




Article

An IoT Architecture for Continuous Livestock Monitoring Using LoRa LPWAN

Lorenzo Germani , Vanni Mecarelli, Giuseppe Baruffa * , Luca Rugini 
and Fabrizio Frescura 

Department of Engineering, University of Perugia, via G. Duranti 93, 06125 Perugia, Italy;
lorenzo.germani@unipg.it (L.G.); vanni.mecarelli@unipg.it (V.M.); luca.rugini@unipg.it (L.R.);
fabrizio.frescura@unipg.it (F.F.)

* Correspondence: giuseppe.baruffa@unipg.it; Tel.: +39-075-585-3627; Fax: +39-075-585-3654

Received: 30 October 2019; Accepted: 22 November 2019; Published: 1 December 2019



Abstract: The Internet of Things (IoT) architecture is quickly becoming popular even outside of its originating scenario of home automation. This paper reports the design, implementation, and performance of an IoT hardware and software architecture conceived for the continuous monitoring of livestock located in barns and during grazing. We have adopted the LoRa low power wide area network (LPWAN) technology to cover the diverse environments, and a suitable configuration of web services to perform data storage, analysis, and visualization. Since the LoRa LPWAN (LoRaWAN) medium access control (MAC) layer does not provide a listen-before-talk (LBT) mechanism, we propose a custom MAC layer with LBT-based carrier-sense multiple access with collision avoidance (CSMA/CA). The devised system has been implemented using off-the-shelf hardware, and its performance has also been estimated with the help of a C++ event-based simulator. The preliminary results of our HW implementation on the field confirm the stability of the conceived system and its reliability.

Keywords: LoRa; Internet of Things; precision livestock farming; wireless sensor network; listen-before-talk

1. Introduction

With the diffusion of the Internet-of-Things (IoT) paradigm [1], objects are being connected to the Internet, thus realizing a global network of connected things. This new concept is being used in several scenarios, such as in smart cities [2,3], industrial manufacturing [4,5], healthcare [6,7] and transportation [8,9]. Another application scenario is in rural areas [10,11], and specifically on smart livestock farming [12], over which we focus our work.

In rural areas, it is important to have sensor nodes that can operate for long time without human intervention, and capable to transmit data over a wireless communication channel [12]. Low Power Wide Area Networks (LPWANs) can satisfy the data communication requirements typical of this environment, which are flexibility, resistance to multipath and other types of interference, and wide coverage. ZigBee, SigFox, NB-IoT, and LoRa are among the most frequently used technologies for the implementation of rural LPWANs [13,14]. With respect to the other competing technologies in the LPWAN scenario, LoRa does not require a third-party infrastructure for channel access, but it can act in an almost decentralized way [13,14]. This means that the LoRa end devices (EDs) do not pay fees for communicating and they must only adhere to regulatory spectrum requirements.

In this work we use LoRa, since it also offers good performance in terms of battery life, at a relatively low cost, and it is resilient to transmission errors while maintaining a wide coverage [15]. Moreover, the typical coverage range of LoRa varies from 2–5 km in urban areas to 20–25 km in

suburban or rural environments [15], making this technology viable for both small and medium-scale applications. It has also been shown that a single LoRa gateway can manage a large number of EDs with little maintenance and easy deployment [5].

LoRa is a proprietary wireless communication standard developed by Semtech, based on a chirp spread spectrum (CSS) modulation [16,17]. The physical layer (PHY) uses chirp waveforms with different chirp rate, to maintain orthogonality among different classes of users, and with different initial frequency, to convey the actual information message [15]. Channel coding is also used, with interleaving and Hamming block codes. Although the details of the LoRa PHY are not public, several authors have tried and succeeded to reverse-engineer, almost completely, its PHY communication techniques [18].

On the other side, LoRaWAN is an open MAC protocol, built on top of the LoRa PHY, that is specifically designed for low-powered devices operating over long-range wireless links. Even though LoRaWAN is commonly used in deploying LoRa networks [19], in this work we have decided to implement a custom MAC layer in order to have more control over the architectural design and to remove some features of the LoRaWAN layer [20] that are not used in our application, such as the large header size when short payloads have to be delivered, and the packet acknowledgments managed by a LoRa server. Continuous and near-real-time monitoring of livestock, in particular of dairy cattle, can help to provide information vital to study the animal behavior and to prevent pathological states that have a negative impact on livestock health [21,22].

In particular, our proposed architecture aims at analyzing, almost in real-time, the status of dairy cattle in a cowshed or during grazing by means of sensors embedded in smart collars. By the way, this approach may be extended also to other kinds of livestock, such as beef cattle, swine, ovine, and even equine.

In this paper we first describe the overall architecture of the proposed livestock monitoring scheme. With respect to other similar systems described in the literature, our architecture uses a simpler MAC layer to alleviate the need for a central server to manage the acknowledgments, and also comprises the data storage platform through which the clients may perform their data analysis tasks. The main purpose of the proposed architecture is to collect, store, and analyze several parameters typical of the cattle livestock scenario, related to the health of each head of the herd and to the habitat where cattle are kept. Then, we show in detail how we designed and implemented the different objects that implement and overview the sensor network. We give a description in terms of both functional hardware and software and firmware used to control the sensors and the system controllers. We will also describe the implementation of a MAC-level simulator that we have designed to test and optimize some parts of the proposed sensor network. Eventually, we will conclude our paper with the results of several simulations, as well as with some preliminary results on the performance of the real system.

2. Related Works

In the literature there are several studies related to object and environmental monitoring by means of LoRa, concerning agriculture and livestock farming [19,21,23–25], health [26], forestry [12,27], smart cities [2,28], industrial [4,5,29] and maritime [9] applications.

For what concerns agriculture and livestock farming, in [23] the authors present the WAZIUP IoT system used, among other things, also for cattle management in rural Africa, which is based on LoRa for long-range connectivity. This research has established some low-cost designs for both the sensor devices and the gateways, which are based on Arduino and Raspberry Pi. The goal is to avoid cattle theft, by monitoring position and speed of the animals, but no information is provided on other biomarkers such as body temperature and motion. Agriculture monitoring is the target of the work described in [24]. In particular, [24] focuses on a sensor network devoted to the monitoring of environmental parameters in the field (temperature, humidity), and to a parallel actuator network that is used to remotely operate on vineyards for automated irrigation. The paper [24] does not provide details of the used hardware, nonetheless [24] adopts a LoRaWAN network and a commercial service

provider for cloud-related tasks. In [25], vital signs of grazing cattle are collected via a LoRa network. The authors of [25] design portable nodes built around the STM32 microcontroller, and equip them also with accelerometers and global navigation satellite system (GNSS) positioning. The gateway uses a Raspberry and a dedicated multi-channel LoRa chip. The investigation is mainly focused on the coverage radius and packet loss: Some findings in [25] show that a maximum distance of 900 m can be covered in an urban scenario with many buildings, which rises to 2 km on open roads. LoRa is also used in [21] to control environmental parameters such as NH_3 concentration and temperature in pig houses. The system in [21] collects the gaseous concentrations and these data feed a neural network, which helps regulating the ventilation of the hog lots. The authors of [21] have also verified the coverage radius of their system, in both indoor and outdoor conditions, finding a low packet loss rate even at 4.5 km distance in outdoor scenarios and on five story buildings in indoor scenarios.

There are also a number of commercial solutions proposed for the deployment of livestock monitoring networks. For these solutions, the information is limited to what reported in data sheets and web sites of the related companies. Cattle Watch [30] uses an undisclosed type of LPWAN for cattle tracking and virtual geo-fencing in unpopulated areas, whereas the gateway relays the data to the data collection system by means of a cellular, satellite, or NB-IoT network. Cattle Traxx operates similarly, although their technological platform is LoRa-based [19]. Sodaq [31], instead, proposes a smart cow collar that adopts a full LoRaWAN implementation for cattle tracking and activity detection. Their system is powered by a small solar cell, thus allowing extended deployment periods before servicing; however, they do not offer a proprietary cloud solution but they rely on external service providers. Allflex SenseHub Beef [32] is another solution for short range (<500 m) cattle monitoring, which is based on IEEE 802.15.4 communication protocols: With this system, however, only a limited amount of information is transmitted from the nodes, to improve battery lifetime. Finally, Moocall [33] is an animal tracking and estrus detection system which uses GSM networks for information relay from sensors to data centers.

In [26] a study of LoRa propagation in indoor scenarios has been conducted, for the purpose of human health monitoring. The study in [26] has found that the packet loss rate can be minimized within a coverage radius of about 200 m, even if the experiment involved only a single node and base station.

Tree plantation monitoring with LoRa is explored in [27]. In that study, the gateway antenna is located at different heights, and the nodes transmit (with limited power) information such as temperature, humidity, soil moisture, and so on. The experiment covered limited distances (less than 500 m) and was mainly devoted to explore the propagation conditions due to the presence of trees and their foliage. The results in [27] show that there is some dependence on the antenna elevation when distances are large. LPWANs for tree farming are also used in [12], where the authors have shown that the height of the receiving antenna has a large impact on the packet loss rate, with the received signal strength indicator (RSSI) depending more on the distance rather than on the adopted spreading factor.

Smart city monitoring is the subject of [2]. In this work, environmental quality parameters (CO_2 concentration, luminance, temperature, etc.) in the city of Bologna are monitored with a LoRa network. The scenario is that of a densely populated urban area, with coverage radius of 1–2 km. The paper [2] focuses on the modeling of the path loss and on the adaptation of this model to a C++ network simulator, with the purpose to predict the performance of the network when the number of installed nodes scales up. The development of a link-level simulator, conceived for the ns-3 network simulator tool, has been investigated in [28], where it is found that a single LoRa gateway, obeying to the spectrum usage restrictions, is able to manage up to 15,000 EDs in smart cities, with a small packet loss rate.

In [5], the authors investigate the use of LoRa for tracking trolleys in an industrial scenario. Trolley positioning may take place either indoor or outdoor, at a maximum distance of about 500 m from the gateway. First, [5] models the propagation scenario and the LoRa communication performance with real devices. Then, [5] establishes a Python-based simulation framework to evaluate the maximum

load that can be sustained by the network under stressing conditions. The simulations in [5] show that a single gateway could be engaged up to 6000 nodes, still keeping an acceptable performance in terms of packet loss rate. Another evaluation of the feasibility of LoRa transmission in an industrial scenario has been carried out in [29], where the authors experimented time-slotted channel-hopping to respect the limitations imposed by spectrum access regulations and to improve the typical node latency down to less than one minute, which is deemed to be acceptable for most industrial control applications. The authors of [4] also proposed the adoption of an enhanced flexible node to supplement gateway functionalities and to act as a range extender. In [9], maritime communication with boats is established using LoRa, in particular for lightweight boats navigating near the coastal line. The coverage extended up to 4 km far, with a limited amount of consumed power on the EDs, as low as 140 mA at 5 V.

In the above presented LoRaWAN approaches [2,4,5,12,21,23–29], the adopted LoRaWAN MAC layer does not include a listen-before-talk (LBT) mechanism that tries to prevent as much as possible packet collisions among nodes [34]. On the contrary, our paper proposes a custom MAC layer that includes LBT-based carrier-sense multiple access with collision avoidance (CSMA/CA), to be incorporated with the physical layer of LoRa. The proposed CA mechanism is based on a random retransmission time that randomizes the access of the nodes to the wireless medium. As a first consequence, the proposed approach allows to reduce the number of collisions with respect to LoRaWAN. As a second consequence, the use of LBT allows to overcome the limits imposed by regulatory bodies on the effective duty cycle of channel occupation [35]: Therefore, our approach can transmit more often than a regular LoRaWAN network. Due to the above-mentioned features, our approach allows a reduced packet loss rate (with respect to LoRaWAN) for a fixed number of EDs, or, alternatively, our approach allows an increased number of EDs (with respect to LoRaWAN) for a fixed packet loss rate.

3. System Architecture

The system is designed as shown in Figure 1. At the base there are the EDs, whose purpose is to collect raw data from one or more locally attached electronic sensors, to provide some preprocessing for reducing the produced data rate, and to send preprocessed data to one or more gateways (GWs) using the LoRa PHY on the wireless channel. The GWs receive the data from the EDs, conditionally apply some further preprocessing to the data, provide a small amount of nonpersistent storage to overcome temporary access problems to the server, and, finally, communicate via a reliable and secure TCP/IP connection with the server. The application server (AS) provides dedicated application programming interfaces (API) for receiving and uploading the data coming from the EDs through the GWs, and also provides long-term persistent and reliable storage for the data. Differently from the LoRaWAN architecture, our solution does not use a centralized server to manage acknowledgments, which are controlled by the GWs in a decentralized way, and uses collision detection policies to allow for an increased number of EDs. Using the same API, web-based graphical user interfaces (GUI) or other authenticated clients can download the stored data for visualization or analysis purposes.

