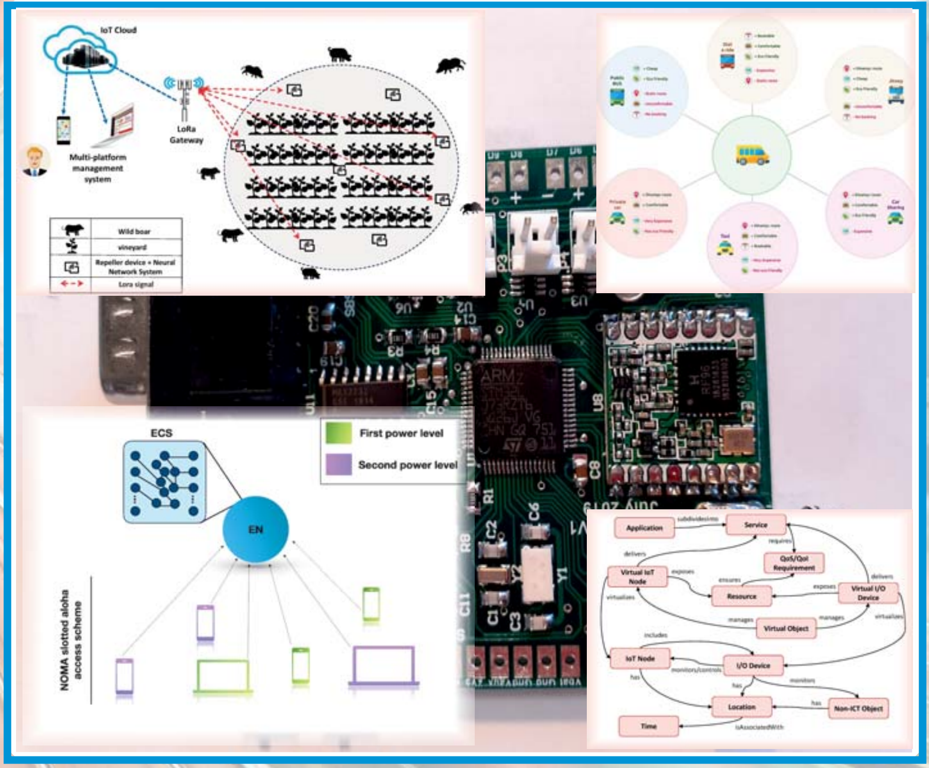
CNIT Technical Report-05

INTERNET OF THINGS: TECHNOLOGIES,CHALLENGES AND IMPACT

CNIT
Technical Report-05

INTERNET OF THINGS:
TECHNOLOGIES, CHALLENGES AND IMPACT



Edited by
Luigi Atzori & Gianluigi Ferrari

cnit consorzio nazionale
interuniversitario
per le telecomunicazioni

cnit consorzio nazionale
interuniversitario
per le telecomunicazioni

TECHNICAL REPORT-05

INTERNET OF THINGS:

TECHNOLOGIES, CHALLENGES AND IMPACT

CNIT TECHNICAL REPORT-05

INTERNET OF THINGS:
TECHNOLOGIES, CHALLENGES AND IMPACT

Edited by Luigi Atzori & Gianluigi Ferrari

Copy-editing: *Gaspare Galati* and *Sergio Pandiscia*
Graphics and Cover Design: *Sergio Pandiscia*

TABLE OF CONTENTS

PREFACE

1. **Low-power Wide-Area Networks:**
   A Comparative Analysis Between LoRaWAN and NB-IoT 1
   by C. Buratti, K. Mikhaylov, R. Marini, R. Verdone

2. **LoRaWAN: current status and research directions** 23
   by L. Vangelista, A. Zanella, M. Zorzi

3. **LoRaWAN: a Deep Dive in a Large Scale Deployment and**
   in Radio Access Optimization Strategies 41
   by G. Bianchi, F. Cuomo, D. Garlisi, P. Pisani, I. Tinnirello

4. **Wide Area Transmission Technologies for IoT** 67
   by A. Abrardo, G. Peruzzi, A. Pozzebon

5. **Enabling technologies for the Internet of Vehicles:**
   standards, research and open challenges 87
   by C. Campolo, A. Molinaro

6. **A Machine Learning Based Non-Orthogonal Multiple**
   Access Scheme for IoT Communications 103
   by R. Fantacci, B. Picano

7. **Resources virtualization and task offloading towards the Edge in the IoT** 123
   by G. Merlino, V. Pilloni

8. **Security and privacy in the IoT: how to enforce standard communication**
   technologies with efficient and flexible mechanisms 141
   by A. Suriano, D. Striccoli, G. Piro, A. Antenore, G. Boggia

9. **The Social Internet of Things: a Survey** 163
   by L. Atzori, A. Iera, G. Morabito

10. **Software Defined Fog/Edge Networking for Internet of Vehicles: a Services-Oriented Reference Architecture** 183
by M. Bonanni, F. Chiti, R. Fantacci

11. **IoT-enabled Smart Monitoring and Optimization for Industry 4.0** 207
by L. Davoli, L. Belli, G. Ferrari

12. **A Flexible Mobility System based on Small and Low-emission Vehicles for Smart and Green Mobility** 227
by S. Ullo, M. Gallo, M. Di Bisceglie, C. Galdi, M. Marinelli,
L. Glielmo, G. Palmieri, P. Amenta, A. Ferrara, M. Ferrucci,
G. Romano, M. Russo, M. De Angelis

13. **An IoT Solution and Real-Time Detection System for Crop Protection against Ungulates** 249
by M. O. Ojo, D. Adami, S. Giordano

PREFACE

More than two decades have passed since scholars from different fields around the world have been tackling various issues related to the realization of solutions to bring the physical world into the digital one to make easier the access to the former. At the beginning, great attention has been devoted to the implementation of RFID-based systems for tracking raw material (when reaching the production plant) and goods (when delivered to the shops to reach the customers), significantly impacting the logistics section. Then, the focus has moved to more complex communication technologies in order to extend the plethora of devices that could be incorporated into the digital world and make almost any objects reachable on the Internet. Afterwards, the attention has shifted to the technologies needed to create virtual copies in the cloud of the physical devices to augment the real world things with additional functionalities, such as increasing the processing and storage capabilities, extend the set of languages the objects could speak, foster interoperability with other devices deployed in other platforms and the like. This brought to the implementation of several platforms that followed different architectural models and were capable of exploiting edge computing facilities. Whereas general-purpose IoT platforms are not developed anymore, quite significant efforts are devoted to their customization needed in diverse application domains. This is a more market-oriented research that is still going on, with the intention to tackle application specific requirements, which could be related to the types of sensors to be used, the radio coverage, specific data processing functionalities and/or the capabilities to mashup services coming from different systems. The most relevant application domains where IoT solutions are now operational are those related to: logistics, with the tracking of good from the source of raw material to product in the hands of the consumer for after-market assistance; mass transport services, with IoT systems capable of tracking the fleets and providing real-time information to the customers; agriculture applications, where rovers can autonomously move in the production field to collect key data to drive the production activities, just to cite a few. In some others, engineering efforts are still going on to deploy effective and successful solutions.

As complete *IoT platforms* are now available and operating in some key market segments, they also disappeared from the Gardner hype cycle for emerging technologies after August 2018. Even if this does not mean that IoT is not a profitable research domain anymore, it highlights that investing in research for general-purpose IoT platforms is no longer fruitful. Indeed, when looking at the 2020 Gardner top ten strategic technologies trends, we observe that some of these rely on the widespread use of the technologies that are predominant in the IoT domain. For instance, one of these is *hyper-automation*, which consists in the creation of a digital twin of the organization that is benefiting from this innovation. It is certainly possible to create such a twin only if the physical world is monitored and accessed through connected IoT devices. *Autonomous things* is another trend mentioned in this list, which means that the connected things (drones, robots, ships and appliances) become autonomous in taking decisions and acting accordingly. And this will happen clearly with a deep use of AI technologies as well as with an extensive knowledge of the current context and past experiences the autonomous things are and were involved in: this can happen only if we have disseminated the environment with sensing things. Similar considerations could be extended to most of the other list of hot Technologies. From this analysis we can say that IoT is not recognized as a hot technology by itself, as most of the characterizing issues have already been fixed and proper solutions provided. However, it is a major and mandatory component in the new trends as there is always the need for obtaining an increasingly accurate view of the physical world into the digital one. The new trends then take the names from the specific applications that are being considered, which can be considered as research subfields that have been generated by it and then threads that will drive the innovation in the future and open new areas of research for our community. Whether in these new areas the IoT will disappear or not does not make the difference to us as researchers in this field!

Most of the key technological areas mentioned above have been considered by the authors that have contributed to this book. The contributions have then been grouped into three parts. In the *first part*, the first four chapters are dedicated to low-power wide area networks. In Chapter 1, Buratti *et al* discuss, in a comparative way (analytically, via simulations and experiments), on the main features of two key technologies: Long Range Wide Area Network (LoRaWAN) and 4G Narrowband IoT (NB-IoT). In Chapter 2, Zanella *et al* discuss the main characteristics of the LoRaWAN technology and present some results that shed light on the effect of different parameters settings in some illustrative scenarios. In Chapter 3, Bianchi *et al* present and discuss the lessons learned from a real-world LoRaWAN deployment carried out by the UNIDATA regional operator in a large terrestrial area. In Chapter 4, Abrardo *et al* focus on wide area transmission technologies for IoT along with their application scenarios and requirements, considering both license-based (i.e., cellular) and license-free (e.g., LoRaWAN and other Sub-GHz technologies) technologies. The two final chapters, which complete the first part of the book, are related to vehicular communications and machine learning-based multiple access. In Chapter 5, Campolo and Molinaro describe the Internet of Vehicles (IoV) status quo, by analyzing the V2X application requirements and the main enabling communications technologies, and outline research challenges and opportunities related to edge computing, virtualization, artificial intelligence, and other IoV enablers. In Chapter 6, Fantacci and Picano propose the application, at a local computation node according to the emerging edge computing paradigm, of an Echo State Network (ESN) machine learning framework to optimize Non-Orthogonal Multiple Access (NOMA) of a very large number of IoT devices with minimized latency and network congestion.

In the *second part*, the four chapters focus on various aspects related to platforms, services, and resource management. In Chapter 7, Merlino and Pilloni propose a middleware architecture that relies on equipping co-located smart Edge and Fog devices with agents that virtualize real objects' features, resources and services, thus pushing computation from the remote Cloud as close to data sources as possible. In Chapter 8, Suriano *et al* focus on IoT security services: first, they highlight the impact that their implementation may have on the behavior of standardized communication protocols; then, some interesting approaches emerged in the recent scientific literature, addressing the uncovered security services, are presented. In Chapter 9, Atzori *et al* focus on the Social Internet of Things (SioT) paradigm for the realization of IoT solutions, discussing different views, major technologies, enabled applications, current challenges and future directions. In Chapter 10, Bonanni *et al* propose an innovative architectural concept for Internet of Vehicles (IoV), integrating multiple paradigms: Software Defined Networking (SDN), Network Function Virtualization (NFV), Mobile Fog/Edge Computing (MFEC).

In the *third part*, illustrative applications to the domain of smart industry, smart transport, and smart agriculture are considered. In Chapter 11, Davoli *et al* present an example of an IoT-based Industry 4.0 “renovation" strategy aimed at allowing to continuously collect heterogeneous Human-to-Things (H2T) and Machine-to-Machine (M2M) data, which can be used to optimize and improve a factory as a whole entity. In Chapter 12, Ullo *et al* introduce an innovative mobility system, based on small and low-emission vehicles, aimed at spreading the Mobility as a Service (MaaS) paradigm through connected heterogeneous intelligent and personalized architecture integrated within an Intelligent Transportation System (ITS) framework. In Chapter 13, Ojo *et al* present an IoT system for crop protection against animal intrusions, describing the deployed experimental solution based on a low-power wireless network, neural network-based processing for real-time animal detection and the back-end system.

We wish to thank all authors for considering this CNIT Technical Report as a venue for disseminating their research outcomes. We express our gratitude to all reviewers who devoted their precious time in providing valuable feedback on all the received contributions. Finally, we hope that this Book will provide a valuable source of reference for researchers, designers, and engineers working in this area of Internet of Things.

Luigi Atzori & Gianluigi Ferrari

Low-power Wide-Area Networks: A Comparative Analysis Between LoRaWAN and NB-IoT

Chiara Buratti*, Konstantin Mikhaylov†, Riccardo Marini*, Roberto Verdone*

*Università di Bologna, †University of Oulu

Abstract: *Low Power Wide Area Networks are becoming the most important enabler for the Internet of Things (IoT) connectivity. Application domains like smart cities, smart agriculture, intelligent logistics and transportation, require communication technologies that combine long transmission ranges, energy efficiency and low infrastructure costs. Recent and future trends make LoRaWAN and 4G/NB-IoT the main drivers of IoT business. In this contribution we briefly discuss the main features of the two technologies, analyzing some important Key Performance Indicators. The presented results have been obtained analytically, via simulations and experiments developed at the University of Bologna via testbeds, that are currently under use to verify different IoT applications and demonstrate their feasibility.*

1 Introduction

The proliferation of embedded systems, wireless technologies, and Internet protocols have enabled the Internet of Things (IoT), to bridge the gap between the virtual and physical world by enabling the monitoring and control of the physical world by data processing systems. A large variety of communication technologies has gradually emerged, reflecting a large diversity of application domains and requirements. Some of these technologies are prevalent in a specific application domain, such as Bluetooth Low Energy in Personal Area Networks [1], and Zigbee in Home Automation systems [2]. Others, like Wi-Fi Low Power, Low Power Wide Area Networks (LPWAN) [3], and cellular communications, such as the 3GPP Long Term Evolution for Machines (LTE-M) and Narrowband IoT (NB-IoT), have a much broader scope. In addition, this landscape is constantly and rapidly evolving, with new technologies being regularly proposed, and with existing ones proliferating into new application domains. In this Chapter, we focus on LoRa and NB-IoT, presenting their main features and characteristics, and some examples of achievable results via analyzing a set of selected generic Key Performance Indicators (KPIs). The Chapter is organized as follows: Section 2 deals with the LoRa technology, while Section 3 reports NB-IoT technology. Finally, Section 4 compares the two solutions and reports drawn conclusions.

2 LoRaWAN Technology: main features and characteristics

2.1 LoRaWAN Technology

2.1.1 Overview

LoRa is a Physical Layer developed by Cycleo (a French company) later acquired by Semtech, on top of which LoRaWAN specifies the link and network layer procedures. The first target of LoRa is to allow very low power operations to ensure with a single battery a long lifetime to the devices - of more than ten years. It also allows long communication ranges (2-5 km in urban areas and up to 15 km in suburban areas) [20, 18]. The downside is low data rates, some tens of bit per second in the most robust options. However, LoRa can offer certain flexibility and can reach a data rate up to 50 kbit/s [4, 5]. LoRa physical layer is based on Chirp Spread Spectrum (CSS) modulation. Using a bandwidth exceeding the necessary one to transmit the data, LoRa performs spectrum spreading, which brings robustness against some characteristics of the channel (e.g., interference, frequency selectivity, Doppler effect). One original characteristic of LoRa is that information is carried by a cyclic shift in the chirp (position modulation).

The transmitter generates chirp signals by varying their frequency over time and keeping phase between adjacent symbols constant. The signal frequency band is usually set to 125, 250 or 500 kHz in the Industrial Scientific Medical (ISM) bands of 863-870 MHz for Europe or 902-928 MHz for US[21]. However, there also exist some narrower bands (7.8 to 62.5 kHz) in the 166 and 433 MHz bands. Finally, a new version at 2.4 GHz has recently emerged. The main characteristics of LoRa's modulation depend on several parameters:

- The Spreading Factor (SF): it is related to the duration of a symbol. For the higher spreading factors, more chips are combined in a single symbol, making the transmission longer (thus reducing the data rate), but increasing its energy thus, potentially, allowing longer communication range. LoRa employs six quasi-orthogonal SFs (numbered 7 to 12). Consequently, up to six frames can be exchanged in the network at the same time over the same frequency channel, as long as each one is configured with unique SF.

- Forward Error Correction (FEC) techniques, and, specifically, Hamming code, are also used to increase receiver sensitivity. The Code Rate (CR) index specifies the number of additional bits added to a LoRa frame. LoRa offers CR = 0, 1, 2, 3 and 4, where CR = 0 means no encoding and the effective coding rate is 4/(4+CR), ranging from 1 (no coding) to 1/2.

- The output of the encoder passes through the Whitening block (optional). Whitening induces randomness, in order to make sure that there are no long chains of 0's and 1's in the payload. An interleaving block is then implemented to avoid bursts of errors. The interleaver uses a diagonal placing method to scramble each codeword.

A packet contains a preamble (for the detection and synchronization purpose), possibly a header (depends on operation mode) and the payload, with a maximum size between 51 bytes and 222 bytes, depending on the SF. The raw on-air data rate varies according to the SF and the bandwidth, and it ranges between 22 bit/s (BW = 7.8 kHz and SF = 12) to 27 kbit/s (BW = 500 kHz and SF = 7). The SF 6 offers another option with a rate of 50 kbit/s. Frequency hopping is exploited at each transmission in order to mitigate external interference. The choice of the bandwidth, the SF and the CR impact the Time-on-Air. An increase in this time will consequently increase the duration of the period the radio has to be off, which imposed by the frequency use regulation. Although few information bits are transmitted per packet, the packet duration can be long, more than one second for large SF and small bandwidth. To decode a packet, first, a receiver has to detect the preamble consisting of successive up-chirps (typically 4 or 6) and two down-chirps (the up-chirp reversed in time). This allows the synchronization and the detection of the beginning of the frame. The decoding consists of multiplying each symbol by a down-chirp. The resulting signal is a sine wave with a fixed frequency, given by the shift. The Fourier transform then exhibits a peak, easy to detect, that allows recovering the bit sequence. Besides, the capture effect allows receiving the target packet even under the interference of a signal with the same SF, given that the interfering signal is weaker than the target one.

LoRaWAN networks are based on single-hop transmissions, leading to a star-of-stars topology. Devices transmit their packets directly to Gateways that relay messages to a central Network Server, through another network (Cellular, Wi-Fi or Ethernet for instance). Bi-directional communications are allowed too. LoRaWAN defines three classes of devices (A, B and C):

- Class A devices, aiming low cost and long life devices, use pure ALOHA to access the channel in the uplink. A Class-A device is always in sleep mode unless it has something to transmit. After transmission, the device listens during two window periods, defined by duration, offset time and a data rate. Feedback can only happen after a successful uplink transmission. The second window can increase robustness in the downlink, and it is disabled when the end-device receives downlink traffic in the first window.

- Class B devices are designed to support additional downlink traffic, at the price of higher energy consumption. A Class-B device synchronizes its internal clock using beacons emitted by the gateway. This process is called a “beacon lock”. After synchronization, the device negotiates its ping-interval. The LoRa Server is then able to schedule downlink transmissions on each ping-interval. By doing so, additional downlink traffic can also be supported without relying on prior successful uplink transmissions.

- Class C devices are always listening to the channel except when they are transmitting.

Class A is intended for End-Devices (EDs). The other classes must remain compatible with Class A. The three classes can coexist in the same network and devices can switch from one class to another. However, there is no specific message defined by LoRaWAN to inform the gateway about the class of a device and, hence, this must be handled by the application.

An essential parameter in LPWANs and networks operating in unlicensed bands is the maximum allowed duty-cycle. It corresponds to the percentage of time during which an

Table 1: LoRa Key Parameters Values

Parameter	Value	Comment
Bit Rate	22 bit/s – 50 kbit/s	Depending on SFr
Frequency Bands	[69, 433, 868] MHz (Europe) 915 MHz (North America)	2.4 GHz version available
Bandwidth	[125, 250, 500] kHz	7.8 - 62.5 kHz Bandwidths available in the 433 MHz band
Topology	Stars of stars	
Link budget	155 dB – 170 dB	Depending on SF
TX Range	Up to 15 km	Few km in urban area
Consumptions (TX)	18 mA at 10 dBm 84 mA at 20 dBm	

end-device can occupy a channel and equals 1% in EU 868 for end-devices. The channel selection is pseudo-random and happens at each transmission.

The most critical parameters are summarized in Table 1.

2.1.2 Protocol Operation

LoRaWAN networks allow EDs to individually use any of the possible combinations of data rate and transmitted power. This is referred to as Adaptive Data Rate (ADR) and is designed in order to increase the battery life of the ED while maximizing the network capacity [8]. In order to determine the optimal data rate, the network needs to take some measurements: this is achieved by estimating the link budget between the ED and the gateway looking at uplink messages. For example, an ED very close to the gateway should transmit with the highest possible data rate (i.e., the lowest possible value of SF), in order to reduce as much as possible the Time on Air (ToA) of the transmitted packet, allowing the ED to reduce its energy consumption whilst also reducing the probability of collisions with other nodes. The algorithm operating on the network server is designed by the developer while the one working on the node is specified by LoRa Alliance [8].

EDs are in charge of deciding if ADR should be used or not. If it is activated (ADR bit in the frame header set to 1), the network server will control the transmission parameters of the ED through ad-hoc commands. When the ADR bit in the downlink packet is set to 1, the server informs the ED that it will send ADR commands; differently, the node will not receive any indication because the server is not able to estimate the best data rate to be used; this happens when the radio channel varies too fast. Besides this, the node should periodically check if the network still receives its uplink frames when ADR is enabled.

Figure 1 shows the algorithm implemented on the ED. Each time the uplink frame counter is incremented, the same happens to the ADR_ACK_CNT counter (except for repeated transmissions that do not increase the counter). After ADR_ACK_LIMIT messages (by default 64) without any downlink response, the device sends a request to the network, which must respond within the next ADR_ACK_DELAY frames (by default 32) with a downlink frame. If no reply is received, the ED must try to reconnect to the network by first setting the transmitted power to its default one and then possibly switching to the next lower data rate (which will provides longer transmission range). The device must lower its data rate every time the ADR_ACK_DELAY expires and, once it reaches the lowest data rate, it must re-enable all the default uplink frequency channels.

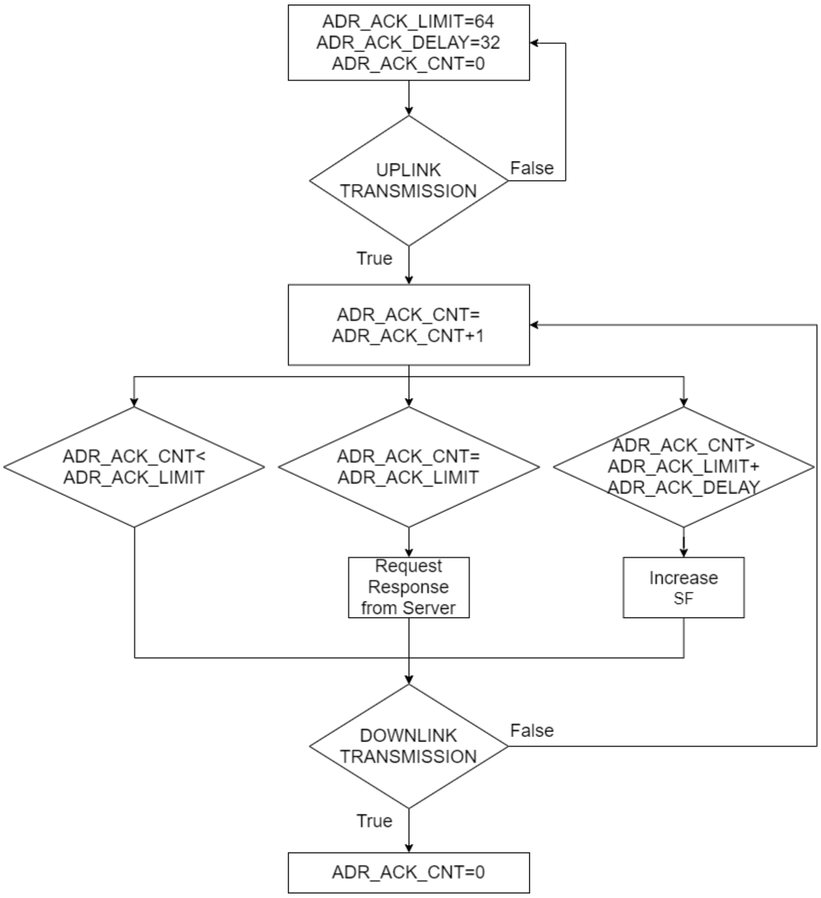

ADR_ACK_LIMIT=64
ADR_ACK_DELAY=32
ADR_ACK_CNT=0
UPLINK TRANSMISSION
False
True
ADR_ACK_CNT= ADR_ACK_CNT+1
ADR_ACK_CNT< ADR_ACK_LIMIT
ADR_ACK_CNT= ADR_ACK_LIMIT
ADR_ACK_CNT> ADR_ACK_LIMIT+ ADR_ACK_DELAY
Request Response from Server
Increase SF
DOWNLINK TRANSMISSION
False
True
ADR_ACK_CNT=0

Figure 1: ED ADR

The ADR algorithm working on the network server exploits data regarding the uplink transmissions processed by the network server in order to define the optimal data rate to be used. One of the most widespread implementations, used by The Things Network or ChirpStack, to mention some of the most famous LoRa Server architectures, is based on Semtech's recommended algorithm [17]. It is shown in Figure 2.

Once the network server detects that the ED is sending a packet with the ADR bit set, it starts collecting Signal-to-Noise Ratio (SNR) measurements of the received signal. It keeps recordings, typically, of the 20 most recent transmissions from each ED. After receiving 20 samples, it computes the SNR margin: $SNR_{margin} = SNR_{max} - SNR_{req} - M$, where SNR_{max} is the maximum SNR among the collected data, M is set as 10 dB by default (it can be controlled by the server administrator), and SNR_{req} is the minimum SNR required to demodulate the received signal correctly, and it varies according to the data rate (or the Spreading Factor) as shown in Table 2.

Data Rate	Spreading Factor	SNR_{req}[dB]
0	12	-20
1	11	-17.5
2	10	-15
3	9	-12.5
4	8	-10
5	7	-7.5

Table 2: SNR_{req} values for different SF with BW=125 kHz.

From SNR_{margin}, N_{step} is computed as: $N_{step} = \left\lfloor \frac{SNR_{margin}}{3} \right\rfloor$.
Then an iterative process starts:

- If $N_{step} < 0$, the transmitted power is increase in each step by 3 dB until the maximum one (according to the regional regulations) is reached; N_{step} is increased by 1;
- If $N_{step} > 0$, first the algorithm tries to increase the data rate in each step until it reaches the maximum one (DR=5) or, if this is not possible anymore, the transmitted power is decreased by 3 dB until it reaches the minimum one; N_{step} is decremented by 1.

The algorithm stops when $N_{step} = 0$ and the server generates a specific packet with the new transmission parameters and sends it to the selected ED; the changes introduced will reflect on the next uplink message if the packet is correctly received.

2.2 LoRa KPI

We concentrate on the following KPIs: Reliability, Network Throughput and the End-to-End (E2E) Delay, and we provide some example of numerical results achieved via simulations and experiments. Simulations have been carried out on Matlab. A squared area with one gateway in the centre has been considered for the sake of simplicity; the square size has been chosen such that nodes using SF7 have a 90% probability of being connected to the gateway, while nodes using SF12 have a connection probability of 100%. In the simulator the path loss is modeled as follows: $Loss = k_0 + k_1 log_{10}(d) + s$, with $k_0 = 10log_{10}\left(\frac{4\pi}{\lambda}\right)^2 \quad and \quad k_1 = 10\beta$, where β is the propagation coefficient and λ is the wavelength, d is the distance between transmitter and receiver. s represents random channel fluctuations, described as a Gaussian r.v. in dB, zero mean and standard deviation σ. The parameters used during simulation are reported in Table 3.

Concerning the experimental platform, Idesio Rigers Board 1.0, a multi-sensor platform specifically designed for smart city applications, has been used. It is equipped with the microchip RN2483 radio transceiver, fully certified 433/868 MHz LoRa module and it supports LoRaWAN Class A devices. Fifteen devices were programmed to work at the same time, divided into three clusters of 5 devices using respectively SF7, SF10 and SF12, sending packets every 60 s.

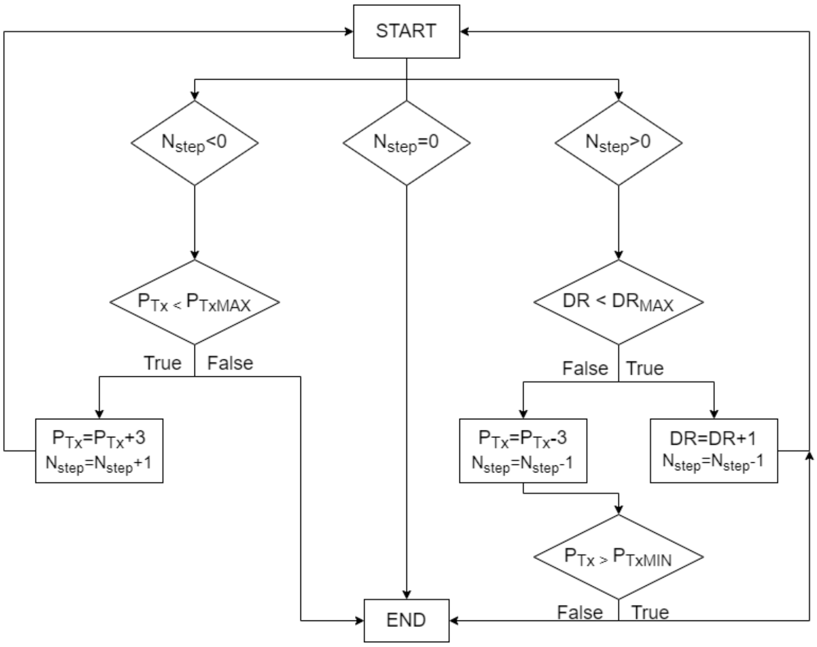


Figure 2: Network Server ADR Algorithm

Packet Periodicity T	60 s	Area Side	6500 m
Packet Size B	16 Bytes	Confirmed Message	Enabled
Preamble Length	8 Symbols	Header	Enabled
f	868,5 MHz	BW	125 kHz
CR	4/5	PTx	13.5 dBm
β	3	σ	3

Table 3: Simulation Parameters

2.2.1 Reliability

Reliability is mainly affected by the use of ALOHA protocol by Class A devices. It is well known that the performance of ALOHA is poor in terms of packet delivery success rate and, further, it does not support a network load increase, due to the interference increase. Hence, as long as the network load is low enough (taking into account the number of SFs, frequency channels, devices), reliability should be enforced. Figure 3 shows the Packet Error Rate (PER) as a function of the number of the nodes in the network, both considering simulations (curve) and experimental results (square points). When computing the PER in the simulations, we account for both, connectivity issues (i.e., probability that an ED is not connected to the gateway) and collisions (i.e., probability that more than one ED transmit at the same time, using the same channel and SF). In the experiments, three measurements sessions of 20 minutes each have been carried out and at the end statistics regarding the PER, obtained by counting the number of packets received/lost, have been computed. In the figure, the different curves are related specific values of SF (in this case, we fixed the same value for all nodes in the network), and to the case of ADR. As for the experiments, no ADR has been considered. As can be seen, the optimum value of SF to be set varies with the number of nodes: when few nodes are present, the PER is mainly limited by connectivity - therefore SF=10 is the best solution. When the traffic load increases, SF=7 becomes better, because it allows keeping under control collisions due to the smaller ToA. Besides this, introducing ADR drastically improves the performance, since it reduces both connectivity and collisions issues managing the SF used by the device, making it able to reach the gateway in almost all cases and distributing different SFs among all the nodes.

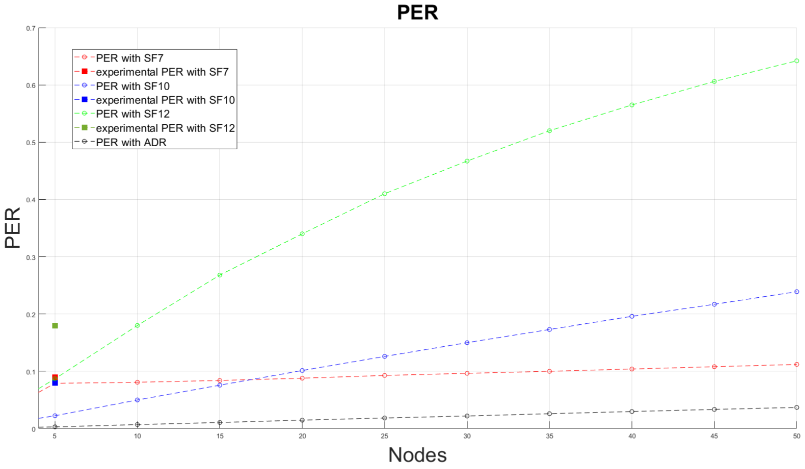


Figure 3: PER versus the number of nodes in the network.

2.2.2 Network Throughput

The network throughput is defined as the number of bits per second correctly received at the gateway, given by $S = \frac{B \cdot N \cdot (1-PER)}{T} \quad [bit/s]$, where N is the number of nodes, T is the period of time between two successive generated packets, and B is the packet size (see Table 3 for parameters settings). The network throughput is depicted in Figure 4 as a function of the number of nodes, and demonstrates a trend similar to that of the PER.

2.2.3 End-to-End Delay

We define the End-to-End (E2E) Delay as the interval of time between the generation of the packet at the network server to be sent in the downlink to a given node, and the instant when the network server receives a reply from the node. Tests and simulations to derive the average End-to-End Delay have been carried out, accounting for the fact that this delay strongly depends on the operating class used by the ED.

In the case of Class C devices, that is assuming the node are always on, the E2E Delay

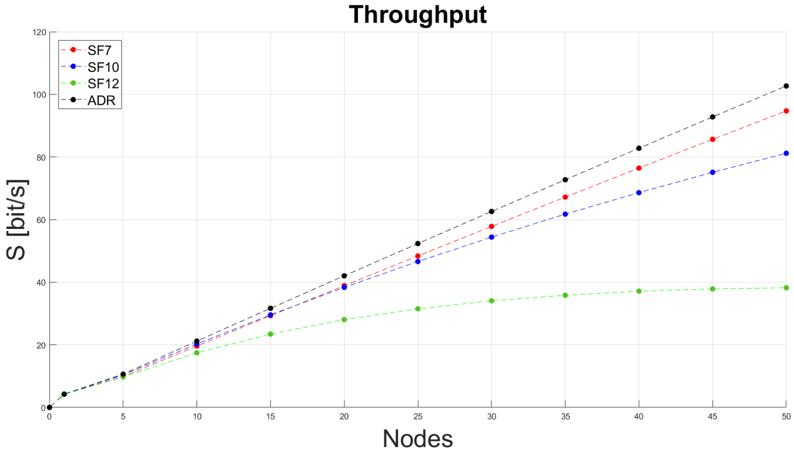


Figure 4: Throughput

is given by:

$$E2E_{Delay} = \tau_{NS-GW} + ToA_{DL} + T_{proc}^{(node)} + ToA_{UL} + \tau_{GW-NS} + T_{proc}^{NS} \quad [s] \qquad (1)$$

where $\tau_{GW-NS} = \tau_{NS-GW}$ is the propagation time from gateway to the network server and viceversa; ToA_{DL} and ToA_{UL} are the ToA of the packets transmitted in downlink and uplink, respectively (they depend on the SF set); $T_{proc}^{(node)}$ is the processing time at the ED and T_{proc}^{NS} is the processing time at the network server.

Since in class A the receive window is opened only after an uplink message, the server is able to send a downlink message (which contains the request) only after the correct transmission of the ED. This means that if for some reason an uplink packet is lost, the network server will not know that the receive window of the device is opened and therefore it will not send the downlink message, waiting for the next uplink. Therefore, in the case of Class A devices the packet to be sent in downlink remains at the network server for a certain amount of time, denoted as $T_{wait-NS}$, which depends on the frequency with which the ED generates packets in uplink and on the probability that this packet is sent with success. Therefore, since in our simulations (and also experiments) EDs generate an uplink packet every T, the E2E Delay for the case of Class A is given by:

$$\begin{aligned} E2E_{Delay} \quad = \quad & T_{wait-NS} + T_{RXwind} + ToA_{UL} + ToA_{DL} + T_{proc}^{(node)} + ToA_{UL} + \\ & \tau_{GW-NS} + T_{proc}^{NS} \quad [s] \end{aligned} \qquad (2)$$

where $T_{wait-NS} = \frac{T}{2} + T \cdot PER$ is the average waiting time of the packet at the network server, given that nodes generate packets in uplink every T and these packets have a probability of being correctly received given by $(1 - PER)$. T_{RXwind} is the interval between the end of uplink transmission and the beginning of the first receive window opened by the ED. The other terms are the same considered for Class C. Results are provided in Figure 5 for Class A and in Table 4 for Class C. Note that in the case of Class

C the network server can send downlink messages almost at any time; therefore there is no variation due to the number of nodes, because delay does not depend on the PER. This deeply reflects the user perception of the system and it appears clearly that Class A LoRaWAN is not thought for real-time application.

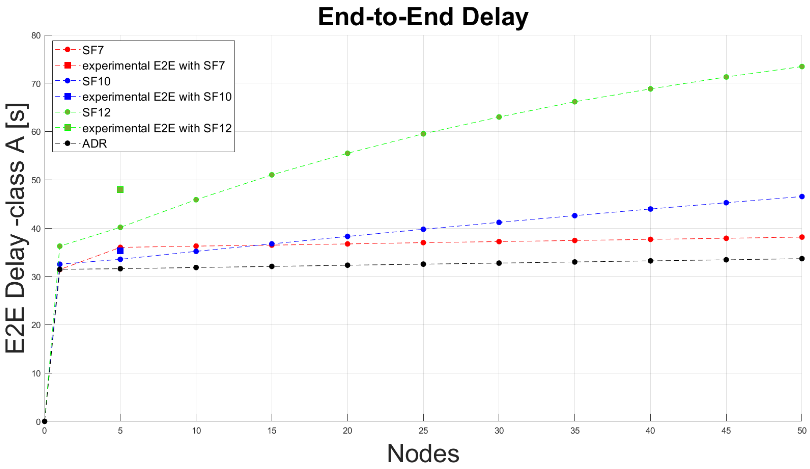


Figure 5: End-to-end delay class A

SF7	SF 10	SF12	ADR
0,38 s	0,9 s	2,79 s	0,38 s

Table 4: End-to-End Delay class C

3 NB-IoT Technology: main features and characteristics

3.1 NB-IoT technology

3.1.1 Overview

NB-IoT is designed to achieve efficient communication in the cellular IoT framework and reach a longer battery life for a massive distribution of nodes. Three key elements characterize it: low cost, a large number of connections per cell and robust coverage, with very good penetration in underground and indoor environments [6]. NB-IoT is introduced in Release 13 (Rel. 13) of 3GPP, emerging as an alternative solution to the LPWA technologies already present on the market (e.g., LoRaWAN). NB-IoT leverages on the LTE standard and numerology, but it is designed for ultra-low-cost Machine Type Communications (MTC), supporting a massive number of devices per cell. From LTE it takes the synchronization, radio access, resources definition and assignment. The standard

Table 5: NB-IoT Key Parameters Values

Parameter	Value	Comment
Bit Rate	up to 253.6 kbit/s	UE capabilities and netw. config.
Frequency Bands	various in [400,2700] MHz	TDD only in band 2490-2690 MHz
Bandwidth	180 kHz	200 kHz in re-farmed GSM
Topology	star of stars	similar to LTE
Link budget	up to 164 dB	netw. config.
TX Range	up to 35 km	expected to reach 100 km [16]
Consumption	TX: 230 mA at 23 dBm RX: 61 mA	[15]

allows modifications to regular LTE by enhancing the link budget and reducing the energy consumption, complexity and costs to a minimum.

While the other cellular systems for MTC are based on existing radio access technologies, NB-IoT can either operate in a stand-alone mode, within the guard bands of LTE carriers or within LTE carriers. It supports a nominal system bandwidth of 180 kHz (equal to the one of an LTE Physical Resource Block (PRB)) in both uplink and downlink. The (narrowband) channel spacing is 15 kHz as in LTE, but it can be decreased to 3.75 kHz in uplink communications [7]. Traditionally, in Rel. 13 and Rel. 14 the NB-IoT was limited to frequency-division duplexing (FDD) operation implying the use of different frequency bands for uplink and downlink transmissions. However, in Rel. 15 (2019) a new option - the time-division duplexing (TDD) - has been introduced allowing to use the same frequency band both for uplink and downlink.

As in LTE, NB-IoT eNBs (enhanced Node-B) employ Orthogonal Frequency Division Multiple Access (OFDMA) in the downlink, and the User Equipments (UEs, the term used in LTE to denote an end node or a user terminal) use Single Carrier Frequency Division Multiple Access (SC-FDMA) in the uplink. However, the modulation schemes are limited to Binary Phase Shift Keying (BPSK) and Quadrature Phase Shift Keying (QPSK) to reduce complexity and ensure a better link budget. A single process Hybrid Automatic Repeat Request (HARQ) is expected in both uplink and downlink by default (this requirement was relaxed in Rel. 14), and the half-duplex operation is allowed. NB-IoT UEs (cat NB1/NB2) implement power control in the uplink, in order to keep low power and consumption where possible.

The expected Coverage Enhancement (CE) is mainly achieved by allowing repetitions (i.e., temporal diversity [19]). The signalling for control information and data is repeated a number of times in different uplink and downlink channels. Each replica has a different coding, and multiple replicas can be combined at the receiver to increase the reception probability.

NB-IoT also introduces a UE categorization in several classes of devices, based on measured power levels. It allows an energy-efficient operation, though keeping an ultra-low device complexity. To further reduce costs, the device searches for only one synchronization sequence and can use a low sampling rate (e.g., 240 kHz) to establish primary time and frequency synchronization to the network. Also, the maximum transport block size is 680 bits/1000 bits in downlink and uplink in Rel. 13 (in Rel. 14 both were increased to 2536 bits) and a single transmit-receive antenna can guarantee the performance objectives of NB-IoT.

Techniques like Power Saving Mode (PSM) and extended Discontinuous Reception (eDRX) are used to increase the battery life for cellular IoT devices. Energy consumption critically depends on the device behaviour when it is not on an active session: these idle time intervals for cellular networks are used to monitor paging and perform mobility measurements. For this reason, PSM and eDRX support a reduced energy consumption by extending the periodicity of paging occasions or requiring no monitoring at all.

3.1.2 Protocol operation

To get a better understanding of the NB-IoT technology, we detail the operation of a UE operating in an FDD-based network, using different frequency resources for uplink and downlink [19]. In case of uplink, the resource grid is composed of multiple subcarrier frequencies with a step (the so-called frequency separation - Δf) of either 3.75 kHz or 15 kHz, and time slots with a duration of 0.5 ms and 2 ms in case of Δf=15 kHz and Δf=3.75 kHz, respectively. On top of this, NB-IoT introduces the notation of a resource unit (RU), denoting a combination of a specific number of consecutive subcarriers (i.e., 1,3,6 or 12) and a number of time-domain slots. The RU represents the minimum element, which can be allocated to a UE for an uplink data transmission. In the case of downlink, the frequency separation is fixed at 15 kHz, and the concept of PRBs is used. A PRB spans 12 subcarriers over 7 OFDM symbols, and a pair of PRBs is the smallest schedulable unit, which is referred to as a single subframe (thus having the total duration of one millisecond). Ten subframes compose a single frame (of 10 ms), and 1024 frames make a hyperframe (10.24 s).

Once powered up, a UE typically starts the procedure of cell search, which is the procedure by which the UE acquires time and frequency synchronization with a cell and identifies it. For this, the UE enables the receiver and searches first for the narrowband primary synchronization signals (NPSS) which are sent by eNB in every 5th subframe of each frame. Then it proceeds with detecting the narrowband secondary synchronization signals (NSSS) which encode the physical cell identity (PCID) and are sent in 9th subframe of each even frame (the transmission of a complete NSSS sequence takes 4 subframes and thus NSSS are repeated every 80 ms). Once finished, the NB-IoT UE proceeds with acquiring the Master Information Block (MIB-NB, [10]) which has a fixed schedule with a periodicity of 640 ms composed of 8 data blocks, each repeated eight times. The elements of MIB (sent in the so-called Narrowband Physical Broadcast Channel - NPBCH) are transmitted in subframe 0 of every single frame. Once decoded, MIB provides the UE with relevant information about the network deployment mode, timings and the scheduling of the first system information (SI) block (SIB1-NB).

The SIB1-NB uses a fixed schedule with a periodicity of 2560 ms in subframe 4 of every other frame in 16 continuous frames [10]. The starting frame of the SIB1-NB depends on the PCID and is derived by the UE from NSSS, while the configuration for repetitions is specified in MIB-NB. The SIB1-NB provides the UE with the information needed to evaluate whether it is allowed to connect the cell and carries the scheduling information of the other SI blocks. To the "required" SI for UE belong MIB-NB, SIB1-NB, SIB2-NB(radio resource configuration), SIBs 3-5-NB(neighbouring cell-related and cell re-selection information), and SIB22-NB(radio resource configuration on non-anchor carriers). The SI messages are sent on Narrow Band Downlink Shared Channel (NPDSCH)[11].

Once possessing all the required SI, the UE may try to establish the connection to the network. For this, it has to execute the special random access (RA) procedure to gain access to a radio channel. Specifically, the UE waits for a scheduled (scheduling specified in SIB2-NB and is periodic with a period ranging between 40 ms and 2.56 s [10]) RA channel (RACH) window, randomly selects one of the preambles (number of which depends on the number of available carriers from 12 to 48) and transmits it. A preamble is sent using single-tone transmission employing frequency hopping between symbol groups. Three different NPRACH preamble formats are currently defined (formats 0 and 1 introduced in Rel. 13 and format 2 added in Rel. 15), featuring the different trade-offs between the on-air time and maximum communication range, which can, potentially, reach 120 km [12]. The basic NPRACH repetition unit consists of four symbol groups for Formats 0 and 1, or 6 symbol groups for Formats 2, with a special relationship between tone frequencies within a repetition unit. Note, that up to three periodic NPRACH windows can be configured in a cell, each associated with a CE level and characterized by the different number of preamble repetitions (ranging from 1 to 128 [10]). The selection of NPRACH to use is made by the device based on its estimation of the radio signal received power (RSRP) from eNB, the network configurations, and the number of previous unsuccessful RA attempts. The NPRACH transmission is sometimes referred to as Message 1 (Msg1) since this is the first message in RA procedure.

After a RACH window, the eNB delivers the scheduling for RA response (RAR or Msg2) in NPDCCH during the Type 2 common search space[11]. The RAR itself is sent in the narrowband physical downlink shared channel (NPDSCH) and allocates the resources and specifies the MCS and the number of repetitions for the next uplink transmission - the radio resource control (RRC) connection request (Msg3) - for the RA preambles it has received. The Msg3 are sent by all the UE which have used the specific RA preamble carrying the unique data identifying the device, i.e., the UE Contention Resolution Identity and which is used to detect the possible collisions. In case of successful Msg3 reception, the eNB replies them with Msg4, i.e., the RRC connection setup, which is also sent in NPDSCH and scheduled through NPDCCH.

Note, that a similar procedure has to be repeated each time an unconnected UE requires to access the radio resources to transmit or receive the data. Note, however, that NB-IoT also supports the contention-free channel access procedure initiated by the eNB, which implies that an eNB dictates a UE it wants to get connected the random access preamble, which no other UE are allowed to use thus ensuring collision avoidance. Following the discussed above RA procedure, the UE and eNB continue exchanging the data sent in NPDSCH in downlink and NPUSCH in uplink having each data transmission scheduled through NPDCCH (the respective procedures are discussed in more details in the following subsection).

Importantly, similarly to LTE, the closing of an active RRC session is handled by the network (i.e., the MME and eNB) and done based on the inactivity timer. In addition to this, the release assistance indication (RAI) procedure has been introduced in Rels. 13 and 14, which allows a UE to signalize the network that the UE has no other data to send and ask for connection release.

Table 6: Uplink and downlink TBS configurations

TBS size (bits) in NPDSCH								
I_{TBS}	Number of subframes (1 ms long)							
0	1	2	3	4	5	6	8	10
1	16	32	56	88	120	152	208	256
2	24	56	88	144	176	208	256	344
3	32	72	144	176	208	256	328	424
4	56	120	208	256	328	408	552	680
5	72	144	224	328	424	504	680	872
6	88	176	256	392	504	600	808	**1032**
7	104	224	328	472	584	**680**	**968**	1224
8	120	256	392	536	680	808	1096	**1352**
9	136	296	456	616	776	936	1256	1544
10	144	328	504	680	872	**1032**	1384	1736
11	*176*	*376*	*584*	*776*	*1000*	*1192*	*1608*	*2024*
12	*208*	*440*	*680*	***904***	*1128*	*1352*	*1800*	*2280*
13	*224*	*488*	*744*	*1032*	*1256*	*1544*	*2024*	*2536*

TBS size (bits) in NPUSCH								
I_{TBS}	N_RU-Number of resource units							
0	1	2	3	4	5	6	8	10
1	16	32	56	88	120	152	208	256
2	24	56	88	144	176	208	256	344
3	32	72	144	176	208	256	328	424
4	56	120	208	256	328	408	552	680
5	72	144	224	328	424	504	680	872
6	88	176	256	392	504	600	808	**1000**
7	104	224	328	472	584	**712**	**1000**	1224
8	120	256	392	536	680	808	1096	**1384**
9	136	296	456	616	776	936	1256	1544
10	144	328	504	680	872	**1000**	1384	1736
11	*176*	*376*	*584*	*776*	*1000*	*1192*	*1608*	*2024*
12	*208*	*440*	*680*	***1000***	*1128*	*1352*	*1800*	*2280*
13	*224*	*488*	*744*	*1032*	*1256*	*1544*	*2024*	*2536*

3.2 NB-IoT KPI

Note, that unless stated otherwise in what follows we imply the NB-IoT operation using frame structure type 1, i.e., the FDD mode. In what follows, we start by discussing the performance of the NB-IoT physical layer and then present the results taking into account the link-layer procedures. Importantly, the results presented below illustrate the NB-IoT performance in different deployment modes and various network and UE configurations, differing with respect to multi-tone support, uplink frequency separation, etc.

3.2.1 Physical layer performance

The peak data rate of NB-IoT at the physical layer is defined by the configurations of the NPDSCH and NPUSCH illustrated in Table 6 .To give an example, for Rel. 13 the 680 bits of downlink data can be sent fastest within three one-millisecond-long subframes, resulting in peak throughput of 226.6 kbit/s. The Rel. 14 has introduced new TBS options (devices implementing these are referred to as class NB2 in contrast to NB1, which denote to devices operating based on Rel. 13), allowing for slightly higher data rates, which can reach 2536bits/10ms=253.6 kbit/s. Even though the TBS allocation tables for NPDSCH and NPUSCH are rather similar, for NPUSCH TBS is allocated in terms of the resource units, duration of which depends on the number of subcarriers and the subcarriers spacing as discussed in [9].

For frame structure type 1 implying FDD operation and NPUSCH format 1, which is used to transfer user data in uplink, in case of 3.75 kHz subcarrier spacing (Δf=3.75 kHz) only single tone transmissions are supported and the maximum I_{TBS} equals 10. Given that the duration of a single resource unit (RU) for Δf=3.75kHz equals 2ms*16slots = 32 ms, the maximum uplink physical layer data rate for Rel. 13 equals 1000 bit/(6*32ms) = 5.208 kbit/s and for Rel. 14 is 1736 bit/(10*32ms) =5.425 kbit/s. The respective values in Table 6 are highlighted with yellow and lime. In the case of 15 kHz subcarrier spacing (Δf=15 kHz) an eNB may assign to the 1,3,6 or 12 sequential tones resulting in the durations of a single resource unit becoming equal to 8, 4, 2, or 1 ms, respectively. Therefore, the maximum uplink throughput (the respective TBS configuration in Table 6 is highlighted with turquoise) for Rel. 13 is 1000 bit/(6*8ms)=20.833 kbit/s, 1000 bit/(4*4ms)=62.5 kbit/s, 1000 bit/(4*2ms)=125 kbit and 1000 bit/(4*1ms)=250 kbit/s for single, 3, 6, and 12 tone transmissions, respectively. In case of Rel. 14 the numbers (TBS configuration highlighted with magenta) are 1736 bits/(10*8ms)=21.275 kbit/s, 2536 bits/(10*4ms)=63.4 kbit/s, 2536 bits/(10*2ms)=126.8 kbit and 2536 bits/(10*1ms)= 253.6 kbit/s for single, 3, 6, and 12 tone transmissions, respectively.

Note, that all the calculations above imply that the minimum number of repetitions configured in the network is one and that the condition of the radio channel between a UE and an eNB is sufficiently good. Otherwise, e.g., for the UE located close to the cell edge or experiencing hard radio signal propagation conditions (e.g., a sensor device in the basement of a building) – a UE may be instructed by the eNB (within the Downlink Control Information (DCI) packet assigning the uplink/downlink resources) to repeat the transmission multiple times. The possible options for downlink (NPDSCH) and uplink (NPUSCH) are listed in Table 7. If repetitions are used, the maximum physical layer data rate decreases proportionally to the increase of the number of repetitions.

However, the discussion above does not account for the protocol-layer features and procedures, which affect directly the throughput experienced by the applications.

3.2.2 Performance of NB-IoT medium access protocol

Note, that for the following discussion, we imply that the RRC session between the UE and an eNB has been already established. First, we consider the uplink transmission scenario, which is illustrated with the respective timings in Fig. 6.

Before sending the actual data in the uplink, a UE has to receive in NPDCCH the DCI of format N0, which carries the information about the resources allocated for NPUSCH,

Table 7: Repetitions in NPDSCH and NPUSCH

Repetitions in NPDSCH																
I_{Rep}	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
N_{Rep}	1	2	4	8	16	32	64	128	192	256	384	512	768	1024	1536	2048
Repetitions in NPUSCH																
I_{Rep}	0	1	2	3	4	5	6	7								
N_{Rep}	1	2	4	8	16	32	64	128								

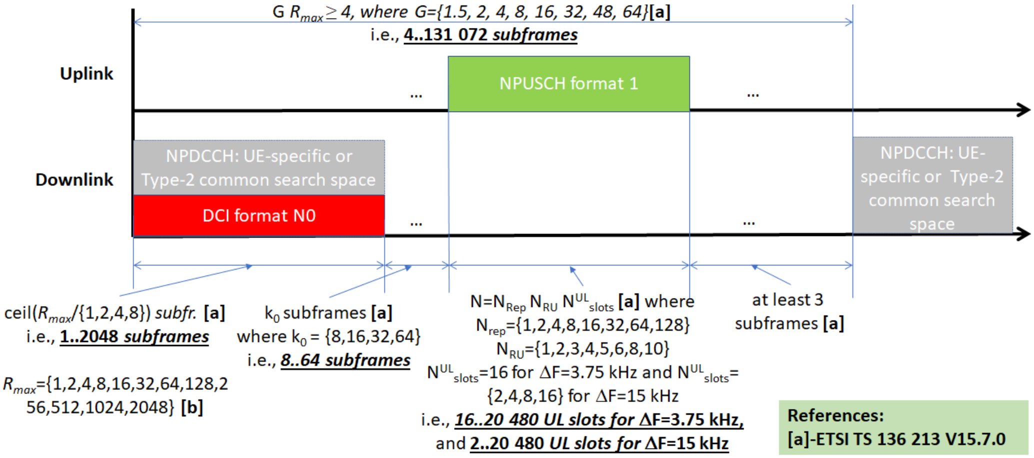
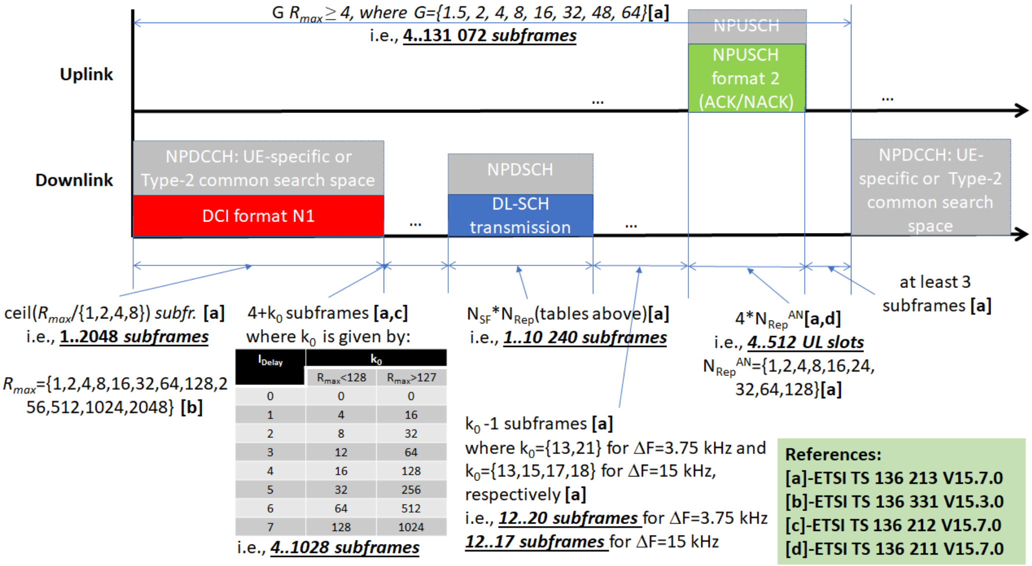

G R_{max}≥ 4, where G={1.5, 2, 4, 8, 16, 32, 48, 64}[a]
i.e., 4..131 072 subframes
Uplink
...
NPUSCH format 1
...
Downlink
NPDCCH: UE-specific or Type-2 common search space
DCI format N0
...
...
NPDCCH: UE-specific or Type-2 common search space
ceil(R_{max}/{1,2,4,8}) subfr. [a]
i.e., 1..2048 subframes
R_{max}={1,2,4,8,16,32,64,128,256,512,1024,2048} [b]
k_0 subframes [a]
where k_0 = {8,16,32,64}
i.e., 8..64 subframes
N=N_{Rep} N_{RU} N^{UL}_{slots} [a] where
N_{rep}={1,2,4,8,16,32,64,128}
N_{RU}={1,2,3,4,5,6,8,10}
N^{UL}_{slots}=16 for ΔF=3.75 kHz and N^{UL}_{slots}={2,4,8,16} for ΔF=15 kHz
i.e., 16..20 480 UL slots for ΔF=3.75 kHz,
and 2..20 480 UL slots for ΔF=15 kHz
at least 3
subframes [a]
References:
[a]-ETSI TS 136 213 V15.7.0

Figure 6: Uplink transmission in NB-IoT and the respective timings

the MSC and the number of repetitions to be used, as well as contains a flag playing a role of a negative acknowledgement and indicating the need of repeating the previous NPUSCH transmission. Depending on the channel conditions the DCI message itself may be repeated multiple (up to 2048) times. Note, that the time windows when an eNB may send a DCI packet, which are referred to in the protocol as "search spaces" are limited and happen periodically, with period depending on the configuration of the network. The minimum period for search space is four subframes (remind, that duration of one subframe equals to 1 ms), but since this leaves not many resources for actual data transfers, in practice the value of the period can be much bigger. After receiving the DCI N0 and before commencing the NPUSCH transmission the UE has to wait for 8 to 64 ms, as specified by the eNB. The NPUSCH transmission itself has been discussed in the previous subsection, and its duration depends on the number of repetitions, the TBS, the number of subcarriers and subcarrier spacing. Following the NPUSCH transmission, the UE waits for an NPDCCH search space, in which the eNB may request (by sending another DCI N0) the repetition of NPUSCH in case it was not received or wants to proceed with the transmission of new data. Note, that the protocol prescribes to have at least a 3-subframe gap between the end of NPUSCH transmission and the start of the next DCI message to this device.

Considering all the implications discussed above, to every NB-IoT UE's uplink transmission, there is an associated signalling overhead of at least 11 subframes for NPDCCH and guard time intervals. Given this, the practical uplink throughput drops up to twice. Specifically, for 12-tone transmission Rel. 13 and Rel. 14 NB-IoT UE can achieve the throughput of 62.5 kbit/s and 115,27 kbit/s, respectively. For single tone and frequency separation of Δf=15 kHz the throughput peaks at 16.67 kbit/s and 18.87 kbit/s for Rel. 13 and Rel. 14, respectively. Finally, for Δf=3.75 kHz due to long uplink resource unit duration the maximum possible throughput does not change significantly, staying at 4.9 kbit/s and 5.22 kbit/s for Rel. 13 and Rel. 14, respectively.

When this comes to the latency, the minimum one in case of uplink data equals the actual on-air time and can be as small as one subframe duration, i.e., 0.5 ms (up to 224-bit TBS with no repetitions in case of a multitone). However, this implies that the eNB has to know exactly when the UE will have data to be transmitted and provide resources to such transmission in advance. This situation is hardly realistic unless the UE traffic is strictly periodic.


G $R_{max}\geq 4$, where G={1.5, 2, 4, 8, 16, 32, 48, 64}[a]
i.e., 4..131 072 subframes
Uplink
Downlink
NPUSCH
NPUSCH format 2 (ACK/NACK)
...
NPDCCH: UE-specific or Type-2 common search space
DCI format N1
NPDSCH
DL-SCH transmission
NPDCCH: UE-specific or Type-2 common search space
ceil(R_{max}/{1,2,4,8}) subfr. [a]
i.e., 1..2048 subframes
R_{max}={1,2,4,8,16,32,64,128,256,512,1024,2048} [b]
4+k_0 subframes [a,c]
where k_0 is given by:
i.e., 4..1028 subframes
N_{SF}*N_{Rep}(tables above)[a]
i.e., 1..10 240 subframes
k_0-1 subframes [a]
where k_0={13,21} for ΔF=3.75 kHz and k_0={13,15,17,18} for ΔF=15 kHz, respectively [a]
i.e., 12..20 subframes for ΔF=3.75 kHz
12..17 subframes for ΔF=15 kHz
4*N_{Rep}^{AN}[a,d]
i.e., 4..512 UL slots
N_{Rep}^{AN}={1,2,4,8,16,24,32,64,128}[a]
at least 3 subframes [a]

I_{Delay}	k_0	
	R_{max}<128	R_{max}>127
0	0	0
1	4	16
2	8	32
3	12	64
4	16	128
5	32	256
6	64	512
7	128	1024

References:
[a]-ETSI TS 136 213 V15.7.0
[b]-ETSI TS 136 331 V15.3.0
[c]-ETSI TS 136 212 V15.7.0
[d]-ETSI TS 136 211 V15.7.0

Figure 7: Downlink transmission in NB-IoT and the respective timings

The phases of NB-IoT downlink transmission and their respective durations are illustrated in Fig. 7. Similarly to uplink transmission, the downlink transmission starts with eNB sending a DCI message within the search space time window. Note, that when arranging a data transfer in the downlink, the format of the DCI message differs from the one used for scheduling an uplink. Specifically, the DCI format N1 message, in addition to the information on the resources and MCS to be used for transmitting the downlink, includes the scheduling information for the uplink acknowledgement (ACK) message to be transmitted by the UE following the downlink. As one can see from Fig. 7, the minimum time gap from the end of DCI message to start of downlink data depends both on the number of downlink repetitions and a scheduling delay and ranges from 4 to 1028 downlink subframe length (equal to 1 ms). The time gap between the downlink transmission and the following ACK transmission depends on both the scheduling and the frequency separation used for uplink resource grid, with the minimum duration of 12 downlink subframe lengths. The ACK is sent in NPUSCH using the special uplink frame type (i.e., the NPUSCH format 2). The size of the frame is fixed, and the number of repetitions is one of the network-specific configuration parameters. Similarly to the uplink, the protocol prescribes to have at least a 3-subframe gap between the end of NPUSCH transmission and the start of the next DCI message.

Considering all these, the maximum feasible downlink throughput for inband deployment in Rel. 13 for the case of uplink frequency separation of Δf=3.75 kHz is 21.25 kbit/s, and for Δf=15 kHz is 26.15 kbit/s. For Rel. 14, the respective numbers are 45.68 kbit/s and 54.25 kbit/s, respectively. For standalone deployment for Rel. 13 and uplink frequency separation of Δf=3.75 kHz the maximum downlink throughput is 21.93 kbit/s and for Δf=15 kHz is 27.2 kbit/s. For Rel. 14 these numbers increase to 66.74 kbit/s and 79.25 kbit/s, respectively.

The minimum latency for downlink transmission is defined by the duration of the DCI, the gap between DCI and NPDSCH transmission and the duration of NPDSCH. As this can be seen from Fig. 7, the cumulative duration of these three phases is six subframes – i.e., 6 ms.

Note that the calculations above do not account for the signalling overhead due to the scheduled RACH windows in the uplink, or transmission of synchronization signals and SI in the downlink. These may introduce additional delays, thus reducing the throughput.

3.3 Important mechanisms

Since its initial introduction in Rel. 13 NB-IoT technology has significantly evolved, having a set of new (often optional) functionalities introduced. Since these modifications have the potential to affect the KPIs, in what follows we briefly discuss some of them.

3.3.1 Two HARQ processes

In Rel. 13 the NB-IoT UE were restricted to have only a single HARQ process both with respect to uplink and downlink. As a result, a UE had to wait for the previous block to be acknowledged before sending/receiving the next one. This, due to the signalling overhead and various gaps, has drastically limited the maximum throughput. This limitation has been relaxed in Rel. 14 introducing for the NB2 devices the optional support of two HARQ processes [13, 12]. In essence, the support of the second HARQ process allows a device to send or receive the second block of data even before the first one is acknowledged, thus increasing the maximum data rate. Depending on the configuration, this can bring up to 50% improvement for the throughput.

3.3.2 Early data transmission (EDT)

The need of using RA procedure and establishing an RRC to send the data in uplink brings with it substantial overheads for both the data delivery time and the energy consumption, which become especially notable in case if the amount of data is small. To address this issue, the EDT mechanism has been introduced as a part of Rel. 15. This mechanism allows to integrate up to 1000 bits of data [13] into Msg3 and have them acknowledged in the following message, without even establishing the connection. Note, that for EDT special RACH windows are defined, different from the ones used for conventional RRC connection establishment.

3.3.3 Other mechanisms

Among other notable mechanisms affecting the NB-IoT performance are the enablement of unacknowledged mode RLC (Rel. 15) [13], the allocation of the NPUSCH resources for periodic buffer status report (BSR) transmission for connected UE (Rel. 15) [9, 13], introduction of the RLC unacknowledged mode (UM) [14] (Rel. 15), introduction of optional additional SIB1-NB transmissions to facilitate acquisition of SI needed to connect to the network (Rel. 15), etc.

4 Comparing the two technologies

Both LoRaWAN and NB-IoT have enormous potential for the development of many different IoT applications. Smart cities and precision agriculture are among the domains that can benefit more from the adoption of these technologies. Indeed, in most cases, the IoT applications from such domains do not demand high throughput or low latency; their requirements are compatible with the performance offered by either of the two technologies.

However, the comparison between LoRaWAN and NB-IoT technologies, and the identification of their actual strengths and limitations, must take into account regulatory issues and business models, besides technical aspects.

From the regulatory viewpoint, there is a clear difference between the two technologies. NB-IoT can be deployed over existing 4G systems. Only Mobile Network Operators (MNOs) who have a 4G license can offer NB-IOT services. This is both an advantage and a drawback. The good side is that for MNOs, deploying the network is just a technical and investment issue. In many countries all over the world, they have already deployed NB-IoT plug-ins, and there is no other issue in exploiting it from the user viewpoint. On the opposite, LoRaWAN operates on a license-exempt ISM band which is regulated differently from country to country. In Europe, the document providing guidelines for the use of LoRaWAN (and other) technologies is CEPT Recommendation number 70 03. Different national authorities interpret it in various ways. To date, in Italy, it is still not possible to operate a LoRaWAN network on the 868 MHz band, based on current limitations posed by the Ministry for Economic Development. In the rest of Europe, the same frequency band is used by many operators delivering IoT services since some time now. Assuming that this will be solved soon also in Italy, from the user viewpoint this frequency band poses constraints: nodes can not go beyond the one per cent duty cycle boundary. This means that users have to ensure that their devices do not generate data too frequently. In most applications, this is not an issue, but potentially such limitation brings complexity on the shoulders of the user.

The business model behind the two technologies is totally different. As mentioned, NB-IoT services can only be offered by MNOs. As long as they deploy the network, it is publicly available (upon payment of a subscriber fee). On the opposite, anyone in principle could offer LoRaWAN coverage; private deployments may be useful for particular applications (especially in remote locations, which are not attractive to MNOs). LoRaWAN networks might be available for free in some areas, as it happens, e.g., with the Things Network - a community of open source LoRaWAN gateway owners. Large LoRaWAN networks are deployed in Italy by some companies for smart city applications. As long as they will be allowed to operate commercially, they will offer subscription-based services in large cities like Milano and others.

Finally, the technical side. Numerous differences characterise the two technologies.

- Latencies. The two systems offer comparable performance in the uplink, with latencies of up to about two seconds. In the downlink, however (e.g. for sending commands to actuators), the two options are quite different. NB-IoT has smaller latency than in the uplink, while the LoRaWAN protocol requires transmission on the uplink first, to piggyback packets in the downlink acknowledgements; therefore, downlink latency can be very large, depending on the uplink transmission rate of the device.

- Throughput. As mentioned above, NB-IoT can offer throughput in the order of some (or tens of) kbit/s. On the opposite, a LoRaWAN device has maximum throughout severely limited by the duty cycle constraint and typically close to few tens of bit/s.

- Security and identification. LoRaWAN with ABP is insecure. On the contrary, NB-IoT has advanced security protocols in place.

- Roaming. Sub-GHz ISM bands (normally used by MNOs for NB-IOT wide coverage) are not uniform around the globe, which complicates LoRaWAN trans-ocean roaming. NB-IoT terminals supporting multiple bands can handle this. Recently, the intra-continental roaming solutions for LoRaWAN (allowing to roam between networks deployed in the same bands) have been delivered. However, their widespread adoption is still underway.

- Energy consumption. In addition to the overall consumption, the potential problem for NB-IoT is high peak consumption and the need for a lot of energy during the initial connection with the network. This may make it hard to enable energy-harvesting powered NB-IoT devices. Overall, for non-frequent transmission of small amounts of data, battery duration of a LoRaWAN-powered sensor system can be one order of magnitude larger than for NB-IoT.

- IP support. NB-IoT supports IP, and many off-the-shelf transceivers implement IP-based protocols like TCP/UDP, FTP, HTTP, CoAP, MQTT. This enables seamless integration between NB-IoT and the Internet; whist LoRaWAN requires some form of adaptation layer (most often based on the NS) in between.

- Handover. LoRaWAN networks do not implement any sort of handover mechanism. NB-IoT has to handle it, though this requires additional signalling.

In conclusion, the two technologies differ in many aspects, and both have strengths and weaknesses. Depending on the specific application, the best solution can be identified based on the above considerations. While this is true for all countries of Europe, Italy still suffers from the lack of a vision: while NB-IoT is available, LoRaWAN networks can not be operated commercially yet. This is affecting the development of the digital agenda of the country.

References

[1] C. Gomez, J. Oller, and J. Paradells. *Overview and evaluation of Bluetooth Low Energy: An emerging low-power wireless technology.* Sensors, vol. 12, no. 9, pp. 11 734–11 753, 2012.

[2] M. Siekkinen, M. Hiienkari, J. K. Nurminen, and J. Nieminen. *How Low Energy is bluetooth low energy? Comparative measurements with Zigbee/802.15.4.* Wireless Communications and Networking Conference Workshops (WCNCW), 2012 IEEE. IEEE, 2012, pp. 232–237.

[3] M. Research. *The need for low cost, high reach, wide area connectivity for the internet of things: A mobile network operator's perspective.* White paper, 2014.

[4] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui and T. Watteyne. *Understanding the Limits of LoRaWAN.* in IEEE Communications Magazine, vol. 55, no. 9, pp. 34-40, Sept. 2017.

[5] Claire Goursaud, Jean-Marie Gorce. *Dedicated networks for IoT: PHY / MAC state of the art and challenges.* EAI endorsed transactions on Internet of Things, 2015.

[6] Olof Liberg, Marten Sundberg, Eric Wang Johan, Bergman Joachim Sachs. *Cellular Internet of Things. Technologies, Standards, and Performance.* Elsevier, September 2017.

[7] 3GPP TS 36.300 Group Radio Access Network. *Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN). Overall description.* 3GPP, December 2018.

[8] LoRa Alliance. *LoRaWAN Specification (V1.0.2)*, 2016.

[9] ETSI TS 136 211 V15.7.0

[10] ETSI TS 136 331 V15.3.0

[11] 3GPP TS 36.523-3 V13.1.0

[12] Olof Liberg, Marten Sundberg, Eric Wang Johan, Bergman Joachim Sachs. *Cellular Internet of Things. Technologies, Standards, and Performance.* Elsevier, December 2019.

[13] GSMA, NB-IoT Deployment Guide to Basic Feature set Requirement, June 2019.

[14] ETSI TR 121 915 V15.0.0

[15] Quectel, BC95 Hardware Design, https://www.quectel.com

[16] A. Weissberger, NB-IoT, https://techblog.comsoc.org/category/nb-iot/

[17] Semtech Corporation *LoRaWAN – simple rate adaptation recommended algorithm* Revision 1.0, October 2016

[18] Gianni Pasolini, Chiara Buratti, Luca Feltrin, Flavio Zabini, Roberto Verdone, Oreste Andrisano, Cristina De Castro, *Smart City Pilot Project Using LoRa* European Wireless 2018; 24th European Wireless Conference, Catania, Italy, 2018, pp. 1-6.

[19] L. Feltrin, M. Condoluci, T. Mahmoodi, M. Dohler and R. Verdone, *NB-IoT: Performance Estimation and Optimal Configuration* European Wireless 2018; 24th European Wireless Conference, Catania, Italy, 2018, pp. 1-6.

[20] 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, June 2018.

[21] LoRa Alliance *LoRaWAN 1.1 Regional Parameters*, 2017

LoRaWAN: current status and research directions

Lorenzo Vangelista, Andrea Zanella, Michele Zorzi

Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

Abstract: *Among the different Low-Power Wide Area Network (LPWAN) technologies, Long-Range Wide Area Network (LoRaWAN) stands out for flexibility, performance, and open specifications, characteristics that have been attracting interest from both the industrial and scientific communities. LoRaWAN, indeed, features a number of tuneable network parameters that, in principle, make it possible to better configure the system according to the specific context, thus improving energy efficiency, fairness, and capacity. In this chapter, we discuss the main characteristics of the LoRaWAN technology and present some results that shed light on the effect of different parameter settings in some illustrative scenarios. Furthermore, we illustrate the most recent features introduced by the LoRaWAN specifications and possible future developments of the technology.*

1 Introduction

As stated in [1], the Internet of Things (IoT) paradigm underpins the place-and-play concept, according to which the end devices, i.e., the "things," just need to be placed where they are needed, and are automatically and seamlessly connected to the rest of the (cyber-physical) world. Cellular networks, with their world-wide established footprint, are the ideal candidates to provide such a service, which is indeed the target of the NB-IoT standard. On the other hand, the signaling and control traffic of NB-IoT, although optimized for sporadic machine-type communications, may become the bottleneck of the system [2].

In the meantime, a number of Low Power Wide Area Network (LPWAN) technologies have appeared in the market, in an attempt to fill this gap. Such technologies are characterized by long-range links (in the orders of kilometers), and typically have a simple star-shaped network topology, where the end nodes are directly connected to a gateway that, in turn, provides legacy IP connectivity with other networks or with the public Internet. Moreover, LPWAN implements robust modulations that guarantee excellent energy-efficiency and coverage range, at the cost of very low bitrates. Finally, most LPWANs operate on unlicensed radio bands, thus avoiding the huge royalties to access reserved frequencies, and adopt uncoordinated access schemes, which make it possible to simplify the hardware and reduce the manufacturing costs and the energy consumption.

LoRaWAN is an LPWAN technology that has been gaining a considerable share of the market in the last years, thanks to some interesting features, such as the chirp modulation used at the PHY layer that allows for long-range, robust communications, with low complexity, low power and low cost receivers, and the simple protocol stack. As a consequence, a strong eco-system is growing around this technology, with partners developing different parts of the systems, and integrators selling the complete solutions for both geographical and residential/industrial types of networks.

This chapter is dedicated to the analysis of the LoRaWAN system, offering a broad overview of its main features, from the physical layer up to the application layer. In particular, Sec. 2 discusses the chirp modulation that characterizes the physical layer of the technology and that, being covered by an industrial patent, is not of public domain, and hence could only be inferred by means of reverse engineering. Sec. 3 presents the protocol stack, as described in the open specifications issued by the LoRa Alliance. Furthermore, the section reports the results of some studies presented in the literature, which investigate the performance of the system with different parameter settings, in an attempt to unravel the intertwining of the different elements of the system. In Sec. 4, we describe the network aspects of the standard, dwelling upon the roaming mechanisms and the protocols to interconnect the end devices with the IP-based world. Sec. 5 addresses a possible extension of the application field of the technology in satellite systems, while Sec. 6 concludes the chapter by recalling the key points, and remarking limitations and potentials of the standard.

2 The physical layer

2.1 The LoRa modulation

The LoRa modulation, i.e., the physical layer of LoRaWAN, has never been officially disclosed by Semtech, the company owning the patents on the LoRa modulation and producing the chips implementing it. Nevertheless, several attempts have been made to reverse engineer the LoRa modulation, the first ones appeared only on-line in Internet webpages, such as [3]. To the best of the authors' knowledge, instead, the first paper published in the archived literature was [4], while the first mathematically rigorous description of the LoRa modulation was presented in [5]. The latest and correct[1] description of the LoRa modulation is provided in [6], which we will take as reference for the notation.

Following [6], we denote by SF the so–called Spreading Factor (SF) parameter, which can take any integer value from 7 to 12, by B the bandwidth of the modulated signal, by f_0 the carrier frequency, and by T_s the symbol period, with $T_s = MT_c$ and $T_c = 1/B$. Therefore, for the LoRa M-ary modulation, with $M = 2^{SF}$, in the band $[f_0 - B/2, f_0 + B/2]$, the complex envelope of the transmitted signal in the symbol interval $[0, T_s)$ for a transmitted symbol $a \in \{0, 1, \ldots M-1\}$ is given by:

$$x(t; a, SF) = \exp\left\{\jmath 2\pi Bt\left(\frac{a}{M} - \frac{1}{2} + \frac{Bt}{2M}\right) - u\left(t - \frac{M-a}{B}\right)\right\} \tag{1}$$

where

$$u(t) = \begin{cases} 1\,, & t \geq 0 \\ 0\,, & \text{otherwise.} \end{cases} \tag{2}$$

[1] [5] contains a minor error, i.e., a 1/2 in the quadratic term of the phase.

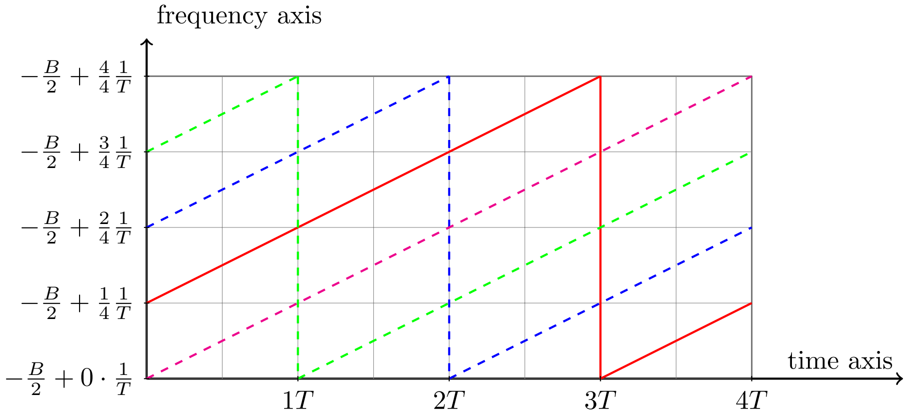


Figure 1: Example of LoRa modulation's instantaneous frequency patterns with SF=2: in magenta the case $a = 0$, in red the case $a = 1$, in blue the case $a = 2$ and eventually in green the case $a = 3 = 2^{SF} - 1$.

A pictorial representation for a simple case (i.e., taking $SF = 2$) of the instantaneous frequency of the LoRa signal for different modulated symbols is shown in Fig. 1. Although the LoRaWAN specifications constrain the Spreading Factor to be in the range 7–12, in Fig. 1 we set $SF = 2$ just to illustrate the concept without making the picture unreadable (with $SF = 7$, we would have had 128 lines). This consideration also applies to Fig. 2.

