

Introduction

Aghiles Djoudi¹², Rafik Zitouni², Nawel Zangar¹ and Laurent George¹

¹LIGM, UMR 8049, École des Ponts, UPEM, ESIEE Paris, CNRS, UPE, France

²ECE Research Lab Paris, 37 Quai de Grenelle, 75015 Paris, France

Email: {aghiles.djoudi, nawel.zangar, laurent.george}@esiee.fr, rafik.zitouni@ece.fr

I. CONTEXT AND MOTIVATION

II. METHODOLOGY AND CONTRIBUTIONS

III. ORGANIZATION OF THE THESIS

Survey A

Aghiles Djoudi¹², Rafik Zitouni², Nawel Zangar¹ and Laurent George¹

¹LIGM, UMR 8049, École des Ponts, UPEM, ESIEE Paris, CNRS, UPE, France

²ECE Research Lab Paris, 37 Quai de Grenelle, 75015 Paris, France

Email: {aghiles.djoudi, nawel.zangar, laurent.george}@esiee.fr, rafik.zitouni@ece.fr

I. ALL

Heterogeneity of applications -> network adaptability to different application network slicing is a solution to -> fixed slicing strategies to maximize resource allocation efficiency -> (on-demand) dynamic slicing strategies maximize resource allocation efficiency

ultra-reliable low-latency communications (URLLC), enhanced mobile broadband (eMBB), and massive machine-type communications (mMTC))

| Year | Factors | Computation Model | Results interpretation |
|------|----------------|--------------------------------|---|
| 2018 | EXPLoRa-SF [1] | connected nodes time-on-air | Estimation Closeness have a high degree of Correlation with privacy score |

Table I. ADR solutions

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 [3] 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) [4]. LoRaWAN operates in ISM bands (863870 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 [4] 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.

In [5] the authors studied the impact of LoRa parameter settings (bandwidth, coding rate, spreading factor, transmission power, etc.) on energy consumption and communication reliability. They proposed a mechanism to automatically select LoRa transmission parameters that satisfy the performance requirements. This solution is optimal for a given application scenario, but it is not convenient when traffic dynamically changes.

3.2 Data Compression

The authors in [6] were interested in data compression in order to reduce the size of the data sent and thus minimize the transmission time and optimize the energy consumption. A swapped huffman tree coding has been applied to transmit the necessary data with a compression ratio of 52.3%. Data compression has been used in various LoRa sensors in the industry [4] in order to reduce energy consumption and thus reduce the data transmission time that will provide better optimization of the LoRa network. The two studies mentioned above were interested in optimizing energy consumption without worrying about the regulatory constraints relating to the channel occupancy time.

3.3 Activity Time Sharing Mechanism

[7], proposes a mechanism for sharing the channel occupancy time in order to improve the overall performance of the network. We give more details on this mechanism, to which we are interested in our work.

[7] proposed an activity time sharing mechanism in a long-range unlicensed LoRa network to face the problem of activity time limitation in the case of video surveillance applications. The proposed mechanism supposes that all devices that will participate in the sharing mechanism register with the LoRa gateway and announce their local remaining activity time (initially can be the total authorized activity time or just a fraction). Thus, the gateway computes the global activity time allowed for usage which can be an addition of the allowed time of each device Global Time (1) or just a fraction of it. After it informs it to all devices which will share it. This step is performed each cycle (every hour). As long as this global activity time allows, a node Di that exhausts its duty cycle (allocated activity time) and needs additional time to send its data borrows the remaining time from the global time. A global

view of the total remaining activity time is maintained by the LoRa Gateway (LR-BS) on reception of packets and sent back to devices at the appropriate moments.

GlobalT ime = $n \times 36$ s (1) In [7], the author did not evaluate nor propose a mechanism for selecting devices that will benefit the shared extra time. Indeed, he limited himself to serving the first applicant.

Moreover, [7] assumes that all the nodes participating in the sharing mechanism must be on standby to be able to receive from the gateway the updated information of the global activity time and the list of nodes involved in the loan. Otherwise they must wake-up periodically to receive this update. This would not correspond to the behavior of class A nodes but rather to class B nodes. We believe that the activity time sharing mechanism proposed in [7] improves the quality of service but lacks an additional time allocation mechanism by a priority classification or a strategy that satisfies a larger number of requesting devices taking into account the range of a device and its battery level in the management of the allocation of additional time. In the next section, we will describe our solution to those above-mentioned issues.

[8] QN8Y27W6

In HT-WLANs, dynamic link adaptation can be classified into two categories as follows. (i) Link adaptation in static environment: MiRA [9] is a dynamic data rate adaptation approach that selects spatial streams and rates. It is based on MIMO technology and the receivers feedback. In poor channel condition, MiRA performs excessive rate selection. Further, RAMAS [10] is a credit-based scheme that also applies MIMO streams. So, this approach incurs overhead of assigning credit to select data rate.

Deek et al. [11] proposed a rate adaptation scheme based on channel bonding. But, the mechanism can not utilize the full strength of all PHY/MAC new features. Minstrel [Multiband Atheros Driver for WiFi] is the default link adaptation algorithm in Linux system and engages the statistical information for channel overhearing. However, it is suitable only for legacy IEEE 802.11 systems. Different MCS values and MIMO are used in **das_link_2008** [12].

Feng et al. [13] developed a link adaptation scheme that applies frame aggregation. All these mechanisms do not consider all PHY/MAC enhancements of HT-WLANs along with their internal trade-offs. Thus, these approaches are not able to meet theoretical achievable data rate of IEEE 802.11n/ac in practical scenarios. Minstrel HT [New Rate Control Module for 802.11n] is the default rate adaptation methodology that is applied by the wireless driver ath9k [-802.11n Wireless Driver]. It perceives the maximum enhancements of PHY/MAC in IEEE 802.11n, but suffers from exhaustive sampling. SampleLite [A Hybrid Approach to 802.11n Link Adaptation] is a pure received signal strength indicator (RSSI) threshold-based algorithm. It can not cope up with all possible wireless network scenarios. In one of our previous works [14], a dynamic link adaptation scheme is designed for IEEE 802.11n. In this work, we consider a limited set of channel conditions measured by RSSI. In our another work [15], an adaptive

learner is designed for link adaptation in IEEE 802.11ac. Sur et al. [Practical MU-MIMO user selection on 802.11ac commodity networks] designed MUSE that is a MU-MIMO-based rate adaptation algorithm for IEEE 802.11ac networks. ESNR is an another rate selection scheme designed in [16]. Specifically, it was designed for IEEE 802.11n (MIMO). All new features of HT-WLANs are not employed in MUSE and ESNR. Moreover, their performances were not evaluated in mobile environment. (ii) Link adaptation in mobile environment: As per our knowledge, no work has yet considered SDN-based framework to design a dynamic link adaptation algorithm for HT-WLANs in mobile environment.

However, Chen et al. [17] proposed a rate adaptation algorithm, RAM, for mobile environment considering only legacy IEEE 802.11 standards. Hence, it is not adjustable with HT-WLANs.

[5] 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], [18], [19]). 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. [20], [21], [18]), transmit rate control (e.g. [22], [sourceforge.net/p/madwi/svn/HEAD/tree/madwi/trunk/ ath rate/minstrel/minstrel.txt], [23]), or a combination of the two as joint transmit power and rate control (e.g. [24], **subramanian_joint_2005**, [25]. 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 than what is feasible with LoRa. For example, the most commonly used transmit rate control algorithm Minstrel [sourceforge.net/p/madwi/svn/HEAD/tree/madwi/trunk/ ath rate/minstrel/minstrel.txt] evaluates its links every 100 ms.

[1] 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, J. Vidler, and U. Roedig, Lora for the

internet of things][26]; ii) papers aiming at analyzing the LoRa capabilities and studying their performance in specific scenarios [27][28][29][30]; 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 [31]. In [32], the authors present a performance and capability analysis of a currently available LoRa transceiver. They describe the transceivers features and demonstrate how it can be used efficiently in a wide-area application scenario. In particular, they demonstrate how unique features such as concurrent non-destructive transmissions and carrier detection can be employed. The experiments demonstrate that six LoRa nodes can form a network covering 1:5:ha in a built up environment, achieving a potential lifetime of 2 year on 2 AA batteries, delivering data within 5 s with reliability of 80%.

Conversely, the EXPLoRa heuristic [1] aims to efficiently distribute the SFs among end-devices: the EXPLoRa-SF tries to equally distribute the SFs among the total number of nodes only constrained by the Receiver Signal Strength Indicator (RSSI) values. A more sophisticated approach, EXPLoRa-AT tries to equalize the Time-on-Air of the transmitted packets among the SF channels. The solutions presented in this paper are an upgrade of the latter approach by taking into account different traffic areas as well as variable payloads and message periods, thereby being able to precisely determine and equalize the traffic load for each SF channel, expressed in terms of symbol time.

[33] I4JZZ98I

There is limited published work discussing scalability of LoRa.

Closest to this paper is the work by Petajajarvi et al. [31] and our own previous work reported in [32]. Petajajarvi et al. present an evaluation of LoRa link behaviour in open spaces. We evaluated LoRa link behaviour in built-up environments. We built upon the results reported in these papers when constructing our communication models for LoRaSim (see Section 3). This previous work, however, does not address general scalability questions of LoRa. A vast number of generic wireless simulation tools such as ns-3 [Modeling and tools for network simulation] or OMNet++ [The OM-NeT++ 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/ledown/down.php?le=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.

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.

[33] I4JZZ98I

There is limited published work discussing scalability of LoRa.

Closest to this paper is the work by Petajajarvi et al. [31] and our own previous work reported in [32]. Petajajarvi et al. present an evaluation of LoRa link behaviour in open spaces. We evaluated LoRa link behaviour in built-up environments. We built upon the results reported in these papers when constructing our communication models for LoRaSim (see Section 3). This previous work, however, does not address general scalability questions of LoRa. A vast number of generic wireless simulation tools such as ns-3 [Modeling and tools for network simulation] or OMNet++ [The OM-NeT++ 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/ledown/down.php?le=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, [32] describes LoRaBlink, an IoT protocol for LoRa transceivers designed to support reliable and energy efficient low-latency bi-directional multi-hop communication. The work in [26] provides an overview of LoRa and an in-depth analysis of its functional components. The physical and data link layer performance are evaluated by field tests and simulations. Based on the analysis and evaluations, the authors show that LoRa physical layer, thanks to the chirp spread spectrum modulation and high receiver sensitivity, offers good resistance to interference. Field tests show that LoRa can offer satisfactory network coverage up to 3 km in a suburban area with dense residential dwellings. The SF has significant impact on the network coverage, as does the data rate.

However, the performance drastically decrease when the link load increases. Limits and potentialities of LoRaWAN are studied by Voigt et al. in [27]. 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 [33][28][29].

Bor et al. in [33] provide a LoRa link behaviour by using practical experiments able to describe (i) communication range in dependence of communication settings of SF and Bandwidth (BW) and (ii) capture effect of LoRa transmissions depending on transmission timings and power. They also provided a LoRa simulator (LoRaSim) and evaluated the LoRa scalability limits in static settings comprising a single sink, and assessed how such limits can be overcome with multiple sinks and dynamic communication parameter settings.

A measurement based assessment of LoRa was also carried out in [31], which captures the Received Signal Strength Indicator (RSSI) by different locations from the base station and derives an heat map able to characterize performance as a function of the distance and of the environmental conditions (on water and on the ground). The paper also derives a channel attenuation model based on the presented measurements results.

Empirical evaluations have been also provided in [34].

In [28] the effects of interference in a single gateway LoRa network have been investigated. Unlike other wireless networks, LoRa employs an adaptive chirp spread spectrum modulation scheme, thus extending the communication range in absence of any interference. Interference is however present when signals simultaneously collide in time, frequency, and spreading factor.

Leveraging tools from stochastic geometry, the authors of [28] have formulated and solved two link-outage conditions that can be used to evaluate the LoRA behavior.

Georgiou et al. in [28] showed how the coverage probability drops exponentially as the number of network devices grows due to the interfering signals using the same spreading factor, which is concluded to be perhaps the most significant limits towards scalability on LoRa.

The paper [29] analyses the performance of the LoRa LPWAN technology by showing that, in accordance to the specifications, a single end device located close to the base station can feature an uplink data transfer channel of only 2 kbit/s at best. In terms of scalability, they show that a single LoRaWAN cell can potentially serve several millions of devices sending few bytes of data per day. Nonetheless, they showed that only a small portion of these devices can be located sufficiently far away from the base station.

Finally, [30] and [35] derive throughput behavior and capacity limits under some ideal conditions (perfect orthogonality of the SFs).

[36] 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 limita-

tions of LoRaWAN [37][38][28][33], and studing its performance under specific settings [35][39]varsier_capacity_2017. (ii) Papers proposing novel approaches and heuristics to optimize the network performance [32][30][40][41][1]. As for the first group, Adelantato et al. surveyed in [37] 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 [37] 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.

Petäjäjärvi et al. in [38] analyzed the capacity of a LoRaWAN cell. For applications requiring transmission of only a single packet per day, the cell may serve up to millions of devices. However, in case of applications reporting messages every minute, only few hundreds of devices may be hosted.

Petäjäjärvi et al. in [38] also evaluated the performance of the LoRa communication under the presence of the doppler shift. The results concluded that with SF = 12 (which enables the longest range) the communication deteriorates when relative speed exceeds 40 km/h whereas with a lower mobility it can be assured a reliable communication; finally, it was also evaluated the coverage attained by a LoRa device transmitting with SF = 12 and a transmission power of 14 dBm; as a result, it was determined the feasibility of communicating within a distance of 2-5 km, and in a range of 15-30 km on the water.

Scalability issues also have been addressed in by Bor et al. in [33] 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.

Magrin et al. in [35] implemented a model using the ns-3 simulator to study the performance in a typical urban environment. It was developed a path loss model where devices inside the buildings may be affected by building penetration losses. There were executed simulations with thousands of devices following a Pareto distribution. It was concluded that LoRaWAN with the ADR scheme may scale well only if there are numerous gateways suitably deployed across the system. i.e., a packet success rate of 95% for 15000 devices is attained only if there are 75 gateways.

Other works dealt with application-tailored deployments such as in [39] 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.

Reynders et al. in [30] developed a scheme to efficiently assign the SF and the transmission power across the devices.

On the other hand, Abdelfadeel et al. presented in [40] a fair adaptive data rate algorithm (FADR) which computes a

data-rate and transmission power allocation in order to achieve fairness in data-rate and reduce collision among nodes.

Other efforts for optimizing the network have been done tackling other solutions such as the usage of new LoRa transceivers [32] or the development of multi-hop communication for choosing the minimal Time-on-Air path [41].

[42] 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 [35].

[35] shows an assignment of SF to each ED based on the GW sensitivity by analyzing the radio frequency power signal at the GW. As a result, it lowers the probability of collisions and minimizes the ToA. Then, the GW is chosen based on the received power and SFs are allocated for the transmission. The GW is configured with 8 received paths with 3 channels in total. These receiving paths are assigned to each channel for uplink transmission. However, in this work, downlink transmission and confirmed mode are not considered. Another approach, RS-LoRa MAC protocol, aims to improve the reliability and scalability of LoRaWAN under free space path loss model [43]. RS-LoRa works in two major steps; in the first step, the GW is responsible for scheduling EDs within its range by measuring the Received Signal Strength Indicator (RSSI) and SF for each channel. In the second step, each ED decides its SF, transmission power, and channel based on the information provided by the GW. This scheduling reduces the collision by carefully selecting an SF in order to improve network reliability, scalability, and capture effect.

In order to tackle the interference and capture effect, an error model is described in [44]. This model is used for determining the range between EDs and a GW as well as analyzing the interference among various concurrent communications. The algorithm considers three different methods for assigning an efficient SF: (i) a random assignment method which assigns SFs based on uniform distribution mechanism, (ii) a fixed assignment method which assigns the same SFs all the time during the simulation period, (iii) Packet Error Rate (PER) based which finds and allocates the lowest SF for which the PER falls under a certain threshold. The PER approach for finding a suitable SF performs better than both the random and fixed-based SF assignment methods and can play a vital role in enhancing the packet delivery ratio. 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 [45]. 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 [30]. However, 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 [46]. 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.

[47] SGS7P626

Some authors deployed LoRa networks and experimentally studied its performance [48] **petajajarvi_coverage_2015-1** [49] [50] [51]. 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.

Blenn et al. [52] performed simulations based on traces from experiments and analyzed results based on real life and large scale measurements from The Things Network but their simulations are limited to the deployed scenario. To and Duda [53] presented LoRa simulations in NS-3 validated in testbed experiments. They considered the capture effect and showed the reduction of the packet drop rate due to collisions with a CSMA approach.

In a system level simulator, Haxhibeqiri et al. [54] studied the scalability for LoRaWAN deployments in terms of the number of nodes per gateway. Simulations are performed for a duty cycle of 1% but they are limited to 1000 nodes. We developed a LoRa simulator to compare the performance in different deployment scenarios for large scale networks based on an accurate model of the LoRa PHY/MAC layers. We simulate several deployment scenarios varying traffic intensity and the number of nodes. C. LoRa Evaluation and Limits Several authors evaluated performance and limits of LoRa networks.

Reynders et al. [55] evaluated Chirp Spread Spectrum (CSS) and ultra-narrow-band networks. They proposed a heuristic equation that gives Bit Error Rate (BER) for a CSS modulation as a function of SF and Signal to Noise Ratio (SNR).

Cattani et al. [56] evaluated the impact of the LoRa physical

layer settings on the data rate and energy efficiency. They evaluated the impact of environmental factors such as temperature on the LoRa network performance and showed that high temperatures degrade the Packet Delivery Ratio (PDR) and Received Signal Strength (RSS).

Goursaud et al. [57] studied the performance of the CSS modulation. They showed the possibility of interference between different SFs and evaluated co-channel rejection for all combinations of SFs.

Feltrin et al. [58] discussed the role of LoRaWAN for IoT and showed its application to many use cases. They considered the effect of non perfect orthogonality of SFs for a link level analysis.

Petajajarvi et al. [38] analyzed the scalability of a LoRa wide area network and showed its good coverage (e.g. until 30km on water for SF12 and transmit power of 14 dBm). They also showed the maximum throughput for different duty cycles per node per channel. Mikhaylov et al. [Analysis of capacity and scalability of the lora and low and power wide and area network and technology] discussed LoRa performance under European frequency regulations. They studied the performance metrics of a single end device, then the spatial distribution of several end devices. They showed LoRa strengths (large coverage and good scalability for low uplink traffic) and weaknesses (low reliability, delays, and poor performance of downlink traffic).

Bor et al. [32] presented a capability and performance analysis of a LoRa transceiver and proposed LoRaBlink protocol for link-level parameter adaptation.

Nunez et al. [59] analytically showed the potential gain of adaptive LoRa solutions that choose suitable radio parameters (i.e., spreading factor, bandwidth, and transmission power) to different deployment topologies (i.e., star and mesh). These studies provide a first view of LoRa performance and its limitations. As a conclusion we need to take into account the capture effect and imperfect orthogonality of SFs. We contribute with an accurate LoRa simulation model considering the co-channel SF interference and the gateway capture effect, allowing accurate performance analysis in large scale simulations for different deployment scenarios. Our study extends the previous evaluations of LoRa limits with the evaluation of reliability, network throughput, and power consumption from sparse to massive access deployment scenarios. D. LoRa Network Deployment Strategies Some authors studied LoRa network deployments and SF allocation strategies.

Bor et al. [33] studied LoRa transceiver capabilities and the limit supported by LoRa system. They showed that LoRa networks can scale if they use dynamic selection of transmission parameters.

Georgiou et al. [28] investigated the effects of interference in a network with a single gateway. They studied two link-outage conditions, one based on SNR and the other one based on co-SF interference. They showed, as expected, that performance decreases when the number of nodes increases and highlighted the interest of studying spatially heterogeneous deployments.

Croce et al. [60] showed the effect of the quasi-orthogonality of SFs and found that overlapped packet transmissions with different SFs may suffer from losses. They validated the findings by experiments and proposed SIR thresholds for all combinations of SFs. They remarked that LoRa networks cannot be studied as a superposition of independent networks because of imperfect SF orthogonality.

Abeele et al. [44] studied the capacity and scalability of LoRaWAN for thousands of nodes per gateway. They showed the importance of considering the capture effect and interference models. They proposed an error model from BER simulations to determine communication ranges and interference. They also analyzed three strategies of network deployments (random SF allocation, a fixed one, and according to related PDR), the last one presenting the best performance.

Lim et al. [46] analyzed the LoRa technology to increase packet success probability and proposed three SF allocation schemes (equal interval based, equal area based, and random based). They found that the equal area scheme results in better performance compared with other schemes because of the reduced influence of SFs. The state of the art indicates the interest in heterogeneous deployments and SF allocation strategies. Thus, we analyze homogeneous and heterogeneous deployments with different SF allocations as a function of the number of nodes and traffic intensity in order to show network performance and the benefits of heterogeneity for large scale networks.

[61] 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 LoRa's performance, especially its transmission range, capacity, and scalability and on interaction between LoRa transmissions.

They include [LoRa transmission parameter selection] [26] [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.

For instance, [5] introduces an algorithm for selecting proper parameters considering a desired energy consumption and link quality. In [LoRa from the city to the mountains: Exploration of hardware and environmental factors], a measurement study shows that vegetation has a big impact on LoRa transmissions. The Spreading Factor (SF) and the transmission data rate have a significant impact on the network coverage according to [26].

LoRa scalability is investigated in [28][27][37]. The authors in [28] analyze a LoRa network using a single gateway. Their results show that with an increase in the number of end devices, the coverage probability drops exponentially, due to their interfering signals.

In [27], simulation is used to show that multiple base stations improve the network performance under interference.

In [37], the authors focus on the performance impact of LoRa on higher layers. Notably from their work is that the down-

link receive window is seen as the limiting factor. This work, like the others, identifies that the main scalability limit of LoRa is its channel access protocol (essentially ALOHA) together with its rather expensive packet acknowledgements.

Finally, the work most closely related to ours is [34], an empirical study of interference between LoRa networks (a pre-print at the time of this writing). The paper investigates the interference case when one LoRa radio uses conventional LoRa modulation and the other one uses 2-GFSK modulation, which is also used in IEEE 802.15.4g. (Support for 2-GFSK is required in the LoRa specification.) The experiments use randomized packet lengths and inter-arrival times for both the sender and the interferer. The inter-arrival times are a significant fraction of (and in some cases longer than) the packet transmission times. This means that the proportion of time that the channel is interfered varies depending on the choice of LoRa transmission parameters. As a consequence, the results reflect a mix of heavily and minimally interfered packets. It is therefore hard to draw conclusions about the interference behavior, beyond the specific empirical observations. By contrast, we are doing much more controlled experiments that allow us to examine the interaction between the two modulations in detail.

[30] 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 SNR or 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.

[62] MWBMBZYE

Many work has been done in estimating the performance of LoRA networks, such as [37] [26] [29]. Although their conclusions are interesting, they only use a simplified 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 [33] [27]. 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 Configuration 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 different modules have been proposed in literature.

The authors in [35] proposed a complete LoRa module. It features MAC commands, different overlapping networks and multigateway support. Besides these interesting features, this module has some drawbacks. First, it can only send LoRa messages, so it is impossible to simulate the effect of interference. Next, similar chirp rates do not have effect on each other. Due to the chirp spread spectrum technique, spreading factor 9 with 125 kHz bandwidth has a similar chirp rate compared to spreading factor 11 with a 250 kHz bandwidth. Another small drawback is that all the gateways in this model are virtually directly connected to the network server, so the packets cannot be routed over IP.

Independently from the previous implementation, the authors in [44] 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 specification. It is highly configurable. Its flexible 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 effect of different spreading factors [30], the effect of interference [55], the reliability and scalability [43], etc. Our model can also be used to study the effect of downlink messages [63] and multiple gateways [44] [27].

