowest RSSI and the highest BER and ToA which represents the cluster with low QoS requirements

Performance metrics:	**value**
time (s):	0.0102
completeness:	0.2105
v-measure:	0.3466

Table 4.5. Clustering performance

The clustering performance of applying the FCM algorithm on LoRa transmission setting are presented in Table 4.5.

Figure 2 Figure 3 and Figure 4 resume the clustering results of the FCM algorithm. Settings are assigned to their cluster based on their membership values.

Figure 2 shows the relationship between the RSSI and BER of LoRa transmission settings, results show that settings with a hight membership values to cluster C0 have RSSI between -135 dBm and -110 dBm, and BER less than 0.02%. Cluster 2 has lower RSSI compared to the tow other cluster and also the highest BER, Cluster 1 could be used for application with hight sensitivity to the BER and less to RSSI

Figure 3 shows the relationship between BER and ToA of different settings. The same settings of cluster c0 which are presented in Figure 2 have the lowest ToA which make them suitable for applications with high sensitivity to time transmission. Settings of cluster C2 which seem to have the same QoS as settings of cluster C1 in Figure 2 have a higher ToA, which make them not suitable for applications with high sensitivity to the delay.

Figure 4 shows the relationship between ToA and RSSI. As said previously,

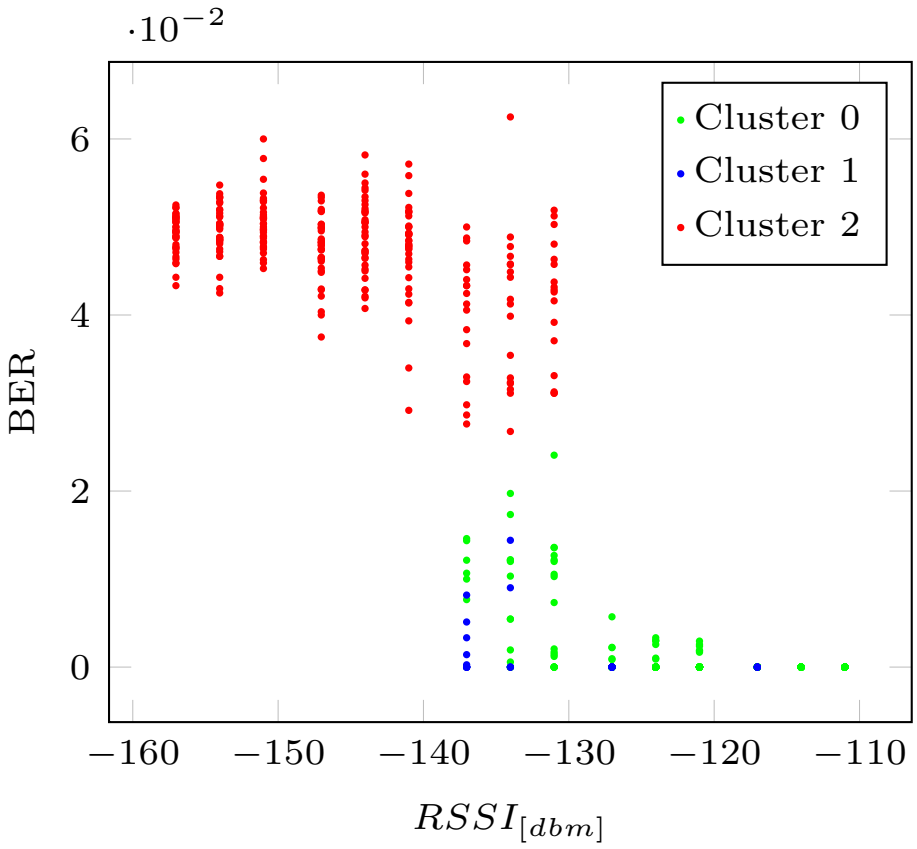

Figure 2. BER vs RSSI.

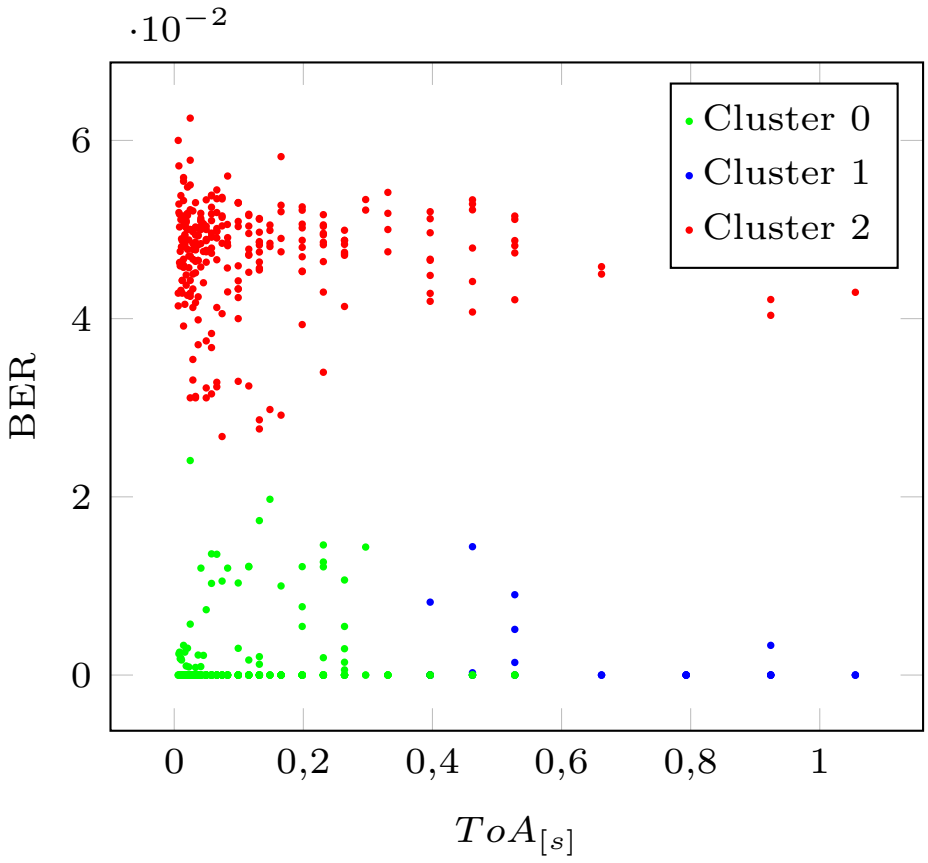

Figure 3. BER vs ToA.

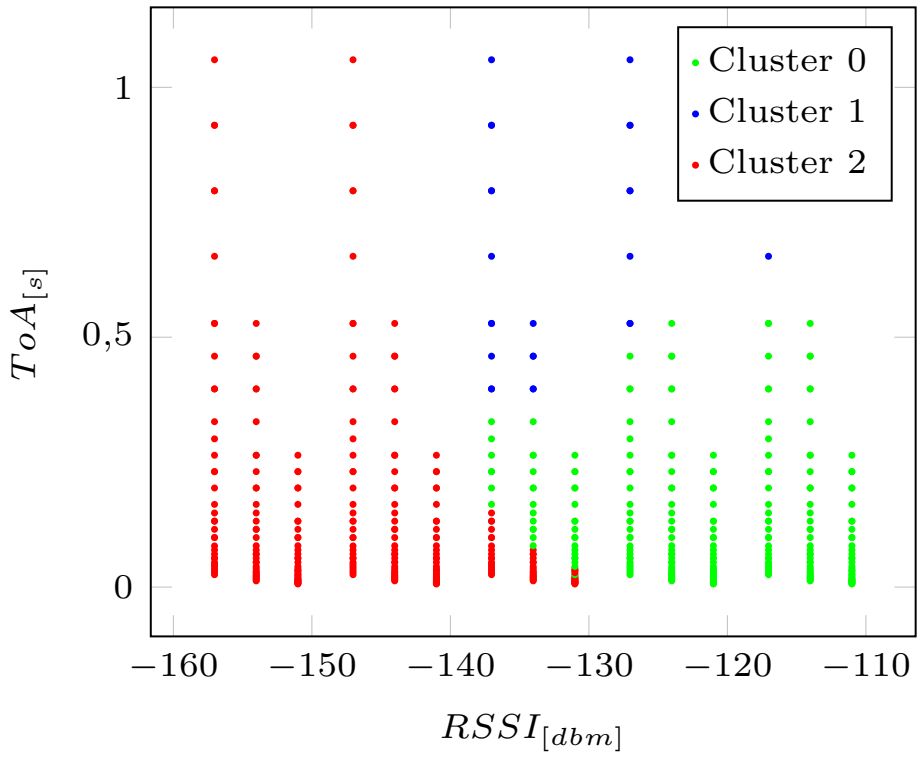

Figure 4. ToA vs RSSI.

7 Conclusion

The main challenge of this work was to build a framework that can easily be plugged in LoRaWAN network servers with less complexity, Our main contribution was to test the effectiveness of applying FCM algorithm to select the transmission setting that best fit a given application requirement. The framework can cope with the large number of possible settings and conflict view points and purposes of applications requirements, The proposed framework has been developed to present and design a solution that consider LoRa parameters, environment conditions and performance metrics required by applications. In this paper we used the FCM algorithm to compute the membership values of each configuration to 3 kinds of applications that require different QoS. Simulation results have shown that FCM algorithm is very effective in identifying the inherent settings that offers the required performance metrics. In addition, FCM could rank the available settings based on their membership values. The network server could then select the configuration with a higher membership value to the application running on end-devices.

5 | Testbed

"Human identity is no longer defined by what one does, but by what one owns. This is not a message of happiness or reassurance, but it is the truth and it is a warning." - Jimmy Carter

Contents

Abstract

1 Introduction [bregell_hardware_2015]

1.1 Problem Statement

1.2 Background

Internet of Things (IoT) is a concept aiming at connecting all things to the Internet [1]. The different kinds of devices range from simple sensor devices to complex machines such as industry

robots. Home automation has been available for a few years in the forms of timers and remotely controlled devices, such as lights, garage door, and climate control equipment. Also in the industry and workplace, there are current systems that have some of the functionality of IoT, e.g, sensors in robots and machines which keep track of the system status so that maintenance can be scheduled at the right time. However, these systems or sensors rarely communicate with each other or make decisions based on other sensor values; instead they depend on input from a user. In the same way cellphones connected people and made them constantly connected to the Internet, IoT will connect devices and make them constantly connected to the Internet [2]. In theory, this could lead to a future with autonomous technology all around us. The benefits could be huge as it would save time and energy for both the individual at home and for the industry [3]. IoT could be used in industry to automate power-heavy tasks to run when the electricity price is low. This principle can also be applied for the home user with laundry machines and charging of e.g. electric cars. This practice would lead to reduced energy consumption and thus a reduced environmental footprint. i3tex AB wants to investigate potential fields of applicability of this upcoming technology. i3tex AB has customers in the automotive, communication, and pulp industries; those customers have made inquiries on how to integrate IoT and sensor networks into production. As technology evolves, size and energy consumption of the IoT devices will decrease and computation power will increase [4]. This reduction in size and energy consumption, together with the increased computing power, will open up new fields for IoT. Thus, i3tex AB want to have an IoT platform to present to their current and potential customers. The interest in IoT is rapidly increasing, and thus, in the near future, the number of devices connected to the Internet is expected to increase rapidly. To support this huge increase in both number of connected devices and the sheer amount of data that will be sent over both wired and wireless networks, the communication technology must be ready [5].

1.3 Purpose (Goal)

The purpose of this project is to find and examine a communication method for devices that are made to be a part of IoT. This will be done by examining the available technologies and then developing a prototype based on the findings, which will be used for examining the communication method. This project will examine the physical, link, and network layers [6, 7] of the Open Systems Interconnection model (OSI model) [8], in order to find suitable technologies on the market. As IoT is still only defined as a concept, there are several technologies to take into consideration and examine in further detail. The prototype will be delivered to i3tex AB together with appropriate documentation, e.g. technical specification, hardware manual, software manual, and API specification.