3.1. Application Server

The hardware of the server is a DELL PowerEdge R440 [36] with an Intel Xeon Silver 4110 CPU with 8 cores and 16 threads, 32 GB of RAM, and 1675 GB of disk storage in RAID 5. As for the software configuration, the whole system runs on Ubuntu Server 18.04 LTS, with a Linux-Apache-PHP-PostgreSQL (LAPP) backend using Apache 2, PHP 7, and PostgreSQL 11.

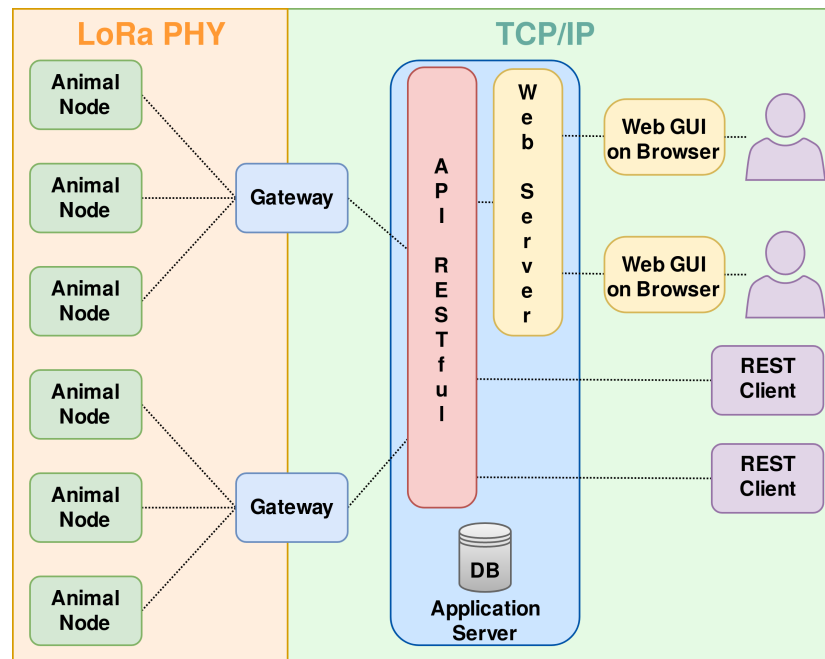


Figure 1. Overall architecture of the system.

3.1.1. API Description

The AS uses a representational state transfer (REST) API on the HTTP interface [37]. This choice is motivated by the type of stateless information that is sent by the sensors, and by the ubiquity of REST in modern IoT platforms for data retrieval in post-processing tasks [38]. The main tasks accomplished with REST are:

- **Data uploading:** The sensor data aggregated on the GW are uploaded via a POST request to the HTTP service. The request carries the sensor identifier, the GW identifier, the ED identifier, the data value, and an access token. If the data are successfully uploaded, the service returns HTTP code 200 (OK), to indicate success. Otherwise, HTTP code 203 is returned to indicate a duplicate data insertion, and HTTP code 403 to denote an unauthorized action; other HTTP codes can also be used, following the standard numbering.
- **Data downloading:** The client can request stored data values over a specified time interval or from some moment in the past up to the current time. The request is done via a POST command and comprises the sensor identifier, the start and stop times of the interval or the time duration in seconds, and an access token. If the command is executed with success, the data are returned in JSON format as an array of couples, where each couple stores the data sample value and the data sample time.

3.1.2. Database Storage

For server storage of the captured data, a regular SQL-based database has been chosen. Also other classes of databases, such as those based on the NoSQL concept, have been considered in the design phase. In other application scenarios, it has been demonstrated the superiority of NoSQL over SQL, especially when the underlying exchange data rate is high [39]. However, in our particular architecture setup, the additional flexibility and speed provided by the NoSQL databases would have not carried any particular advantage over a classic SQL database. Instead, the existence of well-established frameworks for the control and management of SQL databases has been evaluated to better suit the data collection, storage, and analysis purposes of the proposed architecture.

3.2. Gateways

The GW hardware is composed by different devices and sensors, which are all controlled by a Raspberry Pi 3B embedded PC, whose purpose is also to coordinate and manage the data and control communication with the server. Internet access on the GW is granted by the on-board Wi-Fi/Ethernet connection of the Raspberry or by a 3G/4G USB modem. The Raspberry runs the Raspbian Stretch 9.9 OS, with a number of services automatically launched during boot and managed by the *systemd* daemon. Such services have the duty to collect the data from the attached sensors, to receive the data from the visible ED with the LoRa network, to keep an active Internet connection with the server, and to reliably send the collected data to the server. LoRa network access is guaranteed by an Adafruit Feather M0 RFM95 [40], which continually listens for messages arriving from EDs. Thus, the GW is not a LoRaWAN-capable GW. Although this might seem a limitation, indeed represents an advantage in our architecture, for two reasons. First, differently from the chips used in LoRaWAN GWs, it has a lower cost. Secondly, the MAC layer in the GW can be customized and optimized for our particular application. Our architecture entails two different versions of GWs, for either indoor or outdoor usage. The indoor-use GW is designed for installation in sheltered areas such as barns and cowsheds, and is more oriented towards dairy cattle livestock scenarios. Instead, the outdoor-use version is more specific for arrangement in open areas such as paddocks and pasture lands, and it is designed for beef cattle livestock scenarios.

3.2.1. Indoor Version

The indoor GW is conceived for monitoring several important physical parameters typical of the shed environment, such as temperature, relative humidity, illuminance, carbon dioxide (CO₂) and ammonia (NH₃) concentration [41–43].

Sensor data acquisition is managed by an Arduino Mega 2560 board, built around a 16 MHz ATmega2560 microcontroller, with 16 analog inputs and 54 digital input/output pins. This controller board queries the sensors attached to the digital or analog pins (which use different communication protocols and logic levels), reads and parses the raw data from the sensors, and formats the data for UART-based communication with the Raspberry using a common message structure. Another purpose of the rugged Arduino Mega board is also to decouple the sensor electronics from the fragile Raspberry on-board electronics. The used sensors and the overall architecture of the indoor GW are shown in Figure 2. The adopted sensors are:

- Aosong AM2321 [44]: Queried on the I²C bus, records the external temperature with a resolution of 0.1 °C and accuracy of ± 0.5 °C in the range from -40 °C to 80 °C, and the relative humidity with a resolution of 0.1 % and accuracy of ± 3 %.
- Adafruit TLS2561 (GY-2561) [45]: Queried on the I²C bus, records the illuminance in a range from 0.1 lx to 40000 lx with a dynamically varying resolution.
- DFRobot SEN0219 [46]: Queried on the analog pins, records the CO₂ concentration with an accuracy of about ± 50 ppm in the range from 0 ppm to 5000 ppm;
- Winsen MQ137 [47]: Queried on the analog pins, records the NH₃ concentration within a range from 5 ppm to 500 ppm.

The Feather M0, before passing the data over UART to Raspberry, formats them according to the same structure used by the Arduino Mega.

The Raspberry manages the UART communication between the Feather M0 and the Arduino Mega in the same way, listening on the serial port for a known message structure. After detecting a message, it decodes and prepares a new message to send to the server via the API. The Raspberry also implements a buffering strategy for the messages: In case of Internet connection problems, typical in remote rural areas, unsent messages are stored locally and resent once the connection is resumed. The system is equipped by a 25 W, 220 V AC to 5 V DC power supply. All the equipment is kept in a waterproof electrical cabinet; for providing a cooling support for the devices and permitting a fresh

air flow on the sensors, a fan is mounted in front of the sensors, pushing the air inward, as shown in Figure 3.

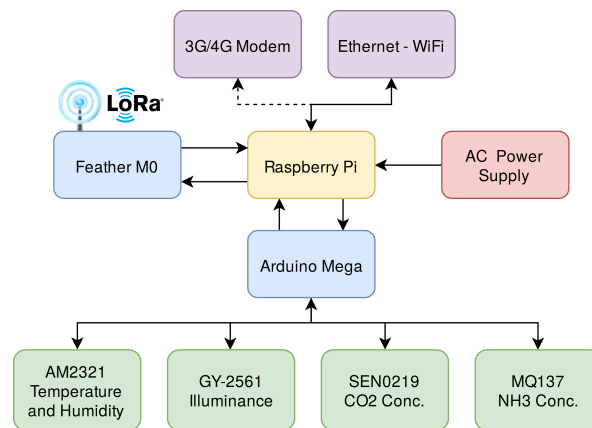


Figure 2. Overall architecture of the indoor gateway (GW).

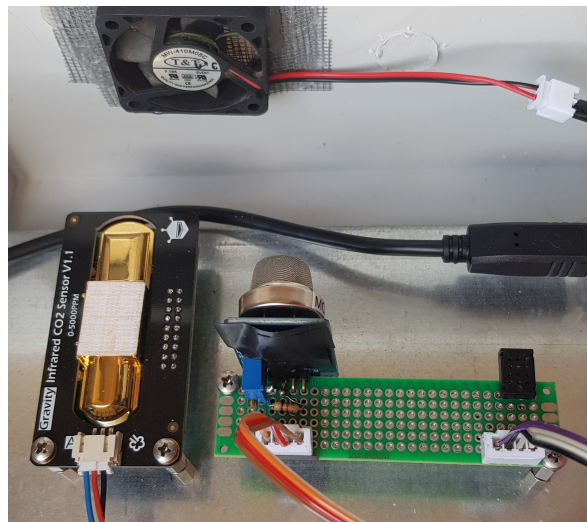


Figure 3. Sensors layout inside the indoor GW cabinet.

3.2.2. Outdoor Version

The main purpose of outdoor GW is to manage EDs in remote areas, far from the shed, directly on the pasture land. Due the open field scenario, only weather parameters (temperature and relative humidity) are collected, for purposes of correlation with the animal health status.

The electrical power is provided by a 20 W solar panel, accumulated in a 12 V 7 Ah lead-acid battery: Charging cycles and load protection are managed by a solar charger controller. The battery current and voltage are also monitored by an INA219 based sensor [48], for purposes of power management. With the typical power consumption of the system, the battery is able to maintain the GW operative for 2–3 days in absence of sufficient solar illumination. To allow for a low power consumption policy, the sensors are directly controlled by the Raspberry via its general-purpose input/output (GPIO) pins. The only sensor used, in this case, is the AM2321 sensor for temperature and humidity measurement, mounted on the exterior of the cabinet and covered by a semipermeable external coating. The architecture of the outdoor GW is shown in Figure 4.

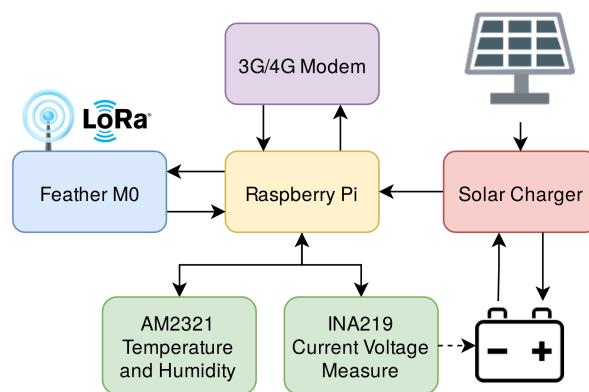


Figure 4. Overall architecture of the outdoor GW.

3.3. End Devices

3.3.1. HW Implementation

The goal of the ED is to gather as much information as possible through several sensors located on the collar worn by cattle. There are several possible parameters related to the animal health that can be monitored [22]. In this project, we have chosen GNSS position, acceleration, rotation, and body temperature. In addition, other parameters related to the operation of the sensor themselves can be captured, such as the accelerometer temperature, the battery voltage, and the RF link strength. The ED hardware layout can be customized to different scenarios: For example, if the animals are always kept in the shed, GNSS positioning is less significant and thus can be disabled for improving battery lifespan, while for grazing cattle the knowledge of the position plays an important role [25].

Since the ED is carried by the animal, it is necessary to avoid disturbing and stressing the animal as much as possible. This also means that the ED must be enclosed by a robust and waterproof package. Thus, it is important to pursue a low power consumption strategy, so as to obtain a longer battery lifespan: Under a typical operating condition (with the GNSS disabled), the battery time is of about 5 weeks. The overall architecture of the ED is shown in Figure 5.

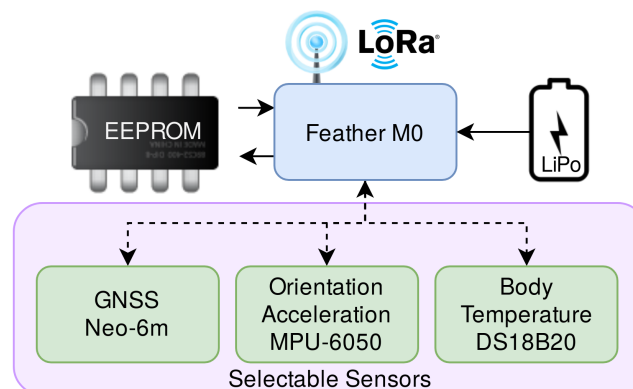


Figure 5. Overall architecture of the animal end device (ED).