[64] P3CSS7S2

LoRa and LoRaWAN technologies are relatively recent standards [26]. Most existing research based on LoRa and LoRaWAN has focused on features such as delay, range, throughput and network capacity [5] [26][54][65]. 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.

Terrasson, et al. present an energy model for Ultra-Low Power sensor nodes in [66] [A Top-Down Approach for the

Design of Low-Power Microsensor Nodes for Wireless Sensor Network]. In these papers, the authors described the modeling of a sensor node dedicated to wireless sensor network applications. However, the RF module used in this study was the CC1100 module (a short range device) which did not include the LoRa technology. An other energy estimation model is presented in [Energy Consumption Estimation of Wireless Sensor Networks in Greenhouse Crop Production], The goal of this work is to obtain a low power consumption of sensor nodes.

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.

Phui, et al. proposed a comparison of LoRaWAN classes and their power consumption in [67]. The objective of this study is to offer an experimental comparison of LoRaWAN classes to verify the published current levels of different operating modes in LoRa datasheets. The measurement results allow to estimate the lifetime of end-node devices. However, in this paper, the authors did not study the effect of different LoRaWAN parameters such as coding rate, communication range and transmission power level on the total consumed energy. Many other studies provide the power consumption of sensor nodes based on LoRa/LoRaWAN. Most of the current values were obtained from the datasheet or by empirical means [68][Design and implementation of the plug-play enabled exible modular wireless sensor and actuator network platform]LoRa Mobile-To-Base-Station Channel Characterization in the Antarctic], without developing an energy model that can estimate and optimize the energy consumption of the wireless sensors.

Casals, et al. developed current models that allow the characterization of LoRaWAN devices lifetime and energy cost in [69]. The proposed models were very important and derived from measurement results using a currently prevalent LoRaWAN platform.

However, Casals, et al. did not include the energy consumption of the processing and the sensor units. In our paper, we have modeled these units for an application scenario of connected sensor. Another major difference is that we have illustrated our energy model with optimization of LoRaWAN parameters such as the spreading factor SF, the coding rate CR, the bandwidth BW, the payload size and the communication range. Optimizing these parameters is very important to reduce the energy consumption of the sensor node. The previous works were proposed to estimate the amount of energy consumption by a sensor node. Some of these studies have not included LoRa technology into their energy models, so they used different RF transceivers which are mainly dedicated to short range communication. Other works did not study the energy optimization of sensor nodes. In fact, optimizing LoRa and LoRaWAN parameters are very interesting to reduce the energy consumption by the communicating sensor.

[70]Z38EPLZL

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

Adelantado et al. (2017) detailed the characteristics and limits of LoRa/LoRaWAN in terms of the relationships between duty cycle and throughput with various packet sizes. Duty cycle defines the ratio of the time that a device can transmit data, in order to regulate signal transmission and avoid signal collisions. They described how LoRa can be used in real-time monitoring, metering, smart transportation and logistics, but not for video surveillance.

As mentioned by Augustin et al. (2016), the packet payload is up to 255 bytes with a low duty cycle limit. Their study led to a number of open research challenges, particularly in channel management, such as time division multiple access over LoRaWAN and random-based access in unlicensed bands. To design and improve LoRa network performance, understanding network-related dependencies is essential.

Bor and Roedig (2017) identified more than 6000 various parameter combinations for LoRa networks and developed an algorithm for automatic selection on LoRa transmission parameters. The performance measurements on our LoRa network also were related to some of their parameter selections. Augustin et al.(2016) also studied network-related parameters for performance improvements and also measured network sensitivity and coverage. Our study proposed an appropriate strategy on gateway placement based on signal intensity and sensitivity in the context of three building density environments in Brisbane, Australia.

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 Elkhodr et al. and Khutsoane et al.

Petri et al. (2016) deployed a LoRa experiment using an Arduino module and a Froggy Factory LoRa shield in the city of Rennes with their key focus on the quality of service. They measured traffic between the gateway and the end devices. The traffic generated was like that of a sensor monitoring network. They focused on an observation of the performance metrics, not on overall LoRa design and deployment.

Our study fills this gap by proposing [70], 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.

Centenaro et al. (2016) stated that LoRa is a viable solution for the deployment of a Smart City and tested LoRa coverage based on one of modulation parameters and then proposed a LoRa system architecture based on the number in a particular urban population in Italy. Our proposed architecture considered practical network coverage and signal attenuation in various building density environments.

[71] WQ72KWJM

Performance evaluation over LoRa networks has been intensively reviewed by many research studies in the literature [49] [51] [Experimental performance evaluation of lorawan: A case study in bangkok]. Other research studies focused on evaluating LoRa scalability [34] while considering co-SF interference that comes from collisions when using the same SF configuration on the same channel [28] whereas others assumed that SFs on a channel are perfectly orthogonal [33] [32]. 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 [72] 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 [5].

In [46], the authors analyze several SF configuration strategies where a group of LoRa devices can be configured with similar or heterogeneous SFs based on their position from the gateway. The goal is to find the scheme that gives the best performance in terms of PDR. However, the impact of the latter configuration on network slicing has not been previously tested. Few research works recently tackled network slicing in IoT and focused on machine critical communications over various wireless networks.

The work in [73] 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 [74] 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 [75]. 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 [37]. 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.

[58] 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 [31].

Different studies have investigated the use of LoRa technology in specific fields of application, as for example, sailing monitoring systems [51], tactical troops tracking systems [50], smart cities [76], 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 [32], where the Authors study packet collisions applying a time offset between each other; in [28] 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 [29], [37]. 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 [29] and [37], 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.

In [33] the Authors estimated, through simulations, the capacity of a LoRa network assuming a simpler collision mechanism and protocol than what we used in this work; these assumptions lead to a lower capacity with respect to what we present in this paper. In [49] the Authors studied experimentally the impact on the coverage of having multiple gateways deployed in a particular area of Glasgow. Our work, though, focuses more on capacity than on coverage: a fact of relevance for LoRa, whose receive sensitivity depends on transmission parameters (the Data Rate).

[77] 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 [28][29]**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 [28], 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 [29], 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 [78][79].

In [78], authors presented a solution to improve the overall security of a LoRaWAN-based IoT system. This proposal employed proxy-nodes for performing the cryptographic operations in order to avoid heavy computation in the constrained end-nodes.

In turn, work in [79] presented an integration of IPv6 into LoRaWAN. Similar to the case of 6LoWPAN, this solution permitted a higher interoperability of the IoT network with the outside world. Although interesting, both works lacked of a detailed performance evaluation to demonstrate the impact of their proposals on the system operation. From a different perspective, other studies presented the results extracted from experimental tests conducted in diverse scenarios and situations [next citation]. These works focused on evaluating the performance of LoRaWAN under different propagation and environmental conditions.

For example, in **petric_measurements_2016-1**, a real LoRaWAN deployment was evaluated and it was found that

the base-station antenna location and elevation have great importance in the network performance. The measurements were carried out using three different base-stations. A similar experimental study was elaborated in **A Study of LoRa: Long Range & Low Power Networks for the Internet of Things**. In this case, authors focused on tuning the LoRaWAN PHY layer, i.e., LoRa, configuration parameters, identifying both Spreading Factor (SF), which will be explained later in detail, and data-rate as the principal factors impacting on the network coverage.

Another coverage study focused on LoRa, was presented in [80], but, in this case it was carried out in an indoor scenario. The presented outcomes demonstrated the robustness of LoRa in adverse industrial environments, even with high data-rates.

In turn, authors of [31] 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 [38], 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 [31] 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 OkumuraHata propagation model over realistic topographic maps.

[69] H6QDESI3

Most of the research done on LoRa/LoRaWAN has focused on features such as coverage, robustness, capacity, scalability, delay and throughput [29] [31] [37] [26] [65] [38] [34] [54] [39]. 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 [39] [81] [82] [LAMBS: Light and Motion Based Safety] [5] [Comparación de Soluciones Basadas en LPWAN e IEEE 802.15.4 Para Aplicaciones de Salud Móvil (m-Health)] [68] [Design and implementation of the plug&play enabled exible modular wireless sensor and actuator network platform] [83] [84] [85] 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][86]. 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 [81] [85] [87] [88]. 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 [87], 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).

In [81], which focuses only on optimizing downlink communications, fixed LoRaWAN settings (i.e., a single DR and acknowledged transmission) are considered. Energy consumption of 0.050.44 mJ and a battery lifetime between 13 and 1 year, respectively, are obtained for a device running on two AA batteries, when transferring data from 1 to 10 times per hour. However, one of the most important values used in the model, the sleep current (of 2 A), is without explanation significantly lower than the corresponding value in the datasheet published by the manufacturer (i.e., 30.9 A).

In [88], only the energy related to the activation of a LoRaWAN node by using the On-The-Air Activation (OTAA) mode is modeled.

In [85], the battery capacity consumed during a day by a device is calculated assuming 100 events detected and 10 frame transmissions performed by the device. The result is 222.66 Ah, which corresponds to a lifetime of 21.5 months for a 150 mAh button cell battery. Neither the impact of using acknowledgments, nor the influence of the DR configured are considered. Finally, the impact of errors due to corruption on LoRaWAN energy performance has not been modeled in publicly available works.

[89] LRLSSXZL

A first study [90] was conducted on stand-alone LoRaWAN base stations that can operate even when Internet is intermittent or non-existent and that have the ability to communicate with each other [91], in order to form a city size extensive network. In this paper, we will first look at the indexes as well as at the practitioners, to find the existing tools or to make one, in order to ensure a good coverage study based on this concept and, then, to propose a testBed for this purpose. Noted that [90] and [91] are steps of [92] that aims to study the feasibility of the smart city in developing countries, especially in Africa. However, saying that Internet is not accessible or is intermittent is only a general observation, so some people can claim to have acceptable connectivity [91] (with a round-trip time less than 100 ms (see Fig.1)) and therefore, will want to go in this direction. This is why it should be wise and judicious to propose an architectural model offering several options of communication on demand and which will remain flexible to future evolutions. This is how we thought to add the Wi-Fi protocol (which is part of the broad family of radio technologies and used for equipment implementing the IEEE 802.11x standard family) to the proposed model in [91] as it is able to achieve these goals. Indeed, it belongs to the Wi-Fi Alliance organization [Wi-Fi Alliance] and operates in the frequency band 2.5 GHz (for 802.11b, 802.11g or 802.11n) or 5 GHz for the 802.11a. We are also seeing continuous improvement of its technologies (see Table 1). Strictly speaking, we all know that sending naked data

[93] QTLET7Q6

A few papers, such as [94], present measurements of existing LoRa systems and study the actual performance of the end devices with respect to their relative distance to the gateways. They aim at optimising the parametrization of LoRa networks. However, due to the limited number of end devices considered in these studies, it is difficult to gauge the scaling properties of LoRa networks, for instance in terms of the maximum number of end devices supported with a given goodput rate. Moreover, these studies do not permit a fine-tuned analysis of collisions in LoRa networks.

In contrast, [54] scrutinizes interference in LoRa in order to establish packet collision rules, which are reproduced next in a simulation model allowing the authors to study the scalability issues of LoRa networks. A real platform to test capture in LoRa networks is used in [33]. The conclusions drawn from the tests lead to a collision model close to that derived in [54] and to similar scalability evaluation results. We recall the collision rules established in [54] and [33] in Section 3.2 since

we integrate them in our stochastic-geometric LoRa network model.

[72] D35WN7JM

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

Specifically, **atanasovski_long-range_2015** focus on LoRa applications and PHY, while in [26] some test-bed and simulation results are presented but with a low number of devices.

In **bankov_limits_2016** authors model LoRaWAN performance analytically in a scenario with high number of devices and propose solutions to improve its performance. Most of the current studies are based on simulation results. Authors in [33] implemented and used the LoRaSim simulator [www.lancaster.ac.uk/scc/sites/lora] to investigate the capacity limits of LoRa networks. They showed that a typical deployment can support only 120 nodes per 3.8 ha, although performance can be improved with multiple gateways.

The paper in [55] compared the performance of LoRa and ultra narrowband (Sigfox-like) networks, showing that ultra narrowband has a larger coverage but LoRa networks are less sensitive to interference. Finally, authors in [Decoding LoRa: realizing a modern LPWAN with SDR] presented details on the patented LoRa PHY and introduced gr-lora, an open source SDR-based implementation of LoRa PHY. All of these works, however, assume that the SFs adopted by LoRa are perfectly orthogonal, thus simplifying the analysis and consequently the network capacity. Instead, in this paper we show that, due to the imperfect orthogonality of the SFs, inter-SF collisions can prevent the correct reception of the transmissions with serious impact on LoRa performances. To the best of our knowledge, we are the first to demonstrate and quantify this performance degradation.

[95] 9AZ7VKCG

Since LoRa has so many transmission parameters to configure, the crucial task of finding a parameter combination to balance packet delivery performance and energy consumption can be difficult. The pTunes framework is a general modelling framework for selecting optimal MAC parameters based on measurements [96]. We propose a similar approach for selecting LoRa parameters. Several

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

[5] [33] **hutchison_data-aware_2013** [97] [56] [LoRa from the city to the mountains: Exploration of hardware and environmental factors] [LoRaWANTM Specification] [98] [31] [39]

or outdoor [26][LoRa from the city to the mountains: Exploration of hardware and environmental factors][98] settings.

Bor and Roedig propose an algorithm for finding the best transmission setting for a specific transmission channel [5]. 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 finding 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 efficiency for three types of channel (indoor, outdoor and underground) under different LoRa parameter settings [56]. An interesting finding 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 effect of environmental parameters on channel performance and found that high temperature at the node reduced packet delivery rate significantly. LoRaWAN is a mesh protocol for LoRa nodes. It specifies 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.

Marcelis et al. measured the spatial and temporal properties of LoRaWAN channels using both static and mobile transmitters [98]. They found that channels at the limits of the transmission range (7.5km) had very low packet reception rates. An efficient coding scheme based on convolutional codes and fountain codes was proposed for data recovery. Their experiments show that with this protocol 99% of the data can be recovered and that, for 10 byte packets, it has 21% more data recovery and 42% lower energy consumption than a naive repetition coding protocol. Research Gaps. Developing a generalised methodology for determining the best parameter values to use for very low power LoRa applications remains an open research question. Another open question is what is the best strategy for achieving high data delivery rate with low energy use. It has been shown that high data delivery rates can be achieved without over-engineering a setting to achieve the highest packet reception rate [56][98]. This paper develops this line of thinking and demonstrates that the LoRa physical layer with data-aware repetition coding is an effective solution for achieving reliable and very low energy LoRa-based sensor networks.

[99] 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, multi-path 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 [37], [33], [54], and [29], 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 [54]. 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.

For example, in [5], a transmission parameter selection algorithm for LoRa is presented with a goal to minimize energy consumption at a specific reliability level. To enable LoRaWAN to achieve a high data rate two different spreading factor allocations algorithms are presented in [1].

Similarly, in [30] power and spreading factor control algorithm is presented to achieve fairness in LoRa-based networks. Existing research work on LoRa has largely ignored the investigation of different channel access control protocols to improve the performance of LoRa in terms of reliability, throughput, and energy consumption. Therefore, here we focus on investigating the impact of a range of channel access control protocols on a LoRa-based network.

[100] D6JC9B4S

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

In [Lorawan specification] and [101], adaptive data rate (ADR) method is specified for energy optimization in static nodes. ADR involves adaptation of LoRa transmission parameters to save transmission energy using feedback mechanism based on previous communications. Similarly, a probing algorithm using trial and error method for selecting the optimal settings is illustrated in [5]. These described methods converge to an optimal setting step by step after multiple transmissions for the fixed distance between the node and the gateways. However, such methods become unsuitable in moving node scenarios. The research involving LoRa moving nodes include several works.

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][Eficient, real-time tracking of public transport, using lorawan and rf transceivers], [Long-range wireless sensor networks for geo-location 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-efcient 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.

[101] CNASAVPC

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

As reported by Petajajarvi, Mikhaylov, Roivainen, et al. [31], the performance of the LoRa's modulation is satisfactory in real-life environment experiments even under relatively strict European regulations [ERC Recommendation 70-03 relating to the use of Short Range Devices, p. 7, band h1.4]. Although multiple simultaneously emitted LoRa's transmissions can be processed by a single gateway exploiting the orthogonality of the spreading factors in addition to using multiple transmission sub-bands, the nature of the ALOHA-based access to the medium inevitably leads to the presence of transmission collisions.

As confirmed by Georgiou and Raza [28], the exponential dependence of the TOA on SF introduces exponential upper bound on the network performance.

As Adelantado, Vilajosana, Tuset-Peiró, et al. state [37], the throughput of the network is upper-bound either by the collision rate (lower DR, shorter TOA) or by the duty-cycle limitation (higher DR, longer TOA). The systems containing end-nodes whose transmission parameters are constant with respect to time are particularly affected.

As a partial solution, Bor, Roedig, Voigt, et al. suggest [33] implementing a dynamic transmission scheme, i.e. ADR in LoRaWANTM, and to densify the infrastructure by adding additional gateways. Since as seen by the network operators, the implementation of an ADR is a CAPEX-efficient way to optimize the capacity of the networks, vendors offering NM

on the market keep their algorithms as a part of the intellectual property. Contrary to this trend, the general customer availability of LoRaWANTM products catalyzed the creation of several open source NM implementation projects which include the globally recognized The Things Network (TTN) and a younger LoRa Server whose ADR algorithms are publicly available [thethingsnetwork.org/wiki/LoRaWAN/ADR]. In this paper, their algorithm was adapted and further improved as part of the effort to support the community project.

[102] 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, SIGFOX, NB-IOT, 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?][[\[28\]](#)[[\[63\]](#)] and real measurements [Performance of a low-power wide-area netw. based on LoRa technol.: Doppler robustness, scalability, and coverage] [[\[103\]](#) [\[68\]](#) [\[104\]](#) [\[105\]](#)] [Long range commun. in urban and rural environ]. Analytic models have been proposed for a variety of scenarios and communication phenomena.

Bor et al. [[\[33\]](#)] experimentally observed the capture effect of LoRa and modeled the capacity of such networks, concluding that LoRa networks with one gateway and conservative operational parameters do not scale well, while networks with dynamic adaptation of operating parameters or multiple gateways scale better.

Georgiou and Raza [[\[28\]](#)] propose an analytic model for LoRa coverage probability that takes into account Rayleigh fading, showing that LoRa networks are sensitive to network usage.

Pop et al. [[\[63\]](#)] evaluated the impact of the LoRa WAN downlink on the uplink goodput and coverage probability. They verified that the downlink becomes unable to deliver

several acknowledgment packets if too many end-devices request delivery confirmation.

Concerning the modeling of communication fading, only Georgiou and Raza [[\[28\]](#)] and Pop et al. [[\[63\]](#)] take this impairment into account. Several works have used measurements to evaluate the performance of LoRa networks.

Petäjäjärvi et al. [Performance of a low-power wide-area netw. based on LoRa technol.: Doppler robustness, scalability, and coverage] 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.

There are similar measurement reports in other environments, including a university campus [[\[103\]](#)], indoor applications [[\[68\]](#)], industry [[\[104\]](#)], dense cities downtown [[\[105\]](#)], and rural areas [Long range commun. in urban and rural environ]. 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 [EXPLoRa: Extending the perf. of lora by suitable spreading factor allocations][[\[5\]](#)[Resource efc. in low-power wide-area netw. for IoT appl] and other LPWAN technologies [Optimization of the predened number of replications in a UNB-based IoT netw][Eval. of macro diversity gain in long range ALOHA netw][Performance eval. of LoRa netw. in a smart city scenario].

Cuomo et al. [EXPLoRa: Extending the perf. of lora by suitable spreading factor allocations] suggest algorithms to replace LoRa WAN adaptive data rate strategy. The algorithms do not base the spreading factor (SF) configuration only on distance and received power, but also take into account the number of connected devices, equalizing the time-on-air (ToA) of packets in each SF.

Bor and Roedig [[\[5\]](#)] explore LoRa configuration [[\[106\]](#) 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 [[\[49\]](#)]. LoRaWAN is not the only communication protocol that uses LoRa as the physical layer: Symphony Link TM and LoRaBlink are other examples. LoRaBlink [[\[32\]](#)] 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 Specication-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) [107]. 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 [33]. 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 nodes 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 [49]. 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 [32]. Using a higher bandwidth shortens the transmission times but reduces the maximum receiver sensitivity [33]. Capacity calculations performed in [29] 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 Designers Guide,]. LoRa's coding rates are 4/(CR + 4) with CR 1, 2, 3, 4. A LoRa packets 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 Designers 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 Designers Guide,]. LoRa can detect channel activity using Carrier Activity Detection (CAD) [32]. 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], [49].

[108] 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][31][26], none of them has considered the impact of ADR on performance.

Bor et al. [33] propose an algorithm to select parameters such that transmission airtime and power are minimized. However, the authors themselves described the proposed algorithm as optimistic and impractical; their goal was to show that improvements in network capacity are possible with dynamic data rates. In contrast, we evaluate the ADR mechanism built into LoRaWAN and suggest a simple yet effective modification to improve its performance. Reynders et al. [30] presented and evaluated a mechanism to optimize the fairness of packet error rates among nodes with different spreading factors. We have applied a similar approach to derive the optimal distribution of spreading factors as a network-aware scheme in comparison with ADR+. However, the scenario considered in [30] is such that all nodes in the network can reach the gateway with every spreading factor and every power setting, i.e., all nodes are close to the gateway. In contrast, we consider a more realistic deployment scenario wherein nodes may only reach the gateway with specific spreading factors and transmission powers.

Varsier et al. **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.

Kim et al. [109] proposed a new ADR algorithm for LoRa networks at the nodes. Their algorithm requires an active feedback channel, i.e., an acknowledgment for every transmission. However, this mechanism would decrease the delivery ratio as downlink traffic has been demonstrated to have an impact on uplink throughput [63]. In contrast, we show that ADR improves the efficiency of LoRa networks without the need for acknowledgments. There are a few articles which present simulation tools 4 to evaluate the performance of LoRa networks.

Bor et al. [33] developed a Python-based discrete event simulator (called LoRaSim) to characterize the capacity of LoRa networks. However, the simulator supports only uplink

transmissions from nodes to the gateway; thus, it cannot be used to evaluate ADR.

Pop et al. [63] extended the LoRaSim simulator by adding support for downlink transmissions. The authors demonstrated that downlink transmissions in the network actually decrease the communication performance of the wireless connections.

Van den Abeele et al. [44] presented a LoRa simulator based on ns-3. The work characterizes the scalability in scenarios with both uplink and downlink transmissions; however, it does not consider dynamic configuration of transmission parameters.

[53] 253585LX

Several authors studied the issue of limits to the capacity of LoRa and its scalability to a large number of devices. Georgiou and Raza [28] provided a stochastic geometry framework for modeling the performance of a single gateway LoRa network. They showed that the coverage probability drops exponentially as the number of contending devices grows. They concluded that LoRa networks will become interference-limited rather than noise-limited in dense deployment scenarios because of the LoRaWAN access method.

Augustin et al. [26] presented throughput measurements on a testbed showing: i) less than 10% of loss rate over a distance of 2 km for SF 9-12 and ii) more than 60% of loss rate over 3.4 km for SF 12. They also simulated the LoRa behavior for a larger number of devices and showed that it behaves closely to ALOHA with the maximum channel capacity of 18% and an increasing collision ratio: for a link load of 0.48, the ratio is around 60%.

Adelantado et al. [37] explored LoRa from the point of view of the capacity and the network size. They observed that for low duty cycles, throughput is limited by collisions, whereas for higher duty cycle values, the maximum duty cycle set by the ETSI regulations prevents devices from increasing their packet transmission rates and limits throughput. For instance, for 1000 devices, the maximum packet rate per node is 38 pkt/hour (packets of 50 B) with the probability of successful reception of only 13%.

Reynders et al. [55] compared the performance of LoRa and Ultra Narrowband (UNB, SIGFOX-like) networks with regard to the range and coexistence. They showed that UNB MAC is slightly better than LoRaWAN: the latter discards both colliding packets at reception, while the UNB network enables reception of the strongest packet thanks to the capture effect. The maximal throughput of the network occurs for 10 5 devices in the network, but results in a packet loss of 63%.