1.4 Limitations

To be able to achieve the project goal within the available time, limitations need to be defined in the three main areas of: Operating System (OS), hardware, and communication method. The OS will not be custom-made, but rather selected amongst those already on the market. Thus, to simplify the hardware selection, only those OSs which already have hardware support that meets the requirements will be taken into consideration. Furthermore, support for either 6LoWPAN, ZigBee, or Bluetooth Low Energy (BLE) as communication method is required, since development to make those standards available is outside the scope of the project. On the hardware side, the limitations will be to only use existing devices and parts as there will be no time for developing hardware or Printed Circuit Boards (PCBs). However, the hardware does not need to have an integrated radio transceiver, but needs to support at least one transceiver supporting IEEE802.15.4 [6]. Thus, communication methods will be primarily selected from specifications building on the IEEE 802.15.4.

1.5 Method

To ensure that the right technologies were selected and investigated, the first phase of the project was a literature study. The study served as a foundation when developing and performing the evaluation of the communication methods. At the end of the phase, a requirements specification was formulated to serve as a platform for the next phase. After the literature study, a selection process was performed, where the most promising technologies that met the requirements were examined in further detail and brought into the development phase. This process included the selection of development tools and other decisions bound to the product development. In the development phase, the chosen set-up was configured and assembled to prepare for testing; it was then tested according to throughput, range, latency, and energy consumption. Throughput was measured in kilobyte per second (KB/s) and tested by transferring data of different sizes in both congested and uncongested network set-ups to simulate real world and lab environments. The same set-up was used to measure the latency of a transmission, which was measured in microseconds (ms). Range was calculated in-

stead of measured, with meters (m) as the unit. The power consumption was measured in watts (W). Each week a meeting with the company supervisors was performed, to keep the work on the right track. Here, feedback was be given and other issues and questions handled.

2 Related work

3 Background

The goal of the implementation phase is to have a working prototype for future assessment. To make the process of implementing the prototype possible, the first part of the implementation process will be to create a set of requirements. When these are set, the process will continue by comparing the data from chapter 2 to find the candidates that fulfil the requirements. After the technologies are selected, the process will continue with setting up the workspace, which includes the platform for development and the required tools to build, debug and test the prototype. Finally, when these three steps have been performed, the next step will be to start with the actual prototype development.

Requirements

As the time dedicated for development is limited, the requirements have to make sure that the development process does not run into any major obstacles. All parts of the total prototype need to fit together seamlessly. However, the hardware platform and the operating system are tied most closely together. Therefore, they need requirements that complement each other, so that they can act as a platform for software development. Naturally, both the hardware and the operating system requirements might have to be altered slightly to enable the best match.

3.1 Hardware

Each platform examined in chapter 2 has different strengths and weaknesses. When looking at the MCU, OpenMote has a ARM Cortex-M3 which is more powerful compared to the other two alternatives: it features a 32bit 32MHz core with 32KB RAM and 512KB flash memory compared to the MSP430 16bit 25MHz core with 10KB/100KB memory configuration. In terms of peripherals all three platforms are comparable, with similar amount of DAC, GPIO pins, and external busses. All of the platforms have a temperature sensor and an accelerometer, but OpenMote also features an light/uv-light sensor and a voltage sensor built into the MCU's ADC. The MSB430 platform has a somewhat lower power consumption in active RF mode thanks to the less power-hungry sub-GHz transceiver. On the other hand, the less powerful MSP430 MCU has a better deep sleep power consumption, but as the radio is not integrated in the chip as it is in the CC2538 SoC, that advantage is offset by the external transceiver. Comparing the transceivers, there are two 2.4GHz models and one subGHz model; the sub-GHz CC1100 has a higher transmit power of 10dBm compared to 0dBm for CC2420 and 7dBm for CC2538. Also the sensitivity is similar for all the alternatives but gives a slight advantage to CC2538 with -97dBm compared to -95dBm and -93dBm for CC2420 and CC1100. Using these numbers and Friis range equation (equation 4.1) the range of each transceiver with a Fade margin (FM) of 20dB can be seen in figure 3.1. The benefits of working with lower frequencies can clearly be seen as the theoretical range of CC1100 is almost 3 times longer than the 2.4GHz transceivers.

Adding all this information together, the choice of platform will land on OpenMote with the CC2538SoC. It both has a MCU with more memory and better performance, and a transceiver with really good characteristics both in terms of energy consumption and range. Also OpenMote is the only option that can act as a border router using OpenBase; it lets the SoC interface with USB, UART, JTAG, and Ethernet, which enables the standalone border router mode without the need to be connected to a computer or other hardware. The OpenBattery extension lets the SoC operate as a node in a mesh network and provides a dual AAA battery slot connected to the PCB.

Resume

• Transceiver with IEEE 802.15.4 or IEEE 802.15.1 • Integrated sensor/sensors • MCU with low power mode under 5MicroA • Wakeup from low power mode with timer • Border router ability • Joint Test Action Group (JTAG) support

3.2 Operating system

As a modern operating system can be compiled to match almost any hardware, the most important thing to have in mind is the out-of-the-box hardware support. Only RIOT and Contiki have full support for the ARM Cortex-M3 of the considered operating systems and thus both TinyOS and freeRTOS are directly eliminated as developing the support would take too much time. Compared to

RIOT, Contiki also has full driver support for the sensors and transceiver, which should decrease the implementation time significantly. When compiled into binary form, RIOT uses less RAM and ROM and thus probably is a bit faster compared to Contiki, which could be important if the application consumes much resources. The lower memory usage might also give RIOT an advantage in being future-proof. Contiki also has support for soft real-time scheduling compared to the hard real-time scheduling of RIOT; this is however not crucial, as the software that will be running on the OS does not have any hard real-time constraints. Both RIOT and Contiki have support for 6LoWPAN but no support for either ZigBee or BLE; this is due to the fact that these are proprietary stacks. Support could be added but would take some time to customize for the given OS. What gives Contiki the largest advantage is that it also have border router software ready for deployment, which in the RIOT case would have to be developed. All in all, as the project has such a limited time frame, Contiki will be selected as the OS; this, mainly because Contiki comes with most advantages time-wise; this choice means that the focus of the software development will be the creation of a test and evaluation system.

Resume

- Support for the SoC/MCU and transceiver • 6LoWPAN, ZigBee or BLE stack • Soft real-time
- RAM and ROM footprint matching the hardware

3.3 Communication protocol

The OpenMote platform has a IEEE 802.15.4 transceiver and thus supports both ZigBee and 6LoWPAN; this means that BLE is not an option. As ZigBee does not have full IPv6 support yet and is not integrated into Contiki, the natural choice is 6LoWPAN. This choice will not only save some development time but also enables evaluation of the header compression. As seen in figure 3.2, the 6LoWPAN stack in Contiki will replace the IP stack while maintaining the same functionality. As the functionality is the same, TCP and HTTP will work with 6LoWPAN, but including them in the source increases the OS build size considerably. On top of the UDP layer, Contiki also has a working implementation of CoAP that can be used for retrieving data from the nodes in a power efficient manner. CoAP is a stateless protocol that uses the HTTP response headers to achieve a very low overhead in transmissions while using application level reliability methods to ensure packet delivery.

Resume

- IPv6 addressing support • Existing OS support • Network type is mesh • UDP

3.4 Workspace and tools

The ARM Cortex-M3 chip that OpenMote and CC2538 is built upon requires the GCC ARM Embedded compiler. This tool-chain is free and runs on both Linux, OSX, and Windows; however, there is no bundled development application so a secondary application for programming is needed. In Windows, there are several Integrated Development Environments (IDEs) such as IAR Workbench ARM [30], Code Composer Studio [31] and the Eclipse plug-in ARM DS-5 [32, 33]; these IDEs use various proprietary toolchains and have a price tag ranging from free to several thousand SEK. Most of the IDEs also have a code size restriction for the free versions. To minimize the costs, the development machine development machine used in this project will run Ubuntu 14.04 LTS, the used tool-chain is GCC ARM Embedded, and Geany is used as the code development application. To analyse the network traffic in real-time, the open source tool Wireshark is used together with a IEEE 802.15.4 packet sniffer. Together with a laptop, the packet sniffer will grant the ability to traverse the mesh network and analyse the network in real-time as it is seen by the nodes.

4 Proposed ...

The goal of the development process is to have a functional border router and at least two nodes to be able to test how response time and throughput differs with each hop in a mesh network. To be able to measure response time and throughput, each node needs to have a CoAP server which can respond to ping and also receive an arbitrary amount of data for throughput measurement. It is desirable for each node to be able to send information about each sensor so the project can be used as a tech-demo. The first part of the development was to set-up of the workspace and tools mentioned in section 3.1.5. Ubuntu OS was installed in a VirtualBox Virtual Machine to make it easier to duplicate and backup; this procedure gave a noticeable decrease in performance and it is recommended to have a dedicated native Ubuntu machine for this type of development. Even though Ubuntu uses an easy-to-use package system, there were some problems in finding a version of GCC ARM Embedded tool-chain that was compatible with Contiki's built-in simulator Cooja [34, 35]; eventually, version

4.82 was used to successfully build Contiki. Cooja is a useful tool for testing and debugging network configurations but does not have support for the CC2538 MCU; instead, nodes called Cooja Motes are simulated with generic hardware. As Cooja is written in Java and runs in a JVM, Oracle Java 1.8 was also installed.

4.1 Drivers and firmware

Figure 3.3 shows an overview of the full system. The foundation is the SoC with the MCU, transceiver, and sensors. The Contiki operating system implements the soft real-time kernel together with the firmware for the SoC/MCU and the drivers for the peripherals and sensors. The last part is the communication stack, which provides TCP and UDP connectivity over 6LoWPAN. On top of the TCP and UDP protocols, HTTP and/or CoAP can be implemented. The firmware required for the OS to work properly on the hardware platform was already implemented. However, the drivers for the I 2 C bus and the sensors were not implemented. The I 2 C driver is required for the sensor drivers which in turn enables the MCU to communicate with the sensors on the OpenBattery platform.

4.2 CoAP server

In order to make each node's sensor data accessible, a CoAP sever was implemented as an application running on top of Contiki. A CoAP server in general can handle any number of resources; in this implementation, one resource was made for each sensor value i.e. temperature, light, humidity, and core voltage. The temperature, light, and humidity sensors all work in a similar fashion. When their value is requested, the I 2 C bus is initialized and then a request is sent over the bus. When the response with the value arrives, that data is put into either a plain-text or JavaScript Object Notation (JSON) formatted message depending on the request and then sent back to the requester. As the core voltage sensor is part of the MCU's ADC, that value is retrieved by simply getting data from a register (a somewhat faster operation). As a buffer for testing throughput speed was also needed, a resource with a circular buffer was implemented. This resource is configured with CoAP's block-wise transfer functionality for arbitrary data size; however, the buffer in itself is only 1024Kb to allow the program variables to fit into the ultra low leakage SRAM. For testing purposes, the data could have been discarded instead of actually saved into the buffer, but then the transfer can not be verified. Resources are defined by paths as CoAP works in a very similar way as HTTP.

Each resource is registered in the server with its path, media type, and content type. When a package arrives on the CoAP port, the server starts to break down the package to be able to direct it to the right resource. It starts with verifying that the package is actually a CoAP package, and then it checks the path and sends it to the correct resource. The resource then inspects the method field in the package header to direct the incoming data to the right function. CoAP package method can be either GET, PUT, POST or DELETE. This function then inspects the request media type and answer content type so that the function can parse the request and send a correctly formatted answer. If the resource does not implement the received method, the server responds with "405 Method not allowed" and if the content/media type is not supported the answer is "415 Unsupported Media Type". The content/media types are text/plain, application/json, application/exi, and application/xml.

4.2.1 Testing

Contiki is shipped with a simulation tool called Cooja which is written in Java; it can simulate an arbitrary number of nodes with different roles and configurations. All simulation data, such as radio packages and node serial output may be viewed through different windows and exported to various formats. Unfortunately Cooja did not have support for ARM Cortex-M3, but the general set-up was still tested by using Cooja Motes, which are nodes without specified hardware, and MSP430 nodes such as Wismote or Skymotes. With this simulator the basic understanding of the communication between nodes was gained; also, before the hardware arrived, early testing was performed to test the OS and application software.