There are three functional blocks common to all kind of EDs: An Adafruit Feather M0 RFM95 microcontroller with embedded LoRa modem, a LiPo rechargeable battery, and an EEPROM memory. The microcontroller takes care of gathering the data from sensors, of pursuing a low power policy (e.g., powering off the sensors when they are not needed), of packet buffering strategies, of the LoRa channel access with CSMA/CA, and of sending the data to the GW. The sensors adopted in the ED are:

- u-blox Neo-6m [49]: GNSS receiver for precise geographic positioning that communicates NMEA messages via UART;

- InvenSense MPU-6050 [50]: Queried on the I²C bus, records accelerations and rotations;
- WINOMO DS18B20 [51]: Queried via a digital pin using One Wire protocol, measures body temperature with an accuracy of ± 0.5 °C in the range from -10 °C to 85 °C.

Figure 6 shows the assembled ED with the microcontroller, sensors, an EEPROM, and a LiPo battery.

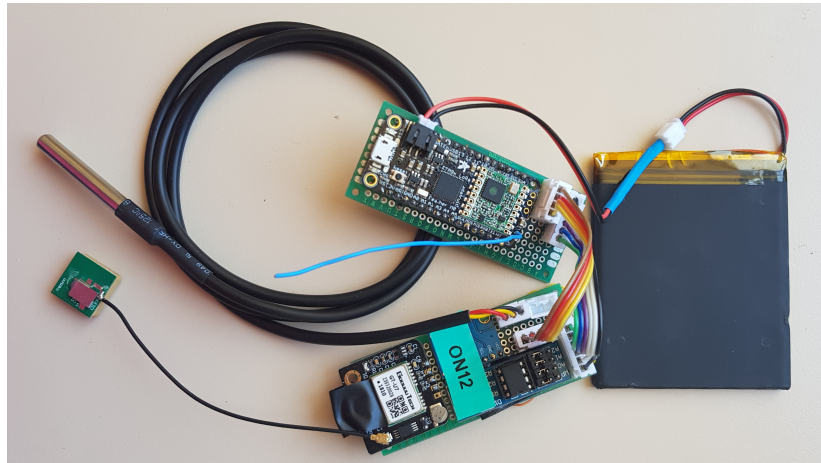


Figure 6. The assembled ED.

Table 1 reports the main parameters collected by each entity described above.

Table 1. Classification of devices by purpose.

| Device Class | Observed Parameters |
|--------------|---|
| Indoor GW | Temperature, relative humidity, illuminance, CO ₂ and NH ₃ concentrations |
| Outdoor GW | Temperature, relative humidity |
| ED | GNSS positioning, acceleration, rotation, body temperature |

3.3.2. Custom Access Protocol

The access protocol is responsible of medium access with CSMA/CA, data compression, data encryption with PSK AES-128 standard, packet acknowledgment, and retransmissions.

The LoRaWAN header minimum size is of 13 bytes and the maximum is of 27 bytes [15]. Instead, the header designed for our proposed protocol is of 4 bytes: This protocol discards many additional features implemented in LoRaWAN, which would not be used in this application.

More details about medium access strategies will be given in Section 4.2.

4. PHY/MAC Level Simulator

The performance of the proposed LoRa architecture has also been evaluated by means of a physical and MAC layer system simulator. The simulator, written in C++, implements a time-based event queue and adopts a simple physical layer model where waveforms are associated to data packets, as well as a MAC level that operates on byte packets. The *evsObject* represents the base class of the objects involved in the time-based event simulation; from this base class, several other classes are created with inheritance. Every object is characterized by an *handler* method, which is invoked whenever an event occurs for that object. There are three main subclasses, which are used to characterize different actors of the simulation system. They are:

evsChannel: This class represents the communication channels, upon which data delivery may take place as a byte packet exchange;

evsDevice: This class represents the communication devices, characterized as a finite state machine with MAC arbitration policies;

evsInterface: This class acts as a connector between the communication devices and the communication channels. For instance, a device may act as bridge between different communication channels by incorporating multiple interfaces.

From the three main subclasses, several objects are then created:

evsInetChannel: This object represents a generic Internet data channel with reliable packet delivery (e.g., TCP/IP);

evsRadioChannel: This object represents a generic radio communication channel characterized by attenuation and packet losses;

evsLoRaEndDevice: This object represents a LoRa ED;

evsLoRaGateway: This object represents a LoRa gateway;

evsDBServer: This object represents the server hosting the database;

evsInetInterface: This object represents the interface towards an Internet communication channel;

evsLoRaRadioInterface: This object represents the interface towards a radio communication channel.

The data transported during the simulation events are represented by the base class *evsPacket*, which has the two following subclasses:

evsInetPacket: A data packet delivered through the reliable Internet communication channel;

evsLoRaRadioPacket: A data packet unreliably delivered through the radio communication channel with the LoRa PHY.

Finally, the *evsEvent* class is used to denote an object event, which will be handled by the proper handler in the destination object: All the events are queued in a time-sorted list. The simulation engine, then, starts to handle the events in the list, from the first to the last. The simulation is organized as graphically described in Figure 7: Several LoRa EDs communicate with the LoRa GWs by means of the LoRa radio channel model. The GWs, instead, communicate with the database server by means of the reliable Internet channel. The data are exchanged between the blocks under the form of either LoRa radio packets or Internet packets.

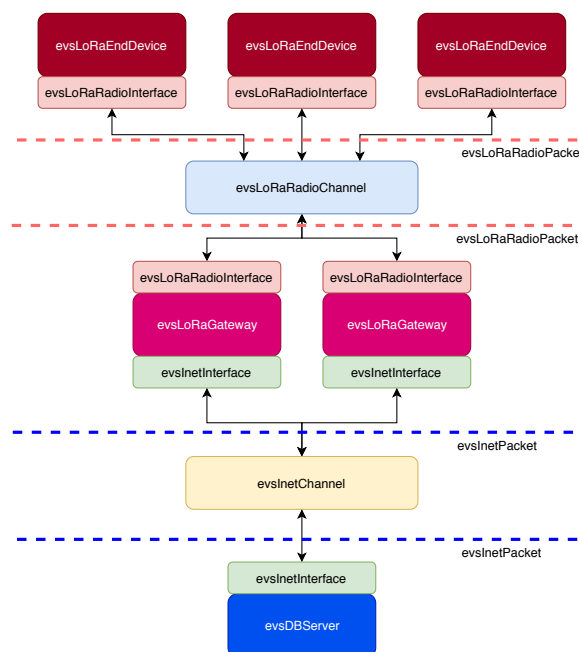


Figure 7. Graphical example of the C++ simulation setup.

4.1. Physical Layer Model

For the simulation of the physical layer, we have adopted a path loss model characterized by a received power [52]

$$P_{RX} = \frac{P_{TX} G_{TX} G_{RX}}{L_a L} = \frac{P_{TX} G_{TX} G_{RX}}{L_a L^{ref}} \left(\frac{d^{ref}}{d} \right)^\alpha, \quad (1)$$

where P_{TX} is the transmitted power (in W), G_{TX} and G_{RX} are the transmit and receive antenna gains, respectively, L_a represents additional losses due to connectors, cabling, antenna misalignment, etc., and L is the path loss. The path loss depends on d , the distance between the transmitter and the receiver (in m), on L^{ref} , which is the path loss at distance $d^{ref} = 1$ m, and on α , the attenuation exponent of the path loss. If we express (1) in dB, then the logarithmic path loss is given by

$$L_{dB} = 10\alpha \log_{10}(d/d^{ref}) + L_{dB}^{ref}, \quad (2)$$

so that the received power (in dBm) can be calculated as

$$P_{RX_{dBm}} = P_{TX_{dBm}} + G_{TX_{dB}} + G_{RX_{dB}} - L_{dB} - L_{a_{dB}}. \quad (3)$$

In order to find the channel parameters α and L^{ref} , we have performed a data transmission experiment using the hardware described in Sections 3.2.1 and 3.3.1. The LoRa GW has been installed in the premises of the Engineering Campus of the University of Perugia, at an height of $h_{GW} = 5$ m with a dipole antenna with gain $G_{RX} = 3$ dB. The LoRa ED is equipped with a quarter wave whip antenna, which can be considered omnidirectional ($G_{TX} = 0$ dB) because the antenna orientation is random. Additional losses (due to antenna misalignment, receiver cable connections, and so on) have been considered for an amount of $L_a = 1$ dB. The LoRa ED has been configured to transmit at a power $P_{TX} = 20$ dBm on the carrier frequency $f = 868.0$ MHz, with a spreading factor $F_S = 7$, a bandwidth $B = 125$ kHz, and a code rate $R_C = 4/5$. The ED has been carried to a maximum operating distance of 2 km at walking speed, traversing various types of transmission environments, such as urban, suburban, and hilly, both in LOS and NLOS conditions. The ED transmitted its GNSS coordinates every 5 s, and the RSSI at the GW was recorded for every received data packet.

Figure 8 plots the measured path loss L as a function of the distance d (in logarithmic scale), as obtained from (2) and (3). After least squares (LS) fitting of the path loss as a function of the logarithmic distance, we have found the average attenuation exponent to be $\alpha = 3.73$ and the reference path loss to be $L^{ref} = 33.39$ dB.

From Figure 8 it appears that some shadowing applies to the total path loss, since the measures are not concentrated along the LS fit line. A better model that also takes into account shadowing [52] can be obtained as

$$\tilde{L}_{dB} = L_{dB} + X_{dB}, \quad (4)$$

where X_{dB} is a Gaussian random variable with zero mean and standard deviation σ that models log-normal shadowing: From our measurements, using (2), the shadowing has a standard deviation $\sigma = 10.60$ dB. Figure 8 also shows the curves $L_{dB} \pm \sigma_{dB}$ due to shadowing. Our experiment shows that the measured path loss is within the range $L_{dB} \pm \sigma_{dB}$ for 71.7% of the times. As a term of comparison, in [2] the authors obtain for a dense urban environment a path loss exponent variable between 3.25 and 3.84, with a mean square error between the actual loss and the predicted one of 7.28 dB and a reference path loss of 31 dB.

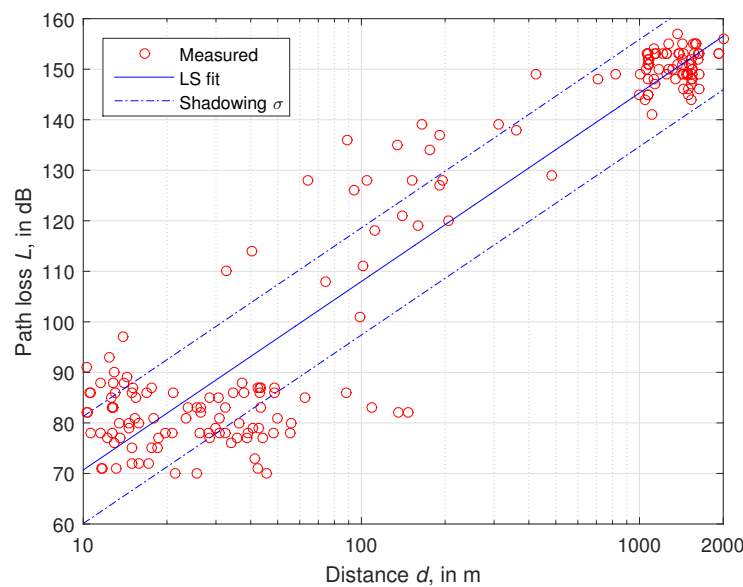


Figure 8. Least squares (LS) fitting of the measured received power vs. the distance, plotted as value of total path loss L .

4.2. MAC and Transport Layer Model

Differently from the ALOHA access scheme used in LoRaWAN, in our implementation the medium access is performed with a listen-before-talk (LBT) strategy [53] at the physical layer, so as to avoid unnecessary collisions between radio packets, jointly with an acknowledgment (ACK)-based retransmission policy implemented at the transport layer. CSMA-CA is implemented in an unslotted way: If the channel is sensed to be idle, i.e., no other LoRa ED is transmitting, then the current transmission happens. Otherwise, if the channel is occupied, a random backoff time is calculated before the transmission attempt takes place. This backoff time is randomly chosen within a contention window (CW), beginning right after the transmission decision and with duration T_{CW} . If the channel is still occupied, then T_{CW} is doubled, and the transmission attempt delayed again. This procedure is repeated several times, each time with a doubled T_{CW} , up to a maximum value of N_{CSMA} transmission attempts. When the maximum number of attempts is reached and the transmission has not been finalized, the radio interface gives up sending the data packet.