Haxhibeqiri et al. [54] investigated the scalability of LoRa in terms of the number of devices per gateway. They used a simulation model based on the measurements of the interference behavior between two nodes to show that when the number of nodes with the duty cycle of 1% increases to 1000 per gateway, losses increase to 32%. However, this level of the loss rate should be considered as low compared to 90% in pure ALOHA for the same load and it results from taking into account the capture effect, which apparently plays an important role in the

LoRa behavior. We compare their measured packet loss rate and collision ratio with the simulation results.

Mikhaylov et al. [29] showed that a LoRa cell can potentially serve a large number of devices, but devices are limited to sending only a few bytes of data per day. The majority of devices need to be located in the vicinity of the gateway: only less than 10% can reside at distances longer than 5 km. Another factor that limits scalability is the use of acknowledgements as the gateway is subject to the same ETSI restrictions on the duty cycle, it cannot acknowledge each packet in a dense network.

Bor et al. [32] developed a LoRa simulation to study its scalability. They showed that a typical Smart City deployment can support 120 nodes per 3.8 ha, which is not sufficient for future IoT deployments. 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.

Ochoa et al. [59] proposed various strategies to adapt LoRa radio parameters to different deployment scenarios. Their simulation results showed that in a star topology, we can achieve the optimal scaling-up/down strategy of LoRa radio parameters to obtain either a high data rate or a long range while respecting low energy consumption. All the analyses show a large space for possible improvement of the LoRa performance. Nevertheless, the proposals for enhanced access methods need to take into account energy consumption along with performance, which is our goal in the next sections.

[110] D8LVPLDY

Raza et al. [111] present the five key challenges for LPWAN and compare proprietary and standard technologies, including LoRa, Sigfox, 802.15.4g or Dash7 to cite a few. These challenges are ultra low-power operation and long range communication, since it is expected to cover wide areas for several years. To this aim, 1-hop networks and duty cycling are employed. Low cost is also another challenge for LPWAN. Finally, scalability and quality of service are also challenges given the expected number of connected objects, and the variety of expected services.

Bor et al. [33] present LoRaSim, a simulator to evaluate the scalability of LoRa networks. The authors detail range of communication options (carrier frequency, spreading factor, bandwidth and coding rate) for a transmitter. Moreover, they study the collision avoidance scenarios as well as the maximum number of transmitters in a LoRa network. Their evaluation results show that to keep the Data Extraction Rate above 0.9, only 120 users are supported per antenna using standard LoRa settings.

Bor et al. [32] also propose LoRaBlink, a MAC and routing cross-layer scheme, above the LoRa physical layer to extend the radio coverage of the gateway. LoRaBlink is self-organized network based on beacons (that contains distance in hops from the sink) and time-slotted channel access method. Their performance evaluation show that LoRaBlink may cover a

network of 1.5 ha in a built up environment, achieve 80% of reliability while having a potential lifetime of 2 years.

Voigt et al. [27] investigate the use of directional antennae and the use of multiple base stations as methods of dealing with internetwork interference. Simulation results show that the use of multiple base stations outperforms the use of directional antennae.

Georgiou and Raza [28] provide a stochastic geometry framework for modeling the performance of a single gateway LoRa network. They model the co-spreading factor interference assuming a single bandwidth frequency for all the nodes and they measure the outage and coverage probabilities based on the signal to noise ratio and the co-spreading sequence interference. Their analysis shows that the coverage probability drops exponentially as the number of end-devices grows due to interfering signals using the same spreading sequence.

Abeele et al. [44] using a LoRa ns-3 module, perform a scalability analysis of LoRaWAN. Their work shows detrimental impact of the downstream traffic on the delivery ratio of the upstream traffic. They, also, show that increasing gateway density can ameliorate but not eliminate this effect, as stringent duty cycle requirements for gateways continue to limit downstream opportunities. The same authors show through simulations that LoRaWAN can send six times more traffic compared to pure Aloha in a single-cell LoRaWAN network for the same number of end devices per gateway when the 125-kHz channel bandwidth is used [54].

In [112], Bankov et al. consider LoRaWAN networks with class A devices operating in acknowledged mode. They detail the data transmission process, considering the difference between power of the signal from different devices and the capture effect. Indeed, they developed a generic mathematical model which can be employed to evaluate network capacity and transmission reliability of LoRaWAN networks when considering Okumura-Hata model for propagation losses. In this paper, unlike other works, we thoroughly study LoRa performance by presenting both packet success probability as well as the average success probability per frequency and spreading factor configuration as a function of distance and density. We solve an optimization problem for each frequency to find the optimal mix of nodes with different configurations that maximize the density and do not violate a minimum average success probability.

[40] 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 [33]. The CIR of all SF pairs has been calculated using simulations in [57] and validated by real LoRa link measurements in [72].

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 [113] and [38], mathematical models in [Mathematical model of lorawan channel access] and [28], or computer simulations in [33] and [35]. Almost all these works have assumed perfectly orthogonal SF s although it has been shown in [34] and [72] 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 [5] 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.

Authors in [1] proposed two SF allocation approaches, namely EXP-SF and EXP-AT, to help LoRaWAN achieve a high overall data rate. EXP-SF equally allocates SF s to N nodes based on the Received Signal Strength Indicator (RSSI), where the first $N/6$ nodes with the highest RSSI get SF 7 assigned and then the next $N/6$ nodes SF 8 and so on. EXP-AT is more dynamic than EXP-SF, where the SF allocation theoretically equalizes the airtime of nodes.

The two aforementioned works [5] and [1] 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 [30], where authors proposed a power and spreading factor control approach to achieve fairness within a LoRaWAN cell. We provide an overview of [30] 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 [79] 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].

[44] J22SN85R

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

In one of the first works on this topic, Mikhaylov et al. [29] present an analysis of the capacity and scalability of LoRa LPWANs. The authors perform an analytical analysis of the maximum throughput for a single LoRaWAN end device, taking into account such factors as RDC and the influence of receive windows. The authors note that receive windows drastically increase the time between subsequent transmissions and that RDC restrictions reduce the maximum throughput further. The authors applied the same methodology to determine the capacity of LoRaWAN based on ALOHA access. While it is true that the LoRaWAN MAC access is an ALOHA scheme, empirical data has shown that the assumptions made in pure ALOHA access do not adequately model a LoRaWAN network (see figure 4 in [33]). Specifically, it fails to model

the interference between concurrent transmissions as pure ALOHA assumes concurrent transmissions are always lost regardless of their received power levels, timings and the presence of forward error correction.

A second, but similarly lacking, pure ALOHA capacity analysis of LoRaWAN is discussed in [26]. In [37] Adelantado et al. also calculate LoRaWAN capacity as the superposition of independent ALOHA-based networks (one for each channel and for each SF). In conclusion, analyses based on pure ALOHA, fail to adequately model interference in LoRaWAN networks and therefor underestimate the capacity of LoRaWAN LPWANs.

In [28], Georgiou and Raza provide a stochastic geometry framework for modeling the performance of a single channel LoRa network. Two independent link-outage conditions are studied, one which is related to SNR (i.e. range) and another one which is related to co-spreading factor interference. The authors argue that LoRa networks will inevitably become interference-limited, as end device coverage probability decays exponentially with increasing number of end devices. The authors report that this is mostly caused by co-spreading factor interference and that the low duty cycle and chirp orthogonality found in LoRa do little to mitigate this. Finally, the authors note that the lack of a packet-level software simulation is hindering the study into the performance of LoRa. It would be interesting to combine the authors modeling of co-spreading factor interference with our ns-3 error model, as in the SINR approach all interference is treated as noise.

The work of Bor et al. [33] studies the limit on the number of transmitters supported by a LoRa system based on an empirical model. The authors performed practical experiments that quantify communication range and capture effect of LoRa

transmissions. These findings were used to build a purpose-built simulator, LoRaSim, with the goal of studying the scalability of LoRa networks. The authors conclude that LoRa networks can scale quite well if they use dynamic transmissions parameter selection and/or multiple sinks. Our study confirms that multiple sinks drastically improve scalability, even though we use a very different approach for modeling interference. Furthermore, our study goes deeper into modeling LoRaWAN as the LoRaWAN MAC layer is modeled and the impact of confirmed messages and downstream traffic is studied.

The recent work presented by Pop et al. in [63] studies the impact of bidirectional traffic in LoRaWAN by extending the LoRaSim simulator to include bidirectional LoRaWAN communication. The resulting simulator is named LoRaWANSim. Both our ns-3 module and LoRaWANSim allow to study the scalability of LoRaWAN networks. Both works find that duty cycle limitations at the gateway limit the number of downlink messages (Ack or data) a gateway can send. This problem grows worse as the end device density increases, but can be partially mitigated by increasing gateway density (see section V-C).

The authors of [63] correctly identify that the absence of an acknowledgement, does not necessarily mean that the link quality has decreased and that a node should decrease its data

rate for subsequent retransmissions. Actually, decreasing the data rate might exacerbate this problem as detailed in [63]. Notable differences between the two simulators include that the LoRaWANSim manuscript is limited to single gateway network, while the ns-3 module provides support for multi-gateway LoRaWAN networks. Secondly, the collision models are quite different. The ns-3 module builds on the error model derived from the complex baseband BER simulations, while LoRaWANSim reuses the empirical model from LoRaSim. Both collision models support the capture effect as well as modeling interference. Under capture effect, we understand the ability to receive an interfered transmission in the presence of one or more interferers as long as the SNR of the interfered transmission is sufficiently high for the transmission to be received error-free. The LoRaWANSim collision model incorrectly assumes perfect orthogonality between spreading factors, while the ns-3 module counts every transmission on the same channel with a different spreading factor as interference. Furthermore, the LoRaWANSim manuscript does not mention the 10% RDC restriction that applies in the sub-band of the RW2 channel in the EU. This underestimates the downlink capacity in RW2. Thirdly, the SpectrumPhy model for the LoRa PHY in ns-3 enables modeling inter-technology interference, which could facilitate studies on the interference between 802.11ah on LoRaWAN. Finally, the LoRaWANSim simulator does not appear to be open source although the manuscript is still under revision at this time.

[52] P8FP2R7R

Vangelista et al. **atanasovski_long-range_2015** present LoRa as one of the most promising technologies for the wide-area IoT and mention that LoRa exhibits certain advantages over the LPWAN technologies Sigfox TM , Weightless TM , and On-Ramp Wireless. The robust chirp signal modulation and the low energy usage in combination with the low cost of end-devices together with the fact that the LoRa Alliance is also actively marketing and pushing interoperability aspects, makes LoRaWAN an interesting choice among available LPWAN technologies. 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.

The expected coverage of LPWANs and especially LoRa was also analyzed by Petäjäjärvi et al. [31], who conducted measurements in Finland. Using a single base-station with an antenna gain of 2 dBi and configuring the nodes to send packets at SF12 using 14 dBm of transmit power, connectivity within a 5 km range in urban environments and 15 km in open space were found to result in packet-loss ratios smaller than 30%. Measurements conducted by sending packets from a node mounted to a boat revealed that packets can be sent over a distance of almost 30 km.

Petäjäjärvi et al. [39] also tested the usage of LoRa in indoor

environments. The results show that very low packet-loss is to be expected with only one base-station to cover an average university campus.

Bor et al. [32] conducted experiments using multiple nodes transmitting data using LoRa. Experiments were conducted in which two devices sent packets at different power levels, but the same spreading factor, to estimate the influence of concurrent transmissions. Additionally, a new media access control (MAC), LoRaBlink, was developed to enable direct connection of nodes without using LoRaWAN. Contrary to the above-mentioned work, we have not confined to a single base-station, but we have provided extensive measurements based on the large-scale The Things Networks network and for a duration of 8 months.

[114] S33CL98I

Several recent related works have sought to evaluate the performance of LoRa networks using analytic modeling [28] [33] [45] [63] [112] and real measurements [38] [39] [103] [68] [104] [105] [115] [116] [113]. 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 [1] [5] [117] [27], UNB/SIG F OX in [118], and for others in [119], [35]. Analytic models have been proposed for a variety of scenarios and communication phenomena.

Bor et al. [33] experimentally observed the capture effect of LoRa and modeled the capacity of such networks, concluding that LoRa networks with only one gateway and conservative operational parameters do not scale well, while networks with dynamic adaptation of operating parameters or multiple gateways tend to scale better.

Georgiou and Raza [28] propose an analytic model that takes into account Rayleigh fading and allows to equate the coverage probabilities of nodes in a network considering two outage probabilities: disconnection and collision. Their work shows that LoRa networks are sensitive to network density.

Gupta et al. [45] modeled the IoT traffic considering periodic messages and event-generated messages and analyzed the impact of traffic variations in LoRa WAN networks. They were able to identify that LoRa gateways do not handle well burst events, which generate a significant amount of messages in a short period, especially when there is a spatial or temporal correlation in the transmission behavior of IoT devices.

Pop et al. [63] evaluated how LoRa WAN downlink impacts LoRa uplink goodput and coverage probability. They considered the medium access control (MAC) layer and, through simulations, verified that if too many end-devices request delivery confirmation, the downlink becomes unstable and unable to deliver several acknowledgment packets, thus forcing network nodes to retry their transmissions, ultimately flooding the network.

Bankov et al. [112] proposed a mathematical model for LoRa WAN channel access taking into account the capture effect and using the Okumura-Hata model, but without fading.

Concerning the modeling of communication fading, only Georgiou and Raza [28] and Pop et al. [63] take this impair-

ment into account, to the best of our knowledge. Several works have used measurements to evaluate the performance of LoRa networks.

Petäjäjärvi et al. [38], [39] 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 [103], indoor applications [Indoor deployment of lowpower wide area networks (LPWAN): A LoRaWAN case study], industry [104], dense cities downtown [105], [Usability of LoRaWAN technol], smart metering [116], and rural areas [113]. 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 .

Cuomo et al. [1] propose algorithms to replace LoRa WAN adaptive data rate strategy. The proposed algorithms do not base the configuration of the spreading factor (SF) on distance and received power measurements, but take into account the number of connected devices, allowing the equalization of the time-on-air (ToA) of the packets in each SF.

Bor and Roedig [5] explore LoRa configuration parameters such as SF, bandwidth, coding rate and transmission power, which result in 6, 720 possible settings, and proposes an optimization problem that minimizes energy spent on data transmission while meeting required communication performance.

Qin and McCann [117] approach the optimization of LPWANs efficiency from a resource allocation perspective. They use game theory to derive an algorithm that allows network nodes to decide which channel and SF to use and, for each channel/SF group, which is the optimal transmit power that maximizes data extraction rate.

Voigt et al. [27] consider the inter-network interference that is likely to take place when several independent LoRa networks get deployed too close. Authors consider using directional antennas in network nodes and using multiple gateways in the network. Results show that directional antennas enhance data extraction rate, although the use of multiple gateways in the covered area tends to perform better. Besides the above works, some authors have explored techniques similar to those proposed in this paper for LoRa , UNB/S IG F OX , or for LPWANs in general.

Mo et al. [118] investigated the optimal number of message replications for use in UNB/S IG F OX networks.

Song et al. [119] consider the macro reception diversity of long-range ALOHA networks, where augmented spatial

diversity arises from allowing several base stations to receive the same packet.

Magrin et al. [35] developed a simulation model for the NS-3 network simulator with which they showed that LoRa networks support a large number of nodes and maintain reasonable network quality if several gateways are carefully placed. 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 [28] as baseline model and extend it to incorporate the proposed techniques. Our work on message replication differs from [118] 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 [118] does not. Our approach using multiple receive antennas differs from [119] and [35] 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.

[120] DFN3W8GF

Noreen et al. described LoRa PHY performance theoretically [121]. 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 [sktict.com], a leading telecommunication operator in Korea, presented its experimental results on LoRa transmission ranges at the IoT-LPWAN 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.

Augustin et al. [26] reported the result of suburban experiments in Paris. Cisco 910 as a gateway was installed outside a residential window at a height of about 5 meters. LoRa nodes were placed in 5 different locations so that their distances from the gateway were 600 m, 1400 m, 2300 m, 2800 m, and 3400 m. There were 3 options for SF values: 7, 9, and 12. The adaptive data rate and retransmission were not used. Experimental results show that the PDR is about 90% when SF 12 is used at 2800 m, and the ratio drops to less than 10% at SF 9. At 3400 m, only about 40% of the received packets have SF 12.

Wixted et al. [49] evaluated the LoRa coverage in the Cen-

tral Business District in Glasgow, Scotland. A LoRa gateway was installed on the roof of the 7th floor of a university building. Carrying nodes with SF 12 in a backpack, a tester moved around the city and measured the received signal strength. The experimental result shows that it is possible to successfully receive packets 2.2 km in the south direction and 1.6 km in the north direction (before passing through the hill). The authors mentioned that the east and west directions were not fully tested but were measured to about 2 km.

Erbati et al. [Analysis of LoRaWAN technology in an Outdoor and an Indoor Scenario in Duisburg-Germany] analyzed LoRa performance in an outdoor environment in Duisburg, Germany. A LoRaWAN gateway was installed in an eight-story building located in the city of Duisburg. Experiments were conducted in a non-LOS real-world environment where there exist various obstacles, such as buildings, cars, and trees. The distance varied from 300 m to 1850 m, and 500 data were sent

from each spot. The following parameter settings were used: 21 bytes of payload, a CR of 4/5, SF 10, and 125 kHz bandwidth. Results show that the Received Signal Strength Indication (RSSI) value decreases logarithmically with increasing distance. A data packet is not decoded anymore when the distance becomes greater than 1,850 m, where a PDR of 69% is achieved.

Newmann et al. [68] deployed both the gateway and nodes inside to study LoRa performance in an indoor environment. In the experiments, the distances between nodes and the gateway were 0.50–60 m. The result shows that RSSI decreases rapidly on a log scale with increasing distance compared with the outdoor environment, but it is reliable enough to be used indoors. However, when the distance becomes very small, packet errors occur frequently as a result of a bad cyclic redundancy check (CRC).

Adelantado et al. [37] referred to the duty-cycle constraint as a major factor limiting LoRa MAC performance. A higher SF value allows a wider communication range but also increases the ToA, which inevitably leads to an increase in the off-period duration under the constraint. The problem can be serious because, generally, nodes using high SF values are much more numerous than nodes using low SF values. The study applied 3 channels and a 1% duty cycle as the fixed parameters and simulated the packet reception ratio for a given packet length and number of nodes. As a result, although the packet length is small at 10 bytes, the packet reception rate decreases exponentially because of the increasing number of collisions as the number of nodes increases. Notably, it converges to almost zero when more than 1500 nodes are deployed. On the contrary, a small number of nodes decreases the probability of collision, but the throughput is limited by the duty cycle.

Abee et al. [44] analyzed the scalability of large-scale LoRaWAN using the NS-3 simulator. LoRa MAC has been modeled as a pure ALOHA network that does not consider the capture effect or interference characteristics. Instead, the authors generated a LoRa error model from a preliminary simulation that measured the bit error rate on the baseband

and used it as an interference model in the simulator. The simulation result shows that the packet reception ratio varies depending on how SF values are assigned when nodes are deployed. Performance is very poor when all nodes have the same SF. Assigning SF values randomly results in better performance than the fixed SF assignment, but it is still low: only a 20% ratio with 5000 nodes. They improved the performance by about two times compared with the random assignment method by using a PER (packet error rate) strategy that assigns the minimum SF value at which the PER falls below a certain threshold. The authors ran another experiment that set the size of an application payload to 8 bytes and used a single channel. In this experiment, they increased the data period from 600 s to 60,000 s, varied the number of gateways from 1 to 4, and then evaluated MAC performance. Their results confirm that the PDR increases exponentially as the data period increases. With respect to the gateways, the PDR achieves 70%, 89%, and 96% when 1, 2, and 4 gateways are deployed in an area with 10,000 nodes that send data every 600 s. This illustrates the possibility of using multiple gateways to improve LoRa performance in a dense environment. Our experiments used the LBT scheme on multiple channels, so it is impossible to compare our results to theirs directly. However, both results indicate that there is a trade-off between the number of nodes and the data period to achieve a given level of LoRa MAC performance.

To and Duda [53] applied a Carrier-Sense Multiple Access (CSMA) technique to LoRa to improve LoRa scalability. When a node has a packet to send, it performs a Clear Channel Assessment (CCA) to test whether there is an ongoing transmission on a channel. The node transmits the packet only if the channel is clear; otherwise, it backs off for a random interval of time, with k slots of 1 s, where $k \in [0, 2^n - 1]$. The authors further proposed CSMA-10, a variant of CSMA, where the node listens to the channel, named the Clear Channel Gap (CCG), for an interval of 10 ms before attempting transmission. Their experimental results show that the proposed method can mitigate the probability of packet collision. This allows the deployment of 5500 nodes with a 90% PDR, whereas a general LoRaWAN deployment only allows less than 500 nodes to achieve the same packet delivery ratio.

[122] EKZHVKL

Mikhaylov et al. [29] analyzed and assessed the throughput of the LoRa technology, determining the airtime of a packet. Therefore, it is possible to estimate the maximum number of nodes that can communicate with a Gateway module. The purpose of this paper was to analyze the ALOHA communication mechanism. The results are obtained at an empirical level. A mathematical model of the access mechanism to the communication channel is presented in [112]. A certain threshold of the network load is also calculated in this paper by estimating the throughput. When this threshold value is reached, the PER (Packet Error Rate) parameter increases rapidly towards 1, because the packet relaying causes an avalanche effect leading to the saturation of the communication channel.

Bor et al. [32] analyzed the access mechanism of the communication channel. From the obtained results, it can be seen that when the same SF (spreading factor) is used, by both the receiver and the transmitter, the packets are received even if a third node attempts to interfere with the transmission. Thus, the separation of channels by using different SF proves effective. One or two simultaneous transmissions can be received with high probability, if there is a separation of at least 3 symbol periods between them. The paper also analyzes the possibility of implementing a carrier activity detection mechanism. An algorithm for the automatic selection of communication parameters is presented in [5] so as to achieve a performance level as high as possible, at the same time ensuring energy efficiency.

Blenn N. et al. [52] obtained a series of experimental and empirical results by analyzing the influence of the payload on the quality of the received signal. The experiments have been conducted over an 8-month period, with the results showing that the LoRa channel occupancy rate is not evenly distributed, a fact that contributes to a decrease in performance. This phenomenon is based on the fact that the majority of LoRa nodes use the default settings programmed by the manufacturer, a fact that causes the overload of certain channels. The purpose of the paper was to use certain user-defined communication channels according to the RF (Radiofrequency) environment congestion.

In [38], the Doppler effect over the LoRa modulation is analyzed, by performing a series of experimental measurements. From the obtained results, the authors conclude that, by using SF=12, a communication range of up to 30 km with a packet loss of 62 % can be obtained.

In **vangelista_frequency_2017** the mathematic model of the LoRa modulation and also of the demodulation process based on signal processing theory is presented. The paper also presents a comparison of the performance levels between the LoRa modulation and the FSK (Frequency-Shift Keying) modulation, regarding the value of the encoded bit error rate parameter. The obtained results show that when an AWGN (Additive White Gaussian Noise) channel is used, the LoRa modulation ensures a higher performance level.

Liao et al. [123] analyze the effect of the simultaneous LoRa transmissions over the performance level. The paper proposes the integration of a CT (Concurrent Transmission) type flooding into the technology. CT is an extremely efficient flooding type protocol that has recently revolutionized the design of the multihop networks based on the IEEE-802.15.4 standard. Instead of attempting to avoid packet collision, CT enables more nodes to send packets with the same content simultaneously, at the same time moment. By allowing such synchronized packets collisions, CT enables rapid back-to-back relaying of packets that considerably improve the efficiency of the network. The paper proposes the implementation of such a strategy for increasing the performance level of the LoRa networks by introducing a multihop mechanism. None of the papers evaluate the maximum number of nodes that can communicate on a channel, taking into consideration a

real implementation scenario.