The complex envelope of the transmitted LoRa signal is then, again with reference to [6],

$$x(t; SF) = \sum_{n=-\infty}^{+\infty} x(t - nT_s; a_n, SF) g_{T_s}(t - nT_s) \; ; \tag{3}$$

where

$$g_{T_s}(t) = \begin{cases} 1\,, & 0 \leq t \leq T_s \\ 0\,, & \text{otherwise.} \end{cases} \tag{4}$$

The LoRa modulation, as defined in (1), exhibits several important properties:

1. it is a *constant envelope* modulation (this is a key enabler for low power consumption in power amplifiers);

2. it is a *continuous phase modulation* i.e., the phase at the beginning and at the end of a symbol is the same;

3. while the sequences $x(kT)$ are orthogonal, *it is not, strictly speaking, an orthogonal modulation* (see [6]);

4. it exhibits *lines in the power spectral density*, although of power almost irrelevant, for $SF \geq 7$, with respect to the power of the continuous part of the power spectral density;

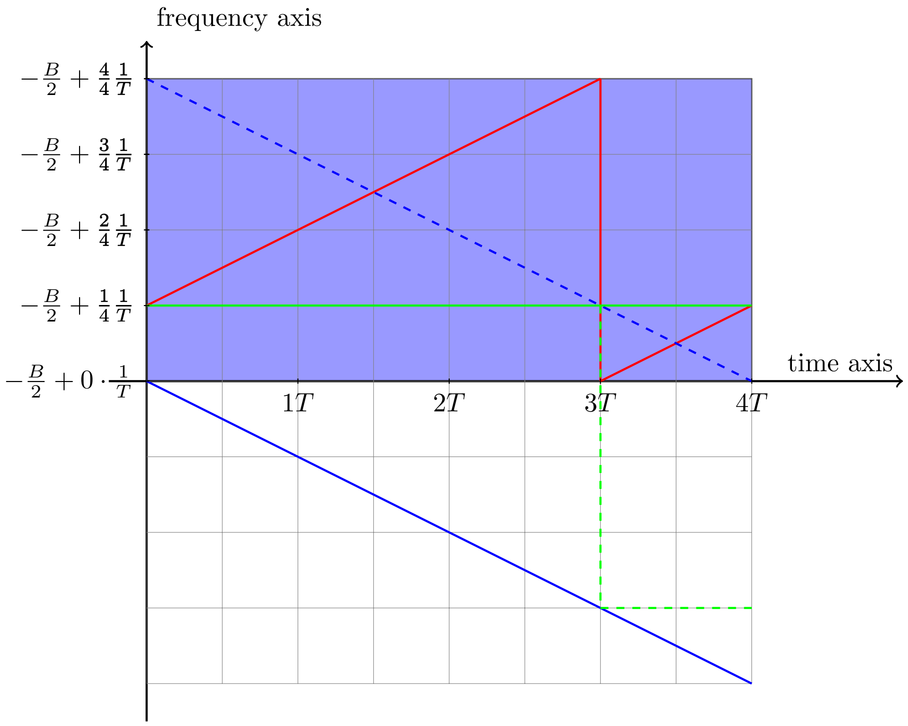

frequency axis
$-\frac{B}{2}+\frac{4}{4}\frac{1}{T}$
$-\frac{B}{2}+\frac{3}{4}\frac{1}{T}$
$-\frac{B}{2}+\frac{2}{4}\frac{1}{T}$
$-\frac{B}{2}+\frac{1}{4}\frac{1}{T}$
$-\frac{B}{2}+0\cdot\frac{1}{T}$
time axis
$1T$
$2T$
$3T$
$4T$

Figure 2: Example of LoRa demodulation's instantaneous frequency patterns for the sampled signal with $SF = 2$ and $a = 1$: in red the instantaneous frequency of the transmitted signal for the case $a = 1$, in solid blue the downchirp, in dashed blue the downchirp folded on top (the frequency domain is periodic) and in green the result of the multiplication.

5. its power spectral density is not limited to the interval $[-B/2, B/2]$, although the power of the energy outside the interval $[-B/2, B/2]$ is negligible for $SF \geq 7$.

The structure of the demodulator for the LoRa modulation is basically made by a sampling of the received signal, followed by a multiplication of the signal by a downchirp

$$e^{\jmath 2\pi \frac{B^2 t^2}{2M}}$$

followed by a Discrete Fourier Transform (DFT). In the case of non-coherent detection, the transmitted symbol a corresponds to the index of the bin with the greatest magnitude in the DFT output, whereas for coherent demodulation it is associated to the index of the bin with the greatest real part. In Fig. 2 a pictorial view of the demodulation in the sampled signal domain is provided.

The computation of the Bit Error Rate for uncoded transmission has been carried out in [7].

2.2 On the orthogonality of LoRa Signals with different Spreading Factors

As we will see in Section 3, in LoRaWAN packets from the different End Nodes (ENs) and from the gateways are multiplexed in time in the same band $[f_0 - B/2, f_0 + B/2]$. The choice of the SF for these packets is determined by the Adaptive Data Rate algorithm (see Section 3) that is applied to each link. Ideally, packets overlapping in time and frequency, but using different SFs, should not interfere, while some mutual interference can be expected when the packets are transmitted with the same SF. In other words, the naive idea which is behind many descriptions (even in the early academic literature) of the LoRaWAN system is that it is made of several parallel and non-interfering channels, one for each SF. Unfortunately, this is not the case, as can be easily demonstrated mathematically.

Although not strictly orthogonal, still transmissions with different SFs generate little mutual interference and are often said to be *quasi–orthogonal*, or it is said that there is *good isolation* between packets with different SFs. The first quantitative results on the isolation of transmissions with different SFs are reported in [4]. A couple of remarks are in order for the results reported in [4]:

1. unfortunately, no description is reported on the methodology used to derive the results;

2. the isolation between packets with the same SF is 6 dB, which means that if two packets with the same SF clash in time and frequency, the strongest one can be correctly demodulated if its received power is at least 6 dB higher than that of the other packet; otherwise both packets cannot be demodulated.

Considerable work has been done in assessing the isolation between the different SFs and – at the time of writing – the state of the art is [8]. In [8] first of all a clear methodology is presented for the software simulations carried out to assess the isolation; furthermore, the simulation results in [8] have been validated with a hardware emulation. A couple of remarks are in order for [8] too:

1. the results differ significantly from those reported in [4] and, as discussed above, they appear to be much more reliable;

2. the results for interfering packets with the same SF show that the isolation is around 0 to 1 dB, which means that, in practice, the packet arriving with stronger power is demodulated correctly. This results into the so–called *capture effect*, which can substantially impact the network level performance.

3 The LoRaWAN protocol stack

One of the most interesting features of LoRaWAN is the simplicity of its architecture and protocol stack [9, 10]. Fig. 3 sketches the protocol stack of the three main elements of the LoRaWAN architecture, namely the End Node (EN), Gateway (GW), and Network Server (NS). The LoRa EN, which is also referred to as End Device (ED), is the most peripheral element of the network, and can be any device (typically equipped with sensing units)

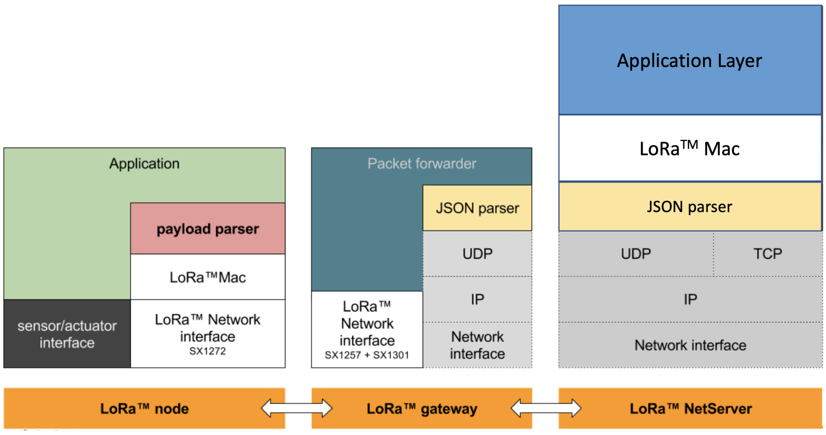


Figure 3: LoRaWAN Protocol Stack.

capable of communicating through the LoRa wireless interface. The LoRa GW provides seamless connection towards the IP legacy technologies, making it possible to exchange data between LoRa nodes and the NS. Finally, the NS is in charge of the management of the whole system: it sets the tuneable parameters, gets the data from the LoRa nodes to the server-side applications and back, sends Acknowledgment (ACK) messages to the peripheral nodes (when required), and so on.

The standard defines three operating modes for the EDs, namely *Class A* (All), *Class B* (Beacon), and *Class C* (Continuous). In Class A mode, the radio interface is activated only to transmit an uplink (UL) data packet. After each transmission, the EN opens two consecutive receive windows during which it can receive downlink (DL) packets from the NS (through one of the GWs). In Class B mode, ENs get time-synchronized by beacons transmitted by the GWs, and wake up periodically to receive possible DL packets. In Class C mode, finally, ENs keep their radio interface always active and can transmit and receive at any time. Clearly, the energy consumption of the nodes is strongly impacted by the operating mode and most battery-powered devices operate in Class A mode.

The ENs can opt for *unconfirmed* transmission, which does not entail any reply from the network, or *confirmed* transmission that, instead, requires an ACK from the NS in one of the two reception opportunities after the transmission. In case of missing ACK, a confirmed packet is re-transmitted for at most $m-1$ additional times, where the setting of the parameter m and of the timing of the retransmissions for the same confirmed message is at the discretion of the EN and may be different for each EN. If still not acknowledged after m transmissions, the packet is generally dropped, although the EN may be programmed to try again with a lower data rate. It is worth noting that the reception windows are opened also for unconfirmed transmissions, in order to receive possible DL packets.

The ENs' transmissions are not addressed to any particular GW, but rather received and forwarded to the NS by all GWs in the coverage range of the transmitter. We note that commercial GWs (e.g., [11]) support multiple parallel reception chains, which make it possible to decode multiple overlapping signals, provided that they use different SFs and have sufficient received powers, as explained in Sec. 2.2. However, the GWs do not support full-duplex transmission, so that, at any given time, they can either transmit or receive.

Duplicate packets are dropped by the NS, which can use side information (e.g., the received signal strength) passed along by the GWs to rank the GWs to be used for possible DL transmission to the EN.

The standard also defines the frequency bands, power, and Duty Cycle (DC) restrictions that apply to the different regions. Tab. 1 shows the configuration mandated for the European region, which entails three default bidirectional channels that shall be implemented in every EN, and a fourth channel reserved for DL transmissions only. We note that, according to the ETSI EN300.220 regulations, the first three channels share the same DC limitation. Indeed, the regulations limit to 36 seconds per hour the total transmission time of a device transmitting in the frequency band from 868.0 MHz to 868.6 MHz, which corresponds to a DC limit of 1%. We remark that this limit applies to the *aggregate* transmission time in this subband, i.e., the DC constraint is shared among the three channels. Conversely, the fourth channel falls in a different frequency range, for which the DC limit is relaxed to 10%.

Table 1: Available LoRaWAN channels in Europe (see [12])

Frequency [MHz]	Use	Duty Cycle
868.1	UL/DL	1%, shared
868.3	UL/DL	1%, shared
868.5	UL/DL	1%, shared
869.525	DL	10%, dedicated

3.1 Performance analysis when varying the system parameters

The performance of LoRaWAN networks has been widely studied in the literature, in the last few years [13, 14]. Most studies assume UL-only traffic, focusing on pure monitoring scenarios. Under these assumptions, LoRaWAN is proved to be able to support hundreds of devices, spread over an area of several square kilometres, that generate traffic with inter-packet periods that vary from 30 minutes to 24 hours, for an aggregate traffic of up to 0.8 packet/s, with a reliability of about 95% with a single GW. The capacity of the system can be further extended by deploying multiple GWs and/or adopting smart strategies to assign the SFs to the different ENs [15]. Also, advanced Adaptive Data Rate (ADR) algorithms can further improve the performance [16].

Interestingly, the presence of DL traffic (including ACKs) is a game changer. As observed in some recent studies, indeed, the use of confirmed traffic (which requires ACKs to be returned by the NS) may enhance the data collection capabilities of the network as long as the overall load is light (less than 0.8 packet/s, according to [17]), but can yield significant performance degradation for higher loads, thus reducing the system scalability. Furthermore, it has been observed that, in scenarios with a mixture of unconfirmed (UL-only) and confirmed traffic, the unconfirmed traffic is particularly penalized, since a number of packets are dropped because of interference at the GW, collision with the transmission of DL ACKs (which are usually prioritized over receptions), or unavailability of free receive paths at the GW.

Some of these inefficiencies can be alleviated by appropriately changing some system settings. A first clear advantage is obtained by relaxing the DC constraint at the GW, which may prevent the transmission of ACKs for successfully received UL packets that require confirmation. Note that, while relaxing the DC constraint of a device would violate the regulations, a similar effect can be lawfully obtained by deploying multiple GWs to be used by the NS for DL transmissions in a load-balancing manner, so as to effectively divide the off time required by the regulations among multiple GWs.

As mentioned, another cause of packet losses is the collision of UL and DL transmissions. In particular, a GW may get a DL packet from the NS to be forwarded to an EN, while busy receiving one (or more) UL transmission(s). In this case, the standard behavior requires the GW to interrupt any ongoing reception, in order to transmit the DL packet. This is indeed meaningful, considering that a missed ACK would determine the retransmission of a packet that was already correctly delivered to the NS. On the other hand, giving priority to the ongoing reception, hence deferring the ACK transmission to the second receive window (if available), may potentially improve the system capacity. This option has been investigated in [17], where it has been observed that it may indeed improve the performance of both confirmed and unconfirmed traffic in mixed traffic scenarios. However, this ACK-prioritization option may also lead to congestion in the DL channel, if the amount of confirmed traffic is significant. A possible workaround suggested by the authors in [17] consists in the definition of two classes for the confirmed traffic, namely *strongly-confirmed* and *loosely-confirmed*. The underlying idea is that applications that need an explicit feedback from the NS should send strongly-confirmed packets, while the loosely-confirmed packets can be used whenever the application's goal is to deliver its packets to the NS, so that ACKs are useful to avoid useless retransmissions, but their reception is not actually critical for the application *per se*. Clearly, the transmission of ACKs for strongly-confirmed packets should be prioritized over reception, while the other ACKs could be sent only if the GW is idle.

Yet another approach to mitigate the bottleneck effect due to the DC limitations of the GW is to change the standard setting of the ACK mechanism. According to the current LoRaWAN specifications, whenever the NS sends an ACK in response to a successfully received confirmed packet, the ACK should be either transmitted in the first receive window (RX1), matching the channel and SF of the UL packet, or in the second receive window (RX2), using the channel reserved to DL transmissions and transmitting with the lowest bitrate, corresponding to $SF = 12$, so as to increase the probability of correct reception of the ACK. However, this approach yields longer transmission times of ACKs, which will rapidly consume the DC budget at the GW. A more efficient solution may consist in adopting an ADR scheme also for ACK transmissions. A possible implementation may consist in simply using the same SF of the UL transmission also for the ACKs sent in RX2, or in increasing by a certain factor such an SF to gain robustness to possible asymmetries in the channel gains, and/or in the receiver sensitivities. As observed in [17], this simple solution may actually bring significant performance gains, in particular in combination with the prioritization of the reception over the transmission of ACKs.

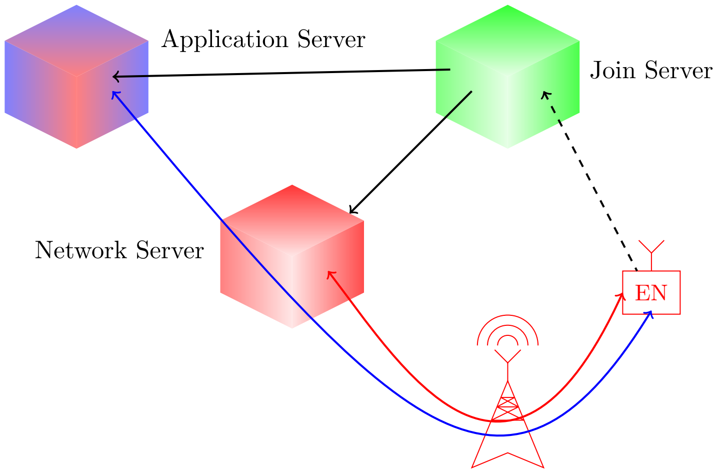


Figure 4: The LoRa network

4 LoRaWAN networking

4.1 LoRaWAN backend

Fig. 4 offers an overview of a LoRaWAN network that, in addition to the three elements already mentioned in Sec. 3, includes two additional servers for security and data delivery (the Join Server and the Application Server). We note that the only elements that belong to the core network of a LoRaWAN operator are the NS and the GWs.

As mentioned, the NS is responsible for the MAC Layer and the radio resource management. In particular, it assigns the SFs to the ENs, manages the band in which the EN should work, and so on, using a control link (shown in red in Fig. 4) encrypted with session keys, which are obtained via the Join Server.

The Application Server is responsible for the services (down to the raw data) associated with an EN, using a data link (shown in blue in Fig. 4), which is also encrypted using session keys provided by the Join Server.

It must be noted that the keys encrypting the data link and the control link are different by design and specific to each EN. As a consequence, an NS cannot access the user data. Furthermore, even if the keys of an EN are compromised, the security of the other ENs would be unaffected.

We should remark that the Join Server is not necessarily owned by the network operator but rather, and more commonly, owned and run by a third party, trusted by both the network operator and the owners of the ENs. Furthermore, different ENs may refer to different Join Servers. The Join Server is actually at the core of the security architecture of the LoRaWAN network as it stores the keys specific to an EN, from which session keys are derived and handed over to the NS and the Application Server. The Join Server can be compared to a certification authority issuing certificates for the usual web.

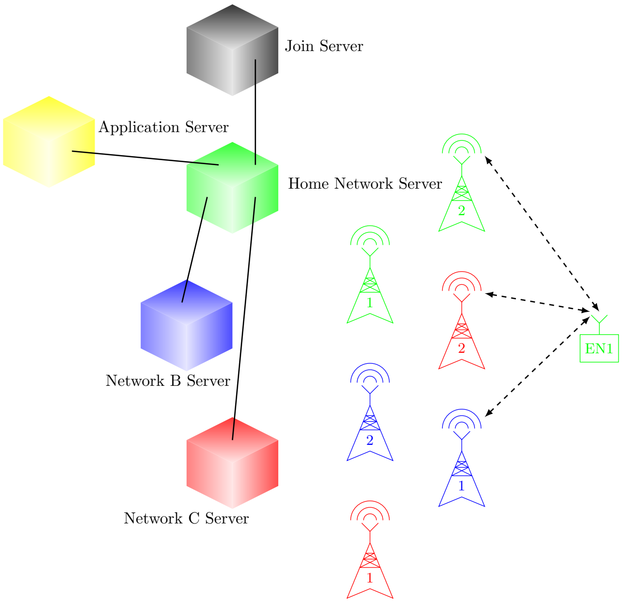


Figure 5: Passive roaming

A final note is about the "master keys" for both the signaling and the user data transfer: the EN manufacturer is responsible for storing them in a secure way in the EN and getting them to a trusted Join Server, once again in a secure way, not using any LoRaWAN network.

4.2 The LoRaWAN global roaming

Roaming in LoRaWAN refers to the possibility for an EN to connect to the *Home Network* (the network which manages the EN, its subscription, billing, etc.) via another network, called *Foreign Network* and run by another operator. Roaming agreements with related charging policies for ENs when in Foreign Networks must be in place for the roaming to work.

There are two types of Roaming in LoRaWAN:

1. Passive Roaming;
2. Active Roaming.

Passive Roaming is illustrated in Fig. 5. In this case we have three networks whose geographical coverage areas are, at least partially, overlapping. In Fig. 5 the Home Network is depicted in green, in all the three components: the NS, the GWs, and a target EN. There are then two Foreign Networks, depicted in red and blue, that are supposed to have roaming agreements with the Home Network. As we can see from Fig. 5, the GWs are covering the same geographical area so that there can be reliable links from the EN to (at least) one GW of any of the three networks. The packet sent by the EN can be collected by both the Home and Foreign Network's GWs and, according to the operational mode for Passive Roaming, is just *relayed*, without taking any further action, to the Home Network Server. The Home Network Server, then, is able to select the version of the packet that is received with the best quality from any of the GWs and uses the selected GW for a possible downlink transmission to the EN and to run the Adaptive Data Rate algorithm.

There are several benefits of Passive Roaming:

- the EN sees a dense network even if its Home Network is not really dense, with benefit for the battery lifetime since most likely the Adaptive Data Rate algorithm will select lower SFs;
- there will be less overall interference in the area, since most of the nodes will get a low SF and a low transmit power (we recall that - although often neglected in academic papers - LoRaWAN supports *power control*);
- the overall Capital Expenditure will be lower, since each operator can install fewer GWs, without compromising the Quality of Service.

It is important to highlight that Passive Roaming is possible in LoRaWAN because the GWs are *stateless* and just relay packets. This is not the case for example for NB-IoT. It is also important to highlight that, in the case of Passive Roaming, the EN is controlled by the Home Network, as far as the Layer 2 protocol is concerned (for example, MAC commands are issued by the Home Network Server).

Active Roaming (available for networks and ENs supporting the version 1.1 and above of the LoRaWAN specifications) is very similar to the usual roaming for cellular wireless networks. In this case, the Layer 2 protocol control is handed over by the Home Network to another network, often called *Visited Network*.

Actually, the two types of roaming can be combined as shown in Fig. 6, where Network B (depicted in blue) represents the Visited Network, while Network C (depicted in red) and Network D (depicted in violet) support Passive Roaming with the Visited Network B. The Layer 2 protocol controller (including the Adaptive Data Rate algorithm and the power control) in this case is the Network Server of Network B, while the Home Network Server (called *Anchor* Network Server) is still in charge of the security and the delivery of the user data to the Application Server.

In conclusion, LoRaWAN provides multiple and flexible roaming modes, enabling different business models. A particularly important business model, which is gaining momentum, corresponds to a large organization that owns the Home Network Server and the Application Server for its devices, which are always in Passive Roaming on other networks. This business model enables such an organization to have full control over its own devices and their specific characteristics without actually entirely owning the physical infrastructure; the full control of the devices through the aforementioned Network Server, and especially the Application Server, allows for the provision of controlled and unified access to the services provided by the ENs of the organization.

4.3 The LoRaWAN IP networking

It is well known that the actual definition of "Internet of Things" involves the connection of the "Things" to the Internet. As a consequence, any communication system for the

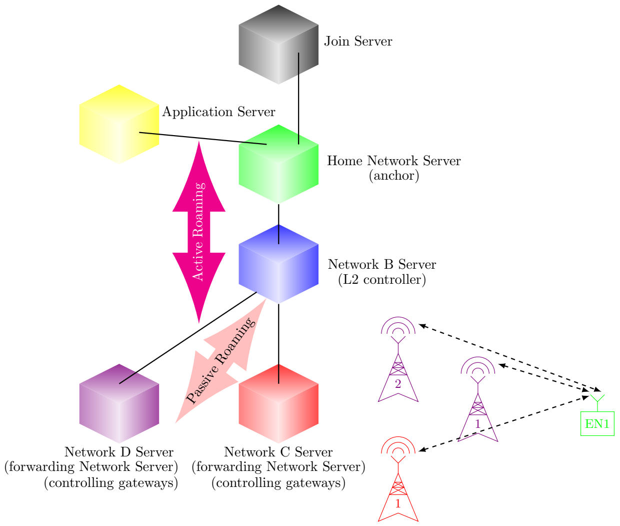


Figure 6: Active and passive roaming

Internet of Things must support the IP (Internet Protocol) down to the “Things.” For the “plain” LoRaWAN architecture, this is actually not true because, in most cases, the data transport from the Application Server is done via the Message Queueing Telemetry Transport (MQTT) protocol [18], in which the node is identified via a “topic” of the MQTT protocol rather than an IP address.

However, the successful work of the IETF “LPWAN Working Group” led to the development of the Static Header Compression Protocol (SCHC, pronounced “chic”), enabling end–to–end IPv6 connectivity for the ENs in a generic Low Power Wide Area Network (LPWAN), such as LoRaWAN. It must be remarked that the SCHC protocol works as well for other LPWANs, such as SigFox and NB-IoT, paving the way for an effective interoperability between, e.g., LoRaWAN and NB-IoT.

SCHC is actually a compression and segmentation protocol and is not the first one developed by IETF, the most important predecessors being ROHC [19] and 6LoWPAN [20]. The need for a new compression protocol stems from the fact that 6LoWPAN relies on some specific mechanisms of the IEEE 802.15.4 standard, which are not supported by the LPWANs, while ROHC relies on a dynamic exchange of information between the source and the destination in order to build the best compression rules and this is again not possible (or too heavy) in LPWANs.

The main characteristic of SCHC is the fact that it is a “static” and stateless protocol. The rules for compression (and decompression) are set *before* the EN enters the network and are stored in the memory of the EN and the NS. It is not possible to add or replace any rule after the EN has been finally commissioned. This enables a strong compression, relying on the fact that the set of messages to and from the ENs is very limited (for example, for a temperature-monitoring sensor, the messages are probably limited to “read the temperature,” “send an alarm in case the temperature is above/below a certain threshold,” and very few others).

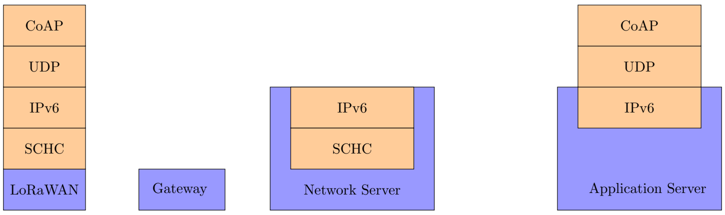


Figure 7: The protocol stacks in the different nodes when using SCHC.

Fig. 7 illustrates a possible architecture for the use of the Constrained Application Protocol (CoAP) in LoRaWAN networks. CoAP is a service layer protocol, designed by the IETF Constrained RESTful Environments Working Group to ease the design of web-based applications in resource-constrained internet devices. As it can be seen from the figure, CoAP relies on the UDP and IPv6 protocols, whose (typically long) headers are compressed by SCHC before transmission over LoRaWAN.

5 Satellite based LoRaWAN networks

In the last few years satellite communications for the Internet of Things has become a hot topic for both industry and academia. In [21], an overview is provided on this topic. Clearly, the interest in such a type of connectivity for the Internet of Things is triggered by the need/opportunity to provide IoT services in areas in which it is not economically or technically viable to provide them via terrestrial networks. Examples include the oceans and the rural areas. LoRaWAN is addressed in [21] as one of the systems capable of providing IoT connectivity via satellite.

The types of satellites are typically Low Earth Orbit (LEO) satellites, whose orbit is about 500-600 km above the earth. LEO constellations for LoRaWAN connectivity are already in a pre–commercialization phase; some satellites have already been launched and the connectivity has been tested by companies like Lacuna Networks, supported by the European Space Agency, and Semtech, the provider of the chips for LoRa ENs and Gateways. Companies like Lacuna Networks are taking advantage of the trend to use (relatively) low cost *CubeSat* satellites.

Unfortunately, while IoT satellite networks are undoubtedly interesting from both the commercial and the technical point of view, there are not so many technical details available in the literature. It is clear that the link from the ENs to the satellites must be of the same type (same type of modulation and coding at least) as that of the terrestrial networks. There are multiple reasons for this, including:

1. economical: the market for purely satellite IoT is not large enough to drive down the costs of the components;

2. regulatory: to re-use for the satellite network existing chipsets originally intended for the terrestrial LoRa links, ENs must use the same bands allocated for the terrestrial networks: the EN, as far as the regulatory framework is concerned, basically "pretends" to transmit to a terrestrial infrastructure but the satellites are "smart" enough to be able to collect the packets from these nodes.

In the architecture of these types of networks, the link from the satellites (acting as flying gateways) to the Network Server is based on specific satellite bands and modulations.

The main challenge at the architectural level is how to provide connectivity from the satellites to the terrestrial ENs. As a matter of fact, the ENs are regular terrestrial nodes but the satellite cannot use the bands allocated for terrestrial communications. A clear solution to this problem – to the best of the authors' knowledge – is still lacking or has not yet been disclosed by the companies that are going to offer the LoRaWAN satellite service. Moreover, one needs to take into account that, most likely, the LoRaWAN protocol cannot accommodate the delays which are typical of satellite networks. As a consequence, modifications in the protocol could be needed, which however would require a reconfiguration of the ENs parameters in case of roaming into terrestrial networks. Such a roaming, on the other hand, is likely needed in case of mobile LoRaWAN nodes connected to a satellite network that transit in a densely populated area. As a matter of fact, a satellite beam will likely "illuminate" (i.e., cover) a wide area. In urban environments, such an area may contain thousands of LoRa nodes, and the satellite gateway will receive *all* the packets generated by such nodes (and gateways) in the illuminated area, irrespective of whether they belong to a terrestrial or satellite network. If the number of such nodes is too large, the satellite gateway may be unable to decode the messages coming from the ENs of its own network because of the interference produced by the other nodes in the area. Therefore, these satellite LoRaWAN networks seem to be able to work only when the covered are contain a limited number of LoRaWAN nodes. In the authors' opinion, this problem can be solved only by letting the mobile nodes of the LoRaWAN satellite network roam into the terrestrial network, when needed.

Summarizing the above discussion, one can see that there are conflicting requirements, and it is not clear how they can be solved in LoRaWAN satellite networks (which are a reality), for example:

1. how to implement the link of the satellite networks toward the ENs;

2. how to have a LoRaWAN certified EN (i.e., obeying the LoRaWAN specifications as set by the LoRa Alliance) working in a satellite LoRaWAN network, if one takes into account the delays;

3. how the satellite gateway can survive the huge interference it collects when flying over areas with terrestrial LoRaWAN networks, taking into account that the satellite beam will most likely collect the transmissions of all the terrestrial ENs connected to tens or hundreds of terrestrial gateways for the same area.

These issues, which are mostly at the network/architectural level, compound with the issue of the link physical layer performance. In [22], the authors (one of whom is with Inmarsat, a leading satellite company which invested in Actility, the company with the largest share in the LoRaWAN network server market) call for a redesign of the LoRa demodulator in the satellite, taking into account the impairments coming, e.g., from the Doppler effects. However, the re-design provided in [22] does not take into account some "tricky" details in the LoRa modulation chain such as the data whitener, etc. Without the full cooperation of Semtech, owning the intellectual property of the complete LoRa modulation chain, it looks then very hard to deploy such a receiver in an FPGA to be placed in the satellite (or, as [22] suggests, in a terrestrial site where we suppose that the signal received from the satellite is relayed). On the other hand, relying on commercial off–the–shelf chipsets does not look possible because of the issues highlighted in [22], as it looks hard to expect that a satellite version of the chipset for the gateway is released by Semtech since the market would be too small to justify the investment.

As a conclusion for this section, we can say that, while promising as a concept, LoRaWAN satellite networks pose a number of problems, both on the commercial and on the technical side, which may prevent their adoption in the near future.

6 Conclusions

This chapter has provided a general introduction to the LoRaWAN architecture and protocols, dwelling upon its main features, such as the chirp modulation at the physical layer, the set of tuneable parameters at the MAC layer, and the roaming among networks managed by different operators.

From this analysis, it appears clearly that LoRaWAN, like other LPWAN systems, fills a gap in the arena of wireless communication technologies, providing a very flexible, secure, energy efficient, easy to deploy, and relatively low cost means to connect a multitude of nodes with low traffic demands. Furthermore, the roaming mechanisms can favor the adoption of this technology, by reducing the capital expenditure for its deployment, while improving the coverage and the quality of service provided to the ENs.

Such features have attracted the attention of both the academic and the industry communities, and the recent literature has been enriched by a number of studies aimed at investigating the challenges and the opportunities provided by this interesting technology.

These studies have revealed some subtle interdependencies among the tuneable parameters of the system, whose standard settings may actually result in suboptimal performance for some application scenarios, in particular in the presence of confirmed traffic. Furthermore, when considering applications that require some downlink traffic, the DC constraint of the GWs may become a bottleneck that limits the capacity of the system. This opens the way to the analysis of alternative frequency-sharing mechanisms, in particular based on Listen-Before-Talk (LBT) techniques that can be used in place of the DC regulations. Such techniques are typically considered for short-range wireless technologies, but are deemed unsuitable for LPWAN systems, which have coverage ranges in the order of several kilometers and a star topology, conditions that exacerbate mutual interference and hidden node problems. Nonetheless, some recent studies have proved that the smart combination of LBT, DC, and rate-adaptation techniques can effectively enlarge the LoRaWAN capacity [23, 24].

In a multi-GW scenario, we can expect better performance for both UL and DL traffic, provided that each EN is covered by more than one GW and the NS wisely distributes the load among such GWs, thus alleviating the DC constraint. On the other hand, the degrees of freedom in the system configuration will also considerably increase with the number of GWs, making the setting of the system parameters more difficult.

These results motivate the development of accurate mathematical models of the system that can capture such interdependencies and provide accurate performance estimates, so as to enable a more accurate and informed network planning.

As a final consideration, we observe that the adoption of LPWAN technologies, including LoRaWAN, is slowed down by the wariness of the stakeholders in betting on one specific technology as the possible winner in the race to conquer the market. In this respect, the SCHC protocol, facilitating the interconnection among the different LPWAN technologies, can act as a catalyst for the massive deployment of such systems.

References

[1] L. Vangelista, A. Zanella, and M. Zorzi, "Long-range IoT technologies: The dawn of LoRa$^{\mathrm{TM}}$," in *Future access enablers of ubiquitous and intelligent infrastructures*. Springer, 2015, pp. 51–58.

[2] A. Biral, M. Centenaro, A. Zanella, L. Vangelista, and M. Zorzi, "The challenges of M2M massive access in wireless cellular networks," *Digital Communications and Networks*, vol. 1, no. 1, pp. 1–19, Feb. 2015.

[3] Bertrik Sikken, "Decoding LoRa," https://revspace.nl/DecodingLora, online; accessed May 20, 2020.

[4] 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, pp. 1–11, Oct. 2015.

[5] L. Vangelista, "Frequency shift chirp modulation: the LoRa modulation," *IEEE Signal Processing Letters*, vol. 24, no. 12, pp. 1818–1821, Dec. 2017.

[6] M. Chiani and A. Elzanaty, "On the LoRa modulation for IoT: Waveform properties and spectral analysis," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8463–8470, May 2019.

[7] T. Elshabrawy and J. Robert, "Closed-form approximation of LoRa modulation BER performance," *IEEE Communications Letters*, vol. 22, no. 9, pp. 1778–1781, Jun. 2018.

[8] 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, Jan. 2018.

[9] M. A. Ertürk, M. A. Aydın, M. T. Büyükakkaşlar, and H. Evirgen, "A survey on LoRaWAN architecture, protocol and technologies," *Future Internet*, vol. 11, no. 10, pp. 1–34, Oct. 2019.

[10] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi, "Long-range communications in unlicensed bands: The rising stars in the IoT and smart city scenarios," *IEEE Wireless Communications*, vol. 23, no. 5, pp. 60–67, Nov. 2016.

[11] Semtech Corporation, *SX1301 datasheet*, June 2014.

[12] "LoRaWAN Regional parameters v1.1, LoRa Alliance," https://lora-alliance.org/resource-hub/lorawanr-specification-v11, online; accessed May 20, 2020.

[13] M. Centenaro, L. Vangelista, and R. Kohno, "On the impact of downlink feedback on LoRa performance," in *IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC).* IEEE, 2017, pp. 1–6.

[14] V. Di Vincenzo, M. Heusse, and B. Tourancheau, "Improving downlink scalability in LoRaWAN," in *IEEE International Conference on Communications (ICC).* IEEE, 2019, pp. 1–7.

[15] 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, Jan. 2018.

[16] N. Benkahla, H. Tounsi, S. Ye-Qiong, and M. Frikha, "Enhanced ADR for LoRaWAN networks with mobility," in *15th International Wireless Communications & Mobile Computing Conference (IWCMC).* IEEE, 2019, pp. 1–6.

[17] D. Magrin, M. Capuzzo, and A. Zanella, "A thorough study of LoRaWAN performance under different parameter settings," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 116–127, Jan. 2020.

[18] "ISO/IEC 20922:2016 [ISO/IEC 20922:2016] information technology — message queuing telemetry transport (MQTT) v3.1.1," https://www.iso.org/standard/69466.html, online; accessed May 20, 2020.

[19] https://datatracker.ietf.org/wg/rohc/charter/, online; accessed May 20, 2020.

[20] C. Bormann, Z. Shelby, "6LoWPAN: the wireless embedded Internet," *John Wiley & Sons*, vol. 43, 2011.

[21] Z. Qu, G. Zhang, H. Cao, and J. Xie, "LEO satellite constellation for Internet of Things," *IEEE Access*, vol. 5, pp. 18 391–18 401, Aug. 2017.

[22] G. Colavolpe, T. Foggi, M. Ricciulli, Y. Zanettini, and J.-P. Mediano-Alameda, "Reception of LoRa signals from LEO satellites," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 6, pp. 3587–3602, Apr. 2019.

[23] J. Ortín, M. Cesana, and A. Redondi, "Augmenting LoRaWAN performance with listen before talk," *IEEE Transactions on Wireless Communications*, vol. 18, no. 6, pp. 3113–3128, Apr. 2019.

[24] D. Zucchetto and A. Zanella, "Uncoordinated Access Schemes for the IoT: Approaches, Regulations, and Performance," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 48–54, Sep. 2017.

LoRaWAN: a Deep Dive in a Large Scale Deployment and in Radio Access Optimization Strategies

Giuseppe Bianchi[1,2], Francesca Cuomo[1,3], Domenico Garlisi[1,4], Patrizio Pisani[5] and Ilenia Tinnirello[1,4]

[1]Consorzio Nazionale Interuniversitario per le Telecomunicazioni (CNIT)

[2]University of Roma ”Tor Vergata”, Italy

[3]University of Rome ”La Sapienza”, Italy

[4]University of Palermo, Italy

[5]UNIDATA S.P.A., Italy

Abstract: *Low-Power Wide Area Networks (LPWANs) represent one key enabling technology for the Internet of Things (IoT). LPWAN capabilities are long range, deep penetration, and ultra-low power consumption. Among the LPWAN emerging technologies, LoRaWAN (Long Range Wide Area Network) is a successful network infrastructure for ultra-low power device communication based on Long Range (LoRa) modulation, patented by Semtech. LoRa exploits free ISM bands and has been conceived for low power consumption (life device up to 10 years) and low data rate applications. LoRaWAN guaranteed easy deployment, indeed a single gateway can cover up to several kilometers. In this chapter, we present and discuss the lessons learned from a real-world LoRaWAN deployment carried out by the UNIDATA regional operator in a large terrestrial area. After presenting the basic LoRa modulation features, we discuss the LoRaWAN architecture and how the network performance are affected by radio access settings and the gateways deployment. We then present the platform designed for data collection and analytics and some statistics gathered in the first year of operation. We then discuss different strategies able to optimize the global network performance. To this aim: i) we define an algorithm to optimize LoRa Data Rate allocation along the nodes, in single and multi gateway scenarios; ii) we propose a methodology to process LoRaWAN packets and perform devices profiling; iii) we propose a methodology to predict the radio space utilization.*

1 Introduction

IoT devices and applications are playing a crucial role in the human everyday life, they are used to deploy smart environments where people live. Thanks to the IoT paradigm homes, buildings, and industries are connected together to apply the smart cities concept. Providing intelligence to these environments means enabling a broad set of operations, such as collecting data, triggering alarms, and reconfiguring smart objects. According to the IoT Analytics forecast of 2018 [1], the market for IoT has seen an unexpected acceleration in the last years. Currently, the number of connected devices exceeds 17 billions, and the number of IoT devices is 7 billions. We expected that the number of interacting devices, based on heterogeneous architectures and capabilities, can be as high as thousands of devices, thus leading to significant volumes of data potentially streamed through a radio interface to an optimization/intelligence application. We need technologies and IoT network architecture capable of managing these amount of data information, with characteristics of scalability and efficiency.

In the context of IoT applications, a promising technology for supporting radio connectivity in wide coverage network is LoRaWAN [2]. LoRaWAN is a new LPWAN technology which enables power efficient wireless communications over very long distances. LoRaWAN uses LoRa modulation [3] and form one-hop networks where every node can reach directly one (or more) internet connected sink nodes. LoRaWAN works on scientific and medical (ISM) radio bands and in following the frequency plan provided in [4]. For the European region, the applied frequency plan is EU863-870, it provides 8 channels and 7 data rates. In this chapter we present a real LoRaWAN deployment in Italy, and we study real scalability problems related to the LoRaWAN deployment. The chapter sections are organized as follows: Section 2 presents the technology and the network architecture. Section 3 describes the target LoRaWAN large scale scenario used in our study and provides LoRa modulation features that can be exploited to improve network optimization. According with the amount of devices and generated data of the IoT scenarios, the Section 4 presents the scalability problem and three different approaches to improve the network performances, namely: i) the 'capture aware water filling' algorithm to optimize LoRa devices parameters, ii) a Machine Learning (ML) algorithm for LoRaWAN devices profiling, and iii) a ML algorithm to predict channel usage and perform network optimization.

2 What is LoRaWAN?

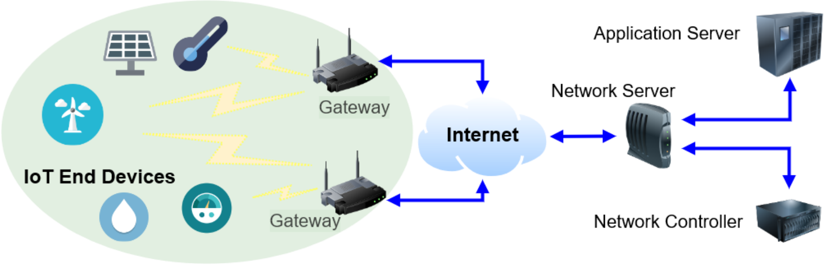


Figure 1: LoRaWAN network architecture.

LoRaWAN includes a simple network architecture, represented by a star-of-stars topology, able to connect thousands of End Devices (EDs) to Gateways (GWs), in an area of few kilometres, with very minimal maintenance. Figure 1 presents the LoRaWAN network architecture. Any ED is not strictly associated to a specific GW but, conversely, can interact with any reachable one. Communication starts with EDs sending packets to GWs which, in turn, forward them to the Network Server (NS). NS is responsible for processing packets, eliminating duplicated packets, and forwarding information data to the IoT applications server and to the Network Controller (NC). The NC is the node responsible for the network configurations. While LoRa defines the physical layer and is a proprietary technology, LoRaWAN entails the overall networking, the Medium Access Control (MAC); furthermore, LoRaWAN deals also with Applications Servers, Security, Firmware Update Over the Air [2].

2.1 LoRa Modulation

LoRa has been proposed by Semtech [3] and is a spread spectrum modulation technique derived from Chirp Spread Spectrum (CSS) technology. LoRa implements a CSS modulation that uses the entire wideband linear frequency to modulate chirp pulses. In LoRa, EDs support multi-rate by exploiting six different Spreading Factors (SF), from 7 to 12. The selection of the SF has an impact on duration and delivery probability of the generated packet. Communication on different SFs in the same channel are in principle orthogonal [5]. The SF defines two fundamental values: i) the number of chips contained in each symbol, that is $N = 2^{SF}$, ii) the number of raw bits that can be encoded by that symbol that is equal to SF. The LoRa Data Rate (DR) depends on the Bandwidth (BW) in Hz, the Spreading Factor sf and the Coding Rate (CR) as:

$$DR = sf \cdot \frac{BW}{2^{sf}} \cdot CR \tag{1}$$

where the symbols/sec are given by $BW/2^{sf}$ with $sf \in \{7 - 12\}$ and the channel coding rate CR is $4/(4 + RDD)$ with the number of redundancy bits $RDD = 1, \cdots, 4$.
The symbol duration (sec) is calculated as follow:

$$T_{sym} = 2^{sf}/BW \tag{2}$$

LoRa devices use a high SF when the signal is weak or there is a strong interference in the adopted channel. Using a high SF means a longer symbol duration and a consequent longer Time on Air (ToA). The selection of the DR is a trade-off between communication range and packet duration. Packets transmitted within different SFs, in principle, generate few interference with each other. To maximize both battery life of the EDs and overall network capacity, the LoRaWAN can manage the DR and RF output for each ED individually by means of an recommended ADR scheme [6]. This mechanism determines the transmission parameters (SF and transmit power) of the ED based on the estimation of the link budget in the uplink and the threshold of the Signal to Noise Ratio (SNR) for decoding the packet correctly at the current DR. In LoRaWAN, the system capacity can be kept large since the receiver can detect multiple simultaneous transmissions by exploiting the orthogonality when different SFs are used. Moreover, if the multiple simultaneous transmissions are generated with the same SF, a low difference in the signal strength (few dB values) can generate a channel capture effect that ensures the correct reception of the stronger signal. Although the presence of different SFs and the relaxed channel capture effect, the LoRaWAN performance suffers when the network scales and the parameters are not correctly tuned.

The limits and the performance expectations for LoRaWAN have been studied in the papers [7], [8] and [9]. The work in [7] provides an overview of the capabilities and limitations of LoRaWAN. Voigt et al. in [9], through simulations based on real experimental data, show the effects of the interference on performance of a LoRa network. Scalability issues in the LoRa system are analyzed in [10][11][12]. In the next Section, we show a real large scale LoRaWAN deployment and we study performance limits of the network.

3 LoRaWAN large scale scenario by UNIDATA operator

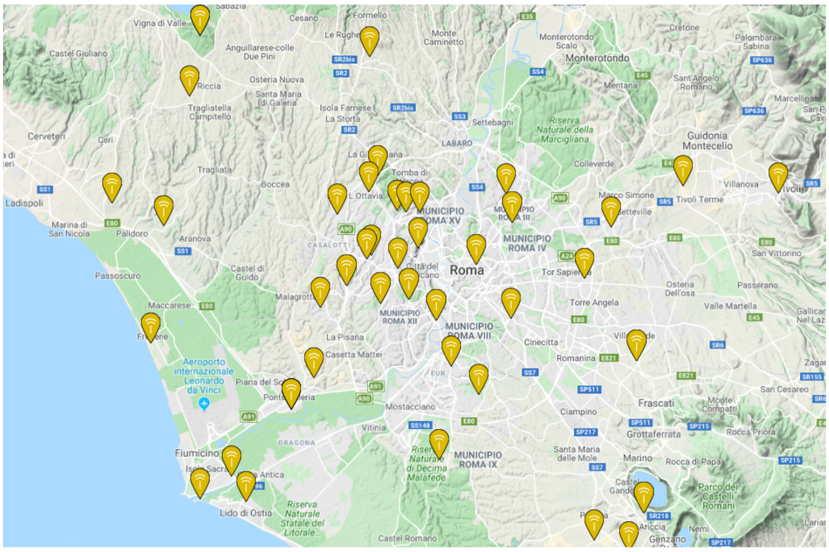
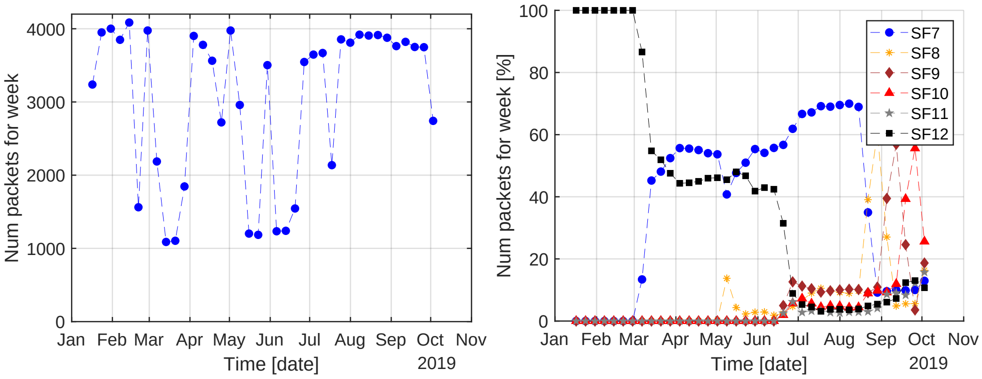


Figure 2: LoRaWAN GWs deployment around the Roma city area.

A quite spread LoRaWAN network deployed on Italy is the one of UNIDATA. UNIDATA provides IoT solutions applied in Smart City, Smart Building, Smart Metering and many other smart applications. The main application services provided by the current network are: i) water metering, ii) energy consumption metering, iii) management of parking slots; iv) GPS tracking, v) smart road lightning. The UNIDATA LoRaWAN network already reaches millions of people in Italy, the Figure 2 shows the current GWs deployment in the Rome city region. More specifically, in the whole Italy territory, the complete network involves **1862 installed EDs**, and **138 installed GWs**. Currently, the total amount of devices that results present in the network is **89.528** (much higher than the ones managed by UNIDATA). **In the 2019 the network has collected 372.119.877 packets** (2,25% are generated from EDs registered with UNIDATA).

The GWs are connected to the NS operator, located in Rome at the operator datacenter, where also the database is deployed. The database contains several indexes, each one representing a particular flow of information gathered from the network. For instance, there are flows representing uplink and downlink packets, information exchanged



(a)

(b)

Figure 3: LoRaWAN network with 300 water metering. Number of received packets per week in a period of 1 year (a), and percentage of packets belonging to each SFs for the same period (b).

between GW and the NS, or between EDs and NS, or packets which have been deduplicated because they have been received several times. The latter case happens when different GWs insist in the same geographic area and packets reach the NS from these different GWs. UNIDATA manages the network by UniOrchestra, a suite for creating and managing IoT LoRaWAN networks and services.

3.1 Performance behavior in large scale scenario

As an example of application provided by LoRaWAN, we considered a sensor network present in a city in the north of Italy and served by the UNIDATA. The network provides water metering for 300 real consumers. Each water counter is equipped with a LoRaWAN module that forwards the measurement around 18 times per week. Two GWs cover the whole area where the 300 water meters are. Packets, sent by the water meters, are forwarded by the GWs to the operator NS. Each received packet is processed by the NS, the payload message is sent to the application service, while the device radio link information are forwarded to the operator NC.

Figure 3(a) shows the number of received packets week by week in a period of 1 year. The NC implements the recommended LoRaWAN ADR algorithm [6] that estimates the best SF and transmission power values for each ED according to the measure Signal to Noise Ratio (SNR). To determine the optimal DR, the NC collects statistics on the packets received by the NS in the uplink. For example, the 10 most recent uplinks. These measurements contain the frame counter and the SNR value related to the best GW, the one receiving the package with a value of greater SNR. For each measurement the NC calculates the so-called "margin", which is the measured SNR minus the required SNR to demodulate a message given the DR. This margin is used to determine how much it is possible to increase the DR (when BW and CR are fixed this mean to decrease the SF) or to reduce the transmit power (if the SF is already the lowest one). For instance, when the network receives a message with DR SF12BW125 (SF=12 and BW=125) and SNR is 5 dB, the resulting margin is 25 dB (required SNR at SF=12 is -20 dB [6]). This is more enough for the LoRa demodulation, it is the possible to improve the performance in terms of airtime, by increasing the DR, and in terms of energy by decreasing the transmission power. If we increase the DR to SF7BW125 we still have a margin of 12.5 dB (required SNR at SF=7 is -7.5 dB [6]), this new configuration reduce the ToA value, ensuring minor energy consumption and collision probability. We could even lower the transmit power to save even more energy and cause less interference. Figure 3(b) shows the percentage of nodes for each SF related to the transmissions in Figure 3(a). At the starting moment the ADR was disabled and from the Figure we can notice that all nodes used SF12. The ADR has been enabled in March, but the margin is kept high for the next 3 months (up to June). In this period only two SFs were used, SF7 for nodes with an higher SNR and SF12 for the others. From July, the margin has been reduced and the nodes are more equally distributed among the SFs.

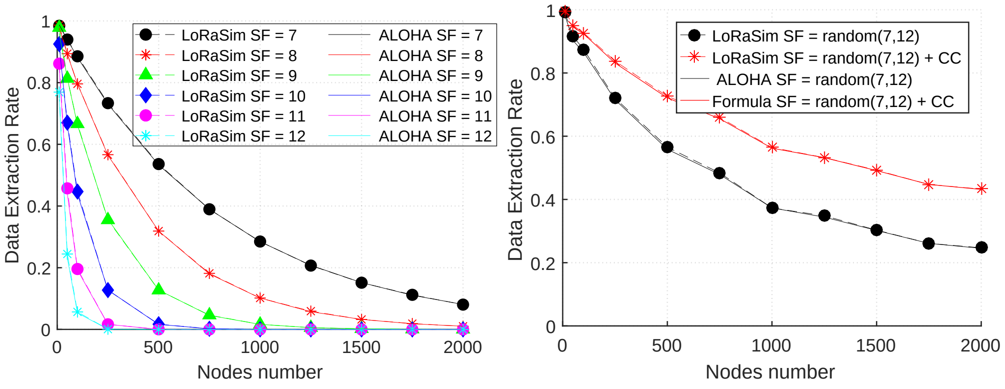
3.2 Capacity of LoRa Cells and channel effects

This sub section will present a study of the capacity of a LoRa cell and the impact of the channel effects on the capacity. The medium access in a LoRa cell works as a non-slotted Aloha system. Under Poisson packet arrivals, it is possible to model the cell throughput as $G \cdot e^{-2G}$, being G the normalized load offered to the cell. The probability of correctly receiving a packet transmission, which is a typical metric considered for characterizing LoRaWAN systems, often called Data Extraction Rate (DER), is given by e^{-2G}. Since different SF are available, the system works as the super-position of multiple coexisting (but independent) Aloha systems, each one experiencing the load due to the nodes employing a given SF. Let n_{sf} be the total number of EDs in the cell employing a SF equal to sf (with $sf \in \{7, 12\}$). The time interval required for transmitting a packet is given by the sum of the preamble time, which lasts m_{ph} symbol times T_{sym} as in (2), and payload transmission time. Since each symbol time codes sf bits and a channel coding with rate $CR = 4/(4 + RDD)$ is applied, the time ToA_{sf} required for transmitting over the air a packet of P bytes with spreading factor sf can be expressed as:

$$ToA_{sf} = (m_{ph} + \lceil \frac{8P}{4sf} \rceil \cdot (4 + RDD)) \cdot T_{sym} \tag{3}$$

Assuming that every ED generates packets with a source rate of s pkt/s, the normalized load using spreading factor sf can be expressed as $G_{sf} = n_{sf} \cdot s \cdot ToA_{sf}$. The total cell capacity results equal to $\sum_{k=7}^{12} G_k e^{-2G_k}$ and can dramatically decrease (down to zero) as the loads G_{sf} increase.

Figure 4(a) compares the DER performance of an Aloha system (lines) with the results obtained by simulating a LoRa cell (points) as a function of the offered load. Simulations have been obtained by using the public available LoRaSim simulator [13], while the offered load is expressed by the number of EDs transmitting in the cell at a source rate of 1 packet every $90sec$, with a packet size of 20 bytes. LoRaSim is a custom-build discrete event simulator implemented in Python, other configuration parameters used for the result in this chapter are reported in Table 1. All the nodes are configured with the same SF and different curves refer to settings which vary from SF 7 to SF 12. From the Figure, we can notice that the Aloha model well describes the system behavior. The performance degrade significantly using high spreading factors because larger ToAs



(a)

(b)

Figure 4: DER as a function of the number of EDs transmitting for each SF. Case of a single SF used for all the EDs in comparison with Aloha formula (a). Case with EDs uniformly distributed among all the SFs (b), without channel capture and with channel capture (+ CC).

Parameter	**Value**	Parameter	**Value**
Carrier Frequency (MHz)	868.3	Bandwidth (kHz)	125
Code Rate (CR)	4/5	TXPower	14 dBm
Path loss values	$\eta = 2.9$, $\sigma^2 = 0$,	$\overline{L_{pl}}(40m)$	$-66\ dB$

Table 1: Simulation parameters.

correspond to higher load conditions. For instance, with 500 EDs, the DER is almost zero for $sf = 11$ or $sf = 12$, while it is still above 0.5 for $sf = 7$.

Capture-Effect LoRa modulation is very robust to Gaussian noise, but also to self-interference due to colliding transmitters [10]. Indeed, in case of collisions between two or more transmitters, a Signal to Interference Ratio (SIR) value of just very few dBs (actually, as little as 1 dB for all the SFs in the experiments described in [14], versus the 6 dB specified in [3]) is enough for correctly demodulating the strongest colliding signal. This phenomenon, called "*channel capture*", has a strong impact on the scalability of LoRa technology, because the deployment of multiple GWs can significantly boost the capture probability and thus the overall network capacity.

A simple approximation of the throughput improvement due to channel capture can be obtained by considering that in most practical cases, a target ED collides with a single interfering signal at time. This assumption is reasonable when the cell works in stable conditions and collisions involving multiple overlapping packets are rare or have a dominant contribution in the interfering power. Under this approximation, a target ED employing a given spreading factor sf is actually competing with a fraction of the total load G_{sf}. In [15] is presented a simplified model for a circular ring radius cell, transmitter is located at distance r from the gateway, and interfering nodes in a circular ring delimited by a distance $\alpha \cdot r$.

Neglecting the effect of random fading and assuming an attenuation law of type $r^{-\eta}$, all the interfering nodes at distance higher than $\alpha \cdot r$, with $\alpha = 10^{SIR/10\eta} > 1$, do not prevent the correct demodulation of signal of the target ED. The smaller the α coefficient, the lower the competing load is. Therefore, the cell throughput in presence of channel capture can be obtained by generalizing the Aloha results as:

$$S_c(G_{sf}) = 2\pi \int_0^{R/\alpha} \delta_{sf} e^{-2\frac{\alpha^2 r^2}{R^2}G_{sf}} r \cdot dr + \delta_{sf}(\pi R^2 - \pi R^2/\alpha^2)e^{-2\cdot G_{sf}} \tag{4}$$

where $\delta_{sf} = G_{sf}/(\pi R^2)$ is the density of load offered to spreading factor i and R the cell radius. The DER is simply given by $S_c(G_{sf})/G_{sf}$.

Figure 4(b) shows the DER achieved with (red curve) and without (black curve) capture effects as a function of the number of EDs in the cell, for a capture SIR threshold of 1 dB (power ratios are higher than 1 dB to generate a capture effect). Each ED is configured on a randomly chosen SF between 7 and 12. Simulation results are plotted using points, while the analytic results are reported by lines. Simulation results match pretty well the upper bound provided by our model, although we ignore accumulation of interference generated by multiple packets.

Interference inter-SF In reality, different SFs are not perfectly orthogonal: it may happen that the reception of a target packet transmitted with a given SF is prevented by an overlapping packet transmitted with a different SF, when the SIR is lower than a rejection threshold (power ratios are lower than the threshold to generate a collision). In [14], it has been experimentally shown that the rejection thresholds are almost independent on the SF of the interfering ED and vary in the range between -10dB (for target packets transmitted at SF 7) and -25dB (for target packets transmitted at SF 12). For simplicity, in the following we refer to a constant inter-SF rejection threshold of -16dB. Because of imperfect orthogonality, a target ED working on SF sf at a generic distance r will compete not only with the load $G_{sf} = n_{sf} \cdot s \cdot ToA_{sf}$ offered to the same SF, but also with a fraction of the load G_{-sf} working with a SF different from sf, corresponding to the EDs closer to the gateway. For a given rejection threshold SIR, only EDs placed in a cell sub-region delimited by a radius $\beta \cdot r$, with $\beta = 10^{SIR/10\eta} < 1$ can interfere with the target ED while transmitting with a different SF. Since such a fraction depends on the distance r and since the number of target EDs grows proportionally to r in case of devices uniformly placed within the cell, the average success rate $DER(sf)$ for a generic target ED working on SF sf can be written as:

$$e^{-2G_{sf}} \cdot \int_0^R e^{-\frac{\beta^2 r^2}{R^2}\sum_{k\neq sf} n_k s\cdot(ToA_k + ToA_{sf})} \frac{2r}{R} dr \tag{5}$$

where each term $e^{-\frac{\beta^2 r^2}{R^2} n_k s\cdot(ToA_k+ToA_{sf})}$ is the probability that no transmission at SF k has been started in the interval ToA_k before the starting of the target packet, and no other one is originated during the following packet transmission time ToA_{sf}. It follows, that in stable conditions, when the load offered to each SF is lower than 0.5, we have:

$$DER(sf) = e^{-2G_{sf}} e^{-\beta^2/2 \sum_k n_k \cdot s \cdot (ToA_k + ToA_{sf})} \tag{6}$$

In other words, the total load L_{sf} competing with the target ED is not only G_{sf}, but also a fraction $\frac{\beta^2}{2}\frac{ToA_k+ToA_{sf}}{2\cdot ToA_k}$ of the load G_k offered by each different SF k (with $k \neq sf$).

4 LoRaWAN scalability problems and solutions

In the previous Section we introduced a real LoRaWAN deployment and we showed how the performance suffers when the network scales and the parameters are not correctly tuned. However, **network capacity can be boosted through an aware EDs parameters configuration that take into account LoRa channel effects** presented before. Network **optimization can be reached by an aware SF allocation that supporting load balancing, as well applying device profiling and prediction schemes based on ML**. The idea is to understand EDs network conditions and choose context aware algorithm to maximize network performance. In this section we present possible solutions to address the scalability issue of the network based on three different approaches: i) balancing of the SFs allocation, ii) profiling of the ED behavior, iii) predicting channels' utilization.

4.1 Algorithm to optimize LoRa parameters (SF balancing)

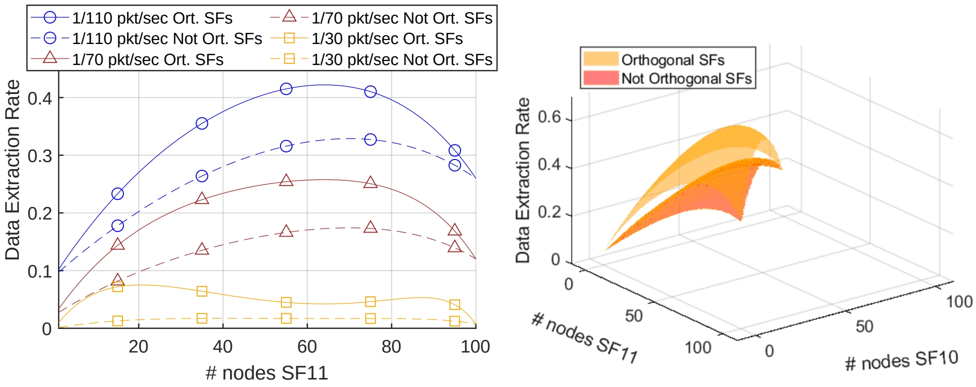
A critical aspect in LoRaWAN networks is represented by the configuration of the channels employed by different GWs (multiple channels can be supported by the same gateway) and by the allocation of SFs to different devices. The SFs allocation has an impact on the distance at which the ED can be located and on the robustness of the radio link in presence of fading. On the other side, **the ToA spent by packets sent at different SFs can be significantly different** (being the ratio between the minimum and the maximum possible ToA about 2^5). **It follows that SFs allocation has also an impact on the system load.** A detailed analysis of the impact of different configuration choices for SFs taken by both the EDs and the network is presented in [16]. The selection of the optimum SF for each device is a function of the ADR. ADR should be enabled whenever an ED has sufficiently stable RF conditions. The recent literature also concentrated on the ADR mechanisms. In [17] is present a strategy, named EXPLORA-AT for implementing a suitable ADR in LoRaWAN systems. The key idea was to assign the SFs to the EDs in a way that assures an equal time-on-air occupation to all the available SFs. In the rest of the subsection, we first present the optimum values of SFs balancing when capture effect and interference inter-Sf exist, and after we present an algorithm to optimize this allocation [15].

Optimal SF Balancing The problem of spreading factor allocation in a network with N total EDs can be modeled by the choice of n_7, n_8, $\cdots$, n_{12}, i.e. the choice of the number of EDs using each available spreading factor, with the constraint that $\sum_{sf} n_{sf} = N$. A possible optimization criterion is to maximize the average data extraction rate:

$$E[DER] = \frac{\sum_{k=7}^{11} n_k e^{-2n_k \cdot s \cdot ToA_k}}{N} \tag{7}$$

If we replace n_{12} with $N - \sum_{k=7}^{11} n_k$ and null all the derivatives $\frac{\partial E[DER]}{\partial n_{sf}}$ with respect to a generic n_{sf} with $sf \neq 12$, we obtain:

$$e^{-2n_{sf} \cdot s \cdot ToA_{sf}}(1-2n_{sf} \cdot s \cdot ToA_{sf}) - e^{-2(N-\sum_{k=7}^{11} n_k) \cdot s \cdot ToA_{12}} \cdot [1-2(N-\sum_{k=7}^{11} n_k) \cdot s \cdot ToA_{12}] = 0 \tag{8}$$



(a) (b)

Figure 5: DER as a function of the number of EDs when only 2 SFs are used, $sf = 11$ and $sf = 12$ (a) and when 3 SFs are used $sf = 10$, $sf = 11$ and $sf = 12$ (b).

By relaxing the constraint of an integer number of end devices per each spreading factor, i.e. by permitting real values for each n_i, from the previous equation, it is evident that the solution $n_{sf} ToA_{sf} = n_{12} ToA_{12}$ $\forall sf$, can be a maximum or a minimum because it nulls all the derivatives. By solving for all the SFs and considering the constraint on the total number of EDs, we have:

$$n^*_{sf} = \frac{ToA_{12}}{ToA_{sf}} \frac{N}{\sum_{k=7}^{12} ToA_{12}/ToA_k} \quad \forall sf \tag{9}$$

If we assume a network working in stable conditions and therefore $1 - 2n_{sf} \cdot s \cdot ToA_{sf}$ is greater than zero, the solution $\mathbf{n}^* = [n^*_7, n^*_8, \cdots n^*_{12}]$ is a maximum. For very high loads, when $1 - 2n_{sf} \cdot s \cdot ToA_{sf} < -1$, the solution $\mathbf{n}^*$ becomes a minimum. In such a condition, it is interesting to see that $E[DER]$ exhibits other maximization solutions, which are obtained by enforcing a normalized offered load equal to 0.5 (i.e. $n_{sf} = \frac{0.5}{s \cdot ToA_{sf}}$) in all the SFs except one $sf = \widehat{sf}$, in which all the residual $N - \sum_{k=7, k \neq \widehat{sf}}^{11} n_k$ are allocated. In other words, for high load conditions, $E[DER]$ is maximized by leaving one SF working in unstable conditions (and in particular, the optimal choice is $\widehat{sf} = 12$), and by optimizing the load of all the remaining SFs by setting $G_{sf} = 0.5$.

Figure 5 visualizes the previous considerations in a system with two (5(a), continues lines) or three (5(b), orange surface) available SFs and a total number N of EDs equal to 100. When only two SFs are available ($sf = 11$ and $sf = 12$), the optimal number of nodes configured on each SF can be determined by studying a single variable function. In the figure 5(a) we can immediately recognize that the point which nulls the derivative of $DER(n_{11})$ is given by the solution of the equality $n_{11} ToA_{11} = n_{12} ToA_{12}$, that is n_{11}=64 nodes, being ToA_{12} about twice as ToA_{11} as results from (2) and (3). As the source rate employed by all the nodes increases, the point changes from a maximum to a minimum point (yellow curve). For high load conditions, the optimal choice is to fix the load for one of the two SF and let the other one become congested (that is n_{11}=21 nodes in figure 5(a) when source rate is 1/30 pkt/sec). Since the number of stations working in stable conditions are maximized when $G_{11} = 0.5$ (rather than $G_{12} = 0.5$), the global maximum is reached when $n_{11} = 0.5/(s \cdot ToA_{11})$. The figure 5(b) shows a 3D plot when an additional SF is considered ($sf = 10$), for a source rate $s = 1/90 pkt/sec$. Also in this case, the vector $[n_{10}, n_{11}]$ which nulls the derivative can be easily recognized (n_{10}=56 and n_{11}=28 for the orthogonal SFs scenario).

The optimization criteria can be extended in presence of inter-SF interference. In such as case, we have:

$$E[DER] \qquad = \qquad \frac{\sum_{sf=7}^{12} n_{sf} e^{-2n_{sf}\cdot s\cdot ToA_{sf} - \sum_{k\neq sf}^{12} \frac{\beta^2}{2} n_k \cdot s \cdot (ToA_k + ToA_{sf})}}{N} \tag{10}$$

If we replace again n_{12} with $N - \sum_{l=7}^{11} n_l$ and compute the derivatives $\frac{\partial E[DER]}{\partial n_{sf}}$ with respect to a generic n_{sf} with $sf \neq 12$, we obtain:

$$\begin{aligned} e^{-2L_{sf}}[1 - n_{sf} \cdot s(2 \cdot ToA_{sf} - \frac{\beta^2}{2} n_{sf} \cdot (ToA_{12} + ToA_{sf})]+ \\ e^{-2L_k} \sum_{k\neq sf}^{11} \frac{\beta^2}{2} n_k \cdot s\,[-(ToA_k + ToA_{sf}) + (ToA_k + ToA_{12})] + \\ e^{-2L_{12}}[-1 + (N - \sum_{l=7}^{11} n_l) \cdot s(-\frac{\beta^2}{2} \cdot (ToA_{12} + ToA_{sf}) + 2 \cdot ToA_{12})] \end{aligned} \tag{11}$$

By permitting real values for each n_k, it is easy to show that the vector of unknown allocations $[n_7^*, n_8^* \cdots n_{11}^*]$ for which $L_k = L_{sf} = L_{12}\ \forall k$ nulls all the derivatives $\frac{\partial E[DER]}{\partial n_{sf}}$ with respect to a generic n_{sf}. Indeed, in such a case we can simplify the exponential terms from the previous expression and note that the sum of all the other terms is equal to $L_{12} - L_{sf}$, which in turns is equal to zero. To find the optimal allocations it is required to solve a linear system in the unknown variables $n_7, n_8, \cdots n_{11}$. It can be easily shown that each component of the system solution can be written as:

$$n_{sf}^* = \frac{ToA_{12}}{ToA_{sf}} \frac{N - \frac{\beta^2}{4} N \left[\sum_{k=7}^{12} (\frac{ToA_{sf}}{ToA_k} - 1) + 2\right]}{\sum_{k=7}^{12} \frac{ToA_{12}}{ToA_k} (1 - \frac{\beta^2}{2})} \qquad \forall sf \tag{12}$$

Such an expression coincides with the one derived in the previous section when β is equal to zero (i.e. in absence of inter-SF interference). Figure 5 also shows the effects of inter-SF interference on the average DER and on the optimal load allocations across SFs. In particular, the dashed lines in figure 5(a) refers to a cell with two SFs, in which the threshold for rejecting inter-SF interference is equal to -10dB. We can see that the optimal load allocations are now achieved by increasing the number of nodes on SF 11 (from 64 to 70). Similarly, in the red surface in figure 5(b), the optimal load allocations across three SFs is shifted towards an increased number of stations on the highest possible rate. Obviously, the DER achieved under optimal allocations is lower than the one achieved without inter-SF interference.

More generally, load balancing can be achieved by updating the proportion between the number of EDs that can be allocated on each SF as derived in equation 12. Table 2 summarizes the load balancing results in the case of orthogonal and non-orthogonal

SF	7	8	9	10	11	12
ToA $[msec]$	49.41	90.62	164.86	329.73	659.46	1187.84
P_{sf} **[%], orth.**	47.02	25.85	14.36	7.18	3.59	2.02
P_{sf} **[%], not orth.**	50.75	26.98	14.07	0.060	0.019	0.002

Table 2: ToA (in ms) as a function of SFs when payload size is 20 byte and coding rate is 4/5; resulting optimal percentages P_{sf} in accordance to optimal load allocation

SFs. In this last case, the rejection threshold has been configured to $-16dB$. For non-orthogonal SFs, the portion of EDs transmitting at SF 7 increases, while the allocations performed at SF 11 and SF 12 are reduced to almost zero.

This result can be used the ADR algorithm, it should be used the optimal percentages P_{sf} to improve the network performance. The idea is addressed by EXPLoRa algorithm [17].

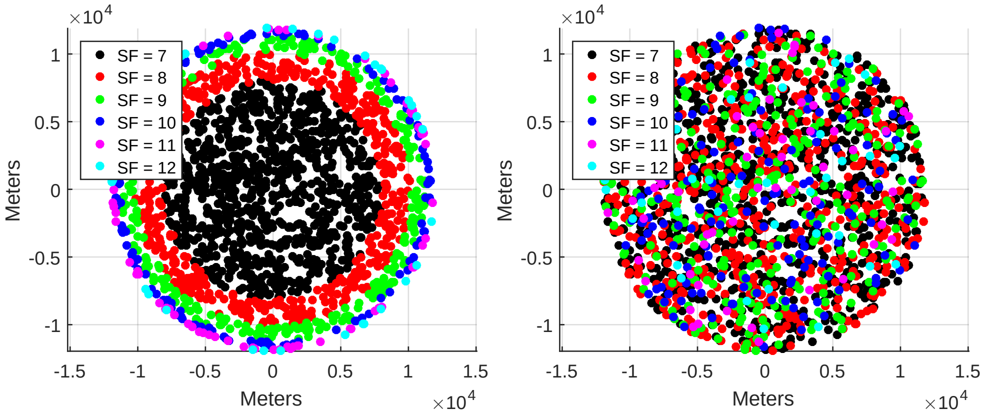
EXPLoRa proposes two different variants: a basic solution, called EXPLoRa-SF, and an enhanced approach named EXPLoRa-Air Time (EXPLoRa-AT). This approach starts from the remark that a rate adaptation strategy merely based on link-level budget/measurements, such as the recommended ADR defined by the LoRa Alliance [2], cannot take advantage of the (quasi) orthogonal nature of different SFs. For an extreme example, if all network devices are very close to the gateway, they will all select $sf = 7$, thus congesting such SF, while all the remaining SFs will remain "empty". Better allocation strategy consists in "forcing" some devices to operate with a higher than necessary SF, thus spending a higher ToA, but gaining from a better allocation of load among the available SF-induced "channels".

The goal of EXPLoRa-SF was to show that performance can increase by distributing users on different SFs. Under EXPLoRA-SF the nodes are equally split between SF sub-channels: although some nodes transmit with a ToA higher than necessary, the reduction on the data rate is compensated by the reduction of the interference caused by simultaneous transmissions from the other nodes. EXPLoRa-AT introduced an allocation strategy than equally splitting the nodes among different SFs, it equally balances the total ToA spent on each SF. The number of nodes in each SF follows the proportion reported in Table 2, row orthogonal.

The effects of load balancing on the proportion of EDs working on the same SF is depicted in Figure 6(a) for the orthogonal case. We can easily recognize that about one half of the nodes, colored in black, are using $sf = 7$ and all the other EDs follow the optimal $P_{sf}(sf)$ proportions.

However, the position of nodes employing the same SF plays a further crucial role, especially due to the fact that LoRa has a somewhat unexpected capability to capture and correctly demodulate a signal even in the presence of a significant interference. All these aspects are jointly accounted in the EXPLoRa-Capture (EXPLoRa-C) algorithm [15], where a sequential allocations in different circular rings is performed. The idea is to choose the SF to be used by each ED.

The first interesting aspect of the EXPLoRa-C is the possibility to implement the load balance criterion in terms of "sequential waterfilling". For facilitating the selection of a data rate compatible with the link budget, EDs are ordered according to their $RSSI$ value (from the highest to the lowest) and SF allocations are performed sequentially (from the highest rate $sf = 7$ to lowest rate $sf = 12$). This procedure ensures to take the advantage



(a)

(b)

Figure 6: Nodes position and allocated SF with EXPLoRa-AT (a) and EXPLoRa-C (b), in single gateway cell.

of the channel capture.

The basic idea of this extension, is exploiting the "spatial" dimension for reducing the effective load experienced on each SF. For a single cell system, this corresponds to spread the EDs working on a given SF within the cell, in order to increase the probability that colliding signals are received with a power ratio higher than the capture threshold. Note that this is generally different from allocating SFs sequentially to the EDs as a function of their ordered $RSSI$ values, because such an allocation could assign the same SF to nodes with similar $RSSI$ values. In other words, EDs employing the same SF should be at different distances from the gateway, rather than concentrated in a circular ring. For a multi-cell system, the spatial dimension can benefit from the availability of multiple gateways: nodes at a similar distance from the closest gateway could indeed be received with very different $RSSI$s from the neighbor gateways, thus resulting in a good capture probability at a different gateway.

In order to map these considerations into a new allocation strategy, the algorithm introduces the concept of distance between EDs, taking into account both the difference between the $RSSI$s at the closest gateway, and difference in the set of gateways in their coverage area. The basic idea of EXPLoRa-C is still based on a sequential allocation of SFs, equally sharing the ToA spent at different SFs, but the allocation is performed in multiple rounds, by skipping in each round the decision on nodes which are at a small distance from the previous decision.

In case of M gateways, $GW_1, GW_2, \ldots GW_M$, are deployed in a network with N nodes, EXPLoRa-C is executed M times. All the EDs are organized into M sets and ordered as a function of the $RSSI$ value perceived by the closest gateway. For each set, EXPLoRa-C is executed by considering the total number of allocations on each SF as equal to $P_{sf}(sf) \cdot N_m$ (with $m \in [1, M]$). Moreover, the concept of distance between EDs is extended by also considering the neighbor gateways. In case nodes have $RSSI$ values whose difference is lower than the capture threshold but the set of GWs in range is difference, they can still be configured on the same SF.

In the following, we present the performance evaluation of EXPLoRa-C by using

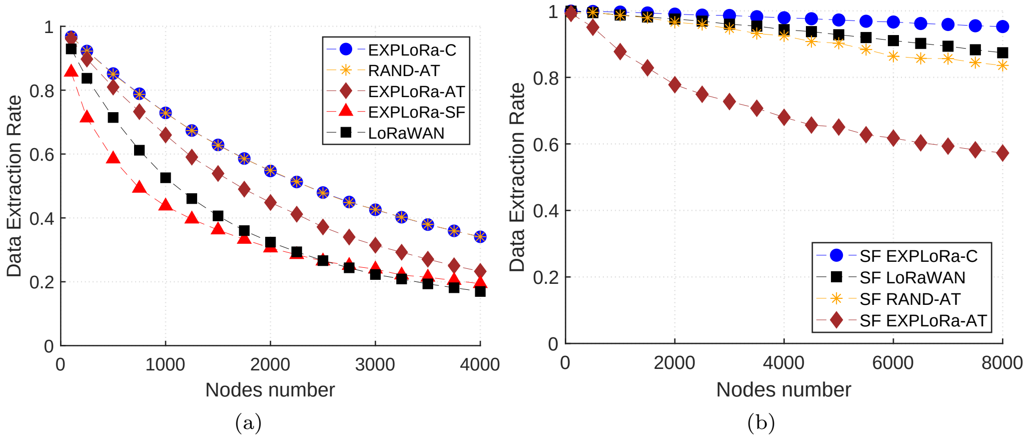
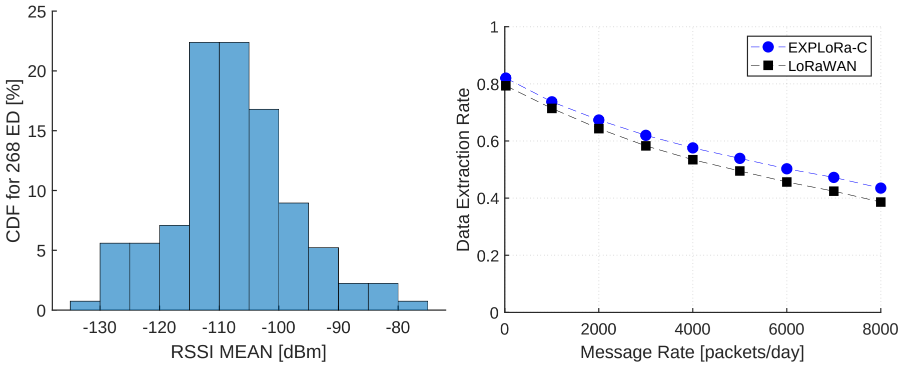


Figure 7: DER as a function of the EDs numbers in presence of channel capture, comparison among EXPLORA-C, RAND-AT, EXPLoRa-AT, EXPLoRa-SF and LoRaWAN, when we consider single cell (a), and multi-gateway scenario (b).

the LoRaSim simulation framework, in both the single cell and multi-gateway scenarios. Simulation parameters are reported in table 1, and channel capture is enabled. The simulation uses a cell dimensions, namely a radius R, of 12 km. It represents an unconstrained deployment. Indeed, for the unconstrained deployment any node can use any SF, because with the considered propagation model and transmission power entails a $RSSI$ at the cell edge that is enough for using $sf = 7$, i.e. the highest rate. We compare the results obtained by performing completely random allocations or by using different variants of EXPLoRa and the LoRaWAN recommenced scheme. Figure 7(a) plots the average DER achieved in the unconstrained deployment and a varying number of EDs (from 100 to 4000), which are uniformly placed within the cell area. We assume that SFs are perfectly orthogonal and do not interfere each other. From the figure, we can observe that just equalling splitting the EDs across all the available SFs is not a good strategy: indeed, EXPLoRa-SF achieves performance which are worse or almost equivalent to the LoRaWAN legacy scheme. This is due to the load experienced on $sf = 12$, which can reach unstable conditions even with a few hundreds of EDs. Specifically, when $s = 1pk/90\ sec$, the normalized load offered on $sf = 12$ by each node is equal to 0.0132 ($ToA_{12} = 1.19\ sec$) and therefore it is enough that n_{12} is equal to 40 nodes (i.e. the total number of EDs N is equal to $6 \cdot n_{12} = 240$) to work in unstable conditions. In the case of LoRaWAN, since the link budget is not a constraint, all the EDs are always configured for working with $sf = 7$, with a waste of cell capacity.

Conversely, EXPLoRa-AT is able to optimize such a capacity, by equally sharing the normalized offered load on each available SF. However, by also optimizing the possibility to achieve channel captures in case of collisions, EXPLoRa-C can further improve the average DER, especially in high load conditions. Since in this scenario all the nodes can be served at the highest data rate, the performance achieved under a completely random SF assignment (named RAND-AT in the Figure) which obeys to the $P_{sf}(sf)$ proportions (i.e. randomly chooses $P_{sf}(sf) \cdot N$ nodes for using a given SF) are equal to the ones achieved with EXPLoRa-C. We could argue that boosting the capture effects can result



(a) (b)

Figure 8: For the 300 devices real LoRaWAN network, CDF of the $RSSI$ mean (a), and performance comparison in terms of DER for EXPLoRa-C and LoRaWAN schemes (b).

in unfair performance between EDs. However, we find that the capture effect improves the DER experienced by some nodes without degrading the performance of the other ones involved into the collisions.