4.2.2 Final prototype

The final prototype consists of four OpenBattery nodes and one OpenBase border router. Both the nodes and the border router are deployed with Contiki. Each node runs a CoAP server, described in section 3.2.2, on top of the OS in its own thread. The border router runs a router software called 6lbr that acts a as translator between Ethernet and IEEE 802.15.4 [36]. Both types of hardware are configured with a 8Hz Radio Duty Cycle (RDC) driver to keep the power consumption to a minimum. RDC is a OS driver that cycles the listening mode of the transceiver to reduce power consumption. As Contiki puts the MCU into Low Power Mode (LPM) when no function is running and the transceiver is off, the RDC driver indirectly controls when the MCU is in LPM. When using the RDC protocol,

the nodes repeatedly send messages until the target node wakes up and sends an Acknowledge packet (ACK); this makes communication seamless, even though most of the time the nodes' transceivers are not active. Also, an always-on RDC driver, where the transceiver is constantly listening, will be used to be able to look at the performance impact of the 8Hz RDC.

5 Experimentation

In this chapter the results from each type of assessment are presented. The first assessment is range, followed by response time, after that connection speed, and finally the power consumption. The only assessment that is not performed on the prototype is the range assessment.

5.1 Range

Range is very hard to measure without advanced equipment and isolated rooms but can be roughly estimated with equation 4.1 called Friis range equation [37]. P t is the sender transmit power, P r the receiver sensitivity, d is the distance between the antennas in meters, f is the signal frequency in hertz, and is the wavelength. G t and G r is the antenna gain for the transmitter and the receiver. The last term in equation 4.1, when inverted, is the Free-space path loss (FSPL) and can be expanded as shown in equation 4.3. P r (dB) = P t + G t + G r + 20 log 10 ( ) 4d c f FSPL(dB) = 20 log 10 (d) + 20 log 10 (f) − 147.56 = (4.1) (4.2) (4.3) Unlike Friis range equation, the Link budget equation 4.4 also takes external loss like FM into account [38]. This is needed to make a correct estimation of the actual range as there are several things in the environment that obstructs and distorts the signal. P r = P t + G t + G r − FM − FSPL (4.4) Combining equation 4.1, 4.3 and 4.4 gives us the equation for the estimated distance as seen in equation 4.5. d = 10 x P t + G t + G r − P r − F M + 147.56 − 20 log 10 (f)

With this equation an estimation of the transceiver range can be made for different FMs and transmit powers. When deployed, the transceiver is configured to only accept packages with a signal strength of -70dBm and above to minimize packet loss and corruption. The antenna gain for OpenMote is 0dBi and can thus be omitted. Figure 4.1 shows a comparison between three different levels of FM: 0dB, 10dB, and 20dB. A FM of 0dB means that there is no signal loss except the FSPL and this is very hard to achieve outside of a lab environment. When increasing the FM to 10dB, which corresponds to a normal home environment, the maximum range drops to 22m. However, in these kind of environments the desired range is usually around 10m which would let the device reduce the transmit power to around 0dBm. Finally, the FM is increased to 20dB which is roughly what it would be in a office or industrial environment. The maximum range in this environment is now reduced to only 7m when transmitting at maximum power.

Response time Before measuring the response time, some theoretical estimations are needed to be able to evaluate the real values. The theoretical values are based upon the radio duty cycle (RDC) and the average response time to reach a node can thus be derived from equation 4.6, 4.8, 4.9 and 4.10. As each node only

5.2 Response time

Before measuring the response time, some theoretical estimations are needed to be able to evaluate the real values. The theoretical values are based upon the radio duty cycle (RDC) and the average response time to reach a node can thus be derived from equation 4.6, 4.8, 4.9 and 4.10. As each node only checks the radio every 125ms, this duration combined with the data packet send time of 4ms (equation 4.7.) and ACK send time corresponds to the worst case delivery, as the node needs to wait a whole cycle before being able to send the package the desired node. When the target node is already listening, the best case delivery time is 5ms. Thus, the average theoretical delivery time to reach any adjacent node is 67.5ms. Radio duty cycle: Transfer time: 1s = 0.125s = 125ms 8 (4.6) 133B + 4B 4ms 31.25KB/s (4.7) Worst case delivery: 125ms + 4ms + 1ms (4.8) Best case delivery: 4ms + 1ms (4.9) 130ms + 5ms = 67.5ms (4.10) 2 The delivery time is only calculating the time to send a packet over a link, but when calculating the response time, the acknowledge (ACK) response has to be included in the calculation. Each ACK also needs to wait for the target node to be awake, adding one more instance of average delivery time, resulting in 125ms in average response time. This time will multiply with each hop, resulting in equation 4.11, 4.12 and 4.13. Avg. delivery: Avg. response time: (2 · 67.5ms) · hops (4.11) Best case response time: (2 · 5ms) · hops (4.12) Worst case response time: (2 · 130ms) · hops (4.13) After doing a test with real nodes set-up with a 8Hz RDC with three hops, as seen in figure 4.2, the values in table 4.1 were obtained. Each node was pinged 200 times at a one minute interval to simulate some traffic on the network. What can clearly be seen in the average field of the table is that the average of 765ms is much higher than the expected average of 135ms; the difference is mainly due to the worst-case pings that in some cases had response times

up to 30 seconds. However, when looking at the geometrical mean which is better at smoothing out big spikes seen in figure 4.3, the observed response time is still 265ms which is a bit longer than the expected worst case response time for one hop. Also, for two and three hops the observed average is high, but the geometrical mean shows that this is due to the spikes. The estimated response time for two hops is 270ms which as seen in the geometrical mean table 4.1 is way off by 500ms. The same observation goes for three hops where the observed geometrical mean response time is 1181ms which compared to the estimated response time of 405ms is significantly higher.

With these observations in mind, the estimation could be described much better with equation 4.14. which would result in an average response time of 266, 532 and 1064 ms for one, two and tree hops. However, this would mean that the response time is doubled for one hop and then doubled for each consequent hop making the response time exponential which should not be the case.

With the RDC disabled, i.e. the transceiver is always listening and the MCU does not go into sleep mode, the response time is completely different. As seen in table 4.2 the average response time is around 12ms per hop and the spikes seen in the response time for 8Hz RDC is gone. Furthermore, the estimated best case response time of 10ms is very close to the observed average response time. The response time also scales to the number of hops as expected and is roughly 12ms per hop. It appears there might be a problem in

5.3 Connection speed

Connection speeds can be measured in several ways each with their own different pros and cons. One of the most popular ways is throughput, i.e. the amount of data over the link is divided by the time it took to reach the target. However, this gives a false picture of how fast the connection actually is from the developer's point of view, as the measured data does not only contain application data but also headers and checksums. IEEE 802.15.4 has a theoretical data rate of 250kb/s as seen in equation 4.15. but this is only a measure of how many bits per second the transceiver is able to output. The application data part, when using no header compression, is only 41% of the total transfer. Thus, resulting in a theoretical application data rate, also called goodput, of only 12.81KB/s. Data rate: 250kb/s = 31.25KB/s 133B − 54B = 0.59 133B Theoretical goodput: 31.25KB/s · (1 − 0.59) = 12.81KB/s Overhead: (4.15) (4.16) (4.17) When using CoAP as the application level protocol, each package can carry either 32 or 64 bytes of application data. In practice, the 64B mode is only applicable when sending packages between nodes on the same mesh network, as the addressing fields then can be fully compressed. When using applications outside the mesh network, each package can only carry 32B of data, resulting in a packet size of 111B as shown in equation 4.18; this does not affect the theoretical data rate but has a noticeable impact on the goodput due to the large overhead of 71as shown in equation4.19. To be able to use the full data rate, the application needs to use a protocol without handshakes, i.e. UDP, as the transceiver then can send the packets as fast as physically possible. CoAP is implemented on top of UDP and thus has a low transport layer overhead, but uses its own mechanism for handshaking, delivery and ordering. The theoretical CoAP application throughput can be estimated by looking at the average response time of the node and then add that time to the the data delivery time. Each package needs to be acknowledged before the next package is sent, and thus the node response time needs to be taken into consideration. When doing so, the throughput as calculated in equation 4.20. is only 1.64KB/s and thus the theoretical goodput is reduced from 12.81KB/s down to 0.48KB/s as shown in equation 4.21. Packet size: 133B − 54B + 32B = 111B Actual overhead: 79B = 0.71 111B (4.18) (4.19)

Using the packet size of 111B together with the theoretical response time from equation 4.11 would give the results shown in figure 4.4. To verify these calculations, the same test set-up as shown in figure 4.2, which also was used in section 4.2, was used to test throughput and goodput at different number of hops. Each node was sent 1KB data each minute for 200 minutes; the time from the first package sent to the final acknowledge packet received was measured for each 1KB transmission. The first test was performed with an RDC of 8Hz and resulted in the values shown in figure 4.5. As the chart shows, the theoretical throughput and goodput is much higher than the observed values, but this is due to the fact that the actual average response time is higher than the theoretical one. With some calculations made the observed throughput and goodput are within range of what is expected, given the observed response times in table 4.1. Equation 4.22 uses the observed values to calculate the average response time, given the values in figure 4.5.

5.4 Power consumption

To measure power on devices that use very low power and also changes the power consumption very rapidly and frequently is not an easy task. According to the currency specification from the CC2538, the different power modes have the consumption seen in table 4.3 using the built in voltage

regulator TSP6750 that switches the input voltage down from the 3V to 2.2V. The components on the OpenBattery supplied directly by the 3V batteries have the current and power consumption specifications as seen in table 4.4.

Given these power profiles, combined with the time it takes to receive and transmit packages, and retrieve a measurement, the theoretical power for one RDC cycle results in the chart seen in figure 4.7. The node starts in sleep mode using 112.81W and after 109ms wakes up and goes into RX mode where a request for a sensor value is received. The node then switches off the radio and fetches the sensor value. After the value is retrieved from the sensor, the radio is once again put in to RX mode for a Channel Clear Assessment (CCA) before entering TX mode and sending the payload. The transmission is successful and the node goes into RX mode to listen for the ACK, when it is received the node enters sleep mode again. For this cycle the average power consumption is 4.8mW which would drain the 2250mWh batteries in 19 days.

However, as the nodes have a RDC running at 8Mz, most of the time there will be no package for the node to receive and thus no measuring and transmitting, as seen in figure 4.8. This cycling reduces the average power consumption to 0.47mW, which would make the batteries last for ca 200 days. The goal is to have a node that can run for one year without having to change the batteries and to be able to do this on 2xAAA batteries with 750mAh the average consumption has to be under 257W as calculated in

To verify these assumptions, we used a Keithley 2280S power supply [39] to measure the total current draw of the prototype. The node was connected to the power supply, which was set up to make 277 measures each second with a supply voltage of 3V. Several measurements were performed. One of the most interesting ones can be seen in figure 4.9. In this picture, we can clearly see the different operating modes, as the node performs 3 transmissions during the interval. In the first transmission at the 1.6s mark, the strobing feature of the RDC protocol is seen as the package is sent 5 times before the receiving node is awake and can receive the package. In the two following transmissions, the package is delivered on the first try. As our measurement is limited to 277Hz, the current peaks when only waking up to listen for traffic are sometimes missed, and the peak value is hard to extract; but the 8Hz RDC cycle is still visible. The average power consumption for these cycles is 8mW, which would make the batteries only last for 11 days. However, when taking the average of a measuring series without any transmissions, the average goes down to 4mW, which increases the battery time to 23 days. The theoretical sleep power of 0.11mW compared to the measured of 3mW is what makes the average power consumption that high.

Reducing this power consumption by a tenfold would result in an average consumption of 0.39mW, which is closer to the theoretical average power consumption. A discovery made when measuring the power was that the nodes consumed less power when supplied with a lower input voltage. Simply by reducing the voltage from 3V to 2.6V reduced the power consumption in LPM by 15%. However, this reduction could affect the range of the nodes.