[124] AAHXK9LU

Multiple works exist for the performance evaluation of LPWANs [29], [125], [37], [55]. 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.

In [29], multiple annuli LoRaWAN R cell structure is well modelled and illustrated with a few applicative scenarios. This structure is considered in our article but channel effects and capture effect are added to our model thus making it more complete and realistic.

In [125], the performances of a random Frequency Division Multiple Access (random FDMA) scenario are studied in the pure Aloha case, but the capture effect with little overlap between packets is not considered.

In [55], the performances in terms of packet delivery ratio and throughput of LoRaWAN R

and Sigfox R are simulated. However, the simulation process and the numerous network parameters are not exposed enough thus lacking of transparency and possibility of reuse.

In [37], some interesting insights on the limits of LoRaWAN R are given, but again the model is based on Poisson process and there is no possible extension to account for the capture effect. In this article, in order to give the limit of the performance, we study the outage probability and throughput of LoRaWAN R and Sigfox R when every node is transmitting as frequently as possible, according to either the ISM band duty cycle constraints or technology-related constraints, which result in message sending periods in the order of 1 to 10 minutes. This scenario of the saturation throughput could be that of packet and object tracking systems [The internet of things: a survey. Information Systems Frontiers]. Other less frequent IoT scenarios such as water & gas metering can be evaluated using our model thanks to its flexibility.

[126] 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 [31], [49] 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.

Other works, such as [5], examine the impact of the modulation parameters on the single communication link between an ED and its GW, without considering more complex network configurations. Note that, from the GW perspective, ACK packets are not distinguishable from any other DL packet and, hence, are subject to the same rules and constraints.

To obtain more general results, [124] uses a stochastic geometry model to jointly analyze interference in the time and frequency domains. It is observed that when implementing a packet repetition strategy, i.e., transmitting each message multiple times, the failure probability reduces, but clearly the average throughput decreases because of the introduced redundancy.

In [127] the author proposes closed-form expressions for collision and packet loss probabilities and, under the assumption of perfect orthogonality between SFs, it is shown that the Poisson distributed process does not accurately model packet collisions in LoRaWAN. Network throughput, latency and collision rate for uplink transmissions are analyzed in [128] that, using queueing theory and considering the Aloha channel access protocol and the regulatory constraints in the use of the different sub-bands, points out the importance of a clever splitting of the traffic in the available sub-bands to improve the network performance.

In [112] the authors present a mathematical model of the network performance, taking into account factors such as the capture effect and a realistic distribution of SFs in the network. However, the model does not include some important network parameters, preventing the study of their effect on the network performance.

A step further is made in [129] where the authors develop a model that makes it possible to consider various parameters configurations, such as the number of ACKs sent by the GW, the SF used for the downlink transmissions, and the DC constraints imposed by the regulations. In this work, however, multiple retransmissions have not been considered.

The study presented in [63] features a system-level analysis of LoRaWAN, and gives significant insights on bottlenecks and network behavior in presence of downlink traffic. However, besides pointing out some flaws in the design of the LoRaWAN medium access scheme, this work does not propose any way to improve the performance of the technology.

In [44], system-level simulations are again employed to assess the performance of confirmed and unconfirmed messages and show the detrimental impact of confirmation traffic on the overall network capacity and throughput. Here, the only proposed solution is the use of multiple gateways, without deeply investigating the specificities of the LoRaWAN standard.

In [62] a module for the ns-3 simulator is proposed and used for a similar scope, comparing the single and multigateway scenarios and the use of unconfirmed and confirmed messages. In this case, the authors correctly implement the GWs multiple reception paths, but do not take into account their association to a specific UL frequency, which usually occurs during network setup: indeed, the number of packets that can be received simultaneously on a given frequency can not be greater than the number of reception paths that are listening on that frequency. Also in this case, the study only focuses on the performance analysis, without proposing any improvement.

The authors in [101], [108] 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 [30], the authors compute the optimal SFs distribution to minimize the collision probability and propose a scheme to improve the fairness for nodes far from the station by optimally assigning SFs and transmit power values to the network nodes, in order to reduce the packet error rate.

In [130] it is shown how the use of a persistent-Carrier Sense Multiple Access (p-CSMA) MAC protocol when transmitting UL messages can improve the packet reception ratio. However, attention must be paid to the fact that having many EDs that defer their transmission because of a low value of p may lead to channel under-utilization.

In [131], 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.

[132] 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 [106], [48], [31], [26]. 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 [38], [77]. Extensive research regarding the ADR scheme has resulted in additions and modifications targeting network performance metrics such as scalability [109], throughput [1], PDR [30], and contention [133]. 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 [109]. 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 [101], additions such as adjusting data rates before incrementing Tx, the averaging of SNR history and accounting for hysteresis is recommended.

A contention aware ADR approach, proposed by [133], tracks the number of nodes per SF and aims to increase the

number of devices using low SFs in order to maximise the networks throughput.

In [77], 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 [48] 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 [77]. 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 [49]. To investigate the impact of downlink traffic in which ACKs materially influence performance, the popular LoRaSim simulator was extended into LoRaWANSim [63]. 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 [63]. 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 [44].

[63] 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], [28], [88] and simulators [33], [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 provide any insights on the interplay between downlink traffic and the gateways 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 [33], 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.

[134] 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 [39], 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], [28], the performance of their different configurations [5], and how these types of networks tolerate download traffic [63], 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, [37] highlighted that the effects of TDC limitations jeopardize the actual capacity of large-scale 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.

[135] 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.

Rémi Bonnefoi, et al[136] present reinforcement learning algorithms to reduce the collision probability and the network latency.

Jaehyu Kim, et al [137] propose a secure device-to-device link establishment scheme to guarantee security features in LoRaWAN and evaluated the performance by comparing the energy consumption.

Ramon Sanchez-Iborra, et al **alsanchez-iborra_performance_2018-1** evaluate the performance of LoRaWAN on different real scenarios. Also the energy consumption is considered by some researchers.

Analytical models of energy consumption and the performance of LoRaWAN are analyzed by Lluís Casals, et al [69]. The impact of LoRa transmission parameter selection on communication performance is presented by Bor, Martin; Roedig, Utz [5], a link probing regime to save energy consumption and improve communication reliability is also developed.

Considering energy constraints of wireless sensor networks in IoT, Fadi Al-Turjman, et al [138] propose a novel traffic model to investigate the effects of multi-hop communication of IoT. C. Tunc, et al [139] 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 and Finn to consider a server in a WDM optical access network.[140]. Wang and Chang exploit the equilibrium joining strategies for customers in an M/M/1 queue with working vacations and vacation interruptions [141]. Kempa and Kobielnik consider a single-channel finite-buffer queuing model with a general independent input stream of customers [142]. 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 [143] aims to integrate LoRaWAN with 4G/5G mobile networks, which will allow the current infrastructures of mobile network operators are reutilized.

On the contrary, although LoRaWAN is rising with a promising future, Adelantado, et al[37] give an objective description of limitations of LoRaWAN in accordance with the development questions in research and application.

Pedro Cruz, et al **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 Luis Alonso, et al [144].

Antonis Tzounis, et al [145]**tzounis_internet_2017** discuss the IoT solutions towards the modernization of agriculture and present a survey of IoT technologies in the agricultural sector.

Yonghua Songa, et al [145] propose a IoT solution in energy management system based on the LPWAN technology.

René Brandborg Sørensen, et al [128] have investigated the performance of LoRaWAN including delay, collision rate, and throughput. These articles have provided valuable works for the research in this paper. According to the literature review, most of the works about LoRa has focused on the study of the network characteristics such as coverage, time delay, or energy consumption. As we know, the network performance and the availability such as collision rate and SDR are significant in elderly care solutions. Moreover, the data delivery policy among the nodes and the gateway in the star topology network is very important for avoiding packets collision and increasing SDR. But it has been carried out limited studies. Therefore, we mainly focus on the performance improvement and optimal cluster allocation policies for maximizing SDR in LoRaWAN. The network QoS factors, including average time delay, SDR and energy consumption, are analyzed. Based on this topic, a smart WPSN solution in elderly care system is implemented and evaluated experimentally.

[146] 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 [33], [108], **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 [108] slightly modify the basic ADR scheme. The maximum operation in the SNR of several recent packets is replaced with the average function.

In [1], EXPLoRa-SF selects spreading factors based on the total number of connected nodes and EXPLoRa-AT equalizes the time-on-air of packets transmitted at the different spreading factors.

In addition, the authors in [5] propose an link probing regime to select transmission parameters in order to achieve lower energy consumption.

In [30], the authors present a scheme to optimize the packet error rate fairness to avoid near-far problems.

[147] 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], [63], 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 [148]. 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 [109]. Subsequent research [37] [30] [43] [108] considered an SF distribution scheme based only on the distance from the node to gateways.

Adelantado et al. [37] showed that an excessive number of nodes (28 % of the network) should use the largest SF (SF12) to ensure the coverage of urban cells. This approach only considered the path loss and ignored the airtime when using SF12.

Reynders et al. [43] provided SF distribution scheme to balance the packet error rate [30] and lightweight scheduling to group the nodes into different power level and selected SFs to improve the reliability and scalability of the LoRaWAN network.

Slabicki et al. [108] showed that a network-aware approach can further improve the delivery ratio of dense networks by using global knowledge of the node locations. However, the proposed algorithms based on ADR are not considered in the same way as for parallel transmission.

Cuomo et al. [1] extended the work on parallel transmission in a single-hop LoRa network. Specifically, these authors proposed to use the airtime to balance the nodes of each group

with a specific SF and to attempt to use a high data rate to offload the traffic to less congested larger SFs. However, these strategies cannot be easily applied to a multi-hop LoRa network because of their lack of consideration in regards to multi-hop relays. In contrast to a single hop network in which the airtime of different groups is only decided by the number of nodes and their data rates, the airtime in a multi-hop network is also determined by the hop count of each subnet. Moreover, the connectivity between multiple relays is still not considered when conducting SF allocation in a single-hop LoRa network. B. SF ALLOCATION IN A MULTI-HOP NETWORK As compared to single-hop LoRa networks, multi-hop networks are more flexible to extend the coverage and more efficient to improve the data transmission without increasing the number of gateways. A practical strategy that transforms the topology from a star to a mesh network when the coverage range exceeds 3.2 km was proposed [59].

Moreover, it was shown [149] that constructing a mesh LoRa network is a good solution to solve the coverage problem in extensively shadowed urban areas. However, few reports that discuss the SF allocation in a mesh LoRa network have been published. On the other hand, similar to SF allocation in a mesh LoRa network, the adoption of parallel transmission by using multiple channels has already been implemented in conventional multi-hop single-root data collection networks.

In data collection networks, the proposed protocols [150], [151] usually construct a static channel assignment approach to maintain the simplicity of channel coordination.

Other researchers [150] proposed TMCP to assign different channels to disjoint trees and operate parallel transmissions among sub-trees for data collection. However, the paper does not discuss the balance between different sub-trees.

A more recent proposal [151] involved a multi-channel multi-path data collection protocol based on Basketball Net Topology (BNT), which maintains not only a tree-based topology but also the connectivities between peer nodes located at the same height in the tree. This protocol enables child nodes to rejoin the network, even when their parent node disappeared from the original tree structure, by using peer links to communicate with other nodes. However, the connectivities of peers extend the hop counts to the sink node, which inversely increases the airtime of the entire network. The use of LoRa enables the coverage range to be extended when a lower data rate with a larger SF is chosen. As compared to multi-channel assignment algorithms, we needed to consider an approach that would decrease the hop count of each sub-tree using a different SF while ensuring that the airtime between different sub-trees remains balanced.

II. IoT APPLICATIONS REQUIREMENTS [152]

A. Summary and discussion

| Use cases | Resources | Mobility | Heterogeneity | Scalability | QoS | Data | standardization | Safety | Security |
|----------------------------|-----------|----------|---------------|-------------|-----|------|-----------------|--------|----------|
| 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 II. Main IoT challenges[153] + [154] hancke_role_2012 [155]

III. IoT WIRELESS NETWORKS (NORMS & STANDARDS)

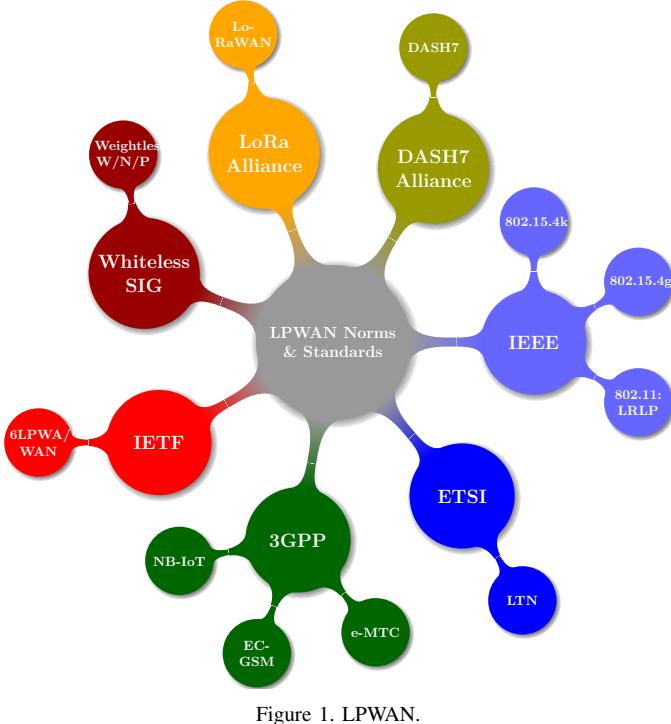


Figure 1. LPWAN.

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.

A. SigFox

B. IETF

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 $\approx 38\%$ 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 $\approx 5\%$, as seen in figures 2.3 and 2.4.

C. 3GPP

1) *NB-IoT*:

2) *EC-GSM*:

3) *e-MTC*:

D. IEEE

1) *IEEE 802.11*:

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 $4+1+1 \approx 4.5\%$; 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.

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.

E. LoRaWAN

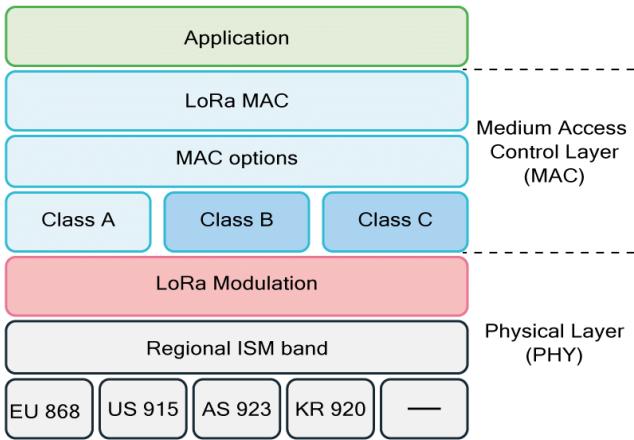


Figure 2. uhuhuh.

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.

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.

1) ALIANCE:

a) Class-A:

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.

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.

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.



Figure 3. Class A.

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 negotiates its ping-interval. LoRa Server is then able to schedule downlink transmissions on each occurring ping-interval.

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.

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.

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 Class Higher.

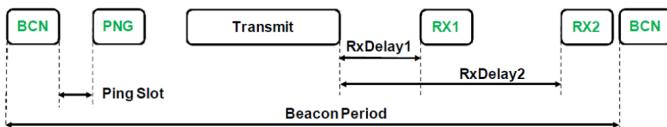


Figure 4. Class B.

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.

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.

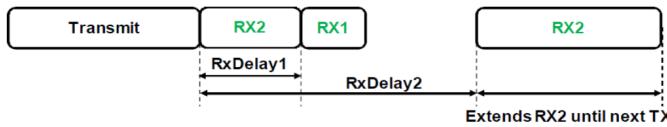


Figure 5. Class C.

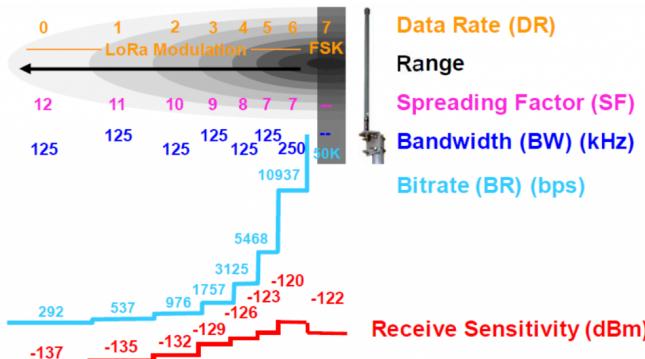


Figure 6. LoraWan Parameters.

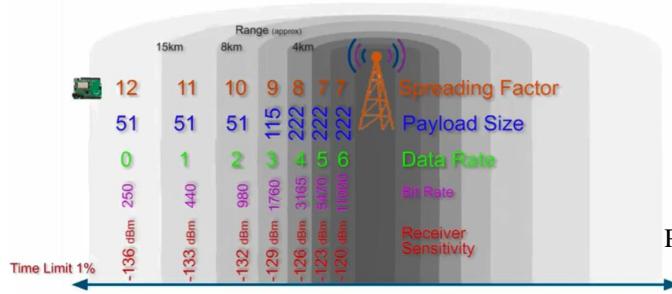


Figure 7. .

2) **SEMTECH:** LoRa has four configurable parameters: **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).

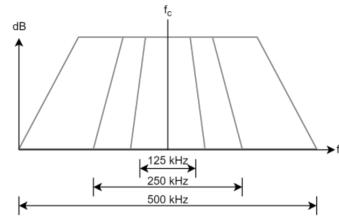


Figure 8. .

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. 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 Transmission power:

Payload Payload length:

F. Divers

1) **IPLC:**

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

Table III. uyuyuy

| 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 IV. gaddam_comparative_2018

2) *BACnet*:

3) *Z-WAze*:

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].

G. Summary and discussion

Chirp Spread Spectrum (Proprietary) (**CSS**) Carrier Frequency (**CF**) Forward error correction (**FEC**) Path loss (**PL**) Link Symmetry (**LS**) Base Station (**BS**) **CSS** Direct Sequence Spread Spectrum (**DSSS**) Ultra narrow band (**UNB**) Data Rate (**DR**) Adaptive Data Rate (**ADR**) Coding Rate (**CR**) Bandwidth (**BW**)

| Characteristics | CF [Hz] | 6LoWPAN | LoRaWAN | SigFox | NB-IoT | INGENU | TELENSA |
|----------------------------------|---------------------------|----------------|----------------------------------|-------------------------------------|---------------------------------------|-------------------------------------|------------------------------------|
| Modulation | 2.4G | O-QPSK | - | - | QSPSK \downarrow | | 2-FSK |
| | 915M | BPSK | LoRa | BPSK \uparrow , GFSK \downarrow | QSPSK n-tone \uparrow | RPMA \uparrow , CDMA \downarrow | 2-FSK |
| | 868M | BPSK | LoRa/GFSK | BPSK \uparrow , GFSK \downarrow | /4-QPSK 1-tone | | 2-FSK |
| Chwidth [KHz] | 2.4G | 16 | 500 - 125 | - | 180 | | |
| | 915M | 10 | 64+8 \uparrow , 8 \downarrow | \times | - | 40 | \times |
| | 868M | 1 | 10 | 360+40 | \times | \times | \times |
| CF [MHz] | 2.4G | \times | - | - | - | \times | ISM |
| | 915M | 902-929 | 902-928 | 902 | \times | \times | 915M |
| | 868M | 868-868.6 | 863-870 and 780 | 868.18-868.22 | \times | \times | 868M/430M |
| BW [Hz] | 2.4G | 5M | - | - | 200K | 1M | \times |
| | 915M | 2M | 125K-500K | \times | \times | \times | \times |
| | 868M | 600M | 125K-250K | 0.1K-1.2K | \times | \times | \times |
| DR [bps] | 2.4G | 250M | - | - | - | 78K \uparrow , 19,5K \downarrow | \times |
| | 915M | 40M | 980-22K | \times | 234.7 \downarrow , 204.8 \uparrow | \times | \times |
| | 868M | 20M | LoRa: 0.3K-37.5K FSK: 50K | 0.1K \uparrow , 0.6K \downarrow | \times | \times | 62.5 \uparrow , 500 \downarrow |
| CR [dBm] | 2.4G | -85 | - | - | - | \times | \times |
| | 915M | -92 | \times | \times | \times | \times | \times |
| | 868M | -92 | -137 | -137 | \times | \times | \times |
| 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 | \times | - | - | - | \times | \times |
| | 915M | \times | \times | \times | \times | \times | \times |
| | 868M | \times | Multi BS | Multi BS | \times | \times | \times |
| msg/day | 2.4G | \times | - | - | - | \times | \times |
| | 915M | \times | \times | \times | \times | \times | \times |
| | 868M | \times | Unlimited | 140 \uparrow , 4 \downarrow | Unlimited | \times | \times |
| PL B | 2.4G | \times | - | - | - | \times | \times |
| | 915M | \times | \times | \times | \times | \times | \times |
| | 868M | \times | 51 - 243 | 12 \uparrow , 8 \downarrow | 1600B | 10KB | \times |
| Coding/Spreading | | DSSS | CSS | UNB | \times | DSSS | UNB |
| Proprietary | | \times | \times | \checkmark | \times | \times | \times |
| Topology | | \times | Star, Stars | Star | \times | Star, Tree | Star |
| ADR | | \times | \checkmark | \times | \times | \checkmark | \times |
| Security | | \times | AES 128b | \times | \times | AES 256B | \times |
| LS | | \times | \checkmark | \times | \times | \times | \times |
| FEC | | \times | AES 128b | \times | \times | \checkmark | \times |
| 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 | | X | | |
| Scalability | | | 1M \uparrow , 100K \downarrow | | 55 k | | |
| <i>Linkbudget</i> $[dB]$ | | | 157 | | 154 | | |
| <i>Sensitivity</i> $[dBm]$ | | | -124 - (-134) | | -141 | | |
| Multi-hop supporter | | | X | | X | | |
| Addressing | | | Broadcast \uparrow , Unicast \downarrow | | Unicast \uparrow , Both \downarrow | | |
| Peak current | | | 32 mA | | 120300 mA | | |
| Sleep current | | | 1 A | | 5 A | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

Table V. LPWAN Characteristics [156], [lopes_design_2019](#), [raza_low_22](#), [157]

| 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 VI. LPWAN Characteristics [158]

| Standard | 802.15.4k | 802.15.4g | Weightless-W | Weightless-N | Weightless-P | DASH 7 Alliance |
|--------------------------|-------------------------------------|-------------------------|----------------------------|--|----------------------------|---------------------------------|
| Modulation | DSSS, FSK | MR-[FSK, OFDMA, QPSK] | 16-QAM, BPSK, QPSK, DBPSK | UNB DBPSK | GMSK, offset-QPSK | GFSK |
| <i>BW</i> | 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 |
| <i>DR</i> | 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 |
| <i>PL</i> | 2047B | 2047B | >10B | 20B | >10B | 256B |
| Security | AES 128b | AES 128b | AES 128b | AES 128b | AES 128/256b | AES 128b |
| Forward error correction | ✓ | ✓ | ✓ | ✗ | ✓ | ✓ |

Table VII. 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 |
| <i>BW</i> (MHz) | 868/915/2400 | 2400 | 860-960 | 868/908/2400 | 700-900 | |
| MAC | TDMA, CSMA/CA | TDMA | ALOHA | CSMA/CA | OFDMA | |
| <i>DR</i> (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 | 8bit | 16bit | | | | |

Table VIII. IoT cloud platforms and their characteristics al-fuqaha_internet_24

| | 802.15.4 | 802.15.4e | 802.15.4g | 802.15.4f |
|-----------------------------|---|---|---|--|
| <i>CF</i> | 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) |
| <i>DR</i> Differences | Upto 250kbps - 127 bytes | Upto 800kbps Time sync and channel hopping N/A | Up to 800kbps Phy Enhancements Up to 2047 bytes | Mac and Phy Enhancements N/A N/A Active RFID |
| <i>PL</i> Range Goals | 1 75+ m General Low-power Sensing/Actuating Many | 1 75+ m Industrial segments | Upto 1km Smart utilities | Upto 1km Smart utilities |
| Products | Few | | Connode (6LoWPAN) | LeanTegra PowerMote |

Table IX. IEEE 802.15.4 standards [159]

| 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 |
| <i>BR</i> Mb/s | 6-54 | 327 | 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 | ✓ | ✗ | ✓ | ✓ | ✓ |
| <i>DR</i> | <640 kbps | 250 kbps | 106424 kbps | ✓ | ✓ |

Table X. An example table.