For assessing EXPLoRa-C performance in a multi-gateway scenario, we considered a regular grid, in which gateways are placed at regular distances and border effects are neglected. The scenario corresponding to the regular grid is given by a topology with 25 gateways, regularly placed with a distance of 12 km from the neighbors. The average DER is presented in figure 7(b) as a function of the number of EDs placed in the whole network (up to 8000 nodes). In this case, EXPLoRa-C provides not only the optimal results, but also a gain on the random allocations implemented by RAND-AT. The gain in comparison to EXPLoRa-AT is about 38%, and 8% in comparison with recommended LoRaWAN ADR.

Application in real scenario In order to test the EXPLoRa-C scheme in a real LoRaWAN deployment, we considered the real sensor network deployment presented in the Subsction 3.1. The evaluation result uses the network database to retrieve the behaviour of the devices in a range period of 1 year, specifically, it was extracted the $RSSI$ mean value for each device and SF assigned by the LoRaWAN ADR. Figure 8(a) shows Cumulative Distribution Function (CDF) of the $RSSI$ for the 300 devices present in the network. The $RSSI$ mean and the SF values, assigned by the LoRaWAN ADR, it was used to recreate the scenario by using the LoRaSim simulator. To reproduce the application scenario it was also set a σ^2 variance of 6 to account the path loss shadowing. Figure 8(b) shows the DER for both EXPLoRa-C and LoRaWAN schemes when the message rate increase from 18 to 8000 packets per day. From the Figure we can observe that EXPLoRa-C gets better performance than LoRaWAN, especially when the message rate increases. In case of a message rate of 8000 packets per day, EXPLoRa-C gets a DER of 5% greater than LoRaWAN.

4.2 LoRaWAN End Device profiling

How see before, the estimation of the ED radio condition potentially helping the network planning. In this direction, a possible strategy can be to apply device profiling related to the utilization of wireless resources in the LoRaWAN networks. In [18] it is proposed a methodology to process LoRaWAN packets and perform profiling of the IoT devices. Specifically, the work uses the k-means algorithm to group devices according to their radio and network behaviour.

More in generally, the approach can be help to: i) monitor the system behaviour and capture anomalies; ii) optimize the network planning; iv) identify different spaces where service provision can be enhanced, that is, new radio resources, more suitable parameter settings, different configurations of the IoT devices and services. This technique allow the clustering of EDs into "activity groups", which consist of ED which tend to have similar activity profiles.

The work uses the k-means algorithm which is a partitional clustering algorithm [19] and, as such, given a dataset $\mathcal{S} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ of n observations, it partitions the data into k non-overlapping clusters, i.e. $\mathcal{S} = \{\mathcal{S}_1, \dots, \mathcal{S}_k\}$ such that $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset$ if $i \neq j$ and $\cup_{i=1}^{k} \mathcal{S}_i = \mathcal{S}$. k-means finds a (sub-)optimal partition of the data in such a way that the intra-cluster variance or Within-Clusters Sum of Squares (WCSS) is minimized:

$$WCSS = \sum_{i=1}^{k} \sum_{\mathbf{x} \in \mathcal{S}_i} \|\mathbf{x} - \mathbf{c}^{(i)}\|^2 \tag{13}$$

where $\mathbf{c}^{(i)}$ is the centroid for cluster i, defined as the center of mass of the cluster itself.

Despite its simplicity, the number of clusters k to be returned is a parameter that must be tuned by the end-user and finding a suitable value is strictly problem and data-dependent and hardly known a-priori. Typically, one tries several k candidates and selects the best value by studying the objective function in Eq. (13) and/or by means of internal validation indices [20]. Common strategies include: i) The Elbow Plot [21] that consists in plotting the WCSS as function of k and choose the first k value corresponding to the point where the curve become flat. ii) The Davies-Bouldin Index [22] that measures the intra-cluster separation against the inter-cluster variance, the Davies-Bouldin index is not bounded within a specific range; however, the closer to 0, the better. iii) The Silhouette Score [23] that quantifies how-well each pattern has been assigned to its own cluster, conversely to the Davies-Bouldin index, the silhouette score is by definition bounded in range $[-1, +1]$: the closer to $+1$, the better.

This subsection presents an application of the k-means algorithm to the LoRaWAN real deployment introduced in Section 3. All the studies have been developed over the database "pre-deduplication" index; in such a way we are sure that all traffic analyzed is coming from the EDs to the NS, passing through different GWs. A subset of the database packet fields has been extracted and pre-processed and they will represent the features for the profiling analysis performed.

The goal of the work is to cluster (group) EDs according to the their behaviour. The dataset used in the analysis regards 12 months of activity over the entire LoRaWAN deployment network (January-Dicember 2019). From the initial dump of the database, packets having NULL device identifier have been discarded: these packets correspond to devices which are either not managed by the operator. In order to properly feed the k-means algorithm, the following pre-processing steps have been performed: i) packets

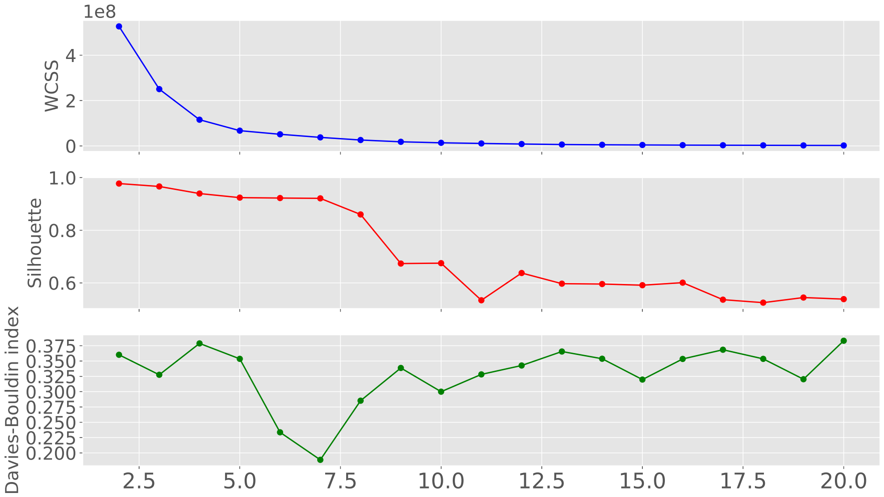


Figure 9: WCSS, Silhouette and Davies-Bouldin indices as function of k.

have been grouped by device; ii)each device has been mapped into an 7-length real-valued vector containing the following statistics amongst its packets: the error rate (elaborated via the FCNT field, that is a counter increasing for each packet sent from ED), the average values of: SNR; RSSI; packet size; and inter-arrival time (i.e., number of hours between two packets), the mode values of: channel frequency; and SF. Amongst the available fields inside a LoRaWAN ED packet, they depict the radio aspect of the EDs. Thanks to this pre-processing stage, the considered devices have been cast into a 7 columns real-valued matrix, suitable for being processed by the k-means, where each row represents an ED.

To evaluate the best k value, we considered several candidates $k = \{2, 3, \ldots, 20\}$ and Figure 9 shows the Davies-Bouldin Index, the Silhouette Index and the WCSS as function of k. By jointly considering the three indices, a suitable value of $k^\star = 7$ has been chosen; indeed, the Silhouette is rather high (> 0.9), the Davies-Bouldin index reaches its minimum value (< 0.2) and $k = 7$ lies pretty much towards the end of the WCSS elbow, which can be easily seen for $k \in [3, 7]$. For the chosen $k^\star$, we further analyzed the clustering solution as returned by k-means.

Specifically, for each cluster, we considered the behaviour of its 'most central' element (i.e., the element closest to the centroid) and Figures 10–11 show their most characteristic features. Figure 10 shows the PDF of the RSSI for the 'most central' elements of the 7 different clusters. Figure 11 shows the bar plot for the error rate values (Fig. 11(a)) and the number of packets sent (Fig. 11(b)).

In all figures, device IDs have been anonymized and replaced by the ID of their respective cluster (reported in the figures legend) and marked with a unique color which consistently spans across Figures 10–11. For the sake of readability, legends are not included in Figure 11.

By looking at Figure 10, three clear-cut profiles emerge in terms of RSSI, they are bounded by $RSSI \approx -100$ and $RSSI \approx -70$: the leftmost part of the spectrum sees

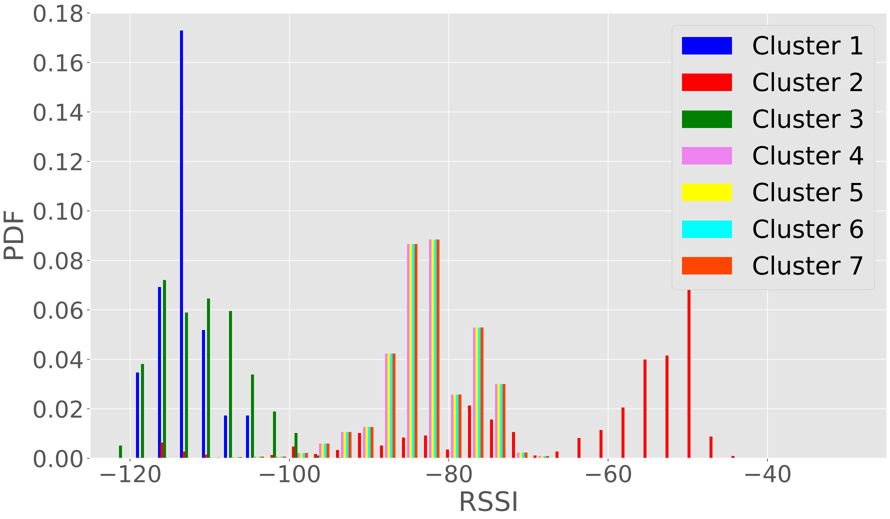
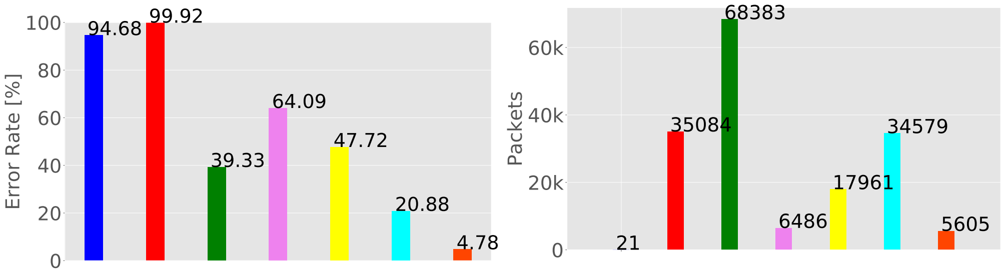


Figure 10: PDF of the RSSI for the 'most central' elements for the 7 clusters.



(a)

(b)

Figure 11: Error rate (a) and number of packet sent (b) for the 'most central' elements for the 7 clusters.

cluster 1 and 3 as the most prominent ones, whereas the rightmost part is dominated by cluster 2. All other clusters lie in the middle part of the spectrum. Figure 11 also shows completely different behaviours amongst the 7 clusters.

A interesting analysis consists in jointly considering these behaviours, which has been used as guideline to further characterize the radio properties of the devices belonging to each group. Notably, field-experts at the network operator recognized some devices having precisely the characteristics presented in these histograms (e.g., the device with a large number of packets sent is, actually, a device which sends about three packets per day, so these clustering results are indeed in line with the network behaviour). At the same time, the cluster analysis was capable to find anomalies inside the network: for example, cluster 1 represents a set of devices which have high error rate (Figure 11(a)) and small number of packets sent (Figure 11(b)). This, along with low RSSI values (Figure 10),

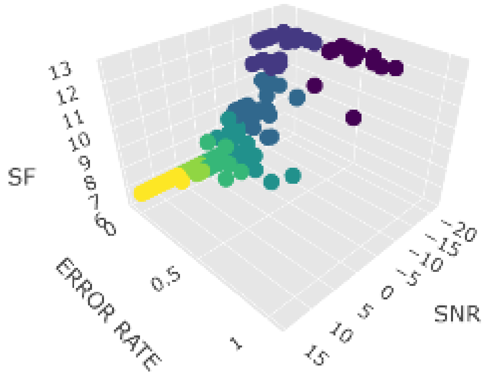


Figure 12: Sensor device representation in terms of SNR, ERROR RATE, and SF.

identifies a misbehaving of such devices.

To better show the potential of the k-means algorithm applied to the real deployment, Figure 12 shows the cluster representation of the water metering network devices presented in the Subsection 3.1, the 3D plot is related to the ERROR RATE, SF and SNR axis. Each cluster is represented by a different color. The Figure shows how 7 different clusters well map the current position of the the devices in the relative representation.

How see in the previous Subsection, the SF balancing is a critical operation in the LoRaWAN network, and is possible use balancing algorithm that starting to the radio characteristics find the best balancing among the EDs, in terms of how many and which EDs distributed for each SF. The clustering algorithm, presented in this Subsection, can be help to evaluate which EDs chose for the candidate SF.

4.3 Prediction of LoRaWAN network activity

Another optimization strategy presented in this chapter is the possibility to predict the behavior of the LoRaWAN scenario, the idea is to anticipate future problem in the network, and perform actions to avoid them. In fact, the knowledge of future device transmission schemes could facilitate the evaluation of the optimal network operating parameters. For example, foreseeing circumstances of saturation peaks in the radio space, we could perform (”in advance”) changing of the radio parameters, and thus avoiding future malfunctions and interruptions of the services connected to the applications.

The idea is to study, design and implement an algorithm capable of forecast the LoRaWAN network activities in terms of: i)periods of node activity, ii) percentage of usage for each channel/SF combination. In this section, we show an algorithm able to predict whether the radio channel, at a certain frequency/SF combination, will be busy or free at a specific time in the future and, if busy, what traffic intensity will be present (number of LoRaWAN packets).

This sort of problem can be modeled as a time series problem, where the channel usage statistics can be converted into a timebased sequence. Prediction of time series is a well-studied problem in the literature and multiple solutions have been published in

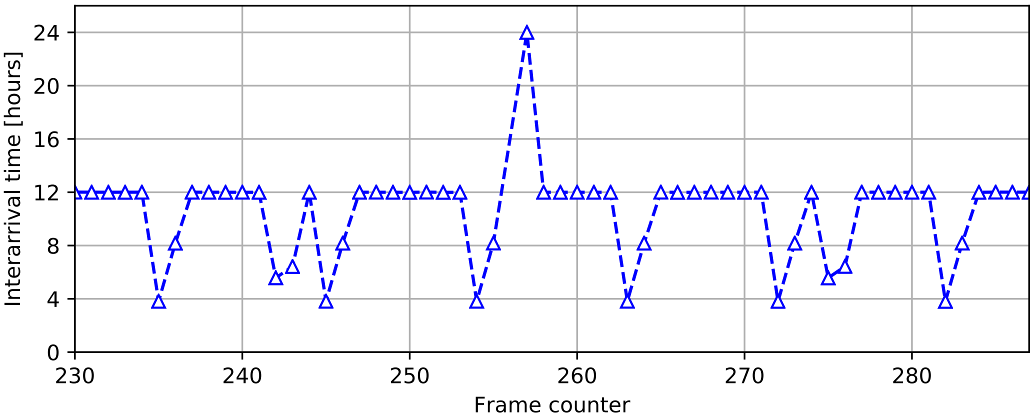
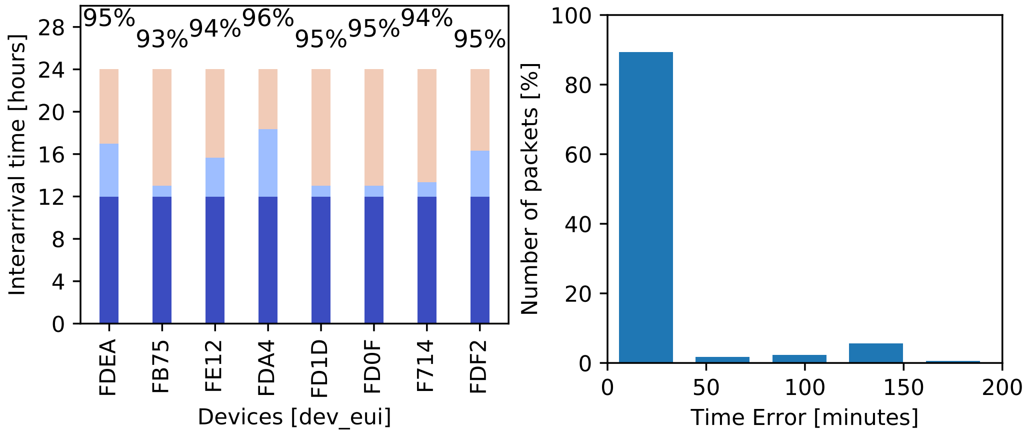


Figure 13: Pattern of the inter arrival time for the LoRaWAN water metering device

order to deal with this specific type of problem. More specifically, the our problem has been modeled as a time series problem in which each fixed interval of use of the channel is treated as a sequence of occupied and free slots [24]. According with the literature, one of the most effective solution methods is represented by the Long Short-Term Memory Neural Networks (LSTM) [25], these are Recurrent Neural Networks (RNN) [26] with backward feedback propagation, which allow information to persist over time.

It is important to notice that the LSTM cell has two inputs and two outputs. As input it takes the input of the system and the previous state, just like the feedback loop of the vanilla RNN. As output it has the output of the system and the current state of the cell. Another fundamental characteristic about RNNs is that they possess the ability to be Multiple Input Multiple Output (MIMO). In the our problem we will use many to one configuration, we will use an input sequence to predict the next element of the sequence. In this section we present an application of the LSTM network to the water metering deployment to predict channel usage, indeed for these commercial devices no information about the transmission time are present. To obtain the input sequence to be fed to the LSTM network, we started from the fields of the LoRaWAN packets saved in the database. However a preprocessing of the series is needed before to fed the LSTM.

Before, we addressed an analysis of the transmission pattern of the network devices, it was deduced that the devices send the packets with a structure that is the combination of two different component, a regular part and a not regular part. Figure 13 shows a detail of the transmissions of a device, the x axis show frame counter of each packet, y axis shows the difference time respect the last packet time, or the inter arrival time. From the Figure 13, we can notice that regular part has a cycle of 9 packets, where the first 7 present an inter arrival time of 12 hours. While last 2 packets present a less temporal distance, but their inter arrival time distance sum is equal to 12 hours (4 and 8 in the case of the Figure). The inter arrival times of the eighth and ninth packet remain constant over time for the same device, but differ from device to device. The Figure also shows a case due to the lack of a packet that has not been received by the NS (at 258 frame counter value). The transmission pattern also has a not regular part, the study carried out has allowed to find the occasional existence of split times that do not respect the regular pattern, but which however have a certain repeated characteristic. Randomly one



(a) (b)

Figure 14: Frequency analysis for the inter arrival time for a sub set of water metering devices (a). Histogram of the error prediction for entire data set (300 devices) (b).

of the packets of the regular pattern can be replaced by two or more packets. The Figure 13 shows one of these situations (at 275 and 276 frame counter values).

To be sure that the pattern found comprehensively represents the behavior of the entire set of devices, a frequency analysis of the split times was made, Figure 14(a) shows the frequencies of the most frequent inter arrival times for a subgroup of devices (each device is represented with a different bar). Each bar has three different colored components, the height of which is equal to the inter arrival time value considered. From the figure, we can see that all the devices have a 12-hour inter arrival time, and two other values, the sum of these two is 12 hours, but they are different from device to device. For all devices, the sum of the occurrences of these 3 split inter arrival times represents more than 92% of the entire set of samples (the exact value is shown above each bar).

Figure 15 shows the inter arrival time series of 5 devices in succession (blue triangular). From the Figure it is easy to notice that the 12 hour (720 minutes) samples are common to all devices, while the other two inter arrival time is different in each device. In the figure, the isolated points (6 hour) represent the irregular parts of the sequence, that has been processed to better fit the prediction algorithm.

How see in the ED pattern study, time series of the distances between two successive received packets can present gaps, because not all the packets are correctly received (due to the channel fading and shadowing). The result is time series with non-continuous data, where, in a completely random way, parts of the sequence may be missing. Furthermore, when we are in the presence of highly corrupt time series, due to the absence of packets, it may be difficult or impossible to find a preprocessing function capable of correcting the sequence, these series can not be used for the learning process of the prediction model. For these devices, the forecast could be made using models obtained with devices of the same behavior profile. In other words, by using a subgroup of devices, we create models that are valid for all devices that have the same operating profile.

Prediction algorithms require a learning phase, in which the inter arrival time series is used for the generation of a model to be used later for the forecasting phase. We used

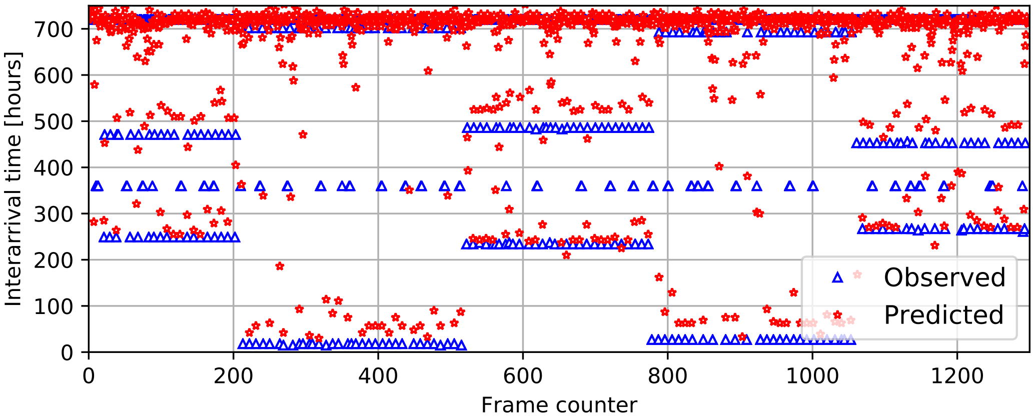


Figure 15: Qualitative result of the LSTM prediction for the test data set.

a subgroup of devices to generate a model capable of processing forecasts for all devices on the network. For the learning and for the forecasting phase, it is necessary to create a sequence of observations called features, features are used to predict future observation called target. Clearly, a single feature per sample is not enough to teach the model an abstraction that recognizes patterns never seen before. In order to create the features series, a data structure is used through a sliding window. Each input of the model is a series, in which each cell will represent the inter arrival time value as a feature, while the last column will be the target to be predicted (in the testing phase, the target will be used separately in comparing the predictions with the actual results). In the next row, the previous target is moved to the last feature column in order to insert the new target.

Following this approach, an LSTM neural network model was created with a many input and a single output, the model hold a single internal layer that has 120 hidden nodes. First, we training the model with a sequence retrieved from 100 ED (selected in random way) and we use the remaining EDs to test the model. Figure 15 shows a qualitative evaluation of the test results. The blue triangles represent the observed values, while the red stars represent the predicted inter arrival time values. As you can see from the Figure, for each device, the star markers mostly gather near the blue triangles.

In order to better show the results on the entire data set, a histogram of the vector of the forecast errors was made. Figure 14(b) shows the histogram of the vector of the errors of prediction of the complete sequence, from the figure it is possible to notice that more than 85% of the errors of prediction remain within 30 minutes.

5 Conclusion

In this chapter we have presented a LoRaWAN large scale scenario where LoRa modulation features are been exploited to improve network optimization. Network scalability issues are analyzed and three different approaches to improve the network performances has been presented. First, we present EXPLoRa-C, a 'capture aware water filling' algorithm to optimize LoRa devices parameters, where the the modulation channel effects are considered. Second, we present an application of the k-means ML algorithm to profile network devices. The clustering results are suitable groups of devices sharing similar behaviour. Third, we present an optimization strategy based on LSTM network algorithm to predict devices behavior and prevent future problems.

Extracted devices information via ML algorithms, clustering and prediction, are used by the optimization strategy that fit best parameters tuning. For example, is possible use balancing algorithm that starting to the extracted devices information find the best balancing among the EDs, in terms of how many and which EDs distributed for each SF.

All the presented solutions are been tested in the real LoRaWAN deployment, addressed by the Italy operator UNIDATA, that in the 2019 has collected 372.119.877 packets (2,25% are generated from EDs registered with the operator).

Acknowledgement

We thank UNIDATA S.p.A. who provided insight and expertise that greatly assisted our research. Part of the activities are related to the AMORE (Advanced Manufacturing e microelettronica per Ottimizzare le REti wireless delle cose) project. KETs, POR FESR Lazio 2014-2020.

References

[1] Zana Diaz Williams Eugenio Pasqua Saverio Romeo Raquel Artes Matt Wopata Knud Lasse Lueth, Padraig Scully. State of the iot & short-term outlook. 2018.

[2] Sornin, N., Yegin, A., et al. LoRaWAN 1.1 Specification. `https://lora-alliance.org/resource-hub/lorawantm-specification-v11`, 2017.

[3] Semtech. LoRa. EP2763321 from 2013 and U.S. Patent 7,791,415 from 2008.

[4] LoRa Alliance Technical CommitteeRegional Parameters Workgroup. LoRaWAN® Regional Parameters RP002-1.0.0. `https://lora-alliance.org/resource-hub/lorawanr-regional-parameters-rp002-100`, 2019.

[5] Daniele Croce, Michele Gucciardo, Ilenia Tinnirello, Domenico Garlisi, and Stefano Mangione. Impact of spreading factor imperfect orthogonality in lora communications. pages 165–179, 2017.

[6] Revision 1.0 Semtech Corporation. LoRaWAN – simple rate adaptation recommended algorithm, 2016.

[7] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne. Understanding the limits of lorawan. *IEEE Communications Magazine*, 55(9):34–40, 2017.

[8] Juha Petajajarvi, Konstantin Mikhaylov, Marko Pettissalo, Janne Janhunen, and Jari 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. 13:1–16, 03 2017.

[9] Thiemo Voigt, Martin Bor, Utz Roedig, and Juan Alonso. Mitigating inter-network interference in lora networks. In *Proceedings of the 2017 Int. Conf. on Embedded Wireless Systems and Networks*, EWSN &#8217;17, pages 323–328, 2017.

[10] Martin C. Bor, Utz Roedig, Thiemo Voigt, and Juan M. Alonso. Do LoRa Low-Power Wide-Area Networks Scale? In *19th ACM Int. Conf. on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, MSWiM '16, pages 59–67, 2016.

[11] Orestis Georgiou and Usman Raza. Low Power Wide Area Network Analysis: Can LoRa Scale? *IEEE Wireless Communications Letters*, 2017.

[12] K. Mikhaylov, Juha Petäjäjärvi, and Tuomo Hänninen. Analysis of capacity and scalability of the lora low power wide area network technology. In *22th European Wireless Conference*, pages 1–6, 2016.

[13] LoRaSim is a discrete-event simulator based on SimPy for simulating collisions in LoRa networks and to analyse scalability. http://www.lancaster.ac.uk/scc/sites/lora/.

[14] D. Croce, M. Gucciardo, S. Mangione, G. Santaromita, and I. Tinnirello. Impact of lora imperfect orthogonality: Analysis of link-level performance. *IEEE Communications Letters*, 22(4):796–799, 2018.

[15] D. Garlisi I. Tinnirello G. Bianchi, F. Cuomo. Capture aware sequential waterfilling for lorawan adaptive data rate, arXiv:1907.12360, [cs.NI], Nov 2019.

[16] S. Li, U. Raza, and A. Khan. How agile is the adaptive data rate mechanism of lorawan? pages 206–212 – IEEE Global Communications Conference (GLOBECOM), Dec 2018.

[17] 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)*, pages 1–8, 2017.

[18] Jacopo Maria Valtorta, Alessio Martino, Francesca Cuomo, and Domenico Garlisi. A clustering approach for profiling lorawan iot devices. In Ioannis Chatzigiannakis, Boris De Ruyter, and Irene Mavrommati, editors, *Ambient Intelligence*, pages 58–74, Cham, 2019. Springer International Publishing.

[19] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

[20] Alessio Martino, Antonello Rizzi, and Fabio Massimo Frattale Mascioli. Distance matrix pre-caching and distributed computation of internal validation indices in k-medoids clustering. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2018.

[21] Robert L. Thorndike. Who belongs in the family? *Psychometrika*, 18(4):267–276, 1953.

[22] David L Davies and Donald W Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227, 1979.

[23] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20, 1987.

[24] Jorge Luis. Hernandez Villapol. Spectrum Analysis and Prediction Using Long Short-Term Memory Neural Networks (LSTMs) and Cognitive Radios. *thesis, December 2017; Denton, Texas. (https://digital.library.unt.edu/ark:/67531/metadc1062877/: accessed May 28, 2020), University of North Texas Libraries.*

[25] F. A. Gers, J. Schmidhuber, and F. Cummins. Learning to forget: Continual prediction with lstm. *Neural Computation*, 2000.

[26] Christian W. Omlin and C. Lee Giles. Constructing deterministic finite-state automata in recurrent neural networks. 1996.

Wide Area Transmission Technologies for IoT

Andrea Abrardo, Giacomo Peruzzi, Alessandro Pozzebon

Department of Information Engineering and Mathematics

University of Siena, Siena, Italy

Abstract: *In this paper, the state of the art of Wide Area transmission technologies for IoT along with their application scenarios and requirements is discussed. In particular, the paper considers the main existing technologies, both license-based (i.e., cellular) and license-free (e.g., LoRaWAN and other Sub-GHz technologies). Then, future trends are analyzed, discussing in particular the advantages expected to be brought by the introduction of the upcoming 5G cellular technology in the Internet of Things (IoT) context, with the possibility of exploiting the edge computing paradigm and the slicing concept. In the second part of the paper, the advantages and drawbacks of the two main technological frameworks (e.g., licensed and license-free) are examined according to the different system requirements. Finally, two specific use cases are presented which are currently enabled by LoRa modulation.*

1 Introduction

The Internet of Things (IoT) has become a promising, valuable and robust paradigm throughout the recent years [1]. Indeed, the IoT is a hot topic nowadays either from a research point of view and from an application perspective as well: from smart cities to environmental monitoring, from home automation to the migration towards the Industry 4.0 domain, from healthcare to smart agriculture, from intelligent transportation systems to autonomous vehicles. All these scenarios are characterized by the presence of dumb objects of broad purpose (e.g., sensor nodes, vehicles, domestic appliances, industrial machineries and so on) with Internet connectivity and computational capabilities so as it is possible to control them and, at the same time, to receive notices from them.

Ideally, IoT devices should be designed to satisfy the following characteristics: high coverage, low cost and low power. Sadly, though, a simultaneous achievement of these features is tough or nearly impossible, hence a trade-off has to be reached in most of the cases. The application scenario determines which of the aforesaid features should be fulfilled the most: a pervasive monitoring infrastructure requires cheap sensor nodes, real time monitoring needs high coverage besides a relevant amount of data to be streamed while for devices relying only on batteries as source of energy to be as low power as possible is mandatory.

The Mobility Report issued by Ericsson in June 2019 [2] foresees that wide-area IoT devices, numbering at about one billion and representing a little more than one-tenth of all IoT deployments today, will grow consistently to reach 4.5 billion devices by 2024.

Short-range IoT devices, based on Bluetooth, ZigBee and 6LoWPAN or IPv6 technologies, represent about 90 percent of deployments today but they are growing at a much slower rate to reach 18 billion devices by 2024. Hence, despite short-range IoT devices are going to continue to exist in the market in the foreseeable future, a major attention will be given to wide-area IoT devices that are going to see an exponential growth.

Wide-area IoT is in general characterized by low power devices transmitting infrequent short bursts of data over a low-power wide area network (LPWAN). These devices typically do not have any external power supply (i.e., they run on batteries and they are installed in areas where frequent batteries substitution is an ineffective option). Accordingly, they are expected to be very power efficient. LPWANs can work in both licensed and unlicensed spectrum. Two of the most common LPWAN technologies that work in unlicensed spectrum are Sigfox [3] and Long-Range Wide Area Network (LoRaWAN) [4], while those working in licensed spectrum are narrowband (NB)-IoT [5] and LTE-M [6] or LTE machine type communication, sometimes referred-to as Cat-M1, both of which are specified by 3GPP in Release 13 and are generally known as cellular IoT. It is expected that LPWANs deployments based on unlicensed spectrum technologies will continue to exist and proliferate, despite at a much lower rate than cellular IoT, particularly with the development of 5G cellular systems [7].

Realizing the full IoT vision can only be achieved through the integration of different technologies. In particular, LPWAN technologies in the unlicensed spectrum are expected to address many of the requirements of massive IoT communications (e.g., low-cost devices with low energy consumption, extended coverage areas, high scalability for effective and fee-free deployment). On the other hand, IoT solutions in the licensed approved spectrum bands can become fundamental for massive to critical IoT use cases, where stringent requirements in terms of latency and reliability are present. This is particularly true when we consider the forthcoming 5G cellular system, that aims at addressing the requirements of a wide range of heterogeneous applications within a single network, exploiting the concept of network slicing [8, 9] and network virtualization [10, 11]. As an example, next generation 5G mobile networks are envisaged to ensure the presence of a massive number of devices and new services such as enhanced Mobile Broadband (eMBB), massive Machine-Type Communications (mMTC), traffic control and industrial control (Drone/Robot/Vehicle) and tactile Internet etc., so that critical communications and network operations can be efficiently supported. Moreover, cellular communication may exploit the security mechanism of existing cellular networks which are based on protecting basic connectivity and privacy of end-users, thus allowing to ensure an enhanced security mechanism addressing issues on authentication, authorization, and accounting (AAA) for heterogeneous interconnected IoT devices.

In this paper we discuss the state of the art of Wide Area transmission technologies employed within the context of different IoT application scenarios. In particular, referring to the existing technologies, we consider both license-based (i.e., cellular) and license-free (e.g., LoRaWAN and other Sub-GHz technologies). Then, future trends are analyzed, discussing in particular the expected advantages to be brought by the introduction of the upcoming 5G cellular technology in the IoT context, with the possibility of exploiting the edge computing paradigm and the slicing concept. In the second part of the paper, the advantages and drawbacks of the two main technological frameworks (e.g., licensed and license-free) are examined according to the different system requirements. Finally, two specific use cases are presented considering either their current development and their conceivable system requisites evolution due to 5G entrance within IoT communication technology framework.

This paper is drawn up as follows. In Section 2 several IoT scenarios are reviewed. Section 3 presents current IoT enabling technologies while Section 4 shows the upcoming ones. Section 5 reports a technologies comparison together with a glance to research challenges and future directions. Section 6 shows two common use cases which are currently enabled by LoRa modulation. Then, a discussion about the architectural transformations that will be entailed by the imminent advent of 5G, is provided. Finally, Section 7 reports conclusions and final remarks.

2 IoT Application Scenarios

LPWANs are massively employed within lots of operating scenarios. Most of the times, they are exploited for monitoring issues. Hereinafter, the main scenarios, along with some examples, are reported.

Smart Cities

There is a huge number of applications which aim at improving and enhancing the quality and lifestyle for city residents through gathering information which are relevant to their needs [12, 13]. As an example, a typical LPWAN for smart cities has sensor nodes deployed within the urban environment so as to monitor the occurrence and the extent of events in order to prevent car crashes, to safeguard inhabitants health, to efficiently manage assets and resources or to enhance the performances and the quality of urban services.

A typical municipal service that can take great advantage by the introduction of IoT is represented by waste management. As an example, in [14] a waste management and monitoring system based on LPWANs is presented. More specifically, a sensor node for the monitoring of the filling level of trashbins is developed. Indeed, having a precise knowledge of this quantity permits to better organize the depletion procedures. Another parameter that is recognized to be very important in the urban environment context is represented by the level of air pollutants. In [15, 16] two wireless sensor networks for wide area monitoring of air pollution are presented. However, even though the previous works on smart cities mainly rely on a single enabling standard, different IoT communication technologies may suit just as well [17]. More generally, the smart city scenario includes a wide range of applications, such as structural health, video surveillance, monitoring of parking spaces [18, 19], noise urban maps, smart lighting [20, 21], and many others [22]. Hence, they can be characterized by a wide range of requirements in regard of tolerable delay, data throughput, update frequency. Nevertheless, they are all nearly always categorized as massive IoT application scenarios, since they do not require stringent latency and reliability requirements.

Environmental Monitoring

The communication devices for this category of applications typically require very low power operation but, since they can also be deployed over very large areas, they may require very long communication range. This context encompasses many scenarios and some of them may overlap, for instance, with smart cities paradigms or transports and logistics as it is shown in [23] where an LPWAN is proposed for environmental monitoring of a high-speed rail station whose parameters of interest are temperature, humidity, luminance, and noise. In addition, similar networks for remote monitoring proved to be a reliable choice even in hostile environments as the marine one [24].

In some situations, environmental monitoring may require a network infrastructure composed by sensor nodes that are able to move within the monitored area. To this end, an interesting proposal is the one set forth in [25] where an LPWAN for environmental monitoring composed by unmanned aerial vehicles equipped with sensors and transceivers is proposed.

Another typical scenario for environmental monitoring envisages the evaluation of both air and water quality taken as indicators of ambience wholesomeness. To this regard, the study in [26] proposes a full IoT solution, while [27] shows a patented system for a water monitoring system.

Finally, another typical environmental monitoring scenario is given by the collection of meteorological parameters. In [29] a solution is proposed, in which a set of remotely available sensors forward meteorological data to the Internet.

The aforementioned scenarios are characterized by very loose constraints in terms of tolerable delay, updated frequency, data rate and, in many cases, packet loss ratio as well [22]. As such, they can fall in the category of massive IoT, but by no means they can be categorized as critical IoT scenarios.

Smart Agriculture

Farmers and breeders may capitalise on LPWANs and IoT technologies since they could enable the tracking of the conditions and of the health of cattle alongside sending data related to soil (e.g., temperature and humidity) so as to control fertilising procedures or irrigation systems as well. Such a task may be fulfilled as suggested in [30], where an IoT smart water management platform for irrigation control is presented. Other interesting applications in this context are presented in [31] and [32], where different baseline technologies are respectively proposed, or even in [33] where a monitoring system for a vineyard based on cellular technologies is set up. Finally, [34] shows a smart agriculture information system which operates thanks to cellular technologies.

Energy efficiency is perhaps the most important aspect of this scenario, in particular when most IoT devices are battery powered and are expected to be operational for a very long period without human intervention. On the other hand, no stringent requirements in terms of latency and data rate are required for this scenario.

Transports, Tracking and Logistics

IoT market received a huge contribution due to the development of LPWANs for transports and logistics monitoring. Thanks to the enabling technologies it is feasible to track vehicles (e.g., from cars or buses to boats and trains) and mobile assets shipped all over the world (e.g., parcels, luggage, crates, containers and packages).

Tracking systems for public transport can be implemented by means of modulations enabling LPWANs rather than the classic GPS techniques as it is shown in [35]: transmitters are installed onto buses and receivers are set up onto bus stops. Hence, the communication takes place whenever a bus enters in an area covered by the signal emitted from the bus stop. Indeed, in urban areas having dense bus stops, a real time tracking could be accomplished avoiding standard GPS thus lowering devices costs.

Tracking could be also a topic which can be considered in a broader sense. Indeed, most of the time vehicles or assets are subject to tracking. However, also runners involved in cross country races may be tracked by making use of LPWAN networks [36].

One of the fundamental requirements for this class of applications is global coverage and mobility. As such, despite the aforementioned related works deal with non cellular technologies since they have been proposed as an efficient and low cost solution in some cases, it is expected that LPWAN cellular communications will represent the most viable option. Indeed, an instance of tracking by exploiting cellular technology can be retrieved in [37].

Industrial Monitoring

Unlike the previous IoT scenarios, the Industrial Internet of Things (IIoT) (especially in process monitoring and control) requires in general very stringent requirements in terms of both latency and reliability [38]. As a matter of fact, manufacturing industries were one of the pioneers in adopting Time Sensitive Networking (TSN) technology through wired connections. At the same time, they have seized Machine to Machine (M2M) connectivity for decades to enhance supply chain efficiency whilst, nowadays, it is exploited to monitor devices as well as to signal and prevent faults. An example of fault in industrial plants could be chemical emissions and an instance of this type of fault detection is shown in [39]. Another study [40] considers wearable sensor nodes for environmental monitoring whose target is to supervise the concentration of harmful gases in industrial plants. Also machine vibrations could be a phenomenon to be monitored within industrial contexts: a solution could be to set up a hybrid network infrastructure enabled by both non-cellular and cellular technologies [41]: the former conveys data from sensor nodes to the gateways while the latter is in charge of forwarding such data to the Internet. Moreover, cellular technologies may also enable the monitoring of water level in industrial tanks [42]. On the other hand, in this application scenario several new challenging aspects emerge: thousands of nodes generate GB/s of aggregate data (massive capacity), automation processes set very stringent requirements in terms of reliability (up to the order of 10-9), communication latency and jitter (less than 100μs and about 1μs, respectively), energy management of the nodes can be over-constrained (e.g., battery-less devices). All these aspects lead to system requirements that cannot be fulfilled by current and near-future wireless technologies, such as the ones defined in the 5G standard. In fact, the current trend to satisfy such requirements, based on cell densification, massive MIMO transmission, and the exploitation of higher frequency bands, is not enough. The IMT-2030, namely the "connectivity beyond 5G", is already defining the requirements for the future connectivity that must satisfy simultaneously the ultra-low-latency, ultra-massive and ultra-reliability requirements, while maintaining an affordable complexity and low energy consumption. To tackle these objectives, a technological and design paradigm shift is needed.

Smart Healthcare

The IoT potentialities are expected to strongly impact and influence the medical and healthcare system [43]. In particular, IoT in healthcare can be seen as a viable mean for facilitating the process of collecting patient data and providing an insurmountable quantity of data that can be used to advance scientific studies in disease cures, diagnosis, etc. To this regard, in [44, 45] low power wearable devices equipped with sensors are proposed to serve as data sourcing platforms for doctors and service providers. In [46, 47], the main enabling technologies (either licensed-based and licensed-free), challenges and opportunities for smart healthcare and remote healthcare monitoring are surveyed. The authors claim that multiple sensors for detecting vital parameters (e.g., respiratory rate, heart rate, blood pressure, body temperature and so on) could be part of an LPWAN that stores such data into a cloud on which machine learning algorithms are applied so that medical staff may take advantage by. Some interesting solutions for monitoring vital parameters, detecting falls on elderly people (and many other criteria) based on non-cellular and cellular technologies, respectively, are presented in [48, 49, 50]. Moreover, [51] proposes a way so to set up smart hospitals solely relying on cellular enabling technologies.

In smart healthcare scenarios, relatively low latency in the order of few seconds are generally needed while stringent requirements in terms of data rate and update frequency can be needed in some cases (e.g., real-time emergency response and remote diagnostics).

Vehicular Communications

Intelligent transportation systems (ITS) are used to ensure that the transportation network is efficiently monitored and controlled [52, 53]. In particular, ITS aims at ensuring that system reliability, availability, efficiency and safety of the transportation network are guaranteed. Despite there are some solutions for implementing vehicular monitoring platforms base on non-cellular LPWANs [54], this is a typical application scenario where the advent of 5G, and beyond 5G cellular communication systems, will play a fundamental role, particularly with the view of on-going potential development in autonomous cars (i.e., self-driving cars). Indeed, in this case, ultra reliable and low latency communications are mandatory [55, 56, 57, 58, 59, 60].

3 Current IoT Enabling Technologies

These technologies can make use of the spectrum belonging to the unlicensed ISM bands or to the licensed cellular spectrum. In the first case, their carrier frequencies are region depending. Therefore, hardware devices must be thoroughly chosen bearing in mind the place in which the network will be deployed. Moreover, for such technologies some of the standards are free whilst some other may require subscription fees. Finally, at the end of this Section, Table 1 recaps some of the previously mentioned related works highlighting the relative operating scenarios and enabling technologies.

3.1 Sigfox

Sigfox is a French company founded more than 10 years ago and Sigfox networks are presently operating all over the world. Commonly with other IoT technologies, also

Sigfox networks work within the Sub-GHz spectrum. Users have to pay a subscription which offers different plans. The network topology is a star one enabling both uplinks and downlinks. Sigfox is characterized by a small size payload (i.e., at most $12\,B$) that turns to be a major drawback, and a limited number of possible transmitting packets per day, depending on the subscription plan. Sigfox also offers a network infrastructure and a cloud service to retrieve, and possibly analyze, the transmitted data. In doing so, users have only to concern about the sensor nodes since all the network side is in charge of Sigfox itself. It operates within ISM bands requiring very narrow band (i.e., only $100\,Hz$) and providing very slow data-rates (i.e., only $100\,bps$). Sigfox takes advantage of the Binary Phase Shift Keying (BPSK) modulation and of the Random Frequency Time Division Multiple Access (RFTDMA) technique to access the channel.

3.2 Ingenu

At its release, more than a decade ago, Ingenu was an innovative technology in the LPWANs scenery. It utilizes higher ISM frequencies than Sigfox. Indeed, it operates at $2.4\,GHz$ that is the same frequency of WiFi or Bluetooth. The advantage of this band is that it is globally available, so that developers do not have to consider in what regions their products will be deployed, differently from Sigfox. Moreover, $2.4\,GHz$ band offers broader bandwidth than sub-GHz ISM bands. Along with the Differential Binary Phase Shift Keying (D-BPSK) modulation, the core of the Ingenu LPWAN is its proprietary and patented technology: the Random Phase Multiple Access (RPMA) [61]. It is both a physic and Medium Access Control (MAC) layer especially developed by Ingenu to satisfy the requirements of an LPWAN: extended battery lifetime, robustness towards interference and wide coverage and high capacity since a single RPMA access point may cover up to $450\,km^2$ and may handle up to 530000 messages per hour. RPMA is also capable to address, among many other features, bi-directional communication and broadcast transmission. Ingenu has built the first wireless Machine Network: the largest IoT network in the world dedicated to connectivity for machines, that has been set up in more than 30 cities (most of them in the US). Moreover, similarly to Sigfox, Ingenu provides all the network infrastructure.

3.3 Weightless

Weightless comprises a set of wireless standards for LPWAN conceived for exchanging data between sensor nodes and base stations [62]. It has been managed by the English Weightless Special Interest Group since 2012. At its early days, three standards were released (i.e., Weightless-P, Weightless-N, Weightless-W): two of them were deprecated in favour of the remaining one. Weightless-N provided only for uplinks, hence it was a mono-directional communication standard. Weightless-W was intended to operate within the unused frequencies belonging to the TV bands (i.e., $470 \div 790\,MHz$). Weightless-P is the only surviving standard, indeed it is simply named as Weightless, since it is bi-directional, it needs a narrow band to run and it exploits all the frequencies belonging to the ISM unlicensed sub-GHz bands (i.e., $163\,MHz$, $433\,MHz$, $470\,MHz$, $780\,MHz$, $868\,MHz$, $915\,MHz$ and $923\,MHz$). In addition, Weightless is also an open standard that makes use of both the Gaussian Minimum Shift Keying (GMSK) and the Offset Quadrature

Phase Shift Keying (OQPSK) modulations and of the Time-Division Multiple Access (TDMA) scheme to access the communication channel.

3.4 LoRa

Long Range (LoRa) modulation is a patented digital wireless communication standard developed in 2012 and owned by the American company Semtech. It is based on the Chirp Spreading Spectrum (CSS) modulation and exploits the Additive Links On-Line Hawaii Area (ALOHA) technique to access the communication channel (i.e., transmission may occur at any time). Nowadays it is managed and controlled by the LoRa Alliance which is a consortium made up by over 500 companies dealing both with hardware and software. Such an institution also released the Long Range Wide Area Network (LoRaWAN) standard (i.e., a communication protocol referring to the MAC layer which is based onto the LoRa modulation). In contrast with the proprietary nature of the modulation by which Semtech receives royalties from chip vendors that sell LoRa modules, LoRaWAN specification is openly available. LoRaWAN is not the only network built on top of the LoRa modulation. Indeed, Link Labs has developed a competing LoRa based LPWAN.

LoRa is a long range wireless communication technology, with coverage ranging from few kilometers in urban areas, up to tens kilometers in rural environments. Similarly to the other LPWAN technologies, LoRa is designed for single-hop communications within star topologies networks. Nevertheless, there exists a recent study that shows the feasibility of a mesh network by means of this modulation [63].

3.5 Cellular IoT

Two cellular IoT technologies, namely narrow-band IoT (NB-IoT) and LTE-M, have been designed within the 3GPP release 13 standardization activity. Such technologies are deployed as part of the LTE network today and are expected to work with 5G networks in the future using the non-standalone (NSA) architecture, where the legacy LTE core network is used for control-plane functionality such as initial access, paging, and mobility, while the radio access network (RAN) functionalities are carried out through the 5G NR [28]. However,, as dictated by the 3GPP Release 15 (i.e., the first 5G standard), in the first phase the radio interface will still be LTE-based, while the core network will be 5G service based architecture (SBA). The two cellular technologies are similar, in that they are both designed with the aim of supporting low device complexity, massive connection density, low device power consumption, low latency and extended coverage. The main difference is that LTE-M supports mobility, higher bandwidth (1.4 MHz) and data rates (up to 1 Mbps), whereas NB-IoT supports 180 KHz bandwidth (one resource block of LTE) with data rates lower than 100 kbps, and it does not support mobility. On the other hand, NB-IoT works at lower frequency bands making it excellent for indoor uses. Finally, NB-IoT has much higher latency (i.e., from 1.5 to 10 s versus 50-100 ms of LTE-M). Given that, it is clear that such cellular technologies can be considered the licensed counterparts to unlicensed LPWAN technologies, in that they are designed for applications with no stringent requirement in terms of latency and data rate. In other terms, all the aforementioned technologies are suitable for massive IoT while they cannot support critical IoT scenarios.

Table 1: literature review recap sorted out by operating scenarios and enabling technologies.

Technologies	Operating Scenarios						
	Smart Cities	Environmental Monitoring	Smart Agriculture	Transports, Tracking and Logistics	Industrial Monitoring	Smart Healthcare	Vehicular Communications
Sigfox		[29]	[32]				
Ingenu	[20]	[27]					
Weightless	[18]						
LoRa	[14] [15] [23] [64]	[23] [24] [25]	[30] [31]	[23] [35] [36]	[39] [41]	[48]	[54]
NB-IoT	[16] [19] [21]		[34]		[41] [42]	[49] [51]	[60]
LTE-M			[33]	[37]		[45] [50]	[37] [59]

4 Upcoming IoT Enabling Technologies

The possibility of implementing critical IoT scenarios together with massive IoT is recognized as one of the fundamental step towards the full realization of the IoT vision, including the full range of application scenarios. Among them, communication with vehicles for self-driving cars, industrial automation and process control, critical healthcare including tactile Internet for tele-surgery, augmented and virtual reality for gaming, tourism, teaching, etc., call for the broad implementation of the 5G vision where a single network may provide a wide set of use cases. Comparing with existing unlicensed options for IoT, there is an urgent need for supporting ultra-reliable and low latency (URLLC) communications, or critical communications. To this aim, 3GPP release 15 [65] and 16 [66] have defined solutions for new radio (NR) at both the physical layer and upper layers to reduce latency and improve system reliability.

The main innovations brought by 5G with respect to 4G are represented by: (*i*) The definition of a flexible frame structure, with shorter slots for low latency transmissions; (*ii*) Non-slot based scheduling and mini-slot scheduling, where URLLC traffics can be scheduled within shorter version of slots; (*iii*) Semi-persistent scheduling for downlink transmissions, where a gNB may transmit a packet towards the device without sending the information about the scheduling decision (grant mechanism); (*iv*) Grant free transmission in the uplink, where a device can transmit towards the gNB without the sending request/grant mechanism; (*v*) Multiplexing of URLLC and eMBB, where an urgent URLLC packet can be scheduled occupying resources already assigned to eMMB without waiting for the next scheduling opportunity; (*vi*) Enhancement of the downlink physical control channel to increase reliability (e.g., by repetition); (*vii*) Enhancements of acknowledgement mechanism, to reduce the latency for URLLC traffics.

At the same time, the Release 16 working activity is elaborating the possibility of extending 5G spectrum allowing access to shared and unlicensed spectrum with the goal of providing more capacity, higher spectrum utilization, and new deployment scenarios. It will benefit mobile operators with licensed spectrum, but also create opportunities for those without licensed spectrum to take advantage of 5G technologies.

The aforementioned innovations allows to increase the flexibility of the radio access network (RAN), thus opening the way for the implementation of a network that can cope with the requirements of different application scenarios, such as those categorized as massive and critical IoT. Moreover, 5G allows to implement virtual networks (i.e., network slicing), in order to provide connectivity more adjusted to specific needs. The creation of virtual programmable networks will increase the service agility that is the possibility of re-configuring a part of the network by applying, for example, different latency or prioritizing them in the connection to the network so that they cannot be affected by possible overloads of the mobile network (e.g., for guaranteeing reliability and latency).

5 Technology Comparison and Research Challenges

In this Section we aim at providing useful guidelines for the selection of the most proper IoT technology, comparing in particular cellular (i.e., licensed) and unlicensed technologies, given the current maturity level of the different options. As a matter of fact, 5G promises to represent a sort of panacea in the next future, but, for the time being, NB-

IoT is the only licensed IoT communications technology available on the market. Hence, given the current situation, the application scenario is the main discriminant and in most of the cases it is the one that dictates the technology to exploit.

For instance, adopting cellular-based solutions is quite inconvenient whenever a multitude of sensor nodes have to be deployed since such a choice entails pretty high subscription costs due to the high number of SIMs to be bought and maintained. On the other hand, adopting unlicensed technologies (e.g., LoRa) and providing Internet connectivity to the gateways by means of cellular solutions, in case no WiFi or Ethernet facilities are available, could be a bold move since it is expected that far less gateways than sensor nodes are required.

On the contrary, in situations in which a high data-load has to be sent and received avoiding data losses, or in real-time applications as well, cellular technologies definitely suit better. Indeed, due to the fact that they exploit licensed frequencies and since they lean on robust and tested network infrastructures (i.e., the cellular ones), achieving no data loss and real-time data streams is smoothly feasible. On the other hand, since telecom operators have not provided so far ad-hoc price plans for IoT users, adopting cellular solutions might result in pretty high running costs. Moreover, existing communication standards are often too energy demanding for sensor nodes lacking of renewable sources of power.

Given the above, what will the future hold for IoT communication technologies? Which one will prevail amid licensed-free and licensed-based solutions? Albeit there is a significant hype on 5G, will it revolutionize the current technology framework? How unlicensed solutions will keep up to it? Without the necessity of staring at the crystal ball, 5G will surely break the equilibria that implicitly have been formed so far since its specifications ensure global coverage, high data-rates, low power consumption and ad-hoc price plans for different slices of the network. With regard to coverage, non cellular communications cannot compete with cellular ones and this is one of the most valuable pros of 5G. However, LoRaWAN standard is on its way to face and bridge such a gap. Indeed, quite recently two different projects [67, 68] have been started with the aim to enable LoRa satellite connectivity. The basic idea is to employ PocketQubes (i.e., $5x5x5\,cm$ satellites) acting as LoRa gateways or repeaters. In particular, [67] exploits LoRa modulation on licensed frequencies (so, sensor nodes have to be re-designed in order to transmit in those bands and users have to pay a fee to broadcast in such a spectrum), while [68] proposes a solution where the same ISM bands of the LoRaWAN standard are exploited for communicating with the satellites, even if users have to pay for each transmitted message towards the space.

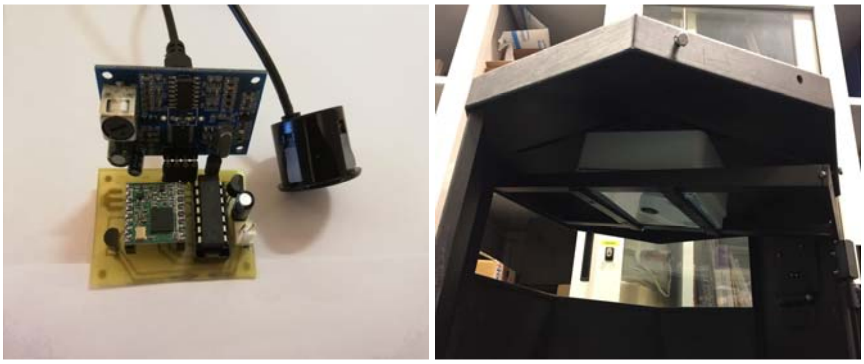
6 Two LoRa-based use cases

In the following, we go into much details in the description of two use cases that have been developed at the Department of Information Engineering and Mathematical Sciences (DIISM) of the University of Siena. The first one belongs to a Smart City scenario and deals with smart waste management [14, 64]. In particular, it faces the problem of solid waste monitoring by using an innovative IoT node architecture so as to monitor the filling of garbage bins spread all over the city. The second one falls both in the environmental monitoring scenario as well as the paradigm of Industry 4.0 for seafarms [24]. More specifically, it consists in providing an IoT solution for remote monitoring of overboard sea farms.

Concerning the first use case, the system requirements for what concerns the sensing platforms (i.e., the so-called ”Smart Bins”) can be summarized as follows: (*i*) Long transmission range; (*ii*) Very low per unit cost, owing to the massive number of required nodes; (*iii*) Low frequency update; (*iv*) Very low power consumption.

The system designed for fulfilling all the above requirements is based on a city-scale LoRaWAN network infrastructure, allowing transmission ranges of nearly 1.5 km in the considered urban environments of the cities of Florence and Siena, Italy.

The architecture of the sensor nodes used for this application is very simple as well. It includes a low power microcontroller (i.e., an ATtiny84 produced by Microchip) driving a LoRa transceiver (i.e., an RFM95 produced by HopeRF embedded in a breakout board produced by Adafruit) that are both powered by a battery pack (i.e., 4 AA lithium batteries providing 6 V and 4800 mAh) passing through a 5 V voltage regulator (i.e., a A7805 produced by Texas Instrument). Three sensors are integrated in the module: an ultrasonic distance sensor for the trash level detection, a temperature sensor and a tilt sensor for overturning detection. This system features a very low cost thanks to the reduced number of electronic components. Duty-cycling policies are actuated at a microcontroller level, thus allowing to notably reduce power consumption and to increase the node life time up to a couple of years. The prototype of the sensor node is shown in Fig. 1a, while in Fig. 1b it was housed within an IP56 box during some laboratory tests simulating its usage inside real trash bins.



(a) (b)

Figure 1: Sensor node for smart waste management: (a) prototype of the sensor node without batteries and antenna, (b) laboratory tests inside a real trash bin.

The second use case has slightly different system requirements. The main difference concerns the number of involved nodes, that in this case is limited to few units. The other requirements are similar, including the very low power consumption, since even though the seamark buoy (which is exploited to install all the elements composing the sensor node) can be exploited to install photovoltaic cells for energy harvesting, such a technique is specifically adopted for running the required sensors (which are non-standard ones due to their operating environment) rather than for the transmitting electronics that, instead, was designed to be battery powered.

Also in this case, we adopted a LoRaWAN network infrastructure. The main goal is that of conveying a bunch of environmental parameter from offshore to ashore, which are both related to air and sea, so as to remotely monitor the overboard productive plants as well as fish wholesomeness. Such data are sampled by employing suitable sensors (e.g., marine temperature probes and current meters) that are installed on board of seamark buoys. These sensors are managed by an ad-hoc control system which is connected to a tailor-made RS232-LoRaWAN interface (see Fig. 2) whose purpose is to arrange data coming from sensors in LoRaWAN packets and to broadcast such information ashore by covering an $8.33\,km$ distance. Test results proved the feasibility and the robustness of the proposed solution [24]. Hence, two gateways (see Fig. 3) were positioned ashore in order to establish a space diversity scheme so as to enhance the delivery reliability.

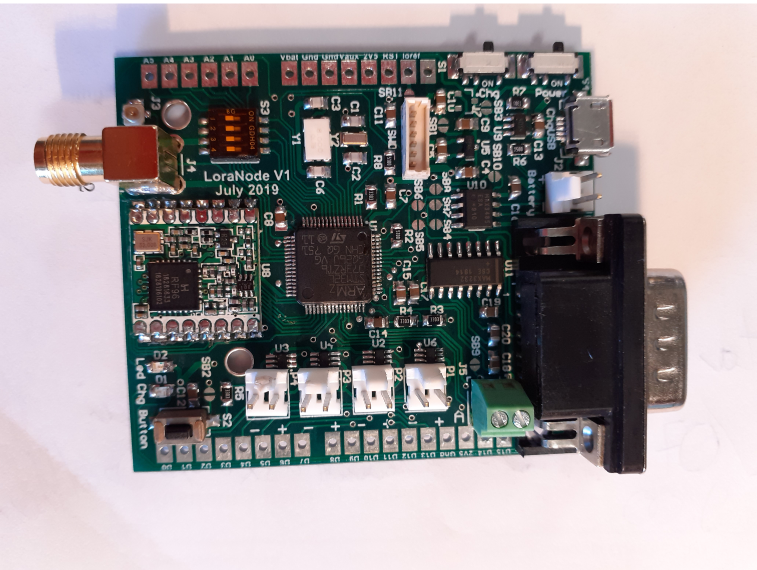


Figure 2: RS232-LoRaWAN interface.

In order to fulfill the very low power consumption requirement, we adopted the following extremely low power components: an STM32L073 microcontroller produced by STMicroelectronics and an RFM95 LoRa transceiver manufactured by HopeRF and miscellaneous low power electronics. Finally, the system is powered by a $3600\,mA$ lithium ions battery providing $4.2\,V$ ensuring nearly 500 days lifetime.

The rationale behind the choice of LoRa in both scenarios can be traced back to the considerations drawn in the previous Section. In particular, as for the first scenario, the number of nodes is very high and hence, adopting cellular solutions might result in pretty high running costs (as long as telecom operators do not provide ad-hoc price plans for IoT users). Moreover, existing cellular solutions are still too energy demanding, whereas LoRa allows to design ad-hoc architectures specifically tailored to save energy, that was one of the main requirements for both scenarios. On the other hand the situation can

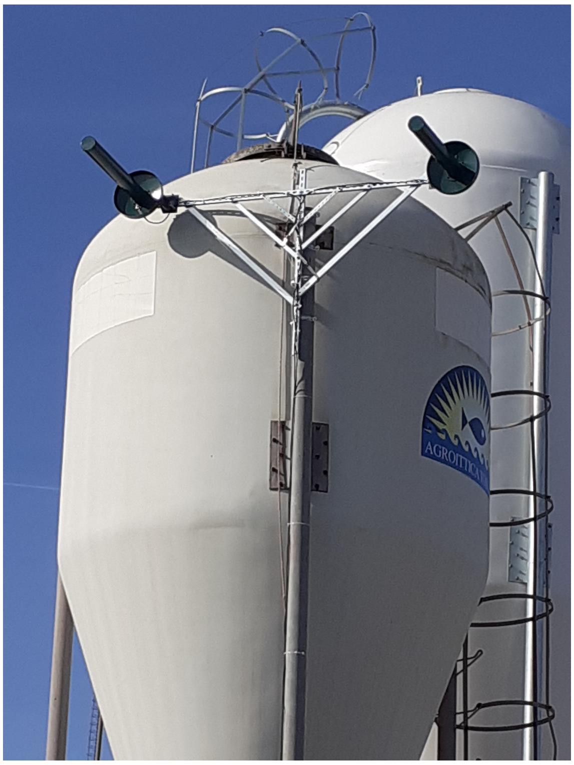


Figure 3: Gateways and their antennas in the installation site.

drastically change in the next future, and hence both systems could be easily re-designed to encompass cellular communications. In particular, the second scenario could take great advantage from the use of a cellular solution since we could avoid deploying ashore gateways for providing connection to the Internet.

7 Conclusions

The aim of this paper was to propose a short survey on the state-of-the-art of long range data transmission technologies, both license-based (i.e., cellular) and license-free (e.g., LoRaWAN and other Sub-GHz technologies), along with their application scenarios and requirements. Due to the upcoming introduction of 5G cellular technologies, all these scenarios are expected to undergo a rapid change in their operating principles and technological layout. It is clear that this change is strictly dependent on the system requirements of each specific application. Hence, we have analyzed the future trends, discussing in particular the advantages expected to be brought by 5G with the possibility of exploiting the edge computing paradigm and the slicing concept. Finally, two specific use cases developed at the Department of Information Engineering and Mathematical Sciences (DIISM) of the University of Siena are presented, discussing the rationale for choosing LoRa as the communication technology.

References

[1] Palattella, M. R., Dohler, M., Grieco, A., Rizzo, G., Torsner, J., Engel, T., & Ladid, L. (2016). Internet of things in the 5G era: Enablers, architecture, and business models. IEEE Journal on Selected Areas in Communications, 34(3), 510-527.

[2] Ericsson report 2019: https://www.ericsson.com/en/mobility-report/reports Accessed on 10 February 2020.

[3] Sigfox. Online Availability: htpp://www.sigfox.com. Accessed on 10 February 2020.

[4] de Carvalho Silva, J., Rodrigues, J. J., Alberti, A. M., Solic, P., & Aquino, A. L. (2017, July). LoRaWAN—A low power WAN protocol for Internet of Things: A review and opportunities. In 2017 2nd International Multidisciplinary Conference on Computer and Energy Science (SpliTech) (pp. 1-6). IEEE.

[5] Popli, S., Jha, R. K., & Jain, S. (2018). A survey on energy efficient narrowband internet of things (NBIoT): Architecture, application and challenges. IEEE Access, 7, 16739-16776.

[6] Ikpehai, A., Adebisi, B., Rabie, K. M., Anoh, K., Ande, R. E., Hammoudeh, M., Gacanin, H., & Mbanaso, U. M. (2018). Low-power wide area network technologies for internet-of-things: A comparative review. IEEE Internet of Things Journal, 6(2), 2225-2240.

[7] Agiwal, M., Roy, A., & Saxena, N. (2016). Next generation 5G wireless networks: A comprehensive survey. IEEE Communications Surveys & Tutorials, 18(3), 1617-1655.

[8] Foukas, X., Patounas, G., Elmokashfi, A., & Marina, M. K. (2017). Network slicing in 5G: Survey and challenges. IEEE Communications Magazine, 55(5), 94-100.

[9] Zhang, H., Liu, N., Chu, X., Long, K., Aghvami, A. H., & Leung, V. C. (2017). Network slicing based 5G and future mobile networks: mobility, resource management, and challenges. IEEE communications magazine, 55(8), 138-145.

[10] Feng, J., Zhang, Q., Dong, G., Cao, P., & Feng, Z. (2017, March). An approach to 5G wireless network virtualization: Architecture and trial environment. In 2017 IEEE Wireless Communications and Networking Conference (WCNC) (pp. 1-6). IEEE.

[11] Abdelwahab, S., Hamdaoui, B., Guizani, M., & Znati, T. (2016). Network function virtualization in 5G. IEEE Communications Magazine, 54(4), 84-91.

[12] Gea, T., Paradells, J., Lamarca, M., & Roldan, D. (2013, July). Smart cities as an application of internet of things: Experiences and lessons learnt in barcelona. In 2013 Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (pp. 552-557). IEEE.

[13] Jin, J., Gubbi, J., Marusic, S., & Palaniswami, M. (2014). An information framework for creating a smart city through internet of things. IEEE Internet of Things journal, 1(2), 112-121.

[14] Cerchecci, M., Luti, F., Mecocci, A., Parrino, S., Peruzzi, G., & Pozzebon, A. (2018). A low power IoT sensor node architecture for waste management within smart cities context. Sensors, 18(4), 1282.

[15] Rossi, M., & Tosato, P. (2017, July). Energy neutral design of an IoT system for pollution monitoring. In 2017 IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems (EESMS) (pp. 1-6). IEEE.

[16] Duangsuwan, S., Takarn, A., & Jamjareegulgarn, P. (2018, September). A Development on Air Pollution Detection Sensors based on NB-IoT Network for Smart Cities. In 2018 18th International Symposium on Communications and Information Technologies (ISCIT) (pp. 313-317). IEEE.

[17] Pham, T. L., Nguyen, H., Nguyen, H., Bui, V., & Jang, Y. M. (2019, October). Low Power Wide Area Network Technologies for Smart Cities Applications. In 2019 International Conference on Information and Communication Technology Convergence (ICTC) (pp. 501-505). IEEE.

[18] https://www.nwave.io/parking-technology Accessed on 17 March 2020.

[19] Shi, J., Jin, L., Li, J., & Fang, Z. (2017, September). A smart parking system based on NB-IoT and third-party payment platform. In 2017 17th International Symposium on Communications and Information Technologies (ISCIT) (pp. 1-5). IEEE.

[20] Myers, T. J., Werner, D. T., Sinsuan, K. C., Wilson, J. R., Reuland, S. L., Singler, P. M., & Huovila, M. J. (2013). U.S. Patent No. 8,477,830. Washington, DC: U.S. Patent and Trademark Office.

[21] Zhao, L., Gao, Q., Wang, R., Fang, N., Jin, Z., Wan, N., & Xu, L. (2018, June). Intelligent Street Light System Based on NB-IoT and Energy-saving Algorithm. In 2018 3rd International Conference on Smart and Sustainable Technologies (SpliTech) (pp. 1-6). IEEE.

[22] Zanella, A., Bui, N., Castellani, A., Vangelista, L., & Zorzi, M. (2014). Internet of things for smart cities. IEEE Internet of Things journal, 1(1), 22-32.

[23] Li, W., Liu, G., & Choi, J. (2019). Environmental monitoring system for intelligent stations. Concurrency and Computation: Practice and Experience, e5131.

[24] Parri, L., Parrino, S., Peruzzi, G., & Pozzebon, A. (2019). Low Power Wide Area Networks (LPWAN) at Sea: Performance Analysis of Offshore Data Transmission by Means of LoRaWAN Connectivity for Marine Monitoring Applications. Sensors, 19(14), 3239.

[25] Tarab, H. (2018). Real Time Performance Testing of LoRa-LPWAN Based Environmental Monitoring UAV System.

[26] Doni, A., Murthy, C., & Kurian, M. Z. (2018). Survey on multi sensor based air and water quality monitoring using IoT. Indian J. Sci. Res, 17(2), 147-153.

[27] Sinsuan, K. C., Myers, T. J., Cohen, L. N., Werner, D. T., Hughes, M., Boesel, R. W., & Singler, P. M. (2014). U.S. Patent No. 8,831,069. Washington, DC: U.S. Patent and Trademark Office.

[28] E. Dahlman, S. Parkvall, J. Skold, ”5G NR, The Next Generation Wireless Access Technology,” Academic Press, 2018.

[29] Joris, L., Dupont, F., Laurent, P., Bellier, P., Stoukatch, S., & Redouté, J. M. (2019). An Autonomous Sigfox Wireless Sensor Node for Environmental Monitoring. IEEE Sensors Letters, 3(7), 01-04.

[30] Kamienski, C., Soininen, J. P., Taumberger, M., Dantas, R., Toscano, A., Salmon Cinotti, T., Filev Maia, R., & Torre Neto, A. (2019). Smart water management platform: Iot-based precision irrigation for agriculture. Sensors, 19(2), 276.

[31] Davcev, D., Mitreski, K., Trajkovic, S., Nikolovski, V., & Koteli, N. (2018, June). IoT agriculture system based on LoRaWAN. In 2018 14th IEEE International Workshop on Factory Communication Systems (WFCS) (pp. 1-4). IEEE.

[32] Zhang, X., Andreyev, A., Zumpf, C., Negri, M. C., Guha, S., & Ghosh, M. (2017, May). Thoreau: A subterranean wireless sensing network for agriculture and the environment. In 2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS) (pp. 78-84). IEEE.

[33] https://www.bce.ca/news-and-media/releases/show/Bell-Huawei-and-BeWhere-bring-new-Internet-of-Things-solution-to-Ontario-winery-1 Accessed on 17 March 2020.

[34] Yao, Z., & Bian, C. (2019). Smart Agriculture Information System Based on Cloud Computing and NB-IoT. DEStech Transactions on Computer Science and Engineering, (cisnrc).

[35] James, J. G., & Nair, S. (2017, November). Efficient, real-time tracking of public transport, using LoRaWAN and RF transceivers. In TENCON 2017-2017 IEEE Region 10 Conference (pp. 2258-2261). IEEE.

[36] Sendra, S., Romero-Díaz, P., García-Navas, J. L., & Lloret, J. (2019). LoRa-Based System for Tracking Runners in Cross Country Races.

[37] Driusso, M., Marshall, C., Sabathy, M., Knutti, F., Mathis, H., & Babich, F. (2016). Vehicular position tracking using LTE signals. IEEE Transactions on Vehicular Technology, 66(4), 3376-3391.

[38] Da Xu, L., He, W., & Li, S. (2014). Internet of things in industries: A survey. IEEE Transactions on industrial informatics, 10(4), 2233-2243.

[39] Addabbo, T., Fort, A., Mugnaini, M., Parri, L., Parrino, S., Pozzebon, A., & Vignoli, V. (2018, April). An IoT framework for the pervasive monitoring of chemical emissions in industrial plants. In 2018 Workshop on Metrology for Industry 4.0 and IoT (pp. 269-273). IEEE.

[40] Antolín, D., Medrano, N., Calvo, B., & Pérez, F. (2017). A wearable wireless sensor network for indoor smart environment monitoring in safety applications. Sensors, 17(2), 365.

[41] Gao, S., Zhang, X., Du, C., & Ji, Q. (2019). A Multichannel Low-Power Wide-Area Network With High-Accuracy Synchronization Ability for Machine Vibration Monitoring. IEEE Internet of Things Journal, 6(3), 5040-5047.

[42] Anand, S., & Regi, R. (2018, February). Remote monitoring of water level in industrial storage tanks using NB-IoT. In 2018 International Conference on Communication information and Computing Technology (ICCICT) (pp. 1-4). IEEE.

[43] Pasluosta, C. F., Gassner, H., Winkler, J., Klucken, J., & Eskofier, B. M. (2015). An emerging era in the management of Parkinson's disease: wearable technologies and the internet of things. IEEE journal of biomedical and health informatics, 19(6), 1873-1881.

[44] Laplante, P. A., & Laplante, N. (2016). The internet of things in healthcare: Potential applications and challenges. It Professional, 18(3), 2-4.

[45] Shakhakarmi, N. (2015). Next generation wearable devices: smart health monitoring device and smart sousveillance hat using Device to Device (D2D) communications in LTE assisted networks. WSEAS Transactions on Communications, 14(2), 25-51.

[46] Baker, S. B., Xiang, W., & Atkinson, I. (2017). Internet of things for smart healthcare: Technologies, challenges, and opportunities. IEEE Access, 5, 26521-26544.

[47] Olatinwo, D. D., Abu-Mahfouz, A., & Hancke, G. (2019). A Survey on LPWAN Technologies in WBAN for Remote Health-Care Monitoring. Sensors, 19(23), 5268.

[48] Catherwood, P. A., Rafferty, J., McComb, S., & McLaughlin, J. (2018, July). LP-WAN wearable intelligent healthcare monitoring for heart failure prevention. In Proceedings of the 32nd International BCS Human Computer Interaction Conference 32 (pp. 1-4).

[49] Manatarinat, W., Poomrittigul, S., & Tantatsanawong, P. (2019, July). Narrowband-Internet of Things (NB-IoT) System for Elderly Healthcare Services. In 2019 5th International Conference on Engineering, Applied Sciences and Technology (ICEAST) (pp. 1-4). IEEE.

[50] Croock, M. S. (2014). LTE Based E-Health Monitoring System. IRAQI JOURNAL OF COMPUTERS, COMMUNICATION AND CONTROL & SYSTEMS ENGINEERING, 14(2), 37-45.

[51] Zhang, H., Li, J., Wen, B., Xun, Y., & Liu, J. (2018). Connecting intelligent things in smart hospitals using NB-IoT. IEEE Internet of Things Journal, 5(3), 1550-1560.

[52] Talcott, C. (2008). Cyber-physical systems and events. In Software-Intensive Systems and New Computing Paradigms (pp. 101-115). Springer, Berlin, Heidelberg.

[53] Yongfu, L., Dihua, S., Weining, L., & Xuebo, Z. (2012, July). A service-oriented architecture for the transportation cyber-physical systems. In Proceedings of the 31st Chinese Control Conference (pp. 7674-7678). IEEE.

[54] Santa, J., Sanchez-Iborra, R., Rodriguez-Rey, P., Bernal-Escobedo, L., & Skarmeta, A. F. (2019). LPWAN-based vehicular monitoring platform with a generic IP network interface. Sensors, 19(2), 264.

[55] Al-Shehri, S., Loskot, P., & Hirsch, M. J. (2020). Localization Enhanced Mobile Networks. In Mobile Computing. IntechOpen.

[56] Afaq, M., Iqbal, J., Ahmed, T., Islam, I. U., Khan, M., & Khan, M. S. (2020). Towards 5G network slicing for vehicular ad-hoc networks: An end-to-end approach. Computer Communications, 149, 252-258.

[57] Shen, X., Fantacci, R., & Chen, S. (2020). Internet of Vehicles. Proceedings of the IEEE, 108(2), 242-245.

[58] Hayes, M., & Omar, T. End to End VANET/IoT Communications A 5G Smart Cities Case Study Approach.

[59] Chen, S., Hu, J., Shi, Y., & Zhao, L. (2016). LTE-V: A TD-LTE-based V2X solution for future vehicular network. IEEE Internet of Things journal, 3(6), 997-1005.

[60] Petrov, V., Samuylov, A., Begishev, V., Moltchanov, D., Andreev, S., Samouylov, K., & Koucheryavy, Y. (2017). Vehicle-based relay assistance for opportunistic crowdsensing over narrowband IoT (NB-IoT). IEEE Internet of Things journal, 5(5), 3710-3723.

[61] RPMA. RPMA Technology for the Internet of Things. Ingenu, Tech. Rep. 2016.

[62] Online Availability: http://www.weightless.org Accessed on 10 February 2020.

[63] Abrardo, A., & Pozzebon, A. (2019). A multi-hop LoRa linear sensor network for the monitoring of underground environments: the case of the Medieval Aqueducts in Siena, Italy. Sensors, 19(2), 402.

[64] Addabbo, T., Fort, A., Mecocci, A., Mugnaini, M., Parrino, S., Pozzebon, A., & Vignoli, V. (2019, March). A LoRa-based IoT Sensor Node for Waste Management Based on a Customized Ultrasonic Transceiver. In 2019 IEEE Sensors Applications Symposium (SAS) (pp. 1-6). IEEE.

[65] https://www.3gpp.org/release-15 Accessed on 10 February 2020.

[66] https://www.3gpp.org/release-16 Accessed on 10 February 2020.

[67] https://fossa.systems Accessed on 10 February 2020.

[68] https://lacuna.space/company/ Accessed on 10 February 2020.

Enabling technologies for the Internet of Vehicles: standards, research and open challenges

Claudia Campolo, Antonella Molinaro

University Mediterranea of Reggio Calabria

Abstract: *The recent technological advancements in the field of sensing, automation, computing and communication technologies for vehicles are revolutionizing transportation systems on a global scale, while fostering large investments in the automotive market. Vehicles, as multi-faceted objects equipped with on board sensors like cameras, radars, positioning receivers, and with storage and processing capabilities, are becoming quite rightly elements of the Internet of Things (IoT). The Internet of Vehicles (IoV) ecosystem represents a prominent instantiation of the IoT, aimed at enabling smart, efficient and green traffic management, intelligent vehicle control, safe, comfortable and pleasant driving and traveling experience. The first step for the IoV is to make vehicles connected and able to interact with nearby/remote people and objects, and among each other, thanks to vehicle-to-everything (V2X) communication technologies. By leveraging efficient air interfaces, a wide range of allocated frequencies, advanced transceivers, multiple radio access technologies, as well as cutting-edge network softwarization principles, fifth generation (5G) systems intend to guarantee ultra-low latency, ultra-high reliability, and high-data rate V2X connectivity. Further promising paradigms and technologies in the IoT and future Internet research arena can contribute to address the needs of vehicular applications. This chapter will describe the IoV status quo, by analyzing the V2X application requirements and the main enabling communications technologies; research challenges and opportunities related to edge computing, virtualization, artificial intelligence, and other IoV enablers will also be part of this chapter.*

1 Introduction

Nowadays, modern vehicles carry on board high-speed processors, high-capacity memory storage, and a multitude of sensors, ranging from side collision radars to global positioning system (GPS) and cameras, which monitor the internal state of the vehicle and its close surroundings. Thanks to on board radio transceivers, vehicles can autonomously exchange their kinematics parameters, sensed data, and driving intentions with neighboring vehicles and coordinate their maneuvers in (semi-)autonomous driving settings, or they can send diagnostics and maintenance information to the cloud servers of manufacturers. Vehicles can as well provide navigation services and other traffic telematics applications, along with on-demand video streaming and online Internet access to passengers. In short, a vehicle, being equipped with vehicle-to-everything (V2X) connectivity, can exchange data with other communicating entities in its surrounding and becomes an essential element of the Internet of Vehicles (IoV) vision, one of the most prominent instantiation of the Internet of Things (IoT).

The ultra-low latency (below 1ms), high-throughput (higher than 1Gbps) and high reliability (close to 100%) requirements of emerging V2X applications, coupled with the high speed and the massive amount of vehicles generating data on the road, severely call into question the current capabilities of available communication technologies. Although more than 20 years passed since a dedicated spectrum was allocated to vehicular communications in 1999, the decision about the technology for V2X communications is still under debate. Connectivity is not the only concern: automotive applications also require access to computing and storage facilities to respectively process a large amount of data (e.g., vehicle diagnostics, traffic information, high-definition maps) and cache them, in most cases, as close as possible to where they are generated and likely consumed.