At the transport layer, a further acknowledgment mechanism is also implemented. If enabled, the transmitter considers a successful packet delivery only if an ACK is received: up to N_{ACK} retransmission attempts are performed in this layer. This entails an effective number of $N_{CSMA}N_{ACK}$ possible retransmissions for a single packet.

5. Simulation Results

The deployment of a full scale LoRa network takes some time, and the refinement of the transmission parameters to improve the efficiency of the whole network can be cumbersome if done on the actual devices. Thus, our purpose is to determine, by means of simulations, how parameters such as the transmission period, the number of EDs, the use of CSMA/CA and LBT, and the acknowledgments impact on the packet delivery rate and on packet losses. Channel occupation and transmission overhead, as well, may be investigated in depth thanks to the simulation environment. Due to the great number of parameters that can be explored in a simulation, we have chosen to restrict the number of possible scenarios by keeping some of them unchanged in every simulation. Table 2 lists the parameters that are kept fixed in all simulations, whereas Table 3 presents the parameters that have been varied in each simulated scenario. The coverage area of size $W_A \times H_A = 200 \text{ m} \times 200 \text{ m}$ has been populated with the selected number of EDs, whose location is uniformly and randomly chosen within

this area at an average height of 1.5 m. The GW has been placed at the center of the coverage area, at an height of 5.5 m above the average height of the EDs.

In our simulator implementation, we have considered a packet to be correctly received if (i) the received power is above the receiver sensitivity threshold S_{RX} and if (ii) there are no collisions with other packets transmitted from other devices. The second condition can be detected in the event-based simulation by letting the receiving device enter an “RX” state upon the reception of any packet. Then, a decision event is issued at the end time of the packet, and a positive decision is taken if no other packet has been detected, in the meanwhile. Moreover, the receiver sensitivity S_{RX} has been inferred from the values given in (Table 10 of [16], pp. 20–21), which depend on the frequency, bandwidth, and spreading factor selected in the receiver. The CW duration has been chosen as $T_{CW} = 400$ ms, which is roughly the duration of a typical LoRa radio packet in our selected configuration ($B = 125$ kHz).

Table 2. C++ simulation fixed parameters.

| Parameter Description | Parameter Value |
|--|---|
| LoRa ED payload size | $N_{PL} = 236$ bytes |
| CW duration | $T_{CW} = 400$ ms |
| Spreading factor | $F_S = 7$ |
| Coding rate | $R_C = 4/5$ |
| LoRa ED transmission power | $P_{TX} = 20$ dBm |
| LoRa ED antenna gain | $G_{TX} = 0$ dB |
| Radio channel path loss exponent | $\alpha = 3.73$ |
| Radio channel reference path loss | $L^{ref} = 33.39$ dB |
| Radio channel shadowing std. deviation | $\sigma = 10.60$ dB |
| Additional losses | $L_a = 1$ dB |
| Coverage area size (width and height) | $W_A \times H_A = 200 \text{ m} \times 200 \text{ m}$ |
| Number of LoRa GWs | $N_{GW} = 1$ |
| Number of database servers | $N_{SRV} = 1$ |
| LoRa GW antenna gain | $G_{RX} = 3$ dB |
| Receiver sensitivity | S_{RX} as in (Table 10 of [16], pp. 20–21) |
| Duration of the network activity | 2 days |

Table 3. C++ simulation variable parameters.

| Parameter | Parameter Value |
|-----------------------------------|----------------------------------|
| Bandwidth | $B \in \{125, 500\}$ kHz |
| Number of LoRa EDs | N_{ED} from 10 to 1000 |
| Transmission period | $T_{ED} \in \{300, 600, 900\}$ s |
| Number of ACK retransmissions | $N_{ACK} \in \{0, 1, 3, 5\}$ |
| Number of CSMA/CA retransmissions | $N_{CSMA} \in \{0, 1, 3, 5\}$ |

5.1. Global Average Success Rate

In first place we analyze the global average success rate, defined as

$$\bar{\rho}_{SUCC} = \frac{1}{N_{ED}} \sum_{i=0}^{N_{ED}-1} \rho_{SUCC_i}, \quad (5)$$

where ρ_{SUCC_i} is defined as the success rate on the i -th ED, given by

$$\rho_{SUCC_i} = \frac{N_{SUCC_i}}{N_{SUCC_i} + N_{FAIL_i}} = \frac{N_{SUCC_i}}{N_{TOT_i} - N_{RETR_i}}. \quad (6)$$

In (6), N_{SUCC_i} is the number of packets correctly received by the i -th entity, to which they were addressed: if the acknowledgment protocol is enabled, then the success condition requires that the original transmitter also receives the ACK back. The number of failed packets is $N_{FAIL_i} = N_{TOT_i} -$

$N_{\text{SUCC}_i} - N_{\text{RETR}_i}$, and it denotes the number of packets that have not been successfully received by the entities to which they were addressed, either with or without acknowledgment: N_{TOT_i} represents the total number of attempted packet transmissions, and N_{RETR_i} is the total number of retransmissions of packets due to either collisions or missed ACKs.

Figures 9–11 show the average success rate for different transmission periods $T_{\text{ED}} = \{300, 600, 900\}$ s of all the simulated combinations: We have chosen $B = 125$ kHz, $N_{\text{CSMA}} = \{0, 1, 5\}$, and $N_{\text{ACK}} = \{0, 1, 5\}$, while the simulation results obtained with the other parameter values have not been plotted for sake of clarity, after considering that the results are very close to the presented ones.

The triangle-marked lines represent the absence of the acknowledgment mechanism ($N_{\text{ACK}} = 0$), the square-marked lines represent $N_{\text{ACK}} = 1$, while the circle-marked lines represent $N_{\text{ACK}} = 5$. The line style denotes different LBT strategies: solid lines represent $N_{\text{CSMA}} = 0$ (i.e., the ALOHA strategy [54]), dashed lines represent $N_{\text{CSMA}} = 1$, and dotted lines represents $N_{\text{CSMA}} = 5$. Note that when, due to the CSMA/CA protocol, a packet transmission cannot be performed, the transport layer considers the packet unacknowledged and, if there is an acknowledgment mechanism, further retries can be attempted.

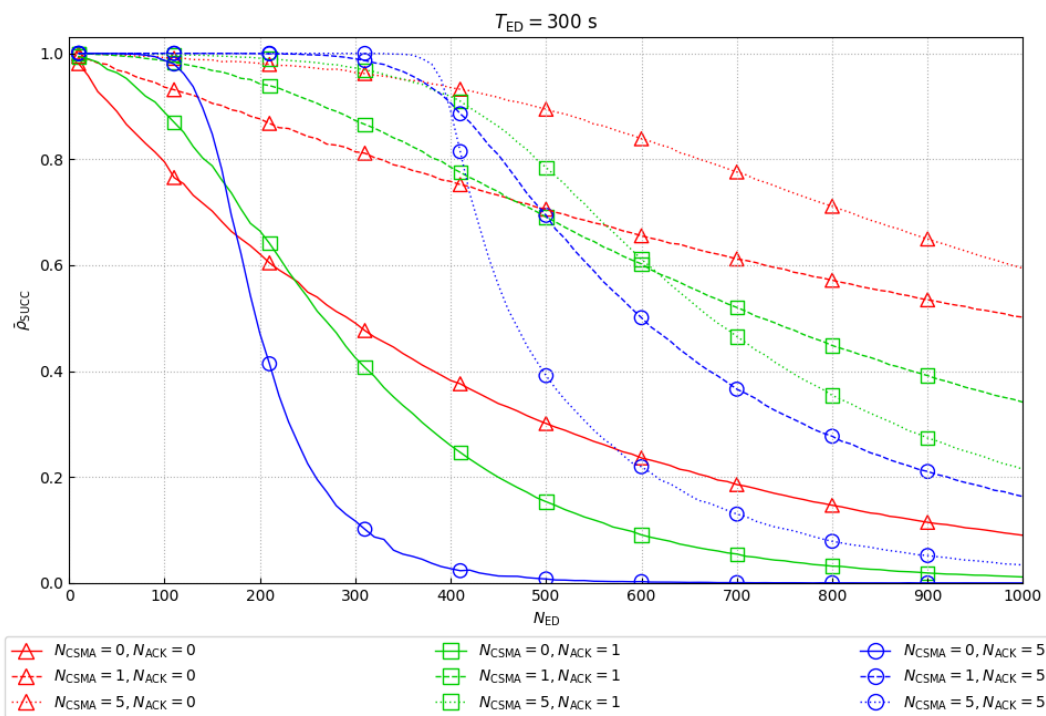


Figure 9. Average success rate $\bar{\rho}_{\text{SUCC}}$ with $T_{\text{ED}} = 300$ s.

We analyze the simulation results by first considering those shown in Figure 9. These results represent the worst case scenario, for two reasons: in first place, the small simulated area of size $W_A \times H_A = 200 \text{ m} \times 200 \text{ m}$ and the same modulation parameters used by every node ($F_S = 7$, $f = 868.0$ MHz and $B = 125$ kHz) constitute a single collision domain, where each node can receive the messages sent by every other node. In second place, each node generates a data packet every $T_{\text{ED}} = 300$ s, resulting in the highest simulated offered traffic.

The solid line with the triangle marker represents the simplest access technique, characterized by the absence of both LBT and acknowledgment mechanisms; in this case, the success rate deteriorates quickly as the number of EDs increases, considering that for $N_{\text{ED}} \approx 100$ the success rate $\bar{\rho}_{\text{SUCC}}$ falls below 80%.

By considering also the two other solid lines ($N_{\text{CSMA}} = 0$), the choice $N_{\text{ACK}} = 5$ (circle markers) provides the best performance until $N_{\text{ED}} \approx 150$ where, after that, $N_{\text{ACK}} = 1$ (square markers) becomes the best choice, replaced by $N_{\text{ACK}} = 0$ (triangle markers) after $N_{\text{ED}} \approx 230$.

This can be explained by an increased collision probability, when more traffic is present on the channel: the acknowledgment mechanism contributes to an avalanche effect that causes a quick decrease of $\bar{\rho}_{\text{SUCC}}$.

If we focus our attention to the case $N_{\text{ACK}} = 0$ (triangle markers), we can appreciate the improvement granted by the introduction of the CSMA/CA mechanism: it can be noticed that a success rate $\bar{\rho}_{\text{SUCC}} = 80\%$ is achieved, at least, up to $N_{\text{ED}} \approx 100$ with $N_{\text{CSMA}} = 0$, up to $N_{\text{ED}} \approx 330$ with $N_{\text{CSMA}} = 1$, and up to $N_{\text{ED}} \approx 680$ with $N_{\text{CSMA}} = 5$. Therefore, our LBT-based approach allows an increased number of EDs with respect to LoRaWAN. Alternatively, if we fix the number of EDs, our LBT-based approach achieves a larger success rate (and hence a reduced packet loss rate) with respect to LoRaWAN.

If we consider the cases $N_{\text{ACK}} = 1$ (square markers) or $N_{\text{ACK}} = 5$ (circle markers), it is interesting to note that at some point ($N_{\text{ED}} \approx 600$ for $N_{\text{ACK}} = 1$ and $N_{\text{ED}} \approx 400$ for $N_{\text{ACK}} = 5$) the performance with $N_{\text{CSMA}} = 1$ improves upon that with $N_{\text{CSMA}} = 5$.

Overall, the best performance is achieved with $N_{\text{CSMA}} = 5$ and $N_{\text{ACK}} = 5$ until $N_{\text{ED}} \approx 400$. After this point, $N_{\text{CSMA}} = 5$ and $N_{\text{ACK}} = 0$ give the best results.

Figures 10 and 11 present the results achieved with longer packet transmission periods of $T_{\text{ED}} = 600$ s and $T_{\text{ED}} = 900$ s, respectively. From these figures we may infer the same characteristics outlined in the previous paragraph for the case $T_{\text{ED}} = 300$ s. The main difference lies in the increase of the number of EDs where parameters configuration switch-over occurs.