Payload size (*PS*)Signal-to-interference & noise ratio (*SINR*)

| Spreading Factor (<i>SF</i>)/ <i>BW</i> | | 125kHz | | | | | | 250kHz | | | | | | 500kHz | | | | | | |
|---|---|-----------------------|-----------|---------|-------------|-----------|-------|-------------|-----------|---------|-------------|-----------|-------------|-----------|---------|-------------|-------------|-----------|---------|-------------|
| | | varsier_capacity_2017 | | [160] | [161] | <i>PS</i> | Byte | Sensitivity | <i>BR</i> | Rx wind | <i>SINR</i> | <i>PS</i> | Sensitivity | <i>BR</i> | Rx wind | <i>SINR</i> | Sensitivity | <i>BR</i> | Rx wind | <i>SINR</i> |
| - | - | Sensitivity | <i>BR</i> | Rx wind | <i>SINR</i> | <i>PS</i> | Byte | Sensitivity | <i>BR</i> | Rx wind | <i>SINR</i> | <i>PS</i> | Sensitivity | <i>BR</i> | Rx wind | <i>SINR</i> | Sensitivity | <i>BR</i> | Rx wind | <i>SINR</i> |
| - | - | [dBm] | [kb/s] | [ms] | [dB] | Byte | [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 XI. 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 intenses. *PS*

| <i>DR</i> | Modulation | | | <i>PS</i> | <i>BR</i> |
|-----------|------------|-----------------|-----------|-----------|-----------|
| | <i>SF</i> | <i>BW</i> [kHz] | <i>CR</i> | 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 XII. oioioi

IV. IoT PROTOCOLS

| Application protocol | DDS | CoAP | AMQP | MQTT | MQTT-SN | XMPP | HTTP |
|----------------------|---------------------|----------------|---------------------|-----------------------|---------|-------------|------|
| Service discovery | mDNS | | | DNS-SD | | | |
| | IPv6 RPL | | | UDP/TCP | | | |
| Transport Network | IPv4/IPv6 | | | 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 XIII. Standardization efforts that support the IoT

A. Application

- 1) *LwM2M*:
 - 2) *CBOR*:
 - 3) *DTLS*:
 - 4) *OSCOAP*:

Figure 9. CoAP Header.

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.

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | | |
|---------------------------------|---|---|-----|---|-----------|---|--------|--|--|
| Message Type | | | UDP | | QoS Level | | Retain | | |
| Remaining length | | | | | | | | | |
| Variable length header | | | | | | | | | |
| Variable length message payload | | | | | | | | | |

Figure 10. CoAP Header.

6) MQTT:

- ▶ **Message type:** CONNECT (1), CONNACK (2), PUBLISH (3), SUBSCRIBE (8) and so on
 - ▶ **DUP flag:** indicates that the message 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

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

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.

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 ask 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 @

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

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 clients temperature

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.

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

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.

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

Table XIV. Application protocols comparison

- ➡ 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.

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
 - ➡ 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 ask 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 @

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

B. Network

1) 6TiSCH:

2) OLSRv2:

3) AODVv2:

4) LoRaWAN:

5) ROHC:

6) IPHC:

7) SCHC:

8) NHC:

9) ROLL:

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.

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-of-order. 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 fragments 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 well-known 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 contexts stored prefix, IPHC compresses the prefix to the contexts 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 XV. 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 its 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 stores 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 XVI. Routing protocols comparison _rp12_

C. 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 XVII

1) Sharing the channel:

a) TDMA, FDMA, CDMA, TSMA:

2) Transmitting information:

a) TFDM, TDSSS, THSS:

D. Radio

1) Digital modulation:

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

2) Hierarchical modulation:

a) QAM, WDM:

3) Spread spectrum:

a) SS, DSSS, FHSS, THSS:

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:

$$Power (in dB) = 10 * \log(10) (50/1) = 10 * 1.7 = 17 \text{ dBm}$$

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.

E. Summary and discussion

V. SDN PLATFORMS

| Plan de contrôle | Plan de gestion | Plan de données |
|---|--|--|
| Contrôle d'admission Réservation de ressources Routage Signalisation | Contrôle et supervision de QoS Gestion de contrats QoS mapping Politique de QoS | Contrôle du trafic Façonnage du trafic Contrôle de congestion Classification de paquets Marquage de paquets Ordonnancements des paquets Gestion de files d'attente |

Table XVIII. An example table.

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

[162] This **shortfall** can be resolved by using the **SDN approach**.

Io_survey_2017 **SDN** also enhances better control of **heterogeneous** network infrastructures.

Io_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.

Io_survey_2017 In cellular networks, OpenRoads presents an approach of introducing **SDN** based **heterogeneity** in wireless networks for operators.

Io_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**.

Io_software_2017 Thus a bottom-up approach application of **SDN** to the realisation of **heterogeneous IoT** is suggested.

Io_software_2017 Perhaps a more complete IoT architecture is proposed, where the authors apply **SDN** principles in IoT **heterogeneous** networks.

redefined_2017 it provides the **SDWSN** with a proper model of network management, especially considering the potential of **heterogeneity** in SDWSN.

| Management architecture | Management feature | Controller configuration | Traffic Control | Configuration and monitoring | Scalability and localization | Communication management |
|---|--|--------------------------|-----------------|------------------------------|------------------------------|--------------------------|
| luo_sensor_2012 Open Flow | SDN support protocol | Distributed | in/out-band | ✓ | ✓ | ✓ |
| costanzo_software_2012 SDWN | Duty cycling, 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-hop | Traffic and energy control | Centralized | in-band | | | ✓ |
| Tiny-SDN | Network task measurement | - | in-band | | | |

Table XIX. SDN-based network and topology management architectures. **ndiaye_software_2017**

VI. BLOCKCHAIN

A. 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 Byzantine Fault Tolerant (PBFT)

B. Summary and discussion

REFERENCES

- [1] F. Cuomo, M. Campo, A. Caponi, G. Bianchi, G. Rossini, and P. Pisani, “ [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)*, 00000, Rome: IEEE, Oct. 2017, pp. 1–8.
- [2] N. Benkahla, H. Tounsi, Y.-Q. Song, and M. Frikha, “ [Enhanced Dynamic Duty Cycle in LoRaWAN Network](#) ”, in *Ad-Hoc, Mobile, and Wireless Networks*, N. Montavont and G. Z. Papadopoulos, Eds., vol. 11104, 00000, Cham: Springer International Publishing, 2018, pp. 147–162.
- [3] A. Springer, W. Gugler, M. Huemer, L. Reindl, C. Ruppel, and R. Weigel, “ [Spread Spectrum Communications Using Chirp Signals](#) ”, in *IEEE/AFCEA EUROCOMM 2000. Information Systems for Enhanced Public Safety and Security (Cat. No.00EX405)*, 00147, Munich, Germany: IEEE, 2000, pp. 166–170.
- [4] L. Alliance. (o). LoraWAN Specification, [Online]. Available: <https://lora-alliance.org/resource-hub/lorawanr-specification-v103> (visited on 08/30/2019).
- [5] M. Bor and U. Roedig, “ [LoRa Transmission Parameter Selection](#) ”, in *2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 00000, Ottawa, ON: IEEE, Jun. 2017, pp. 27–34.
- [6] Y. S. Jang, M. R. Usman, M. A. Usman, and S. Y. Shin, “ [Swapped Huffman Tree Coding Application for Low-Power Wide-Area Network \(LPWAN\)](#) ”, in *2016 International Conference on Smart Green Technology in Electrical and Information Systems (ICSGTEIS)*, 00006, Denpasar, Indonesia: IEEE, Oct. 2016, pp. 53–58.
- [7] C. Pham, “ [QoS for Long-Range Wireless Sensors Under Duty-Cycle Regulations with Shared Activity Time Usage](#) ”, *ACM Transactions on Sensor Networks*, vol. 12, no. 4, pp. 1–31, Sep. 22, 2016, 00014.
- [8] R. Karmakar, S. Chattopadhyay, and S. 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*, 00000, Chennai, India: ACM Press, 2017, pp. 1–6.
- [9] I. Pefkianakis, Y. Hu, S.-B. Lee, C. Peng, S. Sakellaridi, and S. Lu, “ [Window-Based Rate Adaptation in 802.11n Wireless Networks](#) ”, *Mobile Networks and Applications*, vol. 18, no. 1, pp. 156–169, Feb. 2013, 00008.
- [10] D. 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*, 00041, Vancouver, AB, Canada: IEEE, Oct. 2011, pp. 331–340.
- [11] L. Deek, E. Garcia-Villegas, E. Belding, S.-J. Lee, and K. Almeroth, “ [Joint Rate and Channel Width Adaptation for 802.11 MIMO Wireless Networks](#) ”, in *2013 IEEE International Conference on Sensing, Communications and Networking (SECON)*, 00057, New Orleans, LA, USA: IEEE, Jun. 2013, pp. 167–175.
- [12] Qiuyan Xia, M. Hamdi, and K. Ben Letaief, “ [Open-Loop Link Adaptation for Next-Generation IEEE 802.11n Wireless Networks](#) ”, *IEEE Transactions on Vehicular Technology*, vol. 58, no. 7, pp. 3713–3725, Sep. 2009, 00038.
- [13] K.-T. Feng, P.-T. Lin, and W.-J. Liu, “ [Frame-Aggregated Link Adaptation Protocol for Next Generation Wireless Local Area Networks](#) ”, *EURASIP Journal on Wireless Communications and Networking*, vol. 2010, no. 1, p. 164651, Dec. 2010, 00021.
- [14] R. Karmakar, S. Chattopadhyay, and S. Chakraborty, “ [Dynamic Link Adaptation for High Throughput Wireless Access Networks](#) ”, in *2015 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, 00009, Kolkata, India: IEEE, Dec. 2015, pp. 1–6.
- [15] ———, “ [Dynamic Link Adaptation in IEEE 802.11ac: A Distributed Learning Based Approach](#) ”, in *2016 IEEE 41st Conference on Local Computer Networks (LCN)*, 00006, Dubai: IEEE, Nov. 2016, pp. 87–94.
- [16] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, “ [Predictable 802.11 Packet Delivery from Wireless Channel Measurements](#) ”, *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 4, p. 159, Aug. 16, 2010, 00584.
- [17] X. Chen, P. Gangwal, and D. Qiao, “ [RAM: Rate Adaptation in Mobile Environments](#) ”, *IEEE Transactions on Mobile Computing*, vol. 11, no. 3, pp. 464–477, Mar. 2012, 00040.
- [18] S. Lin, J. Zhang, G. Zhou, L. Gu, J. A. Stankovic, and T. He, “ [ATPC: Adaptive Transmission Power Control for Wireless Sensor Networks](#) ”, in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems - SenSys ’06*, 00739, Boulder, Colorado, USA: ACM Press, 2006, p. 223.
- [19] B. Zurita Ares, P. G. Park, C. Fischione, A. Speranzon, and K. H. Johansson, “ [On Power Control for Wireless Sensor Networks: System Model, Middleware Component and Experimental Evaluation](#) ”, in *2007 European Control Conference (ECC)*, 00076, Kos: IEEE, Jul. 2007, pp. 4293–4300.
- [20] J. Monks, V. Bharghavan, and W.-M. 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)*, 00919, vol. 1, Anchorage, AK, USA: IEEE, 2001, pp. 219–228.
- [21] A. Muqattash and M. 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*, 00180, Roppongi Hills, Tokyo, Japan: ACM Press, 2004, p. 210.
- [22] M. Lacage, M. H. Manshaei, and T. 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*, 00000, Venice, Italy: ACM Press, 2004, p. 126.

- [23] S. H. Y. Wong, S. Lu, H. Yang, and V. Bharghavan, “ [Robust Rate Adaptation for 802.11 Wireless Networks](#) ”, in *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking - MobiCom '06*, 00751, Los Angeles, CA, USA: ACM Press, 2006, p. 146.
- [24] K. Ramachandran, R. Kokku, Honghai Zhang, and M. Gruteser, “ [Symphony: Synchronous Two-Phase Rate and Power Control in 802.11 WLANs](#) ”, *IEEE/ACM Transactions on Networking*, vol. 18, no. 4, pp. 1289–1302, Aug. 2010, 00109.
- [25] P. Chevillat, J. Jelitto, and H. L. Truong, “ [Dynamic Data Rate and Transmit Power Adjustment in IEEE 802.11 Wireless LANs](#) ”, *International Journal of Wireless Information Networks*, vol. 12, no. 3, pp. 123–145, Jul. 2005, 00035.
- [26] A. Augustin, J. Yi, T. Clausen, and W. Townsley, “ [A Study of LoRa: Long Range & Low Power Networks for the Internet of Things](#) ”, *Sensors*, vol. 16, no. 9, p. 1466, Sep. 9, 2016, 00000.
- [27] T. Voigt, M. Bor, U. Roedig, and J. Alonso, “ [Mitigating Inter-Network Interference in LoRa Networks](#) ”, Nov. 2, 2016, 00062. arXiv: 1611.00688 [cs].
- [28] O. Georgiou and U. Raza, “ [Low Power Wide Area Network Analysis: Can LoRa Scale?](#) ”, *IEEE Wireless Communications Letters*, vol. 6, no. 2, pp. 162–165, Apr. 2017, 00000. arXiv: 1610.04793.
- [29] K. Mikhaylov, J. Petäjäjärvi, and T. Hänninen, “ [Analysis of Capacity and Scalability of the LoRa Low Power Wide Area Network Technology](#) ”, p. 6, 2016, 00000.
- [30] B. Reynders, W. Meert, and S. Pollin, “ [Power and Spreading Factor Control in Low Power Wide Area Networks](#) ”, in *2017 IEEE International Conference on Communications (ICC)*, 00000, Paris, France: IEEE, May 2017, pp. 1–6.
- [31] J. Petajajarvi, K. Mikhaylov, A. Roivainen, T. Hanninen, and M. Pettissalo, “ [On the Coverage of LPWANs: Range Evaluation and Channel Attenuation Model for LoRa Technology](#) ”, in *2015 14th International Conference on ITS Telecommunications (ITST)*, 00343, Copenhagen, Denmark: IEEE, Dec. 2015, pp. 55–59.
- [32] M. Bor, J. Vidler, and U. Roedig, “ [LoRa for the Internet of Things](#) ”, p. 6, 00170.
- [33] M. C. Bor, U. Roedig, T. Voigt, and J. M. Alonso, “ [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*, 00240, Malta, Malta: ACM Press, 2016, pp. 59–67.
- [34] K. Mikhaylov, J. Petajajarvi, and J. Janhunen, “ [On LoRaWAN Scalability: Empirical Evaluation of Susceptibility to Inter-Network Interference](#) ”, in *2017 European Conference on Networks and Communications (EuCNC)*, 00046, Oulu, Finland: IEEE, Jun. 2017, pp. 1–6.
- [35] D. Magrin, M. Centenaro, and L. Vangelista, “ [Performance Evaluation of LoRa Networks in a Smart City Scenario](#) ”, in *2017 IEEE International Conference on Communications (ICC)*, 00000, Paris, France: IEEE, May 2017, pp. 1–7.
- [36] F. Cuomo, J. C. C. Gamez, A. Maurizio, L. Scipione, M. Campo, A. Caponi, G. Bianchi, G. Rossini, and P. Pisani, “ [Towards Traffic-Oriented Spreading Factor Allocations in LoRaWAN Systems](#) ”, in *2018 17th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, 00000, Capri: IEEE, Jun. 2018, pp. 1–8.
- [37] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia, and T. Watteyne, “ [Understanding the Limits of LoRaWAN](#) ”, Feb. 13, 2017, 00000. arXiv: 1607.08011 [cs].
- [38] J. Petäjäjärvi, K. Mikhaylov, M. Pettissalo, J. Janhunen, and J. Iinatti, “ [Performance of a Low-Power Wide-Area Network Based on LoRa Technology: Doppler Robustness, Scalability, and Coverage](#) ”, *International Journal of Distributed Sensor Networks*, vol. 13, no. 3, p. 155014771769941, Mar. 2017, 00127.
- [39] J. Petäjäjärvi, K. Mikhaylov, R. Yasmin, M. Hämäläinen, and J. Iinatti, “ [Evaluation of LoRa LPWAN Technology for Indoor Remote Health and Wellbeing Monitoring](#) ”, *International Journal of Wireless Information Networks*, vol. 24, no. 2, pp. 153–165, Jun. 2017, 00060.
- [40] K. Q. Abdelfadeel, V. Cionca, and D. Pesch, “ [Fair Adaptive Data Rate Allocation and Power Control in LoRaWAN](#) ”, Feb. 28, 2018, 00000. arXiv: 1802.10338 [cs].
- [41] B. Sartori, S. Thielemans, M. Bezunartea, A. Braeken, and K. Steenhaut, “ [Enabling RPL Multihop Communications Based on LoRa](#) ”, in *2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 00017, Rome: IEEE, Oct. 2017, pp. 1–8.
- [42] A. Farhad, D.-H. Kim, and J.-Y. Pyun, “ [Scalability of LoRaWAN in an Urban Environment: A Simulation Study](#) ”, in *2019 Eleventh International Conference on Ubiquitous and Future Networks (ICUFN)*, 00000, Zagreb, Croatia: IEEE, Jul. 2019, pp. 677–681.
- [43] B. Reynders, Q. Wang, P. Tuset-Peiro, X. Vilajosana, and S. Pollin, “ [Improving Reliability and Scalability of LoRaWANs Through Lightweight Scheduling](#) ”, *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1830–1842, Jun. 2018, 00033.
- [44] F. V. den Abeele, J. Haxhibeqiri, I. Moerman, and J. Hoebelke, “ [Scalability Analysis of Large-Scale LoRaWAN Networks in Ns-3](#) ”, May 16, 2017, 00000. arXiv: 1705.05899 [cs].
- [45] V. Gupta, S. K. Devar, N. H. Kumar, and K. P. Bagadi, “ [Modelling of IoT Traffic and Its Impact on LoRaWAN](#) ”, in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, 00010, Singapore: IEEE, Dec. 2017, pp. 1–6.
- [46] J.-T. Lim and Y. Han, “ [Spreading Factor Allocation for Massive Connectivity in LoRa Systems](#) ”, *IEEE Communications Letters*, vol. 22, no. 4, pp. 800–803, Apr. 2018, 00000.
- [47] M. N. Ochoa, L. Suraty, M. Maman, and A. Duda, “ [Large Scale LoRa Networks: From Homogeneous to Heterogeneous Deployments](#) ”, in *2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 00000, Limassol: IEEE, Oct. 2018, pp. 192–199.
- [48] M. Aref and A. 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*, 00000, Odessa, Ukraine: IEEE, Sep. 2014, pp. 19–23.
- [49] A. J. Wixted, P. Kinnaird, H. Larijani, A. Tait, A. Ahmadiania, and N. Strachan, “ [Evaluation of LoRa and LoRaWAN for Wireless Sensor Networks](#) ”, in *2016 IEEE SENSORS*, 00129, Orlando, FL, USA: IEEE, Oct. 2016, pp. 1–3.
- [50] W. San-Um, P. Lekbunyasin, M. Kodyoo, W. Wongsuwan, J. Makfak, and J. Kerdssi, “ [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)*, 00014, Phuket, Thailand: IEEE, Jan. 2017, pp. 32–35.
- [51] L. Li, J. Ren, and Q. 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)*, 00046, Jackson, WY, USA: IEEE, Feb. 2017, pp. 77–80.