The fifth generation (5G) system appears as the most prominent solution to address the daunting challenges of the vehicular ecosystem. Its highly flexible and programmable end-to-end communication, networking, and computing infrastructure can provide increased performance in terms of throughput, latency, reliability, capacity, and mobility support, while meeting diversified requirements from multiple services. However, the complexity of the IoV landscape spans other domains and goes beyond what 5G technologies can provide. Indeed, other approaches emerging in the future Internet and IoT realms, e.g., virtualization, software-defined networking (SDN), artificial intelligence, information-centric networking (ICN), Social IoT (SIoT), may be required to facilitate service provisioning to and from connected and intelligent vehicles.

In such a context, this chapter aims to provide the following main contributions:

- to introduce vehicular applications by especially dissecting the demands of the most critical ones, i.e., those related to advanced driving (Section 2);
- to provide an overview of the main V2X communication technologies, with a special focus on IEEE 802.11 and cellular-based connectivity solutions (Section 3);
- to analyze cloud computing and its evolution towards edge computing and caching as key enablers of the IoV concept (Section 4);
- to debate the role of virtualization technologies, as initially conceived in IoT, to augment and abstract also vehicular devices (Section 5);
- to argue about the potential of the SDN philosophy in IoV to go well beyond traditional routing (Section 6);
- to analyze the potential of Artificial Intelligence (AI) techniques in IoV (Section 7);
- to unveil the benefits of adding the social dimension to connected vehicles (Section 8).

Such targets will be pursued by analysing the IoV *status quo* and identifying the main intriguing research directions, by scanning the findings of some of the works of the authors, whenever possible, and of the latest literature and standardization progress.

2 Vehicular applications

Vehicular applications cover a plethora of use cases that are typically classified according to their purpose and minimum demands [1]. The 5G Automotive Association (5GAA)[1], which brings together the major telecommunications operators, automotive industries, and chip manufacturers, grouped V2X use cases in four categories: *(i) Safety*, targeting the reduction of the frequency and severity of vehicle collisions through the exchange of warnings; *(ii) Convenience*, providing services to manage the health of vehicles, like diagnostics and software updates; *(iii) Vulnerable Road User (VRU)*, aiming at ensuring safe interactions between vehicles and other non-vehicle road users; *(iv) Advanced driving assistance*, sharing similar objectives with safety use cases, but aiming to support autonomous and semi-autonomous vehicle operation.

The last category exhibits the strictest delivery requirements. Within the Third Generation Partnership Project (3GPP) this category is further classified as follows [2]: *(i) Vehicles platooning*, enabling the dynamic formation of a group of vehicles travelling together with short inter-vehicle distances; *(ii) Advanced driving*, where vehicles coordinate their trajectories and maneuvers by sharing with other vehicles in proximity both data obtained from local sensors and driving intentions; *(iii) Extended sensors*, enabling the exchange of raw/processed sensor data or live videos among vehicles, Road Side Units (RSUs), and VRUs, *(iv) Remote driving*, allowing a remote driver or a cloud application to tele-operate a (private or public) vehicle, when passengers cannot drive themselves (e.g., impaired people) or when the vehicle is located in a dangerous or uncomfortable environment (e.g., earthquake-affected region, road construction work zone).

The requirements for such services get stricter as the degree of vehicle automation increases: e.g., latency requirements pass from 100 ms for information sharing in case of advanced driving, to 5 ms for remote driving. Remote driving also requires very high reliability in the order of 99.999 (five nines) percent. Besides connectivity, these V2X applications also need computing resources, e.g., for developing remote vehicle control systems, for the analysis and aggregation of sensor/video data retrieved from vehicles and other sources. This will be clarified in the following Sections.

3 V2X communication technologies

Connectivity is the first step to make a vehicle part of the IoV. Furthermore, V2X connectivity plays a pivotal role to minimize the environmental impact of transportation and improve traffic efficiency, support cooperative automated driving, enhance road safety and make the traveling experience more pleasant.

In the following, the main candidate technologies enabling V2X connectivity will be presented.

3.1 V2X connectivity options

The term V2X communication collectively gathers multiple types of interactions in the vehicular ecosystem. In 3GPP documents [3] the following modes are considered: *vehicle-to-vehicle (V2V)* communications directly established between vehicles in proximity of each

[1] http://5gaa.org/

other; *vehicle-to-infrastructure (V2I)* between vehicles and nearby roadside infrastructure, such as *RSUs* in traffic lights or eNodeBs; *vehicle-to-pedestrian (V2P)* between vehicles and VRUs, including e.g., pedestrians, motorcyclists, bikers, roller skaters, wheelchairs; *vehicle-to-network (V2N)* allowing vehicles to interact with remote communication entities (e.g., a cloud server) reachable through the cellular infrastructure.

Besides interacting with on-board sensors, with passengers and their smart devices, with other vehicles and RSUs on ground, vehicles can also exchange data with space and aerial platforms (e.g., satellites, drones). Vehicle-to-drone (V2D) communications are investigated in [4]. Drones can help vehicles acquire data from wider geographic areas [5]; they can carry mobile base stations to fill the gap of short-range connections, while complementing the coverage of terrestrial base stations [6].

In addition, IoV is expected to be integrated with other IoT systems, such as smart homes, smart cities, smart grids, hence new types of interactions will be supported, such as vehicle-to-grid (V2G), vehicle-to-home (V2H) [7], and so on.

3.2 IEEE 802.11

IEEE 802.11 initially elicited the interest of the industrial and academic community, due to operation simplicity and native support for V2V communications in a distributed manner. The IEEE 802.11p amendment, now part of the IEEE 802.11 standard [8], was conceived as an enhancement of the IEEE 802.11a, with physical (PHY) and medium access control (MAC) layers' settings and procedures properly adjusted to support outdoor communications under high speed mobility. It was complemented by the IEEE 1609 family of standards [9], collectively referred to as Wireless Access in Vehicular Environment (WAVE), including the architecture, management structure, security, and physical access for wireless vehicular networks.

Worldwide field trials have demonstrated the .11p feasibility of supporting basic safety applications (e.g., emergency brake light, stationary vehicle warning) [10] in low congested scenarios. Nevertheless, this technology suffers from poor performance under high traffic density, so it cannot fulfill the strict requirements for very low latency and high-bandwidth of advanced V2X applications. Such limitations are mainly due to a basic PHY layer and to the lack of a protection mechanism from interference and collisions, especially for broadcast communications, at the MAC layer, as ruled by the distributed carrier sensing multiple access with collision avoidance (CSMA/CA) protocol [10].

Many attempts to improve the IEEE 802.11 performance can be found in the literature. Among them, in [11] the authors propose to leverage full duplex (FD) radios on board to enable collision detection of broadcast packets while transmitting. The reliability of broadcast communications can be improved by recovering from packet losses via informed, other than blind, retransmissions.

Recently, a new IEEE study group, named IEEE 802.11 Next Generation V2X (NGV), now preparing the IEEE 802.11bd amendment, has been created to investigate evolved PHY technologies, such as multiple input multiple output (MIMO) and advanced coding techniques, in order to enhance the .11p coverage and throughput [12].

3.3 Cellular-V2X

The previous role of IEEE 802.11 as the *de-facto* standard for V2X communications has been undermined by the cellular technology, powerfully entering the automotive domain. As argued in the seminal work of the authors in [13], cellular-based V2X communications can adequately cope with the demands of vehicular applications, thanks to the ubiquitous deployed infrastructure, its centralized organization and mature industrial foundation, as well as to the support for V2V communications, even out-of-the coverage of a base station, and for multicast/broadcast message dissemination.

The first stage of 3GPP specifications of what is referred to as cellular-V2X (C-V2X) was completed in June 2017 for Release 14, with further refinements included in Release 15 in 2019. C-V2X is a unified technology that allows vehicles to communicate with nearby and remote entities, and guarantees full coverage and service continuity. Its specifications span the radio access network (RAN) as well as the core network (CN) segments.

RAN. Modifications over the RAN encompass the conventional long term evolution (LTE)-Uu and the PC5 radio interfaces. The former is leveraged for uplink/downlink V2N communications and operates in the licensed spectrum. The latter is meant for direct localized interactions (e.g., V2V, V2P, V2I) in the unlicensed 5.9 GHz band. Most of the 3GPP efforts focused on the PC5 customization to match the vehicular applications' requirements. In particular, two communications modes have been specified, namely *Mode 3* or scheduled, and *Mode 4* or autonomous. In *Mode 3*, operating only in-coverage of an eNodeB, the radio resource allocation is supervised by the network, without specifying a given algorithm, which is left open to the operators. In *Mode 4*, a pool of pre-configured resources can be accessed by vehicles in an autonomous manner, also in out-of coverage of an eNodeB conditions (e.g., in urban canyons, tunnels). The allocation scheme relies on a sensing mechanism aimed at identifying the less interfered resources as potential candidates for transmission. This is coupled with a semi-persistent scheduling (SPS) scheme, which reserves the same resources for multiple transmissions, and well suits the periodic nature of broadcast messages exchanged among vehicles in the one-hop neighborhood, i.e., Cooperative Awareness Messages (CAMs) [14].

CN. Architectural enhancements have been specified to manage V2X communications, with the introduction of two new entities. The *V2X Control Function* is responsible for the V2X policy provisioning and parameters configuration over the PC5 (in- and out-of-coverage) and the LTE-Uu interfaces. The *V2X Application server (AS)* has a wide range of functionalities, including the reception of uplink unicast data from vehicular User Equipments (VUEs), VRUs, RSUs, and the data delivery to VUEs in a target area by unicast and/or multicast transmissions.

5G and beyond evolution. C-V2X kept evolving in the last couple of years in order to be aligned with the 5G new radio (NR) specifications, with enhancements for Release 16 to be issued by early 2020. Envisioned radio interface modifications have the main objectives *(i)* to introduce new/flexible waveforms and numerologies to support the most challenging V2X applications' demands, and *(ii)* to improve Modes 3 and 4 in order to support peculiar V2X traffic patterns, such as aperiodic traffic [15] and unicast/multicast/groupcast (besides broadcast) communications [12]. Non broadcast communication primitives are particularly indicated for platooning applications, where communications must be confined with the platoon. Improvements of the autonomous mode are also suggested for such application, so that resources can be allocated to vehicles with the help of other vehicles, e.g., a vehicle in the middle of the platoon that can better measure the channel conditions, or the platoon leader, as preliminarily argued in the work in [16] anticipating Release 16.

The effectiveness and reliability of the 5G NR features at the PHY layer have been investigated in [17], [18], also with comparison with IEEE 802.11bd. Early but encouraging results about the impact of a flexible numerology in the autonomous resource allocation mode can be also found in the work of the authors in [19].

Other technologies and concepts lie under the 5G-and-beyond umbrella, although not yet prioritized by 3GPP, but which will be likely included in Release 17. Among them there is the usage of frequencies above 6 GHz (millimeter waves). Such frequencies promise larger bandwidth and higher throughput, which can be particularly appealing for: *(i)* V2V communications between very close vehicles, e.g., to support extended sensor applications in high-density platoons, and *(ii)* V2I communications for rapid bulk data transfer to/from a RSU (e.g., for real-time high-definition map download, object detection and recognition). The harsh propagation environment may, however, hamper such benefits. Challenges arise, for example, due to the overhead for the beam training under high mobility and the blockage effect by e.g., pedestrian bodies [20], [21].

The FD technology is another prominent beyond-5G solution. In [22] the authors have demonstrated that FD on board capabilities can improve the performance of the autonomous mode, by letting vehicles adapt the probability of keeping a given resource according to the level of interference detected while transmitting.

Other enhancements are planned by 3GPP at the architectural level. As the 5G CN design in Release 16 is progressing towards a network of functions, it is under discussion whether a C-V2X architecture, simpler than the one designed in Release 14, can be developed by using some of the available 5G Network Functions (NFs).

3.4 Other connectivity options

Although cellular connectivity is considered as the main option to enable V2X interactions, for the sake of completeness it is worth mentioning that other technologies gaining momentum in the IoT landscape could be exploited in the vehicular domain, for at least a subset of V2X services. Indeed, we expect IoV to be supported by a mash-up of different connectivity solutions.

Among short-range technologies, Bluetooth has been considered in [1]. For instance, the work in [23] suggests the usage of Bluetooth to detect VRUs. Low-powered, low-cost transmitters Bluetooth Low Energy (LE) devices can be easily installed into a bicycle and notify nearby Bluetooth LE devices of their presence. However, due to the very limited communication range (around 50 meters), Bluetooth would hardly match the needs of generalized V2X communications.

Also Visible Light Communications (VLC) gained attention as an alternative to the existing radio frequency (RF) based vehicular communications. Such an attention is mainly fueled by the need to reduce the load on the crowded wireless channel [24], through a relatively simple and energy-efficient system. The same light is used for both illumination and as a data carrier. Although VLC can provide high throughput (in the order of 10 Gbps), its operation is limited to line of sight communications. This prevents eavesdropping.

On the other hand, Low Power-Wide Area Network (LP-WAN) technologies, in particular Long Range Wide Area Network (LoRaWAN), have attracted the interest of automotive stakeholders due to the long coverage range and high scalability. They could offer a valuable option for small message delivery with low throughput, such as event notifications or vehicle diagnostics and monitoring messages [25].

4 Vehicular cloud computing

With the proliferation of smart and autonomous vehicles equipped with a massive number of sensors and human computer interaction devices, several vehicular applications can rely on the continuous retrieval and manipulation of big amounts of data, collected from vehicles themselves, road-side sensors, infrastructure nodes and other sources. An autonomous vehicle is envisaged to generate 1 GB data for every second and, on average, 30 TB data per day. Such data can be exploited by different stakeholders (i.e., the driver, the municipality, the road transportation authority, etc.) and used for short-term and long-term, real-time and non real-time decision taking. For instance, collected data can help to predict congested urban areas and to detect anomalous situations in real-time.

The conventional approach is to transfer data, retrieved by on board sensors, to the cloud through V2X connectivity. Storing and processing data in the cloud is adequate for some V2X use cases, but this may be both unreliable and slow for other more demanding use cases (e.g., augmenting the driver's visual perception), due to the long and fluctuating delay to the cloud. Moreover, the massive amount of data to be processed may challenge the virtually unlimited capabilities of cloud platforms. In this view, the multi-access edge computing (MEC) paradigm, by offering cloud-like resources (i.e., storage, processing) at the network edge [26], can offload the core network and enable V2X applications to benefit from reduced access latency. Moreover, edge-based vehicular applications can benefit from additional (context) information that is not directly available to the participating vehicles, e.g., via data fusion from multiple available sources.

The added value of MEC in V2X environments has been recognized by the 5GAA [27] and by the European Telecommunications Standards Institute (ETSI), as well as investigated in the literature [28]. In particular, the ETSI document in [29] analyses relevant V2X use cases and identifies the new requirements, features and functions, with special attention on the support of vehicle mobility and its impact on service provisioning. Indeed, compared to other vertical markets, the automotive domain adds a layer of complexity to the computation offloading at the edge. It may entail the migration of application services from one MEC server to another, in response to vehicles' mobility, so to ensure the shortest latency and the best edge service provisioning. Moving services closer to the end-users may cause, as a side effect, performance degradation and even interruption of ongoing services during the migration procedures [30].

Not only RSUs and eNodeBs can act as edge nodes, but the vehicles themselves, either in moving or parking states, can establish a vehicular cloudlet and share their computing resources with the requesting neighboring vehicles [31]. In so doing, the cloud-to-things continuum vision can be enabled also in the IoV domain, where edge and deep edge facilities will complement the remote cloud capabilities in a synergic manner.

Caching contents at the edge is a further facilitator of IoV. Indeed, contents requested and generated in IoV are typically characterized by a temporal and spatial scope [32]. Moreover, some contents, such as map download and over-the-air software updates, can be popular, hence requested repeatedly by multiple users. Such peculiarities motivated several studies, which were scanned in the pioneering work of the authors in [32] and, more recently, in [33], which propose ICN for content discovery and delivery in the vehicular and edge domains, in analogy to what has been also proposed for the IoT [34], [35]. V2X communications privilege the content (e.g., accident notification; stationary vehicle warning; road works warning) rather than the identity of the communication endpoints. The ICN native in-network caching capability, its support for multicast and multipath forwarding, coupled with a provider-independent naming scheme, well suit IoV where vehicles are interested in the content itself and not in its provenance.

5 The role of virtualization in IoV

Vehicles, acting as computers on wheels and as moving sensors, are allowed to be part of a complex ecosystem as a special kind of connected things. Vehicular devices can be augmented with a degree of intelligence and cognitiveness, which is enabled by a *virtualization layer*, typical of next-generation IoT architectures. This layer is able to abstract vehicular devices and their capabilities.

Virtualization is a well known concept in the IoT domain [36]. A *virtual object* consists of the semantic uniform description of the related physical object, of its resources (e.g., memory, storage, processing) and its capabilities (e.g., sensing, actuation, computing), which are abstracted into a set of attributes. This facilitates the creation and composition of services at the application layer.

The virtualization of vehicular devices is still in its infancy, but the success of the virtual object concept, today available in commercial IoT-cloud platforms (e.g., Amazon Web Services IoT[2]) and further revamped by the *digital twin* concept [37], can greatly contribute to the widespread and quick deployment of virtualization in the IoV domain.

The work in [38] leverages the Docker platform[3] as a container virtualization technology. It aims to virtualize vehicular on board units (OBUs), by assigning to different on-board devices a container, which is well isolated from the others and is in charge of handling the functionalities of the corresponding physical devices (i.e., camera, infotainment system). The solution proposed in [39] targets similar objectives, with the main difference that the virtual OBU (V-OBU) is hosted in MEC facilities, with the aim of offloading processing from the vehicle and serving data-access requests. This deals with potential disconnection periods of vehicles, saves radio resources when accessing the physical OBU, and improves data processing performance. The idea of a virtual vehicle is further elaborated in the work in [30], where, similarly to [38], Docker is chosen as a virtualization platform, and the issue of mobility of the virtualized vehicular service is investigated.

Besides vehicles, also drones and road-side infrastructure nodes, e.g., a traffic light, can be virtualized.

The simple object-based resource model provided by the Open Mobile Alliance (OMA) Lightweight Machine-to-Machine (LwM2M) [40] can be used for the semantic description of the vehicle as an object. This is the approach followed in [41] to model the smartphone when mobility-related data are exploited in an interoperable manner by multiple stakeholders. Different application-layer messaging protocols investigated in the IoT

[2] https://aws.amazon.com/iot-core/features/?nc1=h_ls

[3] https://www.docker.com/

realm, e.g., Message Queue Telemetry Transport (MQTT), Constrained Application Protocol (CoAP), Hyper-text Transfer Protocol (HTTP), can enable the interaction between the physical object and its virtual counterpart. The effectiveness of such protocols in the vehicular context was preliminarily investigated in [42]. There, CoAP is shown to outperform the other two solutions by ensuring shorter latency and higher throughput, under the considered settings.

6 Software-defined IoV

Managing communication, networking and computing resources in IoV is a tough task, which is becoming more challenging with the increased number of radio access technologies (RATs) today available in connected cars, due to the variety of heterogeneous applications provided on the road and to the dynamicity of the environment.

The SDN paradigm breaks the vertical integration of traditional networks by separating the control plane (CP) from the user plane (UP). In doing so, network devices become simple forwarding devices, and the control logic, which determines how data traffic flows should be treated in the network, is implemented in a logically centralized entity, the *controller*, that has a global view of the network.

In the IoV domain, SDN allows the remote configuration of network elements and on demand resource reservation for safety-critical V2X applications. It can steer traffic through the set-up of automatically configured paths for load balancing or traffic prioritization purpose or to react to network failures and changing conditions (e.g., due to intermittent roadside connectivity or to mobility) [43]. Taking forwarding decisions by leveraging global information collected by the SDN controller from multiple sources can significantly improve multi-hop V2X data dissemination [44].

SDN can deliver multiple applications in an isolated manner, while guaranteeing their performance requirements, e.g., by leveraging different RATs/channels, configuring disjoint routing paths, filtering traffic classes at some intermediate nodes. This capability particularly fits the *network slicing* concept, proposed to fulfill conflicting requirements of 5G use cases in a flexible and holistic manner, as analysed in [45]. It configures NFs, network applications, specific radio access settings and underlying computing resources to specific service needs on a common *programmable end-to-end* network infrastructure, where CP and UP NFs are logically isolated. In our early work in [46], we advocated network slicing as a prominent solution to enable the simultaneous support of heterogeneous V2X applications.

The open programmability and the logically centralized knowledge and control features of the SDN paradigm offer an attractive means to orchestrate other functions besides routing, which are developed as network applications [47]. Some of these functions include e.g., setting the transmission power levels to control vehicle interference, enforcing security policies, supporting network selection, enabling caching decisions and placement of computation tasks along the cloud-to-things continuum [48].

7 Artificial intelligence in IoV

IoV will need not only communication, processing and storage, but also learning capabilities. Indeed, AI and, in particular, machine learning (ML) methods can assist key tasks of intelligent vehicles by extracting knowledge from information-rich data [49]. These methods can be leveraged to enhance autonomous driving operations, for instance by producing accurate models of the surrounding environment from the cooperative sensing data, and deriving effective manoeuvring strategies accordingly [50]. By means of ML, vehicles can identify crucial data to be exchanged for cooperative driving purposes, with efficient network resources utilization. Making ML-assisted decisions may also improve radio resource management algorithms [51].

Applying existing ML methods to IoV raises some challenges, which are mainly related to the distributed nature of data produced by multiple sources and to the hard to be predicted vehicular dynamics. Theses issues pave the way to distributed learning methods. The decision about where to run ML algorithms (i.e., on board the vehicle, at the edge, or in the cloud) is another open issue to be addressed [49]. The *edge AI* concept is mentioned among the top 10 strategic technology trends in 2019 by Gartner[4].

ML techniques support the orchestration of edge resources concerning, among others, the decision about the proper placement of computing functionalities along the cloud-to-vehicle continuum [52]. Indeed, optimization problems often assume that few key parameters are known, but actually some of them are difficult to obtain and are subject to dynamic variations, e.g., due to fluctuations of the wireless channel and vehicle mobility. In [53] a relocation policy for applications in the MEC environment is proposed which is based on deep reinforcement learning.

8 Social IoV

Next-generation IoT architectures are expected to realize the radical IoT paradigm shift from connected things to connected and intelligent things. In such an evolution path, the SIoT paradigm [54] is expected to play a pivotal role, since it aims to provide devices with intelligence and awareness to socialize with each other, as humans do. Social relationships are created without the human intervention, based on shared context and mutual interests (e.g., devices are located in the same place, they are owned by the same user, or they frequently meet each other). Establishing social relationships among objects has the potential to facilitate information/service discovery when the IoT is made of a huge number of heterogeneous nodes.

The extension of this paradigm to the IoV domain has been preliminarily investigated in [55] and further analyzed in [56] and references therein. The autonomous establishment of inter-object social links particularly fits the vehicular environment, in which machine-to-machine (M2M) interactions are dominant. V2X entities themselves can proactively request and consume data, seamlessly to the drivers, by exploiting the created social relationship, such as vehicles belonging to the same automaker, or frequently traveling along the same road segment. The ultimate goal can be e.g., improving the road safety or the traveling experience. Minimizing the interactions between the driver and the on-

[4] https://www.gartner.com/en/documents/3956137/hype-cycle-for-edge-computing-2019

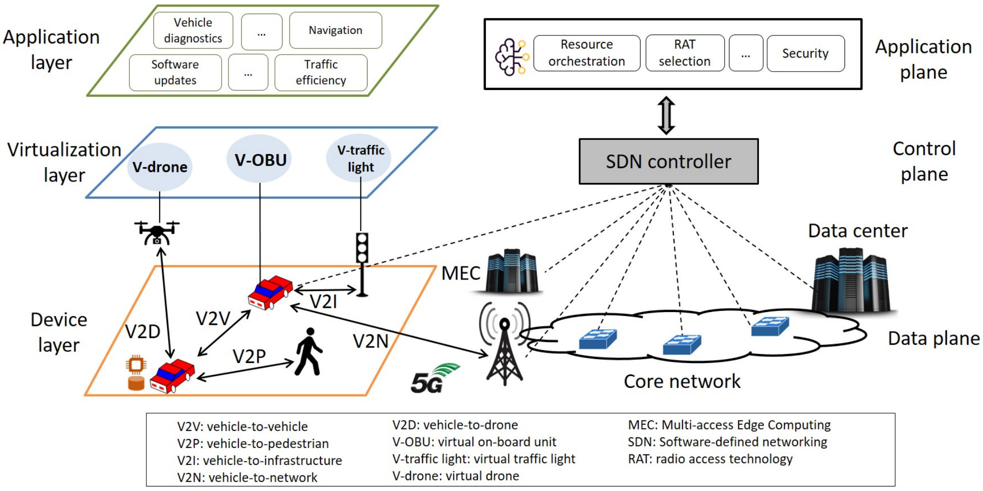


Figure 1: Next-generation IoV reference architecture.

board devices has also the non-negligible side benefit of reducing the driver distractions and the risk of fatalities on the road [43].

9 Conclusions

In this chapter, we presented a concise but exhaustive analysis of the main IoV enablers. The review encompasses both the most consolidated technologies, i.e., the two mainstream V2X connectivity solutions, and groundbreaking solutions and paradigms, still in their infancy in the IoV domain, such as artificial intelligence, virtualization and network softwarization, cloud computing evolution. A reference next-generation IoV architecture, resulting from the mash-up of the aforementioned technologies, is graphically sketched in Fig. 1. Such technologies hold several promises, but several open issues still lie ahead for practically deploying each of them.

It is convincement of the authors that the full potential of IoV can be only disclosed by fostering synergies among heterogeneous fields with the efforts of the entire research community.

References

[1] Z. MacHardy, A. Khan, K. Obana, and S. Iwashina, “V2X access technologies: Regulation, research, and remaining challenges,” *IEEE Comm. Surveys & Tutorials*, vol. 20, no. 3, pp. 1858–1877, 2018.

[2] “3GPP TR 22.186 v16.2.0, Technical specification group services and system aspects. enhancement of 3GPP support for V2X scenarios. Release 15,” June 2019.

[3] "3GPP TR 22.185 v15.0.0, Service requirements for V2X services; Release 15," June 2018.

[4] W. Shi, H. Zhou, J. Li, W. Xu, N. Zhang, and X. Shen, "Drone assisted vehicular networks: Architecture, challenges and opportunities," *IEEE Network*, vol. 32, no. 3, pp. 130–137, 2018.

[5] W. Xu, H. Zhou, N. Cheng, F. Lyu, W. Shi, J. Chen, and X. Shen, "Internet of vehicles in big data era," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 1, pp. 19–35, 2017.

[6] S. Mignardi, C. Buratti, A. Bazzi, and R. Verdone, "Trajectories and resource management of flying base stations for C-V2X," *Sensors*, vol. 19, no. 4, p. 811, 2019.

[7] L.-M. Ang, K. P. Seng, G. K. Ijemaru, and A. M. Zungeru, "Deployment of IoV for smart cities: applications, architecture, and challenges," *IEEE Access*, vol. 7, pp. 6473–6492, 2018.

[8] "IEEE Std. 802.11-2012: "IEEE Standard for Information technology - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," 2012.

[9] C. Campolo and A. Molinaro, "Multichannel communications in vehicular ad hoc networks: a survey," *IEEE Communications Magazine*, vol. 51, no. 5, pp. 158–169, 2013.

[10] C. Campolo, A. Molinaro, and R. Scopigno, "From today's VANETs to tomorrow's planning and the bets for the day after," *Vehicular Communications*, vol. 2, no. 3, pp. 158–171, 2015.

[11] C. Campolo, A. Molinaro, A. O. Berthet, and A. Vinel, "Full-duplex radios for vehicular communications," *IEEE Communications Magazine*, vol. 55, no. 6, pp. 182–189, 2017.

[12] G. Naik, B. Choudhury, and J.-M. Park, "IEEE 802.11 bd & 5G NR V2X: Evolution of radio access technologies for v2x communications," *IEEE Access*, 2019.

[13] G. Araniti, C. Campolo, M. Condoluci, A. Iera, and A. Molinaro, "LTE for vehicular networking: a survey," *IEEE communications magazine*, vol. 51, no. 5, pp. 148–157, 2013.

[14] ETSI EN 102 637-2 v1.3.1, "ITS; Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service," 2014.

[15] C. Campolo, A. Molinaro, A. O. Berthet, and A. Vinel, "On latency and reliability of road hazard warnings over the Cellular V2X sidelink interface," *IEEE Communications Letters*, 2019.

[16] C. Campolo, A. Molinaro, G. Araniti, and A. O. Berthet, "Better platooning control toward autonomous driving: An LTE device-to-device communications strategy that meets ultralow latency requirements," *IEEE Vehicular Technology Magazine*, vol. 12, no. 1, pp. 30–38, 2017.

[17] W. Anwar, A. Traßl, N. Franchi, and G. Fettweis, "On the reliability of NR-V2X and IEEE 802.11 bd," in *IEEE PIMRC 2019*.

[18] W. Anwar, N. Franchi, and G. Fettweis, "Performance evaluation of next generation V2X communication technologies: 5G NR V2V Vs IEEE 802.11 bd," in *IEEE VTC-Fall 2019*.

[19] C. Campolo, A. Molinaro, F. Romeo, A. Bazzi, and A. O. Berthet, "5G NR V2X: On the impact of a flexible numerology on the autonomous sidelink mode," in *IEEE 5G World Forum*, 2019.

[20] J. Choi *et al.*, "Millimeter-wave vehicular communication to support massive automotive sensing," *IEEE Comm. Mag.*, vol. 54, no. 12, pp. 160–167, 2016.

[21] M. Giordani, A. Zanella, and M. Zorzi, "Millimeter wave communication in vehicular networks: Challenges and opportunities," in *2017 6th International Conference on Modern Circuits and Systems Technologies (MOCAST)*. IEEE, 2017, pp. 1–6.

[22] C. Campolo, A. Molinaro, F. Romeo, A. Bazzi, and A. O. Berthet, "Full duplex-aided sensing and scheduling in Cellular-V2X Mode 4," in *Proc. of the 1st ACM MobiHoc Workshop on Technologies, mOdels, and Protocols for Cooperative Connected Cars*, 2019, pp. 19–24.

[23] J. J. Anaya, E. Talavera, D. Giménez, N. Gómez, J. Felipe, and J. E. Naranjo, "Vulnerable road users detection using V2X communications," in *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. IEEE, 2015, pp. 107–112.

[24] A.-M. Căilean and M. Dimian, "Current challenges for visible light communications usage in vehicle applications: A survey," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2681–2703, 2017.

[25] R. Sanchez-Iborra, J. Sánchez-Gómez, J. Santa, P. J. Fernández, and A. F. Skarmeta, "IPv6 communications over LoRa for future IoV services," in *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*. IEEE, 2018, pp. 92–97.

[26] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: A survey of the emerging 5g network edge cloud architecture and orchestration," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1657–1681, 2017.

[27] "5G Automotive Association (5GAA) - Toward fully connected vehicles: Edge computing for advanced automotive communications," December 2017.

[28] F. Giust, V. Sciancalepore, D. Sabella, M. C. Filippou, S. Mangiante, W. Featherstone, and D. Munaretto, "Multi-access Edge Computing: The driver behind the wheel of 5G-connected cars," *IEEE Communications Standards Magazine*, vol. 2, no. 3, pp. 66–73, 2018.

[29] "ETSI - Multi-access Edge Computing (MEC); Study on MEC Support for V2X Use Cases," September 2018.

[30] C. Campolo, A. Iera, A. Molinaro, and G. Ruggeri, "MEC support for 5G-V2X use cases through docker containers," in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2019, pp. 1–6.

[31] R. A. Dziyauddin, D. Niyato, N. C. Luong, M. A. M. Izhar, M. Hadhari, and S. Daud, "Computation offloading and content caching delivery in vehicular edge computing: A survey," *arXiv preprint arXiv:1912.07803*, 2019.

[32] M. Amadeo, C. Campolo, and A. Molinaro, "Information-centric networking for connected vehicles: a survey and future perspectives," *IEEE Communications Magazine*, vol. 54, no. 2, pp. 98–104, 2016.

[33] H. Khelifi, S. Luo, B. Nour, H. Moungla, Y. Faheem, R. Hussain, and A. Ksentini, "Named data networking in vehicular ad hoc networks: State-of-the-art and challenges," *IEEE Communications Surveys & Tutorials*, 2019.

[34] M. Amadeo, G. Ruggeri, C. Campolo, and A. Molinaro, "IoT services allocation at the edge via named data networking: From optimal bounds to practical design," *IEEE Transactions on Network and Service Management*, vol. 16, no. 2, pp. 661–674, 2019.

[35] M. Amadeo, C. Campolo, G. Ruggeri, A. Molinaro, and A. Iera, "SDN-managed provisioning of named computing services in edge infrastructures," *IEEE Transactions on Network and Service Management*, vol. 16, no. 4, pp. 1464–1478, 2019.

[36] M. Nitti, V. Pilloni, G. Colistra, and L. Atzori, "The virtual object as a major element of the Internet of Things: a survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1228–1240, 2015.

[37] A. El Saddik, "Digital twins: The convergence of multimedia technologies," *IEEE MultiMedia*, vol. 25, no. 2, pp. 87–92, 2018.

[38] R. Morabito, R. Petrolo, V. Loscri, N. Mitton, G. Ruggeri, and A. Molinaro, "Lightweight virtualization as enabling technology for future smart cars," in *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. IEEE, 2017, pp. 1238–1245.

[39] J. Santa, P. J. Fernández, J. Ortiz, R. Sanchez-Iborra, and A. F. Skarmeta, "SURROGATES: Virtual OBUs to foster 5G vehicular services," *Electronics*, vol. 8, no. 2, p. 117, 2019.

[40] "Open Mobile Alliance, Lightweight Machine to Machine Technical Specification Core; v1_1-20180612-c," 2018.

[41] C. Campolo, D. Cuzzocrea, G. Genovese, A. Iera, and A. Molinaro, "An OMA lightweight M2M-compliant MEC framework to track multi-modal commuters for MaaS applications," in *2019 IEEE/ACM 23rd International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*. IEEE, 2019, pp. 1–8.

[42] Z. Laaroussi, R. Morabito, and T. Taleb, "Service provisioning in vehicular networks through edge and cloud: an empirical analysis," in *2018 IEEE Conference on Standards for Communications and Networking (CSCN)*. IEEE, 2018, pp. 1–6.

[43] C. Campolo, A. Molinaro, and A. Iera, “A reference framework for social-enhanced vehicle-to-everything communications in 5G scenarios,” *Computer Networks*, vol. 143, pp. 140–152, 2018.

[44] S. Din, A. Paul, and A. Rehman, “5G-enabled hierarchical architecture for software-defined intelligent transportation system,” *Computer Networks*, vol. 150, pp. 81–89, 2019.

[45] S. Zhang, “An overview of network slicing for 5G,” *IEEE Wireless Communications*, vol. 26, no. 3, pp. 111–117, 2019.

[46] C. Campolo, A. Molinaro, A. Iera, and F. Menichella, “5G network slicing for vehicle-to-everything services,” *IEEE Wireless Communications*, vol. 24, no. 6, pp. 38–45, 2017.

[47] K. Z. Ghafoor, L. Kong, D. B. Rawat, E. Hosseini, and A. S. Sadiq, “Quality of service aware routing protocol in software-defined internet of vehicles,” *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2817–2828, 2018.

[48] R. D. R. Fontes, C. Campolo, C. E. Rothenberg, and A. Molinaro, “From theory to experimental evaluation: Resource management in software-defined vehicular networks,” *IEEE Access*, vol. 5, pp. 3069–3076, 2017.

[49] J. Zhang and K. B. Letaief, “Mobile edge intelligence and computing for the internet of vehicles,” *Proceedings of the IEEE*, 2019.

[50] A. E. Sallab, M. Abdou, E. Perot, and S. Yogamani, “Deep reinforcement learning framework for autonomous driving,” *Electronic Imaging*, vol. 2017, no. 19, pp. 70–76, 2017.

[51] H. Ye *et al.*, “Machine learning for vehicular networks: Recent advances and application examples,” *IEEE Veh. Tech. Mag.*, vol. 13, no. 2, pp. 94–101, 2018.

[52] Z. Ning, P. Dong, X. Wang, J. J. Rodrigues, and F. Xia, “Deep reinforcement learning for vehicular edge computing: An intelligent offloading system,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 6, pp. 1–24, 2019.

[53] F. De Vita, G. Nardini, A. Virdis, D. Bruneo, A. Puliafito, and G. Stea, “Using deep reinforcement learning for application relocation in multi-access edge computing,” *IEEE Communications Standards Magazine*, vol. 3, no. 3, pp. 71–78, 2019.

[54] L. Atzori, A. Iera, and G. Morabito, “SIoT: Giving a social structure to the internet of things,” *IEEE communications letters*, vol. 15, no. 11, pp. 1193–1195, 2011.

[55] M. Nitti, R. Girau, A. Floris, and L. Atzori, “On adding the social dimension to the internet of vehicles: Friendship and middleware,” in *2014 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*. IEEE, 2014, pp. 134–138.

[56] K. M. Alam, M. Saini, and A. El Saddik, “Toward Social Internet of Vehicles: Concept, architecture, and applications,” *IEEE Access*, vol. 3, pp. 343–357, 2015.

A Machine Learning Based Non-Orthogonal Multiple Access Scheme for IoT Communications

authors Romano Fantacci, and Benedetta Picano

Università degli Studi di Firenze

Abstract: *The ever increasing diffusion of the IoT devices, able to interact and exchange data with the surrounding environment, is exponentially growing, giving rise to ubiquitous and intelligent ecosystems capable of data gathering and usually requiring low latency processing of time. As a consequence, efficient multiple access schemes have to be identified to handle a usual massive access of IoT devices to a local computation node according to the emerging edge computing paradigm to lower latency and network congestion. This chapter deals with the application of a machine learning technique, specifically an echo state (ESN) machine learning framework is considered. A key challenge in our problem formulation is that we assume that our ESN has not any a priori information on the number and behavior of IoT devices, i.e., on when to transmit and when not to. The goal of the considered ESN is to perform an effective channel access strategy for a sigle carrier two power levels non-orthogonal multiple access (NOMA) scheme with the aim at maximizing the sum throughput and minimizing the mean packet access delay among all the IoT devices. Performance comparisons with a basic, i.e., without resorting to the use of a machine learning approach, NOMA scheme are presented in order to highlight the advantages of the proposed solution. In addition to this, simulation results are also provided to validate the obtained analytical predictions.*

1 Introduction

The unexpected growth of the internet of things (IoT) applications and the forthcoming fifth generation (5G) has opened the door to an unprecedented connectivity demand from smart devices [1, 2]. Recently, with the advent of the promising edge computing paradigm, computation and data storage is moved close to the end-devices, limiting latency and network congestion. In such a landscape, IoTs channel access schemes represent a crucial issue in the network resource management. In this regards, the improvement of the random access techniques have appeared as an attractive solution to make possible IoT devices communications with a reduced signaling overhead and high spectral efficiency. Towards this end, in 5G and beyond 5G cellular communication systems, the use of non-orthogonal multiple access (NOMA) schemes has emerged as a promising technique to enhance throughput performance and improve the spectral efficiency. The basic principle of NOMA is to allow a simultaneous network access to multiple IoT devices over same radio resources with a low inter-IoT device interference.

In particular, NOMA superposes the different IoT devices signals in the power domain and, on the basis of the use of the successive interference cancellation (SIC) method at

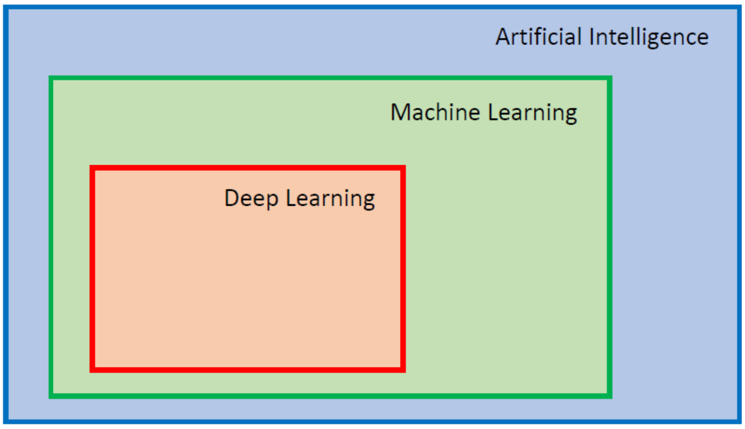


Figure 1: Artificial Intelligence classification

the receiver side, the exploitation of the same spectrum for all the users is allowed [3]. Hence, the clever combination of the simplicity of S-ALOHA technique with the high throughput performance achieved with the NOMA approach due to its capability to resolve access collisions by the use of the SIC scheme makes the S-Aloha NOMA (SAN) scheme an efficient solution to enable communications of low complexity IoT devices. Moreover, a transmission scheme able to dynamically adapt the access policy on the basis of the surrounding network conditions is suitable in the networks due the natural time evolution of the network environment. Therefore, as direct consequence of the ever increasing dynamic behavior of the modern wireless networks, the necessity for flexible and adaptive frameworks is rapidly becoming essential. During years, many approaches and studies have been proposed to improve the management of the random channel contention policies. In this regards, the a priori knowledge about the number of devices is a critical aspect of the majority of the effective solutions presented in literature. In such a context, the machine learning (ML) has recently received attention as powerful and promising tool to provide effective solutions for a wide range of wireless networks problems, among which the access ones. In this regards, it is important to note that there exist several ML methods capable of adapting to the network behavior, without the necessity of the knowledge about the number of devices contending on the shared channel. Such condition justifies the investigation of some ML techniques in reference to the random access problem, since they may represent promising approaches. Generally speaking, the ML includes a wide class of different methods and approaches for different problems, such as prediction, classification and clustering. Moreover, the ML is a specialization of the more wide artificial intelligence (AI) field, as represented in Figure 1. Although the AI research branch focuses on giving a sort of cognitive abilities to the machines, the ML represents the area of the AI which aims at analyzing and understanding patterns beyond data. Roughly speaking, the ML, in its turn, can be differentiated among the following main paradigms [4, 5, 6, 7, 8, 9]

- *Supervised learning*, in which the algorithms are used to map inputs in the outputs, on the basis of some input-output pairs example, properly labeled. The goal of this approach is the accurate mapping approximation, aiming at predicting the outputs on the given new inputs;

- *Unsupervised learning*, which is a self-organized approach, typically used to recognize patterns in data, without any prior labeling, classification, or clustering mechanism;

- *Reinforcement learning*, which focuses on decision making policies, taken in order to maximize a reward function contextualized to an environment, typically represented by a Markov decision process.

Within the unsupervised learning, artificial neural networks (ANNs) have gained attention as solution to better understand the hidden features of data and mapping functions, allowing good approximations and accurate predictions. During years, many different ANNs types have been proposed and studied, diverse in weakness and strengths. Among these, the recurrent neural networks (RNNs) are the the closest model to the biological brains functioning [10]. Despite the effectiveness of the RNNs [11], one weak aspect is constituted by their complexity in terms of training and structure, which make their application critical in several contexts, especially in relation to the limited computation capabilities and power constraints, typical of the IoT devices. This chapter aims at exhibiting the application of a particular type of RNNs, i.e., the echo state networks (ESNs), highly applicable to several problem scenarios, due to their intrinsic simplicity in training and structure [11], to the random access problem within a SAN system. Therefore, the contribution of this chapter can be summarized as follows

- Application of a ESN to enhance the performance of a SAN with two power levels, without the a priori knowledge about the number of the devices in the network;

- The performance validation and comparison of the proposed ESN with the traditional SAN. Furthermore, the behavior of the ESN has been tested also in terms of accuracy in comparison to other methods.

The rest of the chapter has the following structure. In the Section 2, an in-depth review of the related literature is detailed. A wide overview about the main ANNs classes is proposed in Section 3. Furthermore, the Section 4 presents the system model characterization and the chapter objectives. Section 5 describes the framework proposed and details the strategy behavior. Then, in Section 6 the access delay analysis is presented and the goodness of the proposed framework is validated. Finally, the conclusions are drawn in Section 7.

2 Related Works

The SAN contextualization to the next generation networks has been studied in many papers. Paper [12] investigates the application of the SAN scheme in comparison to the carrier sensing multiple access with collision avoidance within the context of the machine to machine communications in the IoT networks. The paper exhibits the improvements due to an enhanced version of the S-Aloha, in which the receiver adaptively learns the number of active device through the multi-hypothesis testing. Similarly, the paper [13] promotes the application of the SAN for the 5G IoT networks and shows the benefits implied by the regulation on the transmission power levels and the number of signals for one slot using a SIC receiver.

Paper [14] proposes an uncoordinated random access protocol which takes into account the limited-power requirements of the IoT devices, discussing a flexible frame structure for the proposed protocol. Furthermore, a multiple hypotheses testing analysis is proposed to identify the number of active IoT devices, on the basis of which the SIC power levels is properly adjusted. Differently, a novel random access protocol for the massive machine-type communications, based on the S-Aloha scheme, is proposed in [15], considering characteristics of the physical aspects such as the channel effects and the number of antennas. In addition, in order to provide energy efficient synchronization between the base station and the active users, in paper [15] a beacon phase is added at the beginning of each phase. Then, the access control barring technology and the transmission acknowledgment are adopted during the beacon phase, enhancing the flexibility and the reliability of the scheme.

As regards the ESNs, authors in [16] propose the binary particle swarm optimization method to determine the suitable matrix weight as connection between the ESN reservoir neurons and the output layer and to provide a time series prediction, while in paper [17] the ESN prediction accuracy in order to perform mobile communication traffic forecasting is improved by using complex network structures such as small worlds topologies. Similarly to [16], authors in [18] focus on the optimization of the ESN on the basis of the particle swarm optimization, performing both the single and the multi-objective optimizer structure. Differently, the paper [19] formulates a new learning algorithm based on the regularization method, providing a stable solution to the approximation function guaranteeing a good tradeoff between accuracy and smoothness. Furthermore, authors in [20] apply the ESN to the problem of the power supply prediction, by exhibiting the high performance reached by the ESN in forecasting temporal series. Likewise, paper [21] aims at predicting the network traffic levels, in order to guarantee quality of service by planning resource allocation, congestion control and so on. Within this context, the ESN plays a crucial role, by avoiding long learning processes.

Differently, during years, other ML techniques have been extensively applied to the networks problems strictly related to the access protocols. Specifically, in paper [22] the authors propose the joint optimization of both the resource allocation and the power control aspects, on the basis of the Multi-Armed Bandit strategy, by assuming the NOMA scheme and the SIC technology. In addition, paper [22] investigates the trade-off between the exploration and the exploitation phase of the ML solution which has been applied in a distributed manner. The Q-learning as been applied in paper [23] focusing on both the machine-to-machine (M2M) and human-to-human (H2H) communications. The paper aims at designing an novel random access channel scheme to control the M2M traffic limiting its impact on the cellular network taken into analysis. Within this landscape, the Q-learning access scheme promoting interaction between the M2M and the H2H communication scheme is presented, assuming a S-Aloha scheme. A Q-learning random access approach has been adopted also by [24], in which the energy efficiency of the

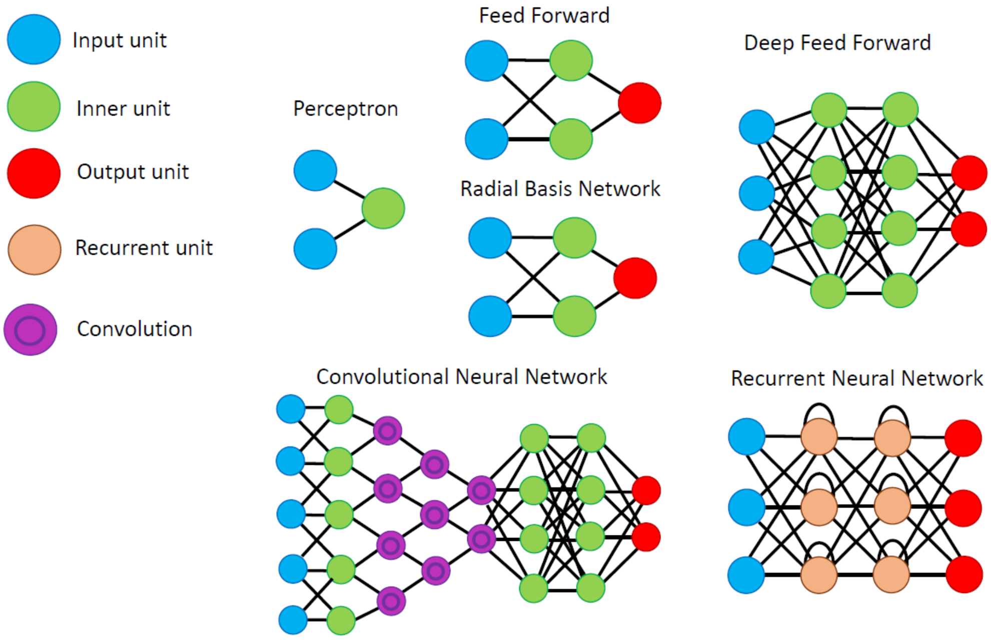


Figure 2: Neural network chart

update Q-value procedure and the storing process is the main objective, together with the minimization of the energy consumption of the whole M2M communication system considered. The support vector machine is applied in [25], in which the problem of the spectrum scarcity in relation to the interference among primary and secondary users is addressed. In this paper the ML is contextualized to the MAC identification, in order to change the users transmission parameters to improve the spectral efficiency. Deep reinforcement learning is adopted in [26], in which the usage of the same slot from different access protocol is taken in exam. More in depth, the agents of the proposed scheme does not have an a priori knowledge about the access scheme of the other networks but, on the basis of successive sampling, they learn the optimal transmission policy despite the presence of the external networks. In paper [27] authors propose the automation of the MAC protocols design on the basis of the machine perspective, particularly suitable for industrial applications and environments.

3 Artificial Neural Networks

The ANNs has recently become a wide-spread tool to provide effective solutions to a large class of problems, such as image classification, clustering, identification of trends data patterns, natural language processing, and recognition to name a few [4, 5, 6, 7, 8, 9]. Despite the massive application of the ANN in many different fields, the search for systematic procedures to strongly simplify the ANN development phase [4, 5, 6, 7, 8, 9] does not stop. Therefore, the ANNs provide a powerful and flexible tool to solve a wide variety of complex and nonlinear problems, justifying their extensive and traversal applications in different research areas. Exactly the way the human brain learns, the ANNs are able to learn by examples. Furthermore, ANNs can also be designed for specific contexts and to have a structure tailor-made for the considered application. The ANNs have numerous similarities with the human brain behavior; they both require a sort of dynamic adjustments of the synaptic connections [4, 5, 6, 7, 8, 9].

There exist many different types of ANNs, typically grouped on the basis of the ANN structure. In general terms, the topology of the ANN determines its behavior and its suitability for the application to different contexts. Therefore, diverse ANNs operate in different ways achieving different outcomes. The crucial point here is that ANNs are designed taking inspiration from how neurons in the brain work. As consequence, they learn more and improve more by increasing the data and their usage. In this sense, one of the key aspect of the ANNs in comparison to other traditional machine learning approaches is that the ANNs have the ability to dramatically improve performance as data as application grows.

Roughly speaking, a ANN consists of a large number of units which operate in a parallel way and are arranged in tiers. The input information arrives at the first tier and from that, the information crosses the network tier by tier, until it reaches the final tier which returns the outcome. The units belonging to different tiers are connected to the units of the previous and successive tier. Furthermore, through a process of weights adjustments, different levels of importance are assigned to the information crossing the network, in order to reach the desired output.

The life-cycle of the ANN can be characterized by two phases: the learning and the operating period. The information patterns cross the network via the input layer, which forward the information to the hidden units, and these in turn arrive at the output units. This wide spread model is namely as feed forward network as depicted in Figure 2. Each unit receives inputs from the units belonging to the previous layer, and the inputs are multiplied by the weights of the connections they cross. Each unit sums up all the inputs it receives and, considering the simplest networks, if the sum is more than a fixed threshold value, the unit triggers the units of the next level to whom they are connected. As previously anticipated, on the basis of its structure, the ANN exhibits strengths and weakness aspects. Therefore, there exist many kinds of ANNs. In addition to the feed forward ANN can be mentioned also the following ANN main structures [4, 5, 6, 7, 8, 9]

- **Perceptron:** The most straightforward neural model which takes and sums the inputs, applies activation function and sends the data to the output layer. The output function of this model in its simplest form can be expressed by

  $$y = sgn\left(\sum_{i=1}^{2} w_i x_i + \theta\right), \tag{1}$$

  with w_i are the weights, x_i represents the input and θ is the bias.

  $$sgn(s) = \begin{cases} 1, & \text{if } s > 0 \\ -1, & \text{otherwise.} \end{cases} \tag{2}$$

- **Radial Basis Function Neural Network:** Such neural networks are formed by two levels, and in the inner layer the features are combined using the radial basis function. This class of ANNs works well in classification and decision making processes. The corresponding output can be expressed as

$$y = \sum_{i=1}^{2} w_i h_i(x), \tag{3}$$

where

$$h(x) = exp\biggl(-\frac{(x-c)^2}{r^2}\biggr), \tag{4}$$

- **Deep Feed Forward:** This type of ANNs is the same of the Perceptron, but it exhibits a larger number of inner units. The output, in its simplest form, can be expressed as in (1) and (2).

- **Convolutional Neural Network:** is formed by one or more convolutional layers, which means that before passing the result to the next layer, the convolutional layer performs a convolution on the input. According to [28], the convolutional operation for each layer can be expressed as

$$F(a,b) = \sum_{c} \sum_{u,v} i_c(x,y) \cdot e_l^k(u,v), \tag{5}$$

where $i_c(x,y)$ represents the input tensor, while e_l^k is the convolutional kernel k^l of the l-th layer [28].

- **Recurrent Neural Network:** In this type of ANN the output of a layer is saved and fed back to the input. Such step significantly helps for predicting the outcome of the layer. As detailed in [29], the two main equations for this type of networks at each time step on the forward pass, are the following

$$\mathbf{h}^{(t)} = \sigma(W^{hx}\mathbf{x}^{(t)} + W^{hh}\mathbf{h}^{(t-1)} + \mathbf{b}_h), \tag{6}$$

and

$$\hat{\mathbf{y}}^{(t)} = softmax(W^{yh}\mathbf{h}^{(t)} + \mathbf{b}_y), \tag{7}$$

in which W^{hx} is the weights matrix between the input and the hidden layer, while W^{hh} represents the matrix of recurrent weights between the hidden layer and itself at adjacent time steps [29]. Furthermore, vectors bias are expressed by b_h and b_y.

As it is straightforward to deduce, different structures imply different strengths and weaknesses, which make the diverse ANNs types suitable or not on the basis of the problem addressed. In the Section 5 the ESN is presented and characterized in relation to the proposed problem scenario, object of study.

4 System Model

In accordance with Figure 3, the considered scenario consists of hierarchical network architecture, represented by one edge node (EN) on the top layer, assumed here located at the base station (BS) site [1] of the cellular 5G network, and J IoTDs at the bottom

[1] Hereafter, for the reader convenience, the terms EN and BS are used interchangeably

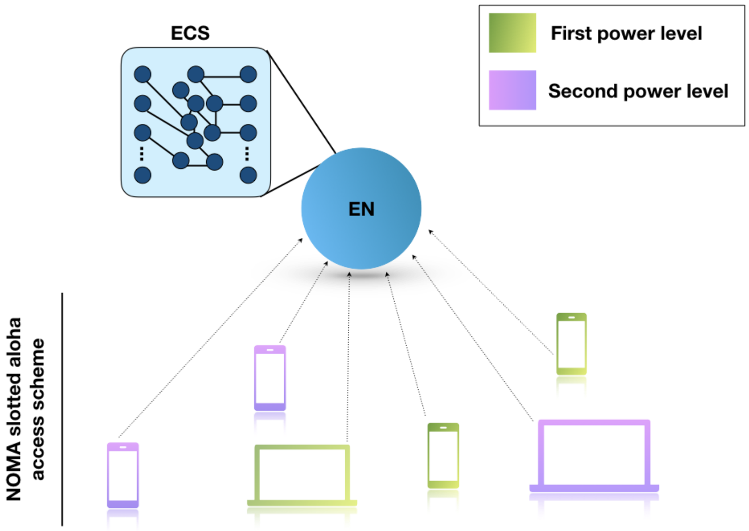


Figure 3: System Scenario

layer. The J IoTDs need to access the EN in order to obtain packet computation before a deadline, expressed by d. Due to the greater time required by the communication time, that spent in computation is negligible in comparison to the first. Furthermore, we assume that IoTDs share a same communication channel to access the EN facilities according to the SAN scheme as in [30]. In the shared channel (uplink) time is assumed divided into slots of a constant size with each slot equal to one packet transmission time. Without loss of generality of our proposal, we focus on a SAN scheme based on a single carrier NOMA technique, however, the proposed methodology can be easily extended to different NOMA alternatives [31]. More in depth, the uplink NOMA is considered compliant with [32], in which we assume that the J IoTDs are clustered in two groups, identifying the transmitter power level classes f_1 and f_2, with $f_1 > f_2$, hereafter used interchangeably to refer at the power levels and the corresponding clusters. Similarly, J' and J'' express the number of IoTDs belonging to the two power levels. In particular, we suppose here that IoTDs exploit channel inversion[2] on the basis of the received downlink reference signal sent out by the BS at the beginning of each IoTDs access period to properly select the power level [33]. According to this, each IoTD accesses the shared channel by selecting one of the two power levels as in [31] and the BS (i.e., the EN) is able to separate and detect packets simultaneously transmitted on the same slot with different power levels by means of the use of a SIC receiver as outlined in [31, 33]. We have assumed that each IoTD performs the channel inversion, i.e., selects the appropriate power level only when it becomes active and that channel conditions remain the same during its

[2] This is done under the assumption of a channel reciprocity.

operating time [32]. Furthermore, we make the abstraction of assuming a transmission attempt as a successful (i.e., associated packet correctly detected) whenever only one IoTD belonging to the cluster f_1, individually accesses the shared channel, independently from the number of transmission attempts of IoTDs belonging to the cluster f_2 on the same slot. Conversely, due to the NOMA-SIC technology, if a collision occurs at the power level f_1, the detection of packets transmitted with lower level f_2 is denied, despite the absence of collisions. Furthermore, we assume that each IoTD can hold only one packet and the outcome of any access attempt is immediately known at the end of the packet transmission time by receiving a positive/negative acknowledgement (ACK/NACK) message by the BS. Finally, we have assumed to adopt the Delayed First Transmission (DFT) approach as in [34, 35], according to which the same transmission probability is considered for all the packets to be sent out by the IoTDs, whether first or next transmission attempts. As a consequence, a newly arriving packet at a given IoTD has to wait for the same random backoff period as any other packet involved in collisions. A further discussion about the formal parts of the proposed model is provided in the next section.

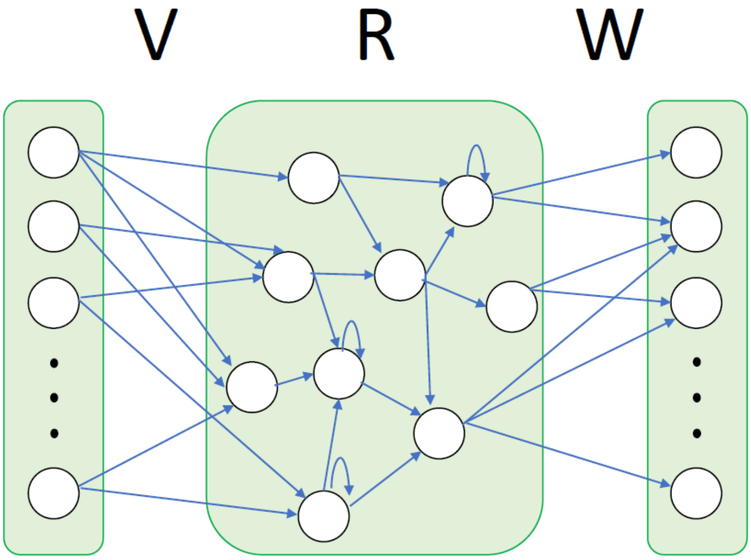


Figure 4: ESN Framework

5 Echo State Network

The general idea behind the ESN is the prediction of an output, on the basis of a given input signal, in reference to an output target $\mathbf{y}^{target}$, in order to minimize the gap between the input and the target values. More in details, the main problem here is to learn a model with output $\mathbf{y}$, lowering the difference between $\mathbf{y}$ and $\mathbf{y}^{target}$ as much as possible. In formal terms, the objective is the minimization of the error $E(\mathbf{y}, \mathbf{y}^{target})$ along a discrete time period T, in which the error can be defined as [10]

$$E(\mathbf{y}, \mathbf{y}^{target}) = \frac{1}{S} \sum_{h=1}^{S} \sqrt{\left(\frac{1}{T} \sum_{q=1}^{T} (y_i(q) - y_i^{target}(q) \right)^2}, \tag{8}$$

where S is the number of the samples considered. Aiming at increasing as much as possible the successful transmissions, the values p_{t_1} and p_{t_2} of the access probability of the IoTDs sharing the same channel are forecast. More in depth, the ESN application is performed to predict both the p_{t_1} and p_{t_2} values. The ESN, drawn in Figure 4, runs at the EN level, and its structure is characterized by the following features [10]

- neurons randomly connected;
- sparse connection links;
- large number of neurons;
- low in energy and time demand.

It is important to highlight here that the ESN application is motivated by its efficiency in terms of low computation time and energy cost [10]. The low computational complexity nature of the ESN finds its motivation in the usage of the reservoir module. In this context, the main benefit is by fixing the reservoir, performing training only at the non-recurrent stage. In detail, the ESN is formed by three main parts: the input weight matrix I, the reservoir weight matrix R, and the matrix W of the output weights. Let $\mathbf{x}^{q\times 1}$ be the input vector, supposing the reservoir weight matrix updating rule given by [10]

$$r^{m\times 1}(q) = tanh(\mathbf{W}_{in}^{m\times q}\mathbf{x}^{q\times 1}(q) + \mathbf{W}_{r}^{m\times 1}(q-1)), \tag{9}$$

where $r^{m\times 1}$ is a vector of internal units in the reservoir part, while $\mathbf{W}_{in}^{m\times q}$ represents the weights matrix associated to the connections existing between the input layer and the reservoir level. Finally, the $\mathbf{W}_{r}^{m\times 1}(q-1)$ is the recurrent weights matrix. Let $v(q)$ be the output vector and $\mathbf{W}_{out}^{q\times m}$ the weight matrix associated to the connection between the reservoir and the output layer. Therefore, the relationship between the reservoir and the output level is the following

$$v(q) = \mathbf{W}_{out}^{q\times m} r^{m\times 1}(q). \tag{10}$$

For a deeper discussion about the theoretical models involved in the ESN framework, we suggest to refer to [10]

5.1 An Algorithm Example

The proposed framework, summarized in Figure 5.1, which runs at the EN site, summarized in the following steps

1. On the basis of the outcome of the access attempts performed on the basis of the access probability predictions for the two clusters, the algorithm updates the p_{t_1} and p_{t_2} values.

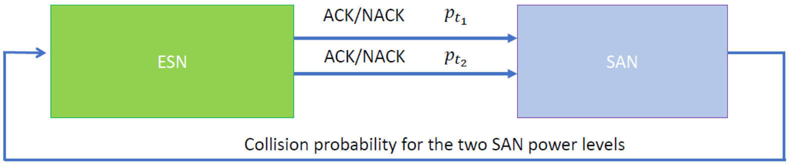


Figure 5: Algorithm example behavior.

2. The associated BS sends out their values embedded in the downlink ACK/NACK messages to all the IoTDs[3].

3. Repeat 1)-3) at each slot.

6 Access Scheme Analysis

The aim of this Section, is the throughput and delay analysis of the proposed ESN-SAN scheme under the assumption of a given number of IoTDs always active, which is useful to study the limits of the proposed solution under different system contexts. The throughput delay performance of a basic SAN scheme, and for the case of the use of deep learning method based on that proposed in [36], are also presented here for comparisons purposes.

As stated before, in order to make affordable our performance analysis due the high number of IoTDs sharing the common access channel to the EN, we have resorted to the use of the DFT model [34, 37]. According to the assumed working conditions, we have that the number of IoTDs contenting for accessing the shared uplink channel is fixed at each slot beginning, i.e., whenever an IoTD obtains a successful transmission attempt, a new packet transmission attempt is started on the next slot by the same or a different IoTD, i.e., the IoTDs number having their buffer full is constant on a slot basis. As previously presented in Section 4, we refer our analysis to the case of a SAN scheme with two power levels, i.e., f_1, f_2, with $f_1 > f_2$ and considering always as a success any IoTD individual access with transmission power f_1 whatever the outcome of the transmission attempt by IoTDs with power level f_2, i.e, success or collision. Hence, the probability that the transmission delay x_{d_1} of a packet generated at a given IoTD, enabled to use the power level f_1, is equal to l slots is

$$P\{x_{d_1} = l\} = p_{t_1}(1 - p_{t_1})^{J'-1}\{1 - p_{t_1}(1 - p_{t_1})^{J'-1}\}^{l-1}, \tag{11}$$

where p denotes the access probability for any IoT device belonging to the cluster f_1 having cardinality, i.e., the number of elements belonging to it, equals to J'.

From (11), we have that x_{d_1} results to be geometrically distributed with parameter $\tau = p_{t_1}(1 - p_{t_1})^{J'-1}$, depending on the p_{t_1} value and the number of IoTDs using the power level f_1. Here, a parameter of interest is the mean packet access delay $\bar{x}_{d_1}$ and its

[3] Note that in the case of idle uplink slots ACK/NACK messages are not sent out, hence, the p_{t_1} and p_{t_2} values remain the same as in the previous update.

$$P\{x_{d_1} = l\} = p_{t_2}(1 - p_{t_2})^{J''-1}[(1 - p_{t_1})^{J'} + J'p_{t_1}(1 - p_{t_1})^{J'-1}]$$
$$\{1 - p_{t_2}(1 - p_{t_2})^{J''-1}[(1 - p_{t_1})^{J'} + J'p_{t_1}(1 - p_{t_1})^{J'-1}]\}^{l-1} \tag{13}$$

reciprocal representing the SAN throughput η_1 for the power level f_1. In particular, $\bar{x}_{d_1}$ results to be

$$\bar{x}_{d_1} = \frac{1}{p_{t_1}(1 - p_{t_1})^{J'-1}} \tag{12}$$

hence, $\eta_1 = J'/\bar{x}_{d_1}$. Similarly, for the cluster transmitting with power f_2, we have (13), and $\bar{x}_{d_2}$ given by:

$$\bar{x}_{d_2} = \frac{1}{p_{t_2}(1 - p_{t_2})^{J''-1}[(1 - p_{t_1})^{J'} + J'p_{t_1}(1 - p_{t_1})^{J'-1}]} \tag{14}$$

with $\eta_2 = J''/\bar{x}_{d_2}$.

For a classical two power levels SAN scheme, p_{t_1} and p_{t_2} are a priori specified and, usually, fixed values. Conversely, in the case of the deep Q-reinforcement learning (DRL) approach the corresponding access probability are derived on a slot basis accordingly to a specific procedure detailed later. We would like to stress again that, as evident in (13), a successful transmission attempt on the power level f_2 strongly depends on the transmission attempt outcome on the power level f_1.

In the case of the DLR approach the corresponding access probability are derived accordingly to the procedure detailed later. We would like to stress again that, as it is evident in equation (13), a successful transmission attempt on the power level f_2 strictly depends on the transmission attempt (if any) outcome on the power level f_1.

In order to validate the goodness of the proposed approach, we have resorted to extensive numerical simulation aiming at deeply investigating the actual performance of the ESN based framework. The performance improvements under the assumed working condition are evident in Figure 6, where the reached SAN sum throughput is depicted for the whole system, i.e., for the two power levels. Therefore, the figure provides the throughput performance reached by the classical two power levels SAN scheme with fixed access probability values set as $p_t = p_{t2} = \frac{1}{K}$ [34, 37, 35] and $J' = J''$. The advantages introduced by the ESN are clearly evident in Figure 6, in which the devices reach greater throughput levels in comparison to the considered conventional SAN scheme. As known from the literature [37], the figure confirms that for the conventional SAN scheme the maximum throughput value is reached when the number of active IoTDs is equal to K. However, it is also evident in the figure that when the IoTDs number changes deviating from the optimum values the throughput performance significantly decreases. Differently, the ESN is able to dynamically rule the access probability on the basis of the system context, i.e., the number of IoTDs, keeping high throughput levels. Furthermore, we can also note in Figure 6 that the ESN approach outperforms the traditional SAN alternative. Furthermore, Figure 6 and Figure 6 depict the delay performance by considering both the clusters f_1 and f_2. Also in this case, the improvements due to the application of the ESN are evident, both in terms of maximum number of supported IoTDs simultaneously active under a fixed maximum mean access delay constraint or, equivalently, as the minimum

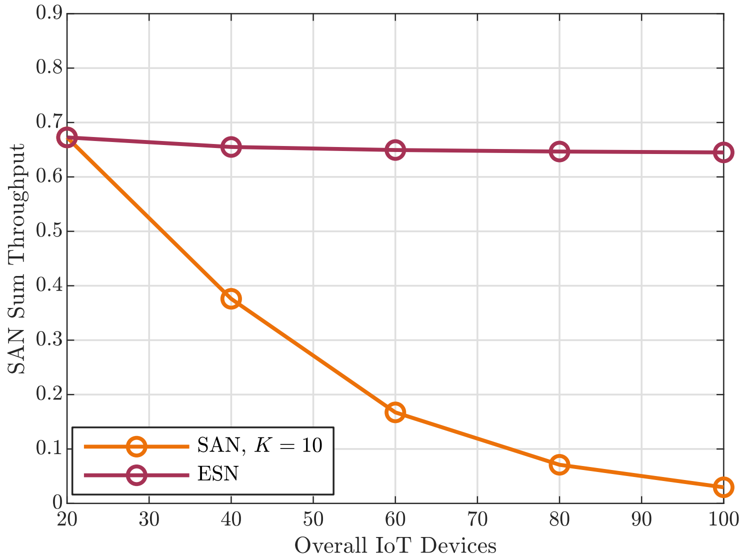


Figure 6: Two power levels SAN scheme sum throughput comparisons.

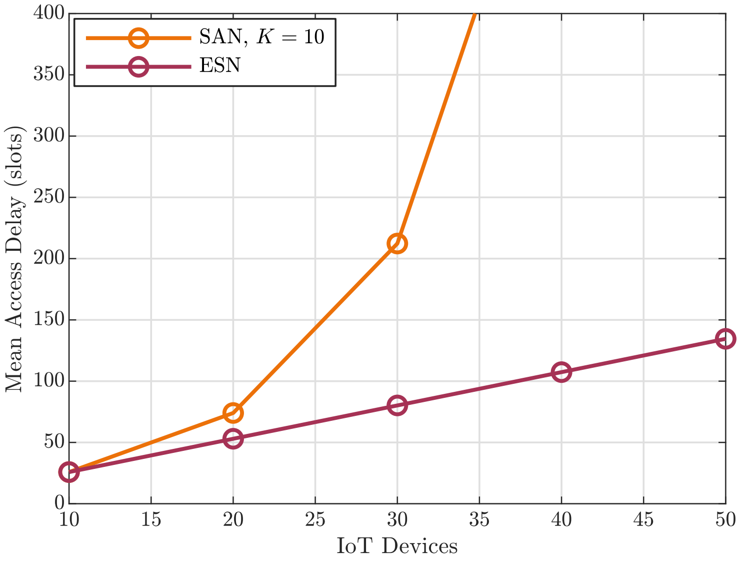


Figure 7: Mean Access Delay for the two power levels SAN scheme - cluster f_1.

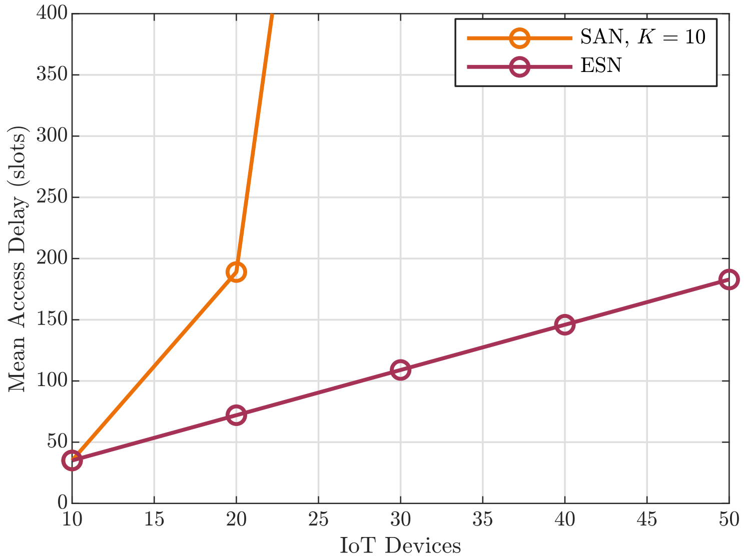


Figure 8: Mean Access Delay for the two power levels SAN scheme - cluster f_2.

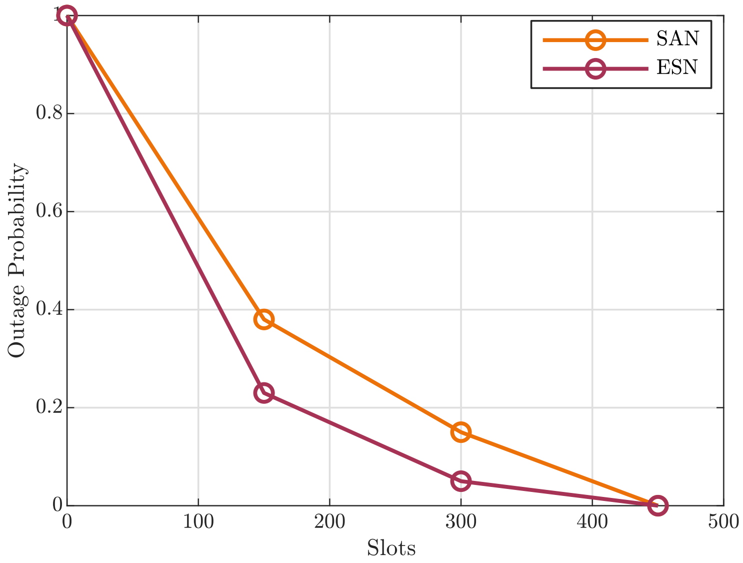


Figure 9: Outage probability for cluster f_1.

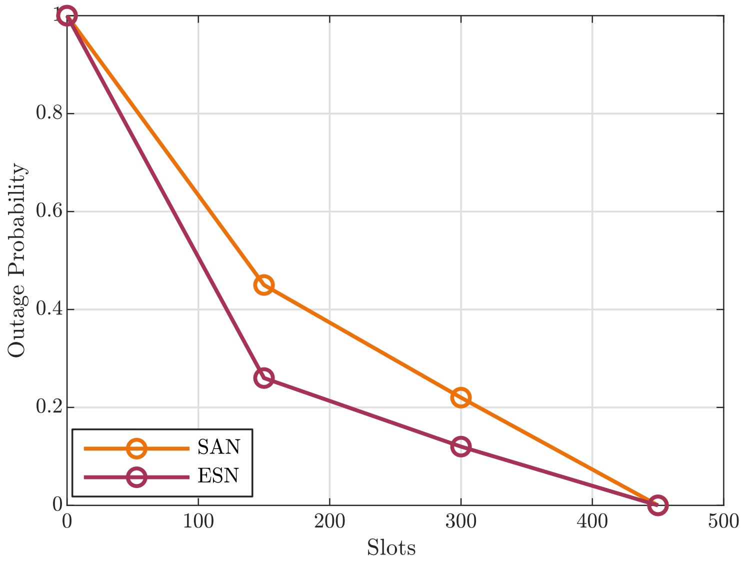


Figure 10: Outage probability for cluster f_2.

mean access delay value guaranteed for specified number of especially for high numbers of IoTDs simultaneously active. In particular, these advantages become even more evident when the number of IoTDs simultaneously active is high.

Furthermore, it is important to stress that if we remove the assumed working condition and consider each IoTD independently active with probability ϕ, roughly speaking, the number of IoTDs handled by the proposed ESN-SAN scheme for each power level has to be considered $1/\phi$ times higher for specified performance target values (i.e., maximum sum throughput or mean access delay). For example, from Figure 6, by assuming a target value for $\bar{x}_{d_1}$ of 150 slots and $\phi = 0.1$, the overall number of IoTDs handled by the cluster f_1 results to be 400 that highlights the effectiveness of the proposed ESN-SAN scheme in facing IoTDs access in 5G and B5G cellular networks. Figure 7 and Figure 8 show the delay complementary cumulative probability distribution, hereafter referred as outage probability, i.e. the probability that the delay is greater than a target value, by varying the mean number of packets that a device has to wait before a successful transmission, for a number of IoTDs equals to 20. As it is straightforward to note from both the Figures, the ESN reaches lower levels of outage probability in comparison to the classical SAN scheme, for both the power levels f_1 and f_2.

7 Conclusion

The chapter has shown the potentialities of the application of the ESN method contextualizing its application to the two levels SAN scheme access problem. The performance of the ESN-SAN framework has been tested in terms of mean packet access delay and sum throughput. Computer simulations and performance comparisons with a classical SAN scheme and with an alternative ML based approach are also presented to validate the goodness of our analysis and the better behavior of the proposed method in handling a massive IoTDs access in future 5G or B5G cellular networks. Interesting open issues and future work directions may consist of moving the ML techniques such as the ESN or the more general deep learning approaches on the devices, despite than on the central unit. This change of vision introduces several challenges which include greater convergence time and possible unstable devices behaviors, that need to be investigated.

References

[1] R. Mitra and D. Agrawal, "5g mobile technology: A survey," *ICT Express*, vol. 1, 01 2016.

[2] Q.-V. Pham, F. Fang, V. N. Ha, M. Jalil Piran, M. Le, L. B. Le, W.-J. Hwang, and Z. Ding, "A survey of multi-access edge computing in 5g and beyond: Fundamentals, technology integration, and state-of-the-art," 2019.

[3] J. Datta and H. Lin, "Detection of uplink noma systems using joint sic and cyclic fresh filtering," in *2018 27th Wireless and Optical Communication Conference (WOCC)*, pp. 1–4, April 2018.

[4] M. Kumar and S. Dargan, "A survey of deep learning and its applications: A new paradigm to machine learning," *Archives of Computational Methods in Engineering*, pp. 1–22, June 2019.

[5] P. N. Druzhkov and V. D. Kustikova, "A survey of deep learning methods and software tools for image classification and object detection," *Pattern Recognition and Image Analysis*, vol. 26, pp. 9–15, Jan 2016.

[6] E. M. Azoff, *Neural Network Time Series Forecasting of Financial Markets.* USA: John Wiley & Sons, Inc., 1st ed., 1994.

[7] H. B. Demuth, M. H. Beale, O. De Jess, and M. T. Hagan, *Neural Network Design.* Stillwater, OK, USA: Martin Hagan, 2nd ed., 2014.

[8] K. Rungta, *TensorFlow in 1 Day: Make your own Neural Network.* Publishdrive, May 2019.

[9] N. Shukla, *Machine Learning with TensorFlow.* USA: Manning Publications Co., 1st ed., 2018.

[10] H. Peng, C. Chen, C.-C. Lai, L.-C. Wang, and Z. Han, "A predictive on-demand placement of uav base stations using echo state network," *2019 IEEE/CIC International Conference on Communications in China (ICCC)*, Aug 2019.

[11] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Artificial neural networks-based machine learning for wireless networks: A tutorial," *IEEE Communications Surveys Tutorials*, vol. 21, pp. 3039–3071, Fourthquarter 2019.

[12] A. Mazin, M. Elkourdi, and R. D. Gitlin, "Comparison of slotted aloha-noma and csma/ca for m2m communications in iot networks," in *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, pp. 1–5, Aug 2018.

[13] E. Balevi and R. D. Gitlin, "A random access scheme for large scale 5g/iot applications," in *2018 IEEE 5G World Forum (5GWF)*, pp. 452–456, July 2018.

[14] M. Elkourdi, A. Mazin, and R. D. Gitlin, "Slotted aloha-noma with mimo beamforming for massive m2m communication in iot networks," in *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, pp. 1–5, Aug 2018.

[15] Q. He, Z. Chen, T. Q. S. Quek, Z. Chen, and S. Li, "A novel cross-layer protocol for random access in massive machine-type communications," in *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1–6, May 2018.

[16] S. Wu, Z. Wang, and D. Ling, "Echo state network prediction based on backtracking search optimization algorithm," in *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, pp. 661–664, March 2019.

[17] P. Yu, L. Miao, and G. Jia, "Clustered complex echo state networks for traffic forecasting with prior knowledge," in *2011 IEEE International Instrumentation and Measurement Technology Conference*, pp. 1–5, May 2011.

[18] N. Chouikhi, R. Fdhila, B. Ammar, N. Rokbani, and A. M. Alimi, "Single- and multi-objective particle swarm optimization of reservoir structure in echo state network," in *2016 International Joint Conference on Neural Networks (IJCNN)*, pp. 440–447, July 2016.

[19] Yong Song, Yibin Li, Qun Wang, and Caihong Li, "Multi-steps prediction of chaotic time series based on echo state network," in *2010 IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA)*, pp. 669–672, Sep. 2010.

[20] Yu Litao, Han Aoyang, Wang Li, Jia Xu, and Zhang Zhisheng, "Short-term load forecasting model for metro power supply system based on echo state neural network," in *2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, pp. 906–909, Aug 2016.