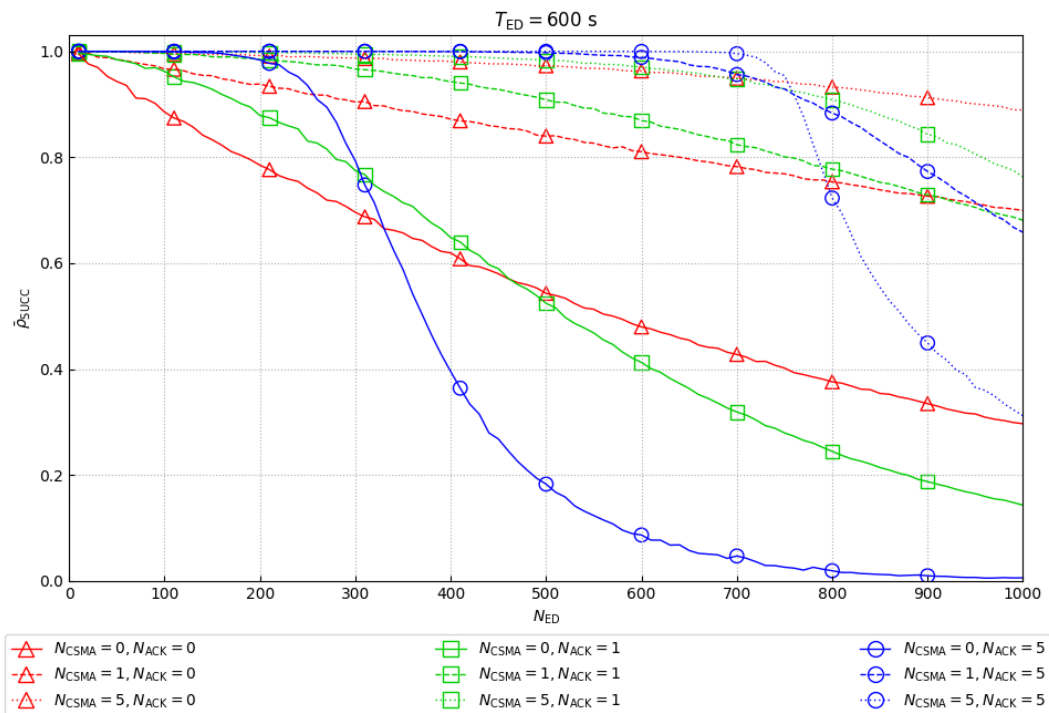


Figure 10. Average success rate $\bar{\rho}_{\text{SUCC}}$ with $T_{\text{ED}} = 600$ s.

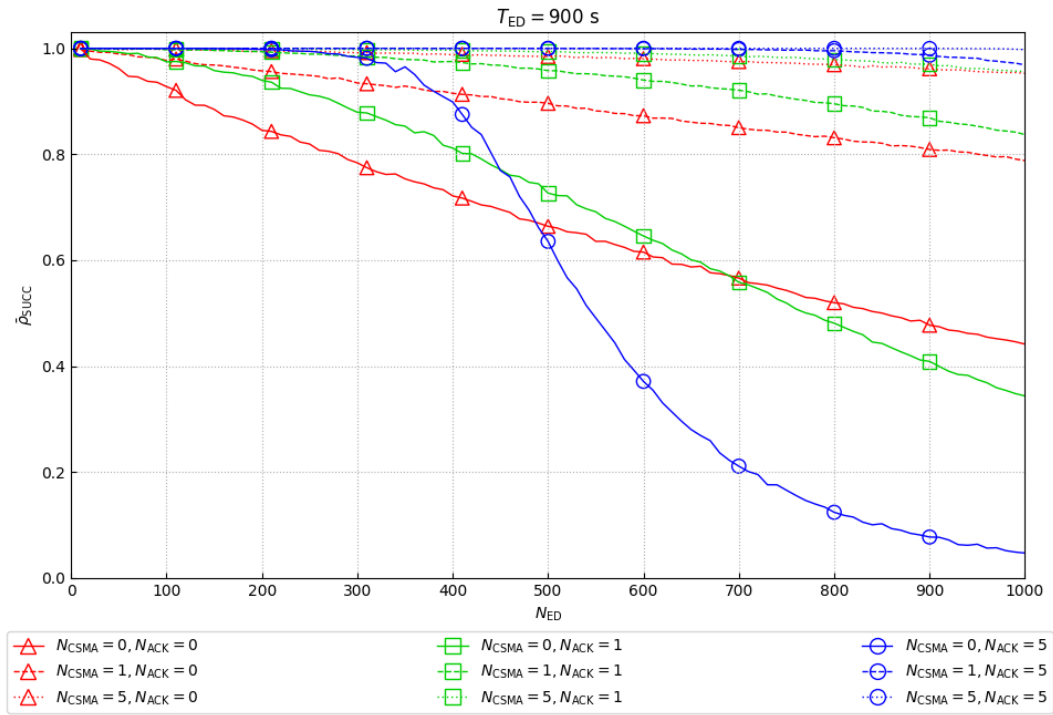


Figure 11. Average success rate $\bar{\rho}_{SUCC}$ with $T_{ED} = 900$ s.

5.2. Traffic Composition

In this section, we analyze the composition of the PHY and transport layer traffic, distinguishing among successful transmissions, packet retransmissions, and failed transmissions. It is interesting to investigate channel occupation and transmission overhead caused by the adoption of different techniques.

To this purpose, we define $R_{P,SUCC}$ as the average successful packet rate per hour and per node, as

$$R_{P,SUCC} = 3600 \frac{\sum_{i=0}^{N_{ED}-1} N_{SUCC,i}}{N_{ED} D_{SIM}}. \quad (7)$$

Similarly, we define $R_{P,FAIL}$ and $R_{P,RETR}$ as

$$R_{P,FAIL} = 3600 \frac{\sum_{i=0}^{N_{ED}-1} N_{FAIL,i}}{N_{ED} D_{SIM}}, \quad (8)$$

$$R_{P,RETR} = 3600 \frac{\sum_{i=0}^{N_{ED}-1} N_{RETR,i}}{N_{ED} D_{SIM}}. \quad (9)$$

Eventually, we also define $R_{P,TOT}$ as the sum of the three rates (7)–(9), as

$$R_{P,TOT} = R_{P,SUCC} + R_{P,FAIL} + R_{P,RETR}. \quad (10)$$

Figures 12–14 show the traffic composition for different transmission periods $T_{ED} = \{300, 600, 900\}$ s of all the simulated combinations. The parameters are represented on a stack chart, beginning with $R_{P,SUCC}$ at the bottom, followed by $R_{P,FAIL}$ in the middle, and terminated by $R_{P,RETR}$ on the top: it is important to note that the top value of the curve also represents $R_{P,TOT}$. With this type of graphical representation, it is possible to visualize the amount of each traffic component and its proportion with respect to the global traffic.

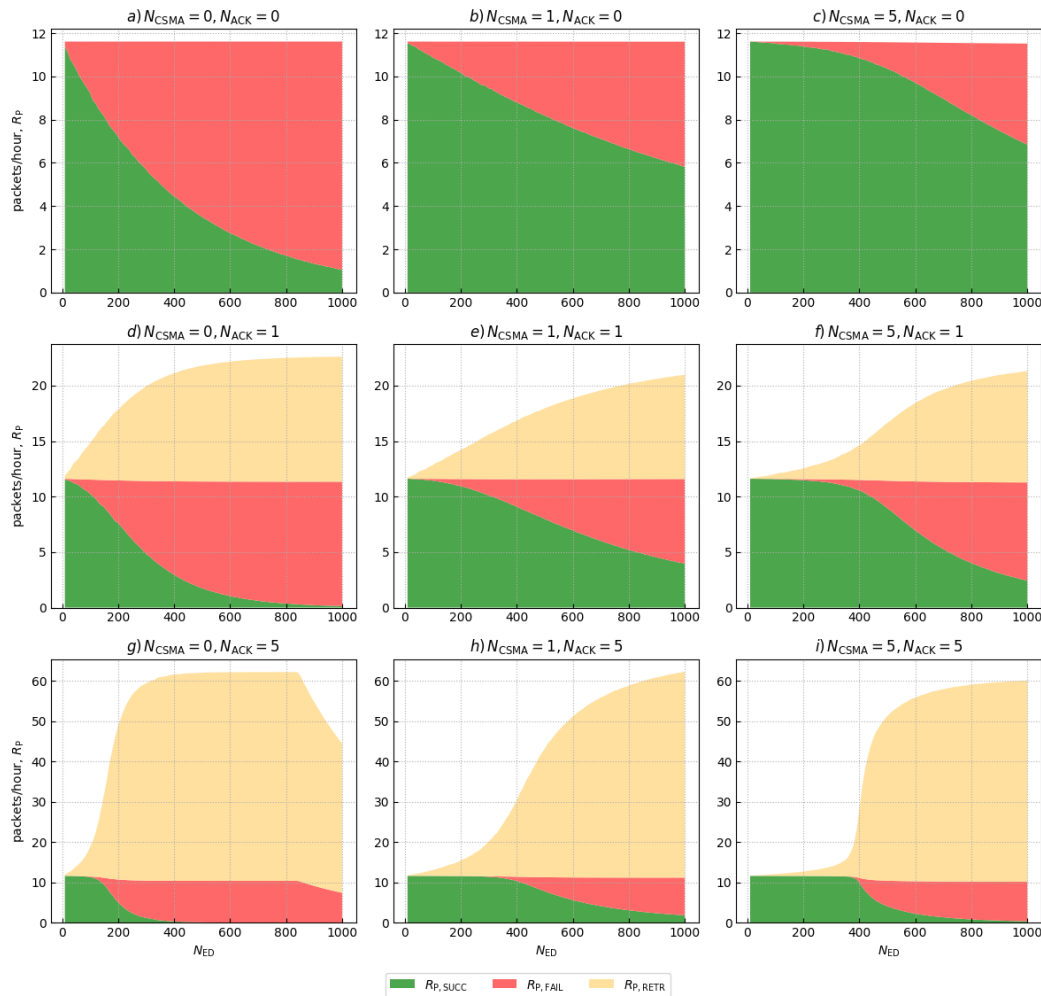


Figure 12. Traffic composition with a transmission period $T_{ED} = 300$ s.

We want to point out that the small number of packets sent each hour by every node (in the order of ten) in our application is motivated by low power consumption constraints. We analyze the simulation results by first considering those shown in Figure 12, since this represents the worst case scenario. Figure 12a presents the configuration without LBT and acknowledgment mechanisms: due to $N_{ACK} = 0$, there is no $R_{p,RETR}$ traffic component and $R_{p,SUCC}$ decreases quickly as N_{ED} grows. In Figure 12b,c we can appreciate the improvement granted by CSMA/CA, which lets $R_{p,SUCC}$ increase without affecting the overall traffic volume. Figure 12d,g report the impact generated by the acknowledgment mechanism, which provides an improvement at low N_{ED} despite an increase of traffic volume due to the retransmission events; this traffic increase, without CSMA/CA, creates an avalanche effect that causes a rapid drop of $R_{p,SUCC}$. This avalanche effect achieves its maximum in Figure 12g, where, after $N_{ED} \approx 850$, there is a traffic breakdown that indicates the saturation of the channel.

Considering that, based on $\bar{\rho}_{SUCC}$, the best performance is achieved with $N_{CSMA} = 5$ and $N_{ACK} = 5$ until $N_{ED} \approx 400$, and, after this point, $N_{CSMA} = 5$ and $N_{ACK} = 0$ give the best results, it is interesting to analyze the traffic composition in these scenarios. The first region is shown in Figure 12i: It can be seen that until $N_{ED} \approx 400$ the overhead is acceptable, considering the high $\bar{\rho}_{SUCC}$ granted. The second region is shown in Figure 12c, where the overhead for $N_{ED} > 400$ is absent, differently from Figure 12i, also guaranteeing a better $\bar{\rho}_{SUCC}$.

We can conclude that the best configuration ($N_{\text{CSMA}} = 5$, $N_{\text{ACK}} = 5$ until $N_{\text{ED}} \approx 400$, and $N_{\text{ACK}} = 0$ after that) resulting by the analysis of $\bar{\rho}_{\text{SUCC}}$ also presents a good channel occupation, avoiding an undue overhead.

Figures 13 and 14 present the results achieved with longer packet transmission periods of $T_{\text{ED}} = 600$ s and $T_{\text{ED}} = 900$ s, respectively. From these figures we may infer the same characteristics outlined in the previous paragraph for the case $T_{\text{ED}} = 300$ s. The main difference lies in the increase of the number of EDs where the described behaviors occur.