- [52] N. Blenn and F. Kuipers, “ [LoRaWAN in the Wild: Measurements from The Things Network](#) ”, Jun. 9, 2017, 00000. arXiv: 1706.03086 [cs].
- [53] T.-H. To and A. Duda, “ [Simulation of LoRa in NS-3: Improving LoRa Performance with CSMA](#) ”, in *2018 IEEE International Conference on Communications (ICC)*, 00000, Kansas City, MO: IEEE, May 2018, pp. 1–7.
- [54] Jetmir Haxhibeqiri, Floris Van den Abeele, Ingrid Moerman, and Jeroen Hoebeke, “ [LoRa Scalability: A Simulation Model Based on Interference Measurements](#) ”, *Sensors*, vol. 17, no. 6, p. 1193, May 23, 2017, 00000.
- [55] B. Reynders, W. Meert, and S. Pollin, “ [Range and Coexistence Analysis of Long Range Unlicensed Communication](#) ”, in *2016 23rd International Conference on Telecommunications (ICT)*, 00000, Thessaloniki, Greece: IEEE, May 2016, pp. 1–6.
- [56] Marco Cattani, Carlo Boano, and Kay Römer, “ [An Experimental Evaluation of the Reliability of LoRa Long-Range Low-Power Wireless Communication](#) ”, *Journal of Sensor and Actuator Networks*, vol. 6, no. 2, p. 7, Jun. 15, 2017, 00000.
- [57] C. Goursaud and J. M. Gorce, “ [Dedicated Networks for IoT: PHY / MAC State of the Art and Challenges](#) ”, *EAI Endorsed Transactions on Internet of Things*, vol. 1, no. 1, p. 150 597, Oct. 26, 2015, 00000.
- [58] L. Feltrin, C. Buratti, E. Vinciarelli, R. De Bonis, and R. Verdone, “ [LoRaWAN: Evaluation of Link- and System-Level Performance](#) ”, *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 2249–2258, Jun. 2018, 00000.
- [59] M. N. Ochoa, A. Guizar, M. Maman, and A. Duda, “ [Evaluating LoRa Energy Efficiency for Adaptive Networks: From Star to Mesh Topologies](#) ”, in *2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 00000, Rome: IEEE, Oct. 2017, pp. 1–8.
- [60] D. Croce, M. Gucciardo, S. Mangione, G. Santaromita, and I. Tinnirello, “ [Impact of LoRa Imperfect Orthogonality: Analysis of Link-Level Performance](#) ”, *IEEE Communications Letters*, vol. 22, no. 4, pp. 796–799, Apr. 2018, 00000.
- [61] C. Orfanidis, L. M. Feeney, M. Jacobsson, and P. Gunningberg, “ [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)*, 00000, Rome: IEEE, Oct. 2017, pp. 1–8.
- [62] B. Reynders, Q. Wang, and S. Pollin, “ [A LoRaWAN Module for Ns-3: Implementation and Evaluation](#) ”, in *Proceedings of the 10th Workshop on Ns-3 - WNS3 '18*, 00000, Surathkal, India: ACM Press, 2018, pp. 61–68.
- [63] A.-I. Pop, U. Raza, P. Kulkarni, and M. Sooriyabandara, “ [Does Bidirectional Traffic Do More Harm Than Good in LoRaWAN Based LPWA Networks?](#) ”, Dec. 14, 2017, 00000. arXiv: 1704.04174 [cs].
- [64] T. Bouguera, J.-F. Diouris, J.-J. Chaillout, R. Jaouadi, and G. Andrieux, “ [Energy Consumption Model for Sensor Nodes Based on LoRa and LoRaWAN](#) ”, *Sensors*, vol. 18, no. 7, p. 2104, Jun. 30, 2018, 00000.
- [65] K. E. Nolan, W. Guibene, and M. 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)*, 00140, Paphos, Cyprus: IEEE, Sep. 2016, pp. 439–444.
- [66] G. Terrasson, A. Llaria, and R. Briand, “ [System Level Dimensioning of Low Power Biomedical Body Sensor Networks](#) ”, in *2014 IEEE Faible Tension Faible Consommation*, 00006, Monaco, Monaco: IEEE, May 2014, pp. 1–4.
- [67] P. S. Cheong, J. Bergs, C. Hawinkel, and J. Famaey, “ [Comparison of LoRaWAN Classes and Their Power Consumption](#) ”, in *2017 IEEE Symposium on Communications and Vehicular Technology (SCVT)*, 00000, Leuven: IEEE, Nov. 2017, pp. 1–6.
- [68] P. Neumann, J. Montavont, and T. 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)*, 00099, New York, NY: IEEE, Oct. 2016, pp. 1–8.
- [69] L. Casals, B. Mir, R. Vidal, and C. Gomez, “ [Modeling the Energy Performance of LoRaWAN](#) ”, *Sensors*, vol. 17, no. 10, p. 2364, Oct. 16, 2017, 00000.
- [70] J. J. Chen, J. E. Chen, V. Liu, L. Fairbairn, and L. Simpson, “ [A Viable LoRa Framework for Smart Cities](#) ”, p. 16, 2018, 00000.
- [71] S. Dawaliby, A. Bradai, and Y. Pousset, “ [Adaptive Dynamic Network Slicing in LoRa Networks](#) ”, *Future Generation Computer Systems*, vol. 98, pp. 697–707, Sep. 2019, 00000.
- [72] D. Croce, M. Gucciardo, I. Tinnirello, D. Garlisi, and S. Mangione, “ [Impact of Spreading Factor Imperfect Orthogonality in LoRa Communications](#) ”, in *Digital Communication. Towards a Smart and Secure Future Internet*, A. Piva, I. Tinnirello, and S. Morosi, Eds., vol. 766, 00000, Cham: Springer International Publishing, 2017, pp. 165–179.
- [73] A. Nakao, P. Du, Y. Kiriba, F. Granelli, A. A. Gebremariam, T. Taleb, and M. Bagaa, “ [End-to-End Network Slicing for 5G Mobile Networks](#) ”, *Journal of Information Processing*, vol. 25, no. 0, pp. 153–163, 2017, 00072.
- [74] Y. Gadallah, M. H. Ahmed, and E. Elalamy, “ [Dynamic LTE Resource Reservation for Critical M2M Deployments](#) ”, *Pervasive and Mobile Computing*, vol. 40, pp. 541–555, Sep. 2017, 00008.
- [75] Y. Hao, D. Tian, G. Fortino, J. Zhang, and I. Humar, “ [Network Slicing Technology in a 5G Wearable Network](#) ”, *IEEE Communications Standards Magazine*, vol. 2, no. 1, pp. 66–71, Mar. 2018, 00010.
- [76] V. A. Stan, R. S. Timnea, and R. A. 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)*, 00014, Ploiesti, Romania: IEEE, Jun. 2016, pp. 1–4.
- [77] R. Sanchez-Iborra, J. Sanchez-Gomez, J. Ballesta-Viñas, M.-D. Cano, and A. Skarmeta, “ [Performance Evaluation of LoRa Considering Scenario Conditions](#) ”, *Sensors*, vol. 18, no. 3, p. 772, Mar. 3, 2018, 00000.
- [78] S. Naoui, M. E. Elhdhili, and L. A. Saidane, “ [Enhancing the Security of the IoT LoraWAN Architecture](#) ”, in *2016 International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks (PEMWN)*, 00038, Paris, France: IEEE, Nov. 2016, pp. 1–7.
- [79] P. Weber, D. Jackle, D. Rahusen, and A. Sikora, “ [IPv6 over LoRaWANTM](#) ”, in *2016 3rd International Symposium on Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS)*, 00020, Offenburg, Germany: IEEE, Sep. 2016, pp. 75–79.
- [80] J. Haxhibeqiri, A. Karaagac, F. Van den Abeele, W. Joseph, I. Moerman, and J. Hoebeke, “ [LoRa Indoor Coverage and Performance in an Industrial Environment: Case Study](#) ”, in *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 00038, Limassol: IEEE, Sep. 2017, pp. 1–8.

- [81] B. Kim and K.-i. Hwang, “ Cooperative Downlink Listening for Low-Power Long-Range Wide-Area Network ”, *Sustainability*, vol. 9, no. 4, p. 627, Apr. 17, 2017, 00007.
- [82] M. S. Mahmoud and A. A. H. Mohamad, “ A Study of Efficient Power Consumption Wireless Communication Techniques/ Modules for Internet of Things (IoT) Applications ”, *Advances in Internet of Things*, vol. 06, no. 02, pp. 19–29, 2016, 00067.
- [83] M. Magno, F. A. Aoudia, M. Gautier, O. Berder, and L. Benini, “ WULoRa: An Energy Efficient IoT End-Node for Energy Harvesting and Heterogeneous Communication ”, in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2017, 00029, Lausanne, Switzerland: IEEE, Mar. 2017, pp. 1528–1533.
- [84] A. Dongare, C. Hesling, K. Bhatia, A. Balanuta, R. L. Pereira, B. Iannucci, and A. Rowe, “ OpenChirp: A Low-Power Wide-Area Networking Architecture ”, in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 00023, Kona, HI: IEEE, Mar. 2017, pp. 569–574.
- [85] G. Conus, G. Lilis, N. A. Zanjani, and M. Kayal, “ 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)*, 00002, Krakow, Poland: IEEE, Jun. 2016, pp. 1–7.
- [86] S. Aguilar, R. Vidal, and C. Gomez, “ Opportunistic Sensor Data Collection with Bluetooth Low Energy ”, *Sensors*, vol. 17, no. 12, p. 159, Jan. 23, 2017, 00033.
- [87] D. Sartori and D. Brunelli, “ A Smart Sensor for Precision Agriculture Powered by Microbial Fuel Cells ”, in *2016 IEEE Sensors Applications Symposium (SAS)*, 00021, Catania, Italy: IEEE, Apr. 2016, pp. 1–6.
- [88] J. Toussaint, N. El Rachid, and A. Guitton, “ Performance Analysis of the On-the-Air Activation in LoRaWAN ”, in *2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, 00000, Vancouver, BC, Canada: IEEE, Oct. 2016, pp. 1–7.
- [89] P. A. Barro, “ A LoRaWAN Coverage testBed and a Multi-Optional Communication Architecture for Smart City Feasibility in Developing Countries ”, p. 12, 00000.
- [90] P. A. Barro, M. Zennaro, and E. Pietrosemoli, “ TLTN The Local Things Network: On the Design of a LoRaWAN Gateway with Autonomous Servers for Disconnected Communities ”, in *2019 Wireless Days (WD)*, 00000, Manchester, United Kingdom: IEEE, Apr. 2019, pp. 1–4.
- [91] P. Barro, M. Zennaro, J. Degila, and E. Pietrosemoli, “ A Smart Cities LoRaWAN Network Based on Autonomous Base Stations (BS) for Some Countries with Limited Internet Access ”, *Future Internet*, vol. 11, no. 4, p. 93, Apr. 8, 2019, 00001.
- [92] P. Barro, J. Degila, M. Zennaro, and S. Wamba, “ Towards Smart and Sustainable Future Cities Based on Internet of Things for Developing Countries: What Approach for Africa? ”, *EAI Endorsed Transactions on Internet of Things*, vol. 4, no. 13, p. 155481, Sep. 11, 2018, 00003.
- [93] B. Baszczyszyn and P. Mühlenthaler, “ 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*, 00000, Palma, Spain: ACM Press, 2019, pp. 119–126.
- [94] T. Petric, M. Goessens, L. Nuaymi, L. Toutain, and A. Pelov, “ Measurements, Performance and Analysis of LoRa FABIAN, a Real-World Implementation of LPWAN ”, in *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 00000, Valencia, Spain: IEEE, Sep. 2016, pp. 1–7.
- [95] B. Dix-Matthews, R. Cardell-Oliver, and C. 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*, 00000, Shenzhen, China: ACM Press, 2018, pp. 37–42.
- [96] M. Zimmerling, F. Ferrari, L. Mottola, T. Voigt, and L. Thiele, “ pTunes: Runtime Parameter Adaptation for Low-Power MAC Protocols ”, in *Proceedings of the 11th International Conference on Information Processing in Sensor Networks - IPSN ’12*, 00121, Beijing, China: ACM Press, 2012, p. 173.
- [97] R. Cardell-Oliver, A. Willig, C. Huebner, T. Buehring, and A. Monsalve, “ Error Control Strategies for Transmit-Only Sensor Networks: A Case Study ”, in *2012 18th IEEE International Conference on Networks (ICON)*, 00004, Singapore, Singapore: IEEE, Dec. 2012, pp. 453–458.
- [98] 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*, 00027, Pittsburgh, PA, USA: ACM Press, 2017, pp. 97–108.
- [99] M. O. Farooq and D. Pesch, “ A Search into a Suitable Channel Access Control Protocol for LoRa-Based Networks ”, in *2018 IEEE 43rd Conference on Local Computer Networks (LCN)*, 00000, Chicago, IL, USA: IEEE, Oct. 2018, pp. 283–286.
- [100] A. Gupta and M. Fujinami, “ Battery Optimal Configuration of Transmission Settings in LoRa Moving Nodes ”, in *2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, 00000, Las Vegas, NV, USA: IEEE, Jan. 2019, pp. 1–6.
- [101] V. Hauser and T. 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)*, 00000, Prague: IEEE, Aug. 2017, pp. 85–90.
- [102] A. Hoeller, R. D. Souza, O. L. Alcaraz Lopez, H. Alves, M. de Noronha Neto, and G. Brante, “ Exploiting Time Diversity of LoRa Networks Through Optimum Message Replication ”, in *2018 15th International Symposium on Wireless Communication Systems (ISWCS)*, 00000, Lisbon: IEEE, Aug. 2018, pp. 1–5.
- [103] S.-Y. Wang, Y.-R. Chen, T.-Y. Chen, C.-H. Chang, Y.-H. Cheng, C.-C. Hsu, and Y.-B. Lin, “ Performance of LoRa-Based IoT Applications on Campus ”, in *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, 00013, Toronto, ON: IEEE, Sep. 2017, pp. 1–6.
- [104] L. Angrisani, P. Arpaia, F. Bonavolonta, M. Conti, and A. Liccardo, “ LoRa Protocol Performance Assessment in Critical Noise Conditions ”, in *2017 IEEE 3rd International Forum on Research and Technologies for Society and Industry (RTSI)*, 00020, Modena, Italy: IEEE, Sep. 2017, pp. 1–5.
- [105] P. Jorke, S. Bocker, F. Liedmann, and C. Wietfeld, “ 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)*, 00023, Montreal, QC: IEEE, Oct. 2017, pp. 1–6.
- [106] J. M. Marais, R. Malekian, and A. M. Abu-Mahfouz, “ LoRa and LoRaWAN Testbeds: A Review ”, in *2017 IEEE AFRICON*, 00000, Cape Town: IEEE, Sep. 2017, pp. 1496–1501.
- [107] B. Reynders and S. Pollin, “ Chirp Spread Spectrum as a Modulation Technique for Long Range Communication ”, in *2016 Symposium on Communications and Vehicular*

- Technologies (SCVT)*, 00000, Mons, Belgium: IEEE, Nov. 2016, pp. 1–5.
- [108] M. Slabicki, G. Premsankar, and M. Di Francesco, “ Adaptive Configuration of Lora Networks for Dense IoT Deployments ”, in *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, 00000, Taipei, Taiwan: IEEE, Apr. 2018, pp. 1–9.
- [109] D.-Y. Kim, S. Kim, H. Hassan, and J. H. Park, “ Adaptive Data Rate Control in Low Power Wide Area Networks for Long Range IoT Services ”, *Journal of Computational Science*, vol. 22, pp. 171–178, Sep. 2017, 00014.
- [110] D. Zorbas, G. Z. Papadopoulos, P. Maille, N. Montavont, and C. Douligeris, “ Improving LoRa Network Capacity Using Multiple Spreading Factor Configurations ”, in *2018 25th International Conference on Telecommunications (ICT)*, 00000, St. Malo: IEEE, Jun. 2018, pp. 516–520.
- [111] U. Raza, P. Kulkarni, and M. Sooriyabandara, “ Low Power Wide Area Networks: An Overview ”, *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 855–873, 22–2017, 00000.
- [112] D. Bankov, E. Khorov, and A. 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)*, 00026, Montreal, QC: IEEE, Oct. 2017, pp. 1–5.
- [113] R. Oliveira, L. Guardalben, and S. Sargent, “ Long Range Communications in Urban and Rural Environments ”, in *2017 IEEE Symposium on Computers and Communications (ISCC)*, 00000, Heraklion, Greece: IEEE, Jul. 2017, pp. 810–817.
- [114] A. Hoeller, R. D. Souza, O. L. Alcaraz Lopez, H. Alves, M. de Noronha Neto, and G. Brante, “ Analysis and Performance Optimization of LoRa Networks With Time and Antenna Diversity ”, *IEEE Access*, vol. 6, pp. 32 820–32 829, 2018, 00000.
- [115] P. J. Radcliffe, K. G. Chavez, P. Beckett, J. Spangaro, and C. Jakob, “ Usability of LoRaWAN Technology in a Central Business District ”, in *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*, 00015, Sydney, NSW: IEEE, Jun. 2017, pp. 1–5.
- [116] M. Rizzi, P. Ferrari, A. Flammini, and E. Sisinni, “ Evaluation of the IoT LoRaWAN Solution for Distributed Measurement Applications ”, *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 12, pp. 3340–3349, Dec. 2017, 00064.
- [117] Z. Qin and J. A. McCann, “ Resource Efficiency in Low-Power Wide-Area Networks for IoT Applications ”, in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, 00020, Singapore: IEEE, Dec. 2017, pp. 1–7.
- [118] Y. Mo, M.-T. Do, C. Goursaud, and J.-M. Gorce, “ Optimization of the Predefined Number of Replications in a Ultra Narrow Band Based IoT Network ”, in *2016 Wireless Days (WD)*, 00015, Toulouse, France: IEEE, Mar. 2016, pp. 1–6.
- [119] Q. Song, X. Lagrange, and L. Nuaymi, “ Evaluation of Macro Diversity Gain in Long Range ALOHA Networks ”, *IEEE Communications Letters*, vol. 21, no. 11, pp. 2472–2475, Nov. 2017, 00010.
- [120] D.-H. Kim, E.-K. Lee, and J. Kim, “ Experiencing LoRa Network Establishment on a Smart Energy Campus Testbed ”, *Sustainability*, vol. 11, no. 7, p. 1917, Mar. 30, 2019, 00000.
- [121] U. Noreen, A. Bounceur, and L. 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)*, 00052, Fez, Morocco: IEEE, May 2017, pp. 1–6.
- [122] A. Lavric and V. Popa, “ Performance Evaluation of LoRaWAN Communication Scalability in Large-Scale Wireless Sensor Networks ”, *Wireless Communications and Mobile Computing*, vol. 2018, pp. 1–9, Jun. 28, 2018, 00000.
- [123] C.-H. Liao, G. Zhu, D. Kuwabara, M. Suzuki, and H. Morikawa, “ Multi-Hop LoRa Networks Enabled by Concurrent Transmission ”, *IEEE Access*, vol. 5, pp. 21 430–21 446, 2017, 00037.
- [124] Z. Li, S. Zozor, J.-M. Brossier, N. Varsier, and Q. Lampin, “ 2D Time-Frequency Interference Modelling Using Stochastic Geometry for Performance Evaluation in Low-Power Wide-Area Networks ”, Nov. 12, 2016, 00000. arXiv: 1606.04791 [cs].
- [125] C. Goursaud and Y. Mo, “ Random Unslotted Time-Frequency ALOHA: Theory and Application to IoT UNB Networks ”, in *2016 23rd International Conference on Telecommunications (ICT)*, 00029, Thessaloniki, Greece: IEEE, May 2016, pp. 1–5.
- [126] D. Magrin, M. Capuzzo, and A. Zanella, “ A Thorough Study of LoRaWAN Performance Under Different Parameter Settings ”, Jun. 12, 2019, 00000. arXiv: 1906.05083 [cs].
- [127] G. Ferre, “ Collision and Packet Loss Analysis in a LoRaWAN Network ”, in *2017 25th European Signal Processing Conference (EUSIPCO)*, 00000, Kos, Greece: IEEE, Aug. 2017, pp. 2586–2590.
- [128] R. B. Sørensen, D. M. Kim, J. J. Nielsen, and P. Popovski, “ Analysis of Latency and MAC-Layer Performance for Class A LoRaWAN ”, *IEEE Wireless Communications Letters*, vol. 6, no. 5, pp. 566–569, Oct. 2017, 00000. arXiv: 1712.05171.
- [129] M. Capuzzo, D. Magrin, and A. Zanella, “ Mathematical Modeling of LoRa WAN Performance with Bi-Directional Traffic ”, in *2018 IEEE Global Communications Conference (GLOBECOM)*, 00000, Abu Dhabi, United Arab Emirates: IEEE, Dec. 2018, pp. 206–212.
- [130] N. Kouvelas, V. Rao, and R. R. V. Prasad, “ Employing P-CSMA on a LoRa Network Simulator ”, May 30, 2018, 00000. arXiv: 1805.12263 [cs].
- [131] D. Zucchetto and A. Zanella, “ Uncoordinated Access Schemes for the IoT: Approaches, Regulations, and Performance ”, *IEEE Communications Magazine*, vol. 55, pp. 48–54, 2017, 00007.
- [132] J. M. Marais, R. Malekian, and A. M. Abu-Mahfouz, “ Evaluating the LoRaWAN Protocol Using a Permanent Outdoor Testbed ”, *IEEE Sensors Journal*, vol. 19, no. 12, pp. 4726–4733, Jun. 15, 2019, 00000.
- [133] S. Kim and Y. Yoo, “ Contention-Aware Adaptive Data Rate for Throughput Optimization in LoRaWAN ”, *Sensors*, vol. 18, no. 6, p. 1716, May 25, 2018, 00003.
- [134] R. M. Sandoval, A.-J. Garcia-Sanchez, J. Garcia-Haro, and T. M. Chen, “ Optimal Policy Derivation for Transmission Duty-Cycle Constrained LPWAN ”, *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 3114–3125, Aug. 2018, 00000.
- [135] G. Yang and H. Liang, “ A Smart Wireless Paging Sensor Network for Elderly Care Application Using LoRaWAN ”, *IEEE Sensors Journal*, vol. 18, no. 22, pp. 9441–9448, Nov. 15, 2018, 00000.
- [136] R. Bonnefoi, C. Moy, and J. Palicot, “ Improvement of the LPWAN AMI Backhaul’s Latency Thanks to Reinforcement Learning Algorithms ”, *EURASIP Journal on Wireless Communications and Networking*, vol. 2018, no. 1, p. 34, Dec. 2018, 00005.
- [137] J. Kim and J. Song, “ A Secure Device-to-Device Link Establishment Scheme for LoRaWAN ”, *IEEE Sensors Journal*, vol. 18, no. 5, pp. 2153–2160, Mar. 1, 2018, 00015.

- [138] F. Al-Turjman, A. Radwan, S. Mumtaz, and J. Rodriguez, “ Mobile Traffic Modelling for Wireless Multimedia Sensor Networks in IoT ”, *Computer Communications*, vol. 112, pp. 109–115, Nov. 2017, 00008.
- [139] C. Tunc and N. Akar, “ Markov Fluid Queue Model of an Energy Harvesting IoT Device with Adaptive Sensing ”, *Performance Evaluation*, vol. 111, pp. 1–16, May 2017, 00021.
- [140] L. Servi and S. Finn, “ M/M/1 Queues with Working Vacations (M/M/1/WV) ”, *Performance Evaluation*, vol. 50, no. 1, pp. 41–52, Oct. 2002, 00469.
- [141] K. Li, J. Wang, Y. Ren, and J. Chang, “ Equilibrium Joining Strategies in M/M/1 Queues with Working Vacation and Vacation Interruptions ”, *RAIRO - Operations Research*, vol. 50, no. 3, pp. 451–471, Jul. 2016, 00004.
- [142] W. M. Kempa and M. Kobielnik, “ Transient Solution for the Queue-Size Distribution in a Finite-Buffer Model with General Independent Input Stream and Single Working Vacation Policy ”, *Applied Mathematical Modelling*, vol. 59, pp. 614–628, Jul. 2018, 00003.
- [143] J. Navarro-Ortiz, S. Sendra, P. Ameigeiras, and J. M. Lopez-Soler, “ Integration of LoRaWAN and 4G/5G for the Industrial Internet of Things ”, *IEEE Communications Magazine*, vol. 56, no. 2, pp. 60–67, Feb. 2018, 00037.
- [144] L. Alonso, J. Barbarán, J. Chen, M. Díaz, L. Llopis, and B. Rubio, “ Middleware and Communication Technologies for Structural Health Monitoring of Critical Infrastructures: A Survey ”, *Computer Standards & Interfaces*, vol. 56, pp. 83–100, Feb. 2018, 00019.
- [145] Y. Song, J. Lin, M. Tang, and S. Dong, “ An Internet of Energy Things Based on Wireless LPWAN ”, *Engineering*, vol. 3, no. 4, pp. 460–466, Aug. 2017, 00043.
- [146] Q. Zhou, J. Xing, L. Hou, R. Xu, and K. Zheng, “ A Novel Rate and Channel Control Scheme Based on Data Extraction Rate for LoRa Networks ”, Feb. 12, 2019, 00000. arXiv: 1902.04383 [eess].
- [147] G. Zhu, C.-H. Liao, T. Sakdejayont, I.-W. Lai, Y. Narusue, and H. Morikawa, “ Improving the Capacity of a Mesh LoRa Network by Spreading-Factor-Based Network Clustering ”, *IEEE Access*, vol. 7, pp. 21 584–21 596, 2019, 00000.
- [148] A. Mahmood, E. Sisinni, L. Guntupalli, R. Rondon, S. A. Hassan, and M. Gidlund, “ Scalability Analysis of a LoRa Network Under Imperfect Orthogonality ”, *IEEE Transactions on Industrial Informatics*, vol. 15, no. 3, pp. 1425–1436, Mar. 2019, 00025.
- [149] K.-H. Ke, Q.-W. Liang, G.-J. Zeng, J.-H. Lin, and H.-C. Lee, “ A LoRa Wireless Mesh Networking Module for Campus-Scale Monitoring: Demo Abstract ”, in *Proceedings of the 16th ACM/IEEE International Conference on Information Processing in Sensor Networks - IPSN '17*, 00011, Pittsburgh, Pennsylvania: ACM Press, 2017, pp. 259–260.
- [150] Y. Wu, J. A. Stankovic, T. He, and S. Lin, “ Realistic and Efficient Multi-Channel Communications in Wireless Sensor Networks ”, in *IEEE INFOCOM 2008 - The 27th Conference on Computer Communications*, 00458, Phoenix, AZ, USA: IEEE, Apr. 2008, pp. 1193–1201.
- [151] S.-Y. Liew, C.-K. Tan, M.-L. Gan, and H. G. Goh, “ A Fast, Adaptive, and Energy-Efficient Data Collection Protocol in Multi-Channel-Multi-Path Wireless Sensor Networks ”, *IEEE Computational Intelligence Magazine*, vol. 13, no. 1, pp. 30–40, Feb. 2018, 00007.
- [152] J. Bregell, “ Hardware and Software Platform for Internet of Things ”, *Master of Science Thesis in Embedded Electronic System Design*, 2015, 00002.
- [153] D. E. Kouicem, A. Bouabdallah, and H. Lakhlef, “ Internet of Things Security: A Top-down Survey ”, *Computer Networks*, vol. 141, pp. 199–221, Aug. 4, 2018, 00029.
- [154] V. P. Venkatesan, C. P. Devi, and M. Sivarajanji, “ 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)*, 00004, Palladam, Tamilnadu, India: IEEE, Feb. 2017, pp. 22–31.
- [155] E. Alba, “ Intelligent Systems for Smart Cities ”, in *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion - GECCO '16 Companion*, 00004, Denver, Colorado, USA: ACM Press, 2016, pp. 823–839.
- [156] H. A. A. Al-Kashoash and A. H. Kemp, “ Comparison of 6LoWPAN and LPWAN for the Internet of Things ”, *Australian Journal of Electrical and Electronics Engineering*, vol. 13, no. 4, pp. 268–274, Oct. 2016, 00000.
- [157] W. Ayoub, A. E. Samhat, F. Nouvel, M. Mroue, and J.-C. Prevotet, “ Internet of Mobile Things: Overview of LoRaWAN, DASH7, and NB-IoT in LPWANs Standards and Supported Mobility ”, *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1561–1581, 22–2019, 00000.
- [158] O. Berder, “ Réseaux & Communications Sans fil ”, p. 593, 2014, 00000.
- [159] U. Sarwar, “ IoT Architecture : Elements of Connectivity Technologies ”, p. 23, 2015, 00000.
- [160] *LoRaWAN® for Developers | LoRa Alliance™*, [Online; accessed 10. Sep. 2019].
- [161] *All About LoRa and LoRaWAN*, [Online; accessed 10. Sep. 2019], Aug. 2019.
- [162] Z. Qin, G. Denker, C. Giannelli, P. Bellavista, and N. Venkatasubramanian, “ A Software Defined Networking Architecture for the Internet-of-Things ”, in *2014 IEEE Network Operations and Management Symposium (NOMS)*, 00000, Krakow, Poland: IEEE, May 2014, pp. 1–9.