[21] O. A. Adeleke, "Echo-state networks for network traffic prediction," in *2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp. 0202–0206, Oct 2019.

[22] M. A. Adjif, O. Habachi, and J. Cances, "Joint channel selection and power control for noma: A multi-armed bandit approach," in *2019 IEEE Wireless Communications and Networking Conference Workshop (WCNCW)*, pp. 1–6, April 2019.

[23] L. M. Bello, P. Mitchell, and D. Grace, "Application of q-learning for rach access to support m2m traffic over a cellular network," in *European Wireless 2014; 20th European Wireless Conference*, pp. 1–6, May 2014.

[24] N. A. Shinkafi, L. M. Bello, D. S. Shu'aibu, and I. Saidu, "Energy efficient learning automata based qlrach (eela-rach) access scheme for cellular m2m communications," in *2019 2nd International Conference of the IEEE Nigeria Computer Chapter (NigeriaComputConf)*, pp. 1–6, Oct 2019.

[25] S. Hu, Y. Yao, and Z. Yang, "Mac protocol identification using support vector machines for cognitive radio networks," *IEEE Wireless Communications*, vol. 21, pp. 52–60, February 2014.

[26] Y. Yu, T. Wang, and S. C. Liew, "Deep-reinforcement learning multiple access for heterogeneous wireless networks," in *2018 IEEE International Conference on Communications (ICC)*, pp. 1–7, May 2018.

[27] H. B. Pasandi and T. Nadeem, "Challenges and limitations in automating the design of mac protocols using machine-learning," in *2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pp. 107–112, Feb 2019.

[28] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *CoRR*, vol. abs/1901.06032, 2019.

[29] Z. C. Lipton, "A critical review of recurrent neural networks for sequence learning," *CoRR*, vol. abs/1506.00019, 2015.

[30] M. Elkourdi, A. Mazin, E. Balevi, and R. D. Gitlin, "Enabling slotted aloha-noma for massive m2m communication in iot networks," in *2018 IEEE 19th Wireless and Microwave Technology Conference (WAMICON)*, pp. 1–4, April 2018.

[31] X. Shan, H. Zhi, P. Li, and Z. Han, "A survey on computation offloading for mobile edge computing information," in *2018 IEEE 4th International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing, (HPSC) and IEEE International Conference on Intelligent Data and Security (IDS)*, pp. 248–251, May 2018.

[32] M. S. Ali, H. Tabassum, and E. Hossain, "Dynamic user clustering and power allocation for uplink and downlink non-orthogonal multiple access (noma) systems," *IEEE Access*, vol. 4, pp. 6325–6343, 2016.

[33] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Communications Surveys Tutorials*, vol. 19, pp. 1628–1656, thirdquarter 2017.

[34] A. Mutairi, S. Roy, and G. Hwang, "Delay analysis of ofdma-aloha," *IEEE Transactions on Wireless Communications*, vol. 12, pp. 89–99, January 2013.

[35] L. Kleinrock and S. Lam, "Packet switching in a multiaccess broadcast channel: Performance evaluation," *IEEE Transactions on Communications*, vol. 23, pp. 410–423, April 1975.

[36] Y. Yu, T. Wang, and S. C. Liew, "Deep-reinforcement learning multiple access for heterogeneous wireless networks," in *2018 IEEE International Conference on Communications (ICC)*, pp. 1–7, May 2018.

[37] F. Tobagi, “Analysis of a two-hop centralized packet radio network - part i: Slotted aloha,” *IEEE Transactions on Communications*, vol. 28, pp. 196–207, February 1980.

Resources virtualization and task offloading towards the Edge in the IoT

Giovanni Merlino[1] **and Virginia Pilloni**[2]

[1] Department of Engineering, University of Messina, Italy
CINI, Smart Cities and Communities Lab, Research Unit of Messina, Italy
gmerlino@unime.it

[2] Department of Electrical and Electronic Engineering, University of Cagliari, Italy
CNIT, Research Unit of Cagliari, Italy
virginia.pilloni@unica.it

Abstract: *A significant role in the Internet of Things (IoT) will be played by mobile and low-cost devices, able to autonomously orchestrate workloads and share resources to meet their assigned application requirements. To increase efficiency, a recent trend promotes pushing computation from the remote Cloud as close to data sources as possible: this is the concept behind the Fog and Edge computing paradigms. Accordingly, distributed orchestration of node-hosted resources, so as to implement intelligent and fair allocation mechanisms, ensure the achievement of better performance at a lower network latency. Focusing on the aforementioned scenario, this Chapter presents a middleware architecture that relies on equipping co-located smart Edge and Fog devices with agents that virtualize real objects' features, resources and services. Thanks to an appropriate and fine-grained assessment of different combinations of offloading patterns, the middleware allows for a flexible and dynamic binding of the requested application workload to the physical IoT resources, in other terms an environment suitable for workload engineering.*

1 Introduction

The age of post-Clouds is already here, where unprecedented volume and variety of data are generated by things at the edge of networks, and many applications are being deployed on the edge networks to consume these Internet of Things (IoT) data. Some of the applications may require very short response times, some may convey personal data, and others may generate vast amounts of data. Cloud-based service models alone are not suitable for these applications.

Novel challenges are surfacing for advanced IoT services, also stemming from recent data production/consumption - and mobility - patterns, such as strict latency, constrained network bandwidth, constrained devices, uninterrupted services with intermittent connectivity, privacy and security due to the IoT environmental changes.

To address such challenges, the integration of Edge computing and IoT has emerged as a promising solution, often describing Fog computing as a paradigm to advance and support analytics [1].

Albeit, hidden behind that, lies the assumption that moving data closer to the site of its processing provides a definitive answer to most aforementioned issues. Whereas a suitably more general framework, to interpret and distill those challenges, consists of a set with fuzzy, ever-evolving boundaries for both data and workloads.

Thus, in more prescriptive terms, workloads need to be considered, as it happens with data, suitable for being parceled out and dispatched where needed, as well as moved elsewhere when required [2], better capturing this notion under the "offloading" [3] umbrella term.

This is particularly applicable in the presence of virtualization primitives, such a Virtual Machines or, more recently - and effectively in the IoT space, *containers* - in any case, self-contained units of (work)load that can be shifted around, across suitable infrastructure, almost effortlessly.

In a sense, even (owned and/or leased) infrastructure is morphing toward an heterogeneous spectrum of resources' sheer capabilities and arrangements, interaction and service models, ownership granularity and trustiness properties.

Thus, task offloading cannot just be seen as (yet another) useful implement in the toolbox of the DevOps community, i.e., in the service of software engineering alone, but also as the mechanism of choice to maximize use of, and value extraction from, globally distributed and decentralized infrastructure, especially when coupled with (suitable) resource virtualization schemes.

In terms of the categories of use cases where the impact of this perspective may be higher and its applicability more natural, any IoT-sourced *data pipeline* belonging to, e.g., the stream-processing layer of a *lambda architecture* [4], is amenable to be enhanced, or even rearranged, on the basis of the virtualization primitives and offloading patterns here proposed.

The remainder of this Chapter is organized as follows. Section 2 provides a description of the reference architecture and of the problem addressed. Section 3 presents the functionalities required to manage virtualization-related processes in the middleware. Section 4 describes the proposed offloading strategy, whereas Section 5 presents a resource allocation algorithm that enables a fair and efficient distribution of resources among the network objects while ensuring that the required quality of information is achieved. In Section 6 some key use cases are presented and discussed. Conclusions and further challenges are outlined in Section 7.

2 Problem Statement and Architecture

Among core challenges arising from the other half of this conceptual framework, i.e., data production/consumption, especially source and/or sinks, such as (typically IoT-hosted) sensors and/or actuators, lies the need to provide a more powerful and complete abstraction model, by including these *inputs/outputs* (I/O) among resources that should be shared according to demand and in compliance with multi-tenancy [5] requirements.

In particular, that translates into exposing resources as abstract handles (e.g., URIs, subject to availability of RESTful interfaces), possibly enhanced through semantic modeling [6], or through deeper, virtualization-level [7], techniques and primitives.

Indeed, workload (i.e., service) mobility is by definition deeply coupled with data mobility, as one cannot (easily) be moved around without also considering shifting the other. Also, data sources (and/or sinks) are especially important in this context, as producers (and/or consumers) of data for the services to process.

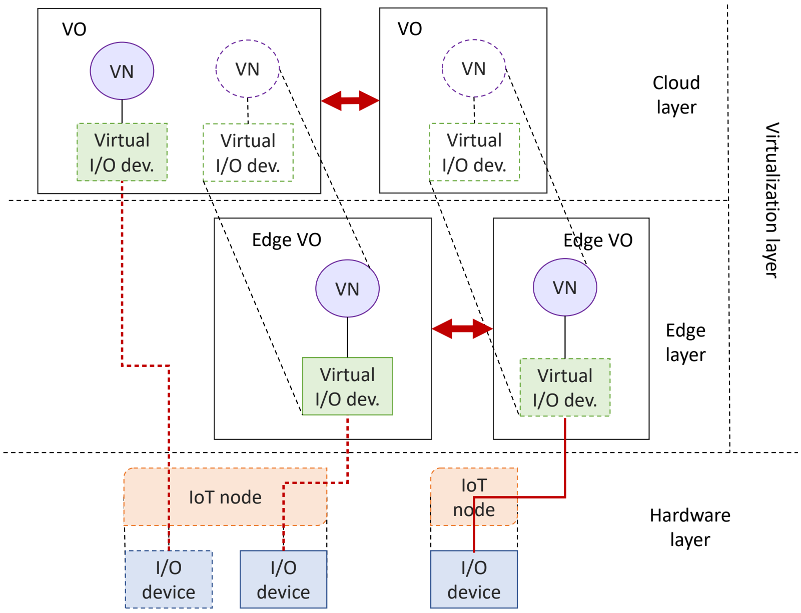


Figure 1: Architecture. Edge VO can actually be implemented either on a gateway or any other connected device: more in general, everywhere across to Cloud-to-Edge-to-Fog continuum

Moreover, from a software engineering perspective as well, it makes sense to address both categories under a unified, cohesive (programming and usage) model.

To proceed with the outline of this model, some definitions need to be introduced:

- The *Virtual Object (VO)* [8] represents the virtual counterpart of one or more real objects, and as such it inherits all their services, features and information mapping. As such, a VO is a digital representation of its counterpart, encapsulating the corresponding description and carrying all metadata required for the lifecycle of its hosted activities to be managed.
- An *IoT node* is any computing unit that hosts physical I/O devices as well as some logic.
- An *I/O device* is an entity (such as a sensor) that can be exported and attached but might not itself be programmed.
- We define a *virtual I/O device* [9] as an instance of a developer-friendly (e.g., pseudo-filesystem-based, or key-value store-based) interface, abstracting I/O primitives from the underlying I/O device, either in its entirety or as a subset or even superset (that is, a logical grouping of several I/O devices). The aforementioned interface can expose a remote I/O device as if it were local to its consumer, through e.g., a messaging system suitable for remoting I/O resources, i.e., a (brokered) dual (publish-subscribe plus remote procedure calls) bus [9].

- A *virtual IoT node (VN)* [9] is a self-contained and isolated environment that might be instantiated on top of either the datacenter-level infrastructure or a physical IoT node at the Edge. A VN can host logic and have attached virtual I/O devices, akin to an actual node.

As described in Fig. 1, VOs include two types of virtual entity, the VNs and the virtual I/O devices, that are the virtual counterparts of IoT nodes and I/O devices respectively. A VO description, and therefore its capabilities, change dynamically whenever a change in its (*backing*) IoT nodes or I/O devices is experienced (e.g., updated geographic location, a change in the amount of available resource, new services provided).

Categorizing a virtual I/O device as a distinct entity with respect to a VN introduces an essential abstraction, separating the virtualization of computing/storage resources from (typically) transducer-backed I/O ones. In particular, the ability to mediate, filter, virtualize, remote, and more in general address the latter, with their unique properties and behavior, separately from the former, provides higher flexibility in designing and assembling the infrastructure, and thus the services, of choice, whilst at the same time providing a more focused scope in the design and implementation of enabling (virtualization) mechanisms. Indeed, this approach provides a finer granularity, and a higher degree of freedom, over one of the core benefits provided by virtualization: the *multiplexing* property.

3 Virtualization

IoT scenarios are characterized by heterogeneous objects, which typically communicate using different languages. Therefore, in order for them to cope with heterogeneity-related issues and be able to interoperate and cooperate, they need to rely on a middleware that provides them with a common hardware abstraction layer, as well as access and communication functionalities [10]. This is made possible thanks to the use of ontologies and semantic description languages, which are used to define objects' characteristics, location, resources, services, and parameters related to the quality of service and information ensured [11]. Indeed, virtualization enables the acquisition, analysis and interpretation of information about the context across heterogeneous platforms. Furthermore, it supports service discovery and mash-up, efficient self-management and mobility management.

According to this vision, the virtualization layer introduced in Section 2 is in charge of creating, maintaining, coordinating and deleting VOs and the virtual IoT nodes and I/O devices that constitute them. Accordingly, the VO instance lifecycle can be described as follows

1. *Creation*: whenever a new object is detected, it is associated with a new VO instance [12], based on the information it provides about the virtual IoT nodes and I/O devices it manages, resources it exposes and services it can deliver. This information is then translated into a semantic language that is understandable by the other objects that belong to the system.

2. *Maintenance and coordination*: during the whole VO instance lifecycle, correct and efficient management and interoperation with the other instances need to be ensured by specific mechanisms. This includes prevention and resolution of conflicts

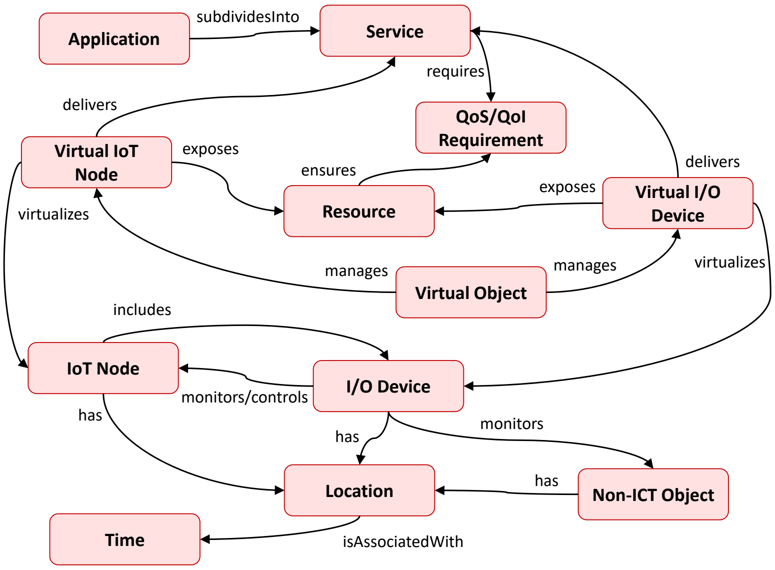


Figure 2: Information model

and instabilities deriving from concurrent invocations of the same VO and/or associated virtual nodes and I/O devices). Furthermore, the VO characteristics change dynamically whenever a change in its related IoT nodes or I/O devices is experienced (e.g. new geographic location, a change in the amount of available resource, new services provided).

3. *Deletion*: VOs have to be deleted when they are not being used anymore, i.e. when their associated IoT nodes and/or I/O devices are not reachable any longer.

To better describe the relationships among VOs, virtual nodes, virtual I/O devices and their real counterparts, i.e. IoT nodes and I/O devices, Fig. 2 introduces the reference information model for the system proposed in this Chapter.

It is modeled using as reference the information model proposed by the iCore FP7 project, where the concept of VO was first introduced [13]. The information model describes in graphic form the VO instance lifecycle introduced above. Specifically, VOs are associated with virtual nodes and I/O devices, which expose *Resources* that are used to deliver *Services* according to specific Quality of Service (QoS) and Quality of Information (QoI) requirements.

Note that services are the atomic functionalities that compose *Applications*. Further details about the efficient allocation of resources for service delivery will be provided in the following Sections. Several ontologies have been proposed in the literature to model measurement and post-measurement transformation of observations, which characterize virtual IoT nodes, virtual I/O devices, resources and QoS/QoI requirements. Some examples are the Sensor Model Language (SensorML) [14], the Semantic Sensor Network (SSN) and the Sensor, Observation, Sample and Actuator (SOSA) [15] ontologies. Services can be modeled using the most common Web service languages, such as Unified Service Description Language (USDL), Web Service Definition Language (WSDL), and Web Application Description Language (WADL) [16].

The *QoS/QoI requirement* block represents the characterization, in terms of some salient attributes represented in the form of metadata, of the goodness of the data collected, processed and flowing through a network [17][18]. QoI concerns the information that meets a specific user's needs at a specific time, place, physical location, and social setting. Some examples of QoI requirements are data sampling rate, precision, and provenance.

The *non-ICT object* block, which is inherited from the iCore information model, represents an object that is not provided with communication capabilities and that can only be virtualized if it is monitored by at least an I/O device (e.g. a door, a room).

Object mobility is supervised by the Location and Time blocks. The *Location* defines, either through coordinates or description (e.g. using the GeoNames ontology [19]), the position of a non-ICT object, an I/O device or an IoT node, which can be either static or mobile. The *Time*, defined in terms of date and time range, is associated with the Location. This association ensures the ability to know at any time when a particular IoT node, I/O device or non-ICT object is available, when it is possible to refer to it, and when the last time it was updated is.

As it will be better explained in the following Section, depending on the capabilities of the IoT nodes and I/O devices available, and on the application to be provided, the VO processes are run in the Cloud, gateway or on the physical devices. Scenarios where the VO services are distributed among these locations are also possible. Whenever an application request is received by the system, the virtualization layer dynamically maps the services that are required to deliver such application to the appropriate VOs, which take charge of their accomplishment by involving the relevant IoT nodes and I/O devices.

4 Offloading

The decentralized approach presented in this Chapter has the advantage of handling I/O devices and IoT nodes locally, without necessarily having to offload all the management burden to the Cloud [20]. This is possible especially when all or some of the VO functionalities are implemented in the Edge VOs. Indeed, since nodes that are located close to each other are often required to cooperate, for example when they are requested to monitor a specific area, they can be able to form a connected group relying on short-range communication technologies. In such a scenario, IoT nodes reside in the same area and are characterized by sufficient computational power and energy. The management process can be distributed among Edge VOs that are installed directly on the involved IoT nodes or an intermediate gateway, in such a way that the management of resources is as close as possible to the point where they are used. Accordingly, the communication between VOs and IoT nodes does not pass through the Internet network, i.e., it does not introduce overhead outside of the local network, and it is faster, characterized by lower latency, less expensive from an energy point of view, and incurs in no connectivity issues.

If the IoT nodes that are required to cooperate are not in the same area, but they are located in different places and at a significant distance, their communication can take place only through the Internet. In this case, there's no advantage of managing them

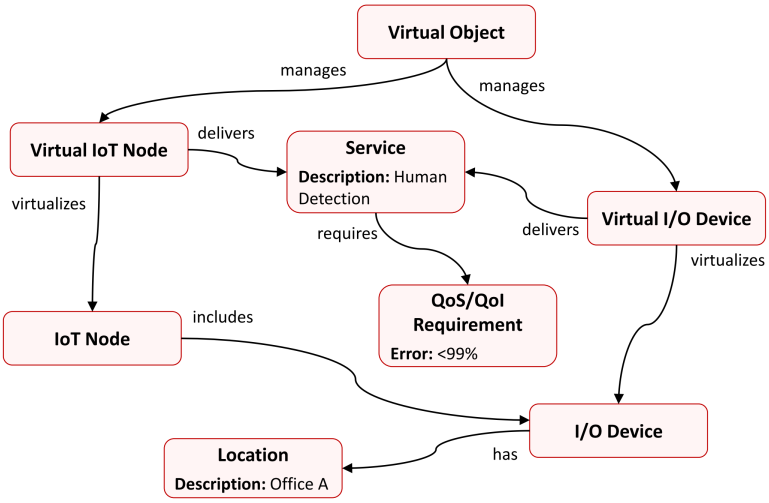


Figure 3: Portion of semantic description for a VO to be able to respond to the query of the running example

locally. Hence, the management process is carried out in the Cloud. Although the Cloud is characterized by greater resources, communication between VOs and IoT nodes has to pass through the Internet. Since VOs have to be frequently updated about the status and available resources of IoT nodes, so that inconsistencies are avoided, having VOs located remotely from IoT nodes would increase the amount of resource needed to synchronize them. Furthermore, higher latency is experienced. Therefore, the first solution, when applicable, is preferable.

It is evident that the choice of offloading strictly depends on the deployed application, i.e., on the correspondence between the services to be delivered and the relative position of the VOs that can provide them. When a new service has to be performed, the virtualization layer searches, among the available VO instances, those that have virtual IoT nodes and/or virtual I/O devices that can deliver it, also considering service requirements, e.g. in terms of QoS/QoI, location and time. To this end, the semantic description of the elements of the system becomes vital to implement effective search functionality. As a result, for each requested service k, a group of VOs capable of performing it is identified.

It is, therefore, crucial to decide how they should contribute to the execution of this service while considering the required QoS/QoI level. In the following, we consider that the target QoS/QoI level corresponds to a comprehensive reference service execution frequency F_k^{ref}. Note that the proposed solution can be generalized to other QoS/QoI requirements.

To explain the process more clearly, a running example is here outlined: suppose that a request was received for the evaluation, every minute, of human presence inside office A, with an error no higher than 99%, minimizing processor usage. Accordingly, the virtualization layer analyses the request and determines the VO instances that can respond to it. Hence, the virtualization layer starts searching for VO instances characterized by parameters equal to those described in Fig. 3. Suppose that the virtualization layer finds three VOs that match the queried one, which correspond to the following virtual I/O devices: a PIR (Passive Infra-Red) sensor, which captures human movements; a camera, which uses pattern recognition to detect human faces; and a thermal sensor, which detects body heat. The fact that the same service can be provided by such heterogeneous devices using so different functionalities, is completely transparent to the virtualization layer, which can manage all of them simply by managing their VOs and related attributes. At this stage, the virtualization layer sends a request for human presence detection to one of the VOs, including also the required frequency $F^{ref} = 1/60$ Hz, and the resource usage to be minimized, i.e., processor usage. The VOs can then start negotiating to find the optimal frequency assignment strategy, using the approach described in Section 5, regardless of their location concerning their corresponding virtual I/O devices. Indeed, since VOs are virtually directly connected between one another, they could negotiate even if one was located in the I/O device, one in an intermediate gateway and one in the Cloud, as it is better explained in Section 2.

5 Resource Allocation

It is evident from the considerations of the previous Sections that assessing the amount of resource available to deliver a specific set of services and allocating them efficiently is crucial for the system not to run into resource starvation. In particular, the resource allocation strategy presented in this Chapter is based on a fair distribution of service workload among the nodes that can deliver it. This means that nodes with greater available resources can take on more workload compared to nodes with lower resources. This prevents situations where some nodes are stuck due to a lack of residual resources, while others are underloaded.

This Section first presents (Subsection 5.1) an analysis of the most critical resources for IoT systems, providing, for each resource type, a model that measures the impact of service workload compared with the amount of resource that is available. Accordingly, an efficient and fair resource allocation model is presented in Subsection 5.2.

5.1 Resource Model

5.1.1 Lifetime

As defined in [21], the lifetime of a node is the time until its full functionality cannot be ensured anymore due to node battery depletion. Therefore, the lifetime of the node associated with VO i at time t is defined as

$$\tau_i^{lftm} = \frac{E_i^{res}}{\sum_k E_{ik}^c \cdot f_{ik}} \tag{1}$$

where E_i^{res} is its residual energy, E_{ik}^c is the energy consumed by the node associated with VO i to perform service k, and f_{ik} is the frequency at which VO i performs service k. This means that the lifetime of a node depends on the frequency at which the services assigned to it are performed.

5.1.2 Storage Capacity

The storage capacity of a node decreases according to the frequency at which data are stored in it and to the amount of data stored. Analogously to the definition of node lifetime, we define the storage capacity depletion time of the node associated with VO i as

$$\tau_i^{stor} = \frac{M_i^{res}}{\sum_k D_k \cdot f_{ik}} \tag{2}$$

with M_i^{res} residual memory expressed in bits, and D_k amount of data to be stored for service k. Note that residual memory can change over time, not only because of its usage but also because its stored data can be moved to another location.

5.1.3 Processor

The processor occupancy is measured as the ratio between the processing speed required to execute a service before its deadline and the available processing speed. If service k is performed at a frequency f_{ik}, this means that the total processor occupancy of the node associated with VO i can be defined as

$$\Theta_i^{proc} = \sum_k \frac{1}{S_i^{proc}} \cdot \frac{N_k^{instr} \cdot f_{ik}}{t_k^{dl}} \tag{3}$$

where N_k^{instr} is the number of instructions that need to be processed to perform service k, t_k^{dl} is service k's deadline, and S_i^{proc} is the processing speed for the node associated with VO i. Note that t_k^{dl} cannot be higher than the amount of the time between two subsequent executions of service k, and is part of the QoS/QoI requirements for service k.

5.1.4 Bandwidth

Analogously to the analysis made for the processor occupancy, and considering that the bandwidth needed by the node associated with VO i to transmit the output data for service k is proportional to its bitrate D_k/t_k^{dl} and to the frequency f_{ik} at which service k is executed, we define the bandwidth occupancy as

$$\Theta_i^{BW} = \sum_k \frac{1}{B_i^{av}} \cdot \frac{D_k \cdot f_{ik}}{t_k^{dl}} \tag{4}$$

where B_i^{av} is the bandwidth available for i.

5.2 Fair Resource Allocation Strategy

The equation that describes the use of the $\mathcal{R}$ set of resources of VO i for the $\mathcal{K}$ set of services can be generalized as

$$\Theta_i(\mathcal{K}) = \frac{1}{\tau_i^{lftm}} + \frac{1}{\tau_i^{stor}} + \Theta_i^{proc} + \Theta_i^{BW} = \sum_{r \in \mathcal{R}} \theta_{ir}(\mathcal{K}) = \sum_{r \in \mathcal{R}} \sum_{k \in \mathcal{K}} R_{ikr} \cdot f_{ik} \tag{5}$$

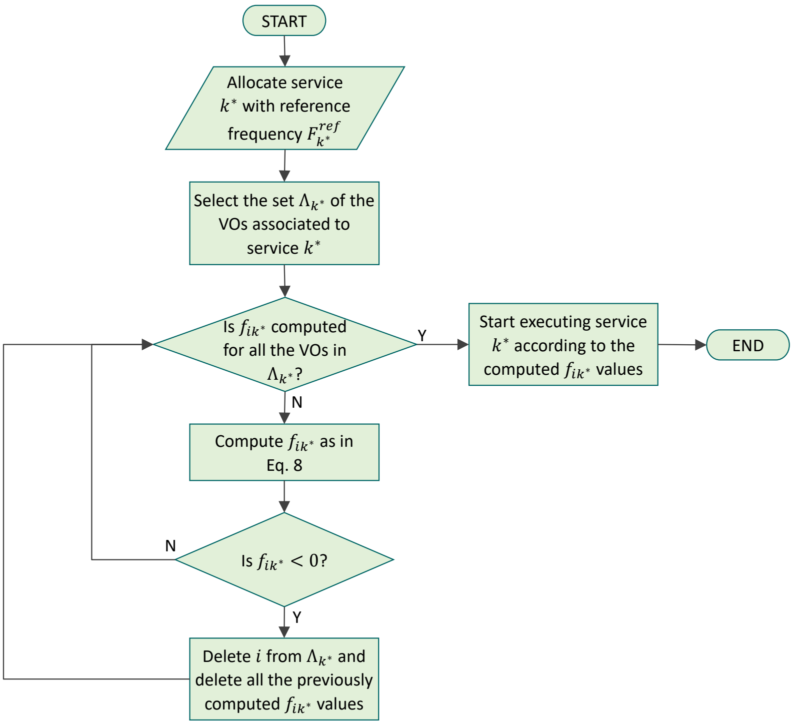

START
Allocate service k^* with reference frequency $F_{k^*}^{ref}$
Select the set Λ_{k^*} of the VOs associated to service k^*
Is f_{ik^*} computed for all the VOs in Λ_{k^*}?
Y
Start executing service k^* according to the computed f_{ik^*} values
END
N
Compute f_{ik^*} as in Eq. 8
Is $f_{ik^*} < 0$?
N
Y
Delete i from Λ_{k^*} and delete all the previously computed f_{ik^*} values

Figure 4: Flowchart of the proposed resource allocation strategy

with

$$R_{ikr} = \begin{cases} E^c_{ik}/E^{res}_i, & \mathit{if}\ r = \text{lifetime} \\ D_k/M^{res}_i, & \mathit{if}\ r = \text{storage} \\ N^{instr}_k/(S^{proc}_i \cdot t^{dl}_k), & \mathit{if}\ r = \text{processing} \\ D_k/(B^{av}_i \cdot t^{dl}_k), & \mathit{if}\ r = \text{bandwidth} \end{cases}$$

Whenever a new service k^* needs resources to be allocated for its execution, a frequency of execution is assigned to each VO i that is associated with that service, i.e. the VOs that belong to the set Λ_{k^*}. To ensure a fair distribution of resources, such frequency of execution f_{ik^*} needs to fulfill the following conditions

$$\begin{aligned} &\sum_{r\in\mathcal{R}} \alpha_r \cdot (R_{ik^*r} \cdot f_{ik^*} + \theta_{ir}(\mathcal{K})) = \sum_{r\in\mathcal{R}} \alpha_r\,(R_{jk^*r} \cdot f_{jk^*} + \theta_{jr}(\mathcal{K})) \quad \forall\{i,j\} \in \Lambda_{k^*} \\ &\sum_{j\in\Lambda_{k^*}} f_{jk^*} = F^{ref}_{k^*} \\ &\sum_{r\in\mathcal{R}} \alpha_r = 1 \qquad\qquad \forall i \in \Lambda_{k^*} \end{aligned} \tag{6}$$

where $F^{ref}_{k^*} = \sum_j f_{jk^*}$ is the reference frequency that has to be ensured according to the QoS/QoI requirements, and $\alpha_r > 0$ is a normalized weighting factor that is used by the system to weigh the impact of resource usage differently according to its needs. If, for instance, the system needs to save storage capacity more than the other resources, its weighting factor is set higher than the others. The first condition ensures that resources are fairly shared among the VOs in Λ_{k^*}. The second condition ensures that the quality requirements are fulfilled.

The first condition can be reformulated to make f_{jk^*} explicit

$$f_{jk^*} = \frac{1}{\sum_{r\in\mathcal{R}} \alpha_r \cdot R_{jk^*r}} \cdot \sum_{r\in\mathcal{R}} \alpha_r \left(R_{ik^*r} \cdot f_{ik^*} + \theta_{ir}(\mathcal{K}) - \theta_{jr}(\mathcal{K})\right) \tag{7}$$

Substituting it in the second condition, it is possible to define it as

$$f_{ik^*} = \frac{1}{\sum_{j\in\Lambda_{k^*}} \frac{1}{\sum_{r\in\mathcal{R}} \alpha_r \cdot R_{jk^*r}}} \cdot \frac{F_{k^*}^{ref}}{\sum_{r\in\mathcal{R}} \alpha_r \cdot R_{ik^*r}} + \frac{\sum_{j\in\Lambda_{k^*}} \sum_{r\in\mathcal{R}} \alpha_r \cdot (\theta_{jr}(\mathcal{K}) - \theta_{ir}(\mathcal{K}))}{\sum_{r\in\mathcal{R}} \alpha_r \cdot R_{ik^*r}} \tag{8}$$

It may happen that Eq. 8 is fulfilled for values of $f_{ik^*} < 0$. This means that the corresponding VO is already consuming much more resources than the other VOs and there are no physically possible values of f_{ik^*} that would lead it to a resource consumption that is comparable to that of the other VOs. Therefore, f_{ik^*} is set to 0 for that VO and the resource allocation is started again considering the remaining VOs in Λ_{k^*}.

Fig. 4 summarizes in graphical form the main steps of the resource allocation mechanism.

6 Key Use Cases

In this Section, some key use cases will be presented and discussed to better clarify the whole process of resource allocation and offloading.

The reference scenario is depicted in Fig. 5. The application that the system is required to execute is identifying when a specific user is at home, at work, or going at home or work, in order to switch on the air conditioner and set it correctly so that the temperature is correct when the user gets in. Therefore, the basic services are three:

Use case 1: Identifying the user's position: this is done thanks to the PIR sensor at home, the Bluetooth beacon at work and the GPS of the phone. The objective, in this case, is to optimize the nodes' lifetime. The reference frequency is 1 sample every 2 minutes, i.e. $F_1^{ref} = 0.0083$ Hz.

Use case 2: Measuring the temperature at home: temperature sensors are used for this purpose. These sensors do not have a network interface and are directly connected to a sink through a wire. The sink is also their gateway to communicate with the Cloud. The objective is to optimize nodes' lifetime. The reference frequency is 1 sample every 5 minutes, i.e. $F_2^{ref} = 0.0033$ Hz.

Use case 3: Measuring the temperature at work: temperature sensors are used in this case as well. However, sensors are equipped with a Bluetooth network interface, through which they can communicate with a gateway that connects them with the Cloud. The objective is to optimize nodes' lifetime and bandwidth.

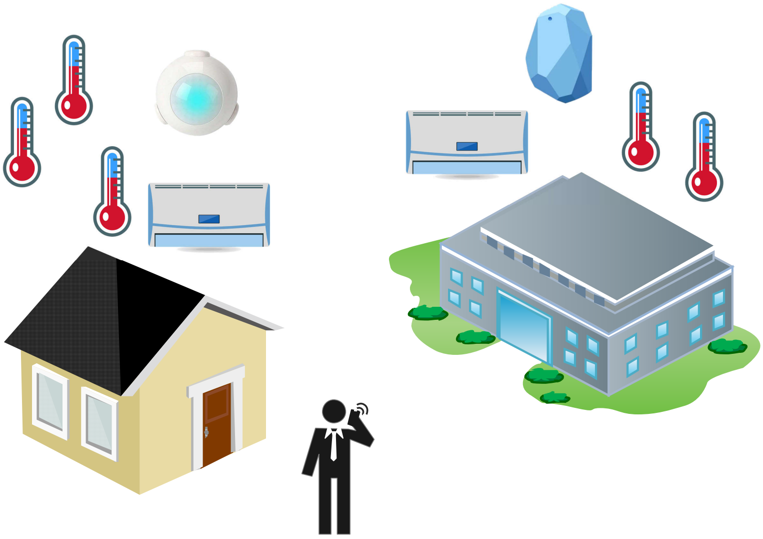


Figure 5: Use case scenario

The reference frequency is again 1 sample every 5 minutes, i.e. $F_3^{ref} = 0.0033$ Hz.

The resources available for the nodes and required for the considered services are described in Table 1.

6.1 Use Case 1

Regardless of the fact that the IoT nodes involved in this use case can host some VO functionalities or not, they are all located in different places. For this reason, the resource allocation can be performed neither locally or on the Edge, but it has to be offloaded to

Device	**Location**	**Provided service**	R_{ik^*1}	$\theta_{i1}(\mathcal{K})$	R_{ik^*4}	$\theta_{i4}(\mathcal{K})$
BT beacon 1	Office 1	1	0.002	0	NA	NA
GPS 1	User (mobile)	1	0.004	0.75	NA	NA
PIR sensor 1	Home	1	0.001	0	NA	NA
Temperature sensor 1	Home	2	0.001	0	NA	NA
Temperature sensor 2	Home	2	0.0007	0	NA	NA
Temperature sensor 3	Home	2	0.005	0	NA	NA
Temperature sensor 4	Office	3	0.002	0	0.1	0
Temperature sensor 5	Office	3	0.003	0	0.08	0

Table 1: Description of resources available for the nodes and required for the services for the reference scenario

Device	f_{i1} **[Hz]**	Θ_i^{init} **[Hz]**	Θ_i^{final} **[Hz]**
BT beacon 1	0.028	0	5.55e-6
GPS 1	0	0.75	0.75
PIR sensor 1	0.056	0	5.55e-6

Table 2: Result of the resource allocation for use case 1

Device	f_{i2} **[Hz]**	Θ_i^{init} **[Hz]**	Θ_i^{final} **[Hz]**
Temperature sensor 1	0.0013	0	1.27e-6
Temperature sensor 2	0.0018	0	1.27e-6
Temperature sensor 3	0.0002	0	1.27e-6

Table 3: Result of the resource allocation for use case 2

the Cloud.

The system retrieves from the VOs the information required for resource allocation and computes f_{i1}. The results are reported in Table 2, where Θ_i^{init} and Θ_i^{final} measure respectively the use of resources before and after service 1 has been allocated. Since the resource consumption for GPS 1 is already high and service 1 does not require a high amount of resources, there are no feasible values of f_{i1} for it. Therefore, the resource allocation involves only BT beacon 1 and PIR sensor 1.

6.2 Use Case 2

In this use case the I/O devices, i.e. the temperature sensors, do not have a network interface and rely upon the sink which serves as a gateway towards the Cloud for all of them. Consequently, their VOs reside in the gateway itself. Therefore, even though they are located close to each other and resource allocation can be performed locally, their VOs' information is stored on the gateway. For this reason, resources are allocated by the gateway, i.e. on the Edge, and the result is used to set the sensors appropriately.

The results are shown in Table 3. Remind that Θ_i^{init} and Θ_i^{final} measure respectively the use of resources before and after service 2 has been allocated. Also in this case, since the reference frequency and the amount of resource required is low, the total use of resources remains low even after the resources have been allocated.

6.3 Use Case 3

The temperature sensors used in this use case are IoT nodes with computational capabilities. For this reason, their VOs reside in the physical node. The resource allocation can be performed by the nodes, which can communicate directly, e.g. using short-range device-to-device communication, to find the optimal frequency value.

This use case has the objective of optimizing two types of resources, namely lifetime and bandwidth. For this reason, α_r should be set considering which resource type is more critical for the system. Fig. 6 and Fig. 7 respectively show how resulting frequency and resource usage for different resource types change when α_r changes. It is evident that when α_1 is low (i.e. α_4 is high) the impact of the bandwidth on resource allocation is higher. Therefore, the optimization is more effective on bandwidth usage, which at the

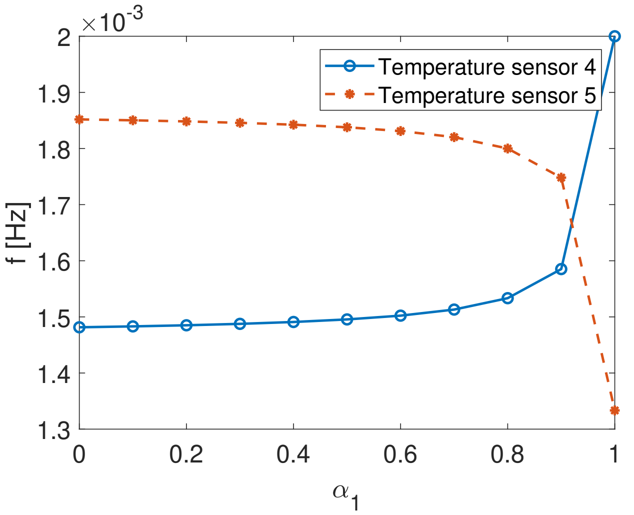
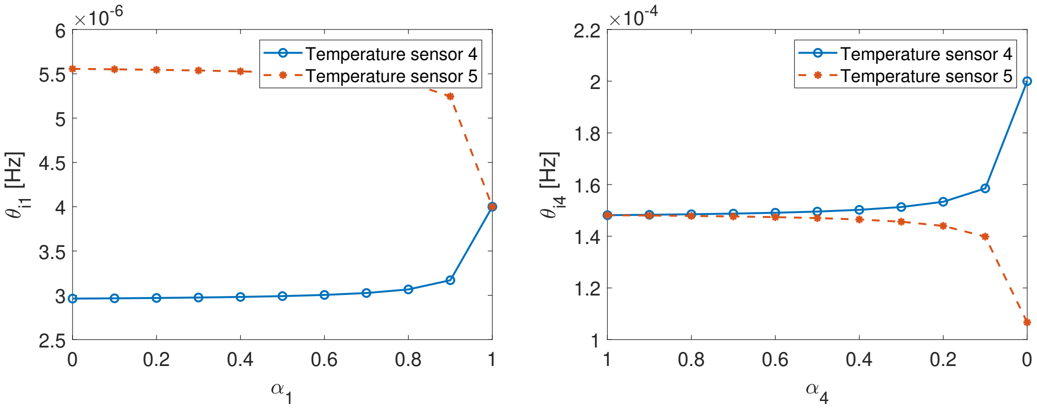


Figure 6: Resulting frequency for different values of α_1, for use case 3



(a) Lifetime

(b) Bandwidth

Figure 7: Resulting resource usage differentiated by resource type for use case 3, considering different values of α_r

end of the allocation is equal for both nodes. On the other end, when α_1 is high (i.e. α_4 is low) the impact of the lifetime on resource allocation is higher. Since service 3 requires more bandwidth usage than lifetime usage (see Table 1), the impact of lifetime usage on resource allocation remains quite low until α_1 is higher than 0.8. This is also confirmed by Fig. 6, where values of $\alpha_1 > 0.9$ correspond to a higher frequency for temperature sensor 4, which has higher bandwidth usage, compared to temperature sensor 5, which has higher lifetime usage.

7 Conclusions and challenges

In order to efficiently control and manage the allocation of services to the heterogeneous resource-constrained objects that characterize the IoT, this Chapter proposed a middleware architecture that makes use of virtualization and offloading mechanisms to flexibly and dynamically orchestrate a fair distribution of applications' workload.

Among key challenges, the choice of the ontology to be used for semantic description is crucial for an effective and efficient selection of the VOs that can respond to a specific query, whilst at the same time compatible with current user demand in terms of multi-tenancy requirements.

Which node coordinates the resource allocation is still to be decided, also taking into account that resource allocation can constitute a not-negligible workload by itself, depending on how it is managed. One strategy to reduce this burden is using a distributed approach, at least when any fallback, such as a centralized (e.g., Cloud-side) coordination, turns out to be unavailable, or undesirable. Indeed, the resource allocation strategy described in Section 5 is particularly suitable for distributed negotiation mechanisms such as consensus algorithms. The use of the most appropriate resource allocation strategy may be also selected from time to time considering the impact of different algorithms on different resource types, considering the ones that are selected to be optimized.

References

[1] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, *Fog Computing: A Platform for Internet of Things and Analytics.* Cham: Springer International Publishing, 2014, pp. 169–186.

[2] G. Merlino, R. Dautov, S. Distefano, and D. Bruneo, "Enabling Workload Engineering in Edge, Fog, and Cloud Computing through OpenStack-based Middleware," *ACM Transactions on Internet Technology*, vol. 19, no. 2, 2019.

[3] H. Flores, Xiang Su, V. Kostakos, A. Y. Ding, P. Nurmi, S. Tarkoma, P. Hui, and Y. Li, "Large-scale offloading in the Internet of Things," in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, March 2017, pp. 479–484.

[4] M. Kiran, P. Murphy, I. Monga, J. Dugan, and S. S. Baveja, "Lambda architecture for cost-effective batch and speed big data processing," in *2015 IEEE International Conference on Big Data (Big Data)*, 2015, pp. 2785–2792.

[5] S. Cherrier, Z. Movahedi, and Y. M. Ghamri-Doudane, "Multi-tenancy in decentralised IoT," in *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, Dec 2015, pp. 256–261.

[6] S. Alam, M. M. R. Chowdhury, and J. Noll, "SenaaS: An event-driven sensor virtualization approach for Internet of Things cloud," in *2010 IEEE International Conference on Networked Embedded Systems for Enterprise Applications*, Nov 2010, pp. 1–6.

[7] M. Samaniego and R. Deters, "Supporting iot multi-tenancy on edge devices," in *2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, Dec 2016, pp. 66–73.

[8] A. Somov, C. Dupont, and R. Giaffreda, "Supporting smart-city mobility with cognitive Internet of Things," in *2013 Future Network & Mobile Summit*. IEEE, 2013, pp. 1–10.

[9] D. Bruneo, S. Distefano, F. Longo, G. Merlino, and A. Puliafito, "I/Ocloud: Adding an IoT dimension to cloud infrastructures," *Computer*, vol. 51, no. 1, pp. 57–65, 2018.

[10] A. H. Ngu, M. Gutierrez, V. Metsis, S. Nepal, and Q. Z. Sheng, "IoT middleware: A survey on issues and enabling technologies," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 1–20, 2016.

[11] M. Nitti, V. Pilloni, G. Colistra, and L. Atzori, "The virtual object as a major element of the Internet of Things: a survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1228–1240, 2015.

[12] iCore, "D3.1: Virtual Object Requirements and Dependencies," 2012.

[13] P. Vlacheas, R. Giaffreda, V. Stavroulaki, D. Kelaidonis, V. Foteinos, G. Poulios, P. Demestichas, A. Somov, A. R. Biswas, and K. Moessner, "Enabling smart cities through a cognitive management framework for the Internet of Things," *IEEE communications magazine*, vol. 51, no. 6, pp. 102–111, 2013.

[14] O. OGC, "SensorML: Model and XML Encoding Standard," *Doc No. OGC*, pp. 12–000, 2014.

[15] K. Janowicz, A. Haller, S. J. Cox, D. Le Phuoc, and M. Lefrançois, "SOSA: A lightweight ontology for sensors, observations, samples, and actuators," *Journal of Web Semantics*, vol. 56, pp. 1–10, 2019.

[16] M. Klusch, P. Kapahnke, S. Schulte, F. Lecue, and A. Bernstein, "Semantic web service search: a brief survey," *KI-Künstliche Intelligenz*, vol. 30, no. 2, pp. 139–147, 2016.

[17] C. Bisdikian, L. M. Kaplan, and M. B. Srivastava, "On the quality and value of information in sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 9, no. 4, p. 48, 2013.

[18] V. Pilloni, L. Atzori, and M. Mallus, “Dynamic involvement of real world objects in the IoT: A consensus-based cooperation approach,” *Sensors*, vol. 17, no. 3, p. 484, 2017.

[19] S. A. U. Nambi, C. Sarkar, R. V. Prasad, and A. Rahim, “A unified semantic knowledge base for IoT,” in *2014 IEEE World Forum on Internet of Things (WF-IoT)*. IEEE, 2014, pp. 575–580.

[20] M. Díaz, C. Martín, and B. Rubio, “State-of-the-art, challenges, and open issues in the integration of Internet of Things and cloud computing,” *Journal of Network and Computer applications*, vol. 67, pp. 99–117, 2016.

[21] H. Yetgin, K. T. K. Cheung, M. El-Hajjar, and L. H. Hanzo, “A survey of network lifetime maximization techniques in wireless sensor networks,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 828–854, 2017.

Security and privacy in the IoT: how to enforce standard communication technologies with efficient and flexible mechanisms

Antonio Suriano, Domenico Striccoli, Giuseppe Piro, Antonio Antenore, Gennaro Boggia

DEI, Politecnico di Bari. Via Orabona 4, Bari, Italy

CNIT, Consorzio Nazionale Interuniversitario per le Telecomunicazioni

Abstract: *The revolutionary Internet of Things paradigm successfully enabled the interaction among smart objects pervasively diffused across the Internet. From the beginning, communication technologies and open standards already provided baseline techniques able to offer security and privacy, also in constrained environments. But, while data confidentiality and data integrity were immediately tackled with robust cryptosystems, authentication and key management have been frequently covered with superficiality. Not only: fine-grained authorization and anomaly and intrusion detection services have been generally ignored in most cases. As a consequence, the worldwide scientific community dedicated a high attention to the design, the implementation, and the evaluation of new addressing these open issues, while taking care about the new requirements characterizing the emerging Internet of Things scenarios and the related security risks and threats. This contribution starts by providing a new set of Internet of Things-oriented definitions of security services and highlights the impact that their implementation may have on the behavior of standardized communication protocols (i.e., computational complexity in constrained environments, energy consumption, and so on). Then, it presents some interesting approaches emerged in the recent scientific literature, addressing the uncovered security services. In summary, they include: key management protocols based on Implicit Certificates and Blockchain, Attribute-Based Access Control in multi-domain and multi-authority ecosystems, anomaly and intrusion detection mechanisms based on machine learning.*

1 Introduction

The Internet of Things (IoT) is an emerging and promising technology which tends to revolutionize the global world through connected physical objects, that can be fruitfully employed to enhance the quality of everyday life. For this reason, IoT is massively present in several application scenarios, including health and environmental monitoring, home automation, smart mobility, industry applications, etc. [1]. The wide variety of public and private environments where IoT devices are employed, together with the evergrowing number of connected devices, unavoidably poses critical security concerns, in terms of privacy, authentication (AuthN) and recovery from attacks, that need to be effectively managed [2]. This challenging task is even more complicated by the spread of communication technologies and approaches proposed for IoT devices, ranging from low power wide area networks (LPWAN) solutions such as SigFox, LoRa, WiFi, Zigbee, Ingenu-RPMA, Weightless and NB-IoT, to 5G New Radio (NR) technology in enhanced Machine Type Communication (eMTC), Bluetooth Low Energy (BLE) and DASH7 [3, 4].

Nevertheless, while data confidentiality and data integrity were immediately tackled with robust cryptosystems, authentication and key management have been frequently covered with superficiality. Not only: fine-grained authorization and anomaly and intrusion detection services have been generally ignored in most cases. Accordingly, high attention must be paid to design, implementation and evaluation of new schemes that address these open issues, at the same time taking into account the new requirements characterizing the emerging IoT scenarios with their security risks and threats. The goal of this contribution is to provide a new set of IoT-oriented definitions of security services, pointing out the impact that their implementation may have on the behavior of standardized communication protocols. Then, some interesting approaches emerged in the recent scientific literature are presented, addressing the uncovered security services. They include key management protocols based on Implicit Certificates and Blockchain, Attribute-Based Access Control in multi-domain and multi-authority ecosystems, anomaly and intrusion detection mechanisms based on machine learning.

The reminder of this chapter is organized as follows. Section 2 provides an overview of employed technologies, focusing on the related security services. Sections 3 and 4 analyze the main authentication and access control schemes, respectively. Section 5 provides an overview of anomaly and intrusion detection schemes. Section 6 concludes this chapter.

2 Security in reference technologies for the IoT

IoT reference technologies can be classified according to the transmission range, depending on whether their communication range is short or medium-long. Based on this classification, each technology will be discussed from a security perspective in what follows.

2.1 Short-range technologies

Wi-Fi IEEE 802.11 is a standard for Wireless LAN (WLAN) (interoperable standard-compliant implementations are referred to as Wi-Fi). IEEE 802.11i [5] specifies security standards for IEEE 802.11 LANs. The original specifications include a set of security features for privacy and authentication that were quite weak. Specifically, the first algorithm was the Wired Equivalent Privacy (WEP) [6], that however contained major weaknesses. Subsequently, the 802.11i task group developed a set of capabilities to overcome the issues of WEP and the Wi-Fi Alliance promulgated the Wi-Fi Protected Access (WPA) [6] as a Wi-Fi standard. This is a set of security mechanisms that eliminated most of the security issues and was based on the original version of the 802.11i standard. The following versions of WPA, i.e. WPA2 and WPA3 have been proposed to solve the security issues of WPA, that no longer was able to provide sufficient security to protect consumers or enterprise Wi-Fi networks. The final form of the 802.11i standard is referred to as Robust Security Network (RSN), that defines different security services, i.e., authentication, access control and privacy with message integrity. There are two versions of WPA2, Personal and Enterprise; the main difference is the authentication phase, since the first uses pre-shared keys (PSK) and is designed for home use, while the second uses IEEE 802.1X to offer enterprise-grade authentication through the Extensible Authentication Protocol (EAP). WPA3 was released to overcome some WPA2 limitations; it also encompasses WPA3-Personal and WPA3-Enterprise versions. It makes it mandatory to use Protected Management Frames (PMF), optional for WPA2, and requires a longer encryption key in the enterprise version. Moreover, WPA3 introduces the Simultaneous Authentication of Equals (SAE) to replace WPA2's PSK exchange protocol. Furthermore, it leverages on a standardized mechanism (Certified Easy Connect) to simplify the provisioning and configuration of IoT devices.

ZigBee Security services for ZigBee include methods for key establishment, key transport, frame protection and device management [7]. The level of security provided by the ZigBee architecture strictly depends on the correct and safe management of the symmetric keys, on the protection mechanisms adopted, and on the proper implementation of the cryptographic mechanisms and the associated security policies. As a consequence, trust in the security architecture reduces to trust in the secure initialization and installation of keys and to trust in the secure management of keys. ZigBee exploits the Advanced Encryption Standard - Counter with Cipher Block Chaining - Message Authentication Code (AES-CCM), as stated in [8]. The offered security services are authentication, confidentiality and integrity, according to the chosen security level. The ZigBee architecture includes security mechanisms at two layers of the protocol stack: the network layer and the application support sublayer. They are responsible for the secure transport of their respective frames. The latter provides also services for the establishment and maintenance of security relationships. Finally, the ZigBee Device Object manages the security policies and configuration of a device.

Bluetooth Low Energy Bluetooth Low Energy (LE) is a wireless personal area network (WPAN) technology based on the IEEE 802.15.1 standard. According to [9], Bluetooth is susceptible to general wireless networking threats (e.g., DoS attacks, eavesdropping, man-in-the-middle, message modification, etc.) plus security vulnerabilities in Bluetooth implementations and specifications. The security services targeted by the Bluetooth specifications are: pairing/bonding, authentication, authorization and confidentiality, and message integrity. Pairing can be obtained by creating one or more shared secret keys and storing them for use in subsequent connections to form a trusted device pair. To this end, the AES algorithm is adopted. Authentication is obtained by verifying the identity of communicating devices based on their Bluetooth address. To this end, the AES-CCM algorithm is used [9]. The standard proposes (from version 2.1) 4 security modes the security procedures are initiated with, before or after physical and logical link setup. Authorization and confidentiality are obtained by allowing the control of resources by ensuring that a device is authorized to use a service before permitting it to do so. Finally, message integrity is obtained by preventing information compromise, ensuring that only the authorized devices can access and view transmitted data, and verifying that a message sent between two Bluetooth devices has not been altered in transit. The encryption algorithm used is the AES-CCM.

2.2 Medium-long range technologies

SigFox SigFox is a technology aiming to build wireless networks to connect low power objects in IoT scenarios. For its technology, security is highly important. The key aspect is that security comes with devices: between the connected devices and the SigFox cloud there is an end-to-end authentication method based on a secret key. The secret key is stored in a non-accessible memory associated with a visible and specific ID stored in a read only memory. This secret key is used by messages sent by devices to create a signature, that is unique for each message, and that will authenticate the sender. Furthermore, the SigFox base stations are connected with the SigFox cloud through a point-to-point link using an encrypted Virtual Private Network (VPN). At the chain end, the user platforms are connected to the SigFox cloud using HTTPS encrypted interfaces [10]. SigFox uses an authentication method that exploits a symmetrical authentication key [11].

LoRa Long Range (LoRa) is a low power wide area network (LPWAN) technology, developed by the LoRa Alliance [12]. LoRaWAN specifications define two layers of cryptography: a unique 128-bit Network Session Key shared between the end device and network server and a unique 128-bit Application Session Key (AppSKey) shared end-to-end at the application level. LoRaWAN security targets a full end-to-end encryption for IoT applications. The security framework has been designed to offer the following security services: mutual authentication, integrity protection and confidentiality. Mutual authentication is established between a LoRA end device and the network as a part of the network joining procedure. This ensures that only genuine and authorized devices will be joined to genuine authentic networks. To this end, a symmetrical authentication key is exploited [11]. Furthermore, messages are authenticated, protected and encrypted at both MAC and application layers, thus ensuring that network traffic cannot be altered and is coming from a legitimate device. Security in LoRaWAN is based on the AES cryptographic primitive.

Narrowband Internet of Things NarrowBand Internet of Things (NB-IoT) is a LPWAN technology standard developed by 3GPP to enable a wide range of cellular devices and services. The security requirements of NB-IoT are similar to those of traditional IoT, but there are many differences mainly due to the hardware equipment. The algorithms that should be adopted are data encryption, identity authentication and integrity verification. The most frequently used cryptographic mechanisms include random key pre-allocation, deterministic key pre-allocation and password based on identity [13]; furthermore, in this technology the authentication is mutual, i.e., the device authenticates to the network and vice versa [11].

Enhanced Machine Type Communication Enhanced Machine Type Communication (eMTC), also called as LTE Cat-M1, or CAT-M, is a LPWAN technology introduced by 3GPP with the Release-13 of the LTE standard. This technology is an enhancement for LTE networks to support MTC for the IoT. LTE systems provide security by integrating different security algorithms, such as mutual authentication, integrity and encryption [14]; however, in addition to security for communications, most of the e-MTC applications require security for data. Some solutions, as proposed in [14], are the utilization of new mechanisms to achieve a higher level of security (e.g., the operation of e-MTC applications could be restricted only to authenticated devices), the exploitation of physical-layer security adopting radio-frequency fingerprinting, and the use of asymmetric security schemes.

5G New Radio 5G New Radio (NR) is a radio access technology developed by 3GPP for the 5G mobile network. The 5G NR network is expected to cater for massive and critical IoT scenarios, as the demand for machine type communication is rapidly growing [15]. The most important security features are the primary authentication and the key agreement. The goal of these procedures is to enable mutual authentication between the UE and the network and provide keying material that can be used between the UE and the serving network in subsequent security procedures [16, 17].

Ingenu-RPMA Ingenu Random Phase Multiple Access (RPMA) technology is designed specifically and exclusively for wireless machine-to-machine communication [18]. This technology addresses six security services: message confidentiality, message integrity and replay protection, mutual authentication, device anonymity, authentic firmware upgrade, and secure multicast. In particular, the authentication scheme provides for mutual authentication [11].

DASH7 DASH7 is a long range low power wireless technology, and is an extension of active Radio-Frequency Identification (RFID), where communication can take place directly between devices and can be also used for non-RFID applications [15]. As stated in [19], the security in DASH7 is organized in the network layer and is similar to the security of IEEE 802.15.4 [20], using AES-Cypher Block Chaining (AES-CBC) for authentication and AES-CCM for authentication and encryption.

Weightless Weightless is a set of LPWAN open wireless technologies adopted in public or private networks with end devices with IoT requirements [21]. Weightless security aims to provide adequate protection for a variety of use cases. Its goal is to guarantee the authenticity, integrity and confidentiality of user and control data. AES-128/256 is the algorithm used for the encryption and authentication of both the terminal and the network, that guarantees integrity while temporary device identifiers offer anonymity for maximum security and privacy. Over-the-air (OTA) security key negotiation or replacement is possible, while a future-proof cipher negotiation scheme with a minimum key length of 128 bits protects long term investment in the network integrity. Weightless offers also mutual authentication with trusted third party.

The security services addressed by the IoT reference technologies are summarized in Table 1.

3 Authentication techniques and Key Management in IoT

Authentication is the process of determining whether someone or something is who or what it is declared to be. It is of great importance the implementation of distributed identity and trust management strategies in IoT scenarios, where a huge number of devices

TECHNOLOGY	SECURITY SERVICES	COVERAGE RANGE
SigFox	AuthN, Confidentiality and Integrity	Medium-long
LoRa	AuthN, Confidentiality and Integrity	Medium-long
Wi-Fi	AuthN, Confidentiality, Port-based Access Control and Integrity	Short
Zigbee	AuthN, Confidentiality, Access Control List and Integrity	Short
NB-IoT	AuthN, Confidentiality and Integrity	Medium-long
E-MTC	AuthN, Confidentiality and Integrity	Medium-long
5G New Radio	AuthN, Confidentiality and Integrity	Medium-long
Bluetooth LE	Pairing, AuthN, Service-level Authorization, Confidentiality and Integrity	Short
Ingenu-RPMA	AuthN, Confidentiality, Integrity, device anonimity, replay protection and authentic firmware upgrades	Medium-long
DASH-7	AuthN, Confidentiality and Integrity	Medium-long
Weightless	AuthN, Confidentiality and Integrity	Medium-long

Table 1: Security services offered by the reference IoT technologies

must be authenticated and carry out trusted communications. The classical authentication process consist of three steps: (i) obtaining the required authentication process (e.g., a password); (ii) analyzing data (e.g., compare the received password with the stored one); and (iii) determining if they are really associated with the original ones (e.g., confirm if they are the same). Authentication techniques must be accompanied by mechanisms of secure data exchange through cryptographic keys. Accordingly, a secure management of such keys is a major aspect in IoT environments, where scalability is a critical issue because of the potentially huge number of connected devices [22]. The different Key Management Protocols (KMPs) proposed in the recent literature aim to tackle this problem, by proposing strategies for secure key management (i.e., their generation, exchange, storage, distribution and replacement) and the related cryptographic and authentication schemes adopted.

This section presents a taxonomy of different IoT authentication schemes, that have been classified by the scientific literature into different criteria, including authentication factors, use of tokens, authentication architectures, and hardware-based schemes [23]. For what concerns KMPs, they are mostly based on asymmetric (or public key) cryptography, that is suitable for resource-constrained devices because it is scalable and with low computational complexity and overhead because of the small amount of data (the keys) to be encrypted. The main challenges in this context are mainly related to the optimization of the cryptography protocols to match the requirements of scalability, the adoption of hardware-based solutions to face the problem of the high computational effort peculiar of asymmetric cryptography, and the adoption of novel technologies, like blockchains, to ensure a high security level in IoT [22].

3.1 Authentication Factors

An authentication factor is a category of credentials that aims to verify that an entity involved in a communication or requesting access to a system is who, or what, it is declared to be. Identity-based authentication schemes can utilize one, more, or a combination of hash and symmetric or asymmetric cryptographic algorithms. Typical schemes don't fit well in constrained environments for IoT, thus new approaches are needed. Some works use identity-based public key cryptosystem and identity-based aggregate signature. Such kinds of approaches are often applied to vehicular communications [24], because of their security and privacy-preserving requirements, and in smart grid scenarios [25]. The cited studies develop smart grid authentication scheme (SGAS) and smart grid key management (SGKM) protocols, so that different kinds of attacks (e.g., brute force, replay, man-in-the-middle, etc.) can be prevented.

3.2 Use of Tokens

A token is a piece of data created by the authentication server to uniquely identify a user or a device [26]. An authentication scheme is called token-based if the authentication method uses a token. Token-based authentication schemes can be further distinguished into soft and hard token-based schemes. Representative examples of token-based approaches can be found in [27, 28]. In [27] the authentication protocol, is payload-based. It is able to detect the attacks based on the cluster formation between neighbouring nodes and their nearest cluster head in a Internet of Sensors (IoS) scenario. In [28] the token-based authentication is based on the use of nonces, that are resistant to different attacks (e.g., man-in-the-middle, replay, impersonation, etc.). This approach is used to guarantee mutual authentication in a wireless sensor network (WSN). On the other side, non-token based authentication involves the use of the credentials (username, password) each time there is a need to exchange data. An example of such approach is found in [29], where the authentication scheme is implemented on existing Internet standards, especially the Datagram Transport Layer Security (DTLS), and applied it in a machine-to-machine (M2M) scenario.

3.3 Authentication Architecture

The authentication architecture can be distributed, if it uses a distributed authentication method between the communicating parties, or centralized, if it uses a centralized server or a trusted third party to deliver and manage the credentials used for the authentication. In both cases, the authentication architecture can be hierarchical, where a multi-level architecture is used to handle the authentication procedure, or flat, where no hierachical structure is used to manage the authentication procedure. The proposal in [30] is an example of a distributed approach used for privacy-preserving authentication in Vehicular ad hoc networks (VANETs) for the Internet of Vehicles (IoV) use case. The proposed protocol is based on a technique based on multiple trusted authority and one-time identity-based aggregate signature (OTIBAS), and multiplicative secret sharing (MSS). This approach promises to be more practical, since it requires only realistic tamper-proof devices (TPD).

3.4 Key management protocols for authentication procedures

Let us consider a scenario in which two entities want to communicate with each other. Three different authentication procedures can take place: (i) *One-way Authentication*, if only one entity will authenticate itself to the other, while the other still remains unauthenticated; (ii) *Two-way Authentication*, if both the entities authenticate each other; and (iii) *Three-way Authentication*, if there is also a central authority that authenticates the two parties separately and then helps them to mutually authenticate themselves. One of the main concerns in authentication procedures is the management of the keys involved in the process. Several protocols and strategies, the so called KMPs, have been proposed in the recent literature to tackle this issue [22]. A first classification of them is performed on the basis of the methodology adopted for the secure delivery of the keys, that can be based on key transport mechanisms to securely transfer a secret key generated by one party to all the other parties, or it is derived by all the parties as a function of a common information. Another classification of KMPs can be done on the basis of the cryptographic primitive adopted by the protocol, that determines the way the keys are managed. Schemes tackling this issue are focused on the pre-distribution of a symmetric key which is shared among all the parties, or schemes based on an asymmetric key that follow the rules of Public Key Cryptography (PKC) and ensure confidentiality, authentication, non-repudiation and integrity, but require high computational effort [22]. One of the goals of authentication mechanisms is to bind the key(s) with the identity of a device. It is noteworthy that some protocols provide implicit authentication, some others not. The authentication mechanisms can adopt identity-based authentication, PKI, or certificate-based authentication [22].

Among the certificate-based authentication schemes, an interesting approach is presented in [31]. This study addresses the problem of public key authentication and key agreement by exploiting implicit certificates. The authors propose a KMP that integrates implicit certificates with a standard elliptic curve Diffie-Hellman exchange, and performs authentication and key derivation. This approach, when applied to sensor networks, guarantees maximal airtime savings and efficient protection against replay attacks. Another promising approach for key management in IoT scenarios is the use of blockchains, as testified by [32, 33]. In [32] the potential of the blockchain technology is discussed to provide IoT security. In [33] a key agreement methodology integrating the blockchain technology is proposed for the IoT scenario. Blockchain is used to store X.509 certificates related to the initial fixed public keys of devices, to publish new ephemeral public keys, and to help the verification of the authenticity of ephemeral public keys without sending signatures and additional X.509 certificates. In this way a low communication overhead, a limited energy consumption and acceptable latencies are guaranteed.

3.5 Hardware-based schemes

In some cases, the authentication process might require the use of physical characteristics of the hardware itself. Based on this concept, the authentication process can be:

- *Implicit hardware-based*: it uses the physical characteristics of the hardware to enhance the authentication such as Physical Unclonable Function (PUF) or True Random Number Generator (TRNG).

- *Explicit hardware-based*: it is based on the use of a Trusted Platform Module (TPM), that is a chip that stores and processes the keys used for hardware authentication.

Examples of hardware-based approaches are [29,34,35]. The hardware-based approach presented in [34] uses a hardware fingerprint to authenticate IoT devices with its PUFs. A software model of the PUF is also derived through Machine Learning based attacks on PUF. Other PUF-based algorithms for IoT devices are presented and implemented in [35]. They use PUF-based elliptic curve for device enrollment, authentication, decryption, and digital signature. In [29] the proposed authentication scheme is based on the RSA algorithm with the use of Trusted Platform Module (TPM).

A schematic representation of the different authentication approaches described above is summarized in Table 2.

Approach	**Use Case**	**Communication Technologies**	**Work**
Identity-based	Vehicular communications	not specified	[24]
	Smart Grid	not specified	[25]
Token-based	IoS	not specified	[27]
	WSN	not specified	[28]
Non-token based	M2M	not specified	[29]
Implicit Certificates	WSN	IEEE 802.15.4	[31]
Blockchain-based	generic IoT	Wi-Fi	[33]
Distributed	IoV	not specified	[30]
Hardware-based	generic IoT	ZigBee	[34]
		not specified	[35]
		not specified	[29]

Table 2: Approaches used in the scientific literature

4 Access control techniques in IoT

Traditional access control models have focused on closed systems where all users are known and primarily utilize a server-side reference authorization entity within the system. The complexity and the heterogeneity of an IoT environment collapse the entire authorization architecture exploited until now. Most of the IoT technologies do not implement fine-grained access control mechanisms directly into the standard. In fact, they rely on an on-off mechanism in which authentication and authorization are coupled together. Indeed, the models that will be described in the sequel are frequently implemented in the upper layers of the protocol stack. Recent works such as [36] highlight that the recent research trend is moving towards the design of approaches that embrace the decoupling between authorization and authentication, a fine-grained authorization, the protection against collusion attacks, a time-limited authorization, the protection of user privacy, the revocation access rights and the support of offline authorization. According to [37], existing approaches can be classified in Role-Based Access Control (RBAC), Attribute-

Based Access Control (ABAC), Usage CONtrol (UCON), Capability-Based Access Control (CapBAC), Organizational-Based Access Control (OrBAC), and other models. The following subsections provide an analysis of some innovative studies and models designed in the access control security field. Then a brief classification of some innovative access control schemes can be found in Table 3.

4.1 Role-Based Access Control (RBAC) models

RBAC relies on the pair role/permission according to which users inherit the permissions assigned to the roles they have. Roles are often structured hierarchically, so that the inheritance of permission between different roles is clearly defined. An explanation of RBAC models and concepts can be found in [38]. Specifically, a RBAC-based model has four principal components: users, roles, permissions and sessions. Firstly, each user receives one or more roles, that are named job functions within the organization, and that describe the authority and responsibility conferred on a member of the roles; this is called User Assignment relationship. Usually, each role can receive one or more permissions, and this is called as the Permission Assignment relationship. Finally, a user may establish one or more sessions in which he/she can activate a subset of roles he/she belongs to. Moreover, advanced models introduce features like Role Hierarchies and constraints. Using an RBAC approach a security administrator assigns the appropriate permissions for the role according to the characteristics and context of IoT devices, and specifies an appropriate range for a user according to his role. This kind of model allows sophisticated context-based services and a dynamic rights management [37]. An extension of traditional RBAC systems can be integrated in medical environments based on IEEE 802.15.4, where context-aware capabilities are exploited in a complete distributed access control architecture. In this approach a two-layer context-aware role-based access system is used. It encompasses a data layer that comprises all the information required for access control decisions and an engine layer that manages the access control decisions on the disclosure of medical data or on the access to the sensors of a Patient Area Network [39].

4.2 Attribute-Based Access Control (ABAC) models

ABAC is an access control technique where resource access is granted depending on the attributes of subjects, objects, actions and the environment related to a request. Policies and access requests are defined in terms of required and owned attribute. The access right to a resource is determined by comparing the attributes in the request with the attributes in the policy. ABAC models often provides constructs to combine policies authored by different stakeholders and mechanisms to solve conflicts that can arise from these policies [40]. More in detail, the logical actors involved in an ABAC model are: Attribute Authorities (AA), Policy Enforcement Point (PEP), Policy Decision Point (PDP), and Policy Administrator Point (PAP). AA is responsible for binding attributes to an entity of interest; PEP manages the authorization request and enforces the corresponding decisions; PDP is responsible for evaluating the applicable policies and making the authorization decision; PAP creates and manages the access control policies. ABAC can be integrated within larger frameworks in order to achieve mutual authentication between user and nodes (e.g. ECC-based authentication) and fine-grained access control in Wireless Sensor

Networks (WSNs) [37]. According to [39], ABAC is more suitable and used in environments encompassing constrained devices; in fact, ABAC can also be built in REST-based architectures of low powered devices and CoAP clients [37]. On the same line, an ABAC scheme can be implemented to address the lightweight, scalability, decentralization, policy distribution and management requirements in a REST-based environment [41]. Moreover, incorporating a semantic model (e.g an ontology) this approach enables new applications to be developed independently from the concrete environment so that the supported protocol suite can be amplified. Exploiting the benefits of the Software Defined Network (SDN) technology in Intelligent Transportation Systems (ITS), ABAC can enable a complete distributed approach for authentication and authorization purposes. This approach deploys PAP in the cloud, whereas the other components of the access control system are deployed in edge of the network [39]. ABAC is also suitable for the Fog Computing paradigm in IoT for healthcare systems, where all the access control components are spread through the cloud, fog and sensing layers. The main benefits of the distribution are to minimize latency and improve availability while preserving data security [42]. A multi-authority access control scheme for IoT can be suitable also for federated and cloud-assisted Cyber-Physical Systems [36]. The approach is based on the ABAC logic and realized through the decentralized multi-authority ciphertext policy attribute-based encryption algorithm. Several security issues can be found, like the decoupling between authentication and authorization, fine-grained, offline, and time-limited authorization, protection against collusion attacks, access rights revocation, and user privacy [42].

4.3 Usage CONtrol (UCON) models

Similarly to ABAC, UCON allows the definition of resource access policies through subjects and objects attributes and constraints regarding the environment. Moreover, it allows the specification of two additional properties that are mutability of attributes and continuity of decision [43]. More specifically, there are six principal components: subjects and their attributes, objects and their attributes, rights, authorizations, obligations and conditions. An authorization grants or denies an access based on the subject and object attributes, obligations are actions that have to be performed during an access and conditions are constraints linked to environment's state, which are uncorrelated to actors' attributes. Moreover, the most important difference between UCON and the other access control techniques in IoT is the possibility to specify the previous cited properties of mutability and continuity, i.e., object attributes can be modified as a result of an access, and an access policy can be enforced not only before the access but also during the period in which the access is performed. UCON in IoT can allow a cross-layer access control based on fuzzy theory that concerns in a network layer modeling, the approach gives the role of subject in the usage control to a service in the application layer, and defines the object as a device in the sensing layer [39]. Based on UCON integrated with Capability-Based Access Control, an hybrid access control model in IoT scenarios and cloud computing can be built, as shown in [44]. The architecture is built based on SDN and Network Function Virtualization (NFV); it allows to authorize legitimate IoT entities to access resources and to ensure better security of these networks. The proposed architecture integrates both authorization and authentication capabilities, so that authorized IoT devices and users can not only access the required data based on the level of trust, but they are authenticated as well.

4.4 Capability-Based Access Control (CapBAC) models

CapBAC models are based on a ticket, or token, held by some entity which grants certain permissions. Usually, it relies on an Access Control Matrix , which can be seen as an evolved Access Control List (ACL) [45]. More in detail, the CapBAC architecture elements are: a resource that will be accessed, an authorization capability that details the granted rights, a capability revocation able to inform the service in charge of managing the resource that specific capabilities have to be considered no more valid, a PDP that manages resource access request validation, a resource manager that is the service that checks the acceptability of the capability token received along to the resource request, and a revocation service that manages the capability revocations. The combination of ACL, CapBAC, RBAC, Organizational-Based Access Control (explained in the next subsection), ABAC and UCON produces an authorization scheme which is transparent, user friendly, fully decentralized, scalable, fault tolerant and compatible with a wide range of access control models that can also leverage on Blockchain-based architecture for IoT access authorizations [39]. For IoT enabled healthcare in a WiFi and Bluetooth Low Energy network the integration of RBAC, ABAC and CapBAC can provide a fine-grained and flexible authorization scheme. This approach is aimed to generate the least number of policies that need to be managed, in fact the parameterization of capabilities allows policy specifications to be applied to multiple user-thing relationships without more policy development, while the use of attributes provides a fine-grained control [46].

4.5 Organizational-Based Access Control (OrBAC) models

OrBAC has been developed in order to extend RBAC in the implementation of a more flexible approach and a new abstraction layer. Each security policy is defined for and by one or more organizations. Thus, the policies are completely parameterized to give the possibility of handling simultaneous several security policies associated with different organizations. Specifically, the most important entity is the Organization that can be seen as an organized group of active entities, i.e. subjects or entities playing a specific role. Moreover, a Subject is an active entity while the entity Object mainly covers inactive entities such as data files or records. Security policies specify the authorized access to inactive entities by active entities and regulate the actions carried out in the system by a set of permissions, prohibitions, obligations and recommendations. Each of these elements define a relationship between the main components of OrBAC that are Organization, Role (that is how organization is employing subject) Context, View (that refers to how Organization is using objects) and Activity (that specifies how organization is performing actions) [47]. An example of hybrid centralized-distributed approach is found in [39], where a cross-domain access control for IoT environment is designed. It relies on abstraction layers that distribute processing between constrained and less constrained devices, indeed the model is based on the partitioning of the access control process into functional layers depending on the capabilities offered to each one [39]. Deployed on ABAC and OrBAC is the Pervasive-Based Access Control that exploits the abstraction concept of OrBAC and the concept of attributes of ABAC [48]. It also introduces a collaborative layer that may rely on distributed structure based on blockchain. The result is a proactive, multi-layer model that enables the smart use of attributes and existing abstract entities [48].

Approach	Use case	Communication Technologies	Work
RBAC	Medical environment	IEEE 802.15.4	[39]
ABAC	generic IoT	not specified	[37, 41]
	Healtcare Systems	not specified	[42]
	Smart Home	ZigBee	[39]
	Intelligent Trasporation Systems	not specified	[39]
	Multi-authority federated platforms	not specified	[36]
UCON	generic IoT	not specified	[39, 44]
CapBAC	generic IoT	not specified	[39]
	Healthcare Systems	WiFi, BLE	[46]
OrBAC	generic IoT	not specified	[39, 48]
OAuth-based	generic IoT	IEEE 802.15.4	[39]
	generic IoT	not specified	[49]
UMA-based	IoT devices and Intelligent agents	not specified	[39]
Trust-Based	Environment Monitoring	not specified	[39]
SDN-based	Smart Home	not specified	[50]
Social-based	generic IoT	not specified	[39]

Table 3: Some innovative access control schemes in IoT

4.6 Other models

There are several studies about the integration and the extension of well-known access control schemes into IoT environments. Examples are Extensible Access Control Markup Language (XACML) and User-Managed Access (UMA), which derives from the Open Authorization (OAuth) 2.0 framework. These schemes are already employed in other scenarios, but they have also been extended for IoT environments [37]. The OAuth modeling into IoT is often based on the lightweight protocol utilization (e.g. MQTT 3.1 or CoAP) through authorization delegation to third party entity. The behaviour of this approach in IEEE 802.15.4 networks shows that this scheme can increase the energy consumption, but other issues, like memory footprint and dynamic configuration capabilities, make the implementation of this kind of framework preferable. Moreover, a scheme that tries to make access control policies independently from the nature of the IoT entities can allow an easy migration of the policy concept to communication among agents and a better management of different roles and permissions [39]. An UMA-based model can also provide a unified access control scheme for an heterogeneous hybrid architecture of IoT devices and intelligent agents [39]. A flexible authentication and authorization framework for IoT leveraging OAuth, different token formats, and the IoT protocol suite can be designed as testified in [49]. The core element in this approach is the gateway that is in charge of implementing OAuth authorization functionalities and collects data taking into account the limited capabilities of constrained devices. A solution based on cryptographic protection can achieve access control, but it creates overhead in terms of time and energy consumption. For this reason a Fuzzy approach of trust management can meet the IoT requirements. An energy efficient solution can be found by joining the Trust-Based Access Control paradigm with the notion of trust levels for identity management [39]. An access control in smart home environments based on REST architecture can also be realized based on existing social structures and ACL. In such approach, policy configurations are automatically assigned to users based on the social relationship between the home owner and the user, which is inferred based on a social network graph information or phone usage information [39]. Static and dynamic access control schemes based on the SDN framework can also be found to enhance the smart home IoT security [50]. This approach allows the manufacturers to enforce the least privileged policy for IoT, to reduce the risk associated with exposing IoT to the Internet and enables users to customize IoT access based on social and contextual needs (e.g. allowing only LAN access to the IoT through the user mobile terminal), to reduce the attack surface within the network.

5 Anomaly and intrusion detection techniques in IoT

Intrusion Detection System (IDS) in IoT is a very hot topic on which several research groups are working on; in fact, a huge amount of approaches and architectures exists on this topic. IDSs are mainly categorized into Host IDSs (HIDS), designed to be implemented on a single system, and Network IDSs (NIDSs) that gather information about incoming and outgoing traffic from the network, to protect a system from intrusions that could harm it. In an IoT environment NIDSs are seen as more suitable for effective anomaly detection. This is counterbalanced by the excess of resource consumption required, which is a disadvantage if seen on resource constrained IoT nodes [51]. Moreover, different detection techniques are developed, based on several attack types and network layers, but none of them appears to be comprehensive in terms of attack type detected, wireless technologies, and networking layers. In addition, due to the open and insecure physical layer, implementing an IDS on any IoT node itself can never guarantee its reliability. According to [51] the main goals common to the majority of IDSs developed for IoT are high performance in terms of true positive and false negative, a lightweight implementation and a real time anomaly and intrusion detection. Due to the enormous number of studies on this topic in the following sections an overview of innovative detection techniques is given, then an analysis of architectural features of IDSs is presented and finally a summary is presented in Table 4, that shows the researching trend of IDSs in IoT in terms of detection types, technologies and architectures.

5.1 Detection types

Different types of anomaly detection exist, each with pros and cons in terms of accuracy, detected attacks and feasibility on constrained IoT nodes. Due to heavy resource requirements of typical existing detection methods, that are not well suited to the constrained resources of IoT embedded devices, several adapted (but also novel) approaches have been proposed in literature [52]. Specifically, detection techniques can be classified in misuse, anomaly, specification and hybrid approaches.

5.1.1 Misuse detection techniques

Misuse detection techniques deal with monitoring activities such as network traffic or system-level actions that are compared to signatures within a database of malicious code patterns and intrusions, to detect well-known attacks. Misuse techniques are difficult to adapt in the IoT scenarios, due to the resource constraints of the involved devices that make infeasible the storage of an exhaustive database of signatures. This explains why these techniques are mostly seen within centralized architectures, which provide greater resilience against subversion, but could maintain an incomplete vision of network activity if the architecture is not well organized. Moreover, the ever increasing prominence of zero-day attacks in IoT networks nullifies their effectiveness. Several optimized pattern matching algorithms have been designed for IoT, often implemented through a remote NIDS [52, 53]. For Bluetooth and 6LoWPAN technologies auxiliary shifting method and the early decision scheme successfully reduce the workload by skipping a large number of unnecessary matching operations [52]. Moreover and IDS can be combined with other security modules such as an Intrusion Preventions System (IPS) or a Network Security Monitoring (NSM) in order to fit better in IoT environment. A framework capable of detecting Denial of Service (DoS) attacks in 6LoWPAN-based networks can also be developed [53].