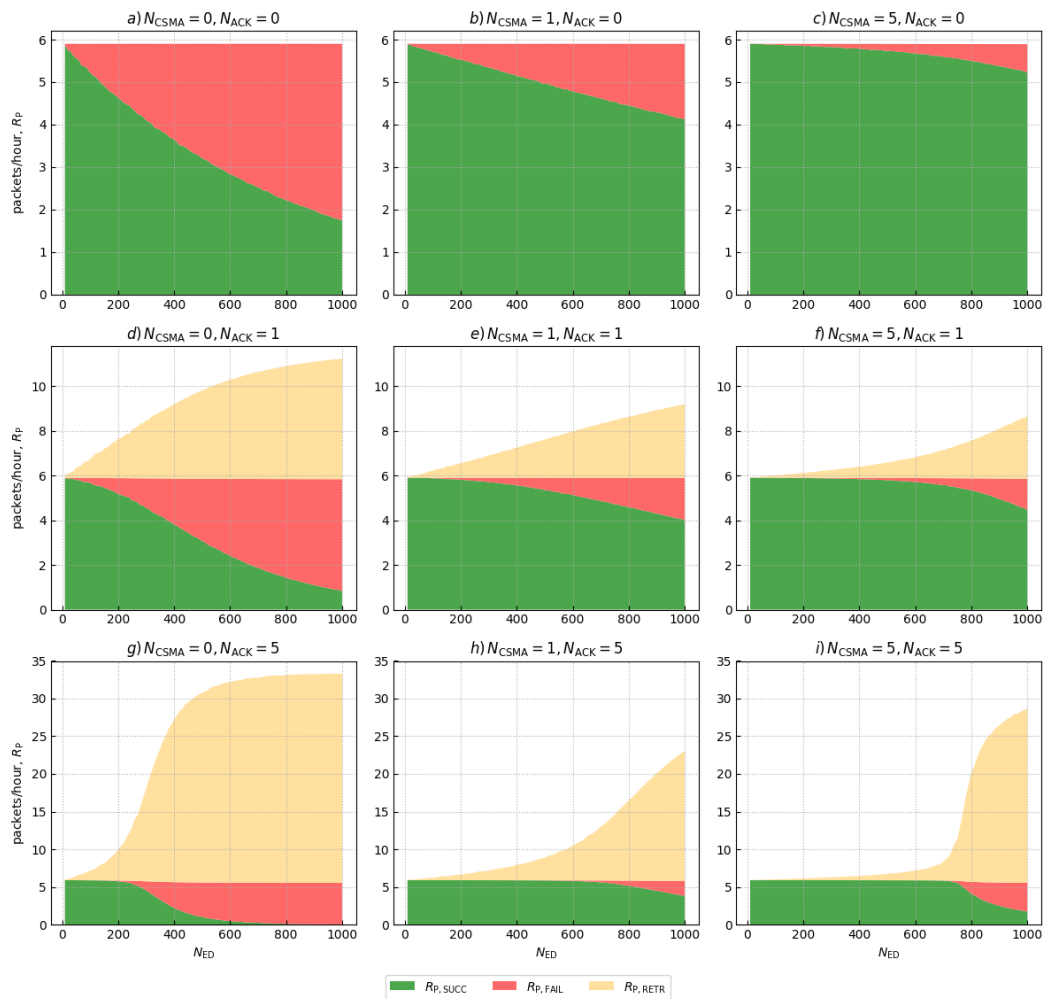


Figure 13. Traffic composition with a transmission period $T_{\text{ED}} = 600$ s.

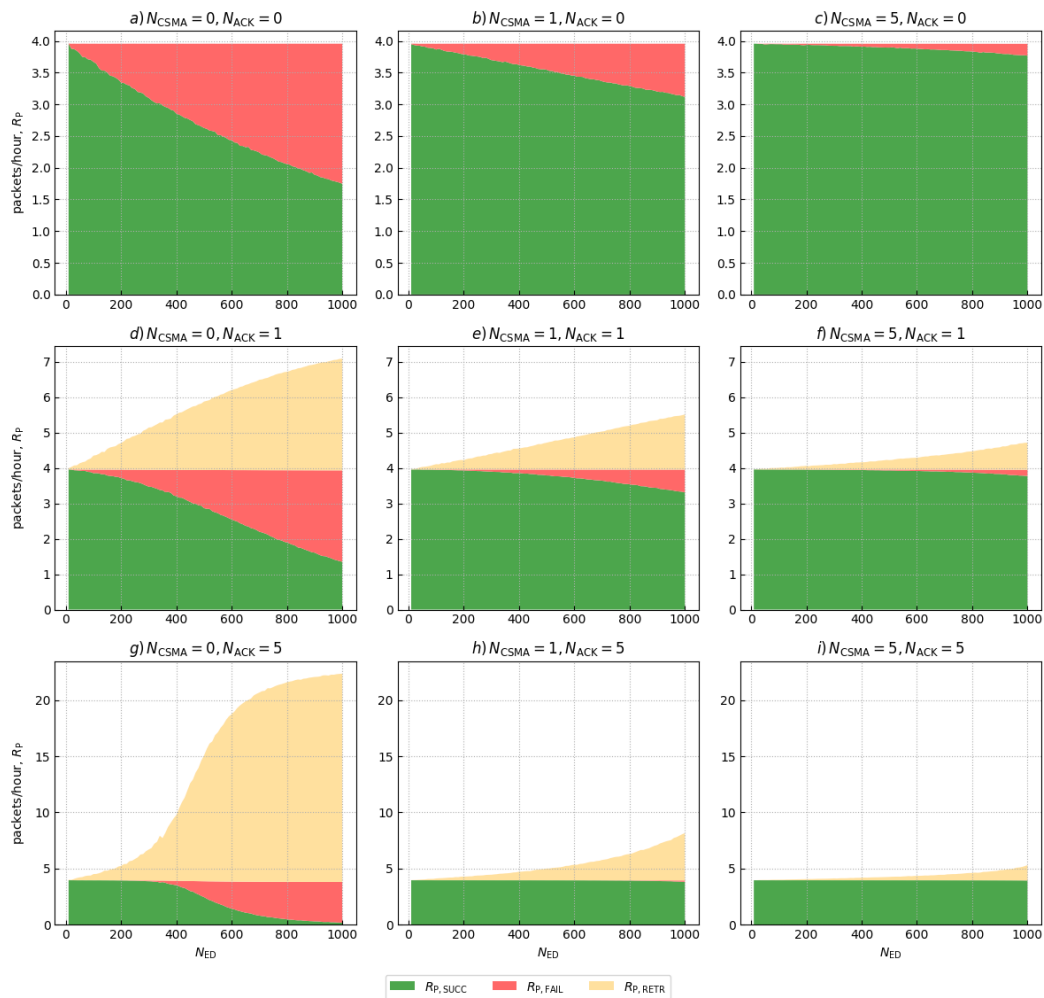


Figure 14. Traffic composition with a transmission period $T_{ED} = 900$ s.

6. Experimental Results

6.1. Range Performance

Experimental measurements were carried out to study the typical coverage area of the wireless transmission, with the parameters listed in Table 4.

Table 4. Range performance parameters.

| Parameter Name | Parameter Value |
|--------------------|-------------------|
| Bandwidth | $B = 125$ kHz |
| Frequency | $f = 863.5$ MHz |
| Spreading factor | $F_S = 12$ |
| Coding rate | $R_C = 4/5$ |
| Transmission power | $P_{TX} = 20$ dBm |
| GW antenna height | $h_{GW} = 5$ m |

Figure 15 displays the path of the coverage test; the start and finish point, where the gateway is located, is the engineering campus of University of Perugia, and it is marked by the blue pin. The path was traveled by foot, with the device sending its GNSS coordinates every 30 s to the LoRa GW.

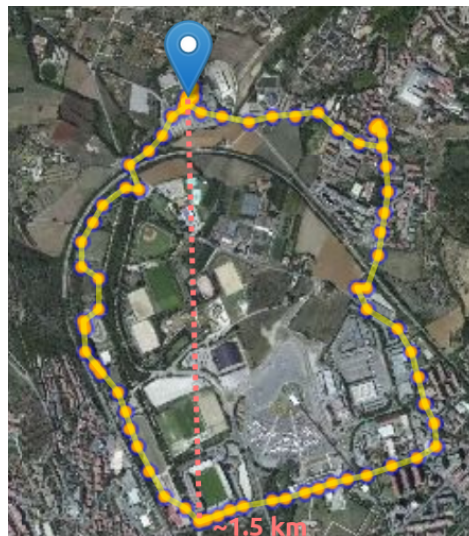


Figure 15. Coverage and transmission range test area.

The path unwinds in a suburban-hilly terrain, which is tougher than that of a typical grazing scenario and far larger than that of a shed. Anyway, we were able to achieve full link coverage in almost every point of the path, reaching more than 1.5 km of maximum coverage distance in non-line-of-sight (NLOS) conditions. If the same terrain conditions were assumed along all the directions, an approximate coverage area of 7 km² would be possible. For reference, in [2], a maximum coverage distance of 2.5 km is obtained in a dense urban terrain, but with an antenna height of 70 m, whereas in [25] the achieved maximum range is 2.03 km on a hilly environment.

To understand the differences between indoor and outdoor scenarios, we have measured the RSSI in two indoor environments (two cowsheds with different geometry, number and type of pillars, walls, ceilings, and metallic fences) and in one outdoor case, assuming a similar coverage radius of 50 m. We have found out that the standard deviation of the RSSI is 8.2 dB and 11.9 dB in the indoor environments, and 6.5 dB in the outdoor environment. It appears evident that the standard deviation of the RSSI is larger in indoor conditions, due to the presence of more obstacles than in the outdoor case. However, the variability of the RSSI also depends on the type of indoor environment. As a reference, we can compare these results with those presented in [26], where the indoor path loss standard deviation varies from 4.65 dB to 10.51 dB, also in this case depending on the type of indoor propagation environment.

6.1.1. Battery Lifespan of LoRa EDs

The battery-operated lifespan of an ED has been investigated by considering the dairy cattle shed environment, with the device characteristic parameters shown in Table 5.

Table 5. Configuration of the LoRa ED used for battery duration test.

| Parameter Name | Parameter Value |
|----------------------------------|----------------------------|
| LiPo battery voltage | 3.7 V |
| LiPo battery capacity | 2800 mAh |
| Transmission power | $P_{TX} = 20$ dBm |
| Transmission period | $T_{ED} = 4200$ s (70 min) |
| Payload size | $N_{PL} = 256$ bytes |
| Board status sampling period | 3600 s |
| Body temperature sampling period | 900 s |
| Accelerometer sampling period | 5 s |

The ED checks the body temperature sensor and the accelerometer with different sampling periods, as well as the electrical parameters of the board (such as battery voltage and microcontroller temperature). The acceleration data is represented by a binary array where ‘1’ means that the acceleration magnitude exceeded a given threshold during the accelerometer sampling period, ‘0’ otherwise. These data are stored in the ED and, approximately every 70 min (4200 s), they are sent with a LoRa data packet of size $N_{PL} = 256$ bytes, together with the body temperature and board data. As summarized in Table 6, experimental measurements show that there are three different current intensity levels, which repeat cyclically depending on the operating mode of the ED. During LoRa packet transmission mode, a current of nearly 23 mA is drawn from the battery, whereas only 15 mA are needed during the sampling mode (to sample the sensors and the microcontroller parameters). At all other times, the device rests idle in sleep mode, consuming only 0.3 mA. By considering the estimated relative duration of each mode, on the average a current intensity of 2.19 mA is required. Under the simplified hypothesis of linear discharge of the battery, a rough estimate of the expected battery lifespan is $2800/2.19 \approx 1280$ h (approximately 53 days).

Table 6. Estimated current intensity levels of the ED configured for battery tests, during the different operating modes.

| Description | Current Level | Relative Duration |
|----------------------------|---------------|-------------------|
| LoRa transmission | 23 mA | 0.01 % |
| Board and sensors sampling | 15 mA | 12.85 % |
| Sleep | 0.3 mA | 87.14 % |

Figure 16 shows the battery discharge curve, in V, measured on the ED over more than one month: The battery starts from a fully charged status, with a voltage of 4.26 V, and slowly descends below the nominal battery voltage of 3.7 V. This shows that the battery correctly works above the nominal value for more than one month.

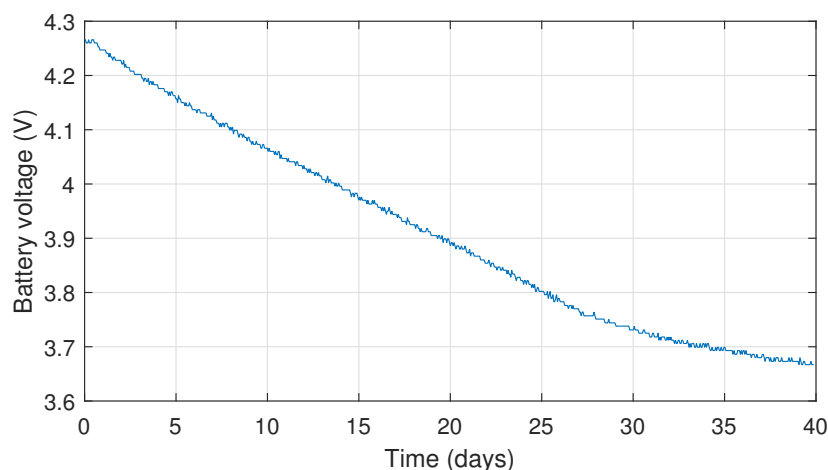


Figure 16. Battery discharge curve, represented by the measured voltage.