Internet of Things can be realised in several ways as there are still many viable options on the market, mainly in terms of hardware, operating systems, and communication standards. Given the recent development in the field, Thingsquare recently released a technology demo using the same practices as used in this thesis; the choices taken are on track with the latest development [40]. Also, both Google and Microsoft have announced that they are developing IoT OSs. When these products are released, it would be very interesting to compare them with Contiki. It would be exciting to see if an open-source project can surpass the commercial offerings in terms of speed, RAM and ROM footprint, and device support. Furthermore, an in-depth comparison between RIOT and Contiki would give much insight into the kind of OS practices that benefit IoT development the most. Google have also started to develop a substitute for 6LoWPAN and UDP that they have named Thread [41]. As 6LoWPAN and ZigBee, it runs on top of IEEE 802.15.4 and thus might be able to out-compete the existing implementations. Google promises lower latencies and power consumption compared to the existing technologies.

The prototype

The prototype development took more time than initially planned; mostly because of the complexity of the OS, but also due to bugs in the untested drivers. The prototype combines the technology from each field, i.e. hardware, OS, and communication protocol, and fulfils the requirements set in section 3.1.1. Even though the OS is relatively simple, compared to Linux, Windows, and OS X, understanding the mechanics of the RDC driver and the LPM driver was difficult, but necessary to be able to interpret the test results. The prototype worked very well during most of the testing, with only a few unforeseen deviations. One occurred during the power measurement, where the power consumption in low power mode tripled in one of the test series; this behaviour could not be reproduced

and is therefore not included in the results. Also, in the early stages when working with the 8Hz RDC driver, packet losses over 50% were recorded for packets with more than one hop; this problem was solved, when a new version of radio driver was released by the OS development team. Selecting OpenMote to be the hardware platform together with Contiki as the OS, was a very good choice as companies are starting to build their IoT solutions around Contiki and similar hardware platforms [42, 43]. Already in the beginning of the development, several benefits were noticed; new drivers and bug-fixes were released increasing the stability and functionality of the OS. The active community around the combination of OpenMote and Contiki was really helpful when developing the drivers for the I 2 C and sensor drivers. Example projects for other platforms could be used as references, giving much insight to how the programming for this type of OS worked. It would have been interesting to examine the differences between two operating systems; not only to test which one has the better performance, but also to compare which one that has the more favourable code structure and development procedure.

6 Results

Collecting the data went well and were reasonably straight forward; it was easy to transition between the two different test set-ups and thus making several test scenarios. Assessments were made in the areas of range, response time, connection speed, and power consumption. In each area, the theoretical values were first calculated and then compared to the retrieved measurements; except in the range case, as the required equipment for measuring was not economically justifiable to purchase.

6.1 Range

The theoretical range for OpenMote when transmitting at full power in an office environment is only 7m. As measuring the range was not a viable option due to the cost of measuring equipment, only distance estimations from the placement of the nodes when maintaining a stable connection can be used as a reference. Using a map of the office and the position of the nodes the range seems to be around 10m, which would mean that the effective FM of the office is around 16dB using the always-on RDC. The FM changed a bit when using the 8Hz RDC as more packages congested the air and the range dropped to somewhere around 5m; resulting in an effective FM of 23dB. To increase the range of the transceiver, a switch to the 860MHz frequency band would be the most effective solution; with a FM of 23dB, the theoretical range would increase to 14m with the same transceiver properties, and with a FM of 16dB the range would be 31m. Usually, transceivers with a lower frequency output also have a lower power consumption while transmitting. Working in sub-GHz also gives the benefit of less interference as fewer other devices uses those frequencies. Changing to a sub-GHz band would thus decrease the power consumption and increase the range, without changing the functionality of the nodes.

6.2 Response time

Initially when measuring the response time the always-on RDC was used and the measured response time was very close to the theoretical value. However, when using the 8Hz RDC protocol the values started to drastically differ from the theory. This behaviour is likely to originate from the way the RDC driver predicts the next time when the target node should be awake. The procedure is called phase optimization; when enabled, the node saves the time when the node was last seen, it then uses this value to predict the next time the node should be awake based on the RDC cycle. However, this prediction is based on the node's internal clock. As the clock can differ from those of the other nodes, misalignments seem to occur, resulting in misses when trying to reach the target node. Each misalignment increases the time it takes to reach the target node as the node then needs to strobe the package until the target nodes wakes up again. In theory, when sending strobes the target node should wake up and receive the package within one cycle (125ms); however, this is not guaranteed as other transmissions might occupy the air, further increasing the response time. If the phase optimization could be improved to guarantee the alignment between the nodes, the response time should get much closer to the theoretical value; as the time to reach the node would be maximum one cycle and the air would not be as congested by nodes sending strobes.

6.3 Connection speed

The connection speed, when using CoAP or any other protocol with perpacket ACK, is directly bound to the response time. IEEE 802.15.4 has a relatively low data-rate, only 250kbps, compared to other solutions, e.g. BLE (1Mbps) and WLAN (>54Mbps). As throughput is based on datarate over a longer period of time, both the overhead and the response time is needed to make a good estimation. CoAP has a very low header size compared to many other communication protocols, but

due to the very small frame size, the overhead is still relatively high. As of now, the results clearly show that when a reliable transfer is desired the connection speed of IEEE 802.15.4 and CoAP is only sufficient for data exchanges around 32 bytes. When the nodes use the always-on RDC, the goodput is less than 3KB/s for one hop and is halved for every hop; however, when the 8Hz RDC is enabled, the goodput is reduced to under 0.1KB/s. Using messages without per-packet ACK, thus removing the response time from the equation, would let the nodes transfer real-time audio and maybe even highly compressed video. However, using messages without the per-packet ACK disables the reliable transmission guarantee, and thus it can only be used with data streams where packet loss is acceptable.

6.4 Power consumption

Making a rough estimation of the power consumption of the platform was straight forward task and so was measuring the actual consumption. When comparing, the two the values differed by a factor of 30, which was not expected. The reason probably originates from the clock interrupt which is triggered every 8ms. Initially, this interrupt was assumed to be disabled when the system entered the lower power modes, but this was not the case. As the interrupt fires at 125Hz and the time to wake up and go back to LPM is only 272s, the power spikes from these interrupts were not seen on the measuring instruments. As seen in figure 4.9, even the peaks from the listening cycles were hard to record and those lasted for at least 4ms; instead, the power consumption from the clock timer looks like an increased LPM power consumption. At the time this was discovered there was no time to fix it, but doing so should decrease the average power consumption to within the limits, granting the nodes the ability to run on battery power for a year. As no delays from calculation could be observed, the clock speed on MCU could, in all probability, have been reduced to save power on the nodes. However, this reduction would only have affected the consumption when the node was in active mode, which is only a few percent of the total cycle time. The OpenMote chip has a step-down DC-DC converter for this purpose which is switched off in LPM mode to reduce quiescent currents; however, as most of the time is spent in LPM, reducing the input voltage to 2.1V by changing battery type and removing the step-down converter would be preferable as it would reduce the power consumption. These changes could affect the range of the device, but this has to be assessed.

6.5 Project execution

Looking at the time plan and the milestones, as seen in Appendix A and B, each milestone matches a task or transition in the time plan. The planning report was not submitted to the examiner until the 6/2-15, which is two weeks behind schedule, exceeding the time planned for milestone M1. The first draft was submitted before deadline, but several revisions were necessary. In retrospect, the litterature study should probably have been planned in parallel with the planning report, as the information from the study helped with the report. Milestone M2 marks the switch from the literature study and selection of technology to the development phase. This milestone was met and development could begin in the following week. As seen in Appendix C, the development phase have several risks to consider. The only risk encountered in this phase was R4, as one of the hardware platforms was delivered with a broken sensor. However, this malfunction did not affect the time plan as the development could continue regardless of the malfunction. The end of the development phase was defined by milestone M3, approval of prototype, which was completed ahead of schedule granting an early transition into the assessment phase. In the assessment phase, it could be argued that risk R9 was encountered when measuring the power consumption, as the results from those measurements did not properly show the wake-ups from the clock timer. This phase contained milestone M4 and M5, of which of only M5 was done in time. The Half-time presentation, milestone M4, was performed on the 8/4-15 in the form of a meeting, where the progress, results and continuation plan were discussed. Also, a half-time version of the report was sent the 17/4-15 and approved by the examiner. Milestone M6, deliver the final prototype, was completed a few days before the set deadline which eliminated risk R11 and gave more time to work on the writing and the presentation. Both of the oral presentations were attended on the 1/6-15 to grant some experience in how the presentation and opposition are carried out, thus now following the time plan. However, there were not many presentations to watch during the planned weeks, as the presentation schedule follow the academic semesters. The presentation for this thesis was not performed until the 3/6-15, thus being two weeks behind schedule. However, it was scheduled on the first available date suggested by the institution. The final version of the report will be submitted to the examiner before the 19/6-15, thus successfully completing milestone M7.

7 Discussion

The purpose of the project was to find and examine a communication protocol that could be suitable for IoT applications, by investigating the current hardware, OS, and communication protocols

and building a prototype from the selected choices. What can be said about the investigation is that it is difficult to examine all candidates in detail; this means that a rough selection has to be made based on initial knowledge potentially discarding good options. The general feeling is, however, that all of the examined candidates in this project were relevant and added valuable insights to the current technology status. The assessment gave relevant and interesting results that improved the understanding in what IoT can be used for, and what further areas of investigation could be. One of the most interesting areas of further investigation would be the RDC driver, as it directly affects the response time and thus also the connection speed. Even though the power consumption was not in line with the expectations, the reason has been found and can be resolved. Another conclusion is that IoT is not ready for real-time applications as the latency is much higher than expected, for the technologies assessed in this thesis, and also has a high spread. As the latency increases for each subsequent network hop and the minimum observed latency per hop is 11ms, when using the always-on RDC, this type of communication will probably only be used for applications where response time can vary greatly, without affecting the functionality. CoAP as a communication protocol shows a lot of promise when combined with 6LoWPAN and IEEE 802.15.4. It performs well given its simplicity but has one disadvantage: the large overhead which comes from the MAC addressing fields in the IEEE 802.15.4 frame. If this overhead could be reduced from the current 71% to only 30the goodput would double. A solution would be to use a similar mechanism as BLE where the packet size varies depending on application. Each node also has computing time left as the MCU is more powerful than needed for the given application; an improvement would be to use a less powerful MCU, like the ARM Cortex-M0+, to reduce the clock speed as suggested in the discussion. When looking at the future-proof aspect the later suggestion is probably the better, as the clock then could be increased if more computing power is needed. In the future, batteries will hopefully be able to store more energy, thus increasing the time between battery changes or reducing the battery size.