5.1.2 Anomaly detection techniques

In anomaly-based intrusion detection systems, a normal data pattern is created based on data from normal users, and is then compared with current data patterns. Specifically, the current activity is compared against an activity model, and gaps between current activity and the model are classified as ambiguous, to provide an online way to detect anomalies. Anomaly-based detection methods are attractive in IoT due to their smaller memory footprint, and are proved to be more effective on typical constrained IoT protocols. An anomaly detection IDS can rely on several strategies, such as data mining, protocol, payload, signal processing or statistical models, to detect intrusions [54]. Anomaly detection techniques in IoT are often designed based on power usage of devices such as a system based on energy consumption and battery exhaustion monitoring in ZigBee network in which energy consumed by the nodes in normal behaviour is used to build a model. On the same line, smart hardware can be used to optimize the detection, for example the exploitation of smart batteries to create a model, but a tradeoff between the battery charge life and the device performance must always be found. Leveraging the trust management concerning the monitoring of activities in a network, a network trust based IDS for WSN can be developed. This approach builds a trust regulation table which is used as the baseline threshold for detecting several attacks [52]. Moreover, the monitoring of the number of packets shared between nodes in 6LoWPAN networks allows wormhole detection. In this scheme, the router is used for heavy processes, while the IoT nodes collect neighbor information. Another interesting lightweight approach is the feature selection by bit-pattern [53]. A direct representation of the feature space for the discrimination function can be used, where the detector can make a fast packet classification decision. Another interesting approach consists of a high performance, real time IDS that exploits Complex Event Processing (CEP), which has the benefit of an ability to identify complex patterns via real-time data processing [54].

5.1.3 Specification based techniques

This approach combines attributes of anomaly and misuse detection, and is advantageous because of a high accuracy, even if it introduces a delay in the creation of a signature due to the human interaction, which causes the process not to be timely and cannot be suitable for IoT [52]. The detection of anomalous activity derives from a pre-defined model, based on explicit policies that determine the normal behavior of the system. These security specifications are designed based on the applications and system-specific security parameters. Thus, operating patterns that are not included in the system normal behavior are considered as an anomaly or security violation. To detect sinkhole attacks and to automatically isolate malicious nodes, each node can act as a monitoring node to monitor the operation of all of its neighbors [53]. To this end, the network is divided in several clusters and data are periodically sent to the IDS agents [54]. A finite state machine can also be used to model the system state. Each IoT node detects intrusions from local trace and neighboring IDS agents cooperatively participate in global intrusion detection actions when an anomaly is detected.

5.1.4 Hybrid based techniques

IoT technologies are wide and varied, and the classification of detection techniques and technologies can be difficult. A possible path to be followed is the combination of some of the abovementioned techniques. Accordingly, different IDSs can be integrated with the aim to maximize the advantages and minimize the limitations of the single mechanisms [53–56]. A specification-based IDS in the IoT nodes can be combined with an anomaly-based IDS in the root node in 6LoWPAN network. The main feature of this system is the reduction in the number of communication messages due to the lack of additional control messages, and the applicability of this system to large-scale networks [54]. Designing manually a specific hybrid technique for each IoT application, protocol and technology is infeasible, so intelligent techniques focused on Machine Learning (ML) and Artificial Intelligence, specifically on Deep Learning (DL) algorithms, have been adopted to design optimal IDS for IoT, such as a Supervised Fuzzy C-Means IDS or a self-learning IDS using Autoencoders (AE), that automatically learns new attacks [53]. The huge number of ML algorithms can be combined into innovative technologies already integrated in IoT, such as an SDN-based IDS leveraging Restricted Boltzman machine (RBM) or Deep Neural Network (DNN), in which the IDS analyzes the traffic over the controller, or a Long-Short Term Memory (LSTM) integrated with a Recurrent Neural Network (RNN). Moreover, in the Industrial IoT scenario, a Deep Auto Encoder (DAE) can determine the hyper-parameters of a DNN [55]. Goals and challenges of this kind of IDS and a report of the most common databases adopted to test and validate these approaches can be found in [56].

5.2 Architecture types

A variety of IDS architectures exist. Therefore, it is necessary to evaluate the correct architectural implementation that can exploit better the detection capabilities provided by the approaches described in the previous sections. According to the classification presented in [52], an IDS can be:

Detection type	**Architecture**	**Communication Technologies**	**Work**
Misuse	Centralized	BLE	[52]
	Centralized	not specified	[53]
Anomaly	Centralized	ZigBee	[52]
	Distributed	BLE, WiFi	[52]
	Distributed	not specified	[52]
	Hierarchical	not specified	[53]
	Centralized	not specified	[53]
	Distributed	BLE	[54]
Specification	Hierarchical	not specified	[54]
	Distributed	not specified	[53]
Hybrid	Hierarchical	not specified	[54]
	Distributed	not specified	[53, 55]

Table 4: A summary of innovative IDSs in IoT environment

- **Centralized**: Centralized IDSs monitor an IoT system from a single location and process data on an external agent. This has the advantage of not imposing an overhead on the sensing nodes. Moreover these systems do not create additional points for subversion and allow for greater depth of processing. However, by moving the data analysis to an external device, they create a single possible point of failure [52].

- **Distributed**: In a distributed IDS, each physical object contributes to the IDS work and implements a part of intrusion detection routine. Moreover, these IDSs must meet the resource constraints of the IoT nodes [53]. A common approach is the implementation of an anomaly detection system employing a distributed architecture but in a watch-dog based manner, that is, a subset of the network nodes have to monitor the other nodes [52].

- **Hierarchical**: Hierachiacal architectures are IDSs in which some nodes have a greater responsibility for processing than others. Decentralized architectures are often grouped under hierarchical architectures. Such systems fit very well across large and heterogeneous IoT networks, although there are many issues which must be considered regarding the system complexity [52].

6 Conclusions

In this contribution the most representative Internet of Things-oriented definitions of security services have been presented, highlighting the impact of their implementation on the already standardized communication protocols. In addition, some interesting approaches, protocols and schemes peculiar of the IoT scenarios have been analyzed, addressing the uncovered security services and taking into account authentication, authorization, key management and intrusion detection mechanisms, to provide an overview of the security services that can be developed or adapted to different IoT-based architectures and multi-domain/multi-authority ecosystems.

References

[1] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, and N. Ghani, "Demystifying IoT Security: An Exhaustive Survey on IoT Vulnerabilities and a First Empirical Look on Internet-Scale IoT Exploitations," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2702–2733, thirdquarter 2019, doi: 10.1109.

[2] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures," *IEEE Access*, vol. 7, pp. 82 721–82 743, Jun. 2019, doi: 10.1109/ACCESS.2019.2924045.

[3] M. M. Alam, H. Malik, M. I. Khan, T. Pardy, A. Kuusik, and Y. Le Moullec, "A Survey on the Roles of Communication Technologies in IoT-Based Personalized Healthcare Applications," *IEEE Access*, vol. 6, pp. 36 611–36 631, Jul. 2018, doi: 10.1109/ACCESS.2018.2853148.

[4] L. Chettri and R. Bera, "A Comprehensive Survey on Internet of Things (IoT) Towards 5G Wireless Systems," *IEEE Internet of Things Journal*, pp. 1 – 18, Oct. 2019, doi: 10.1109/JIOT.2019.2948888.

[5] I. S. Association, "IEEE Standard for Information technology—Telecommunications and information exchange between systemsLocal and metropolitan area networks—Specific requirements," Nov. 2016, part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.

[6] A. H. Lashkari, M. Mansoor, and A. S. Danesh, "Wired equivalent privacy (WEP) versus Wi-Fi protected access (WPA)," *2009 International Conference on Signal Processing Systems*, pp. 445–449, 2009.

[7] Z. Alliance, "ZigBee Specification," Aug. 2015, zigBee Document 05-3474-21.

[8] W. R. L. Xueqi Fan, Fransisca Susan and S. Li, "Security Analysis of Zigbee," May 2017.

[9] J. Padgette, J. Bahr, M. Batra, M. Holtmann, R. Smithbey, L. Chen, and K. Scarfone, "Guide to Bluetooth Security," *NIST Special Publication 800-121*, May 2017.

[10] https://www.sigfox.com/. Last accessed: 8 Jan., 2019.

[11] J. P. S. Sundaram, W. Du, and Z. Zhao, "A Survey on LoRa Networking: Research Problems, Current Solutions and Open Issues," *IEEE Communications Surveys & Tutorials*, Oct. 2019, doi: 10.1109/COMST.2019.2949598.

[12] https://lora-alliance.org/. Last accessed: 8 Jan., 2019.

[13] M. Chen, Y. Miao, Y. Hao, and K. Hwang, "Narrow band internet of things," *IEEE Access*, vol. 5, pp. 20 557–20 577, Oct. 2017.

[14] H. Shariatmadari, R. Ratasuk, S. Iraji, A. Laya, T. Taleb, R. Jäntti, and A. Ghosh, “Machine-type communications: current status and future perspectives toward 5G systems,” *IEEE Communications Magazine*, pp. 10–17, Sep. 2015.

[15] G. A. Akpakwu, B. J. Silva, G. P. Hancke, and A. M. Abu-Mahfouz, “A Survey on 5G Networks for the Internet of Things: Communication Technologies and Challenges,” *IEEE Access*, vol. 6, pp. 3619–3647, 2018, doi: 10.1109/ACCESS.2017.2779844.

[16] https://www.3gpp.org/release-15. Last accessed: 8 Jan., 2019.

[17] ETSI, “5G; Security architecture and procedures for 5G System (3GPP TS 33.501 version 15.1.0 Release 15),” Jul. 2018.

[18] https://www.ingenu.com/. Last accessed: 8 Jan., 2019.

[19] M. Weyn, G. Ergeerts, R. Berkvens, B. Wojciechowski, and Y. Tabakov, “DASH7 alliance protocol 1.0: Low-power, mid-range sensor and actuator communication,” *2015 IEEE Conference on Standards for Communications and Networking (CSCN)*, pp. 54–59, Dec. 2015.

[20] N. Sastry and D. Wagner, “Security considerations for IEEE 802.15. 4 networks,” *Proceedings of the 3rd ACM workshop on Wireless security*, pp. 32–42, Oct. 2004.

[21] http://www.weightless.org/. Last accessed: 8 Jan., 2019.

[22] M. Malik, M. Dutta, and J. Granjal, “A Survey of Key Bootstrapping Protocols Based on Public Key Cryptography in the Internet of Things,” *IEEE Access*, vol. 7, pp. 27 443–27 464, 2019, doi: 10.1109/ACCESS.2019.2900957.

[23] M. El-hajj, M. Chamoun, A. Fadlallah, and A. Serhrouchni, “Taxonomy of authentication techniques in Internet of Things (IoT),” pp. 67–71, 2017, doi: 10.1109/SCORED.2017.8305419.

[24] L. Zhang, C. Hu, Q. Wu, J. Domingo-Ferrer, and B. Qin, “Privacy-preserving vehicular communication authentication with hierarchical aggregation and fast response,” *IEEE Transactions on Computers*, vol. 65, no. 8, pp. 2562–2574, Aug. 2015.

[25] H. Nicanfar, P. Jokar, and V. C. Leung, “Smart grid authentication and key management for unicast and multicast communications,” *2011 IEEE PES Innovative Smart Grid Technologies*, pp. 1–8, Nov. 2011.

[26] T. Nandy, M. Y. I. B. Idris, R. M. Noor, M. L. M. Kiah, L. S. Lun, N. B. A. Juma’at, I. Ahmedy, N. A. Ghani, and S. Bhattacharyya, “Review on Security of Internet of Things Authentication Mechanism,” *IEEE Access*, vol. 7, pp. 151 054–151 089, Oct. 2019, doi: 10.1109/ACCESS.2019.2947723.

[27] M. Jan, P. Nanda, M. Usman, and X. He, “PAWN: a payload-based mutual authentication scheme for wireless sensor networks,” *Concurrency and Computation: Practice and Experience*, vol. 29, no. 17, 2017.

[28] M. Turkanović, B. Brumen, and M. Hölbl, “A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion,” *Ad Hoc Networks*, vol. 20, pp. 96–112, Sep. 2014.

[29] T. Kothmayr, C. Schmitt, W. Hu, M. Brünig, and G. Carle, “A DTLS based end-to-end security architecture for the Internet of Things with two-way authentication,” *37th Annual IEEE Conference on Local Computer Networks-Workshops*, pp. 956–963, 2012.

[30] L. Zhang, Q. Wu, J. Domingo-Ferrer, B. Qin, and C. Hu, “Distributed aggregate privacy-preserving authentication in VANETs,” *Distributed aggregate privacy-preserving authentication in VANETs*, vol. 18, no. 3, pp. 512–526, Jul. 2016.

[31] S. Sciancalepore, G. Piro, G. Boggia, and G. Bianchi, “Public key authentication and key agreement in IoT devices with minimal airtime consumption,” *IEEE Embedded Systems Letters*, vol. 9, no. 1, pp. 1–4, Mar. 2016.

[32] S. M. Muzammal and R. K. Murugesan, “A study on secured authentication and authorization in internet of things: Potential of blockchain technology,” in *Advances in Cyber Security (ACeS 2019)*, vol. 1132, Jan. 2020, pp. 18–32.

[33] P. Tedeschi, G. Piro, and G. Boggia, “When Blockchain Makes Ephemeral Keys Authentic: a Novel Key Agreement Mechanism in the IoT World,” *2018 IEEE Globecom Workshops (GC Wkshps)*, pp. 1–6, 2018.

[34] D. Mukhopadhyay, “PUFs as promising tools for security in Internet of things,” *IEEE Design & Test*, vol. 33, no. 3, pp. 130–115, Jun. 2016.

[35] J. R. Wallrabenstein, “Practical and secure IoT device authentication using physical unclonable functions,” *2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud)*, pp. 99–106, Aug. 2016.

[36] S. Sciancalepore, G. Piro, D. Caldarola, G. Boggia, and G. Bianchi, “On the design of a decentralized and multiauthority access control scheme in federated and cloud-assisted cyber-physical systems,” *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 5190–5204, Dec. 2018, doi: 10.1109/JIOT.2018.2864300.

[37] L. Cruz-Piris, D. Rivera, I. Marsa-Maestre, E. D. L. Hoz, and J. R. Velasco, “ Access Control Mechanism for IoT Environments Based on Modelling Communication Procedures as Resources,” *Sensors*, vol. 18, no. 3, pp. 1–21, Mar. 2018, doi: 10.3390/s18030917.

[38] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, “Role-based access control models,” *Computer*, vol. 29, no. 2, pp. 38–47, Feb. 1996, doi: 10.1109/2.485845.

[39] S. Ravidas, A. Lekidis, F. Paci, and N. Zannone, “Access control in internet-of-things: A survey,” *Journal of Network and Computer Applications*, vol. 144, pp. 79 – 101, 2019, doi: 10.1016/j.jnca.2019.06.017.

[40] E. Yuan and J. Tong, "Attributed based access control (abac) for web services," in *IEEE International Conference on Web Services (ICWS'05)*, July 2005, p. 569, doi: 10.1109/ICWS.2005.25.

[41] G. Fedrecheski, L. C. C. De Biase, P. C. Calcina-Ccori, and M. K. Zuffo, "Attribute-based access control for the swarm with distributed policy management," *IEEE Transactions on Consumer Electronics*, vol. 65, no. 1, pp. 90–98, Feb. 2019, doi: 10.1109/TCE.2018.2883382.

[42] S. Alnefaie, A. Cherif, and S. Alshehri, "Towards a distributed access control model for iot in healthcare," in *2019 2nd International Conference on Computer Applications Information Security (ICCAIS)*, May 2019, pp. 1–6, doi:10.1109/CAIS.2019.8769462.

[43] X. Zhang, F. Parisi Presicce, R. Sandhu, and J. Park, "Formal model and policy specification of usage control," *ACM Trans. Inf. Syst. Secur.*, vol. 8, pp. 351–387, Nov. 2005, doi: 10.1145/1108906.1108908.

[44] A. Lohachab and Karambir, "Next generation computing: Enabling multilevel centralized access control using ucon and capbac model for securing iot networks," Feb. 2018, pp. 159–164, doi: 10.1109/IC3IoT.2018.8668191.

[45] S. Gusmeroli, S. Piccione, and D. Rotondi, "A capability-based security approach to manage access control in the internet of things," *Mathematical and Computer Modelling*, vol. 58, no. 5, pp. 1189 – 1205, 2013, doi : 10.1016/j.mcm.2013.02.006.

[46] S. Pal, M. Hitchens, V. Varadharajan, and T. Rabehaja, "On design of a fine-grained access control architecture for securing iot-enabled smart healthcare systems," Nov. 2017, pp. 432–441, doi :10.1145/3144457.3144485.

[47] A. Kalam, R. Baida, P. Balbiani, S. Benferhat, F. Cuppens, Y. Deswarte, A. Miege, C. Saurel, and G. Trouessin, "Organization based access control," July 2003, pp. 120 – 131, doi: 10.1109/POLICY.2003.1206966.

[48] S. El Bouanani, M. A. El Kiram, O. Achbarou, and A. Outchakoucht, "Pervasive-based access control model for iot environments," *IEEE Access*, vol. 7, pp. 54 575–54 585, 2019, doi: 10.1109/ACCESS.2019.2912975.

[49] S. Sciancalepore, G. Piro, D. Caldarola, G. Boggia, and G. Bianchi, "Oauth-iot: An access control framework for the internet of things based on open standards," July 2017, doi: 10.1109/ISCC.2017.8024606.

[50] M. Al-Shaboti, I. Welch, A. Chen, and M. A. Mahmood, "Towards secure smart home iot: Manufacturer and user network access control framework," in *2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)*, May 2018, pp. 892–899, doi: 10.1109/AINA.2018.00131.

[51] S. Hajiheidari, K. Wakil, M. Badri, and N. J. Navimipour, "Intrusion detection systems in the internet of things: A comprehensive investigation," *Computer Networks*, vol. 160, pp. 165 – 191, 2019, doi: 10.1016/j.comnet.2019.05.014.

[52] E. Benkhelifa, T. Welsh, and W. Hamouda, “A Critical Review of Practices and Challenges in Intrusion Detection Systems for IoT: Toward Universal and Resilient Systems,” *IEEE Communications Surveys Tutorials*, vol. 20, no. 4, pp. 3496 – 3509, Fourthquarter 2018, doi: 10.1109/COMST.2018.2844742.

[53] A. Tabassum, A. Erbad, and M. Guizani, “A survey on recent approaches in intrusion detection system in iots,” in *2019 15th International Wireless Communications Mobile Computing Conference (IWCMC)*, June 2019, pp. 1190–1197, doi: 10.1109/IWCMC.2019.8766455.

[54] M. F. Elrawy, A. Awad, and H. Hamed, “Intrusion detection systems for iot-based smart environments: a survey,” vol. 7, Dec. 2018, doi: 10.1186/s13677-018-0123-6.

[55] M. Ugurlu and I. Dogru, “A survey on deep learning based intrusion detection systems,” in *2019 4th International Conference on Computer Science and Engineering (UBMK)*, Sep. 2019, pp. 223–228, doi: 10.1109/UBMK.2019.8907206.

[56] K. Costa, J. Papa, C. Lisboa, R. Munoz, and V. Albuquerque, “Internet of things: A survey on machine learning-based intrusion detection approaches,” *Computer Networks*, vol. 151, Jan. 2019, doi: 10.1016/j.comnet.2019.01.023.

The Social Internet of Things: a Survey

Luigi Atzori*, Antonio Iera, Giacomo Morabito*****

*UdR CNIT of Cagliari, University of Cagliari
piazza d'Armi, 09123 Cagliari, Italy
email: l.atzori@diee.unica.it

**UdR CNIT of Cosenza, University of Calabria
DIMES- Via Pietro Bucci, 87036, Arcavacata di Rende (CS), Italy
email: antonio.iera@dimes.unical.it

***UdR CNIT of Catania, University of Catania
DIEEI - V.le A. Doria 6, 95125 Catania, Italy
email: giacomo.morabito@unict.it

Abstract: *The Social Internet of Things represents a distinctive paradigm for the realization of IoT solutions which relies on the integration of social networking technologies with machine-to-machine communications. Accordingly, objects are capable of interacting in a social way with external services to improve the efficiency and the trust in the communications. As a result, a social network involving objects, services and humans is created, which supports the social network nodes communications and the deployment of IoT applications. Whereas these are the major features, not all the research groups that work in this area converge on the same vision. In this article, we discuss the different views to highlight the common features and understand the rationales behind each view. We then present the major technologies involved in the implementation of this paradigm and the applications that are enabled. The paper concludes discussing the current challenges and the future directions.*

1. Introduction

The number of objects that are reachable over the Internet is now close to 10 billion and this number is increasing very fast [1]. These devices produce a huge amount of data and provide a remarkable number of services which need to be mashed up and interconnected to extract the real value for the benefit of the society. This can be achieved through centralized approaches, where objects belonging to each platform are connected and managed by a centralized component that takes care of blending the data coming from different objects to extract the useful information. The different platforms can then be interconnected to avoid the formation of the often criticized *silos effect* of the Intranets of Things. Still, the control of interactions and information flows will be on the hands of the central components of each platform, which will decide what can get out of each realm and how it can be shared with the external world.

Partially in contrast with this approach, during the last decade, the communities of researchers working in the areas of *social networks* and *Internet of Things* have been studying a different strategy: *exploiting the potentialities of social networking technologies to develop a decentralized approach to foster the interactions among objects that belong to communities of trillions of members*. Indeed, the use of the social network technologies may bring to a different vision: the one where the objects have the possibility to interact with other mates in an autonomous way and opportunistically on the basis of the applications needs and mimicking the socialization activities of the humans. Expected advantages are the improvements in terms of navigability, scalability and the possibility to implement advanced strategies for the evaluation of the trust among the peers in the social network of objects. These advantages have also been demonstrated in real deployments for several application fields, such as; transportation, energy management and ehealth. Whereas many researchers have found this approach intriguing and full of potential benefits, not all agree on the same definition and different communities exploit different features. To shed the lights on this fast evolving paradigm of objects interaction in the IoT, this chapter intends to provide a review of the major works with particular reference to the different visions that have been developed in the past and the key implementation of this paradigm. Accordingly, the chapter is organized as follows. Section 2 is devoted to the analysis of the evolving definitions and visions of the Social Internet of Things. Section 3 analyses the major technologies that have been exploited over time to develop the different solutions in the field. Section 4 presents the different implementations and platforms that have been deployed. Final conclusions are drawn in the final section 5.

2. Definitions/Taxonomy, and evolutions

The paradigm of *social networks among IoT objects* can be declined in different ways, based on factors such as the participation (or the independence from them) of people in the network of objects, the nature of the objects participating in the social network, the finalization to a specific IoT service.

In the literature there are several definitions, sometimes discordant from each other, of “a social network of objects”, intended as “an ecosystem that allows people and smart devices to interact within a social framework” [2], or “a novel paradigm of social network of intelligent objects, based on the notion of social relationships among objects” [3], [4], or as a paradigm “where things become the social entities rather than their owners, which aligns very well with the vision of smart cities” [5]. Some others may be found.

To clarify the meaning of a social network of IoT objects, we can start from what was one of the first needs felt to make the concept of Social Networks converge with that of Internet of Things, which is due to researchers at Ericsson User Experience Lab. They started from the idea that the main issue with the Internet of Things is not to understand what is IoT from the technological point of view but to understand how it works (complex dynamics to exchange data, interaction with the services offered by objects hidden everywhere, control of our private sphere, etc..). They showed in [6] that the mental model associated to a network is something like “very many point-to-point connections” and concluded that the cited old mental model related to cables is only sufficient to understand the technological side of the Internet of Things. Thus, to avoid confusions (and refusal) and to make interacting users feel comfortable with the novel IoT paradigm, it is interesting to resort to a mental model that man has developed for thousands of years: social behavior and social relations that are the basis of other ecosystems and other types of networks (not technological but human ... for example). Ericsson researchers thought that a solution to both the practical scalability issues and the mental model/pedagogical issue could have been to simply “dress” a network of things as if it was a social network! This is achieved by envisaging a social networks of objects with relevant services that allows them to interact, to express in a natural way which data they need and which services they offer to users, and to collaborate with each other to create collective services to offer to the users. The resulting concept is referred to as ***Social Web of Objects***.

Main peculiarities of this concept can be summarized as follows: (i) online social networks and their APIs (Application Programming Interfaces) are used to allow smart objects to communicate with users by relying on Web protocols [7], (ii) users’ social network accounts support service operations for IoT, such as using location data or publishing device status [8], (iii) social networks are used as an interface to control smart objects [9]; collaboration between social networks and smart objects enable smart devices to “talk” with other objects, to share experience about certain situations and to seek for help [10], (iv) people share services offered by smart objects with friends/objects [11]. Actually, all these features are enabled by relying on Web technologies. Differently, the basic idea deriving from the researches in [3][4] is the definition of a “social network of intelligent objects”, named ***Social Internet of Things*** (***SIoT***). In analogy with Social Networks Services (SNS) for human beings, the novel paradigm introduces the concept of social relationships among objects. Advantages are the possibility to: (i) give the IoT a structure that can be shaped as required to guarantee network navigability, so as that object and service discovery is effectively performed and scalability is guaranteed like in human social networks, (ii) extend the use of models developed for social networks to analyze IoT related issues (intrinsically related to extensive networks of interconnected objects).

This paradigm differs from the previous one for several reasons. It focuses on the establishment and exploitation of social relationships among things, not among their owners. The owner mediation can be foreseen but objects have to play the key role, as it happens in novel IoT-based applications. Through social relationships things can crawl the IoT and discover services and resources; this provides a distributed solution that is expected to be effective, efficient and, most important, relieves the humans from doing it. The envisioned IoT architecture is not a mere service platform centered on the concept of web of things, yet a real platform for Social Network Services with suitable components introduced to cope with the presence of objects instead of human beings.

The latter paradigm has undoubtedly played the role of game changer and since its introduction to date it has given and is giving rise to different declinations in different research contexts and different market verticals; however, it has always been based on the building blocks originally identified (such as, the types of friendship between objects, the policies for their triggering, etc.).

A prominent example is represented by the variation of SIoT in the field of automotive applications and systems, designed for their use in the smart cities and having as reference objects the devices involved in so called Vehicular ad-hoc Networks (cars, roadside units, drivers' phones, etc.).

The ***Social Internet of Vehicle*** (***SIoV***) system introduced in [5] leverages existing VANETs technologies such as vehicle-to-vehicle, vehicle-to-infrastructure and vehicle-to-internet communications and presents a vehicular social network platform. The SIoV system uses social relationships among physical components to encourage different types of communications and stores the information (e.g., safety, efficiency, and infotainment messages) as a social graph that provides near real time or offline use cases for the intelligent transport systems (ITS). The near real time applications offer safe and efficient travel of the vehicle users, and the offline data ensures smart behavior of the vehicles or big data analysis for the transportation authorities.

In a different vertical market (Industry 4.0) the SIoT notion has also been exploited with different objectives. In [12], the rise of data collection and analytics in SIoT has naturally progressed to discussions in the context of industrial plants and has brought to the definition of the ***Social Internet of Industrial Things*** **(*SIoIT*)**. In SIoIT, intelligent machines with social properties, namely, Social Assets, share status information and cooperate via a social network to achieve a common goal: optimal system level performance. Starting from the cited paper, in [13] the role of the Social Operator 4.0 is explored in the context of smart and social factory environments, where humans, machines and software systems will cooperate (socialise) in real-time to support manufacturing and services operations.

A further promising evolution of the SIoT is used in the area of support to elderly people who may need permanent medical monitoring and constant assistance to conduct daily activities. In [14], the paradigm of Social Internet of **Things (SIoT) is recognized as an enabler of a framework for *E-Health systems*** based on IoT, which can offer efficient and effective services to people in the need to receive constant care.

Finally, the great transversality of the SIoT paradigm is recently evidenced by works, such as the one in [15], in which an agent-based approach that leverages Edge Computing and Social Internet of Things paradigms in order to address topics of scalability and interoperability together with discovery and reputation assessment of services and objects with reference to ***Large-scale Smart Environments*** development.

The most recent stage of evolution of the Social IoT is represented by the possibility of defining, monitoring and exploiting social relations and interactions between the Virtual representation of physical IoT objects in virtualized platforms, which has brought to the concept of ***Social Internet of Virtual Objects***. As one of the early examples, in [16] a framework is presented, for the decentralized and autonomous management of Things based on service-, interaction-, location- and reputation-oriented principles, that supports real-virtual world integration by representing Things and groups of Things of the real world via their counterparts in the Cyberworld: Virtual

Entities (VEs). By following the Social Internet of Things (SIoT) paradigm, the platform defines, monitors and exploits social relations and interactions between the VEs, and uses technologies and services from the domain of the social media. This trend has continued and, currently, several examples of solutions based on the concept of connecting virtual images of IoT objects through social bonds are available. Their aim is, among others, to exploit social graphs to carry out a more efficient discovery of the services offered by Things, to deal with interoperability problems, to implement effective and safe group communication mechanisms between IoT devices. As an example, in [17] it is proposed to exploit the Social Internet of Things (SIoT) concept, coupled with Identifier/Locator Splitting (ILS) techniques, to address the issues of ensuring scalability, session continuity, and mobility across heterogeneous multi-access Internet of Things (IoT) networks. More specifically, a scheme is introduced that browses the social graph of devices to find information about the locator of the intended destination(s) and it has been demonstrated that it outperforms the alternative ILS approaches in terms of many performance metrics, at the cost of just a slight increase in the storage demands to track social relationships.

Into the same research line falls the work in [18], wherein the authors leverage social networking technologies and concepts towards the definition of a game-changing network primitive for the future Internet of Things (IoT), called Sociocast, which enables trusted group-oriented communications, in-network publish&subscribe mechanisms, dynamic and selective firewalling, flexible data casting.

These latest works are early examples of the great potential that the paradigm in discussion can have in the Future Internet and in next-to-come 5G platforms.

3. Major technologies

Different technologies have been exploited in the design and implementation of SIoT solutions. In the following, we describe how the major ones have been utilized and summarize the major features in Table I.

Table I. Major technologies exploited for the design and implementation of the SIoT solution and main benefits.

Virtualization technologies	
Social virtual objects	These are used to implement the social virtual objects in the cloud and network infrastructure so as to augment the physical devices with additional storage and computation capabilities which are key for the social interaction.
Fog technologies	
Task offloading	It allows for social clone offloading, both vertically (from physical device to fog) and horizontally (fog nodes share their resources by dynamically performing inter-fog task offloading).
Semantic Web	
Ontologies to describe social objects	It fosters the automatization of key functionalities of the SIoT, such as the creation of new kinds of relationships within the network of users and objects and the search of services and information among the nodes of the social network.
Wireless communications	
Detection of nearby devices	Physical devices scan the radio channels to look for other devices that transmit beacons (e.g., WiFi Access Points, Bluetooth beacons) to identify nearby devices and trigger the establishment of some relationships if relevant conditions are met.

Virtualization technologies are largely employed in the IoT field as they allow for empowering the physical objects with additional (and necessary nowadays) capabilities that are not provided by the devices themselves. This results in the increase of: the set of protocols the object is capable of understanding, the data storage capabilities, the data processing rate, and the availability of the relevant services. Object virtualization is typically implemented in the cloud, which may allocate in a flexible way all the resources that are needed to support the devices management procedures and the deployed IoT applications. When applied to SIoT, these technologies are exploited to implement the social capabilities in the virtual objects: continuous check whether new mates should enter the circle of object friends, discovery of service and information needed by the applications, evaluation of the trust level of the friends [19].

Other than the cloud, **fog technologies** are more and more used to support SIoT deployments as they allow to reduce the response times even if the amount of resources are typically lower than in the cloud environment. [20] is a major research work in this direction, which analyzes all the benefits of the fog towards the deployment of the SIoT paradigm. According to this study, each social physical device has a clone in the fog which is indeed the node with social capabilities. Obviously, this is similar to the cloud implementation, where clones are most often called social virtual objects. The key role of the fog is to make it easier to move the SIoT social network from the IoT physical layer up to the virtual overlay infrastructure. For this purpose, each physical thing is mapped onto a software virtual clone that acts as a virtual processor and exploits the resources of the hosting fog server to execute tasks on behalf of the cloned thing. Container-based technology is emerging as the most suitable means allowing for dense virtualization of physical things [21]. According to the proposed solution, a virtualized physical server running at a fog node is composed of a number of clones of containers (each container acts as a lightweight virtual clone for the associated physical thing), a container engine that dynamically multiplexes the resources of the Fog server over the set of hosted containers, and a host operating system that is shared by all hosted containers. Each clone implements the social capabilities of the IoT nodes. Essentially, it is composed of the data that describe the entity (included its friends) to support two basic functions, namely, Relationship Management and Navigation of the corresponding social network. These are two major SIoT functionalities directly implemented in the fog nodes. Accordingly, below is a list of key features of the fog computing technologies that are exploited :

- Container-based virtualization: virtual clones of the served physical things are dynamically instantiated onto fog servers as lightweight containers.

- Vertical task offloading: thing devices may offload their tasks to the serving fog nodes. This provides thing augmentation on an on-demand basis through dynamic multiplexing of the Fog resources.
- Horizontal task offloading: fog nodes share their resources by dynamically performing inter-fog task offloading.

Semantic web technologies have been used in the SIoT to describe objects' features and functionalities and to foster objects' interactions. In this context, Resource Description Framework (RDF), as well as Web Ontology Language (OWL), are the major tools used to explain the data or services coming from devices and physical agents. Relying on these technologies, in [22] the authors have introduced the Social Smart Object's Relationships Ontology (SSOR-Ont) as the one characterizing the major current SIoT implementations. It represents the types of social relationships which characterize the SIoT, where the main entity is the SocialFriend, which is of type Object. It has several attributes to characterize its major features: ownership, which friends it is linked to, which types of friends it is linked to, location, reputation, and others. Then, the authors propose their own ontology, the SONS-Ont, which describes the needs of a smart agent (for instance in terms of consuming services), the services it produces, and the type of the agent (which is the device agent). Accordingly, in contrast to SSOR-Ont, the main entity of SONS-Ont is the Agent. Its type can be one of three classes: DeviceAgent, UserAgent and TaskAgent, which are the agents of the classes Object, User and Application respectively.

The defined ontology together with semantic rules are exploited to lead to cognitive friendship and goal management. Accordingly, a number of semantic rules are generated, which are typically presented in SWRL (Semantic Web Rule Language) format to automate the behavior of smart objects in several scenarios. Semantic rules defining the smart entities goals could be entered by the application owner or the user, and could be used to exploit the knowledge derived by machine learning techniques in order to automate the process of subscribing to a topic or adding, deleting or making a friend request. On the basis of the defined rules the entity may: (i) delete the least important link that it is connected to, if the number of neighbors is greater than a threshold, (ii) make a friend request, if the importance value of a non-neighbor node is greater than a neighbor node (iii) add an object that has made a friend request to it (if it is more importance than existing friends).

Other works have proposed other ontologies and the use of semantic technologies to support the implementation of key functionalities of the SIoT, such as the creation of new kinds of relationships among the network of users and objects that can be discovered through inference engines and the analysis of the consistency of the association between the components of the social network ([23], [24]).

Wireless communication technologies are central in the implementation of any IoT solution and this is clearly the case also for the SIoT. In this context, other than enabling the communication of the physical devices with the Internet, these technologies are also exploited in the relationship management module to detect the mobility patterns of the objects and to understand whether relationships between nearby devices should be created. In [25] and [26], the authors present algorithms which rely on the presence of anchor points to make individual devices understand they are close to each other and then trigger the relationship management action. The algorithm can be implemented by devices able to scan a radio channel to look for other devices that transmit beacons, such as: WiFi Access Points, Bluetooth beacons for indoor navigation or for advertisement, mobile devices in hotspot mode. The basic approach behind this algorithm is the use of one of these devices as reference point. The objective is then to detect the situations where two devices are under the same radio coverage of the same reference point. Parameters such as, power threshold to detect the presence of anchor point, frequency of scanning and length of visibility are important for the performance of the detection algorithm as evaluated in [25].

4. Reference implementations and platforms

In this section we provide an overview on the SIoT platforms implemented so far. More specifically, we can observe that throughout the years several platforms have been developed to realize the SIoT concept. More specifically, we can distinguish three major evolution phases:

1. Implementation of applications exploiting existing on-line social network platforms to share data and services among *friends*
2. Extension of the semantic supported by existing IoT platforms to take the social dimension of objects into account
3. Implementation of platforms managing virtualizations of IoT objects enriched with social features

Phase 1: Online Social Networks-based SIoT platforms

The most obvious way of implementing the SIoT basic idea is to allow users to share the data produced by their things via existing online social network services such as Facebook and Twitter. The idea is, therefore, to exploit the

APIs offered by such services to allow a user to post the data produced by her devices into her profile.

Solutions implementing such an approach have not been necessarily labeled as Social Internet of Things. However, they implement the idea of including smart devices into the social domain. Early examples of this approach include the smart home environment proposed by Kamilaris et al. in [27], wherein the smart devices monitoring and controlling the environment establish a community, centered around their owner, through Facebook.

In the same group of SIoT solutions, the one developed by the company Lifely[1] can be included. Lifely's solution envisions smart devices representing plants, interacting with their owners and their friends through Facebook and Twitter.

The same category includes all the solutions that post summary of fitness performance gathered by smart wearables or gym machines in some social network decided by their owner, e.g., MotoBody by Motorola, Google Fit, etc.

These solutions have the major advantages that they can be implemented very easily and can exploit existing and well established online social network services.

However, they have several drawbacks as well. The most sensitive is most probably related to privacy. A recent article published by the Wall Street Journal[2] reports that health data monitored by smart wearables and shared through social networks represents a serious threat to personal privacy.

A further major problem of such an approach is that existing online social network services are built around humans and therefore, enable the interactions between humans and things only, i.e., there is scarce support for thing-to-thing interactions.

With reference to the key technologies described in the previous section, virtualization is frequently used in these platforms even if the resulting virtual objects do not have a completely autonomous social behavior in the cloud. However, most of them are supported by the virtual counterpart to improve the processing of the data generated by the sensors and to search for the needed services.

[1] http://www.lifely.cc

[2] https://www.wsj.com/articles/your-health-data-isnt-as-safe-as-you-think-11574418606

Phase 2: Adding the social dimension to IoT data collection platforms

In the transition from Phase 1 to Phase 2 we can position the Social Web of Things developed by Ericsson. In such an approach, objects interact with each other by exploiting a specifically developed online social networking service. Therefore, the two major drawbacks identified for Phase 1 solutions can be solved while new problems arise. The major of these problems is that the proposed solution mostly supports interactions between objects belonging to the same person or organization.

To address this issue, Phase 2 solutions instead exploit IoT data collection platforms like ThingSpeak[3] and semantically give a social meaning to some data. Early example of solutions following this approach is the Social Internet of Things platform available at *http://social-iot.org*. In this case, a server has been deployed in which registered objects are assigned one or several *channels.* For each channel a *read key* and a *write key* are defined. Possession of the *write key* is necessary to write in the channel, whereas possession of the *read key* is necessary for reading in the channel. Therefore,

- In case the thing is a sensor: the write key is owned by the sensor itself whereas the read key is available to all the elements that are allowed to read the values measured.
- In case the thing is an actuator: the write key is owned by the elements that have the permissions to control the actuator whereas the reading key will be available at the actuator itself that periodically read in the channel to identify whether a new action must be executed.

More complex cases can be easily implemented in order to enable operations such as the configuration of the sensor settings or the reading of the current actuator status.

The *channel* can also be used to store information useful to establish social relationships between objects. More specifically, it is assumed that an external service reads all relevant data and, based on a given logic, decides whether new relationships should be created or existing ones should be deleted. Such a service will exploit the write key to update the resulting information in the appropriate channel. In *http://social-iot.org* the source code is available for Android SIoT clients.

The solution named iSapiens aims at realizing smart environment and exploits the above SIoT platform to include new devices [28].

Likewise *Socialite* is a solution which can be included in this category [29]. Major feature of Socialite is the attempt to exploit semantic technologies to

[3] https://thingspeak.com

implement a comprehensive representation of the SIoT entities and relationships.

One problem of this approach is that the platform stores and manages all data concerning the social profile of the object. Therefore, the user must disclose her sensitive information, thus being exposed to serious privacy threats.

Another disadvantage is that the proposed approach is centralized and scalability issues are likely to arise at IoT scales.

Other than virtualization technologies, these platforms have mostly exploited semantic technologies to describe capabilities and performed activities of objects so as to trigger the creation of social links among them.

Phase 3: Virtualized social objects platforms

The transition to solutions belonging to phase 3 has been mostly driven by increasing popularity of virtualization techniques. The first relevant example of such solutions is Lysis [30]. The architecture of Lysis is represented in the following Figure 1.

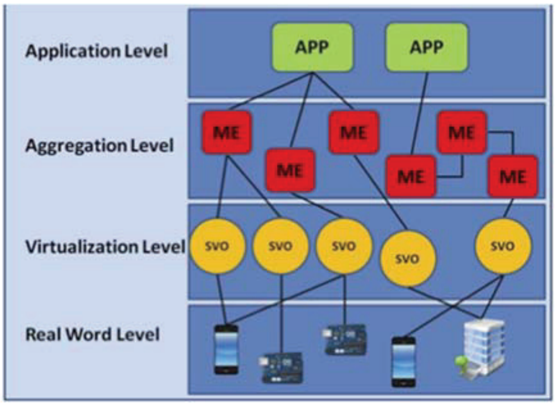


Figure 1. The Lysis architecture

Lysis envisions four levels:

1. The *Real World Level* includes the physical objects that will be included in SIoT.
2. The *Virtualization Level* includes the virtualization of the real world objects enriched with social features, such elements are called *Social Virtual Objects* (SVO)s and run as a *software agent* in the cloud. The SVO is instantiated starting by a *template* which is the same for all types of real world objects. SVOs are enriched with social features

and have two APIs one towards the lower level for communication with the real world objects and the other towards the higher level. At this level, Lysis implements the Relationship Management functions as well.

3. The Aggregation Level implements part of the application logic and is responsible for composing different SVOs to set up entities with augmented functions called *micro engines* (ME).
4. The Application Level finally implements several applications.

Observe that in the such types of solutions social objects can implement any logic and therefore are much more powerful than the *channels* envisioned in the solutions of Phase 2. The use of Lysis has been proposed for monitoring the coast [31].

Observe that Lysis has been designed to run in the cloud and therefore, even if it can programmed and managed as a centralized solution, it can instead be executed on several servers.

Nevertheless, the Lysis platform is managed by a unique organization which might be an obstacle to the deployment of a unique SIoT. To address such a problem, a SIoT platform realized according to the peer-to-peer paradigm has been recently developed [32].

The logical architecture of this solution is similar to the one of Lysis but it is implemented according to the Apache Ignite framework that supports peer-to-peer deployment as shown in Figure 2.

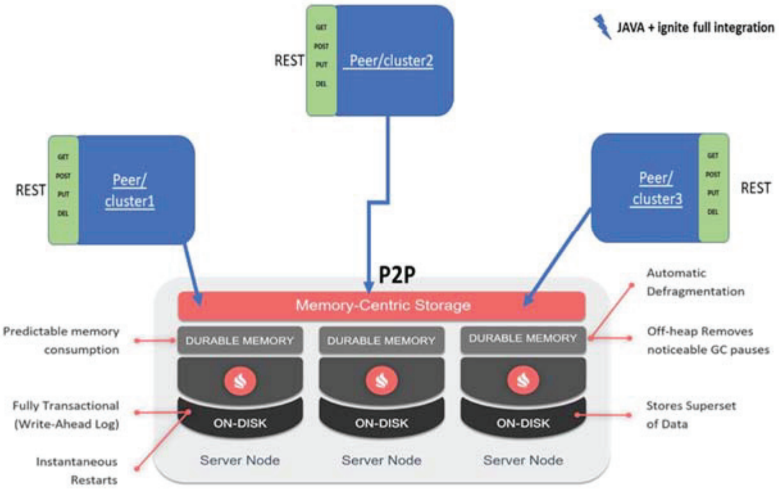


Figure 2. SIoT peer-to-peer implementation on top of Apache Ignite

This design choice allows to improve the performance of IoT solutions implementing the identifier/location splitting approach [ACD18] and is being used for logistics and postal operations within the H2020 project COG-LO[4]. In these implementations, all the key technologies mentioned in section 3 have been exploited. In Lysis and its implementation within the Apache Ignite platform, the cloud and the fog resources are jointly exploited to allocate the needed resources closer to physical devices. Additionally, semantic Web technologies are still used by the relationship manager to activate and update friendships. Additionally, wireless communications technologies are used to detect the events when physical devices are nearby. Indeed, both Bluetooth and WiFi beacons heard by two devices at the same time trigger the creation of a new relationship (or an update) if the needed criteria are fulfilled.

[4] http://www.cog-lo.eu

5. Current challenges and future directions

The communities that conduct research activities in the mentioned fields have still to face several challenges towards the full deployment and exploitation of the SIoT; some of them are discussed in the following.

A big obstacle is represented by the common issue of the *mass adoption* of these technologies that is required to test the proposed solutions and evaluate their benefits, which in turn may push towards its adoption. In this closed loop, the only approach that could be followed is to leverage on applications with a widespread adoption from the community of user, for instance applied to the transport sector with different objects involved, such as cars, smartphones, traffic signals, road panels, coil. By embedding the SIoT functionalities in these applications, it would be still possible to collect data on the possible objects activities and relevant relationship and perform an evaluation of the benefits in terms of navigability and trust management. Up to know this objective has not been achieved apart from small weak attempts. Accordingly, this represents an area where the Social IoT community should put some efforts towards further advancements.

As it has been discussed in section 4, virtualization technologies are characterizing major advancements in the social IoT, bringing to a cyber-physical world where every object can be found, activated, probed, interconnected, and updated at both the virtual and the physical levels. This environment allows for the implementation of the socialization functionalities to make any component interact with each other in an easy way. A component that is missing in this environment is the user, with her profile, attitude, wishes and activities carried out in both virtual and physical worlds. As for the objects, the *virtual users* should represent the virtual counterpart of the IoT users with the objective of simplifying the relationship of the users with the digital world, at least for those activities where an AI-enabled user clone can reliably substitute us (e.g., configuring and updating IoT services and objects, acting on behalf of the user when basic operations are required). Additionally, the virtual user should help in best exploiting the IoT potentialities, taking always into account the user profile and interests. It should be social when interacting with the other peers and the social objects. This vision introduces several challenges, which are mostly related to the definition of the functionalities, training of the virtualized clone, and ethics.

A major issue raised by SIoT skeptics is related to privacy. In fact, the operations executed to build social relationships between social objects can be the cause of private information leakages. Malicious nodes that take part to the SIoT social network may exploit the social links and misuse the trust of other peers to steal important data. However, the social network itself could be used to estimate the level of trust among all the nodes and limit the interactions with the objects belonging to the community that are performing some anomalous behaviors. This could leverage the same principles followed by humans to evaluate the level of trust of any acquaintances. Indeed, in the SIoT network the peers can exchange opinions about other nodes and create a common view on the reliability of the different nodes.

Another thread could come from the widespread adoption of fog computing resources that may host the social virtual objects. It may happen that the object relies on local resources that could be hacked by external services and take control over the hosted virtual objects. All these security aspects need to be extensively study in the future to support the adoption of the SIoT technologies.

6. References

[1] Knud Lasse Lueth, “State of the IoT 2018: Number of IoT devices now at 7B – Market accelerating,” IoT Analytics, August 2018

[2] Antonio M. Ortiz, Dina Hussein, Soochang Park, Son N. Han, and Noel Crespi, “The Cluster Between Internet of Things and Social Networks: Review and Research Challenges,” *IEEE Internet of Things Journal*, vol. 1, no. 3, June 2014.

[3] L. Atzori, A. Iera, G. Morabito, “SIoT: Giving a Social Structure to the Internet of Things”, *IEEE Communications Letters*, vol. 15, no. 11, pp.: 1193-1195. Nov. 2011.

[4] L. Atzori, A. Iera, G. Morabito, M. Nitti, “The Social Internet of Things (SIoT) – When social networks meet the Internet of Things: Concept, architecture and network characterization”, *Computer Networks*, vol. 56, no. 16, 14 Nov. 2012.

[5] Alam, Kazi Masudul, Mukesh Saini, and Abdulmotaleb El Saddik. “Toward social internet of vehicles: Concept, architecture, and applications,” IEEE access 3 (2015): 343-357.

[6] A Social Web of Things by Joakim Forno, available at http://www.ericsson.com/uxblog/2012/04/a-social-web-of-things/

[7] D. Guinard, “A web of things application architecture: integrating the real-world into the web”, *Ph.D. dissertation*, 2011.

[8] A. Pintus et al., “Paraimpu: a platform for a social web of things”, *ACM World Wide Web Conf.*, 2012.

[9] C. Zhang, et al., “Architecture design for social web of things”, *ACM Workshop on Context Discovery and Data Mining*, 2012.

[10] J. I. Vazquez and D. Lopez-De-Ipina, “Social devices: autonomous artifacts that communicate on the internet,” *The Internet of Things*, Springer, 2008, pp. 308–324.

[11] D. Guinard, et al., “Sharing using social networks in a composable web of things”, *IEEE PERCOM*, 2010.

[12] Li, Hao, and Ajith Parlikad, “Social internet of industrial things for industrial and manufacturing assets,” (2016), *IFAC (International Federation of Automatic Control)* papers online, 10.1016/j.ifacol.2016.11.036

[13] D. Romerco, et al. “Social factory architecture: social networking services and production scenarios through the social internet of things, services and people for the social operator 4.0,” *IFIP International Conference on Advances in Production Management Systems*, Springer, Cham, 2017. p. 265-273.

[14] G. Ruggeri, O. Briante, “A framework for iot and e-health systems integration based on the social internet of things paradigm,” *2017 international symposium on wireless communication systems (ISWCS)*, 2017, p. 426-431.

[15] F. Cicirelli, et al. "Edge computing and social internet of things for large-scale smart environments development," *IEEE Internet of Things Journal*, 2017, 5.4: 2557-2571.

[16] O. Voutyras, et al. "An architecture supporting knowledge flow in social internet of things systems,"*2014 IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, IEEE, 2014. p. 100-105.

[17] L. Atzori, et al. "*Leveraging social notions to improve ID-to-locator mapping in IoT identity oriented networks*," *2018 IEEE 4th World Forum on Internet of Things (WF-IoT). IEEE*, 2018.

[18] Luigi Atzori, Antonio Iera, and Giacomo Morabito. "Sociocast: A New Network Primitive for IoT," *IEEE Communications Magazine*, 57.6 (2019): 62-67.

[19] I Farris, R Girau, L Militano, M Nitti, L Atzori, A Iera, G Morabito, "Social virtual objects in the edge cloud," IEEE Cloud Computing 2 (6), 20-28, 2015.

[20] Enzo Baccarelli, Michele Scarpiniti, Paola G. Vinueza Naranjo, and Leticia Vaca-Cardenas, "Fog of Social IoT: When the Fog Becomes Social," *IEEE Networks*, July/August 2018

[21] R. Morabito, "Virtualization on Internet of Things Edge Devices with Container Technologies: A Performance Evaluation," *IEEE Access*, 2017.

[22] Panagiotis Kasnesis, Charalampos Z. Patrikakis, Dimitris Kogias, Lazaros Toumanidis, Iakovos S. Venieris, "Cognitive friendship and goal management for the social IoT," *Computers and Electrical Engineering*, 2016

[23] Z.U. Shamszaman, M.I. Ali, "Toward a smart society through semantic virtual object enabled real-time management framework in the Social Internet of Things," *IEEE Internet Things J.*, 5 (4) (2018) 2572–2579.

[24] A. Li, X. Ye, H. Ning, "Thing relation modeling in the Internet of Things," *IEEE Access* 5 (2017) 17117–17125.

[25] R Girau, S Martis, L Atzori, "Neighbor discovery algorithms for friendship establishment in the social Internet of Things," *IEEE 3rd World Forum on Internet of Things (WF-IoT)*, 165-170, 2016

[26] Luigi Atzori, Claudia Campolo, Bin Da, Roberto Girau, Antonio Iera, Giacomo Morabito, Salvatore Quattropani, "Smart devices in the social loops: Criteria and algorithms for the creation of the social links," *Future Generation Computer Systems*, 2019

[27] A. Kamilaris and A. Pitsillides, "Social networking of the Smart Home," *21st Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, 2010

[28] F. Cicirelli, A. Guerrieri, G. Spezzano, A. Vinci, O. Briante, G. Ruggeri, "iSapiens: A platform for social and pervasive smart environments," *IEEE 3rd World Forum on Internet of Things (WF-IoT)*, 2016

[29] J.E. Kim, A. Maron, D. Mosse, "Socialite: A flexible framework for Social Internet of Things," *Proc. of 16th IEEE International Conference on Mobile Data Management*, 2015

[30] R. Girau, S. Martis, L. Atzori, "Lysis: A Platform for IoT Distributed Applications Over Socially Connected Objects," *IEEE Internet of Things Journal*, vol. 4, no.1, 2017

[31] R. Girau, M. Anedda, M. Fadda, M. Farina, A. Floris, M. Sole, D. Giusto, "Coastal Monitoring System based on Social Internet of Things Platform," *IEEE Internet of Things Journal*, 2019

[32] L. Atzori, C. Campolo, B. Da, R. Girau, A. Iera, G. Morabito, and S. Quattropani, "Enhancing identifier/locator splitting through social internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 2. 2018

Software Defined Fog/Edge Networking for Internet of Vehicles: a Services-Oriented Reference Architecture

Michele Bonanni, Francesco Chiti and Romano Fantacci

Department of Information Engineering

University of Florence, via di Santa Marta 3, I-50139 ITALY

Abstract: *This chapter addresses an innovative architectural concept by integrating Software Defined Networking (SDN) and Network Function Virtualization (NFV) paradigms, together with Mobile Fog/Edge Computing (MFEC). In addition, we applied this reference architectural model to the Internet of Vehicles (IoV) domain by addressing complex mobile Intelligent Transportation Services (ITSs) with stringent temporal and spatial requirements. To this purpose, we design the vehicular communications ecosystem according to a 5G vision, as it supports both Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) links by means of the cellular-enabled Vehicle-to-Everything (V2X) interface. In particular, we focus on a generalized SDN-controlled handover which is capable to handle complex traffic patterns, i.e., supporting groups of mobile nodes, geocast information delivering, and even the involvement of multiple eNBs. The idea behind our proposal is to provide a complete ITS migration that seamlessly allows to move the virtualized service (i.e., a VNF), requested by some vehicles, towards the most suited Fog/Edge Server (FESs). Finally, a strategy, called Swapping Migration, that can optimize both the resource utilization and availability, together with the Quality of Service and Experience (QoSE), has been introduced. This could enable innovative services like complex and composite route planner, autonomous driving, Tactile Internet (TI) inspired vehicles controlling in a green mobility perspective.*

1 Introduction and State of Art

The ever-increasing number of vehicles and their densification within typical mega-cities pose a remarkable challenge to the automotive industry and to public administrations in order to provide transportations safety and efficiency in a Smart City by relying on *smart* cars [2]. To this purpose, the emphasis should be put on *group* intelligence, rather than on individual capabilities of a device, which is, in fact, constrained and not able to face time-varying and unpredictable operative conditions. Recently, information and communication technologies (ICTs) have been regarded as the way to lead vehicular networks toward a revolutionary road. Connecting vehicles via Vehicular Ad-hoc Networks (VANET) represented an early attempt to support this vision, where closer vehicles are allowed to communicate with each other via Vehicle-to-Vehicle (V2V) or Vehicle-to-Infrastructure (V2I) interfaces to notify or collect traffic-related information, respectively [6][7]. Thanks to their high data rates and low latency communications, 5G

Cellular Networks (CNs) represent a promising enabling technology for VANETs, and the standardisation of the Vehicle-to-Everything (V2X) interface is paving the floor for advanced vehicular services [13]. To extend VANETs capabilities, the Internet of Vehicles (IoV) concept has been proposed to network federate vehicles by heterogeneous communication systems with the aim at providing complex applications, mainly vehicle (automated) driving, urban traffic management, or even logistics [20]. IoV represents a typical application of Internet of Things (IoT) technology in Intelligent Transportation Systems (ITS) which allows data collecting and information sharing about vehicles, roads and their surroundings [24]. Differently from IoT, whose main objective is the data-awareness of connected things, the IoV is mainly focused on the integration of vehicles and humans. Based on the cooperation and communication, IoV can handle and process a large amount of data to improve the sustainability of complex vehicular applications, which can be classified as 1) *Active road safety applications*, 2) *Traffic efficiency and management applications* and 3) *Infotainment applications* [1].

1. *Active road safety applications*: applications mainly employed to decrease the probability of traffic accidents and the loss of life.

2. *Traffic efficiency and management applications*: applications used to improve the vehicle traffic flow, traffic coordination and traffic assistance.

3. *Infotainment applications*: applications focused on infotainment that can be obtain from locally or globally based services.

In addition, it is required a paradigm shift from the traditional Cloud Computing (CC) oriented toward hybrid architectures integrating Fog Computing (FC) and Networking (FCN) to support real-time and location-aware processing and control [12]. In particular, FCN relies on a set of Fog Servers (FSs)/Fog Nodes(FNs) generally deployed *closer* to the end users and able to provide *ad hoc* and opportunistic services, i.e., personalised in time, space, clients' profile and, above all, *context*. As a consequence, FCN avoids to upload *all* the vehicles and sensors data to a remote data center and therefore, communication latencies and network congestions are reduced. One of the most important challenges to be faced when CN technology is adopted in this scenario is the specific vehicular mobility patterns. This requires more complex and effective handover policies, where not only users, but also services are transferred *among* cells, to maintain service continuity and real-time requirements. Network Function Virtualization (NFV) is a disruptive paradigm according to which, *network functions* become software applications, i.e., Virtual Network Functions (VNFs), that can be instantiated on generic servers by means of Virtual Machines (VMs) or containers. The latter approach only virtualises the Operating System (OS) without the need of a hypervisor; containers are consequently lighter than VM and present benefits in terms of boot-time and migration-time. VNFs can be dynamically created/destroyed, migrated from one physical device to another and, finally, composed together to offer innovative services. This approach, previously introduced in CC, still represents an open issue within the IoV [8] for what the vehicular VNFs placement and migration among FSs are concerned. Specifically, the reference network architecture requires a decoupling from Data Plane (DP) and Control Plane (CP) to allow a *Controller* to get a comprehensive *view* of the underlying network, and, hence, to apply specific policies, according to the Software Defined Networking (SDN) approach [18]. When SDN is applied to IoV scenarios, it can provide several benefits in terms of re-programmability, agility, scalability, elasticity and flexibility. Moreover, it allows the automatic and fast development of network applications and complex services. The potential benefits of the SDN and FC application to the IoV scenario, especially with cellular connectivity, have been addressed in the literature. Firstly, an architecture, integrating FC and CC managed by SDN in order to reduce IoV latencies and support mobility, is proposed in [5]. The authors in [14] presented a framework for vehicular CC in which vehicles together form a Cloud resource unit. This is used to handle a network slice at the WiFi Access Point (AP) in a densely crowded environment. Moreover, a methodology to share bandwidth and transmission opportunities of an AP dedicated to different slices is proposed. Khakimov *et al.* combined SDN and FC approaches in IoT ecosystem oriented framework in [17], where resources available in the OpenFlow Switches are not only used for forwarding functions but also for providing some limited services. A protocol architecture for SDN based IoV, which allows data packets forwarding over V2V and V2X links, is proposed in [22]. Iqbal *et al.* presented in [15] a data analytic framework at the Fog Layer of IoV reference model able to offer context-aware real-time and batch services at the network edge. A scheme to support seamless handover between different computation APs is presented in [10], which implicitly supports task pre-migration mechanisms when the handover is expected to happen, but it is limited to low-to-medium mobility patterns. In [3], the SDN paradigm is applied to mobile networks in order to efficiently manage mobility in the context of 5G or evolved LTE. A comprehensive overview of the migration techniques and a real Fog testbed is addressed in [19]. Li *et al.* proposed in [23] a QoS-aware approach based on the existing handover procedures to support real-time services, applying the service migration in a FC enabled cellular network under restrictive hypotheses. In particular, each FS is associated with only one Base Station (BS), while its computation and storage resources are always considered sufficient. In addition, the downtime in the pre-copy migration scheme keeps constant when data rate increases. Finally, an approach for VNFs placement and deployment a hybrid Cloud/Fog network is defined and discussed in [25], but without any migration scheme.

It is worth noting that most of the existing investigations only propose strategies to optimise the service latency under specific time constraints, without properly considering the potentialities of service migration in IoV environment, or limiting it to simplified scenarios. To face the above introduced problem, in Sec.3, we propose a framework, called *Mist* Computing (since applications *condense* from the FSs directly *onto* the user path), to support innovative time-space constrained services for a group of (semi) automated vehicles in dynamic scenarios. More in details, the main contributions of this chapter include: (1) the design of an architecture for vehicular communication which combines SDN, NFV and FC and to enable the service migration; (2) a proactive handover based on SDN; (3) a Service Management and Orchestrator implementation; (4) a swapping migration scheme, enhancing the existing IoV service migration strategies; and (5) a detailed performance evaluation of the proposed scheme, as well as, a comparison with several benchmarks, are carried out in Sec. 4, by means of numerical simulations performed over realistic scenarios in terms of end-to-end (e2e) delay, reliability and resources utilization at the FSs.

The chapter is organized as it follows. Section II introduces the most important technologies involved in the Mist Computing approach. Specifically, the concepts of FC, SDN, NFV and VM and Container Migration are reviewed. Moreover, two specific open issues are detailed. Section III describes the proposed model together with its behaviour. Analysis and discussion of the results are given in Section IV. Finally, Section V concludes the chapter.

2 Background knowledge

In this Section the most important concepts needed to understand our proposed approach are reviewed.

2.1 Fog Computing

FC is a decentralized architecture in which the resources are placed in a location between the data source and the Cloud [16]. The data produced by devices are not directly uploaded to the Cloud but are pre-processed by small data center placed close to the end-user. The intermediate network level created by the FC approach leads to the following main advantages:

- **Less core network traffic**: FC reduces the traffic between the IoT devices and the Cloud.
- **Latency reduction**: FC reduces latency between the data source and the processing center.
- **Cost saving of external network**: the cost for a fast Cloud data upload can be avoided.
- **Better security**: FNs can be protected using same procedures followed in IT environment.
- **Better privacy**: Users data are analysed locally instead of sending them to the Cloud.
- **Development of new services**: given the low latency and the proximity of FSs, it is possible to provide services with strong temporal and/or spatial constrained.

A typical FC architecture is composed of three layers, as shown in Fig. 1. The Edge-Layer (EL) includes all the *smart* devices (Edge devices) of an IoT architecture. In this layer, data generated by the IoT devices are directly processed by the devices itself or are forwarded to the FNs within the Fog-Layer (FL). The Fog-Layer is composed of a number of powerful servers that receive and pre-process data from the underlying level. Finally, if further analysis are needed, the FNs upload the data into the Cloud. The Cloud-Layer (CL) is the central endpoint of the FC architecture and it collects all the data without any stringent temporal or spatial requirements that need further processing.

IoV represents a typical application of IoT technology in a mobility scenario, which allows data gathering and information sharing about vehicles, pedestrians, roads and their surroundings. In this context, FC approach can be efficiently used to provide vehicular applications or services (e.g. autonomous driving, forward/backward collision warnings,

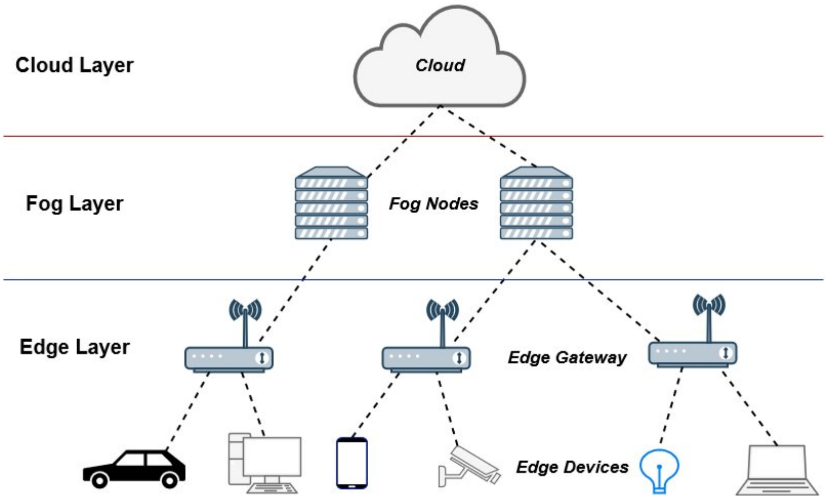


Figure 1: Fog Architecture

lane change assistance) to end users. FNs can be implemented in Road Side Units (RSUs) or Base Stations (BSs) and can host services with strict latency constraints. Instead, traffic efficiency and management applications, together and comfort and infotainment services, which do not have stringent requirements, can be directly executed into the Cloud.

2.2 Software Defined Networking

SDN is a new network architecture which introduce a clear separation between the CP and the DP w.r.t the traditional network, as shown in Fig. 2 [4]. This separation allows to get a complete view of the underlying network which can be used to solve some problems present into the traditional network:

- difficult monitoring of large network;
- complex implementation of new management policies due to the presence of many proprietary protocols.

The CP represents the network intelligence and contains one or more distributed routing protocols which are used to discover the best path from a source to a destination. The DP processes incoming packets in accord with the results obtained by the CP and, if a packet matches a routing flow entry, it is forwarded on the physical port that leads to the next device on the path. As shown in Fig. 3, in a SDN network, the DP is composed of forwarding devices, also known as Switches, whereas the CP is implemented in one o more servers, known as SDN controllers. The more important characteristic of the network devices in an SDN network is that these devices perform a simple forwarding function, without embedded software to make autonomous decision. The SDN controller maps the service requests coming from the AppP, into specific commands and directives to

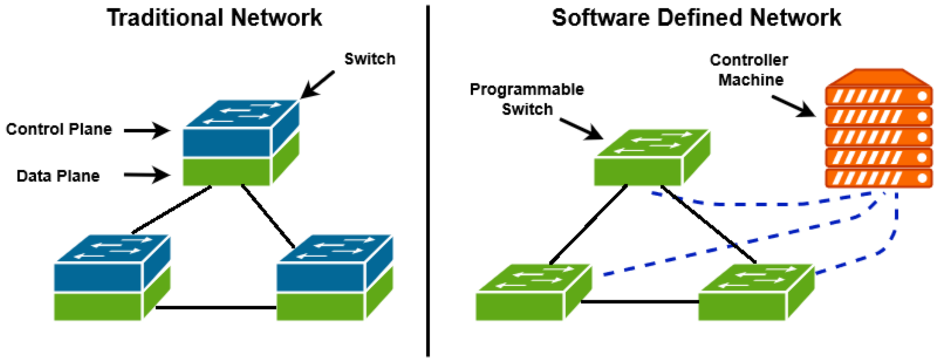


Figure 2: Traditional Network and SDN

DP switches and supplies applications with information about DP topology and activity. Moreover, every controller should provide some essential functions to the AppP, such as:

- **Shortest path forwarding**: uses information collected by the forwarding devices to establish routing path;
- **Notification manager**: processes and sends to an application events related to the network, such as security alarms;
- **Topology manager**: builds and provides to the upper layer an abstract view of the network;
- **Statistics manager**: collects traffic data of the underlying network.

The AppP is the higher level of the SDN architecture and contains applications or network services that monitor, control and manage the network and its resources. The northbound interface (i.e. REST API) enables the network applications to use the CP functions and services without needing to know the details of the underlying network. CP can be of two different types: centralized or distributed. In the former case, only one SDN controller is used, whereas in the latter, multiple SDN controllers are deployed in a hierarchical or peer-to-peer fashion and they communicate each other through the East/Westbound interface. Moreover, this interface is used to connect SDN autonomous systems to traditional non-SDN autonomous ones and to execute the Border Gateway Protocol (BGP). Finally, the southbound interface provides the logical connection between the SDN Controller(s) and the DP forwarding devices. The most commonly implemented southbound API which represents the *de facto* standard is OpenFlow.

SDN, has been mainly designed to simplify the management of wired networks and data centers. However, its features in terms of re-programmability, agility, scalability, and elasticity can be efficiently extended to other networks, such as cellular networks, wireless sensor networks (WSN) and IoV networks. In the latter case, SDN can reduce the gap between IoV and safety applications and can improve the utilization of network resources. More in detail, the network view possessed by the Controller(s) can be used in conjunction

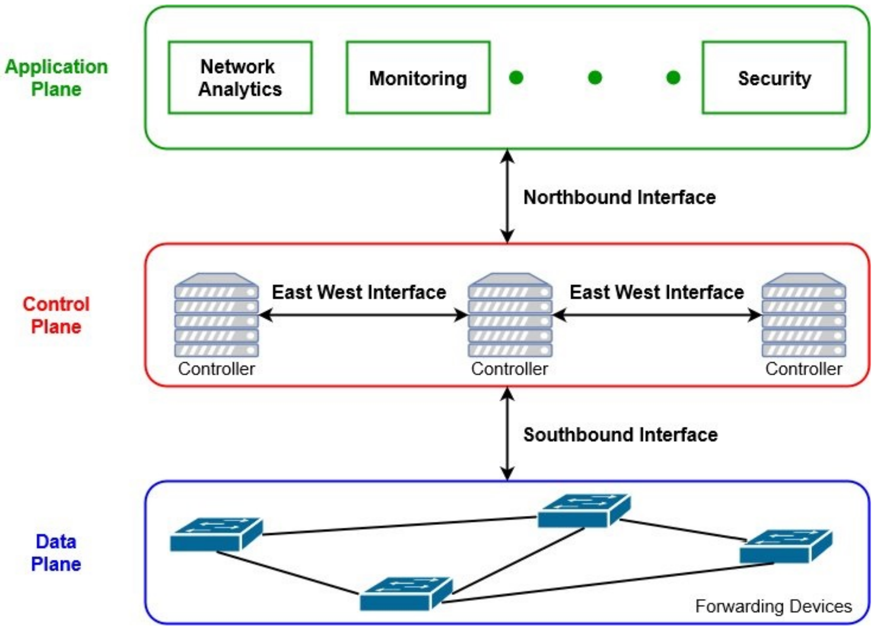


Figure 3: SDN Architecture

with a vehicle traffic prediction model to optimize the data routing between vehicles, pedestrians and infrastructure. In this scenario, Controller(s) has to respond quickly to network changes, thus a combination of SDN and FCN is needed. It is important to point out that the design of a vehicular network architecture, which can combine these two paradigms in a wireless scenario, raises two open issues that should be addressed.

The first one is the mapping of the software defined devices into the FCN architecture. In the IoV perspective, the FCN architecture is composed of many FSs/FNs, whose main objective is to quickly process data coming from IoV devices and to immediately provide results. Thus, it can be assumed that part of the SDN Application Plane (AppP) is located into the FSs. The remaining part, which includes applications/services not real-time may be placed into the Cloud. The SDN Controllers do not have a fixed position, therefore they can be implemented on the same FSs that already host the network applications or can be placed in other servers far from the access network. According to the first approach, Controllers can quickly manage the forwarding devices, but since they are placed closer to the end-devices, they can only be aware of a limited part of the network and therefore a *distributed* CP is needed. With the latter one, instead, Controllers have a complete view of the underlying network, but the latencies introduced are higher than in the former case and it can be unacceptable for real-time services. A possible solution is to merge the previous two cases in a hierarchical architecture in which, (i) a lower and distributed CP manage small areas and quickly answer to applications with stringent temporal constraints and, (ii) an upper and more centralized Controllers handle the communication among the underlying Controllers and between them and the Cloud. Since there are no specific requirements, the IoV devices can be used as forwarding devices becoming both data source and Switch. Edge devices together with the gateway/access point, usually use a wireless channel to communicate among each other. However, although SDN and OpenFlow work efficiently in wired network, the integration of these new concepts over a wireless link represents the second open issue that needs to be addressed. Two feasible solution could be implemented, namely, (i) the use of the gateway as a *proxy* which translates the OpenFlow commands in messages that can travel on a wireless channel, or (ii) directly enabling the OpenFlow protocol over wireless links.

In conclusion, it can be pointed out that the FCN can host different layers of the SDN architecture, but the correct mapping between the software defined devices and the FSs still represents an open issue. Indeed, in the resulting Software Defined Fog Network (SDFN), the devices may simultaneously belong to different SDN layers. Moreover, although the DP of an SDN could be extended until the edge network, an OpenFlow version which can transit on a wireless channel has not been yet implemented.

2.3 Network Function Virtualization

NFV can be defined as the separation process of the network functions (i.e. Network Address Translation (NAT), firewalling, intrusion detection, Domain Name Service (DNS)) from proprietary hardware appliances. The core of NFV is composed of VNFs that handle specific network functions [11]. A VNF can run on VM or container on commercial off-the-shelf (COTS) servers.

A VM can be defined as a virtual environment which emulates the behaviour of a physical machine. The hosting machine shares its hardware resources (RAM, CPU, GPU and Hard disk/SSD) with the VMs, allowing their simultaneously execution. The component that orchestrates the physical resources for the VMs and allows its coexistence is known as Hypervisor. Differently by the VMs, containers do not virtualize the underlying hardware but share the same OS, the kernel, the network connection and the libraries. Moreover, the instances are executed inside a separate space, guaranteeing a reduction of the CPU consumption.

Multiple VNFs can be combined or connected together (VNF service chaining) to create end-to-end services. As shown in Fig 4, the NFV framework mainly consists of three domains of operation:

- **Virtualized network functions**: set of VNFs, that run over the NFVI. A VNF may be composed of one or more VNF components (VNFC), each of which implements a VNF functionality.
- **NFV infrastructure (NFVI)**: comprises the hardware and software resources that create the environment on which VNFs are deployed. NFVI virtualizes physical computing, storage, and networking and places them into resource pools. The virtualization is the most important operation in the NFVI domain, because it abstracts the hardware resources and guarantees the life-cycle independence between the VNFs and the underlying hardware.
- **NFV management and orchestration (MANO)**: manage the software/hardware resources and the life-cycle of VNF instances on top of the NFVI. Examples include VNF instance creation, VNF service chaining, monitoring and shutdown.

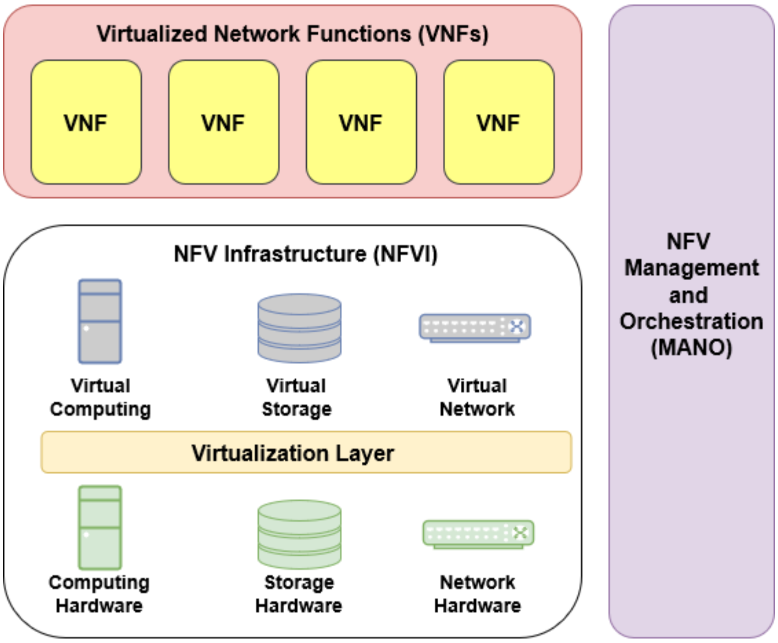


Figure 4: High level NFV framework

If the NFV concept is implemented efficiently, it can bring several of benefits compared with the traditional networking approach. The most important ones are: (i) reduction of the Operating Expense (OPEX) and the Capital Expenditure (CAPEX), (ii) faster deployment of new innovative services at lower risk and, (iii) higher flexibility and agility, by scaling up and down the services and their resources.

In the IoV scenario, NFV can improve network flexibility, while increasing the network resources consumption. More in detail, it can distribute vehicular applications and services and can dynamically manage the VNF resources according to the end user's service requests and the vehicle traffic evolution. As previously mentioned, VNFs can run in VMs or containers on COTS server or into the Cloud, thus a scenario that combines NFV and FCN has to be considered. Moreover, NFV can cooperate with SDN to effectively achieve the service chaining technique and then provide more complex services to end users.

2.4 Virtual machines and Containers Migration

As explained in the previous Section, the NFV can bring many benefits to the network operator. Furthermore, since the network functions are implemented through VMs or containers on standard server, it is possible to move them from one physical machine to another one depending on the needs. This procedure, known as *migration*, brings high flexibility and adaptability to the network and it is mainly implemented in the Cloud but can be extended to the Fog environments [9]. The most relevant characteristics that have to be considered when the migration is applied in Fog scenarios are:

- Fog environments are composed of heterogeneous devices with different capabilities, resources and operating system. Hence, a virtualization platform which can run on different devices is needed.

- Unlike the Cloud approach, FSs are interconnected through wide area network (WAN) and therefore the packet transmission is subject to higher delays and lower throughputs.

- The migration time is an important parameter that has to be considered in the Fog network because its prolonged duration can degrade the overall migration performance.

- The hardware resources of FSs are limited and lower than the resources of a Cloud data center.

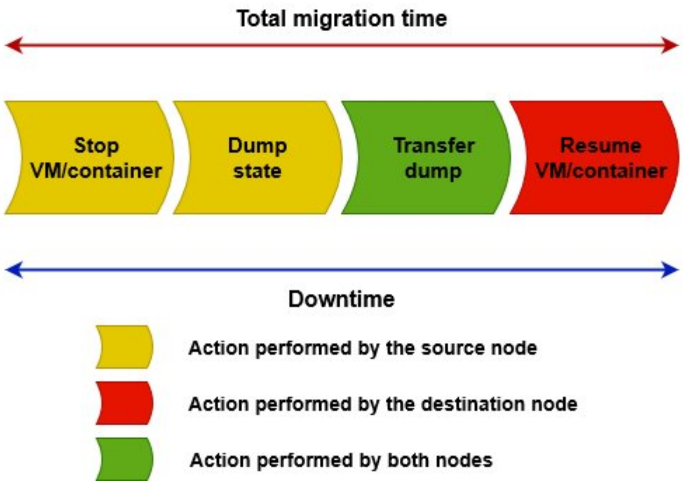

Total migration time
Stop VM/container
Dump state
Transfer dump
Resume VM/container
Downtime
Action performed by the source node
Action performed by the destination node
Action performed by both nodes

Figure 5: Cold Migration

The migration techniques may be either *stateless* or *stateful.* The former simply starts with setting up a new container on the destination node and quitting the old container on the source node. The latter, instead, moves the memory pages and the execution state from the source to the destination. Therefore, the new container starts exactly from where the old container stopped. The stateful techniques may be further categorized into two other classes known as cold and live migration.

As shown in Fig. 5, the *cold migration* executes the following steps:(i) stops the VM/container to ensure that it will no longer change the state; (ii) dumps the entire state and transfers it to destination leaving the source VM/container stopped; and (iii) resumes the VM/container at destination only when all the state has been correctly transferred. The main consequence of this procedure is that the time required to perform the migration, called migration time, is equal to the downtime, which is the time interval during which the VM/container is not running. To avoid this drawback two live migration techniques have been proposed. The first, known as *Pre-copy migration* or *iterative migration* transfers, through multiple iterations, the execution state and the memory pages modified during the previous iteration, also known as dirty pages, to the destination node before the stopping of the source VM/container. Then, like the cold migration, a final dump is performed and the last modified pages along with the changes in the execution state are transferred to the destination where the VM/container will be eventually resumed.

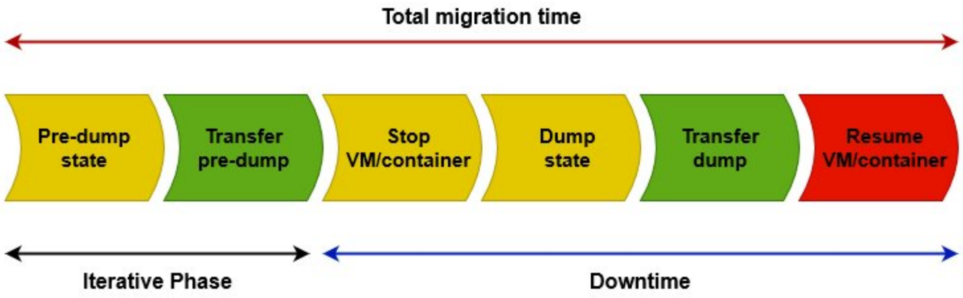


Figure 6: Pre-copy Migration

As specified in Fig. 6, the downtime is lower than the total migration time and generally is lower than the downtime of the cold migration scheme. This is primarily due to the nature of their dumps. The dump in cold migration includes all the memory pages and the execution state, whereas in pre-copy migration it contains only the memory pages modified during the previous iteration along with the last modification in the execution state. Hence, since less data are transferred in pre-copy migration during the source VM/container suspension, the total downtime is reduced. The main drawback of this approach, that can increase the downtime, is the number of dirty pages. In fact, a large page dirtying rate increases the quantity of dirty pages that have to be transferred each time and, consequently, the time to forward them to the destination node. The second live migration technique that reduces the cold migration downtime and allows to perform part of the migration process without stop the source VM/container is known as, *post-copy migration*. This approach first stops the VM/container on the source node and transfers the execution state to the destination, thus the new VM/container can resume its execution from there. Only after this step, all the memory pages can be copied to the destination machine through different techniques. One of these, called *lazy* migration transfers the memory pages only when the resumed VM/container cannot find them at the destination server. As shown in Fig. 7, if the requested memory page are not present, the VM/container generates a page fault and contacts the page server on the source node. This server then transfers the faulted pages to the destination.