6.2. Exploratory Data Analysis of Sensor Measurements

In this section the preliminary data collected by our platform are presented and analyzed with an exploratory data analysis (EDA) approach [55].

As a first example, the data plotted in Figure 17 are the values of temperature, relative humidity, illuminance, and NH_3 concentration captured by an indoor GW, installed in a dairy cattle shed, during the day of 6 September 2019. This kind of result visualization is typically experimented by the users of the web-based GUI, who want to check the parameters variation during the previous hours.

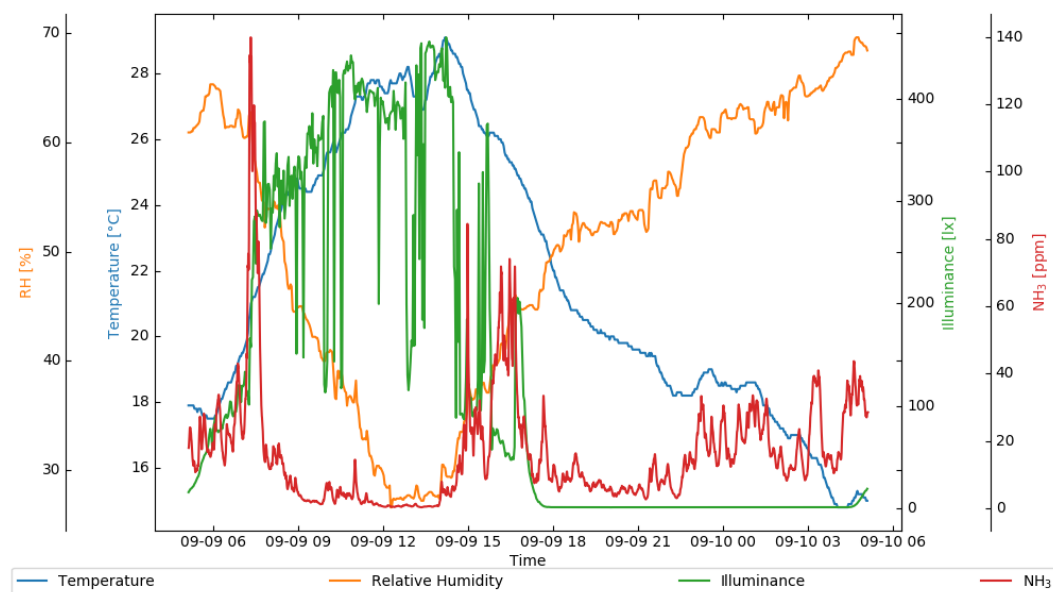


Figure 17. Temperature, relative humidity, illuminance, and NH_3 concentration data collected on 6 September 2019.

As another example, we consider all the measurements collected from 9 July 2019 to 10 September 2019 and we make a preliminary analysis about seasonality and trend using a seasonal decomposition with the moving average method [56].

Figure 18 shows the shed temperature variation and the extrapolated trend using seasonal decomposition. This type of data analysis is performed to infer a trend from time series, to remove fast variations and other disturbances from the observed data.

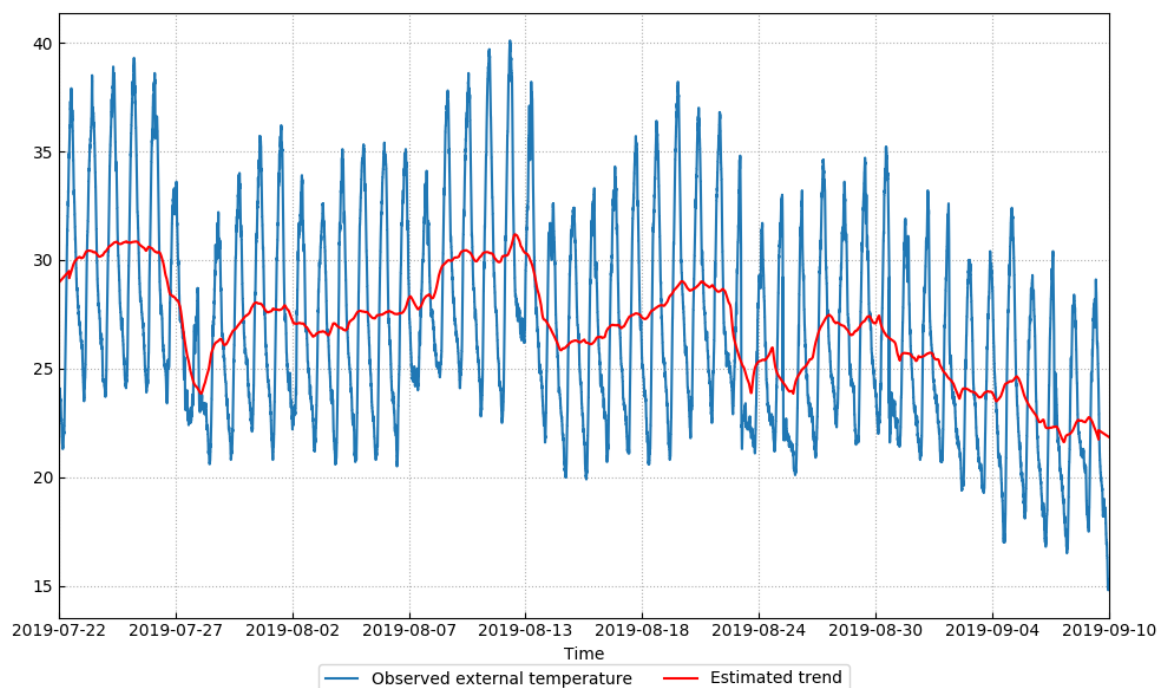


Figure 18. Temperature seasonal decomposition.

Figure 19, instead, shows all the seasonal trends: in this way, it is possible to visualize correlations among the parameters. For instance, during the day of 29 July 2019, a decrease of temperature and illuminance, as well as an increase of relative humidity, are observed. At the same time, the CO_2 and NH_3 concentrations increase. This correlation should suggest the farmer to investigate further on the event.

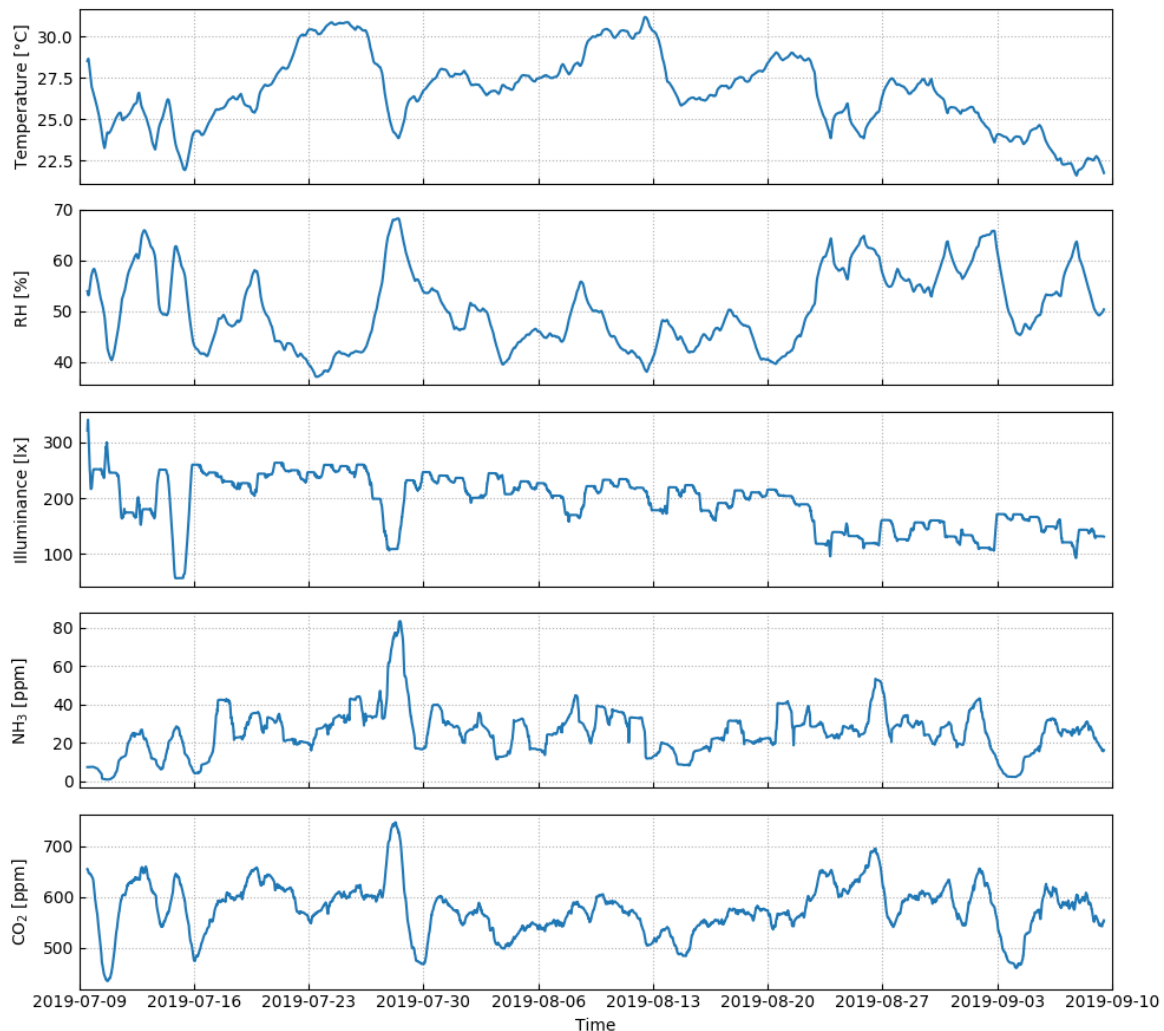


Figure 19. Seasonal data trends of the collected parameters.

A further analysis on the parameters correlation is necessary but, since the parameters have different sampling periods, we compare the hourly average.

Figure 20 shows the resulting data visualization matrix after analyzing the data with the Seaborn Python3 tool [57]. Different statistical analyses have been employed: the histograms and the probability density function (pdf) estimated with kernel density estimation (KDE) method [58] are represented along the main diagonal. On the upper triangular portion of the matrix, the correlation coefficients between the various couples of parameters are reported; on the lower triangular portion, scatter diagrams represent one parameter plotted against another, and the black line denotes the locally weighted linear regression model [59].

The well-known relation between temperature and relative humidity emerges from the strong correlation value (-0.85) and from the scatter plot, where the inverse proportionality is evident. The other trivial relation between temperature and illuminance is confirmed by the strong correlation value (0.76) and clear scatter plot, where the proportionality is shown.

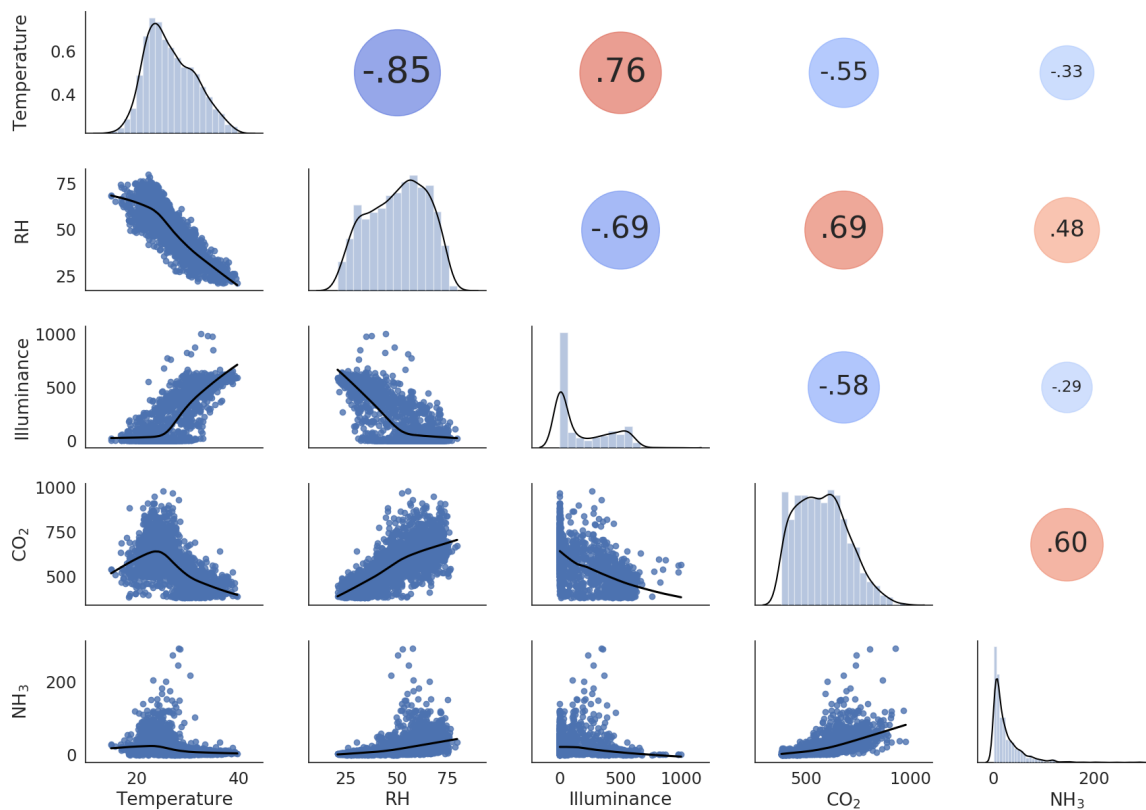


Figure 20. Statistical analysis with scattering plots, cross-correlation values, and kernel density estimation (KDE). The five plots in the main diagonal are the calculated histograms of the corresponding five quantities, together with the estimated Gaussian mixture pdf. The ten plots in the lower triangular part represent the scattering plots of each couple of quantities, together with the locally weighted regression curve. The ten values in the upper triangular part, instead, represent the correlation coefficients between each couple of quantities, where reddish (blueish) colors represent a positive (negative) correlation, and the size of the circle is proportional to the correlation magnitude.