6 | Conclusion

"Everything that has a beginning has an ending.
Make your peace with that and all will be well" -
Jack Kornfield

Contents

1 Conclusion

2 Perspectives

A | Appendix

"In any conflict, discover the one who rubs his hands ... You'll see that it's never the one who fights !" - Marc Roussel

CSS Carrier Frequency (CF) Forward error correction (FEC) Path loss (PL) Link Symmetry (LS) Base Station (BS) CSS Direct Sequence Spread Spectrum (DSSS) UNB Data Rate (DR) ADR CR BW

<table>
<tr><td colspan="2">Preamble</td><td>Sync msg</td><td>PHY Header</td><td>PHDR-CRC</td><td colspan="15">PHY Payload</td><td colspan="2">CRC</td></tr>
<tr><td>Modula-tion</td><td>length</td><td>Sync msg</td><td>PHY Header</td><td>PHDR-CRC</td><td colspan="3">MAC Header</td><td colspan="11">MAC Payload</td><td>MIC</td><td>CRC Type</td><td>Polyno-mial</td></tr>
<tr><td>Modula-tion</td><td>length</td><td>Sync msg</td><td>PHY Header</td><td>PHDR-CRC</td><td>MType</td><td>RFU</td><td>Major</td><td colspan="9">Frame Header</td><td>FPort</td><td>Frame Payload</td><td>MIC</td><td>CRC Type</td><td>Polyno-mial</td></tr>
<tr><td>Modula-tion</td><td>length</td><td>Sync msg</td><td>PHY Header</td><td>PHDR-CRC</td><td>MType</td><td>RFU</td><td>Major</td><td colspan="2">Dev Address</td><td colspan="5">FCtrl</td><td>FCnt</td><td>FOpts</td><td>FPort</td><td>Frame Payload</td><td>MIC</td><td>CRC Type</td><td>Polyno-mial</td></tr>
<tr><td>Modula-tion</td><td>length</td><td>Sync msg</td><td>PHY Header</td><td>PHDR-CRC</td><td>MType</td><td>RFU</td><td>Major</td><td>NwkID</td><td>NwkAddr</td><td>ADR</td><td>ADRACK-Req</td><td>ACK</td><td>FPending /RFU</td><td>FOptsLen</td><td>FCnt</td><td>FOpts</td><td>FPort</td><td>Frame Payload</td><td>MIC</td><td>CRC Type</td><td>Polyno-mial</td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td><td>21</td></tr>
</table>

0) **Modulation :**
   - Lora: 8 Symbols, 0x34 (Sync Word)
   - FSK: 5 Bytes, 0xC194C1 (Sync Word)
1) **Length :**
2) **Sync msg :**
3) **PHY Header :** It contains:
   - The Payload length (Bytes)
   - **The Code rate**
   - Optional 16bit CRC for payload
4) **Phy Header :** CRC It contains CRC of Physical Layer Header
5) **MType :** is the message type (uplink or a downlink)
   - whether or not it is a confirmed message (reqst ack)
   - 000 Join Request
   - 001 Join Accept
   - 010 Unconfirmed Data Up
   - 011 Unconfirmed Data Down
   - 100 Confirmed Data Up
   - 101 Confirmed Data Down
   - 110 RFU
   - 111 Proprietary
6) **RFU :** Reserved for Future Use
7) **Major :** is the LoRaWAN version; currently, only a value of zero is valid
   - 00 LoRaWAN R1
   - 01-11 RFU
8) **NwkID :** the short address of the device (Network ID): 31th to 25th
9) **NwkAddr :** the short address of the device (Network Address): 24th to 0th
10) **ADR :** Network server will change the data rate through appropriate MAC commands
    - 1 To change the data rate
    - 0 No change
11) **ADRACKReq :** (Adaptive Data Rate ACK Request): if network doesn't respont in 'ADR-ACK-DELAY' time, end-device switch to next lower data rate.
    - 1 if (ADR-ACK-CNT) >= (ADR-ACK-Limit)
    - 0 otherwise
12) **ACK :** (Message Acknowledgement): If end-device is the sender then gateway will send the ACK in next receive window else if gateway is the sender then end-device will send the ACK in next transmission.
    - 1 if confirmed data message
    - 0 otherwise
13) **FPending↓ /RFU ↑ :** (Only in downlink), if gateway has more data pending to be send then it asks end-device to open another receive window ASAP
    - 1 to ask for more receive windows
    - 0 otherwise
14) **FOptsLen :** is the length of the FOpts field in bytes 0000 to 1111
15) **FCnt :** 2 type of frame counters
    - FCntUp: counter for uplink data frame, MAX-FCNT-GAP
    - FCntDown: counter for downlink data frame, MAX-FCNY-GAP
16) **FOpts :** is used to piggyback MAC commands on a data message
17) **FPort :** a multiplexing port field
    - 0 the payload contains only MAC commands
    - 1 to 223 Application Specific
    - 224 & 225 RFU
18) **FRMPayload :** (Frame Payload) Encrypted (AES, 128 key length) Data
19) **MIC :** is a cryptographic message integrity code
    - computed over the fields MHDR, FHDR, FPort and the encrypted FRMPayload.
20) **CRC :** (only in uplink),
    - CCITT $x^{16} + x^{12} + x^5 + 1$
    - IBM $x^{16} + x^{15} + x^5 + 1$

Characteristics	$CF_{[Hz]}$	6LoWPAN	LoRaWAN	SigFox	NB-IoT	INGENU	TELE
Modulation	2.4G	O-QPSK	-	-	QSPSK		2-FSK
	915M	BPSK	LoRa	BPSK,GFSK	QSPSK n-tone	RPMA, CDMA	2-FSK
	868M	BPSK	LoRa/GFSK	BPSK,GFSK	/4-QPSK 1-tone		2-FSK
Chwidth$_{[\mathbf{KHz}]}$			500 - 125		180		
	2.4G	16	-	-	-	40	✗
Channels	915M	10	64+8, 8	✗	✗	✗	✗
	868M	1	10	360+40	✗	✗	✗
$CF_{[MHz]}$	2.4G	✗	-	-	-	✗	ISM
	915M	902-929	902-928	902	✗	✗	915M
	868M	868-868.6	863-870 and 780	868.18-868.22	✗	✗	868M/
$BW_{[Hz]}$	2.4G	5M	-	-	200K	1M	✗
	915M	2M	125K-500K	✗	✗	✗	✗
	868M	600M	125K-250K	0.1K-1.2K	✗	✗	✗
$DR_{[bps]}$	2.4G	250M	-	-	-	78K, 19,5K	✗
	915M	40M	980-22K	✗	234.7, 204.8	✗	✗
	868M	20M	LoRa: 0.3K-37.5K	0.1K,0.6K	✗	✗	62.5, 5
.			FSK: 50K				
$CR_{[dBm]}$	2.4G	-85	-	-	-	✗	✗
	915M	-92	✗	✗	✗	✗	✗
	868M	-92	-137	-137	✗	✗	✗
$ChipR_{[chip/s]}$	2.4G						
	915M						
	868M						
Range	2.4G						
	915M						
	868M	10-100 m	5-15 Km	10-50 Km	1Km	15-? Km	1Km-?
Handover	2.4G	✗	-	-	-	✗	✗
	915M	✗	✗	✗	✗	✗	✗
	868M	✗	Multi BS	Multi BS	✗	✗	✗
msg/day	2.4G	✗	-	-	-	✗	✗
	915M	✗	✗	✗	✗	✗	✗
	868M	✗	Unlimited	140,4	Unlimited	✗	✗
PL B	2.4G	✗	-	-	-	✗	✗
	915M	✗	✗	✗	✗	✗	✗
	868M	✗	51 - 243	12,8	1600B	10KB	✗
Coding/Spreading		DSSS	CSS	UNB	✗	DSSS	UNB
Proprietary		✗	✗	✓	✗	✗	✗
Topology		✗	Star, Stars	Star	✗	Star, Tree	Star
ADR		✗	✓	✗	✗	✓	✗
Security		✗	AES 128b	✗	✗	AES 256B	✗
LS		✗	✓	✗	✗	✗	✗
FEC		✗	AES 128b	✗	✗	✓	✗
Battery		1-2 years	<10 years	<10 years	<10 years		
Cost		Free	35e	25e	1020e		
Standar		IETF	LoRa Alliance		3GPP		
Duplex			Half		Half		
Mob support			High,Simple		High,complex		
Mob latency			Low		High (1.6-10s)		
$Tx_{[dBm]}$			+14 - +27		20/23		
Real-Time			Class C		✗		
Scalability			1M, 100K		55 k		
$Linkbudget_{[dB]}$			157		154		
$Sensitivity_{[dBm]}$			-124 - (-134)		-141		
Multi-hop supporter			✗		✗		
Addressing			Broadcast, Unicast		Unicast, Both		
Peak current			32 mA		120–300 mA		
Sleep current			1 A		5 A		

Table A.1. LPWAN Characteristics [122], [**lopes__design__2019**], [**raza__low__22**], [123]

Characteristics	$CF_{[Hz]}$	ZigBee	LoRaWAN	SigFox	NB-IoT	INGENU	TELENSA
Modulation	2.4G 915M 868M	O-QPSK BPSK BPSK					
Channels	2.4G 915M 868M	16 10 1					
$CF_{[MHz]}$	2.4G 915M 868M	2.4835 902, 928 868, 868.6					
$BW_{[Hz]}$	2.4G 915M 868M						
$DR_{[b/s]}$	2.4G 915M 868M	250 kbps 40 kbps 20 kbps					
$CR_{[dBm]}$	2.4G 915M 868M						
$ChipR_{[chip/s]}$	2.4G 915M 868M	2M 600K 300K					
Handover	2.4G 915M 868M						
msg/day	2.4G 915M 868M						
PL B	2.4G 915M 868M						
Coding							
Proprietary							
Topology							
ADR							
Security							
LS							
FEC							
Range							
Battery							
Cost							
Standar	IEEE 802.15.4						

Table A.2. LPWAN Characteristics [**berder_reseaux_2014**]

Standard	802.15.4k	802.15.4g	Weightless-W	Weightless-N	Weightless-P	DASH 7 Alliance
Modulation	DSSS, FSK	MR-[FSK, OFDMA, OQPSK]	16-QAM, BPSK, QPSK, DBPSK	UNB DBPSK	GMSK, offset-QPSK	GFSK
BW	ISM S UB -GH Z, 2.4GHz	ISM S UB -GH Z, 2.4GHz	TV white spaces 470-790MHz	ISM S UB -GH Z EU (868MHz), US (915MHz)	S UB -GH Z ISM or licensed	UB -GH Z 433MHz, 868MHz, 915MHz
DR	1.5 bps-128 kbps	4.8 kbps-800 kbps	1 kbps-10 Mbps	30 kbps-100 kbps	200 bps-100kbps	9.6,55.6,166.7 kbps
Range	5 km (URBAN)	up to several kms	5 km (URBAN)	3 km (URBAN)	2 km (URBAN)	0-5 km (URBAN)
MAC	CSMA/CA, CSMA/CA or A LOHA with PCA	CSMA/CA	TDMA/FDMA	slotted A LOHA	TDMA/FDMA	CSMA/CA
Topology	star	tar, mesh, peer-to-peer	star	star	star	tree, star
PL	2047B	2047B	>10B	20B	>10B	256B
Security	AES 128b	AES 128b	AES 128b	AES 128b	AES 128/256b	AES 128b
Forward error correction	✓	✓	✓	✗	✓	✓

Table A.3. [**raza__low__22**]

Phy protocol	IEEE 802.15.4	BLE	EPCglobal	Z-Wave	LTE-M	ZigBee
Standard		IEEE 802.15.1				IEEE 802.15.4, ZigBee Alliance
BW(MHz)	868/915/2400	2400	860-960	868/908/2400	700-900	
MAC	TDMA, CSMA/CA	TDMA	ALOHA	CSMA/CA	OFDMA	
DR (bps)	20/40/250 K	1024K	varies 5-640K	40K	1G (up), 500M (down)	
Throughput				9.6, 40, 200kbps		
Scalability	65K nodes	5917 slaves	-	232 nodes	-	
Range	10-20m	10-100m				
Addressing	8|16bit	16bit				

Table A.4. IoT cloud platforms and their characteristics [**al-fuqaha__internet__24**]

	802.15.4	802.15.4e	802.15.4g	802.15.4f
CF	2.4Ghz (DSSS + oQPSK) 868Mhz (DSSS + BPSK) 915Mhz (DSSS + BPSK)	2.4Ghz (DSSS + oQPSK, CSS+DQPSK) 868Mhz (DSSS + BPSK) 915Mhz (DSSS + BPSK)	2.4Ghz (DSSS + oQPSK, CSS+DQPSK) 868Mhz (DSSS + BPSK) 915Mhz (DSSS + BPSK)	2.4Ghz (DSSS + oQPSK,CSS+DQPSK 868Mhz (DSSS + BPSK) 915Mhz (DSSS + BPSK) 3∼10Ghz (BPM+BPSK)
DR	Upto 250kbps	Upto 800kbps	Up to 800kbps	
Differences	-	Time sync and channel hopping	Phy Enhancements	Mac and Phy Enhancements
PL	127 bytes	N/A	Up to 2047 bytes	N/A
Range	1 – 75+ m	1 – 75+ m	Upto 1km	N/A
Goals	General Low-power Sensing/Actuating	Industrial segments	Smart utilities	Active RFID
Products	Many	Few	Connode (6LoWPAN)	LeanTegra PowerMote

Table A.5. IEEE 802.15.4 standards [**sarwar__iot__2015**]

Feature	Wi-Fi	802.11p	UMTS	LTE	LTE-A
Channel MHz	20	10	5	1.4, 3, 5, 10, 15, 20	<100
Frequency band(s) GHz	2.4 , 5.2	5.86-5.92	0.7-2.6	0.7-2.69	0.45-4.99
BR Mb/s	6-54	3–27	2	<300	<1000
Range km	<0.1	<1	<10	<30	<30
Capacity	Medium	Medium	✗	✓	✓
Coverage	Intermittent	Intermittent	Ubiquitous	Ubiquitous	Ubiquitous
Mobility support km/h	✗	Medium	✓	<350	<350
QoS support	EDCA Enhanced Distributed Channel Access	EDCA Enhanced Distributed Channel Access	QoS classes and bearer selection	QCI and bearer selection	QCI and bearer selection
Broadcast/multicast support	Native broadcast	Native broadcast	Through MBMS	Through eMBMS	Through eMBMS
V2I support	✓	✓	✓	✓	✓
V2V support	Native (ad hoc)	Native (ad hoc)	✗	✗	Through D2D
Market penetration	✓	✗	✓	✓	✓
DR	<640 kbps	250 kbps	106–424 kbps	✓	✓

Table A.6. An example table.