The post-copy migration presents a downtime lower than the cold migration downtime and comparable with that of pre-copy migration. This is mainly due to the dump phase, because only the execution state without any memory pages is transferred. Differently by the pre-copy solution, the performance of this live migration technique does not depend of the page dirtying rate but it is subject to a non-immediately availability of the memory pages at the destination node that may be unacceptable for time-constrained services.

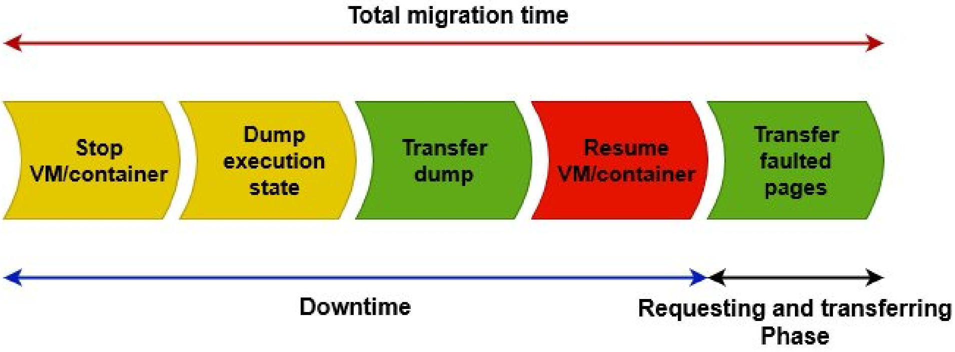


Figure 7: Post-copy Migration

Migration techniques can be applied to wireless network, and especially to IoV network to maintain low latencies between *Client* and *Server* and obtain a higher Quality of Experience. The main idea, which represents the basis of the Mist Computing approach, is to use the migration concept to move the user's service(s) from one server to another when it is needed. More details of our approach are described in the following Section.

3 Overall System Architecture and Service Management Model

The proposed Mist Computing architecture, combining FC, SDN and NFV concepts, is depicted in Fig. 8, where it is evident the classical three-layer arrangement (DP, CP, and AppP). The building elements have been properly adapted to the vehicular ecosystem requirements, where the wireless access is provided by the LTE Advanced access network. In particular, it involves specific APs (i.e., eNodeBs), but, unlike to the classical LTE architecture, in which they are directly connected to the Mobility Management Entity (MME) and the Serving Gateway (SGW), they are paired with an OF Switch.

In our vision, the several entities that compose the LTE CN are virtualized in VNFs and are placed into the Cloud. The routing of data and signalling traffic is still IP-based but is managed by one or more SDN Controllers. The VNFs that are involved in the handover procedure can access to the SDN Controller(s), modify the data flow from one eNB to another one, and achieve both hard and soft handover, as introduced in [21].

The SDN principle gives to the Controller(s) a complete network view to optimally manage the system in terms of (i) traffic flows forwarding and (ii) ad hoc deployment of innovative applications and algorithms without modifying the devices fabric. In the proposed architecture, DP is comprised of eNBs and OF Switches, the latter only being connected to the CP through the OpenFlow protocol. FSs belong to an intermediate level, called Fog Plane (FP), as represented in Fig. 8; they are connected to OF Switches and pre-loaded with the application's *images* provided by the Service Provider (SP), which usually adopt Operational and Business Support Systems (OSS/BSS). According to a *publish & subscribe* pattern, vehicles select a specific service to be personalised

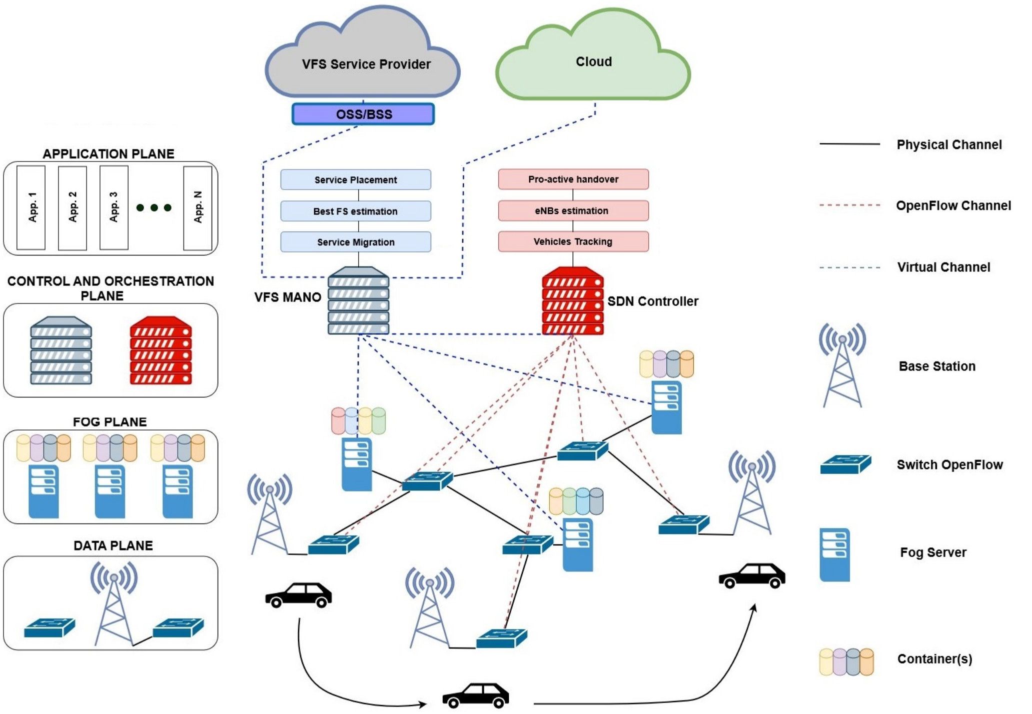


Figure 8: Mist Computing architecture, highlighting actors and interfaces of vehicular ecosystem.

which is executed a specific container. In our approach, Controller(s) have additional capabilities to support *generalised* handover related to a *group* rather than individual mobility, if compared with the features available on MMEs and eNBs of traditional 4G cellular system [21].

The *handover VNF* running on the Controller acts as a *proxy* between the eNB and the CN (MME, S-GW, etc.) and collects data from vehicles. In particular, the eNB sends handover requests to the aforementioned VNF, which will adequately manage them. In case of INTRA-MME/SWG handover with enabled X2 interface, it will only capture information without directly acting on the handover procedure. If the X2 interface is not enabled, the handover VNF will manage the procedure without contacting the CN entities and it will use the SDN Controller to modify data flow towards the destination eNB. Finally, if the handover is INTER-MME or INTER-MME/SGW, the VNF will transparently forward the request to the CN, which will manage the handover as in LTE Advanced. In this way, the overall traffic overhead and latencies are reduced and the SDN Controller can intercept vehicles that are about to perform the handover.

Furthermore, all the network information owned by the Controller is used to perform the service migration. This involves the adoption of a Vehicular Functions and Services Management and Orchestrator (VFS MANO), able to both manage FSs resources and create, delete and migrate the vehicular services (e.g., autonomous driving, remote driving, pre-crash warning, platooning). The service migration aims at maintaining services close to the users, then minimising the latency between the currently associated eNB and the FS hosting the service.

The Mist Computing approach relies on the service migration concept to offload the user's service(s) from one FS to another one, in accordance with the vehicle's path, which has been previously predicted and processed. However, it requires an additional delay to transfer service status (usually a file or a page) to the target FS. Thus, when this information is not negligible, the delay may become quite larger, so that the service migration should not be performed every time a vehicles initiates a handover, as pointed out in [19].

In this chapter, the benefits provided by a correct service migration to a vehicular network are investigated. To this purpose, we analyse two schemes, that is, no migration (Scheme 1) and the proposed optimized pre-copy migration for mobile scenarios (Scheme 2). Since both of them are not able to properly support services with hard time constraints, we propose a swapping migration scheme (Scheme 3) in which service migration is performed to both optimise the overall delay and the services availability at each FS. In both Scheme 1 and 2, a service requested by a vehicle is placed in a FS able to satisfy the QoS requirements. If no FSs are available, that request is rejected or redirected to CC. Scheme 1 is considered as benchmark in which no service migration is pro-actively performed while the vehicles are moving. In order to guarantee the service continuity, the vehicles maintain the connection with the FS where the requested service is running on a container. Consequently, the e2e latency, which includes the propagation, transmission and queuing delay, becomes larger as the car moves away from the FS that is hosting the service. In scheme 2, our optimized pre-copy migration mechanism is considered. In order to select the *best* FS to perform the service migration on, the VFS MANO predicts vehicle(s) position(s): it is usually based on the previously evaluated journey since this represents the core service for a traffic *navigation* application, which is also regularly updated by the underlying *tracking* service. Once obtained the vehicle path and the eNBs list along it, the VFS MANO applies the Djkstra algorithm over the network view owned by the SDN Controller to evaluate the best FS to migrate the service. Then the VFS MANO starts a *pre-copy* container migration by means of an iterative *pre-dumping* phase on the selected FS. Resorting to the SDN Controller, the VFS MANO can track the vehicle position and activate the last phase of the pre-copy migration when it is close *enough* to the target FS, while performing the handover. Differently from the pre-copy migration conventionally used for mobile networks [23], in which the entire migration starts when the vehicle connects to the new eNB, in this optimized scheme only the *last* phase of the pre-copy migration is executed during the handover, therefore, the overall downtime is reduced. In particular, this delay is composed of the handover downtime and migration downtime. In Scheme 2, both are overlapped and, thus, the total downtime is lower than the conventional approach. The main drawbacks of this solution is the scalability: when the service requests are more than resources available in all the FSs, the system drops new requests or redirects them to CC, hence, increasing the service outage or the e2e delay, respectively. In Scheme 3, we propose an optimisation of the Scheme 2, called *swapping-migration*, which is activated *only* if it is necessary to free up resources on a certain FS to be used for another container. The active container *releasing* consists in migrating it to another appropriate FS capable of hosting the service, without degrading its performance but guaranteeing QoS requirements. Specifically, the VFS MANO is aware of the containers/services distribution within FSs, together with the resources needed to satisfy the QoS requirements. Moreover, by interacting with the SDN Controller, the VFS MANO gets a complete view of the underlying network features so that it can: (1) evaluate the best target FS to migrate the active container in order to free up resources to host a new service and, (2) avoid that the migration could not match the QoS requirements of the running service.

4 Results Analysis

The integrated management approach presented in Sec. 3 has been evaluated by means of numerical simulations resorting to (i) an ad hoc Python 3 framework to model the container migration and the VFS MANO, (ii) the Simulation of Urban MObility (SUMO) to generate vehicular traffic and (III) the Mininet WiFi to emulate the Software Defined Wireless Network.

The test scenario is composed of 20 eNBs, 30 FSs, 20 Switches OF, 1 SDN Controller and 1 VFS MANO, representing a realistic urban case study, with 2500 vehicles over this area. To allow service migration, FSs has been supposed to be connected via high-capacity links, such as based on fiber or millimeter-wave.

Table 1: Parameters adopted in simulations.

Parameter	Value
Coverage of the urban area	$150\,\mathrm{km}^2$
Total number of vehicle	2500
Link speed for upstream and downstream in FN	1 Gbps
Resources needed by the vehicular services	10−100 Mbits
Maximum number of containers in a FS	500
Coverage for a single eNB	$8\,\mathrm{km}^2$
Number of eNB	20
Number of FS	30
Number of OpenFlow Switch	20
Number of SDN Controller	1
Number of VFS MANO	1
Handover interruption time	10 ms

Table 2: Number of requests.

Level of services demand	Value
Low	0−5000
Medium	5000−10000
High	10000−15000

The eNBs maximum coverage radius is 8 km, while the length of the wired links (dotted lines in Fig. 8) does not exceed 15 km and with a capacity of 1Gbps. Moreover, there are 30 services offered by the VFS SP, which are different in terms of the resources (RAM, CPU) requested for their execution, whereas FSs are characterised in terms of available resources so that they can accommodate only a variable number of containers and services. More in detail, the size of each container hosting the service is comprised of 10 to 100 Mbits and the maximum number of containers that a server can host is 500. The maximum number of service requests is 15000, in order to saturate the resources available on the FSs and verify our approach. OF Switches interconnecting eNBs and FSs are managed by the SDN Controller, which dynamically modifies the flow rules. All the simulation parameters are summarised in Table 1.

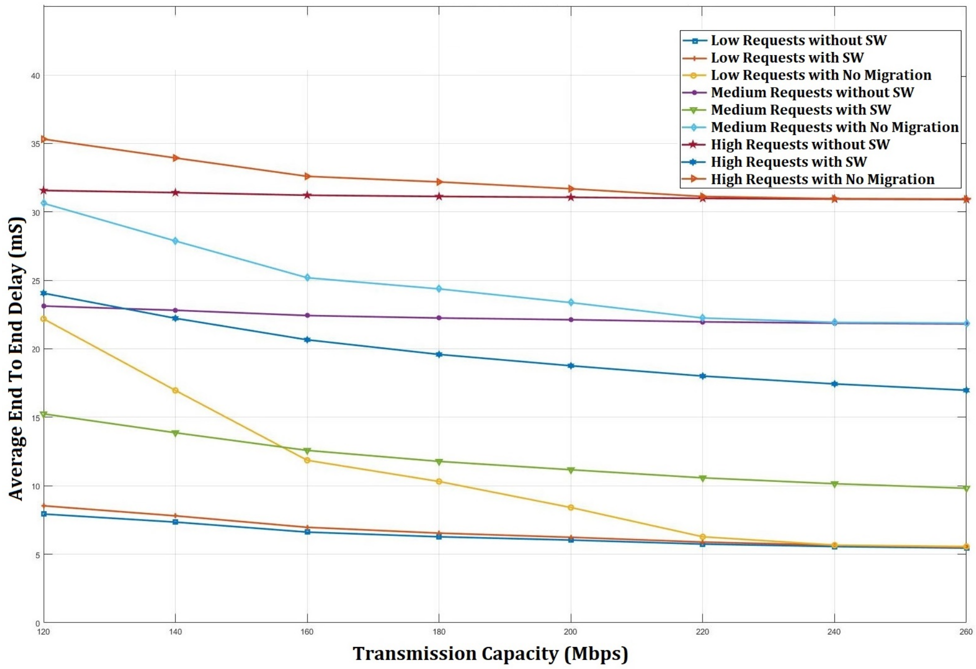


Figure 9: Average e2e service completion latency for different transmission capacities and service request rates.

Vehicles trajectories are randomly generated, and, in our optimized pre-copy migration scheme, they are shared with the *eNBs estimation* application running on the SDN Controller, in order to evaluate the probable eNBs along with them. The VFS MANO, in turn, uses this results (i) to point out the possible FSs on which to migrate the service (provided that they have enough resources), (ii) to load on those FS the proper containers, (iii) to activate the pre-copy migration and, (iv) to execute the last phase of the pre-copy migration when the vehicle is about to perform the handover to the *target* eNB, which represents the last eNB after which the e2e latency is greater than the service delay constraint. In particular, the latter procedure is initiated by the VFS MANO monitoring the vehicles movements through the SDN Controller. Finally, as described in Sec. 3, to overcome some limitations of the optimised pre-copy migration, the swapping-migration scheme is introduced. These proposed solutions (with or without container swapping) are besides compared in terms of service e2e latency, resource allocation and service outage w.r.t. a static vehicular services placement over FSs, for different values of capacity and service requests, as shown in Table 2.

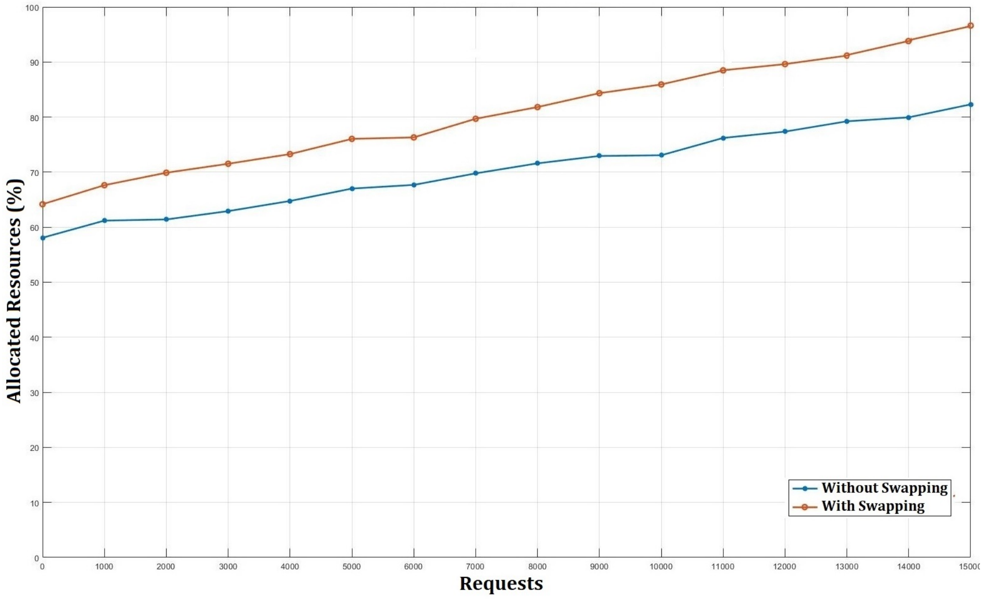


Figure 10: Average normalised resource utilisation per FS.

In Fig. 9, the average e2e service completion latency for the considered three Schemes with different transmission capacities and services request rates, is sketched. It is worth pointing out that in general delay decrease as the transmission capacity of the Fog network increases, since higher bitrate leads to shorter packet transmission time, which reduces the packet queuing delay in the forwarding devices and results in the smaller e2e delay. However, the three Schemes behave in a different way at the increasing of vehicular service request rate, as it causes both a decrease of available resources in the FSs and a more intense use of the traditional CC, therefore, a worsening of the e2e delay performance. The optimised pre-copy migration (Scheme 2 or migration without swapping) outperforms the one without migration support, especially for low transmission capacity, since the service is *deployed* along the trajectory of the requesting vehicle and the overall delay is kept low. In both cases, if the resources available in the FSs are not enough to host a new service, the CC approach is applied, causing an increase of the latency. The swapping oriented Scheme 3 follows the same procedure of the optimised pre-copy migration but, if resources are limited, it tries to satisfy the request using the FC instead of the CC. As explained in Sec. 3, this is performed by migrating an active container to another FS without degrading its performance or violating QoS requirements; once the migration is completed, some resources of the selected FS are released for hosting new services. Even though this procedure requires some additional latency to be accomplished, the average e2e delay remains lower than the previous two approaches, especially for high service request rate. From Fig. 10 it is evident that the swapping-migration increases the normalised resources utilisation (η) within FSs of about 13%, w.r.t. the migration without swapping.

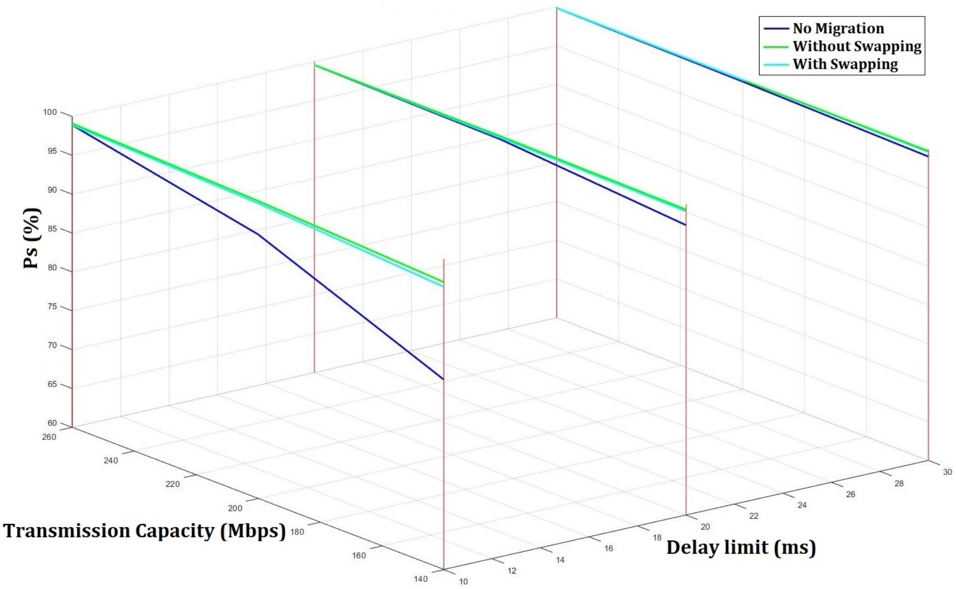


Figure 11: P_s achieved by the three different schemes for low service request rate, variable transmission capacities and different delay limit δ^* values, in the case of low service request rate.

Table 3: Reliability with low service demand.

	No Migration (%)			**Without SW (%)**			**With SW (%)**		
Mbps	10 ms	20 ms	30 ms	10 ms	20 ms	30 ms	10 ms	20 ms	30 ms
140	84.46	97.3	99.1	97	99.3	99.8	96.4	99.1	99.7
200	94	99.2	99.6	98.3	99.6	99.8	98	99.4	99.8
260	98.9	99.6	99.9	99.1	99.6	99.9	98.9	99.6	99.9

As previously described, this is due to the migration of active containers into available FSs and, consequently, the hosting of additional services. It is worth noting that the gap between the curves increases as the service request rate grows. For low-to-medium request rates both Scheme 2 and 3 mainly resort to resources available in the FSs making η to increase. For high request rate values, instead, the Scheme 2 (i.e., without swapping) uses CC more than FC, causing a resource wasting in the FSs. Conversely, Scheme 3 thank to the swapping procedure still uses FSs resources in an optimal way approaching the maximum value of η. In the case of strictly real-time vehicular services, a most relevant performance metric is represented by the success probability P_s, which is defined as the probability that the e2e service completion latency does not exceed a maximum value δ^*.
In Fig. 11, Fig. 12 and Fig. 13, P_s is investigated for all the three different schemes with variable transmission capacities and δ^* equal to 10, 20 and 30 ms. Particularly, the

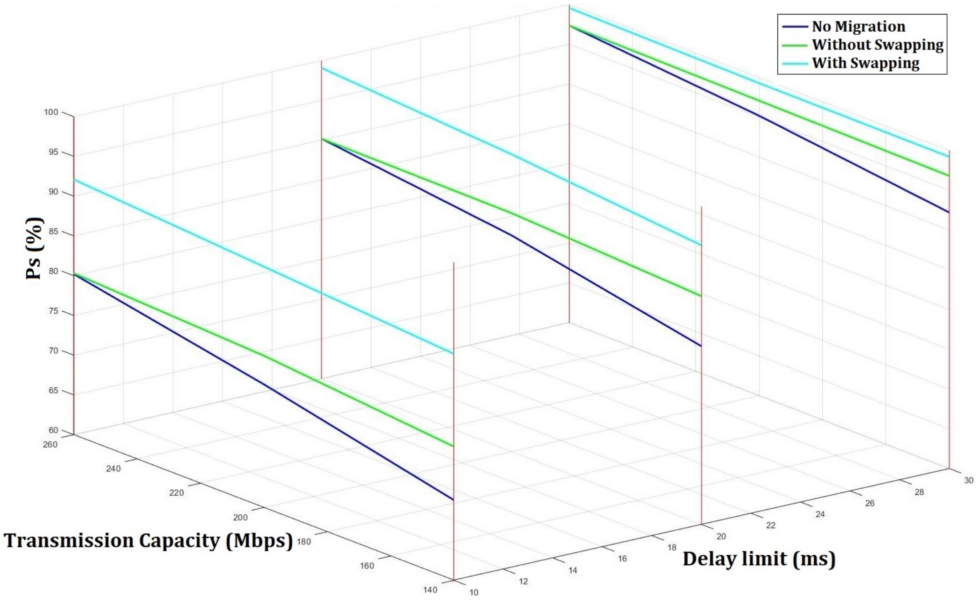


Figure 12: P_s achieved by the three different schemes for medium service request rate, variable transmission capacities and different delay limit δ^* values, in the case of medium service request rate.

Table 4: Reliability with medium service demand.

	No Migration(%)			**Without SW (%)**			**With SW (%)**		
Mbps	10 ms	20 ms	30 ms	10 ms	20 ms	30 ms	10 ms	20 ms	30 ms
140	70.1	82.4	92.2	76.83	88.7	96.8	88.49	95.1	99.2
200	75.5	87.2	95.2	79.1	90	97.1	90.3	97.4	99.3
260	80.2	90.2	97.4	80.3	90.2	97.4	92.1	99.1	99.6

three figures show the evolution of the P_s in case of low, medium and high service request rate, respectively. For a better understanding, the results obtained are also summarized in table 3, 4 and 5.

In all three figures, the P_s increases as the transmission capacity in the Fog Network increases. For low request demand, Fig.11, the migration with and without swapping has about the same performance, keeping the probability higher than 96.4% for all the three latency levels. The approach without migration, instead, needs an higher transmission capacity or a lower delay constraint to achieve the same performance. For medium and high service request rate, Fig.12 and Fig.13, it is evident that the swapping-migration scheme always outperforms the other ones, with a more remarkable gain in the case of both low delay constraints δ^* and transmission capacities.

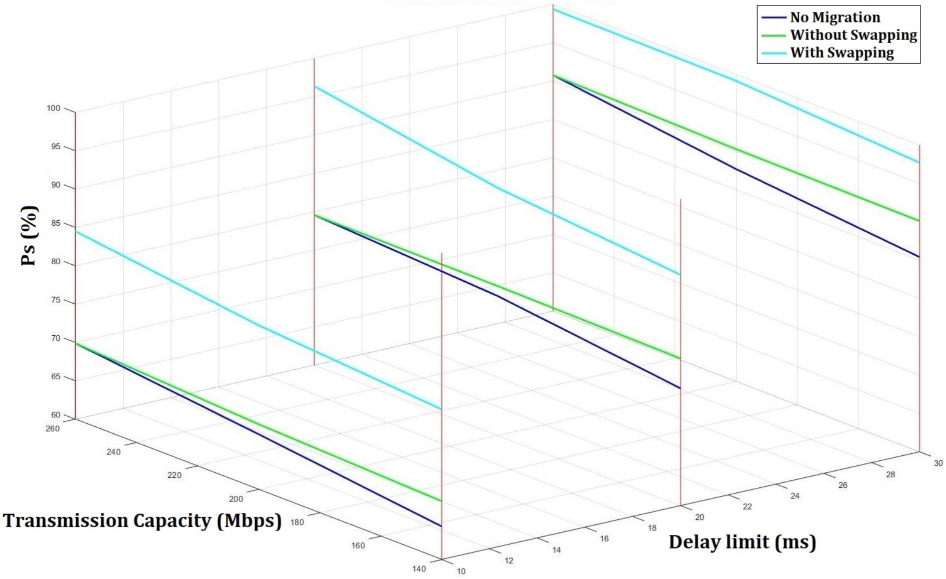


Figure 13: P_s achieved by the three different schemes for high service request rate, variable transmission capacities and different delay limit δ^* values, in the case of high service request rate.

Table 5: Reliability with high service demand.

	No Migration(%)			**Without SW (%)**			**With SW (%)**		
Mbps	10 ms	20 ms	30 ms	10 ms	20 ms	30 ms	10 ms	20 ms	30 ms
140	64.32	75.3	85.4	67.62	79.2	90.1	79.6	90.1	97.7
200	67.2	78.2	87.7	68.5	79.5	90.3	81.4	92.3	99.2
260	69.9	79.6	90.8	69.9	79.6	90.8	84.5	96.4	99.4

In Fig.12, the swapping-migration scheme achieves the maximum probability of 99.6%, when the maximum delay limit is set to 30 ms and the transmission capacity is 260 Mbps, whereas, the other two schemes achieve a maximum value that not exceed 97.4%. In Fig.13 it is possible to notice that for high service request rate, when the δ^* is equal to 10 ms and capacity assumes the lower value, the scheme with swapping achieves a maximum P_s value of about 80% which is considerably higher than the no swapping and static schemes of 16% and 20%, respectively. Vice versa, if the delay limit is set up to 30 ms and the transmission capacity is the maximum attainable, the P_s of the swapping scheme is about 99,4%, while that of the other two approaches does not exceed the 91%.

5 Conclusions

In this chapter, we proposed a vehicular services oriented framework to provide mobile real-time applications. To this end, we derived a general network architecture based on LTE Advanced technology, which properly combines the innovative concepts of SDN, NFV and FC. To address these specific services issues, various service migration approaches have been evaluated and a near optimal procedure, called swapping-migration, is introduced. In particular, in our scheme, the VFS MANO cooperates with the SDN Controller to both predict the best FSs on which pre-migrates the service and the time instant in which to start the final phase of the pre-migration. Moreover, in order to maximise the FSs resources utilisation, the proposed scheme performs active services additional migrations between FSs. The performance of this approach, compared with other ones available in the literature, has been evaluated in terms of e2e service completion delay, outage probability and resource allocation by means of realistic simulations over an urban area. All the tests pointed out that the proposed swapping-migration scheme outperforms the alternatives especially for high service request rate and low capacity, since it always keeps the service topologically *close* to the requesting users, while optimising the use of the Fog Network. As future developments, we are planning to apply Machine Learning (ML) techniques running on collected dataset to perform accurate vehicular traffic and service request predictions with the aim of deriving optimal policies (usually in the Reinforcement Learning (RL) class), which can be implemented into the VFS MANO to improve the service migration. With reference to this aspect, advanced migration procedures could be investigated and their performance tested in vehicle scenarios. Another interesting topic to be addressed is concerned with the overall reliability since the VFS MANO and the SDN Controller represent a single point of failure, so that a *distributed* CP and *multiple* VFS MANOs need to be considered in future design.

References

[1] 2011. *Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions* - IEEE communications surveys & tutorials, 13(4), 584-616.

[2] 2014. *Connected Vehicles: Solutions and Challenges* - IEEE Internet of Things Journal, 1(4), 289-299.

[3] 2014. *Handover management in SDN-based mobile networks* - 2014 IEEE Globecom Workshops (GC Wkshps), 194-200.

[4] 2014. *Software-defined networking: A comprehensive survey* - Proceedings of the IEEE, 103(1), 14-76.

[5] 2016. *A novel load balancing strategy of software-defined cloud/fog networking in the Internet of Vehicles* - China Communications, 13(2), 140-149.

[6] 2016. *Connected-Vehicles Applications Are Emerging [Connected Vehicles]* - IEEE Vehicular Technology Magazine, 11(1), 25-96.

[7] 2016. *Interworking of DSRC and Cellular Network Technologies for V2X Communications: A Survey* - IEEE Transactions on Vehicular Technology, 65(12), 9457-9470.

[8] 2016. *Providing flexible services for heterogeneous vehicles: an NFV-based approach* - IEEE Network, 30(3), 64-71.

[9] 2017. *A critical survey of live virtual machine migration techniques* - Journal of Cloud Computing, 6(1), 23.

[10] 2017. *Follow me fog: Toward seamless handover timing schemes in a fog computing environment* - IEEE Communications Magazine, 55(11), 72-78.

[11] 2017. *Network functions virtualization: An overview and open-source projects* - 2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM), 1-6.

[12] 2017. *Vehicular Fog Computing: Architecture, Use Case, and Security and Forensic Challenges* - IEEE Communications Magazine, 55(11), 105-111.

[13] 2018. *5G for Vehicular Communications* - IEEE Communications Magazine, 56(1), 111-117.

[14] 2018. *Congestion mitigation in densely crowded environments for augmenting qos in vehicular clouds* - Proceedings of the 8th ACM Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications, 49-56.

[15] 2018. *Context-aware data-driven intelligent framework for fog infrastructures in Internet of vehicles* - IEEE Access, 6, 58182-58194.

[16] 2018. *Fog computing architecture, evaluation, and future research directions* - IEEE Communications Magazine, 56(5), 46-52.

[17] 2018. *IoT-fog based system structure with SDN enabled* - Proceedings of the 2nd International Conference on Future Networks and Distributed Systems, 62.

[18] 2018. *Software-Defined and Fog-Computing-Based Next Generation Vehicular Networks* - IEEE Communications Magazine, 56(9), 34-41.

[19] 2019. *Container Migration in the Fog: A Performance Evaluation* - Sensors, 19(7):1488.

[20] 2019. *Internet of Autonomous Vehicles: Architecture, Features, and Socio-Technological Challenges* - IEEE Wireless Communications, 26(4), 21-29.

[21] 2019. *Mobility Management for Intro/Inter Domain Handover in Software-Defined Networks* - IEEE Journal on Selected Areas in Communications, 37(8): 1739-1754.

[22] 2019. *SD-IoV: SDN enabled routing for internet of vehicles in road-aware approach* - Journal of Ambient Intelligence and Humanized Computing, 1-16.

[23] 2019. *Service Migration in Fog Computing Enabled Cellular Networks to Support Real-Time Vehicular Communications* - IEEE Access, 7, 13704-13714.

[24] 2019. *Vehicular Ad Hoc NETworks versus Internet of Vehicles-A Comparative View* - 2019 International Conference on Networking and Advanced Systems (ICNAS), 1-6.

[25] 2019. *VNF placement optimization at the edge and cloud* - Future Internet, 11(3), 69.

IoT-enabled Smart Monitoring and Optimization for Industry 4.0

Luca Davoli, Laura Belli, Gianluigi Ferrari

Internet of Things (IoT) Lab, University of Parma & things2i s.r.l.

CNIT, Research Unit of Parma

Abstract: *In the last decades, forward-looking companies have introduced Internet of Things (IoT) concepts in several industrial application scenarios, leading to the so-called Industrial IoT (IIoT) and, restricting to the manufacturing scenario, to Industry 4.0. Their ambition is to enhance, through proper field data collection and analysis, the productivity of their facilities and the creation of real-time digital twins of different industrial scenarios, aiming to significantly improve industrial management and business processes. Moreover, since modern companies should be as "smart" as possible and should adapt themselves to the varying nature of the digital supply chains, they need different mechanisms in order to (i) enhance the control of the production plant and (ii) comply with high-layer data analysis and fusion tools that can foster the most appropriate evolution of the company itself (thus lowering the risk of machine failures) by adopting a predictive approach. Focusing on the overall company management, in this chapter we present an example of a "renovation" process, based on: (i) digitization of the control quality process on multiple production lines, aiming at digitally collecting and processing information already available in the company environment; (ii) monitoring and optimization of the production planning activity through innovative approaches, aiming at extending the quantity of collected data and providing a new perspective of the overall current status of a factory; and (iii) a predictive maintenance approach, based on a set of heterogeneous analytical mechanisms to be applied to on-field data collected in different production lines, together with the integration of sensor-based data, toward a paradigm that can be denoted as Maintenance-as-a-Service (MaaS). In particular, these data are related to the operational status of production machines and the currently available warehouse supplies. Our overall goal is to show that IoT-based Industry 4.0 strategies allow to continuously collect heterogeneous Human-to-Things (H2T) and Machine-to-Machine (M2M) data, which can be used to optimize and improve a factory as a whole entity.*

1 Introduction

The wide adoption of Internet of Things (IoT) technologies has lead to a greater connectivity in industrial systems, i.e., to the paradigm of Industrial IoT (IIoT). The recent literature provides several relevant definitions for IIoT, which can be summarized as a collection of Smart Objects (SOs), cyber-physical assets, together with generic information technologies and Cloud or Edge Computing platforms allowing real-time and intelligent access, collection, analysis, and exchange of information related to processes, products or services, within the industrial environment. The main objective of IIoT is to optimize the overall production value in terms of service delivery and productivity, cost reduction, energy consumption, and the definition of the build-to-order cycle [5]. Related to IIoT, the recent concept of Industry 4.0 identifies the ongoing fourth industrial revolution focusing on the manufacturing industry scenario and can be considered as a subset of IIoT. The terms IIoT and Industry 4.0 are often used as synonyms; however, there is a difference between them.

Industry 4.0 has been initially proposed to describe the developing German economy in 2011 [30, 37] and mainly focuses on the manufacturing industry. IIoT was first introduced in 2012 as industrial Internet entailing the adoption of IoT in general industrial context (both manufacturing and non manufacturing). This definition is backed by the Industrial Internet Consortium, which was formed in 2014 with the support of Cisco, IBM, GE, Intel, and AT&T. The primary actors in Industry 4.0 are academic institutions, whereas IIoT is more business-oriented and mostly driven by private companies and some academic institutions [1]. Both IIoT and Industry 4.0 aim at making systems robust, faster, and secure, and are characterized by the extensive use of Cyber Physical Systems (CPSs), digital twins, and heterogeneous data collection.

CPSs can be considered as the core of Industry 4.0, being focused on sensors and actuators, and, through the integration of computing, communication, and control, providing dynamic control, information feedback, and real-time sensing for complex systems. Hence, CPSs allow to fulfill the dynamic requirements of industrial production and improve the effectiveness and efficiency of the entire industry. Digital twins, instead, are more focused on the definition of a physical system's digital copy, in order to perform real-time optimization: this is done by creating virtual models of physical objects in virtual space, in order to simulate their real behaviors and provide feedback [36]. As a result, being more focused on models and data, digital twins enable companies to detect and quickly predict physical issues, and optimize processes. In the end, for both CPSs and digital twins, the physical part senses and collects data, and executes decisions based on the digital part, while this latter process and analyzes, thus making decisions [17]. Nevertheless, it is important to highlight that not every system that has entities in the cyber space corresponds to a CPS, since often cyber dynamics are not just a replica of some operational variables in the digital space (e.g., in the Cloud). In the context of CPSs and digital twins, communications play a key role, as efficient information flows from physical and cyber spaces is critical.

Hence, the Industry 4.0 leverages the integration of a set of complementary technologies and paradigms, including Enterprise Resource Planning (ERP), Internet of Things (IoT), Cloud Computing, and so on. Hence, the first change in industrial scenarios introduced by the Industry 4.0 corresponds to an advanced digitization process that, in combination with Internet-based and future-oriented technologies (including, as an example, Smart Objects (SOs) deployment), favors the vision of a production factory as a modular and efficient manufacturing system, in which products control their own manufacturing process.

In Figure 1, we show the main modules of an Industry 4.0-oriented infrastructure, which relies on a set of *data sources*, including: data provided by industrial machines; data collected by externally deployed IoT devices; information manually provided by

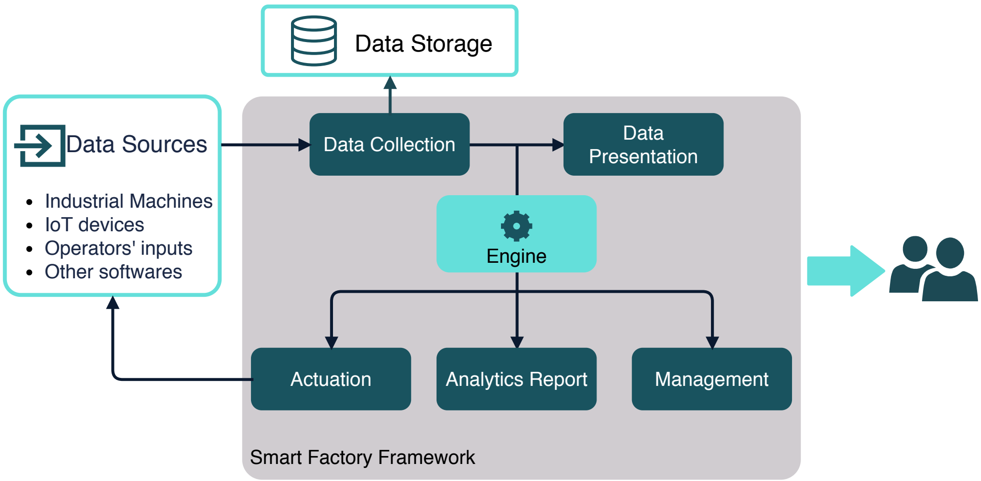


Figure 1: Industry 4.0 modules.

operators and employees in the company; and digital data imported from other third-party services, or pre-existent ERP. The framework building this Industry 4.0 scenario leverages the following main modules.

- A *data collection* module, integrating data from different sources and storing them in a coherent way, in order to efficiently perform queries and analyses.
- A *data presentation* module, devoted to the presentation of processed data to end-users, thus relying on clear and effective User Interfaces (UIs), in order to highlighting useful information.
- A customized *engine logic*, processing the collected raw data and performing different analyses in order to extract a higher knowledge level of the factory system.
- Different *high-layer interaction* modules, in which processed information can be employed to perform actuation tasks on industrial processes, as well as being presented in analytic reports or being employed to support factory employees in their activities, such as production planning, organization, optimization, and so on.

In this work, a description of the process, based on the Industry 4.0 IoT-oriented paradigm and implemented in a generic company C, in order to improve its "performance" in the overall departments, is presented. More in detail, we consider a realistic use case where company C is a successful company producing hoses, which follows high-quality standards for its products and services, and drives its constant attention to technological innovation and modern research, in order to continuously improve each stage of production and organization activities.

Even if the company C has defined a systematic and precise protocol to manage the quality of produced hoses and plan the activities on the production lines, these tasks cannot be considered *smart*, since they (i) do not involve any digitalized information and are performed through the usage of paper forms hand-written by operators, (ii) do not include any IoT-related technology to automatically collect data from industrial machines, and (iii) do not have a foundation of data that can be analyzed, in order to optimize and control the factory system has a whole. As a last consideration, the company C already relies on an ERP system, but not all processes are managed in a "smart" way.

In this work, the possible renovation steps for a company C, through the introduction of modern technologies in the business processes in both control quality and production planning tasks, are presented and discussed. The renovation process can be seen as based on the following applications: (i) the first application, denoted as *SmartFactory*, is a Web-based tool that has been developed in order to improve the production monitoring and the control quality activities; (ii) the second application corresponds to the introduction of planning capabilities aiming at supporting the production planning staff in the complex activity of scheduling the manufacturing orders on the production lines proper of the company C; and (iii) the third application relies on the introduction of an IoT-oriented infrastructure, to improve and enhance the overall activities in the company C. We then conclude discussing the introduction of predictive optimization-oriented approaches, to lower factory employees' risks and plant faults.

The rest of this work is organized as follows. In Section 2, a brief analysis on the context of Industry 4.0 is given, while in Section 3 a set of guidelines for the digitization of a target factory are presented. Section 4 is devoted to the analysis of different approaches to be applied for the monitoring of production lines and machines, while in Section 5 the positive impact of predictive optimization is discussed. Finally, in Section 6 conclusions are drawn.

2 Related Works and Motivations

The Industry 4.0 concept has recently gained particular attention, as it encompasses a heterogeneous set of research fields, being closely related not only to IoT, CPS, Information and Communications Technologies (ICT), and Cloud Computing, but also to Enterprise Architecture (EA) and Enterprise Integration (EI).

The work in [21] represents one of the first review on the content, scope, and findings of Industry 4.0 in ICT-oriented scenarios. In [21], the authors identify 5 main research categories: (i) concept and perspectives of Industry 4.0; (ii) CPS-based Industry 4.0; (iii) interoperability of Industry 4.0; (iv) key technologies of Industry 4.0; and (v) smart factory and manufacturing. In [24], it is highlighted how the benefits brought by Industry 4.0 are not available only to large companies, but they are accessible and attractive also for Small and Medium Enterprises (SMEs). More in detail, the authors of [21] adopt the definition provided in [16], considering Industry 4.0 as *"a new approach for controlling production processes by providing real-time synchronization of flows and by enabling the unitary and customized fabrication of products."* Finally, the authors conclude that applications are mostly related to monitoring of production processes and to the improvement of current capabilities and flexibility, through the introduction of new technologies, such as Cloud Computing and Radio-Frequency IDentification (RFID). However, at the same time, most of the possible opportunities (e.g., CPS, Machine-to-Machine (M2M), Big Data, or collaborative robots) are under-exploited, if not ignored, by SMEs.

Another key point is that the evolution process transforming a traditional company into a "smart industry" is generally smooth. One of the first steps relates to the digitization (or digital transformation process) which has been identified as one of the major trends changing society and business. Digitization, in fact, leads to changes in the companies in both organizational and operational environments through the introduction of new technologies. In [27, 45], it is highlighted how the changes introduced by the digitization cover different levels in a factory, such as: (i) the process level, in which processes are optimized reducing manual steps and adopting new digital tools; (ii) the organizational level, where obsolete practices are discarded and new services are integrated; (iii) the business domain level, in which value chains and roles inside ecosystems are changed; and (iv) the societal level (e.g., changing type of work). Moreover, in [14, 28], it is shown that replacing paper and manual processes with software-based solutions allows to automatically and quickly collect data that can be adopted to better understand the risk causes and the process performance. Finally, in [15, 40] the authors highlight the importance of User Interfaces (UIs) which a digitization process has to be equipped with, where real-time reports and dashboards on digital process performance allow managers to address problems before they become critical.

The current advancements in IoT, together with the development of new cost-effective and high-performance wireless communication systems, allow to connect devices and objects, thus giving them the possibility to share information related to the surrounding environment. This enables the creation of effective CPSs which can continuously monitor and control the environment in the industrial domain. Therefore, the second phase that can transform a traditional company into a smart industry is the introduction of new IoT technologies, in order to collect data inside the factory and monitor processes. This paradigm is also denoted as *Industrial IoT* (IIoT), which an example of is provided in [12], where it is highlighted how the general assessment of the machine operational condition is crucial for a smart and efficient industrial processes management. In [12], a prognostic approach to the detection of incipient faults of rotating machines, by means of their vibrational status monitoring, is proposed. Another IIoT-based solution is described in [8], where a particular mechanism, specifically designed to enable a pervasive monitoring of industrial machinery through battery-powered IoT sensing devices, is presented: the industrial scenario covered a time period of two months and was based on thirty-three IoT sensing devices performing advanced temperature and vibration monitoring tasks, while evaluating transmission delays and system operating life time through power consumption measures. The adopted IoT protocols guarantee that each node is reachable through IP addressing with an acceptable delay.

Another IIoT example is proposed in [23], in which the application of Low-Power Wide-Area Networks (LPWANs) in an industrial scenario is proposed. More in detail, the authors focus their work on the open LoRaWAN network standard [34], thus proposing a comparison with the IEEE 802.15.4 network protocol, which is another IoT protocol widely adopted in the industrial context. The authors conclude that LoRaWAN represents a strongly viable opportunity, providing high reliability and timeliness, while ensuring very low energy consumption.

After the digitization and monitoring through IIoT technologies phases, the last step for a smart industry is related to optimization of the company processes, leveraging the analysis of the collected information. In [29], the authors observe how the widespread adoption of IoT technologies is enabling a faster and more informative sensing, generating data abundance, more than ever. At the same time, technology advances provide also the computational resources needed to process this large amount of data, transforming them into actionable information in a reasonably short time. A critical overview of trends characterizing the industrial process monitoring activity since its appearance (almost 100 years ago) is provided, showing how this task has changed, from simple statistical analysis, to detection and, finally, to diagnosis and prognosis.

3 Digitization

3.1 Organization of the Company C

In order to discuss the evolution of a traditional company to an Industry 4.0-organized smart company, consider that the production activity of a generic company C is divided into N departments, each performing a specific production activity. Each produced article crosses consecutive departments, where semi-finished goods are manufactured with different machines' configurations, among the production processes of several articles. Moreover, each department has a variable number of lines L_{N} that can work simultaneously.

The three main classes of actors operating in the company C and involved in the activities are the following.

- **Production Scheduling Managers (PSMs)**: they are in charge of controlling and organizing the production schedule of all lines available in the factory, taking into account the stock policy and the commissions placed from customers.
- **Quality Inspectors (QIs)**: they are responsible to perform continuous checks on products and semi-finished products directly on the production lines in the factory, in order to guarantee conformity to quality standards. Then, QIs follow the schedule defined by PSMs and move between lines, in order to inspect the production process.
- **Line Operators (LOs)**: they are responsible for preparing and activating the production machines on the production lines, following the schedule. During the production, LOs take measurement to monitor the production, thus also performing quality checks on the hoses, which are then validated by QIs.

Moreover, consider that the company C has defined a precise protocol to follow in the hoses production process. More in detail, after the definition of the schedule for each line, performed by PSMs, each article to be produced is separated into a set of D Manufacturing Orders (MOs), one per department involved in the productive process. Each MO is then assigned to a specific line in the factory. In this way, a LO working on a production line is guaranteed to have a daily activity (namely, a list of MOs) to perform on the line which he/she is responsible for. The manufacturing monitoring task of the company C is performed through the use of MO Forms (MOFs) printed on paper sheets. Each department in C has a specific MOF layout, in terms of input information types and number of sections. Some of the sections are descriptive, aiming at showing some important information for the manufacturing process (i.e., configurations, measures, customizations); other sections are instead input sections that should be filled by LOs or QIs during the production with department-specific data or measurements. The possible

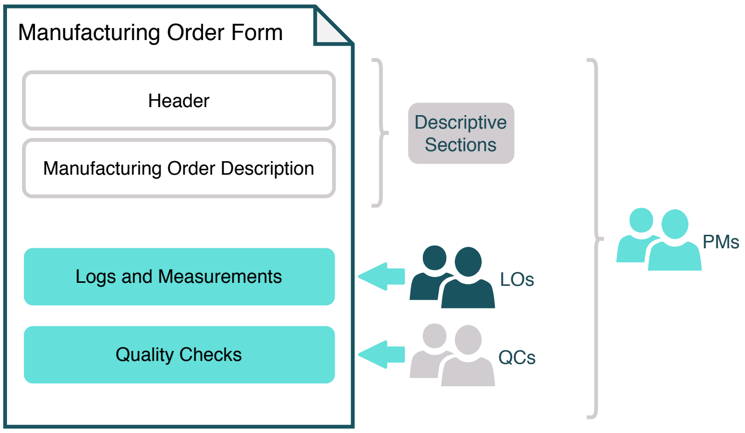


Figure 2: MOFs structure.

section types that can be identified for the production quality control of the company C are described in the following and depicted in Figure 2.

The *Header* section simply identifies the MOF type, among the available ones, through the title together with a start date and an order alphanumeric code. The *Manufacturing Order Description* section is another part containing a list of textual information describing the steps needed for order completion and machine configuration. The *Log and Measurements* is the part that has to be filled by LOs with the quantity of produced products, the quantity of raw materials employed on the line, the development status of a order (if it requires multiple work shifts) and all other measurements that are required during the production. The *Quality Checks* section contains the results of data surveys performed during the production process, in order to verify that the products comply with the required standards. This section can have a different layout depending on the MOF type, as different production articles require specific quality checks. This section is periodically filled out by QIs. The protocol adopted by the company C in order to monitor the productive process is based on the following daily steps.

1. Each morning, the PSM staff prints the MOFs corresponding to the MOs planned for the current work day and distributed to all factory departments, on each department's production line.

2. Each LO starts his/her work and, during the shift, fills out all MOFs received by his/her department supervisor.

3. Each QI starts his/her work and, moving among the lines, he/she periodically completes the quality section of the MOFs.

4. In the evening, each department supervisor collects the MOFs and delivers them to the PSM staff. Finally, all paper sheets are manually scanned and stored as digital images, where inputs are only hand-written.

The production planning of the company C is an extremely important and time-consuming task, as it strongly affects both results and performance of the subsequent activities. The planning, on one side, should fulfill the customer requirements and, on the other side, should try to efficiently use both machines and human resources on the lines. This activity, in the company C, is performed by the PSM staff and managed through the use of virtual spreadsheets, with different layouts and rules depending on the specific department. An example of structure spreadsheet is shown in [2].

A manual process for the planning activity can be extremely complicated and time-consuming for the PSMs, since it requires to take into account several factors (such as customers' orders, machine configurations and delay, employees availability), and is not exempt from possible errors during data insertion. A complete description of aspects related to digitization of control quality and planning processes in a real company are provided in [2]—in the following subsection, only some aspects are highlighted abiding by a general perspective.

3.2 Digital Quality Control

In order to replace the use of paper-based MOFs, the company C has integrated in its workflow a smart Web-based application, here denoted as *SmartFactory*, allowing to:

- collect data, related to the production, in a fast and easy way;
- get rid of costs related to the print-scanning process;
- facilitate the work of QIs and LOs in the different departments, through the adoption of both mobile devices (e.g., tablets) and PCs;
- efficiently manage updates and changes in the production process;
- save digital data through integration with the company IT and ERP systems.

The SmartFactory application, after a login process authenticating and authorizing users, should present different views for PSMs, QCs and LOs. This login phase can leverage on pre-existent technologies, like the operators' personal Near Field Communication (NFC) badges, if they are already employed in the company C to access to the buildings. After the login phase, SmartFactory can redirect the user to a separate home page, customized depending on his/her role with a custom data visibility and privileges on different modules (namely, Read-Only, RO, and Read-Write, RW). As an example, LOs can write on the Logs and Measurements section, and have RO privileges on the Quality Checks section.

Another essential feature required by the process of digitization of company C, is to simply find data of interest. For this reason, SmartFactory should include a research functionality for both QIs and PSMs allowing them to find, in real-time, data related to orders (both in production and historical). Another important functionality is the possibility to show orders requiring a supervision from the PSMs, in a separate and specific application view.

3.3 Digital Planning Management

The planning activity is generally assigned to highly qualified staff, with a deep knowledge of all mechanisms regulating the whole company workflow, from provision office to sales department. Since a second stage of the digitization process in a generic company C should include the production planning department, with the aim of replacing the spreadsheet-based method, the SmartFactory application should include a Web-based planning tool allowing to: (i) simplify and speed up the planning process for PMs; (ii) hide the complexity behind planning allocation calculations; and (iii) avoid planning errors. Moreover, this SmartFactory application's module is only used by PSMs through a PC and is integrated with the company IT and ERP systems through a software extension able to store all data related to the planning activity, which are not already registered. More in detail, the planning tool can include several modules. The *Scheduling Suggestions Module* is responsible for accessing the ERP system and retrieving, for the PSMs, a list of articles to be produced, thus representing the starting point onto which work. The *Shifts Manager Module* allows to manage all information related to LOs' shifts, such as the hours availability in the department of interest. Then, the *Planning Events Module* manages the Graphical User Interface (GUI) replacing the virtual spreadsheet used by PSMs, and allowing them to *drag-and-drop* manufacturing orders on a calendar-based view, in which columns show the working days and rows represent the production lines in the department subject to the planning activity (hence with representation similar to the spreadsheets ones). The SmartFactory application is also responsible for calculating the real duration of a MO event once it is placed on a specific cell. This calculation is performed considering: (i) the requested length of the hose; (ii) the production line's velocity, retrieved analyzing historical data from the ERP; (iii) the shift duration planned for the specific day; and (iv) configurations and setup delays. Finally, the *Production Manager Module* stores data inserted by PSMs and manages the interaction with the company C's ERP system, in order to move planned events to the production system, making digital MOs visible also in the quality control part of the SmartFactory application.

3.4 Main Advantages

The overall architecture of the SmartFactory application, with reference to the modules previously highlighted and their interactions, is shown in Figure 3.

It can be easily estimated that a first (and tangible) effect of the introduction of the SmartFactory application in a generic company C can be related to time saving aspects, due to the fact that forms are no longer printed on paper sheets nor manually scanned. Even the way to insert data from different actors may be certainly simplified, since SmartFactory allows a simultaneous access to MOFs, permitting QIs, LOs, and PSMs to input and view data on the same production order without interfering with each other. Another important aspect to be considered is the quantity of structured data, made available by the adopted digitization revolution, that are continuously collected and can be employed to monitor the status of the production in real-time, but also constitute a basis for further analysis. The last advantage is related to the reduction of errors and faulty products, as non-regular behaviors can be detected and recorded in real-time for further performance analysis.

Furthermore, with regard to the planning functionality that can be introduced in the SmartFactory application, the first advantage relates to the reduction of the number of hours directly spent by PSMs for the MOs' scheduling activity, but it is important to consider also the reduction of time needed to train a new person dedicated to planning activities. In fact, having the SmartFactory application hiding all aspects related to MOs'

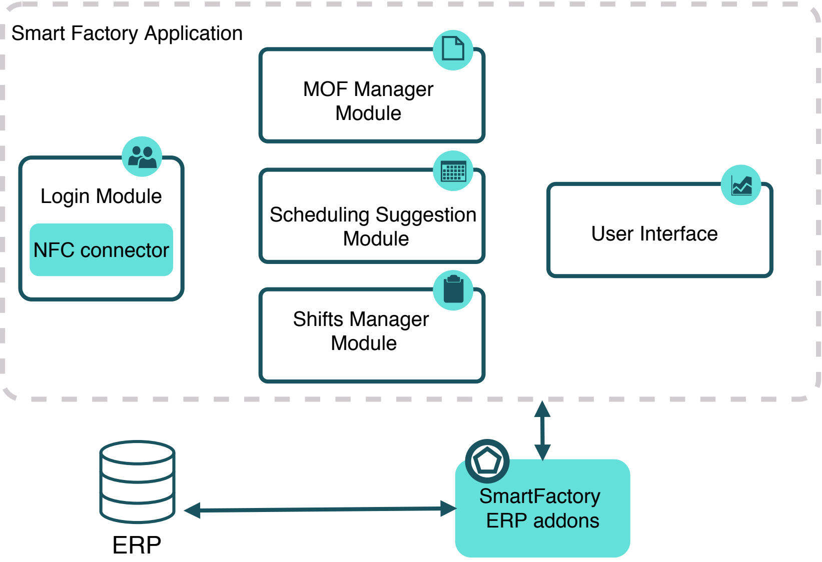


Figure 3: SmartFactory application overview.

duration calculations and providing the user with all required information in single view, a strong simplification of the work of PSMs is obtained, making possible for the company to assign this activity also to other people, with a minimized learning phase. Finally, the complete digitization and automation of the planning process allows to make the complete company plan available to all interested users.

4 Monitoring

On the basis of the concepts highlighted in Section 3, it has been clarified how a generic company C may benefit from a proper planning and quality control methodology, in order to improve its productivity and lower downtime and errors in resource allocation [44]. Looking at the company C as a *step-by-step* production chain, it is clear that data, on the basis of which planning is performed, should be as accurate and delivered *on-time* as possible.

This can be carried out in several ways. A first approach consists in extending the functionalities of the software tools described in Section 3, in order to create a framework of cooperating tools which also includes other aspects of the company C's management. More in detail, the proposed planning tool can be integrated with a more general Customer Relationship Management (CRM) tool, aiming at managing information related to customers orders and due dates, products' stock levels, and planned activities, as shown in Figure 4. As a second step, the company C's software framework can be extended with a Supply Chain Management (SCM) tool, allowing operators to have a clear representation of the status of the working supply chain, through the use of a dashboard showing running activities, production delays, warnings, and alarms related to working

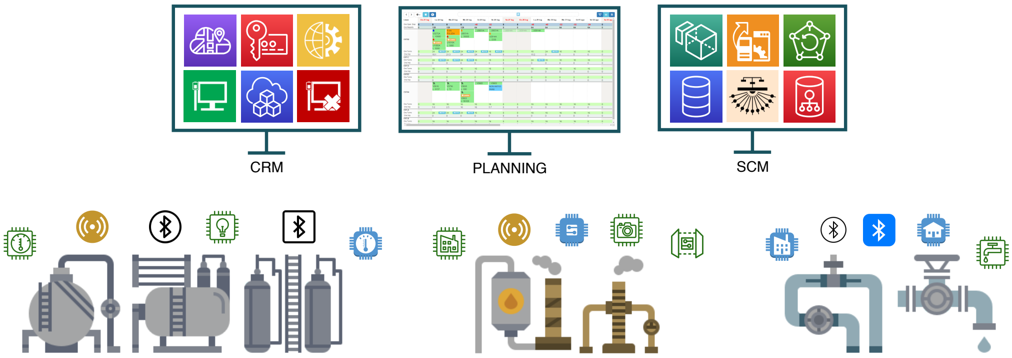


Figure 4: Industry 4.0 monitoring with sensors on production machines, and planning and reporting dashboards.

machinery. This framework is generally built on top of a set of microservices, with its effectiveness strictly related to the quality of information collected from the company environment. This entails the introduction of IoT technologies, with the deployment of sensor networks—preferably Wireless Sensor Networks (WSNs) [19]—directly inside the factory environments and around each production machine, even involving different communication and processing technologies, in order to cover different needs [20].

In the case of a production machine characterized, for its internal manufacturing, by vibrations of some kind, it can be possible to equip the machine itself with some "sensing node" composed by a sensing element (e.g., a vibration sensor) connected with a 1-wire link to a "core" module, in charge of processing the incoming data (either analog or digital) and of doing additional tasks based on data themselves, such as sending the data to an upper-layer data collector, as well as internally storing the collected data for further analysis and as a safe backup. Even though this approach is general, the communication paradigm adopted for the harvested data forwarded to upper-layer systems may be addressed based on the specific characteristics representing the industrial environment in which the production machine is placed, as shown in Figure 4. In other words, this sensing module can forward its collected data through an IEEE 802.3 (Ethernet) connection, if this protocol is available and useful on the production line, as well as taking advantage of the availability of an IEEE 802.11 (Wi-Fi) connection, which several sensing nodes can attach to and participate as IEEE 802.11 clients.

As can be easily understood, a sensing node can be equipped with various types of sensors, able to detect different situations. An example can be a camera-based monitoring system, in which the sensor is video and its goal is to monitor a particular part of the production machine, as shown in Figure 5. In this scenario, the monitoring device provides several degrees of freedom, meaning that it can be customized with different setups and configurations (e.g., different types of cameras) based on the task that the IoT node should perform. In case the IoT system has to monitor with a certain accuracy a specific (and limited in space) region of the overall environment, then the camera should be chosen with certain characteristics (e.g., a High Quality (HQ) camera with an adequate frame ratio). If, instead, the IoT camera has to monitor with lower accuracy, then it can be a Low Quality (LQ) camera, thus lowering the overall price of the Industry 4.0 monitoring node.

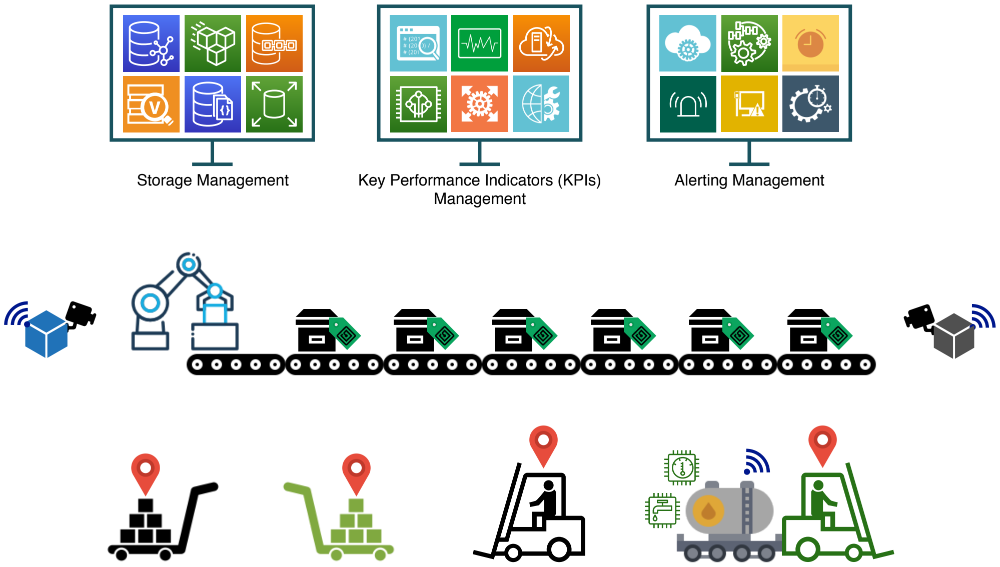


Figure 5: Industry 4.0 monitoring with camera sensors, vehicle positioning, and performance dashboards.

Obviously, these considerations certainly affect the core component of the sensing device, since it has to be defined on the basis of the need of analyzing a video stream, as well as a single frame taken by the camera. Moreover, in the case of a precise monitoring task, the processing element should perform proper processing, thus requiring a high amount of volatile memory (e.g., RAM), as well as a suitable Operating System, OS (possibly a real-time OS, RTOS). If monitoring should be less accurate, then processing can be carried out with more commercial (and less specialized) hardware.

Another way to monitor a machine involved in an Industry 4.0-oriented renovation process can be the definition and deployment of a network of IoT nodes organized with a proper topology, targeting the optimization of the communications inside the network itself and the reachability of the information from outside the network. An illustrative case study is given by a monitoring network composed by sensing nodes involving an *on-field* sensor, harvesting data from both the machine and the environment (as shown in Figure 4), and a processing module enabling the overall device to join and participate to a Bluetooth Low Energy (BLE)-based network [13]. Looking at topological level, in this scenario each sensing node acts as a BLE slave, while in the production plant a BLE master should be elected to own and manage the BLE network and, in turn, to collect data from its slaves. The BLE network topology can be selected depending on the needs and environment.

Another monitoring activity that an Industry 4.0-targeting company may consider is to exploit the possibilities "natively provided" by its production machines. The most relevant one is to access the Programmable Logic Controllers (PLCs) [3] managing the production machines, in order to collect (hopefully in a real-time mode) the production data exposed by the machine itself. This can be performed equipping each machine with a properly made "collector" node, composed of (i) a processing entity, in charge of collecting, storing, and forwarding *on-field* collected data to external—and high-layer—Supervisory Control and Data Acquisition (SCADA) systems [4]; and (ii) an *on-field* network interface, able to directly talk with the production machine. In relation to the latter component, the evolution that has taken place in the field of real-time production data collection, allows the company C to extend its *on-field* interfaces to support multiple communication protocols, ranging from ModBus (either through RS-485 or RS-232 cables) to Ethernet and CANBus [18]. Otherwise, if the data generated from the production machine needs to be restricted to a small amount of bytes (e.g., because they represent only an aggregated metric processed a few times in a certain time period), then the generic company C can consider to use some properly M2M-defined SIM cards, generally enabled to support a fixed traffic amount and low speed, but acceptable for this kind of industrial scenarios. With regard to wired technologies (such as ModBus, CANBus, and Industrial Ethernet), it is worth to specify that ModBus can be used as both fieldbus- and controller-level protocol, meanwhile CANBus is usually employed as fieldbus protocol only [39]. Moreover, both protocols work at application layer, leaving the possibility to adopt different low-level physical protocols (e.g., wired technologies, such as RS485 [41], but also wireless ones, such as the uprising 5G [25, 22]) for transmitting information on the field.

It is important to underline that M2M is related not only to cellular communications, but it generally identifies the interaction between heterogeneous smart devices, thus being one of the IoT's pillars. M2M can enable machines in the company C to exchange messages to each other, in order to achieve a predefined objective, to provide a specific service, or to complete a task [11, 6]. Abiding by this paradigm on monitoring devices in the company C's environment and making communication flows to converge in "collector" nodes, allows to create a M2M services layer, which can interact with SCM and planning tools. This can be useful in: (i) identifying problems, errors or breakdowns to be reported to operators; (ii) providing a real-time monitoring of company machinery status; and (iii) reducing the quantity of input required to operators working on production lines (e.g., during quality control activities).

Finally, for the company C it can be interesting to collect, in addiction to M2M-related data from the production machines, also data related to the interactions between employees and machines and, in general, with the industrial environment, in a Human-to-Machine (H2M)-oriented way. In this scenario, one could consider the introduction of localization-aware infrastructures for the sake of safety and security of the employees in their work environment. In detail, it can be advisable the adoption of *on-board* precise tracking technologies (e.g., based on Ultra-WideBand (UWB) technology [33]) on industrial vehicles which are in charge of moving materials (e.g., from warehouse to production lines, as well as among production plants). In this way, as shown in Figure 5, the resulting benefit is two-fold: on one hand, it is possible to plan vehicles' trajectories, thus better managing the good movements; on the other hand, workers can be aware of their surrounding vehicles' positions, thus increasing their safety and security. In order to improve these risk avoidance measures, the company C can define the adoption of additional sensors (e.g., proximity sensors) directly on the vehicles, thus reducing even further (and, hopefully, completely avoiding) accidents involving humans and/or fixed obstacles (e.g., shelving).

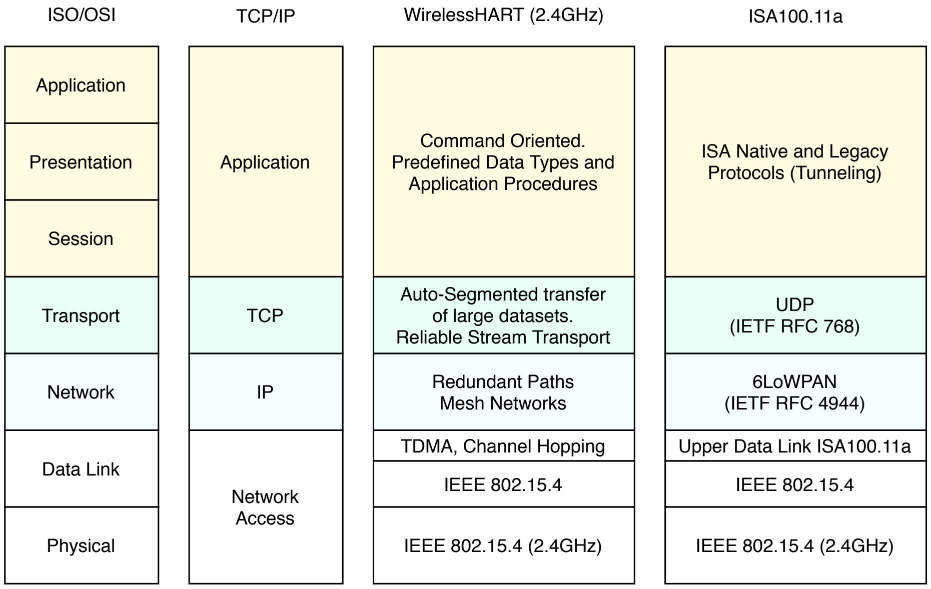
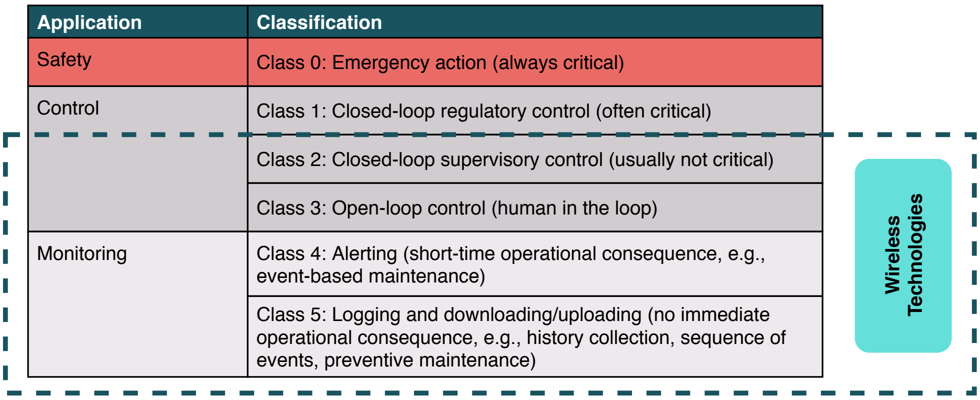


Figure 6: Layered stacks view of WirelessHART and ISA100.11a protocols.

As a conclusion, it is quite clear how, among all cited communication technologies, the wireless ones represent an essential business enabler for the industrial world, because of their reliability, fast deployment, flexibility, cost effectiveness, and capacity to be adequate in (i) pulling data from deployed devices and (ii) sending supervisory control commands to working machinery (e.g., open/close to a valve and start/stop to an actuator) [38]. To this end, the evolving SCADA technology continues to take advantage of emerging technologies at different layers, with the drawback of deploying various heterogeneous and fragmented wireless platforms. This is also due to a limited number of certified wireless instrumentation devices complying with WirelessHART or ISA100.11a [26, 10] specifications, whose layered protocol structures are shown in Figure 6 and directly compared with the 7-layer Open System Interconnect (OSI) model and TCP/IP protocol stack.

On the basis of their coverage area, industrial wireless technologies can be classified into three main categories: (i) WSNs, including ISA100.11a, WirelessHART, ZigBee, and IPv6 over Low-power Wireless Personal Area Network (6LoWPAN); (ii) backbone networks, dominated by IEEE 802.11a/b/g/n/ac protocol; and (iii) backhaul networks, involving Ultra High Frequency (UHF) radio and evolving toward 4G Long-Term Evolution (LTE)/5G, satellite, and microwave technologies. Referring to the ISA100 standard, wireless applications can be grouped into three classes: monitoring, control, and safety. As shown in Figure 7, wireless technologies should be used for noncritical control (Class 2 and Class 3) and monitoring (Class 4 and Class 5) applications, while safety applications, devoted to always-critical emergency situations, should be handled by certainly-available and reliable wired technologies.



Application	Classification
Safety	Class 0: Emergency action (always critical)
Control	Class 1: Closed-loop regulatory control (often critical)
	Class 2: Closed-loop supervisory control (usually not critical)
	Class 3: Open-loop control (human in the loop)
Monitoring	Class 4: Alerting (short-time operational consequence, e.g., event-based maintenance)
	Class 5: Logging and downloading/uploading (no immediate operational consequence, e.g., history collection, sequence of events, preventive maintenance)

Figure 7: Wireless application classification.

5 Predictive Optimization

The techniques previously shown in Section 3 and Section 4 can be used by the company C to collect data from different (and heterogeneous) sources, targeting a more accurate overall situation monitoring of the company itself. As widely known, in the last years there is an increasing interest on the use of the large amount of data which can be collected in these scenarios, in order to extract relevant information. As shown in Figure 8, most companies have only recently started to take advantage of the possibilities introduced by both *on-premise* and *far-from-home* storage and processing mechanisms (e.g., Edge Computing, Fog Computing, Cloud Computing) for their large amount of collected data [31, 42].

Focusing on digitization activities, the quality control and planning tasks described in Section 3 can be enhanced giving to the technical departments more precise information that can help the staff in improving the planning of the activities to be performed inside the factory. These activities include, for example, the estimation of the warehouse stocks utilization over a certain time period (as well as on multiple time periods, for the sake of comparison), either predicting how a specific material employed during the production of hoses will be entirely consumed, as well as highlighting possible misuses or excessive utilization. The final goal is to optimize the use of resources, avoiding, or at least reducing, the waste of materials (which also has a relevant financial impact for the company C).

The data collected from the production machines can be exploited to forecast the productivity level of a certain product over a particular time interval. This, for sure, may help to optimize the utilization of materials required by an employee from the warehouse. The analysis of the *on-field* production data allows to optimize the maintenance of the production machines. In detail, the ability to collect data from production machines allows to timely perform specific part replacement before its actual fault. This can happen in two ways: (i) the part will be replaced in a *preemptive* way [7], knowing its exact lifetime; or (ii) the replacement will happen following a *predictive* approach [35, 9], based on a more accurate information processing and analysis, involving a more comprehensive dataset (possibly composed by heterogeneous data regarding different aspects of

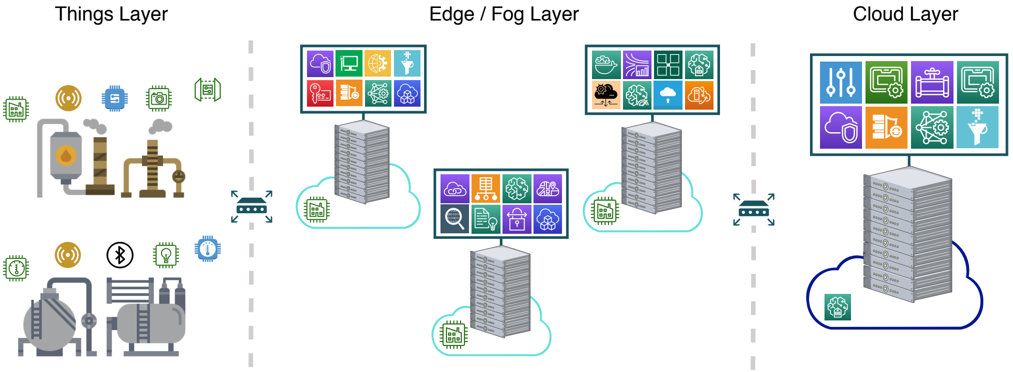


Figure 8: Industry 4.0 predictive optimization to be performed both *on-premise* and *far-from-home*.

the production machine) [32]. Both these maintenance strategies lead to a new concept, that can be denoted as Maintenance-as-a-Service (MaaS), in which the focus moves from sparse data to their aggregated meaning [46, 43]. This data aggregation and processing can be performed either inside the company (namely, company C) or by involving upper-layer systems, relying on the already recalled Edge Computing/Fog Computing/Cloud Computing paradigms.

6 Conclusions

In this work, we have presented an overview on how a generic company C can enhance its internal organization, from a traditional one to an IIoT-based one. Our experience shows that the best way for companies (especially for SMEs) to deal with this process is to face it gradually, integrating *step-by-step* the required set of technologies and know-how. More in detail, in order to drive a generic company C to become IIoT-oriented, the following three transformation stages have been identified: (i) *digitization* stage, aiming at digitally collecting and processing information already available in the company environment; (ii) *monitoring* stage, aimed at extending the quantity of collected data and providing a new perspective of the overall current status of a factory; and (iii) *prediction* stage, involving a set of heterogeneous analytical mechanisms to process the collected data. The last stage has the goal of highlighting problems, errors, or inefficiencies in the company environment, while, at the same time, performing actions on the company C itself through revisions or corrections on the planned tasks.

Acknowledgments

The work of Luca Davoli is partially funded by the University of Parma, under “Iniziative di Sostegno alla Ricerca di Ateneo” program, “Multi-interface IoT sYstems for Multi-layer Information Processing” (MIoTYMIP) project.

References

[1] Mohammad Aazam, Sherali Zeadally, and Khaled A Harras. “Deploying fog computing in industrial internet of things and industry 4.0”. In: *IEEE Transactions on Industrial Informatics* 14.10 (2018), pp. 4674–4682. DOI: `10.1109/TII.2018.2855198`.

[2] Laura Belli et al. “Toward Industry 4.0 with IoT: Optimizing Business Processes in an Evolving Manufacturing Factory”. In: *Frontiers in ICT* 6 (2019), p. 17. ISSN: 2297-198X. DOI: `10.3389/fict.2019.00017`.

[3] William Bolton. *Programmable Logic Controllers.* 4th ed. Newnes, 2015. 292 pp. ISBN: 978-0-7506-8112-4.

[4] Stuart A Boyer. *SCADA: Supervisory Control And Data Acquisition.* 4th ed. International Society of Automation, 2009. 257 pp. ISBN: 978-1-9360-0709-7.

[5] Hugh Boyes et al. “The industrial internet of things (IIoT): An analysis framework”. In: *Computers in Industry* 101 (2018), pp. 1–12.

[6] Jim Brodie Brazell et al. *M2M: the wireless revolution.* Tech. rep. 2005.

[7] Donato Catenazzo, Brendan O’Flynn, and Michael Walsh. “On the Use of Wireless Sensor Networks in Preventative Maintenance for Industry 4.0”. In: *2018 12th International Conference on Sensing Technology (ICST).* Limerick, Ireland, Dec. 2018, pp. 256–262. DOI: `10.1109/ICSensT.2018.8603669`.

[8] Federico Civerchia et al. “Industrial Internet of Things Monitoring Solution for Advanced Predictive Maintenance Applications”. In: *Journal of Industrial Information Integration* 7 (2017), pp. 4–12. ISSN: 2452-414X. DOI: `10.1016/j.jii.2017.02.003`.

[9] Michele Compare, Piero Baraldi, and Enrico Zio. “Challenges to IoT-enabled Predictive Maintenance for Industry 4.0”. In: *IEEE Internet of Things Journal* (2019), pp. 1–1. ISSN: 232-2541. DOI: `10.1109/JIOT.2019.2957029`.

[10] Marcio S. Costa and Jorge L. M. Amaral. *Analysis of wireless industrial automation standards: ISA-100.11a and WirelessHART.* https://tinyurl.com/isa100whart. Accessed: 2020-04-05. Nov. 2012.

[11] Mathias Santos De Brito et al. “Towards programmable fog nodes in smart factories”. In: *2016 IEEE International Workshops on Foundations and Applications of Self* Systems (FAS* W).* IEEE. Augsburg, Germany, 2016, pp. 236–241. DOI: `10.1109/FAS-W.2016.57`.

[12] Giuseppe Dinardo, Laura Fabbiano, and Gaetano Vacca. “A Smart and Intuitive Machine Condition Monitoring in the Industry 4.0 Scenario”. In: *Measurement* 126 (2018), pp. 1–12. ISSN: 0263-2241. DOI: `10.1016/j.measurement.2018.05.041`.

[13] Carles Gomez, Joaquim Oller, and Josep Paradells. “Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology”. In: *Sensors* 12.9 (2012), pp. 11734–11753. ISSN: 1424-8220. DOI: `10.3390/s120911734`.