Genetic Algorithm For LoRa

Aghiles Djoudi¹², Rafik Zitouni², Nawel Zangar¹ and Laurent George¹

¹LIGM, UMR 8049, École des Ponts, UPEM, ESIEE Paris, CNRS, UPE, France

²ECE Research Lab Paris, 37 Quai de Grenelle, 75015 Paris, France

Email: {aghiles.djoudi, nawel.zangar, laurent.george}@esiee.fr, rafik.zitouni@ece.fr

Abstract—The exponential growth of Internet of things (*IoT*) applications in both industry and academic research raises many questions in wireless sensor networks. Heterogeneous networks of IoT devices strongly depend on the ability of IoT devices to adapt their data transmission parameters to each application requirement. One of the most important problem of the emerging IoT networks is the limitation in terms of energy consumption and computation capability. These limitations could be addressed by using the edge computing to unload IoT devices from additional computation tasks. Our work is motivated by the idea of matching each transmission configuration with a reward and cost values to satisfy applications constraints. Our goal is to make IoT devices able to select the optimal configuration and send their data to the gateway with the QoS required by IoT applications. In this work, we use LoRa network to evaluate the efficiency of our algorithm. Determining the best configuration among 6720 LaRa transmission settings is challenging. The difficulty is mainly due to the lack of tools that could take all applications requirements into account to select the best settings. To address this problem, we use a genetic algorithm in an edge computing to select the transmission parameters needed by the application. Each LoRa configuration represents a feature that needs to be selected to match better the QoS criteria. Particularly, we analyze the impact of selecting one configuration in 3 kinds of applications: text, voice and image transmission by modeling a new adaptive data rate selection process.

Index Terms—Genetic algorithm; Fuzzy logic; LoRaWAN; Adaptive Data Rate (ADR).

I. INTRODUCTION

The need of Low Power Wide Area Networks (*LPWAN*) increased significantly these five last years. The main factor is that IoT devices require low power consumption to transmit data in a wide area. LoRa, Sigfox and Narrowband IoT (*NB – IoT*) are the most known technologies that satisfy these requirements. Applications like smart building and smart environment are one of hundreds use cases that need to be deployed with such technologies. Unlike Sigfox and NB-IoT, LoRa is more open for academic research because the specification that governs it is relatively open. The transmission could be configured with 4 parameters: *SF*, Transmission Power (P^{tx}), *CR* and *BW*, to achieve better performance.

The main LPWAN research directions are about link optimization, adaptability and large scale networks to support massive number of devices. The selection of an appropriate transmission parameter for IoT networks typically depends on the nature of the application. In this paper, we investigate

the performance of heterogeneous networks (i.e., when each IoT device selects its LoRa transmission parameters according to its link budget and the application requirements). For that purpose, we have developed a LoRa transmission adaptation mechanism. Both ns-3 simulator and the Low cost LoRa Gateway [1] are used to validate our approach. The computation tasks of the selection process will run on the Gateway device (Raspberry-pi) and the required settings will be sent to nodes for the next transmission.

This paper is organized as follows. Section II elucidates summary of related works. In Section III, we propose our approach to solve LoRa parameter selection problem. Our experiments is presented in Section IV. Section V concludes this paper.

II. RELATED WORK

Transmission parameter configuration mechanisms, such as *ADR* scheme [2] need to be developed to fit each application requirement in terms of power consumption, delay and packet delivery ratio. Solutions running on LoRa node should be less complex to match computation limitation of *IoT* devices as required in LoRaWAN specification. However, LoRa network server could run complex management mechanism, which can be developed to improve network performance. In this paper, we focus on the server-side mechanisms.

The basic *ADR* scheme [2] provided by LoRaWAN predicts channel conditions using the maximum received Signal Noise Rate (*SNR*) in the last 20 packets. The basic *ADR* scheme is sufficient when the variance of the channel is low, it reduces the interference compared with the static data rate [3][4]. However, their simplicity causes many potential drawbacks. First, the diversity of LoRa Gateway models that measures *SNR* make the measurement inaccurate as a result of hardware calibration and interfering transmissions. Second, selecting the maximum *SNR* each 20 packets received could be a very long period in many IoT applications that require less uplink transmission. Third, transmission parameters adjustment considers only the link of a single node. If many LoRa nodes are connected to the near gateway, all nodes connected to this one will use the fastest data rate. In this case, the number of LoRa nodes using the same data rate will increase and the probability of collisions also increases dramatically.

For example, the authors in [4] slightly modify the basic *ADR* scheme by replacing the maximum *SNR* with the average function. In this paper, we focus on building a framework that help IoT devices to adapt their transmission parameters to the application requirements in a server side.

III. PROPOSED FRAMEWORK

Data: population: a set of individuals, FITNESS-FN: a function that measures the fitness of an individual

Result: the best individual in population , according to FITNESS-FN

1 GA(population, FITNESS-FN):

2 **repeat**

3 new population empty set **for** $i = 1$ to S **IZE** (population) **do**

4 x RANDOM-SELECTION (population, FITNESS-FN) y

5 RANDOM-SELECTION (population, FITNESS-FN) child REPRODUCE (x,y)

6 **if** (small random probability) **then**

7 | child MUTATE (child)

8 **else**

9 | a

10 **end**

11 dd child to new population

12 **end**

13 population new population

14 **until** until some individual is fit enough, or enough time has elapsed;

15 **end**

Data: x,y,parent individuals

Result: an individual

14 REPRODUCE(x,y):

15 n LENGTH (x); c random number from 1 to n

16 individual APPEND (SUBSTRING (x , 1, c), SUBSTRING (y, c + 1, n))

17 **end**

The selection process scheme illustrated in Figure 1 can be described following these five steps:

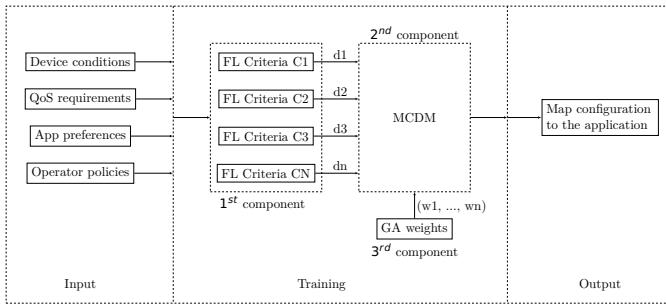


Figure 1. The proposed scheme for LoRa transmission parameters selection based on *GA*, *FL* and Multi-Criteria Decision Making *MCDM*.

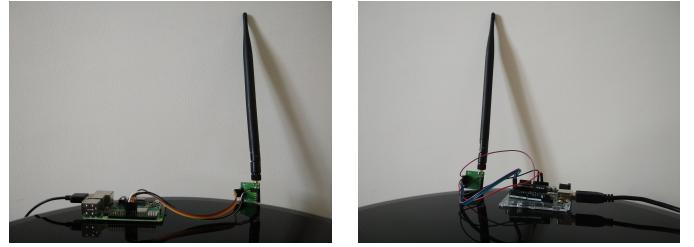


Figure 2. Gateway & Sensor node.

- 1) According to the Semtech SX1276 LoRa transceiver [5], there are 6720 possible settings (s_1, \dots, s_{6720}) and the framework has to select the most optimal one or to rank them according to their relevance.
- 2) The first step of the selection process depends on multiple criteria up to i (c_1, \dots, c_i). Different type of criteria can be measured from different sources to cover the maximum point of views, as an example, the network server requirements, the applications requirements and the devices conditions.
- 3) The Fuzzy Logic (FL) based subsystem gives an initial score for each configuration that reflects its relevance. The different sets of scores (d_1, \dots, d_i) are sent to the *MCDM* in the 5th step.
- 4) At the same time, the *GA* [6] assigns a suitable weight (w_1, \dots, w_i) for each initial selection decision, this selection is made according to the objective function that is required by the application.
- 5) Using the initial scores coming from the 3rd step and the weights using the 4th step, the multi criteria decision making *MCDM* will select the most relevant settings and rank them according to their reward.

IV. EXPERIMENTS

For our experiments we use both real environment (Figure 2) and ns-3 simulator with SX1276 LoRa module. However, to test the scalability of genetic algorithm with numerous IoT devices in a real environment, we use FIT IoT-LAB platform among other platforms presented in [7]. This choice is motivated by the number of devices supported by this platform (up to 2000 nodes).

Figure 2a presents the LoRa gateway that we build using a low cost LoRa gateway [1] on a Raspberry-pi. Figure 2b presents one of the two Arduino boards equipped with an antenna that cover both 868 and 433 MHz band with a SX1276 LoRa Transceiver.

Due to the energy constraints of LoRa nodes of class A that we use, our framework will send commands through the FCtrl fields to ask nodes to adapt their transmission behavior to the new application or the new environment conditions.

V. DISCUSSION

Our main contribution was to build 3 applications that requires 3 different levels of QoS, such as text, sound and

image transmission. We used a low cost LoRa gateway on a Raspberry-pi with 2 Arduino boards equipped with 2 LoRa Transceivers based on the Semtech SX1276 specification. The main challenge **addressed in this work** was to explore the application of genetic algorithm in LoRa transmission parameter selection. To measure the accuracy of applying genetic algorithm in an edge computing we expect to compare our approach with other adaptive data rate solutions.

REFERENCES

- [1] Pham. (o). Low Cost LoRa Gateway, [Online]. Available: <https://github.com/CongducPham/LowCostLoRaGw> (visited on 08/30/2019).
- [2] L. Alliance. (o). LoraWAN Specification, [Online]. Available: <https://lora-alliance.org/resource-hub/lorawanr-specification-v103> (visited on 08/30/2019).
- [3] M. C. Bor, U. Roedig, T. Voigt, and J. M. Alonso, “ 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*, 00240, Malta, Malta: ACM Press, 2016, pp. 59–67.
- [4] M. Slabicki, G. Premankar, and M. Di Francesco, “ Adaptive Configuration of Lora Networks for Dense IoT Deployments ”, in *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, 00000, Taipei, Taiwan: IEEE, Apr. 2018, pp. 1–9.
- [5] Semtech. (o). Semtech LoRa Technology Overview, [Online]. Available: <https://www.semtech.com/lora> (visited on 09/01/2019).
- [6] M. Alkhawlani and A. Ayesh, “ Access Network Selection Based on Fuzzy Logic and Genetic Algorithms ”, *Advances in Artificial Intelligence*, vol. 2008, pp. 1–12, 2008, 00089.
- [7] A.-S. Tonneau, N. Mitton, and J. Vandaele, “ How to Choose an Experimentation Platform for Wireless Sensor Networks? A Survey on Static and Mobile Wireless Sensor Network Experimentation Facilities ”, *Ad Hoc Networks*, vol. 30, pp. 115–127, Jul. 2015, 00046.

EES

Aghiles Djoudi^{1,2}, Rafik Zitouni², Nawel Zangar¹ and Laurent George¹

¹LIGM, UMR 8049, École des Ponts, UPEM, ESIEE Paris, CNRS, UPE, France

²ECE Research Lab Paris, 37 Quai de Grenelle, 75015 Paris, France

Email: {aghiles.djoudi, nawel.zangar, laurent.george}@esiee.fr, rafik.zitouni@ece.fr

Abstract—The number of building sites increased considerably these few last years. Emerging countries invest more and more in building and construction sites to overcome the need of citizens like education, residential and administration buildings. Such sites should be equipped with evacuation systems to allow bricklayers and other persons to be evacuated quickly. Furthermore, evacuation systems should communicate with each other to coordinate evacuation emergency of people. One of the most important problem of such systems is the need of a powerful network technology with sufficient range that could make communication in sites of several kilometers squares. The lack of such mechanisms make the life of billion citizens in a danger. This problem could be addressed by building a novel evacuation emergency application in *IoT* devices with *LPWAN*. One of the known *LPWAN* networks is Long Range (*LoRa*) network. Our work is motivated by the deployment of an Emergency Evacuation Systems (*EES*) in building sites. Our goal is to measure the Quality of Service (*QoS*) of IEEE.802.15.4 and LoRa technologies and adapt their configurations to the emergency situation of the site.

I. INTRODUCTION

Modern and even old cities have several issues regarding building management, urban mobility, disaster recovery, which can be supported by *IoT* applications. Actually, different kinds of services appear to help city management by automating tasks and predicting critical situations before they occur. *EES* for example could help to locate survivors and plans for rescue missions in case of disasters.

EES require high quality level of communication to ensure their proper operations in critical situations. This constraint is more complex when devices need to communicate in a large area like building sites. Thus, the need for a powerful network technology that could transmit data with a long range and sufficient *DR* is very challenging.

First the architecture that we propose are based on three component, *IoT* devices, smart gateway and edge computing server. *IoT* applications, especially emergency ones in case of disasters, require data to be received and bounded in time intervals. So, service orchestrations is an important rigid constraint. Hence, the orchestration algorithms should adapt the network configuration to the emergency situation of the site. Furthermore, the delay to adapt network behavior to the emergency situation has to be optimized to react quickly when an alarm is triggered. For this reason, we opt for the use of

an edge computing and a smart gateway which could map transmission tuning to the emergency state.

This paper is organized as follows. Section II elucidates summary of related works. Section III highlights the required network technologies that could be used in *EES*. In Section IV, we propose our architecture scheme for *EES*. Section V concludes this paper.

II. RELATED WORK

Research communities are paying more and more attention for *EES*, this particular attention is mainly due to the interest of building companies to deploy smart sensors in their construction sites. The main factors of a good deployment of such systems is about the quality of the network that makes all emergency agents communicate and collaborate to make good decisions. In another hand, the need of population of new tools to deal with the permanently dynamic changes of pedestrian and car paths in their city, becomes very challenging. For example, Qiu, et al [1] proposed an emergency response system with sensor nodes for real time application they proposed an iterative method for delay optimizing with the lossless transmission, authors tried to find a trade-off between the increased energy consumption and the reduced network lifetime. The framework proposed in by Al Turjman, et al [2] is a multi hop routing method for disaster management in real time. To achieve a higher efficiency, the system, they propose, is designed for using limited multiple hops with the help of left over energy may sometimes result with loss of data due the exhausted energy level.

III. BACKGROUND

To select the wireless communication that best fit smart building application requirements, four main parameters are generally used: (i) cost; (ii) data rate; (iii) autonomy and (iv) communication range **lopes_design_2019**. As data rate and communication range are inversely proportional to autonomy, devices need to consume more energy to communicate at higher bit rates and greater distances. *IoT* devices in emerging evaluation system are particularly vulnerable to power consumption. This vulnerability is mainly due to the awareness aspect of their task and the need to be powered all the time without carrying about battery level for a long period of time.

| Characteristics | CF [Hz] | 6LoWPAN | LoRaWAN |
|-----------------|-----------|-----------|------------------------------|
| Modulation | 2.4G | O-QPSK | - |
| | 915M | BPSK | LoRa |
| | 868M | BPSK | LoRa/GFSK |
| Channels | 2.4G | 16 | - |
| | 915M | 10 | 64+8↑, 8↓ |
| | 868M | 1 | 10 |
| CF [MHz] | 2.4G | X | - |
| | 915M | 902-929 | 902-928 |
| | 868M | 868-868.6 | 863-870 and 780 |
| BW [Hz] | 2.4G | 5M | - |
| | 915M | 2M | 125K-500K |
| | 868M | 600M | 125K-250K |
| DR [bps] | 2.4G | 250M | - |
| | 915M | 40M | 980-21.9K |
| | 868M | 20M | LoRa: 0.3K-37.5K FSK: 50K |
| CR [dBm] | 2.4G | -85 | - |
| | 915M | -92 | X |
| | 868M | -92 | -137 |

Table I. LoRa and 6LowPAN characteristics [3].

Table 1 compiles the two more promising standards that are being used in smart building applications, notably LoRa and 6LowPAN.

CF DR CR BW

raza_low_22 lopes_design_2019

IV. ARCHITECTURE

Research in *IoT* field usually pay attention to energy consumption by optimizing the data transmission rate or by aggregating the data. However, in an urgent situation like a disaster monitoring service, data accuracy becomes very important. Furthermore, these data are required to achieve the network server quickly over Constrained Application Protocol (*CoAP*) or Message Queuing Telemetry Transport (*MQTT*) as shown in Figure ?? to make the appropriate decisions to evacuate people. For this reason, we use an edge computing server that receives data from *IoT* devices by means of IEEE.802.15.4 and LoRa technologies. This requires to make the LoRa gateway able to forward received packets to the IEEE802.15.4 border router and vice versa.



Figure 1. IoT architecture.

V. CONCLUSION

LPWAN, Wireless Sensor Network (*WSN*) and *IoT* architecture are the first candidates to enhanced disaster monitoring and management systems. Particularly, IEEE802.15.4 and LoRa networks gives new insight for effective *EES*. This work gives an overview of deployment of *IoT* architecture for *EES*, Such services and demand for edge computing in real-time poses new architectural and service orchestration challenges. As a future work, we plan to study the efficiency of using artificial intelligence/machine learning to adapt these two networks to the emergency situation of the building.

REFERENCES

- [1] T. Qiu, Y. Lv, F. Xia, N. Chen, J. Wan, and A. Tolba, “ *ERGID: An Efficient Routing Protocol for Emergency Response Internet of Things* ”, *Journal of Network and Computer Applications*, vol. 72, pp. 104–112, Sep. 2016, 00052.
- [2] F. Al-Turjman, “ *Cognitive Routing Protocol for Disaster-Inspired Internet of Things* ”, *Future Generation Computer Systems*, vol. 92, pp. 1103–1115, Mar. 2019, 00050.
- [3] H. A. A. Al-Kashoash and A. H. Kemp, “ *Comparison of 6LoWPAN and LPWAN for the Internet of Things* ”, *Australian Journal of Electrical and Electronics Engineering*, vol. 13, no. 4, pp. 268–274, Oct. 2016, 00000.

Q-Learning

Aghiles Djoudi¹², Rafik Zitouni², Nawel Zangar¹ and Laurent George¹

¹LIGM, UMR 8049, École des Ponts, UPEM, ESIEE Paris, CNRS, UPE, France

²ECE Research Lab Paris, 37 Quai de Grenelle, 75015 Paris, France

Email: {aghiles.djoudi, nawel.zangar, laurent.george}@esiee.fr, rafik.zitouni@ece.fr

Abstract—One of the most important problem ... This limitation could be addressed by Our work is motivated by Our goal is to

The difficulty is mainly due to To address this problem, Particularly,
We validate our approach by
Simulation results show that
Furthermore,

I. INTRODUCTION

The need of the main factor is
The difficulty to build such system is
For that purpose, we ... In this work we ...

This paper is organized as follows. Section II elucidates summary of related works. Section ?? provide the required background. In Section IV, we ... our approach to solve ... Our experimentation is presented in Section V. Our findings are presented in section ?? . Section VII concludes this paper.

II. RELATED WORK

III. BACKGROUND

Selection of technology

- A. Hardware
- B. Operating system
- C. Communication protocol
- D. Workspace and tools

IV. APPROACH

$$\min_{W, Z} F(W, Z) = \sum_{l=1}^k \sum_{i=1}^n w_{li}^x d(\mathbf{z}_l, \mathbf{x}_i) \text{ such that } 0 \leq w_{li} \leq 1, \quad 1 \leq l \leq k, \quad 1 \leq i \leq n \quad \sum_{l=1}^k w_{li} = 1, \quad 1 \leq i \leq n \quad 0 < \sum_{i=1}^n w_{li} < n, \quad 1 \leq l \leq k$$

where:

- n is the number of possible configurations x_i with $1 < i < n$.
- k is the number of application clusters c_l with $1 < l < k$.
- d is the number of metrics a_j of each object x_i with $1 < j < d$.
- c is the number of metrics categories (physical, mac) c_m with $1 < m < c$.
- $A = A_1, \dots, A_d$ is a set of categorical attributes with $A_{(1 < j < d)} = a_{j1}, \dots, a_{jc}$
- $D = x_1, \dots, x_n$ is a set of n objects.
- $D = x_1, \dots, x_n$ categorical data set with $x_{(1 < i < n)} \in A_{(1 < j < d)}$

- $Z_l = z_{l1}, \dots, z_{ld}$ is a set of cluster centers.
- $W = w_{li}$ is a $k \times n$ fuzzy membership matrix.
- $\alpha \in [1,]$ is a weighting exponent.
- $d(z_l, x_i)$ is a certain distance measure between cluster center z_l and the object x_i .

A. The FL-based Control Component

Our framework contains four FL-based subsystems. Each subsystem considers one of the *ADR* important criteria.

1) Inputs:

a) *Physical subsystem*: The Physical subsystem considers the received signal strength criteria. It has two input variables RSS1 to describe the received signal strength from LoRa-based network, and RSS2 to describe the received signal strength from GFSK-based network.

b) *Envirement subsystem*: The Envirement subsystem considers the mobile station speed criteria. It has only one input variable MSS to describe the mobile station speed.

c) *Sercice subsystem*: The Sercice subsystem considers the service-type criteria. It has two input variables, the first is DelayReqc to describe the one-way delay needed for the required service, and the second is RateReqc to describe the bit rate needed for the required service.

d) *User subsystem*: The User subsystem considers the user preference and price criteria. It has only one input variable Price to describe the user preferred price.

membership functions: Every input variable has three membership functions

- low
- medium
- high

Figure 3 shows the membership functions of the input variables.

2) Outputs:

Every subsystem has two output variables,
a) *CDMA network*: one variable to describe the probability of acceptance for the new user in the CDMA network

b) *CDMA network*: one variable to describe the probability of acceptance for the new user in the TDMA network.

The subsystems output variables are

- RSS_{c1}, RSS_{c2} for RSS subsystem,
- MSS_{c1}, MSS_{c2} for MSS subsystem,
- ST_{c1}, ST_{c2} for ST subsystem,
- UP_{c1}, UP_{c2} for UP subsystem.

membership functions: Each output variable has four membership functions

- ➡ TR (totally reject)
- ➡ PR (probability reject)
- ➡ PA (probability accept)
- ➡ TA (totally accept)

Figure 4 shows ST c1 variable with its membership functions as a sample for the output variables.

B. The GA Component

The GA has been used in our scheme to help the users or network operator to find suitable values for the weights W_v , W_s , W_t , W_u .

Beside the GA advantages that are summarized on Section 2.4, our decision to use GA was based on the nature of our objective functions that have several dynamic and stochastic components, where any other derivative-based optimization method cannot perform well. Another important issue that encouraged the selection of GA to our problem is the high interaction between different variables.