PSSignal-to-interference & noise ratio (SINR)

SF/BW	125kHz					250kHz				500kHz			
-	[varsier_capacity_2017]		[124]	[125]									
-	Sensitivity	BR	Rx wind	SINR	PS	Sensitivity	BR	Rx wind	SINR	Sensitivity	BR	Rx wind	SINR
-	[dBm]	[kb/s]	[ms]	[dB]	Byte	[dBm]	[kb/s]	[ms]	[dB]	[dBm]	[kb/s]	[ms]	[dB]
6	-118				242+13	-115				-111			
7	-123	5.468	5.1	-7.5	242+13	-120				-116			
8	-126	3.125	10.2	-10	242+13	-123				-119			
9	-129	1.757	20.5	-12.5	115+13	-125				-122			
10	-132	0.976	41.0	-15	51+13	-128				-125			
11	-133	0.537	81.9	-17.5	51+13	-130				-128			
12	-136	0.293	163.8	-20	51+13	-133				-130			

Table A.7. Receiver sensitivity [dBm]

[**_evaluation_**] Nous avons vu en effet plus haut qu'il a été démontré que la méthode CSMA est plus efficace pour le traitement des faibles trafics, tandis que TDMA est nettement plus appropriée pour supporter les trafics intensesj. PS

DR	Modulation			PS	BR
	SF	BW [kHz]	CR	Byte	x kbit/s
0	12	125	4/6	51+13	0.25
1	11	125	4/6	51+13	0.44
2	10	125	4/5	51+13	0.98
3	9	125	4/5	115+13	1.76
4	8	125	4/5	242+13	3.125
5	7	125	4/5	242+13	5.47
6	7	125	4/5	242+13	11
7		125	4/5	242+13	50

Table A.8. oioioi

Bibliography

"A quote in a speech, article or book is like a gun in the hands of a soldier. It speaks with authority."

5g

[52] Akihiro Nakao et al. End-to-End Network Slicing for 5G Mobile Networks. In: *Journal of Information Processing* 25.0 (2017), pp. 153–163.

[53] Yasser Gadallah, Mohamed H. Ahmed, and Ehab Elalamy. Dynamic LTE Resource Reservation for Critical M2M Deployments. In: *Pervasive and Mobile Computing* 40 (Sept. 2017), pp. 541–555.

[54] Yixue Hao et al. Network Slicing Technology in a 5G Wearable Network. In: *IEEE Communications Standards Magazine* 2.1 (Mar. 2018), pp. 66–71.

lora-channel

[9] B. Zurita Ares et al. On Power Control for Wireless Sensor Networks: System Model, Middleware Component and Experimental Evaluation. In: *2007 European Control Conference (ECC)*. European Control Conference 2007 (ECC). Kos: IEEE, July 2007, pp. 4293–4300.

[11] Alaa Muqattash and Marwan Krunz. A Single-Channel Solution for Transmission Power Control in Wireless Ad Hoc Networks. In: *Proceedings of the 5th ACM International Symposium on Mobile Ad Hoc Networking and Computing - MobiHoc '04*. The 5th ACM International Symposium. Roppongi Hills, Tokyo, Japan: ACM Press, 2004, p. 210.

[67] Davide Sartori and Davide Brunelli. A Smart Sensor for Precision Agriculture Powered by Microbial Fuel Cells. In: *2016 IEEE Sensors Applications Symposium (SAS)*. 2016 IEEE Sensors Applications Symposium (SAS). Catania, Italy: IEEE, Apr. 2016, pp. 1–6.

[83] Pascal Jorke et al. Urban Channel Models for Smart City IoT-Networks Based on Empirical Measurements of LoRa-Links at 433 and 868 MHz. In: *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC). Montreal, QC: IEEE, Oct. 2017, pp. 1–6.

[87] Qipeng Song, Xavier Lagrange, and Loutfi Nuaymi. Evaluation of Macro Diversity Gain in Long Range ALOHA Networks. In: *IEEE Communications Letters* 21.11 (Nov. 2017), pp. 2472–2475.

[95] Dmitry Bankov, Evgeny Khorov, and Andrey Lyakhov. Mathematical Model of LoRaWAN Channel Access with Capture Effect. In: *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC). Montreal, QC: IEEE, Oct. 2017, pp. 1–5.

[102] Claire Goursaud and Yuqi Mo. Random Unslotted Time-Frequency ALOHA: Theory and Application to IoT UNB Networks. In: *2016 23rd International Conference on Telecommunications (ICT)*. 2016 23rd International Conference on Telecommunications (ICT). Thessaloniki, Greece: IEEE, May 2016, pp. 1–5.

[110] C. Tunc and N. Akar. Markov Fluid Queue Model of an Energy Harvesting IoT Device with Adaptive Sensing. In: *Performance Evaluation* 111 (May 2017), pp. 1–16.

[120] MADWIFI. URL: `https://sourceforge.net/p/madwifi/svn/HEAD/tree/madwifi/trunk/ath_rate/minstrel/minstrel.txt` (visited on 01/31/2020).

Others

[1] Norhane Benkahla et al. Enhanced Dynamic Duty Cycle in LoRaWAN Network. In: *Ad-Hoc, Mobile, and Wireless Networks*. Ed. by Nicolas Montavont and Georgios Z. Papadopoulos. Vol. 11104. Cham: Springer International Publishing, 2018, pp. 147–162.

[2] A. Springer et al. Spread Spectrum Communications Using Chirp Signals. In: *IEEE/AFCEA EUROCOMM 2000. Information Systems for Enhanced Public Safety and Security (Cat. No.00EX405)*. On Information Systems for Enhanced Public Safety and Security (EUROComm 2000). Munich, Germany: IEEE, 2000, pp. 166–170.

[3] LoRa Alliance. LoraWAN Specification. URL: `https://lora-alliance.org/resource-hub/lorawanr-specification-v103` (visited on 08/30/2019).

[4] Raja Karmakar, Samiran Chattopadhyay, and Sandip Chakraborty. Linkcon: Adaptive Link Configuration over SDN Controlled Wireless Access Networks. In: *Proceedings of the ACM Workshop on Distributed Information Processing in Wireless Networks - DIPWN'17*. The ACM Workshop. Chennai, India: ACM Press, 2017, pp. 1–6.

[5] Ioannis Pefkianakis et al. Window-Based Rate Adaptation in 802.11n Wireless Networks. In: *Mobile Networks and Applications* 18.1 (Feb. 2013), pp. 156–169.

[6] Duy Nguyen and J. J. Garcia-Luna-Aceves. A Practical Approach to Rate Adaptation for Multi-Antenna Systems. In: *2011 19th IEEE International Conference on Network Protocols*. 2011 19th IEEE International Conference on Network Protocols (ICNP). Vancouver, AB, Canada: IEEE, Oct. 2011, pp. 331–340.

[7] Martin Bor and Utz Roedig. LoRa Transmission Parameter Selection. In: *2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. 2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS). Ottawa, ON: IEEE, June 2017, pp. 27–34.

[8] Shan Lin et al. ATPC: Adaptive Transmission Power Control for Wireless Sensor Networks. In: *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems - SenSys '06*. The 4th International Conference. Boulder, Colorado, USA: ACM Press, 2006, p. 223.

[10] J.P. Monks, V. Bharghavan, and W.-M.W. Hwu. A Power Controlled Multiple Access Protocol for Wireless Packet Networks. In: *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No.01CH37213)*. IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society. Vol. 1. Anchorage, AK, USA: IEEE, 2001, pp. 219–228.

[12] Mathieu Lacage, Mohammad Hossein Manshaei, and Thierry Turletti. IEEE 802.11 Rate Adaptation: A Practical Approach. In: *Proceedings of the 7th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems - MSWiM '04*. The 7th ACM International Symposium. Venice, Italy: ACM Press, 2004, p. 126.

[13] Starsky H. Y. Wong et al. Robust Rate Adaptation for 802.11 Wireless Networks. In: *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking - MobiCom '06*. The 12th Annual International Conference. Los Angeles, CA, USA: ACM Press, 2006, p. 146.

[14] K Ramachandran et al. Symphony: Synchronous Two-Phase Rate and Power Control in 802.11 WLANs. In: *IEEE/ACM Transactions on Networking* 18.4 (Aug. 2010), pp. 1289–1302.

[15] Pierre Chevillat, Jens Jelitto, and Hong Linh Truong. Dynamic Data Rate and Transmit Power Adjustment in IEEE 802.11 Wireless LANs. In: *International Journal of Wireless Information Networks* 12.3 (July 2005), pp. 123–145.

[16] Francesca Cuomo et al. EXPLoRa: Extending the Performance of LoRa by Suitable Spreading Factor Allocations. In: *2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. 2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob). Rome: IEEE, Oct. 2017, pp. 1–8.

[17] Aloÿs Augustin et al. A Study of LoRa: Long Range & Low Power Networks for the Internet of Things. In: *Sensors* 16.9 (Sept. 9, 2016), p. 1466.

[18] Thiemo Voigt et al. Mitigating Inter-Network Interference in LoRa Networks. In: (Nov. 2, 2016). arXiv: 1611.00688 [cs].

[19] Orestis Georgiou and Usman Raza. Low Power Wide Area Network Analysis: Can LoRa Scale? In: *IEEE Wireless Communications Letters* 6.2 (Apr. 2017), pp. 162–165. arXiv: 1610.04793.

[20] Konstantin Mikhaylov, Juha Petäjäjärvi, and Tuomo Hänninen. Analysis of Capacity and Scalability of the LoRa Low Power Wide Area Network Technology. In: (2016), p. 6.

[21] Brecht Reynders, Wannes Meert, and Sofie Pollin. Power and Spreading Factor Control in Low Power Wide Area Networks. In: *2017 IEEE International Conference on Communications (ICC)*. ICC 2017 - 2017 IEEE International Conference on Communications. Paris, France: IEEE, May 2017, pp. 1–6.

[22] Juha Petajajarvi et al. On the Coverage of LPWANs: Range Evaluation and Channel Attenuation Model for LoRa Technology. In: *2015 14th International Conference on ITS Telecommunications (ITST)*. 2015 14th International Conference on ITS Telecommunications (ITST). Copenhagen, Denmark: IEEE, Dec. 2015, pp. 55–59.

[23] Martin C. Bor et al. Do LoRa Low-Power Wide-Area Networks Scale? In: *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems - MSWiM '16*. The 19th ACM International Conference. Malta, Malta: ACM Press, 2016, pp. 59–67.