[14] Mark Huberty. “Awaiting the Second Big Data Revolution: From Digital Noise to Value Creation”. In: *Journal of Industry, Competition and Trade* 15.1 (Mar. 2015), pp. 35–47. ISSN: 1573-7012. DOI: `10.1007/s10842-014-0190-4`.

[15] Marika Miriam Iivari et al. "Toward Ecosystemic Business Models in the Context of Industrial Internet". In: *Journal of Business Models* 4.2 (2016), pp. 42–59. ISSN: 2246-2465. DOI: 10.5278/ojs.jbm.v4i2.1624.

[16] Dorothée Kohler and Jean-Daniel Weisz. *Industrie 4.0: Les Défis de la Transformation Numérique du Modèle Industriel Allemand*. La Documentation française Paris, 2016. 176 pp. ISBN: 978-2-1101-0210-2.

[17] C. Koulamas and A. Kalogeras. "Cyber-Physical Systems and Digital Twins in the Industrial Internet of Things". In: *Computer* 51.11 (Nov. 2018), pp. 95–98. ISSN: 1558-0814. DOI: 10.1109/MC.2018.2876181.

[18] Wolfhard Lawrenz. *CAN System Engineering. From Theory to Practical Applications*. 2nd ed. Springer-Verlag London, 2013. 353 pp. ISBN: 978-1-4471-5612-3.

[19] Xiaomin Li et al. "A Review of Industrial Wireless Networks in the Context of Industry 4.0". In: *Wireless Networks* 23.1 (Jan. 2017), pp. 23–41. ISSN: 1572-8196. DOI: 10.1007/s11276-015-1133-7.

[20] Chun-Cheng Lin et al. "Key Design of Driving Industry 4.0: Joint Energy-Efficient Deployment and Scheduling in Group-Based Industrial Wireless Sensor Networks". In: *IEEE Communications Magazine* 54.10 (Oct. 2016), pp. 46–52. ISSN: 1558-1896. DOI: 10.1109/MCOM.2016.7588228.

[21] Yang Lu. "Industry 4.0: A Survey on Technologies, Applications and Open Research Issues". In: *Journal of Industrial Information Integration* 6 (2017), pp. 1–10. ISSN: 2452-414X. DOI: 10.1016/j.jii.2017.04.005.

[22] M. Luvisotto et al. "Real-time Wireless Extensions of Industrial Ethernet Networks". In: *2017 IEEE International Conference on Industrial Informatics (INDIN)*. Emden, Germany, Nov. 2017, pp. 363–368. DOI: 10.1109/INDIN.2017.8104799.

[23] Michele Luvisotto et al. "On the Use of LoRaWAN for Indoor Industrial IoT Applications". In: *Wireless Communications and Mobile Computing* 2018 (2018). ISSN: 1530-8669. DOI: 10.1155/2018/3982646.

[24] Alexandre Moeuf et al. "The Industrial Management of SMEs in the Era of Industry 4.0". In: *International Journal of Production Research* 56.3 (2018), pp. 1118–1136. DOI: 10.1080/00207543.2017.1372647.

[25] A. Neumann et al. "Towards Integration of Industrial Ethernet with 5G Mobile Networks". In: *2018 IEEE International Workshop on Factory Communication Systems (WFCS)*. Imperia, Italy, June 2018, pp. 1–4. DOI: 10.1109/WFCS.2018.8402373.

[26] Mark Nixon. *A Comparison of WirelessHART and ISA100.11a*. Tech. rep. Accessed: 2020-04-04. Emerson Process Management, Sept. 2012, pp. 1–39. URL: https://www.emerson.com/documents/automation/white-paper-a-comparison-of-wirelesshart-isa100-11a-en-42598.pdf.

[27] Päivi Parviainen et al. "Tackling the Digitalisation Challenge: How to Benefit from Digitalisation in Practice". In: *International Journal of Information Systems and Project Management* 5.1 (2017), pp. 63–77. ISSN: 2182-7788. DOI: 10.12821/ijispm050104.

[28] Sarah Quinton and Lyndon Simkin. "The Digital Journey: Reflected Learnings and Emerging Challenges". In: *International Journal of Management Reviews* 19.4 (2016), pp. 455–472. DOI: 10.1111/ijmr.12104.

[29] Marco S. Reis and Geert Gins. "Industrial Process Monitoring in the Big Data-Industry 4.0 Era: from Detection, to Diagnosis, to Prognosis". In: *Processes* 5.3 (2017), p. 35. ISSN: 2227-9717. DOI: 10.3390/pr5030035.

[30] Vasja Roblek, Maja Meško, and Alojz Krapež. "A Complex View of Industry 4.0". In: *SAGE Open* 6.2 (2016), pp. 1–11. DOI: 10.1177/2158244016653987.

[31] Radhya Sahal, John G. Breslin, and Muhammad Intizar Ali. "Big Data and Stream Processing Platforms for Industry 4.0 Requirements Mapping for a Predictive Maintenance Use Case". In: *Journal of Manufacturing Systems* 54 (2020), pp. 138–151. ISSN: 0278-6125. DOI: 10.1016/j.jmsy.2019.11.004.

[32] Erim Sezer et al. "An Industry 4.0-Enabled Low Cost Predictive Maintenance Approach for SMEs". In: *2018 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC)*. Stuttgart, Germany, June 2018, pp. 1–8. DOI: 10.1109/ICE.2018.8436307.

[33] Kazimierz Siwiak. "Ultra-Wide Band Radio: Introducing a New Technology". In: *IEEE VTS Vehicular Technology Conference, Spring 2001. Proceedings (Cat. No. 01CH37202)*. Vol. 2. Rhodes, Greece, Greece, May 2001, pp. 1088–1093. DOI: 10.1109/VETECS.2001.944546.

[34] Nicholas Sornin et al. *LoRaWAN Specification*. https://tinyurl.com/loraspecv10. Accessed: 2019-01-27. Jan. 2015.

[35] Lukas Spendla et al. "Concept of Predictive Maintenance of Production Systems in Accordance with Industry 4.0". In: *2017 IEEE International Symposium on Applied Machine Intelligence and Informatics (SAMI)*. Herl'any, Slovakia, Jan. 2017, pp. 405–410. DOI: 10.1109/SAMI.2017.7880343.

[36] Fei Tao et al. "Digital Twins and Cyber–Physical Systems toward Smart Manufacturing and Industry 4.0: Correlation and Comparison"". In: *Engineering* 5.4 (2019), pp. 653–661. ISSN: 2095-8099. DOI: 10.1016/j.eng.2019.01.014.

[37] Birgit Vogel-Heuser and Dieter Hess. "Guest Editorial Industry 4.0—-Prerequisites and Visions". In: *IEEE Transactions on Automation Science and Engineering* 13.2 (Apr. 2016), pp. 411–413. ISSN: 1558-3783. DOI: 10.1109/TASE.2016.2523639.

[38] Soliman A. Al-Walaie. *Industrial wireless evolution - The next-generation process automation wireless technology*. https://www.isa.org/intech/20151001/. Accessed: 2020-04-03. Sept. 2015.

[39] S. Wang et al. "An Integrated Industrial Ethernet Solution for the Implementation of Smart Factory". In: *IEEE Access* 5 (Nov. 2017), pp. 25455–25462. DOI: 10.1109/ACCESS.2017.2770180.

[40] Andreas Wank et al. "Using a Learning Factory Approach to Transfer Industrie 4.0 Approaches to Small- and Medium-sized Enterprises". In: *Procedia CIRP* 54 (2016). 6th CIRP Conference on Learning Factories, pp. 89–94. ISSN: 2212-8271. DOI: 10.1016/j.procir.2016.05.068.

[41] Xuepei Wu and Lihua Xie. “Performance Evaluation of Industrial Ethernet Protocols for Networked Control Application”. In: *Control Engineering Practice* 84 (2019), pp. 208–217. ISSN: 0967-0661. DOI: 10.1016/j.conengprac.2018.11.022.

[42] Jihong Yan et al. “Industrial Big Data in an Industry 4.0 Environment: Challenges, Schemes, and Applications for Predictive Maintenance”. In: *IEEE Access* 5 (2017), pp. 23484–23491. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2017.2765544.

[43] Theodore Zahariadis et al. “Preventive Maintenance of Critical Infrastructures using 5G Networks Drones”. In: *2017 IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS).* Lecce, Italy, Aug. 2017, pp. 1–4. DOI: 10.1109/AVSS.2017.8078465.

[44] Ray Y. Zhong et al. “Intelligent Manufacturing in the Context of Industry 4.0: A Review”. In: *Engineering* 3.5 (2017), pp. 616–630. ISSN: 2095-8099. DOI: 10.1016/J.ENG.2017.05.015.

[45] Alfred Zimmermann et al. “Adaptive Enterprise Architecture for Digital Transformation”. In: *Advances in Service-Oriented and Cloud Computing.* Ed. by Antonio Celesti and Philipp Leitner. Taormina, Italy: Springer International Publishing, 2016, pp. 308–319. ISBN: 978-3-319-33313-7. DOI: 10.1007/978-3-319-33313-7_24.

[46] Marian Zoll, Daniel Jack, and Marcus Vogt. “Evaluation of Predictive Maintenance-as-a-Service Business Models in the Internet of Things”. In: *IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC).* Stuttgart, Germany, June 2018, pp. 1–9. DOI: 10.1109/ICE.2018.8436272.

A Flexible Mobility System based on Small and Low-emission Vehicles for Smart and Green Mobility

**Silvia Ullo[1,*], Mariano Gallo[1], Maurizio Di Bisceglie[1], Carmela Galdi[1], Mario Marinelli[1],
Luigi Glielmo[1], Giovanni Palmieri[1], Pietro Amenta[2], Antonella Ferrara[3], Michele Ferrucci[4], Gianpaolo Romano[4], Mariarosaria Russo[5], Marco De Angelis[6]**

[1]University of Sannio, Department of Engineering
Piazza Roma, 21, 82100 Benevento, ITALY
email: ullo@unisannio.it, gallo@unisannio.it, dibisceg@unisannio.it, galdi@unisannio.it, , mario.marinelli@unisannio.it, glielmo@unisannio.it, giovanni.palmieri@unisannio.it

[2]University of Sannio, Department of Law, Economics, Management and Quantitative Methods
Piazza Arechi II, 82100 Benevento, ITALY
email: amenta@unisannio.it

[3]University of Pavia, Department of Industrial and Information Engineering
Via A. Ferrata, 5, 27100 Pavia, ITALY
email: antonella.ferrara@unipv.it

[4]CIRA - Centro Italiano Ricerche Aerospaziali
Via Maiorise, 81043 Capua (CE), ITALY
email: m.ferrucci@cira.it, g.romano@cira.it

[5]KES S.r.l.
Via Mario Vetrone, 82100 Benevento, ITALY
email: rosy.russo@gmail.com

[6]Consorzio iCampus
Via Acquasanta, 31, 84131 Salerno, ITALY
email: m.deangelis@gematica.com

Abstract: *The environmental impact of transportation systems is considered a crucial point when new solutions must be adopted to enhance life quality in urban areas. All countries are seriously supporting and adopting green* *transportation systems according to increasingly stringent environmental quality targets. In this context, this paper introduces an innovative mobility system based on small and low-emission vehicles aimed to spread Mobility as a Service (MaaS) paradigm. The proposed system is based on a connected heterogeneous intelligent and personalized architecture integrated within an Intelligent Transportation System (ITS) framework. We assume the mobility as no longer a product, but the interconnection of different transport modes to find the best alternative based on comfort and convenience. As a result, this mobility system will provide a service with intermediate features between the private car and transit systems without fixed-scheduling and fixed-routes.*

1. Introduction

Urban and extra-urban mobility management is becoming highly complex when considering the main external impacts linked to traffic congestion, air pollution, noise and accidents.

If these impacts are neglected, it results in a loss of efficiency, increasing costs of public and private resources, and an unacceptable level of costs for sustainability.

According to the White Paper [1], the European Commission is setting up increasingly stringent environmental quality targets, such as a 60% reduction for transport emissions within 2050. Consequently, all governments should set up a broader range of leverages, policies, adoption of state-of-the-art technologies, revised economic models, tax incentives and contributions to driving investment towards new sustainable, intermodal and innovative mobility systems. In particular, the White Paper specifies the main strategies to be adopted like the improvement of vehicles' energy efficiency and more efficient use of transport modes, considering Intelligent Transportation Systems for information to users and traffic management. Moreover, strategies for urban transport and commuting involve the gradual elimination of conventional fuel vehicles, encouraging the use of smaller, lighter and specialized passenger cars, and the increase of collective transport demand.

In 2014, the Italian Ministry of Infrastructure and Transport adopted the National Plan for Intelligent Transport Systems (ITS) [2]. It pushes towards

ITS solutions such as multimodal mobility systems by integrating local public transport with private and alternative modes.

According to these main objectives, we aim to give a significant contribution to the way of reducing pollutant emissions by aligning with some of the strategies outlined by the European Commission and the Italian Government. We propose an ITS framework for information and traffic management to promote the use of low emission and small-sized vehicles integrated with the collective transport system. The proposed framework pushes more and more towards Smart Mobility, including Mobility as a Service (MaaS) and the reduction of private cars for collective transport systems.

A new formulation of real-time services is given: through a mobile device application the user is able to book a mobility service to reach the desired destination by using a vehicle close to his/her position. The system will identify travel times based on the most convenient routes and known congestion conditions and indicate one or more possible travel alternatives obtained by integrating existing mobility services from an intermodal perspective. The target is to design and develop a mobility-oriented ITS framework equipped with sensors, acquisition devices and transmission system.

The proposed framework is able to obtain and process a wide information base, in order to integrate it into a MaaS. Using Big Data Analytics methodologies and Machine Learning techniques, it will discover, extract, expose, share, and connect, in a fast and effective way, high frequency updated information to users, who can interact with the system through a simple mobile application.

In [3] and [4] a project proposal based on these concepts (NETCHIP project) is described.

The paper is structured as follows. In Section 2, we frame the work in the MaaS paradigm. The transport system model is introduced in Section 3. The overall architecture of the system is described in Section 4. The issues related to data analysis, transmission, collection and security are studied in Section 5. Section 6 concludes.

2. The Mobility-as-a-Service paradigm

In the last twenty years, mobility is changing the classical form of separated services among different transport modes towards a new concept based on a single mobility service accessible on demand. This new mobility paradigm, known as Mobility-as-a-Service (MaaS), provides a list of mobility options obtained as a combination of local public transport, ride-, car- or bike-sharing, taxi or car rental.

Thus, assuming a system where mobility is no longer a product but the interconnection of existing transport systems, we propose an innovative MaaS prototype which aims to find the best alternative based on comfort and convenience, by selecting paths and costs that are no longer standardized on a predefined line but flexible to user needs.

The main aim is to design an innovative flexible transit service, with similar features of a Paratransit system [5]. The Paratransit class includes several mobility services like the followings:

- **Car Sharing**: this system provides that users can use shared cars and hence no owners, typically in urban areas, by paying a fare based on their usage time and kilometres travelled.
- **Taxi**: this service is practically present in all medium-large cities of the world and is very similar to personal cars, in terms of flexibility in time and space; it is characterized by high fares so that its use is limited to occasional trips.
- **Dial-a-ride**: these systems are very similar to classic bus systems. This is a service, usually carried out with small size vehicles and with variable frequency, in which the user usually books the trip in advance (from the day before to 4-6 hours before) indicating the origin, destination and time desired beginning or ending journey. The service is organised to meet most user requests. Dial-a-ride system also includes route diversion services; these are traditional bus line services that can, however, deviate from the predetermined route to collect users in low-demand areas.
- **Jitneys**: these systems are commonly used in developing countries and consist of cars or minivan (from 5 to 15 seats), owned by privates, who operate on fixed routes, in some cases with small route deviations, which pick up and leave users along the route, often with no fixed stops. No reservation service is provided.

The MaaS-based system is able to find different rides between an origin-destination pair using public transportation services already present in the user urban area. As a result, this mobility system will provide a service with intermediate features between the private car and the fixed-scheduling and fixed-routes transit systems.

Moreover, the overall management system requires the use of road traffic simulation models which allow calculating network performances (travel times, average speed, etc.). The obtained information is used to organize and manage in real-time vehicles travelling in the road network (paths to follow, users to withdraw, etc.), using dynamic routing algorithms.

The end-user will interact with the system using a mobile application through the following main steps:

1. the user is geo-located, and he/she selects the desired destination;
2. the system computes the available multimodal routes and delivers them to the user, providing paths and prices attributes;
3. the user selects one of the provided alternatives and completes the procedure with payment by credit card: the system state will not change until the end of the procedure.

Behind such a service, a control strategy, based on some optimization criteria, is required to find different ride solutions (the fastest, the cheapest, the most comfortable). In this "active" approach, the user can reserve his/her seat on the ride and the system will coordinate the available vehicles according to the selected solution. The chosen solution enables the automatic procedures of real-time seat reservation and real-time routing of the involved services.

The different ride solutions with their different prices are the output of complex mathematical problems that should be solved in a very short time to guarantee a real-time service. Thus, to reduce the computational complexity, the control and optimization strategy is made of two different hierarchical tasks. The first one, at the high level, is to identify the starting and ending points that the involved vehicle must travel between. Then, the proposed solutions are integrated with the public transport system. It could be possible to obtain solutions that require the use of public transport in order to reach the starting and/or ending point of the ride performed by a vehicle of the fleet. For this reason, it is not guaranteed that the starting and ending points performed by the vehicle of the fleet will coincide with the origin and destination points as requested by the user. Once the starting and ending points are computed, at the lower level, the control strategy is responsible for computing the shortest path (in terms of total travel time), the cheapest path (in terms of ride price) and the most convenient path (in terms of walking distance).

The proposed service is designed to serve urban areas. Figure 1 shows a qualitative chart that frames the proposed system in the different classes of road transit systems. The proposed system is characterized by high flexibility, slightly lower than the one offered by the taxi, but with costs considerably lower than the taxi service, although higher than the traditional bus lines.

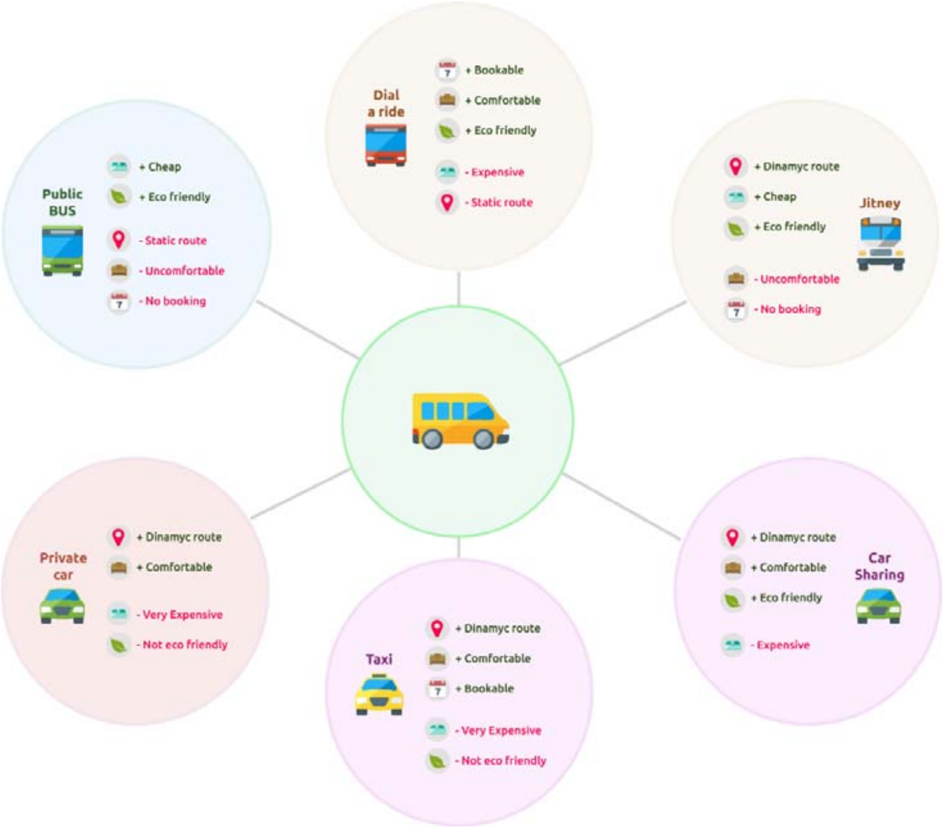


Figure 1. The proposed system

3. The transport system model

The system management requires road traffic simulation models to evaluate network performances and provide up-to-date information to users. To this aim, we formulate a new hybrid model of the transportation system that can be regarded as a micro-macro model.

In the literature, three different classes of road traffic models are generally considered:

- *macroscopic models*, where traffic flows are assimilated to a fluid using hydrodynamics relationships and provide an aggregate representation of traffic conditions. Examples are the well-known Cell Transportation Model (CTM) [6]-[7] or the recent Variable Length Model (VLM) [8];

- *microscopic models*, which simulate single-vehicle dynamics and its interaction with surrounding vehicles [9];
- *mesoscopic models* [10]-[11], which combine the macroscopic flow dynamics description with the microscopic interaction of packets made up of vehicles with homogenous characteristics.

The road network, where the flexible public transport system will serve users' requests, is modelled as a graph consisting of nodes and arcs. The spatial and temporal resolution of the transport network model (i.e., for instance, the number of flow transmission cells in which each arc is split) depends on the data which can be acquired in real-time. The model is used to make predictions about traffic flow dynamics, and it is re-initialized whenever new data are acquired to obtain simulations as reliable as possible. Once formulated, the model is also validated using historical data.

The macroscopic model is used as the basis for the decision-making process to find the best alternatives based on criteria such as comfort, economic convenience, and minimum arrival time at the destination.

The vehicles' dynamics in the flexible public transport system are simulated using a microscopic modelling approach. Each vehicle is considered as an agent with its characteristics in the dynamic network assignment to obtain a resulting coordinated behavior in order to reach the quality of service and emissions' reduction objectives. In the overall modelling approach, vehicles are simulated as embedded in the macroscopic traffic flow, which motivates the use of the term "micro-macro" [12, 13] for the proposed model.

After designing the overall urban traffic model, the implementation of control strategies is made using a simplified urban traffic model to predict traffic flows in a short time and to ensure real-time application. The simplified model is validated using comparative simulations.

Reduced Model Design is a modelling activity in which some details or some dynamics of complete simulation models are neglected without loss of the desired information. This process requires an accurate study of the complete model in order to understand which detail or which dynamics could be neglected without generating a significant error. The reduction process should be validated with parallel simulations between the complete and the reduced model to compare the evolution of the two models and check the prediction error.

The micro-macro model is used as a tool in the decision-making process aimed to propose different multimodal transport alternatives to the user. The alternatives are formulated by the decision-making module based on the different criteria specified by the user when requesting the service. Criteria can include comfort, affordability, time to reach the destination, but also the smallest volume of environmentally harmful emissions, with the aim of encouraging more eco-friendly behavior among users.

The main system management features of the transport system model are the followings:

- *Road network data retrieval*. As far as the transport system model is concerned, it is expected to receive information on actual traffic conditions from the sensors when they are available. Such data sending will result in a re-initialization of model state variables so as to keep the model as close to reality as possible. Sensors are expected to send data synchronously or asynchronously depending on their nature.
- *Steady-state management*. In the absence of service requests, the micro-macro model of the transport system will work in real-time, remaining synchronized with the physical traffic system.
- *User requests management*. The transport system model will receive a service request when an interested user sends a "query" to the system itself. The interface module with the transport system model will have to extract information from the query, such as the origin-destination pair, and the user-selected criteria among those available. Such information is sent from the interface module to the transport system model.
- *Solutions generation management*. The transport system model, after a service request, generates a temporary instance to compute as fast as possible the various eligible solutions according to the user-selected criteria. The identified solutions (i.e., optimal paths) are sent to the decision module in a predetermined format (e.g., sequences of network graph nodes and arcs indices). At the decision-making level, these solutions can be then compared and integrated with those based on other available transport modes in the geographic area (urban trains, subway, urban buses, etc.). The aim is to offer multimodal solutions to the user if more convenient than the unimodal ones elaborated by the micro-macro model.

Finally, the micro-macro model may also receive a service request directly from a vehicle monitoring module. This can occur whenever unexpected events (incidents, work in progress, etc.) happen at the time a certain solution is proposed by the micro-macro model to the decision-making module (and then accepted by the user). In this case, the values previously computed exceed a predetermined threshold and, thus, the model is helpful for the decision-making module in evaluating re-routing solutions and/or suggesting multimodal alternatives to users already travelling.

4. The overall system architecture

The system architecture is designed by defining new methodologies for data transmission and gathering. The system design is mainly based on the implementation of two parts:

1. a system for collecting and transmitting data from the served area and vehicles, equipped with intelligence able to detect patterns and provide real-time results;
2. a cloud-based infrastructure for the propagation of services to end-user and administrators.

From a transportation point of view, the system is able to:

- monitor traffic conditions, with ground sensors and vehicles used as probes;
- estimate real-time traffic conditions based on historical data and in conjunction with the traffic monitoring system;
- collect real-time user requests, in terms of origin-to-destination trips, desired departure time or arrival time, and other specifications could be provided;
- monitor the position of vehicles on the network, their routes, occupancy status and service availability;
- elaborate user requests in real-time by assigning to each request a path, possibly intermodal, that involves one or more trip segments served by the system;
- offer travel prices to the user that are dynamically calculated according to the available demand and supply in the required time window and subject to other user specifications;
- complete the service purchase agreement and manage payments with money-less services (credit cards, Paypal, etc.);
- elaborate and send information in real-time to vehicle drivers about user origin-destination pair, users to be served along the selected route, taking into account traffic conditions obtained by the monitoring system;
- manage vehicles' refueling based on estimated consumption without service interruption;
- gather user feedback on service;

- provide to public administration (municipality) data on traffic monitoring, service effectiveness, and reduction of external costs obtained by the service adoption.

Traffic monitoring and data acquisition are set up by using a sensors network. Sensors represent an important task since they are necessary for surveying in real-time the state of the system. The sensors network is composed by (ground) fixed sensors (e.g., inductive loop detectors, radars, etc.) and mobile sensors (e.g., probe vehicles). The latter ones can be provided using the vehicles of the service [14]-[18]. Other methods for transportation monitoring can also include Unmanned Aerial Vehicles (UAVs) and satellites. Especially, UAVs have several advantages over traditional traffic sensors, such as zero impact on ground traffic, good maneuverability, wide field of view. Moreover, considering the recent price reduction of UAV products, these devices are becoming prominent in transportation safety, planning and operations. Among UAV-based application, vehicle detection is possible to execute traffic monitoring [19].

The functional architecture focuses mainly on the specification and implementation of the data analysis module. A possible framework is reported in Figure 2. We use the Complex Event Processing (CEP), a technique for continuous data flow processing, in order to infer new knowledge and to provide support for early actions. Therefore, the analysis system can process online incoming data, as well as store them for offline processing.

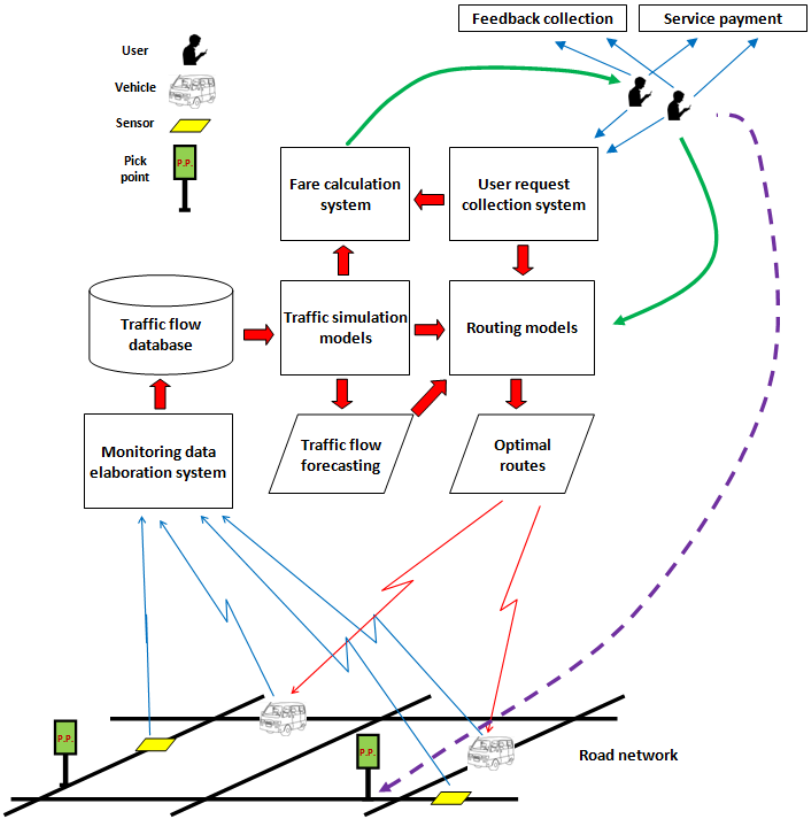


Figure 2. A possible functional framework [3]

The analysis process requires the definition of patterns already classified and determined, for example, by a predictive algorithm using training datasets. The latter represents an analytical model which acts as input to the Complex Event Processing system. Finally, a dashboard-based user interface displays the collected data time sequence and results to the analyst. From a data transmission and acquisition point of view, the system is able to:

- provide appropriate connectivity solutions to the identified data requirements;
- ensure high reliability;
- transmit and receive data from LPT vehicles;
- ensure high scalability to include future city expansions.

The Big Data Analytics system is intended to provide end-users with an easy-to-use tool for examining resource availability data (available vehicles, traffic flows, etc.) and developing strategies that involve all concerned stakeholders (producers, consumers, institutional bodies, etc.), who are in charge of adopting solutions for achieving sustainable mobility goals.

5. Data collecting, analysis, transmission and security

The system is intended to correlate, analyze and synthesize large amounts of data using dataset integration, to enable data coming from different sources, shared within users' communities and referred to identical objects or object correlated in a complex way, to point out their qualities.

An important aspect is the business branch referred to the "renting" of the whole system architecture using a cloud approach: an innovative instrument of route analysis for companies that manage micro-transit systems, using the predictive modelling, and optimizes routes for other companies, or using just a part of the IoT architecture and transmission.

Datasets that interact with the Decision Support System, representing the system inputs, can be divided into:

- *Traffic data*: collected through sensors already on the road network or available in the Open Data online repository concerning average vehicle flows along certain roads, the average density of road sections, congestion level, speed, average number of vehicles passing through an area.
- *Environmental data*: collected through existing control units or via APIs related to both atmospheric parameters (e.g., temperature, humidity, rainfall) and pollutant parameters (e.g., PM10, benzene, carbon monoxide, nitrogen dioxide).
- *Public transport data*: collected through direct interaction with transport companies or using Istat and Open Data repositories concerning the number and type of available means of transport, routes and stops, user basin, number of subscribers, the volume of demand and supply, presence of car and bike-sharing.
- *Urban data*: collected through direct interaction with Public Administrations and the implementation of linking APIs with cartographic systems to delineate the induced characterization of a given area, such as the presence of schools, hospitals, main link roads, presence of commercial activities and transit restrictions;
- *Means of transport data*: collected through devices installed on board of the means of transport, they are able to provide information

regarding the position of the vehicle, fuel consumption, distance travelled, and any anomalies related to the vehicle.

We propose to use transmission technologies considering up-to-date network architectures and communication protocols taking into account sensors network critical issues. Some technologies which are not yet widely used at the application level such as LORA Wan, Wi-Fi, Wi-Fi-HaLow, 6LowPAN, Bluetooth Low Energy, 4G-LTE, NB-IoT can be considered. More innovative communication technologies, such as 5G, can be analyzed. Considering the involved application scenarios, the 5G offers greater bandwidth, reduced latency and less packet loss compared to existing transmission technologies. Additionally, the 5G is able to support up to one million concurrent connections per square kilometer, enabling a variety of services such as those that could be offered by advanced systems for a Smart City [20], [21].

Moreover, in the future activities the new paradigm, called *fog computing*, will be considered. Although many technologies, such as multiple-input and multiple-output (MIMO), are necessary to enhance capacity in the fourth-generation era, it is not economically feasible to deploy such technologies in the fifth generation. The *fog* expands cloud computing and services on the edge of the network and produces data, computing, storage, and application services to users that can be on the edge of the networks such as access points, and therefore it is a perfect substitute as a candidate technology for networks that exceed 5G, where the cloud is to be deployed among the client's devices. Besides this, *fog* and Software Define Networking (SDN) based architectures are estimated as efficient systems to be employed specifically for vehicular environments, that is the case of our project [22], [23].

The proposed system aggregates information coming from several sources, including sensors network, with the aim of building decisions support models for better mobility management. Consequently, the communication technologies should transmit in real-time a big quantity of data from/to vehicles, users, control centre and infrastructures, and supported by simulations [24]-[29]. Thus, heterogeneous data coming from the sensors network are used to build the prediction model of the state of the system [30]. The prediction model should provide timely information to operators, control systems and users about urban mobility flows. Unfortunately, data diversity makes the construction of the prediction model particularly complex, since it requires a data flow management system characterized by complex structures due to the heterogeneity of sources. However, the data fusion process from multiple sources could give better accuracy, greater robustness and confidence in traffic flow estimation. One of the main challenges of fusing multi-source data comes from the fact that different sources usually provide information at different spatio-temporal resolutions. A solution to this complexity in building the prediction model is the use of state-space, both static and dynamic, models like the Kalman filter [31], well-known for its predictive capacity [32]. Specifically, the model is considered dynamic when the traffic-related variables vary over time. The goal is to compute traffic control measures based on the forecast provided by the model. Moreover, based on new data acquired by the monitoring system, it will be possible to apply correction techniques aimed at reducing the prediction error.

Alternatively, artificial neural network (ANN)-based models [33]-[34] represent a good solution among the machine learning techniques [35]-[38], especially for short-term forecasting. Artificial neural networks are particularly versatile in managing vast amounts of data and allow complexity management since not based on linear forms as Kalman filters. Thus, ANNs can be adopted with confidence for rapidly forecasting traffic flows in transportation networks when explicit relationships between flows are not necessary. However, in the domain of neural networks, hybrid methods are necessary to combine static and dynamic networks, also taking into account spatial and temporal data dependency structures.

Regarding the analysis of correlations of a large amount of data from heterogeneous sources (e.g., air pollution data from the local environmental sensor network, traffic data coming both from fixed detection devices and intravehicular sensors, meteorological data), Big Data Analytics approaches should be used together with pre-processing techniques of raw data. Data correlation analysis aims to extrapolate forecasting models based on the local correlation of pollution, transport and meteorological variables, given a certain geographical area of interest, and based on a highly available number of local data sources sufficiently dense and reliable enough to build a temporally deep database. Using statistical methodologies, the cross-domain correlation among data can be investigated to select the key features that best fit the analysis requirements. Moreover, Data Mining and Machine Learning techniques and methodologies are necessary to find and report anomalies in traffic conditions or forecast local pollution at a short-medium term.

The security for a sensors network is important because the validity of the collected data is a fundamental premise to have decisions support models based on real data. In particular, the collected data must be:

a. attributable with certainty to a specific sensor, because the information is strictly related to the road network;
b. immutable by external attacks, because data must be authentic;
c. not readable by unauthorized users, because the reported information may be sensitive (e.g., people or vehicles location).

All the possible risks or attacks scenarios should be analyzed, identifying for each case the possible remedies, and functional/operational requirements for the data communication system should also be defined. The final aim is to define a set of guidelines which allow the identification of security and data communication technologies during the designing phase. Another goal is to define the right protocol for monitoring data communication, with a focus on the ciphering methods to assure efficient access and manipulation of the collected data.

The proposed system operates in a contest with data coming from several different sources and all these sources are classified. Finally, the way the digital identity services are interconnected with the sensors network and the interconnection of sensors networks in one unique system is another important aspect.

6. System performance assessment

The performance of the proposed system has to be properly monitored and evaluated from different points of view. Five main areas of evaluation can be identified:

1. *effectiveness*;
2. *efficiency*;
3. *quality*;
4. *environment*;
5. *financial sustainability*.

For each of these areas, a set of Key Performance Indicators (KPIs) can be defined to measure system performance.

Some of these indicators can also be evaluated in the forecasting phase, in order to guide the design of the system, while others need to be monitored during the system operations, to improve the management phase.

Table 1 describes the main indicators that can be used for each area. The list is not exhaustive: the need for other indicators may arise after the implementation of the system.

Table 1. Main Key Performance Indicators (1/2)

Area of evaluation	**Indicator**	**Description**	**Before**	**After**
Effectiveness	Users/day	It measures the average number of daily users of the system. Data are obtained directly from the centralized management system.		X
	Modal share	It measures the percentage, on a daily basis, of trips using the system, compared to total trips.		X
	Transit share	It measures the percentage, on a daily basis, of trips using the system, compared to trips on the transit system.		X
Efficiency	Cost per passenger-km	It measures the average cost to the operator in relation to the passenger-km transported, on a monthly or annual basis. It can be calculated as a function of the production cost of the service and the total number of kilometers that passengers have travelled onboard the system.		X
	Cost per veh-km	It measures the average cost to the operator in relation to the km travelled, on a monthly or annual basis. It can be calculated on the basis of the production cost of the service and the total number of km travelled by the vehicles in the system, including km without passengers on board.		X
	Passenger-km/veh-km	It measures the ratio between passenger-km and vehicle-km, on a daily, monthly or annual basis.		X

Table 1. Main Key Performance Indicators (2/2)

Area of evaluation	Indicator	Description	Before	After
Quality	Average waiting service time	It measures the average waiting time of the users who have requested the service, in order to have the booking confirmation.		X
	Average departure delay	It measures the average delay that the users experience in relation to the scheduled time of departure.		X
	Average arrival delay	It measures the average delay that the users experience in relation to the scheduled time of arrival.		X
Environment	CO_2 emissions saved	Estimation of CO_2 saved with the implementation of the system. The calculation of this indicator requires accurate forecast models and assumptions to be verified also with user surveys.	X	X
	PM emissions saved	Idem, related to Particulate Matter.	X	X
Financial sustainability	Revenue/cost ratio	It calculates the ratio between revenues from the sale of the service and its total costs, on an annual basis. This indicator must be estimated in advance and recorded during the operation phase of the system. Preventive estimation requires the implementation of accurate forecasting models.	X	X

In addition to these indicators, others should be added concerning the performance of the technological components of the system, from the point of view of reliability, security and protection of personal data.

7. Conclusions

The paradigm of Mobility as a Service (MaaS) is increasingly emerging as a possible alternative to reduce congestion, land use and external impacts of transport systems. Research in this field is necessarily interdisciplinary because it requires expertise in many areas: from transport systems engineering to operational research, from big data analysis to communication and management technologies, from automation processes to environmental and traffic monitoring methodologies.

In this work, preparatory to the proposal of future research initiatives, an attempt has been made to identify the main components of the system, their interconnections, the problems that need to be addressed and resolved, as well as the opportunities and perspectives that this research topic can offer.

References:

[1] European Commission, White Paper. Roadmap to a Single European Transport Area - Towards a competitive and resource efficient transport System, COM(2011), Brussels, 2011.

[2] Ministero delle Infrastrutture e dei Trasporti, Piano di Azione Nazionale sui Sistemi Intelligenti di Trasporto (ITS), 2014.

[3] M. Gallo, S. Ullo, P. Amenta, G. Palmieri, A. Ferrara, M. Ferrucci, M. Russo, M. De Angelis "A Flexible Mobility System based on CHIP Architectures: the NETCHIP Research Project," Proceedings of 2018 IEEE International Conference on Environmental and Electrical Engineering and 2018 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe), pp. 602-607, 2018.

[4] S. Ullo, M. Gallo, G. Palmieri, P. Amenta, M. Russo, G. Romano, M. Ferrucci, A. Ferrara, M. De Angelis, Application of Wireless Sensor Networks to Environmental Monitoring for Sustainable Mobility. Proceedings of 2018 IEEE International Conference on Environmental Engineering, 2018.

[5] V.R. Vuchic, Urban Transit Systems and Technology, John Wiley & Sons, Inc., 2007.

[6] C. Daganzo, "The cell transmission model: a dynamic representation of highway traffic consistent with the hydrodynamic theory," Transportation Research Part B, vol. 28, pp. 269-287, 1994.

[7] C. Daganzo, "The cell transmission model, Part II: Network traffic," Transportation Research Part B, vol. 29, pp. 79-93, 1995.

[8] C. Canudas-de-Wit, and A. Ferrara, “A Variable-Length cell road traffic model: Application to ring road speed limit optimization,” 2016 IEEE 55th Conference on Decision and Control (CDC), Las Vegas, NV, pp. 6745-6752, 2016.

[9] Q. Yang, and H.N. Koutsopoulos, “A microscopic traffic simulator for evaluation of dynamic traffic management systems,” Transportation Research Part C, vol. 4, pp. 113-129, 1996.

[10] E. Cascetta, and G.E. Cantarella, “A day-to-day and within-day dynamic stochastic assignment model,” Transportation Research Part A, vol. 25, pp. 277–291, 1991.

[11] M. Di Gangi, and A. Polimeni, “A Model to Simulate Multimodality in a Mesoscopic Dynamic Network Loading Framework,” Journal of Advanced Transportation, vol. 2017, art. no. 8436821, pp. 1-16, 2017.

[12] J. Lebacque, J. Lesort, and F. Giorgi, “Introducing buses into first-order macroscopic traffic flow models,” Transportation Research Record, vol. 1644, pp. 70-79, 1998.

[13] G. Piacentini, P. Goatin, and A. Ferrara, “Traffic control via moving bottleneck of coordinated vehicles,” 15th IFAC Symposium on Control in Transportation Systems (CTS 2018), Savona, Italy, June 6-8, 2018.

[14] L. D’Acierno, A. Cartenì, and B. Montella, “Estimation of urban traffic conditions using an Automatic Vehicle Location (AVL) System,”. European Journal of Operational Research, vol. 196, pp. 719-736, 2009.

[15] G.A. Klunder, H. Taale, L. Kester, and S. Hoogendoorn, “Improvement of network performance by in-vehicle routing using floating car data,” Journal of Advanced Transportation, vol. 2017, art. no. 8483750, pp. 1-16, 2017.

[16] P.B.C. van Erp, V.L. Knoop, and S. Hoogendoorn, “Macroscopic traffic state estimation: understanding traffic sensing data-based estimation errors,” Journal of Advanced Transportation, vol. 2017, art. no. 5730648, pp. 1-11, 2017.

[17] X. Yang, Y. Lu, and W. Hao, “Origin-destination estimation using probe vehicle trajectory and link counts”, Journal of Advanced Transportation, vol. 2017, art. no. 4341532, pp. 1-18, 2017.

[18] M. Carminati, O. Kanoun, S. L. Ullo and S. Marcuccio, “Prospects of Distributed Wireless Sensor Networks for Urban Environmental Monitoring," in IEEE Aerospace and Electronic Systems Magazine, vol. 34, no. 6, pp.44-52, 1 June 2019.

[19] Y. Xu, G. Yu, Y. Wang, X. Wu, and Y. Ma, “Car Detection from Low-Altitude UAV Imagery with the Faster R-CNN,” Journal of Advanced Transportation, vol. 2017, art. no. 2823617, pp. 1-10, 2017

[20] N. Hassan, K. A. Yau and C. Wu, "Edge Computing in 5G: A Review," in *IEEE Access*, vol. 7, pp. 127276-127289, 2019.

[21] Y. Ai, M. Peng, and K. Zhang, "Edge computing technologies for Internet of Things: a primer", Digital Communications and Networks, Volume 4, Issue 2, 2018, Pages 77-86,

[22] A. Baktayan, M. Na, S. Alhomdy, "Fog Computing for Network Slicing in 5G Networks: An Overview", Journal of Telecommunications System & Management, Volume 7, Issue 2, pp. 1-13, 2018.

[23] S. Aggarwal, N. Kumar, " Fog Computing for 5G-Enabled Tactile Internet: Research Issues, Challenges, and Future Research Directions". Mobile Netw Appl (2019).

[24] J.J. Fernández-Lozano, M. Martín-Guzmán, J. Martín-Ávila, and A. García-Cerezo, "A Wireless Sensor Network for Urban Traffic Characterization and Trend Monitoring," Sensors, vol. 15, pp. 26143-26169, 2015.

[25] W.D. Jones, "Forecasting Traffic Flow," IEEE Spectrum, vol. 38, pp. 90-91, 2001.

[26] Y. Zhao, "Mobile phone location determination and its impact on intelligent transportation systems," IEEE Transactions on Intelligent Transportation Systems, vol. 1, pp. 55-64, 2000.

[27] J. Mathew, and P.M. Xavier, "A Survey on Using Wireless Signals for Road Traffic Detection," International Journal of Research in Engineering and Technology, vol. 3, pp. 97-102, 2014.

[28] A. Janecek, D. Valerio, K.A. Hummel, F. Ricciato and H. Hlavacs, "The Cellular Network as a Sensor: From Mobile Phone Data to Real-Time Road Traffic Monitoring," IEEE Transactions on Intelligent Transportation Systems, vol.16, pp. 2551-2572, 2015.

[29] J.F. Raquet, M.M. Miller, and T.Q. Nguyen, "Issues and Approaches for Navigation Using Signals of Opportunity," Proceedings of the 2007 National Technical Meeting of The Institute of Navigation, San Diego, CA, January 2007, pp. 1073-1080, 2007.

[30] A. Nantes, D. Ngoduy, A. Bhaskar, M. Miska, and Edward Chung, "Real-time traffic state estimation in urban corridors from heterogeneous data," Transportation Research Part C, vol. 66, pp. 99-118, 2016.

[31] J. Durbin, and S.J. Koopman, Time Series Analysis by State Space Methods: Second Edition, Oxford Statistical Science Series, 2012.

[32] S. Galit, C. Kenneth, and J. Lichtendahl, Practical Time Series Forecasting with R, Axelrod Schnall Publishers. 2 edition, 2016.

[33] K.-L. Du, and M.N.S. Swamy, Neural Networks and Statistical Learning, Springer, 2014.

[34] G. De Luca, and M. Gallo, "Artificial Neural Networks for forecasting user flows in transportation networks: literature review, limits, potentialities and open challenges," Proceedings of 2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems, Naples, Italy, 26-28 June 2017, pp. 919-923, 2017.

[35] A. Azzalini, and B. Scarpa, Data Analysis and Data Mining, Oxford University Press, 2012.

[36] B. Efron, and T. Hastie, Computer Age Statistical Inference, Cambridge University Press, 2016.

[37] T. Hastie, R. Tibshirani, and J. Friedman, The elements of statistical learning, 2nd ed., Springer, 2008.

[38] T. Hastie, R. Tibshirani, and M. Wainwright, Statistical learning with sparsity, 2nd ed., CRC, 2015.

An IoT Solution and Real-Time Detection System for Crop Protection against Ungulates

Mike O. Ojo[1], Davide Adami[2] and Stefano Giordano[1]

[1]Department of Information Engineering, Università di Pisa, Italy

[2]CNIT Research Unit, Department of Information Engineering, Università di Pisa, Italy

Abstract: *Advances in standards and protocols driven by the increasing interest in the Internet of Things (IoT) are allowing for more choice in design, thus making smart agriculture, among other sectors, to be more cost effective. The plethora of IoT compatible sensors, wireless transmission systems, and other system elements is already allowing designers a much higher degree of flexibility in realizing new designs for heretofore special purpose systems. This paper presents the development of IoT applications for crop protection to prevent animal intrusions in the crop field. A repelling and a real-time monitoring system are provided to prevent potential damages in agriculture from wild animal attacks. Moreover, this paper provides an in-depth description of a complete solution we designed and deployed, that consists of the low-power wireless network, the neural network solution for animal detection and the back-end system. Specifically, we develop a methodology for deploying the network and present the open-source tools to assist with the deployment, and to monitor the network. The system also allows the integration of neural networks that allows a real-time animal detection. Lastly, this paper also discusses how the technology used is the right one for smart agriculture in relation to crop protection.*

1 Introduction

Statistical data shows that there has been a massive surge in the loss of wine production due to the crop damages caused by animal attacks for the past 3 decades. In general, especially in Europe, the number of wine production losses due to animals' amount to 75% of the total wine production loss. Similarly, frightening statistics come from the United states wine production, that apart from the California fires, wildlife attacks to crop production has been on the increase. Taking Italy as an example, the annual production loss in the wine industry is estimated to be 13 million euros, with an annual cost to the government estimated around 3 million euro [1, 2].

Considering the above, there have been several ways to keep animals from destroying crop, which can be lethal means and sometimes non-lethal means. Lethal ways such as shooting, trapping, string and stone, are very cruel and not environment friendly, while non-lethal means such as scarecrow, chemical repellents, organic substances, fencing are sometimes inadequate, non-substantial, time consuming and also expensive [3]. Some of these methods even have environmental pollution effect on both humans and animals [4]. Technology assistance at various stages of agricultural processes can significantly enhance the crop yield [5].

The use of Ultrasound emission is an excellent system to repel animals, without simultaneously disturbing humans. Animals generally have a sound sensitive threshold that is far higher than humans. They can hear sounds having lower frequencies with respect to human ear. For instance, while the audible range for humans is from $64Hz - 23KHz$, the corresponding range of cows, sheep, dogs and cats is $23Hz - 35KHz$, $10Hz - 30KHz$, $67Hz - 45KHz$ and $45Hz - 64KHz$ respectively.

Researchers have shown that generating ultrasounds within the critical perceptible range causes animals to be disturbed, thus making them to move away from the sound source. At the same time, these ultrasounds are not problems to the human ear even when the frequency range is beyond the human ear. The human eardrum has a far lower specific resonant frequency than animals and cannot vibrate at ultrasound frequency. Human ear can perceive sounds up to $23KHz$ frequency.

Our proposed method for protecting crops from animal attacks is based on the generation of ultrasounds and utilizes IoT technology for providing a system with repelling and monitoring capabilities. IoT technologies are no longer buzz phrases or hypes, as they have found real life applications, such as in our case, smart agriculture.

The emergence of IoT is a phenomenon that owes to the conjunction of several factors such as inexpensive devices [6], low-power wireless technologies, availability of cloud data centers for storage and processing, management frameworks for dealing with unstructured data, high performance computing resources computational intelligence algorithms to deal with this enormous amount of data. Among different and various wireless technologies for the IoT, LoRa and LoRaWAN, global de facto standards of Low-Power Wide Area Networks (LPWAN) are gaining significant interest and positive momentum thanks to

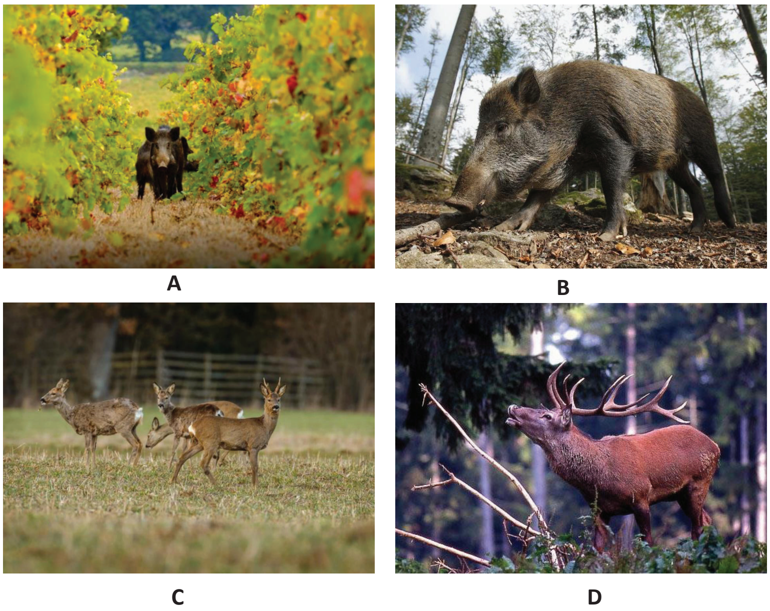


Figure 1: (A) wild boar in a vineyard, (B) wild boar (C) roe deer (D) fallow deer

their excellent features in covering long distances and in large scaling with low cost and high energy efficiency.

Moreover, with the upsurge of machine learning, deep learning algorithms have been extensively used in agriculture-related applications [7]. Real-time detection in agriculture is one of the most effective ways in helping farmers and agronomists in their decision making processes and management skills. This paper presents an animal repelling and dense real-time monitoring solution based on IoT technology to address the problems of crop damages especially vineyard in the Tuscany region of Italy. Figure 1 shows the most common ungulates in Tuscany region affecting crop production.

The main contributions of this article are:

1. We provide a thorough, complete and "brutally honest" description of the design and deployment of a real-time animal detection and repelling system.

2. We provide a methodology for deploying network and software infrastructures for such type of IoT applications.

3. We contribute with open source software to perform pre-deployment tests, manage the network, and store/display information about the status of IoT and network devices.

The rest of this paper is organized as follows. Section 2 describe the state-of-the-art and the related works. Section 3 describes the overall system architecture, whereas the network deployment is discussed in Section 4. Section 5 highlights the lessons learnt and finally, Section 6 provides some final remarks.

2 Background and Related Works

In this section, we present a short overview of the current activities related to smart agriculture applications deployments focusing on crop production. Next, we provide some background about LoRa and LoRaWAN, the wireless network technologies adopted in our application.

2.1 Smart Agriculture

The work in [8] proposed a model to measure the crop productivity and anticipate the potential problems. The proposed model combines a short and medium wireless network range system with a prediction engine to anticipate crop productivity dysfunctions proactively. Muangprathub *et.al* [9] proposed a system for watering agricultural crops based on wireless sensor network. The proposed method introduced a control system using node sensors in the crop field with data management via smart-phone and a web application. The model also utilized data mining to analyze the data for predicting suitable temperature, humidity, and soil moisture for optimal future management of crops growth. The work in [10] developed an e-Agriculture application based on the KM-knowledge base to monitor crop productivity.

The work in [11] presented an integrated approach in the field of IoT for smart Agriculture based on low power devices and open source systems for repelling and monitoring system for crop protection against animal attacks and weather conditions. Nóbrega *et.al* [12] proposed an animal behaviour monitoring platform based on IoT technologies. It includes an IoT network for gathering data, a cloud platform which incorporates machine learning features. The work in [13] presented a real-time rice crop monitoring system to increase its productivity. Chen *et.al* [14] proposed a low-cost agri talk IoT-based platform for the precision farming of soil cultivation.

To compare related works, and also highlight opportunities and gaps, we introduce in Table 1 considering four elements, namely (1) communication technologies (2) Data Analysis (3) Data display (4) Application purpose. Some researchers are working with various wireless technologies in agriculture, but we observed the lack of studies on applying LoRa in a practical and reproducible way.

Table 1: Related Work Comparison

Article	**Communication Technology**	**Data Analysis**	**Data Display**	**Application**
[8]	LoRa	Neural Network	Yes	Arugula cultivation
[9]	WSN	Data Mining	Yes	N/A
[10]	3G, WIFI	N/A	No	N/A
[11]	6lowPAN	N/A	No	wine protection
[12]	3G	Decision tree, KNN, SVM	No	sheep monitoring
[13]	GSM	N/A	Yes	rice production
[14]	4G	random forest-based	Yes	Soil cultivation

2.2 LoRa and LoRaWAN Basics

In this section, the structure and main parameters of LoRa and LoRaWAN are briefly presented. LoRa is very robust over a long distance due to Chirp Spreading Spectrum (CSS), where the physical channel is logically separated by the spreading factors (SF) due to their orthogonality. The carrier frequency varies over a designated amount of time, thus achieving low power and long-range communication links [15].

2.2.1 Network Structure

A network structure based on LoRa technology consists of four individual layers namely the end devices, the gateway, the network server and the application server. A brief description of each layer is highlighted below:

- ***LoRa End Nodes*** are devices embedded with LoRa chips. There are 3 classes of end-devices: Class A (for All), B (for Beacon) and C (for Continuously listening), each associated with a different operating mode. The devices broadcast their sensor values to all gateways in all range which forward data packets to a single network server over an IP based network.

- ***LoRa Gateways*** are intermediate devices running an operating system that forwards data packets coming from the end nodes to a network server over an IP-based backhaul network. In a LoRaWAN deployment, there can be multiple gateways receiving data packets from a LoRa end device.

- ***LoRa Network Server*** perform a lot of functions such as filtering redundant packets, performing adaptive rate, performing security checks and generally manages the network. For example, it knows about active nodes, and when a new LoRa end node joins the network.

- ***LoRa Application Server*** is responsible for encryption, decryption, and processing of data from the network server. The application server allows users to access and manage the gateway, nodes and applications.

2.2.2 LoRa Main Parameters

The LoRa technology is defined by the main parameters which are spreading factor, bandwidth and code rate. It is possible to configure different LoRa parameters in order to adapt the technology to the working scenario. We briefly describe each parameter below:

- ***Bandwidth (BW)*** is the range of transmission frequencies varying between $7.8kHz$ and $500kHz$. The more the bandwidth value, the more the transmitted data, thus reducing transmission time and resulting in lower sensitivity.

- ***Spreading factor (SF)*** characterizes the number of bits sent in each LoRa symbol. SF take values between 7 and 12 resulting in different time-on-air (ToA), thus, varying receiver sensitivity. Having a higher SF such as $SF = 12$ denotes longer range with low bit rate and better receiver sensitivity. The relationship between the LoRa transmission (ToA) and the used LoRa parameters is denoted as $ToA = 2^{SF}/BW$.

- ***Code Rate (CR)*** is related to the number of redundant bits used to improve the packet error rate in the presence of noise and interference. The possible values of CR are 4/5, 4/6, 4/7 and 4/8. A lower coding rate results in better robustness at the expense of increased transmission time and high energy consumption.

- ***Transmission Power (TP)*** is also an essential LoRa parameter varying from region to region. In Europe for example, with $863 - 870MHz$, the TP can be $2dBm$, $6dBm$, $8dBm$, $12dBm$, and $14dBm$.

LoRa MAC layer is basically an ALOHA protocol that is controlled by the network server. The gateway can receive from multiple end devices simultaneously. Vangelista [16] pointed out that the number of transmitting devices that can be received by the gateway simultaneously is limited to 9 due to the orthogonality of transmission sub-bands and quasi-orthogonality of the spreading factors. Moreover, the end devices have to respect the regulatory restrictions with a duty cycle of less than e.g. 1% in each of the European (EU) 868MHz bands.

3 System Architecture

In this section, we present the architecture of the overall system (see Figure 2), which consists of three main components: the ultrasound repeller device, the real-time detection system and the back-end system.

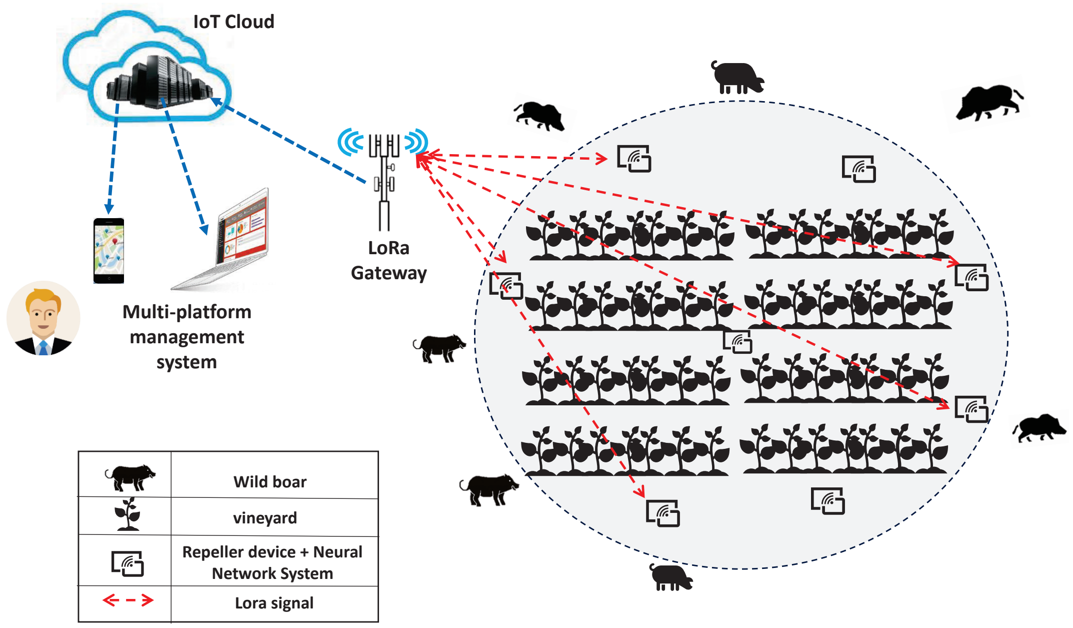

IoT Cloud
LoRa Gateway
Multi-platform management system
Wild boar
vineyard
Repeller device + Neural Network System
Lora signal

Figure 2: System Architecture

3.1 Ultrasound Repeller Device

The Natech Escape [17] is an IoT development board that provides animal repelling capabilities through the generation of ultrasounds. Specifically, the Natech Escape board uses a ATSAMD21G18A 32-bit ARM Cortex® M0+ core architecture clocked at 48MHz and paired with 32KB of RAM and 512KB of flash. It also features a LoRa module RN2483A and xbee radio module supported by the LoRaWAN and IEEE 802.15.4 standard respectively. All the pins of the micro-controller are exposed, allowing a developer to interface sensors and actuators over digital and analog interfaces. The device uses a solar panel along with LiPo batteries that are charged with a battery recharging system. The battery recharging system is equipped with a Pulse-width modulation (PWM) charge regulator that ensures the highest possible output for off-grid applications, making it an autonomous device able to work even in periods of partial or total darkness. Figure 3 shows the Natech Esacpe board in an enclosure with the lid open.

To improve the energy efficiency of the device, we made use of a Passive Infrared Sensor (PIR) sensor (see Section 4.1.1), which activates the driver responsible for the ultrasound generation and as well as the networking communication only when an animal is

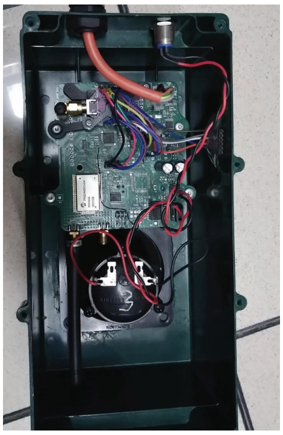


Figure 3: The repeller device encased in an enclosure

detected. The tweeter in the device is capable of generating ultrasound signals at pressure close to $110dB$ SPL in ∼1m distance and in a wide band of $18kHz - 27kHz$ that allows us to tune the device according to the animal that is desired to be repelled. The Natech Escape device can be equipped with two different kinds of lenses: (1) the long range lens, which is used for border protection and (2), the wide range lens, which is used for area protection. To transmit, process and store the information retrieved by the device, we are using a Proxy software that collects the "activities", i.e. animal detected events, and transmits this information via LoRa first to the LoRa gateway, and then to the back-end system. The Natech Escape board will be exposed to direct sunlight, freezing temperatures, dust and rain. Therefore all device and sensors are protected by an enclosure with internal protection 65 (IP65) rating[1].

As of the Operating System (OS), we make use of an open source solution called RIOT [18], which offers a lot of features like, multi-threading, efficient network stack and memory allocation. Our decision for adopting RIOT OS is based on the viable characteristics it offers to the IoT community. The RIOT OS was released in 2013 and it is based on the microkernel architecture which makes it perfect for real-time use cases. As regards to Networking, RIOT OS uses a network stack that is based on IP with support of IEEE 802.15.4, 6LoWPAN, IPv6, RPL, UDP and CoAP. As for the Programmability, it uses a C/C++ syntax with support of multiple threads and a memory passing inter-process communication (IPC) among the threads. The performance evaluation of RIOT OS has been demonstrated in [19]. In our implementation, we exploited the features of RIOT OS mentioned above by adding one thread for the detection using a PIR and transmitting a message to the LoRa gateway through the LoRa module. Another thread is used for communicating with the neural network system, through the xbee radio by exchanging UDP messages that will be discussed later in this article. Figure 4 shows the natechescape ultrasound device used as a virtual fence.

[1] Totally dust tight and protection against low pressure water jets in all direction

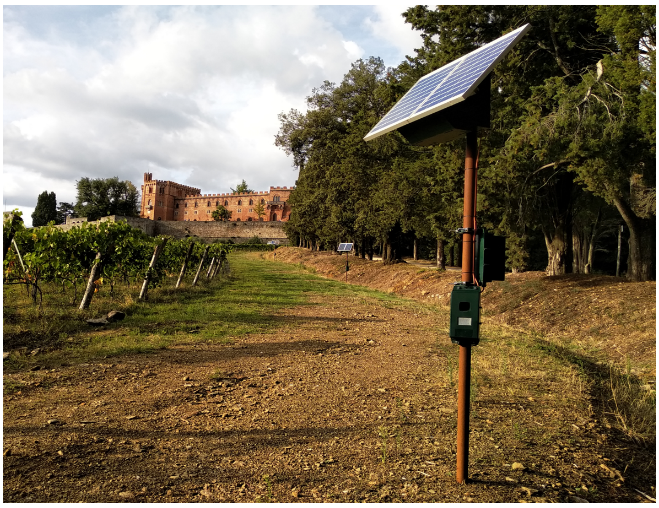


Figure 4: Ultrasound as a virtual fence

3.2 Real-Time Animal Detection System

Object detection and recognition is a common term for computer vision techniques aimed at detecting and recognizing objects in pictures or videos. Deep learning methods are already used widely in real time object detection. Real time smart solutions inspired from deep learning, must possess key capabilities such as energy efficiency, affordable and small form factor, as well as fine balance between accuracy and power consumption. Recent object detection and recognition techniques are mostly established on the utilization of convolutional networks (CNN). There are three main methods in the field of deep learning-based object detection and recognition: Single Shot Detector (SSD), Faster Region CNN (F-RCNN) and You Only Look Once (YOLO).

The YOLO method [20], [21] deals with the classification and the localization as a regression problem. A YOLO network directly performs regression to detect targets in the image without region proposal network, hence it is fast and can be implemented in real-time applications. The state-of-art version (YOLOv3) [21] not only has high detection accuracy and speed, but also performs well with detecting small targets. Optimization of the YOLOv3 model parameters reduces the computational complexities and thus is needed to deploy it on edge devices such as Nvidia Jetson Nano and Raspberry Pi [6].

In this article, we make use of the OpenCV deep neural network library to perform object detection and recognition. We opt for YOLOv3-tiny model as the classifier and localizer, because it is light-weight, fast-speed compared to YOLOv3 which requires a lot more processing power making YOLOv3-tiny the ideal model for our proposed framework. YOLOv3-tiny is said to produce better accuracy and frame rate than YOLOv3 on embedded platforms. In our work, we deployed the YOLOv3-tiny algorithm on embedded platforms such as Raspberry Pi (hereinafter, RPi) 3B+ with Intel Movidius Neural Computing Stick (NCS) and Nvidia Jetson Nano. Table 2 shows a comparison between the main specifications of the selected embedded hardware while Figure 5 shows the different hardware platforms that are considered.

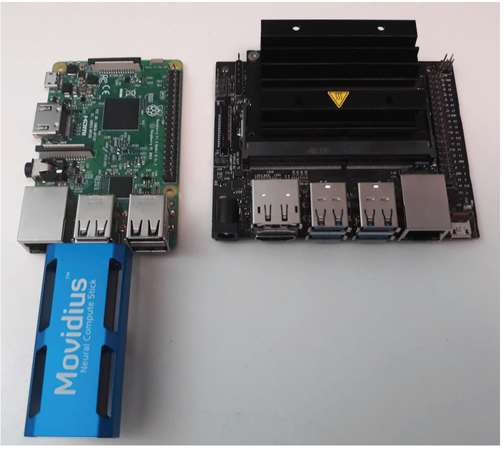


Figure 5: The analyzed embedded platforms: RPi 3B+ with Intel Movidius NCS on the *left* and Nvidia Jetson Nano on the *right*

Table 2: Main specifications of the platforms investigated in this study

Features	**Intel Movidius NCS**	**Jetson Nano**
Size	73 × 26 mm	73 × 45 mm
HW Accelerator	Myriad 2 VPU	128-core NVIDIA Maxwell GPU
CPU	N.A	Quad-core ARM A57 @ 1.43GHz
Memory	4GB LPDDR3	4GB LPDDR4
Nominal Power	1W	5/10W
Weight	18g	140g
Peak Performance	100GFLOPs	472GFLOPs

3.2.1 Data Set Description

We consider two popular ungulates (wild boar and deer) in the Tuscany region of Italy in our study. Image acquisition was conducted systematically in various areas of the region using a digital camera with 20 megapixels during different times and days of June to August.

3.2.2 Hardware Description

Edge-AI (Artificial Intelligence) consists of performing computations locally on an embedded system in real-time. Since the training process requires a lot more computational power as compared to the inference process, it cannot be performed on the embedded platform in a reasonable amount of time, but a high performance computing (HPC) system. The HPC system used for training was equipped with an NVIDIA RTX 2080Ti GPU with CUDA 10 and 64GB of DDR4 SDRAM. This GPU model features 544 Tensor Cores, an NVIDIA technology specifically designed to boost matrix multiplication performance, thus able to speed up the training process of deep learning models. For the embedded implementation of the model, to improve real-time performance, two different hardware platforms have been considered: RPi 3B+ with Intel Movidius NCS and Nvidia Jetson Nano.

3.2.3 Methodology

YOLOv3-tiny, as already introduced, makes use of only convolutional layers, making it a fully convolutional network that can accept inputs of different sizes during and after training. It can be divided into two main blocks: the first one is the feature extractor or backbone dubbed darknet-19. Its principal and fundamental role are to extract features in a hierarchical fashion starting from raw pixels coming from the input layer. Indeed, the extracted representations are later used as starting points by the other modules of the network. On the other hand, the second block of the YOLOv3-tiny architecture analyzes the produced backbone representations, and it predicts position and class of belonging of the different objects present in the input raw image.

At first, we processed raw data as shown in the data set description. The raw data are stored in PNG format, acquired from the test site. We used $n = 600$ sample images, and in order to speed up the label generation, we used an extensive and accurate version of YOLOv3 known as YOLOv3-spp, pre-trained on the COCO data set, to create a draft of the ground truth labels. Each image was resized and normalized. Using transfer learning, we started our training from a pre-trained backbone of the original model. That greatly speeds up the training, drastically reducing the number of training samples required to achieve a high level of accuracy. All training was carried out on the HPC system described above. After training, the model was deployed on the different hardware platforms presented in the hardware description section. We compared the performance of the hardware used in terms of absorbed power and frame rate. The Jetson Nano is the most performing platform, being able to reach 12 frames per second (fps), while the RPi 3B+ with Intel Movidius NCS was able to reach a 8 fps.

3.3 Back-end System

This section illustrates our back-end system. Our system consists of a LORaWAN network for data transmission between the Natech Escape devices and the back-end cloud services. Monitoring data about the status of Natech Escape devices are periodically generated and transmitted to the LoRa gateway, and to a LoRa Network Server in The Things Network (TTN). The TTN is an open platform for registration of LoRaWAN devices (IoT nodes) and gateways. It has an implementation of all needed back-end services for LoRaWAN gateway operation i.e., all functions needed for the data and the transport layer, along with all required security layer.

To get and store the data in our monitoring platform, we first need to create the device in the TTN console and register the devices. Then, we use a monitoring application that has been developed on a Docker with 4 containers. A Node-Red[2] is used to retrieve packets

[2] nodered.org

from the TTN temporary data storage and to store them in our time-series database (InfluxDB [3]). In fact, we opted for two different databases: InfluxDB and MongoDB. The first database is a scalable database for storing metrics, events, and real-time analytics while the second database is classified as NoSQL database, and commonly used because of its support for unstructured data. Therefore, the packet payload and metadata are retrieved by an application developed in a Node-RED server and stored in 2 databases:

- InfluxDB which is then used to produce Grafana[4] monitoring dashboards;
- MongoDB which stores raw data about the devices (i.e. GPS coordinates) to generate reports using python scripts.

The monitoring dashboard provided by InfluxDB and Grafana offers a useful insight to drive the experiment without high maintenance cost. The Node-Red server and the databases are running in the cloud and are accessible remotely for flexibility purposes.

4 Network Deployment and Modes of Operation

The deployment consists in a virtual fence of poles, where each pole is a full, autonomous, networked real-time animal detection and repelling system. So far, our testbed consists of 10 poles deployed through the San Rossore Park in Tuscany that leverage the state-of-the-art LoRaWAN network technology to communicate with a LoRa gateway. Each pole is equipped with two Natech Escape devices, each of them equipped with a Lora module RN2483A, xbee radio module and a PIR. All devices were calibrated at the manufacturer facilities prior to deployments. Data is pushed to the cloud in real-time. The real-time animal detection system is also mounted on the same pole of the Natech Escape devices. The activity is being transmitted from the repeller devices to the LoRa gateway in a periodic manner. We make use of a commercial gateway embedded with LoRa capabilities, developed for outdoor use. The LoRa gateway is connected to a satellite gateway that provides connectivity with the Internet and finally to the TTN network server.

Our tested operates using the TTN, its gateway forward traffic to the EU TTN network server, formatted in a predefined format (e.g. JSON, XML etc.), and then transmitted to the database system and the presentation Graphical User Interface system. Our gateway performs a Cyclic redundancy check (CRC): if CRC is valid, the gateway forwards the packet to the TTN server, otherwise, it drops the packet. Both the database and the presentation systems are Time Series based tools, which means that the storage and the presentation of the data is based on time. The back-hauling of the gateway can be obtained via satellite communication. Moreover, the devices/networks must be able to run “on their own” without any obligatory user intervention. They must be able to withstand important weather conditions, self-heal in case of environmental changes, and run for several years without human intervention.

As this implementation is critical for agriculture, we had to make sure that the repeller devices will operate for a long time before maintenance will be required. Our system uses two 3.7V rechargeable batteries with a capacity of 4200mAh as the power source and supports charging by both power adapter and a 25W solar panel. The batteries were

[3] https://www.influxdata.com/

[4] https://grafana.com/

tested every four days in order to guarantee the effectiveness of the solar power charging board.

4.1 Operational Mode

In this section, we show how the operation can be performed in two different modes namely (1) Detection through PIR and (2) Detection through Camera.

4.1.1 Detection through PIR

When a movement is detected through the PIR sensor, the SAMD21 microprocessor sends an "activity" message to the embedded system (RPi with Intel Movidus NCS or Jetson Nano) through the xbee radio. Note that, the embedded system is also equipped with the xbee radio, which makes it to be embedded with IEEE 802.15.4 capabilities. The embedded system activates the camera, then executes its deep learning algorithms to identify the target, and if an animal is detected, it sends back a message to the Natech Escape including the type of ultrasound to be generated according to the category of the animal. The "activity" message is also transmitted from the repeller device via LoRa to the LoRa gateway, which then forwards the packet to the TTN server.

4.1.2 Detection through Camera

In this case, the PIR sensor is not used and the real-time animal detection system on the embedded system is executed when the camera detects a target. Based on the category of animal detected, the embedded system sends a message, with the ultrasound waveform to be generated, to the Natech Escape through the IEEE 802.15.4 interface. Moreover, based on the operation sent to the Natech Escape device, the activity is then transmitted from the repeller devices via Lora to the LoRa gateway, which then forwards the packet to the TTN server.

5 Lessons Learnt

The most important lesson learnt is that IoT is the right technology for this precision agriculture application, and that using it makes a huge monetary difference. Moreover, as seen throughout this work, IoT technology is ready for this type of application. We strongly believe that the combination of technologies demonstrated through this work is the right one for a large number of "Smart Agriculture" applications. We hope that the present article can guide end users put together the right technical solution.

The main outcome of this work is a perfectly working end-to-end low-power wireless sensor systems, built from commercial off-the-shelf components, but also custom designed and devloped devices, such as the ultrasound repeller and detection system elements. One decade ago, Langendoen et al [22] wrote a foundational paper listing everything that went wrong in a precision agriculture deployment similar to our project, which include board failure, batteries running out etc. So what went right this time, is that the field of low-power wireless has evolved substantially, and has radically changed in that decade. IoT technology has successfully transitioned from the academic to the commercial world.

6 Conclusions

This article provided an in-depth technical description of the Natech Escape repeller device, the real-time animal detection system and lastly the back-end system. We showed the integration of a neural network to the system allowing a real time animal detection. We also showed how data is pushed to the cloud in real-time, thus making it accessible to the scientific community in real-time. We hope this article can be informative to the scientific community by providing testing feedback and limitations and also the agronomists by describing a solution that works.

References

[1] N. Squires, *"Tuscan wine makers back cull of 250,000 wild boar and deer."* 2016. *Accessed on: February 26, 2020 [Online].* Available: https://www.telegraph.co.uk/news/worldnews/europe/italy/12105887/Tuscan-wine-makers-back-cull-of-250000-wild-boar-and-deer.html

[2] A. Amici, F. Serrani, C. M. Rossi, and R. Primi, *"Increase in crop damage caused by wild boar (sus scrofa l.): the "refuge effect","* Agronomy for sustainable development, vol. 32, no. 3, pp. 683–692, 2012.

[3] S. Wang, P. Curtis, and J. Lassoie, *"Farmer Perceptions of Crop Damage by Wildlife in Jigme Singye Wangchuck National Park, Bhutan"* - In Wildlife Society Bulletin, Vol.34, No.2, pp. 359-365, 2006.

[4] B. Hamrick, T. Campbell, B. Higginbotham, and S. Lapidge, *"Managing an invasion: effective measures to control wild pigs,"* 2011.

[5] A. R. Tiedemann, T. Quigley, L. White *et al.*, *"Electronic (fenceless) control of livestock."* Res. Pap. PNW-RP-510. Portland, OR: US Department of Agriculture, Forest Service, Pacific Northwest Research Station, vol. 510, 1999.

[6] M. O. Ojo, S. Giordano, G. Procissi, and I. N. Seitanidis, *"A review of low-end, middle-end, and high-end IoT devices,"* IEEE Access, vol. 6, pp. 70 528–70 554, 2018.

[7] A. Kamilaris and F. X. Prenafeta-Boldú, *"Deep learning in agriculture: A survey,"* Comput. Electron. Agricult., vol. 147, pp. 70–90, Apr. 2018.

[8] U. J. L. dos Santos, G. Pessin, C. A. da Costa, and R. da Rosa Righi, *"AgriPrediction: A proactive Internet of Things model to anticipate problems and improve production in agricultural crops"* - In Computers and electronics in agriculture, 161, 202-213, 2019.

[9] J. Muangprathub, N. Boonnam, S. Kajornkasirat, N. Lekbangpong, A. Wanichsombat & P. Nillaor *"IoT and agriculture data analysis for smart farm"* - In Computers and electronics in agriculture, 156, 467-474, 2019.

[10] I. Mohanraj, K. Ashokumar, & J. Naren *"Field monitoring and automation using IOT in agriculture domain"* - In Procedia Computer Science, 93, 931-939, 2019.

[11] S. Giordano, I. Seitanidis, M. Ojo, D. Adami and F. Vignoli, F. 2018 *IoT solutions for crop protection against wild animal attacks.* In 2018 IEEE International Conference on Environmental Engineering (EE) (pp. 1-5). IEEE.

[12] L. Nóbrega, A. Tavares, A. Cardoso, and P. Gonçalves, *"Animal monitoring based on IoT technologies,"* in 2018 IoT Vertical and Topical Summit on Agriculture-Tuscany (IOT Tuscany). IEEE, 2018, pp. 1–5.

[13] N. Sakthipriya, *"An effective method for crop monitoring using wireless sensor network,"* Middle-East Journal of Scientific Research, vol. 20, no. 9, pp. 1127–1132, 2014.

[14] W. L. Chen, Y. B. Lin, Y. W. Lin, R. Chen, J. K. Liao, F. L. Ng, Y. Y. Chan, Y. C. Liu, C. C. Wang, C. H. Chiu and T. H. Yen, *"Agritalk: IoT for precision soil farming of turmeric cultivation,"* IEEE Internet of Things Journal, vol. 6, no. 3, pp. 5209–5223, 2019.

[15] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne, *"Understanding the limits of LoRaWAN,"* IEEE Communications magazine, vol. 55, no. 9, pp. 34–40, 2017.

[16] L. Vangelista, A. Zanella, and M. Zorzi, *"Long-range IoT technologies: The dawn of LoRa™,"* in Future access enablers of ubiquitous and intelligent infrastructures. Springer, 2015, pp. 51–58.

[17] Natech Home Page, *Accessed on: February 26, 2020 [Online].* Available: https://www.natechescape.com/

[18] E. Baccelli, C. Gündoğan, O. Hahm, P. Kietzmann, M. S. Lenders, H. Petersen, K. Schleiser, T. C. Schmidt, and M. Wählisch, *"RIOT: an open source operating system for low-end embedded devices in the IoT,"* IEEE Internet of Things Journal, vol. 5, no. 6, pp. 4428–4440, 2018.

[19] M. Ojo, D. Adami, and S. Giordano, *"Performance evaluation of energy saving MAC protocols in WSN operating systems,"* in Performance Evaluation of Computer and Telecommunication Systems (SPECTS), 2016 International Symposium on. IEEE, 2016, pp. 1–7.

[20] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, *"You only look once: Unified, real-time object detection,"* in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2016, pp. 779–788.

[21] J. Redmon and A. Farhadi, *"YOLOv3: An incremental improvement"* 2018, arXiv:1804.02767. *Accessed on: February 26, 2020 [Online].* Available: https://arxiv.org/abs/1804.02767/

[22] K. Langendoen, A. Baggio, and O. Visser, *"Murphy loves potatoes: Experiences from a pilot sensor network deployment in precision agriculture*," in Proceedings 20th IEEE international parallel & distributed processing symposium. IEEE, 2006, pp. 8–pp.