Since the *ADR* is a multi criteria problem in nature, different objectives need to be optimized.

In this paper, three different objective functions are proposed to cover the different and opposite objectives and requirements of the users, QoS and operators. The proposed objective functions are outlined in the following points.

1) *Objective function 1 (UP):* The first objective is to maximize the percentage of satisfied users, that is, users who are assigned to a network of their preference (P_u): since the algorithm considers multiple criteria, it may assign a user to a network different from his/her original preference. P_u is considered appropriate for performance evaluation from the user point of view. To maximize P_u , the objective function ObjFun1 shown in Algorithm 1 is used.

2) *Objective function 2 (RSS):* The second objective is to maximize the percentage of users assigned to the networks with stronger signal strength (P_q): since the algorithm considers multiple criteria, it may assign a user to a network with weaker signal strength. P_q is considered as a simple indicator for performance evaluation from the QoS point of view. To maximize P_q , the objective function ObjFun2 shown in Algorithm 2 is used.

3) *Objective function 3:* The third objective is to achieve the load balancing between networks: P_o is the percentage between the number of users in both networks. It is used as a basic indicator for load balancing. If the resources of both networks have the same importance and cost for network operator, this objective is considered appropriate for performance evaluation from network operator point of view. To achieve load balancing, the objective function ObjFun3 shown in Algorithm 3 is used.

The weights of the input criteria W_v , W_s , W_t , W_u have been encoded using real encoding method. The length of the real-valued encoding is 4 real floating numbers. The length of each floating number is depending on the internal precision and roundoff used by the computer to define the precision

of the floating numbers. The results achieved by extensive comparisons of GA performance as a function of the different GA parameters (i.e., the population size, the mutation rate, and the crossover fraction) and the different GA operators (i.e., the selection and crossover operators) have been summarized by [27].

They have concluded that crossover fraction, selection operator, and crossover operator are not of much importance. On the other side, the population size and the mutation rate have the most significant impact on the ability of the GA to find better minimum value for objective function. Based on that, several sets of experiments to determine good population size and mutation rate for our proposed objective functions have been carried out. Each experiment has been repeated several iterations while the performance is recorded. The results from all the iterations are then combined by calculating the average (and standard deviation) for each experiment. Usually, testing GAs includes mainly two issues; how far the result obtained by GA is from the benchmark results, which can be measured by average fitness value. The second issue is how fast GA is in finding the best solution, which can be measured by the total number of function evaluations.

a) *Parameter initialization:* Since our GA is working offline, only the average best fitness value achieved by the GA is used. The average best fitness and the standard deviation are plotted against the tested parameters. When studying a parameter all the other parameters and operators are kept constant. Only a sample for the experiments results achieved is given in this paper as shown in Figure 5.

The lower plot in Figure 5 shows the me *ADR* and standard deviations of the ObjFun1 function best fitness values over 20 runs, for each of the values of the population sizes between 10 and 100 with step size equals to 10 individuals.

The upper plot shows a color-coded display of the best fitness values in each run where dark color indicates better results. For ObjFun1 fitness function, setting the population size to 20 individuals or more yields better result. Based on these results and the other experiments results for other objective functions, the population size chosen in our GAs has been set to 20.

Other sets of experiments have been carried out to determine a suitable mutation rate. Based on the experiments, the mutation rate has been set to 0.1. In addition, our GA uses a roulette selection function.

b) *Elitism:* To ensure elitism, the number of individuals that are guaranteed to survive to the next generation has been set to 4 individuals. It uses two-point crossover function and a uniform mutation function. The GA terminates if the maximum number of the GA iterations reaches to 300 iterations, if there is no improvement in the best fitness value for number of consecutive generations equals to 100 generation, or if there is no improvement in the best fitness value for an interval of time equals to 300 seconds.

C. The MCDM Component

The input criteria of the MCDM are the outputs of the FL-based control subsystems in the first component. The GA component assigns the weight w_i for criteria i to reflect its relative importance. The criteria with more importance to the operator and user can be assigned higher weight using the objective function of the GA specified by the operator. Since all the outputs of FL subsystems are in the range $[0, 1]$, there is not be any need to scale the criteria performance against alternatives.

a) simple multi-attribute rating technique (SMART):

Enhanced version of simple multi-attribute rating technique (SMART) has been used. SMART is one of the simplest MCDM and most efficient methods. The ranking value x_j of alternative A_j is obtained simply as the weighted algebraic mean of the utility values associated with it, that is, a_{ij} according to (1):

SMART employs relatively uncomplicated and straightforward manipulation method, which makes it stronger and easier to use in a hybrid and more complex models such as the proposed one in this paper. With the aid of both FL and GA, SMART has all the capabilities required to address the specific considerations that are involved in the *ADR* decision making process. SMART can be quickly and easily understood by the inexperienced decision makers. However, to get better understanding and deeper insights into the selection decision making, other advanced MCDM methods can be used in the future work and compared with SMART method. In our algorithm, there are two alternatives for the MCDM, one is a CDMA-based network and the other is a TDMA-based network.

Ranking function: The ranking value of CDMA network x_c and the ranking value of TDMA network x_t can be calculated as follows:

where:

- W_v is the assigned weight for the MSS subsystem.
- W_s is the assigned weight for the RSS subsystem.
- W_t is the assigned weight for the ST subsystem.
- W_u is the assigned weight for the user UP subsystem.
- T_w is the total weight and is calculated using (3):

V. EXPERIMENTS

Our proposed scheme and SA are evaluated using the simulation approach.

MATLAB mathematical software and a set of functions called RUNE [1] have been used for the simulation. The system model considers the coexistence of a CDMA-based WWAN network with seven macrocells with omnidirectional antenna and cell radius = 1000 m, and a TDMA-based WLAN network with twelve microcells with omnidirectional antenna and cell radius = 500 m. In the system environment, each mobile has a velocity and is moved with a random distance and a random direction at defined time steps. The velocity is a vector quantity with magnitude and direction. The traffic is modeled according to Poisson process. The main call holding

time is assumed to be 50 seconds. Four types of services are considered in our simulation,

- he voice calls,
- the low bit rate real-time video telephony,
- the high bit rate streaming video,
- and the nonreal-time data traffic.

The velocity of the i th mobile is updated according to (4);

$$G = GD + GF + GR + GA \quad (2)$$

where:

- U_i is the complex speed [m/s].
- P is the correlation of the velocity between time steps.
- X is Rayleigh distributed magnitude with mean 1 and a random direction.
- V_m is the mean speed of mobiles.

The propagation model used in this paper can be described in logarithmic scale as in (5)

$$G = GD + GF + GR + GA \quad (3)$$

where:

- 1) The first component is the distance attenuation GD, which is given by the Okumura-Hata formula (x): where:
 - d is the distance to the transmitter,
 - is a constant used to model the effect of carrier frequency, the antenna size and other physical parameters. It has been set to 28 dB
 - The parameter is the distance attenuation coefficient. It has been set to 3.5.
- 2) The second component is the shadow fading GF, which is modeled as a log-normal distribution with standard deviation of 6 dB and 0 dB mean.
- 3) The third component is the Rayleigh fading GR, which is a Rayleigh distributed.
- 4) The fourth component is the antenna gain GA.

VI. RESULTS EXPLOITATION

This section shows some simulation results and compares the performance of our proposed solution to three different reference *ADR* algorithms.

A. servicetype-based selection algorithm

The first algorithm is a servicetype-based selection algorithm where high bit services with low propagation delay requirements are sent to the WLAN and the low bit rate services with the high propagation delay requirements are sent to the WWAN.

B. terminal speed-based selection algorithm

The second algorithm is a terminal speed-based selection algorithm where highspeed users are sent to the large-coverage network (i.e., WWAN) and the low-speed users are sent to the smallcoverage network (i.e., WLAN).

C. random-based selection algorith

The third algorithm is a random-based selection algorithm where the users are assigned randomly to the two networks.

All solutions have been simulated, evaluated, and compared for the three different objectives addressed in Section x. Several runs of simulation have been carried out. Each run uses a different number of users.

Results interpretation

Figure x shows P_u values in all solutions. The horizontal axis shows the number of users while the vertical axis shows the P_u values.

From both Figure x and the numerical samples for P_u values shown in Table x, the great improvement in the number of the satisfied users in our solution can be seen. For example, with 840 users in the environment, the percentage of satisfied users with all different reference solutions is around 50% while the same percentage in our proposed solution is around 80.7%. In general, our proposed solution achieves around 31% enhancement over the different reference algorithms.

In Figure x, the lower plot shows the weights assigned by the GA for the different criteria using the first objective function. The upper plot shows the best function value in each generation versus iteration number.

Figure x shows P_q values in all solutions. The horizontal axis shows the number of users while the vertical axis shows the P_q values.

From both Figure x and the numerical samples for P_q values shown in Table x, the great improvement in the percentage of the users assigned to networks with stronger received signal can be seen. For example, with 840 users in the environment P_q with the different reference algorithms is around 50% while the same percentage in our proposed solution is around 64%. In average, our proposed solution achieves around 26.5% enhancement over the random-based selection.

Figure x shows the weights assigned by the GA for the second objective. The results of P_o produced by all solutions for the third objective are shown in Table x.

Our proposed solution achieves good and comparable results against the results achieved by other solutions. Figure x shows the weights assigned by the GA for the third objective.

VII. CONCLUSION

A novel framework to solve the *ADR* problem has been presented in this paper.

The proposed framework is scalable and is able to handle the huge number of configurations with a large set of criteria.

The framework can cope with the different and conflict view points and purposes of the operator and users, it can react to the accumulated end users experience about the *ADR*.

The proposed framework has been developed to present and design a multi-criteria *ADR* solution that considered the end user, the application QoS requirements, the environment conditions, and the operator view points.

The main focus of our work is to use a multi-objective optimization method to find the optimum weights jointly,

rather than using the single-objective GA that optimize each weight independently.

As the proposed membership functions and rules of the fuzzy subsystems are still subjective, they can be tuned using a suitable learning method such as the neural networks or the genetic algorithms.

Q-Learning

Aghiles Djoudi¹², Rafik Zitouni², Nawel Zangar¹ and Laurent George¹

¹LIGM, UMR 8049, École des Ponts, UPEM, ESIEE Paris, CNRS, UPE, France

²ECE Research Lab Paris, 37 Quai de Grenelle, 75015 Paris, France

Email: {aghiles.djoudi, nawel.zangar, laurent.george}@esiee.fr, rafik.zitouni@ece.fr

Abstract—One of the most important problem ... This limitation could be addressed by Our work is motivated by Our goal is to

The difficulty is mainly due to To address this problem,
Particularly,
We validate our approach by
Simulation results show that
Furthermore,

I. INTRODUCTION

The need of The main factor is
The difficulty to build such system is
For that purpose, we ... In this work we ...

This paper is organized as follows. Section II elucidates summary of related works. Section ?? provide the required background. In Section IV, we ... our approach to solve ... Our experimentation is presented in Section V. Our findings are presented in section ?? . Section VII concludes this paper.

II. RELATED WORK

III. BACKGROUND

Selection of technology

- A. *Hardware*
- B. *Operating system*
- C. *Communication protocol*
- D. *Workspace and tools*

IV. APPROACH

V. EXPERIMENTATION

VI. RESULTS EXPLOITATION

- A. *Range*
- B. *Response time*
- C. *Connection speed*
- D. *Power consumption*

VII. CONCLUSION

The main challenge of this work was to The efficiency of such algorithms

Our main contribution was to
To measure the accuracy of
As we find that ..., we plan to ..

...

Aghiles Djoudi¹², Rafik Zitouni², Nawel Zangar¹ and Laurent George¹

¹LIGM, UMR 8049, École des Ponts, UPEM, ESIEE Paris, CNRS, UPE, France

²ECE Research Lab Paris, 37 Quai de Grenelle, 75015 Paris, France

Email: {aghiles.djoudi, nawel.zangar, laurent.george}@esiee.fr, rafik.zitouni@ece.fr

Abstract—The purpose of this meeting was to talk about our approach to use genetic algorithm to select the LoRa transmission setting that best fit the application requirements. Beside this, we talked about the ability to adapt this work to a building application and particularly to track persons, objects in a construction site. Clustering algorithms could be helpful in this kind of application

Because allows devices to be located and placed in in based on their

One of the most important problem ... This limitation could be addressed by Our work is motivated by Our goal is to

The difficulty is mainly due to To address this problem,

Particularly,

We validate our approach by

Simulation results show that

Furthermore,

I. INTRODUCTION

The need of The main factor is

The difficulty to build such system is

For that purpose, we ... In this work we ...

This paper is organized as follows. Section II elucidates summary of related works. Section ?? provide the required background. In Section IV, we ... our approach to solve ... Our experimentation is presented in Section V. Our findings are presented in section ?. Section VII concludes this paper.

II. RELATED WORK

A. Objenious

III. BACKGROUND

Selection of technology

A. Hardware

B. Operating system

C. Communication protocol

D. Workspace and tools

IV. APPROACH

V. EXPERIMENTATION

VI. RESULTS EXPLOITATION

A. Range

B. Response time

C. Connection speed

D. Power consumption

VII. CONCLUSION

The main challenge of this work was to The efficiency of such algorithms

Our main contribution was to

To measure the accuracy of

As we find that ..., we plan to ..

Testbed

Aghiles Djoudi¹², Rafik Zitouni², Nawel Zangar¹ and Laurent George¹

¹LIGM, UMR 8049, École des Ponts, UPEM, ESIEE Paris, CNRS, UPE, France

²ECE Research Lab Paris, 37 Quai de Grenelle, 75015 Paris, France

Email: {aghiles.djoudi, nawel.zangar, laurent.george}@esiee.fr, rafik.zitouni@ece.fr

Abstract—

I. INTRODUCTION [1]

A. Problem Statement

B. 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].

C. 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.

D. 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.

E. 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 instead 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 given and other issues and questions handled.

II. RELATED WORK

III. 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.

A. 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 MCUs 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

B. 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

C. 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

D. 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.

IV. 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 Contikis 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.

A. 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.

B. CoAP server

In order to make each nodes 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 MCUs 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 CoAPs 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.

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.

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 as a 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.

V. 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.

A. 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 λ 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. $P_r(\text{dB}) = P_t + G_t + G_r + 20 \log 10(\lambda) 4d c f \text{FSPL}(\text{dB}) = 20 \log 10(d) + 20 \log 10(f) + 147.56$ = (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. $P_r = P_t + G_t + G_r - \text{FM} - \text{FSPL}$ (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. $d = 10 \times P_t + G_t + G_r - P_r - \text{FM} - 147.56 - 20 \log 10(f)$

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

B. 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 to 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: $(2 \times 67.5ms) \times \text{hops}$ (4.11) Best case response time: $(2 \times 5ms) \times \text{hops}$ (4.12) Worst case response time: $(2 \times 130ms) \times \text{hops}$ (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

C. 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 developers 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: $31.25KB/s \times (1 - 0.59) = 12.81KB/s$ 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 equation 4.19. 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: $79B = 0.71 \times 111B$ (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.

D. 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 I²C 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.

VI. 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.

A. 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.

B. 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 nodes 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.

C. 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.

D. 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.

E. 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.

VII. 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 30% the 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.

REFERENCES

- [1] J. Bregell, “ [Hardware and Software Platform for Internet of Things](#) ”, *Master of Science Thesis in Embedded Electronic System Design*, 2015, 00002.

Conclusion

Aghiles Djoudi¹², Rafik Zitouni², Nawel Zangar¹ and Laurent George¹

¹LIGM, UMR 8049, École des Ponts, UPEM, ESIEE Paris, CNRS, UPE, France

²ECE Research Lab Paris, 37 Quai de Grenelle, 75015 Paris, France

Email: {aghiles.djoudi, nawel.zangar, laurent.george}@esiee.fr, rafik.zitouni@ece.fr

I. CONCLUSION

II. PERSPECTIVES

Survey platforms

Aghiles Djoudi¹², Rafik Zitouni², Nawel Zangar¹ and Laurent George¹

¹LIGM, UMR 8049, École des Ponts, UPEM, ESIEE Paris, CNRS, UPE, France

²ECE Research Lab Paris, 37 Quai de Grenelle, 75015 Paris, France

Email: {aghiles.djoudi, nawel.zangar, laurent.george}@esiee.fr, rafik.zitouni@ece.fr

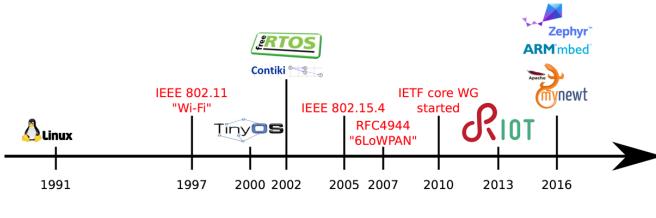


Figure 1. .

I. IoT END DEVICES

A. 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 to 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.

1) *Contiki*: Contiki is an 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 Phototreads [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 Phototreads. 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 aspects 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/html, 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:

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.

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: through 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) *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, preemptive 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.

| | LiteOS | Nano-RK | MANTIS | Contiki |
|-------------------------------|----------------------------|------------------------------|----------------------|---------------------------|
| Architecture | Monolithic | Layered | Modular | Modular |
| Scheduling | Round Robin | Monotonic harmonized | Priority classes | Interrupts execute w.r.t. |
| Memory | | | | |
| 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 I. 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/semaphore | Synchronization/primitives |

Table II. Common operating systems used in IoT environment
al-fuqaha_internet_24

5) Summary and conclusion:

B. Hardware platform

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 (I2C), and Controller Area Network (CAN). These interfaces allow developers to attach 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 Moores 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.

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)

3) Sensing Unit:

- a) GPS:*
- b) Humidity:*
- c) Temperature:*

| Ref | Module | Fre-quency MHz | Tx power | Rx power | Sens-itiv-ity | Chan-nels | Dis-tance |
|---------------|-----------|------------------------------|----------|----------|---------------|-----------|-----------|
| libelium_wasp | Stem1017 | 863-870 (EU) 902-928 (US) | 14 dBm | dBm | -134 dBm | 8 13 | 22+ km |
| libelium_wasp | mm1243017 | | | | | | |

Table III

C. Summary and discussion

D. Lora

| Preamble | | Sync msg | | PHY Header | | PHDR-CRC | | PHY Payload | | | | | | | | | | | | CRC | |
|------------|--------|----------|------------|------------|------------|-------------|-------|--------------|---------|-----|-----------|-----|-------------|----------|------|------|-------|---------------|---------------|---------------|---------------|
| Modulation | length | Sync msg | PHY Header | PHDR-CRC | MAC Header | MAC Payload | | | | | | | | | | | | MIC | CRC Type | Polynomial | |
| Modulation | length | Sync msg | PHY Header | PHDR-CRC | MType | RFU | Major | Frame Header | | | | | | | | | | | | Frame Payload | Frame Payload |
| Modulation | length | Sync msg | PHY Header | PHDR-CRC | MType | RFU | Major | FCtrl | | | | | | | | | | | | Frame Payload | Frame Payload |
| Modulation | length | Sync msg | PHY Header | PHDR-CRC | MType | RFU | Major | NwkID | NwkAddr | ADR | ADRACKReq | ACK | FPendingRFU | FOptsLen | FCnt | FOps | FPort | Frame Payload | Frame Payload | CRC Type | Polynomial |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |

- 0) **Modulation :**
 - ↳ LorA: 8 Symbols, 0x34 (Sync Word)
 - ↳ FSK: 5 Bytes, 0xC1194C1 (Sync Word)
- 1) **Length :**
 - 2) **Sync msg :**
 - ↳ 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)
 - ↳ 0000 Join Request
 - ↳ 0011 Join Accept
 - ↳ 0100 Unconfirmed Data Up
 - ↳ 0111 Unconfirmed Data Down
 - ↳ 1000 Confirmed Data Up
 - ↳ 1011 Confirmed Data Down
 - ↳ 1100 RFU
 - ↳ 1111 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
- 11) **ADRACKReq :** (Adaptive Data Rate ACK Request): if network doesn't respond in 'ADR-ACK-DELAY'
 - ↳ 0 No change
 - ↳ 1 if (ADR-ACK-CNT) \geq (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-FCNT-GAP
 - ↳ 0 otherwise
- 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 $x^{16} + x^{15} + x^5 + 1$

II. DIVERS

$$\begin{aligned}
& \text{SF} = \log_2 \frac{R_c}{SR} \quad (4) \\
& R_c[\text{chips/s}] = BW \quad (5) \\
& R_S[\text{symbols/s}] = \frac{R_c}{2^{\text{SF}}} = \frac{BW}{2^{\text{SF}}} \quad (6) \\
& R_m = DR[\text{bps}] = SF \cdot RS = SF * \frac{BW[\text{Hz}]}{2^{\text{SF}}} * CR \quad (7) \\
& SF = \log_2 \frac{R_c}{SR} \quad (8) \\
& \mathbf{T}_s = \frac{2^{\text{SF}}}{BW[\text{Hz}]} \quad (9) \\
& SR[\text{sps}] = \frac{BW}{2^{\text{SF}}} \quad (10) \\
& DR[\text{bps}] = SF * \frac{BW[\text{Hz}]}{2^{\text{SF}}} * CR \quad (11) \\
& BR[\text{bps}] = SF * \frac{4}{\frac{4+CR}{2^{\text{SF}}}} \quad (12) \\
& Sen[\text{dBm}] = -174 + 10 \log_{10} BW + NF + SNR \quad (13) \\
& SNR[\text{dB}] = 20 \log \left(\frac{S}{N} \right) \quad (14) \\
& SNR[\text{dB}] = 20 \log \left(\frac{S}{N} \right) \quad (15) \\
& BER[\text{bps}] = \frac{8}{15} \cdot \frac{1}{16} \cdot \sum k = 216 - 1^k \left(\frac{16}{k} \right) e^{20 \cdot SINR \left(\frac{1}{k} - 1 \right)} \quad (16) \\
& BER[\text{bps}] = 10^{\alpha \cdot e^{\beta \cdot SNR}} \quad (17) \\
& PER[\text{pps}] = 1 - (1 - BER)^{n_{\text{bits}}} \quad (18) \\
& PRR = (1 - BER)^L \quad (19) \\
& RSSI = Tx_{\text{power}} \cdot \frac{Rayleigh_{\text{power}}}{PL} \quad (20) \\
& \quad (21)
\end{aligned}$$

fakhfakh_deep_2017

$$MSE = \frac{1}{n} \sum_{i=1}^n (p_i - r_i)^2 \quad (22)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (p_i - r_i)^2} \quad (23)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |p_i - r_i| \quad (24)$$

$$Recall = \frac{TP}{TP + FN} \quad (25)$$

$$Precision = \frac{TP}{TP + FP} \quad (26)$$

$$F1_Score = \frac{2 \times Precision \times Recall}{precision + recall} \quad (27)$$

$$TPR = \frac{TP}{TP + FN} \quad (28)$$

$$FPR = \frac{FP}{FP + TN} \quad (29)$$

$$ROC = (TPR, FPR) \quad (30)$$

$$Novelty = \sum_{i \in L} \frac{\log_2 P_i}{n} \text{ where } P_i = \frac{n - rank_i}{n - 1} \quad (31)$$

$$Serendipity = \frac{1}{n} \sum_{i \in n} \max(P_{\text{user}} - P_U, 0) \times rel_i \quad (32)$$

$$diversity = \frac{a}{c} \sum_{i=1}^c \frac{1}{n} \sum_{j=1}^n i_j \quad (33)$$

$$Coverage = 100 \times \frac{u}{U} \quad (34)$$

$$Stability = \frac{1}{P_2} \sum_{i \in P_2} |P_{2,i} - P_{1,i}| \quad (35)$$

$$DCG = rel_1 + \sum_{i=2}^{\text{pos}} \frac{rel_i}{\log_2 i} \quad (36)$$

$$IDCG = rel_1 + \sum_{i=2}^{|h|-1} \frac{rel_i}{\log_2 i} \quad (37)$$

$$NDCG = \frac{DCG}{IDCG} \quad (38)$$