[24] Davide Magrin, Marco Centenaro, and Lorenzo Vangelista. Performance Evaluation of LoRa Networks in a Smart City Scenario. In: *2017 IEEE International Conference on Communications (ICC)*. ICC 2017 - 2017 IEEE International Conference on Communications. Paris, France: IEEE, May 2017, pp. 1–7.

[25] Francesca Cuomo et al. Towards Traffic-Oriented Spreading Factor Allocations in LoRaWAN Systems. In: *2018 17th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*. 2018 17th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net). Capri: IEEE, June 2018, pp. 1–8.

[26] Ferran Adelantado et al. Understanding the Limits of LoRaWAN. In: (Feb. 13, 2017). arXiv: 1607.08011 [cs].

[27] Juha Petäjäjärvi et al. Performance of a Low-Power Wide-Area Network Based on LoRa Technology: Doppler Robustness, Scalability, and Coverage. In: *International Journal of Distributed Sensor Networks* 13.3 (Mar. 2017), p. 155014771769941.

[28] Juha Petäjäjärvi et al. Evaluation of LoRa LPWAN Technology for Indoor Remote Health and Wellbeing Monitoring. In: *International Journal of Wireless Information Networks* 24.2 (June 2017), pp. 153–165.

[29] Khaled Q. Abdelfadeel, Victor Cionca, and Dirk Pesch. Fair Adaptive Data Rate Allocation and Power Control in LoRaWAN. In: (Feb. 28, 2018). arXiv: 1802.10338 [cs].

[30] Benjamin Sartori et al. Enabling RPL Multihop Communications Based on LoRa. In: *2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. 2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob). Rome: IEEE, Oct. 2017, pp. 1–8.

[31] Arshad Farhad, Dae-Ho Kim, and Jae-Young Pyun. Scalability of LoRaWAN in an Urban Environment: A Simulation Study. In: *2019 Eleventh International Conference on Ubiquitous and Future Networks (ICUFN)*. 2019 Eleventh International Conference on Ubiquitous and Future Networks (ICUFN). Zagreb, Croatia: IEEE, July 2019, pp. 677–681.

[32] Vinay Gupta et al. Modelling of IoT Traffic and Its Impact on LoRaWAN. In: *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*. 2017 IEEE Global Communications Conference (GLOBECOM 2017). Singapore: IEEE, Dec. 2017, pp. 1–6.

[33] Jin-Taek Lim and Youngnam Han. Spreading Factor Allocation for Massive Connectivity in LoRa Systems. In: *IEEE Communications Letters* 22.4 (Apr. 2018), pp. 800–803.

[34] Moises Nunez Ochoa et al. Large Scale LoRa Networks: From Homogeneous to Heterogeneous Deployments. In: *2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. 2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob). Limassol: IEEE, Oct. 2018, pp. 192–199.

[35] Mohamed Aref and Axel Sikora. Free Space Range Measurements with Semtech Lora™ Technology. In: *2014 2nd International Symposium on Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems*. 2014 2nd International Symposium on Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS-SWS). Odessa, Ukraine: IEEE, Sept. 2014, pp. 19–23.

[36] Andrew J Wixted et al. Evaluation of LoRa and LoRaWAN for Wireless Sensor Networks. In: *2016 IEEE SENSORS*. 2016 IEEE SENSORS. Orlando, FL, USA: IEEE, Oct. 2016, pp. 1–3.

[37] Wimol San-Um et al. A Long-Range Low-Power Wireless Sensor Network Based on U-LoRa Technology for Tactical Troops Tracking Systems. In: *2017 Third Asian Conference on Defence Technology (ACDT)*. 2017 Third Asian Conference on Defence Technology (ACDT). Phuket, Thailand: IEEE, Jan. 2017, pp. 32–35.

[38] Lingling Li, Jiuchun Ren, and Qian Zhu. On the Application of LoRa LPWAN Technology in Sailing Monitoring System. In: *2017 13th Annual Conference on Wireless On-Demand Network Systems and Services (WONS)*. 2017 13th Annual Conference on Wireless On-Demand Network Systems and Services (WONS). Jackson, WY, USA: IEEE, Feb. 2017, pp. 77–80.

[39] Charalampos Orfanidis et al. Investigating Interference between LoRa and IEEE 802.15.4g Networks. In: *2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. 2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob). Rome: IEEE, Oct. 2017, pp. 1–8.

[40] Brecht Reynders, Qing Wang, and Sofie Pollin. A LoRaWAN Module for Ns-3: Implementation and Evaluation. In: *Proceedings of the 10th Workshop on Ns-3 - WNS3 '18*. The 10th Workshop. Surathkal, India: ACM Press, 2018, pp. 61–68.

[41] Floris Van den Abeele et al. Scalability Analysis of Large-Scale LoRaWAN Networks in Ns-3. In: (May 16, 2017). arXiv: 1705.05899 [cs].

[42] Brecht Reynders, Wannes Meert, and Sofie Pollin. Range and Coexistence Analysis of Long Range Unlicensed Communication. In: *2016 23rd International Conference on Telecommunications (ICT)*. 2016 23rd International Conference on Telecommunications (ICT). Thessaloniki, Greece: IEEE, May 2016, pp. 1–6.

[43] Brecht Reynders et al. Improving Reliability and Scalability of LoRaWANs Through Lightweight Scheduling. In: *IEEE Internet of Things Journal* 5.3 (June 2018), pp. 1830–1842.

[44] Alexandru-Ioan Pop et al. Does Bidirectional Traffic Do More Harm Than Good in LoRaWAN Based LPWA Networks? In: (Dec. 14, 2017). arXiv: 1704.04174 [cs].

[45] Taoufik Bouguera et al. Energy Consumption Model for Sensor Nodes Based on LoRa and LoRaWAN. In: *Sensors* 18.7 (June 30, 2018), p. 2104.

[46] Jetmir Haxhibeqiri et al. LoRa Scalability: A Simulation Model Based on Interference Measurements. In: *Sensors* 17.6 (May 23, 2017), p. 1193.

[47] Keith E. Nolan, Wael Guibene, and Mark Y. Kelly. An Evaluation of Low Power Wide Area Network Technologies for the Internet of Things. In: *2016 International Wireless Communications and Mobile Computing Conference (IWCMC)*. 2016 International Wireless Communications and Mobile Computing Conference (IWCMC). Paphos, Cyprus: IEEE, Sept. 2016, pp. 439–444.

[48] Jiaming James Chen et al. A Viable LoRa Framework for Smart Cities. In: (2018), p. 16.

[49] Samir Dawaliby, Abbas Bradai, and Yannis Pousset. Adaptive Dynamic Network Slicing in LoRa Networks. In: *Future Generation Computer Systems* 98 (Sept. 2019), pp. 697–707.

[50] Konstantin Mikhaylov, Juha Petajajarvi, and Janne Janhunen. On LoRaWAN Scalability: Empirical Evaluation of Susceptibility to Inter-Network Interference. In: *2017 European Conference on Networks and Communications (EuCNC)*. 2017 European Conference on Networks and Communications (EuCNC). Oulu, Finland: IEEE, June 2017, pp. 1–6.

[51] Daniele Croce et al. Impact of Spreading Factor Imperfect Orthogonality in LoRa Communications. In: *Digital Communication. Towards a Smart and Secure Future Internet*. Ed. by Alessandro Piva, Ilenia Tinnirello, and Simone Morosi. Vol. 766. Cham: Springer International Publishing, 2017, pp. 165–179.

[55] Luca Feltrin et al. LoRaWAN: Evaluation of Link- and System-Level Performance. In: *IEEE Internet of Things Journal* 5.3 (June 2018), pp. 2249–2258.

[56] Valentin Alexandru Stan, Radu Serban Timnea, and Razvan Andrei Gheorghiu. Overview of High Reliable Radio Data Infrastructures for Public Automation Applications: LoRa Networks. In: *2016 8th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*. 2016 8th International Conference on Electronics, Computers and Artificial Intelligence (ECAI). Ploiesti, Romania: IEEE, June 2016, pp. 1–4.

[57] C. Goursaud and J. M. Gorce. Dedicated Networks for IoT: PHY / MAC State of the Art and Challenges. In: *EAI Endorsed Transactions on Internet of Things* 1.1 (Oct. 26, 2015), p. 150597.

[58] Ramon Sanchez-Iborra et al. Performance Evaluation of LoRa Considering Scenario Conditions. In: *Sensors* 18.3 (Mar. 3, 2018), p. 772.

[59] Sarra Naoui, Mohamed Elhoucine Elhdhili, and Leila Azouz Saidane. Enhancing the Security of the IoT LoraWAN Architecture. In: *2016 International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks (PEMWN)*. 2016 International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks (PEMWN). Paris, France: IEEE, Nov. 2016, pp. 1–7.

[60] Patrick Weber et al. IPv6 over LoRaWAN™. In: *2016 3rd International Symposium on Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS)*. 2016 3rd International Symposium on Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS). Offenburg, Germany: IEEE, Sept. 2016, pp. 75–79.

[61] Lluís Casals et al. Modeling the Energy Performance of LoRaWAN. In: *Sensors* 17.10 (Oct. 16, 2017), p. 2364.

[62] Byoungwook Kim and Kwang-il Hwang. Cooperative Downlink Listening for Low-Power Long-Range Wide-Area Network. In: *Sustainability* 9.4 (Apr. 17, 2017), p. 627.

[63] Pierre Neumann, Julien Montavont, and Thomas Noel. Indoor Deployment of Low-Power Wide Area Networks (LPWAN): A LoRaWAN Case Study. In: *2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. 2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob). New York, NY: IEEE, Oct. 2016, pp. 1–8.

[64] Michele Magno et al. WULoRa: An Energy Efficient IoT End-Node for Energy Harvesting and Heterogeneous Communication. In: *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017*. 2017 Design, Automation & Test in Europe Conference & Exhibition (DATE). Lausanne, Switzerland: IEEE, Mar. 2017, pp. 1528–1533.

[65] Adwait Dongare et al. OpenChirp: A Low-Power Wide-Area Networking Architecture. In: *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. 2017 IEEE International Conference on Pervasive Computing and Communications: Workshops (PerCom Workshops). Kona, HI: IEEE, Mar. 2017, pp. 569–574.

[66] Gilbert Conus et al. An Event-Driven Low Power Electronics for Loads Metering and Control in Smart Buildings. In: *2016 Second International Conference on Event-Based Control, Communication, and Signal Processing (EBCCSP)*. 2016 Second International Conference on Event-Based Control, Communication, and Signal Processing (EBCCSP). Krakow, Poland: IEEE, June 2016, pp. 1–7.

[68] Joel Toussaint, Nancy El Rachkidy, and Alexandre Guitton. Performance Analysis of the On-the-Air Activation in LoRaWAN. In: *2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*. 2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON). Vancouver, BC, Canada: IEEE, Oct. 2016, pp. 1–7.

[69] Pape Abdoulaye Barro. A LoRaWAN Coverage testBed and a Multi-Optional Communication Architecture for Smart City Feasibility in Developing Countries. In: (), p. 12.

[70] Bartłtomiej Błaszczyszyn and Paul Mühlethaler. Analyzing LoRa Long-Range, Low-Power, Wide-Area Networks Using Stochastic Geometry. In: *Proceedings of the 12th EAI International Conference on Performance Evaluation Methodologies and Tools - VALUETOOLS 2019*. The 12th EAI International Conference. Palma, Spain: ACM Press, 2019, pp. 119–126.

[71] Lorenzo Vangelista, Andrea Zanella, and Michele Zorzi. Long-Range IoT Technologies: The Dawn of LoRa™. In: *Future Access Enablers for Ubiquitous and Intelligent Infrastructures*. Ed. by Vladimir Atanasovski and Alberto Leon-Garcia. Vol. 159. Cham: Springer International Publishing, 2015, pp. 51–58.

[72] Benjamin Dix-Matthews, Rachel Cardell-Oliver, and Christof Hübner. LoRa Parameter Choice for Minimal Energy Usage. In: *Proceedings of the 7th International Workshop on Real-World Embedded Wireless Systems and Networks - RealWSN'18*. The 7th International Workshop. Shenzhen, China: ACM Press, 2018, pp. 37–42.

[73] Marco Zimmerling et al. pTunes: Runtime Parameter Adaptation for Low-Power MAC Protocols. In: *Proceedings of the 11th International Conference on Information Processing in Sensor Networks - IPSN '12*. The 11th International Conference. Beijing, China: ACM Press, 2012, p. 173.

[74] Rachel Cardell-Oliver et al. Error Control Strategies for Transmit-Only Sensor Networks: A Case Study. In: *2012 18th IEEE International Conference on Networks (ICON)*. 2012 18th IEEE International Conference on Networks (ICON). Singapore, Singapore: IEEE, Dec. 2012, pp. 453–458.

[75] Marco Cattani, Carlo Boano, and Kay Römer. An Experimental Evaluation of the Reliability of LoRa Long-Range Low-Power Wireless Communication. In: *Journal of Sensor and Actuator Networks* 6.2 (June 15, 2017), p. 7.

[76] P. J. Marcelis, V. Rao, and R. V. Prasad. DaRe: Data Recovery through Application Layer Coding for LoRaWAN. In: *Proceedings of the Second International Conference on Internet-of-Things Design and Implementation - IoTDI '17*. The Second International Conference. Pittsburgh, PA, USA: ACM Press, 2017, pp. 97–108.

[77] Muhammad Omer Farooq and Dirk Pesch. A Search into a Suitable Channel Access Control Protocol for LoRa-Based Networks. In: *2018 IEEE 43rd Conference on Local Computer Networks (LCN)*. 2018 IEEE 43rd Conference on Local Computer Networks (LCN). Chicago, IL, USA: IEEE, Oct. 2018, pp. 283–286.

[78] Ashirwad Gupta and Makoto Fujinami. Battery Optimal Configuration of Transmission Settings in LoRa Moving Nodes. In: *2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. 2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC). Las Vegas, NV, USA: IEEE, Jan. 2019, pp. 1–6.

[79] Vojtech Hauser and Tomas Hegr. Proposal of Adaptive Data Rate Algorithm for LoRaWAN-Based Infrastructure. In: *2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud)*. 2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud). Prague: IEEE, Aug. 2017, pp. 85–90.

[80] Arliones Hoeller et al. Exploiting Time Diversity of LoRa Networks Through Optimum Message Replication. In: *2018 15th International Symposium on Wireless Communication Systems (ISWCS)*. 2018 15th International Symposium on Wireless Communication Systems (ISWCS). Lisbon: IEEE, Aug. 2018, pp. 1–5.

[81] Shie-Yuan Wang et al. Performance of LoRa-Based IoT Applications on Campus. In: *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*. 2017 IEEE 86th Vehicular Technology Conference (VTC-Fall). Toronto, ON: IEEE, Sept. 2017, pp. 1–6.

[82] L. Angrisani et al. LoRa Protocol Performance Assessment in Critical Noise Conditions. In: *2017 IEEE 3rd International Forum on Research and Technologies for Society and Industry (RTSI)*. 2017 IEEE 3rd International Forum on Research and Technologies for Society and Industry - Innovation to Shape the Future for Society and Industry (RTSI). Modena, Italy: IEEE, Sept. 2017, pp. 1–5.

[84] Ruben Oliveira, Lucas Guardalben, and Susana Sargento. Long Range Communications in Urban and Rural Environments. In: *2017 IEEE Symposium on Computers and Communications (ISCC)*. 2017 IEEE Symposium on Computers and Communications (ISCC). Heraklion, Greece: IEEE, July 2017, pp. 810–817.

[85] Zhijin Qin and Julie A. McCann. Resource Efficiency in Low-Power Wide-Area Networks for IoT Applications. In: *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*. 2017 IEEE Global Communications Conference (GLOBECOM 2017). Singapore: IEEE, Dec. 2017, pp. 1–7.

[86] Yuqi Mo et al. Optimization of the Predefined Number of Replications in a Ultra Narrow Band Based IoT Network. In: *2016 Wireless Days (WD)*. 2016 Wireless Days (WD). Toulouse, France: IEEE, Mar. 2016, pp. 1–6.

[88] Jaco M. Marais, Reza Malekian, and Adnan M. Abu-Mahfouz. LoRa and LoRaWAN Testbeds: A Review. In: *2017 IEEE AFRICON*. 2017 IEEE AFRICON. Cape Town: IEEE, Sept. 2017, pp. 1496–1501.

[89] Brecht Reynders and Sofie Pollin. Chirp Spread Spectrum as a Modulation Technique for Long Range Communication. In: *2016 Symposium on Communications and Vehicular Technologies (SCVT)*. 2016 Symposium on Communications and Vehicular Technologies (SCVT). Mons, Belgium: IEEE, Nov. 2016, pp. 1–5.

[90] Mariusz Slabicki, Gopika Premsankar, and Mario Di Francesco. Adaptive Configuration of Lora Networks for Dense IoT Deployments. In: *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*. NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium. Taipei, Taiwan: IEEE, Apr. 2018, pp. 1–9.

[91] Thanh-Hai To and Andrzej Duda. Simulation of LoRa in NS-3: Improving LoRa Performance with CSMA. In: *2018 IEEE International Conference on Communications (ICC)*. 2018 IEEE International Conference on Communications (ICC 2018). Kansas City, MO: IEEE, May 2018, pp. 1–7.

[92] Dimitrios Zorbas et al. Improving LoRa Network Capacity Using Multiple Spreading Factor Configurations. In: *2018 25th International Conference on Telecommunications (ICT)*. 2018 25th International Conference on Telecommunications (ICT). St. Malo: IEEE, June 2018, pp. 516–520.

[93] Norbert Blenn and Fernando Kuipers. LoRaWAN in the Wild: Measurements from The Things Network. In: (June 9, 2017). arXiv: 1706.03086 [cs].

[94] Arliones Hoeller et al. Analysis and Performance Optimization of LoRa Networks With Time and Antenna Diversity. In: *IEEE Access* 6 (2018), pp. 32820–32829.

[96] P. J. Radcliffe et al. Usability of LoRaWAN Technology in a Central Business District. In: *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*. 2017 IEEE 85th Vehicular Technology Conference (VTC Spring). Sydney, NSW: IEEE, June 2017, pp. 1–5.

[97] Mattia Rizzi et al. Evaluation of the IoT LoRaWAN Solution for Distributed Measurement Applications. In: *IEEE Transactions on Instrumentation and Measurement* 66.12 (Dec. 2017), pp. 3340–3349.

[98] Dong-Hoon Kim, Eun-Kyu Lee, and Jibum Kim. Experiencing LoRa Network Establishment on a Smart Energy Campus Testbed. In: *Sustainability* 11.7 (Mar. 30, 2019), p. 1917.

[99] Umber Noreen, Ahcene Bounceur, and Laurent Clavier. A Study of LoRa Low Power and Wide Area Network Technology. In: *2017 International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*. 2017 International Conference on Advanced Technologies for Signal and Image Processing (ATSIP). Fez, Morocco: IEEE, May 2017, pp. 1–6.

[100] Alexandru Lavric and Valentin Popa. Performance Evaluation of LoRaWAN Communication Scalability in Large-Scale Wireless Sensor Networks. In: *Wireless Communications and Mobile Computing* 2018 (June 28, 2018), pp. 1–9.

[101] Zhuocheng Li et al. 2D Time-Frequency Interference Modelling Using Stochastic Geometry for Performance Evaluation in Low-Power Wide-Area Networks. In: (Nov. 12, 2016). arXiv: 1606.04791 [cs].

[103] Davide Magrin, Martina Capuzzo, and Andrea Zanella. A Thorough Study of LoRaWAN Performance Under Different Parameter Settings. In: (June 12, 2019). arXiv: 1906.05083 [cs].

[104] Jaco Morne Marais, Reza Malekian, and Adnan M. Abu-Mahfouz. Evaluating the LoRaWAN Protocol Using a Permanent Outdoor Testbed. In: *IEEE Sensors Journal* 19.12 (June 15, 2019), pp. 4726–4733.

[105] Dae-Young Kim et al. Adaptive Data Rate Control in Low Power Wide Area Networks for Long Range IoT Services. In: *Journal of Computational Science* 22 (Sept. 2017), pp. 171–178.

[106] Sungryul Kim and Younghwan Yoo. Contention-Aware Adaptive Data Rate for Throughput Optimization in LoRaWAN. In: *Sensors* 18.6 (May 25, 2018), p. 1716.

[107] Ruben M. Sandoval et al. Optimal Policy Derivation for Transmission Duty-Cycle Constrained LPWAN. In: *IEEE Internet of Things Journal* 5.4 (Aug. 2018), pp. 3114–3125.

[108] Guangxiang Yang and Hua Liang. A Smart Wireless Paging Sensor Network for Elderly Care Application Using LoRaWAN. In: *IEEE Sensors Journal* 18.22 (Nov. 15, 2018), pp. 9441–9448.

[109] Fadi Al-Turjman et al. Mobile Traffic Modelling for Wireless Multimedia Sensor Networks in IoT. In: *Computer Communications* 112 (Nov. 2017), pp. 109–115.

[111] Wojciech M. Kempa and Martyna Kobielnik. Transient Solution for the Queue-Size Distribution in a Finite-Buffer Model with General Independent Input Stream and Single Working Vacation Policy. In: *Applied Mathematical Modelling* 59 (July 2018), pp. 614–628.

[112] Jorge Navarro-Ortiz et al. Integration of LoRaWAN and 4G/5G for the Industrial Internet of Things. In: *IEEE Communications Magazine* 56.2 (Feb. 2018), pp. 60–67.

[113] Qihao Zhou et al. A Novel Rate and Channel Control Scheme Based on Data Extraction Rate for LoRa Networks. In: (Feb. 12, 2019). arXiv: 1902.04383 [eess].

[114] Guibing Zhu et al. Improving the Capacity of a Mesh LoRa Network by Spreading-Factor-Based Network Clustering. In: *IEEE Access* 7 (2019), pp. 21584–21596.

[115] Aamir Mahmood et al. Scalability Analysis of a LoRa Network Under Imperfect Orthogonality. In: *IEEE Transactions on Industrial Informatics* 15.3 (Mar. 2019), pp. 1425–1436.

[116] V. Prasanna Venkatesan, C. Punitha Devi, and M. Sivaranjani. Design of a Smart Gateway Solution Based on the Exploration of Specific Challenges in IoT. In: *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*. 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC). Palladam, Tamilnadu, India: IEEE, Feb. 2017, pp. 22–31.

[117] Kais Mekki et al. A Comparative Study of LPWAN Technologies for Large-Scale IoT Deployment. In: *ICT Express* 5.1 (Mar. 2019), pp. 1–7.

[118] Jetmir Haxhibeqiri et al. A Survey of LoRaWAN for IoT: From Technology to Application. In: *Sensors* 18.11 (Nov. 16, 2018), p. 3995.

[119] Mehmet Ali Ertürk et al. A Survey on LoRaWAN Architecture, Protocol and Technologies. In: *Future Internet* 11.10 (Oct. 17, 2019), p. 216.

[121] G. Gan, J. Wu, and Z. Yang. A Genetic Fuzzy K-Modes Algorithm for Clustering Categorical Data. In: *Expert Systems with Applications* 36.2 (Mar. 2009), pp. 1615–1620.

[122] H. A. A. Al-Kashoash and Andrew H. Kemp. Comparison of 6LoWPAN and LPWAN for the Internet of Things. In: *Australi of Electrical and Electronics Engineering* 13.4 (Oct. 2016), pp. 268–274.

[123] Wael Ayoub et al. Internet of Mobile Things: Overview of LoRaWAN, DASH7, and NB-IoT in LPWANs Standards and Mobility. In: *IEEE Communications Surveys & Tutorials* 21.2 (22–2019), pp. 1561–1581.

[124] LoRaWAN® for Developers | LoRa Alliance™.

[125] All About LoRa and LoRaWAN. Aug. 2019.