CO₂ and NH₃ present a moderate correlation (0.60) and an almost direct proportionality; this is explainable by the gaseous nature of the parameters that may be affected by the same external condition, like strong wind or cattle distribution inside the shed.

Other parameters present a quite strong correlation and evident proportionality (i.e., CO₂ vs. RH, illuminance vs. RH, CO₂ vs. illuminance); however, it is not easy to infer the direct cause of those correlations.

As another example of collected data, we consider the accelerations captured by the collar worn by cattle. Figure 21 shows the cluster plot of the x , y , and z average acceleration components captured with a sampling period of 30 s. Each measured sample results from the averaging of the instantaneous acceleration over 5 s, for an overall capture interval of 3 days. Samples are approximately distributed over the surface of a spherical dome with radius of about 9.8 m/s^2 . A simple cluster analysis obtained with the k -means algorithm has been carried out to classify the acceleration values into 4 classes, shown with different colors in the figure. By using this technique, together with other parameters obtained from the acceleration and gyroscopic measurements, it would be in principle possible to classify the activity of each monitored head.

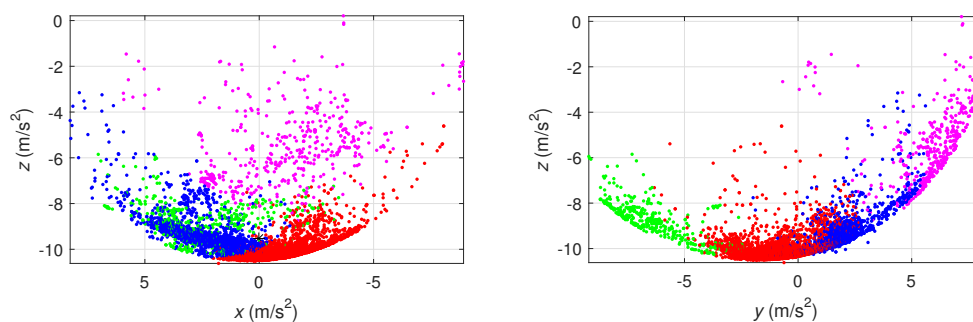


Figure 21. Cluster analysis of the average acceleration data with k -means, using 4 classes over a 3-day period. Different colors represent different classes.

7. Conclusions

In this paper we have presented the design and the implementation of a small-scale sensor network devoted to the monitoring of cattle vital parameters and cowsheds environmental parameters using the LoRa LPWAN technology. The choice of the LoRa network technology has proven advantageous, due to the resilient transmission, the wide coverage, and the long-lasting battery lifespan, which meet the needs of rural area IoT applications. As discussed in Section 6.1, it is possible to cover an area of about 7 km², in a sub-urban/hilly propagation environment, which appears to be harsher than that of a typical rural area, greater than that of a single cowshed, and more associated to that of grazing. As shown in Section 5 by means of MAC-level simulations, it is possible to manage a large number of EDs with a single gateway, thanks to the proposed architecture and to a careful implementation of a custom MAC layer, which has been tailored to the transmission intervals typically adopted in this kind of applications. Specifically, with respect to LoRaWAN, our LBT-based approach allows an increased number of EDs or a larger success rate. For scalability, it is possible to add more gateways and to assign different frequencies and spreading factors to the EDs. It is also worth noting that the use of the custom MAC with LBT capabilities allowed to decrease the number of required data retransmissions, as well as to reduce the power consumption of the EDs. Finally, in Section 6.2 we have also shown a preliminary data analysis concerning the deployed prototypes, four gateways and four EDs, which are currently in operation and collecting data, and we briefly discussed the correlations among the parameters.

Author Contributions: Conceptualization, L.G., V.M., G.B., L.R. and F.F.; methodology, L.G., V.M., G.B. and L.R.; software, L.G., V.M. and G.B.; validation, L.G. and V.M.; investigation, L.G., V.M., G.B., L.R. and F.F.; writing—original draft preparation, L.G., V.M., G.B. and L.R.; writing—review and editing, L.G., V.M., G.B., L.R. and F.F.; supervision, G.B. and F.F.; project administration, F.F.; funding acquisition, F.F.

Funding: This research was funded by the Project “Livestock Smart Farming” (PSR 2014-2020 16.1, Regione Umbria) and by the project “Signal processing and machine learning” (Fondo di Ricerca di Base 2017, University of Perugia).

Acknowledgments: The authors thank L. Ipericon and A. and C. Taglioni for the logistic support. The authors also thank M. Malfagia and S. Pignani for providing some of the hardware used in the simulation and testing.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Choi, Y.J.; Kang, H.J.; Lee, I.G. Scalable and secure Internet of Things connectivity. *Electronics* **2019**, *8*, 752. doi:10.3390/electronics8070752. [CrossRef]
2. Pasolini, G.; Buratti, C.; Feltrin, L.; Zabini, F.; De Castro, C.; Verdone, R.; Andrisano, O. Smart City pilot projects using LoRa and IEEE802.15.4 technologies. *Sensors* **2018**, *18*, 1118. doi:10.3390/s18041118. [CrossRef] [PubMed]
3. Kabalci, Y.; Kabalci, E.; Padmanaban, S.; Holm-Nielsen, J.B.; Blaabjerg, F. Internet of Things applications as energy internet in Smart Grids and Smart Environments. *Electronics* **2019**, *8*, 972. doi:10.3390/electronics8090972. [CrossRef]

4. Sisinni, E.; Carvalho, D.F.; Ferrari, P.; Flammini, A.; Silva, D.R.C.; Da Silva, I.M.D. Enhanced flexible LoRaWAN node for industrial IoT. In Proceedings of the 2018 14th IEEE International Workshop on Factory Communication Systems (WFCS), Imperia, Italy, 13–15 June 2018; pp. 1–4. doi:10.1109/WFCS.2018.8402367. [\[CrossRef\]](#)
5. Haxhibeqiri, J.; Karaagac, A.; Van den Abeele, F.; Joseph, W.; Moerman, I.; Hoebeke, J. LoRa indoor coverage and performance in an industrial environment: Case study. In Proceedings of the 2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Limassol, Cyprus, 12–15 September 2017; pp. 1–8. doi:10.1109/ETFA.2017.8247601. [\[CrossRef\]](#)
6. Islam, S.M.R.; Kwak, D.; Kabir, M.H.; Hossain, M.; Kwak, K. The Internet of Things for health care: A comprehensive survey. *IEEE Access* **2015**, *3*, 678–708. doi:10.1109/ACCESS.2015.2437951. [\[CrossRef\]](#)
7. Pereira, C.; Mesquita, J.; Guimarães, D.; Santos, F.; Almeida, L.; Aguiar, A. Open IoT architecture for continuous patient monitoring in emergency wards. *Electronics* **2019**, *8*, 1074. doi:10.3390/electronics8101074. [\[CrossRef\]](#)
8. Kumar, N.M.; Dash, A. The Internet of Things: An opportunity for transportation and logistics. In Proceedings of the International Conference on Inventive Computing and Informatics (ICICI 2017), Coimbatore, India, 23–24 November 2017; pp. 194–197.
9. Sanchez-Iborra, R.; Liaño, I.G.; Simoes, C.; Couñago, E.; Skarmeta, A.F. Tracking and monitoring system based on LoRa technology for lightweight boats. *Electronics* **2018**, *8*, 15. doi:10.3390/electronics8010015. [\[CrossRef\]](#)
10. Ojha, T.; Misra, S.; Raghuwanshi, N.S. Wireless sensor networks for agriculture: The state-of-the-art in practice and future challenges. *Comput. Electron. Agric.* **2015**, *118*, 66–84. doi:10.1016/j.compag.2015.08.011. [\[CrossRef\]](#)
11. Jawad, H.M.; Nordin, R.; Gharghan, S.K.; Jawad, A.M.; Ismail, M. Energy-Efficient wireless sensor networks for precision agriculture: A review. *Sensors* **2017**, *17*, 1781. doi:10.3390/s17081781. [\[CrossRef\]](#) [\[PubMed\]](#)
12. Yim, D.; Chung, J.; Cho, Y.; Song, H.; Jin, D.; Kim, S.; Ko, S.; Smith, A.; Riegsecker, A. An experimental LoRa performance evaluation in tree farm. In Proceedings of the 2018 IEEE Sensors Applications Symposium (SAS), Seoul, Korea, 12–14 March 2018; pp. 1–6. doi:10.1109/SAS.2018.8336764. [\[CrossRef\]](#)
13. Raza, U.; Kulkarni, P.; Sooriyabandara, M. Low power wide area networks: An overview. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 855–873. doi:10.1109/COMST.2017.2652320. [\[CrossRef\]](#)
14. Barker, P.; Hammoudeh, M. A survey on low power network protocols for the Internet of Things and wireless sensor networks. In Proceedings of the International Conference on Future Networks and Distributed Systems, Cambridge, UK, 19–20 July 2017; p. 44. doi:10.1145/3102304.3102348. [\[CrossRef\]](#)
15. Augustin, A.; Yi, J.; Clausen, T.; Townsley, W.M. A study of LoRa: Long range & low power networks for the Internet of Things. *Sensors* **2016**, *16*, 1466. doi:10.3390/s16091466. [\[CrossRef\]](#)
16. Semtech. *SX1272/3/6/7/8: LoRa Modem Designer's Guide*; Semtech Corporation: Camarillo, CA, USA, 2013.
17. Baruffa, G.; Rugini, L.; Mecarelli, V.; Germani, L.; Frescura, F. Coded LoRa performance in wireless channels. In Proceedings of the 2019 IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (IEEE PIMRC 2019), Istanbul, Turkey, 8–11 September 2019; pp. 1–6.
18. Sondrol, T.; Jalaian, B.; Suri, N. Investigating LoRa for the Internet of Battlefield Things: A cyber perspective. In Proceedings of the MILCOM 2018—2018 IEEE Military Communications Conference (MILCOM), Los Angeles, CA, USA, 29–31 October 2018; pp. 749–756. doi:10.1109/MILCOM.2018.8599805. [\[CrossRef\]](#)
19. de Carvalho Silva, J.; Rodrigues, J.J.P.C.; Alberti, A.M.; Solic, P.; Aquino, A.L.L. LoRaWAN—A low power WAN protocol for Internet of Things: A review and opportunities. In Proceedings of the 2017 2nd International Multidisciplinary Conference on Computer and Energy Science (SpliTech), Split, Croatia, 12–14 July 2017; pp. 1–6.
20. Adelantado, F.; Vilajosana, X.; Tuset-Peiro, P.; Martinez, B.; Melia-Segui, J.; Watteyne, T. Understanding the limits of LoRaWAN. *IEEE Commun. Mag.* **2017**, *55*, 34–40. doi:10.1109/MCOM.2017.1600613. [\[CrossRef\]](#)
21. Liu, X.; Huo, C. Research on remote measurement and control system of piggery environment based on LoRa. In Proceedings of the 2017 Chinese Automation Congress (CAC), Jinan, China, 20–22 October 2017; pp. 7016–7019. doi:10.1109/CAC.2017.8244042. [\[CrossRef\]](#)
22. Sharma, B.; Koundal, D. Cattle health monitoring system using wireless sensor network: A survey from innovation perspective. *IET Wirel. Sens. Syst.* **2018**, *8*, 143–151. doi:10.1049/iet-wss.2017.0060. [\[CrossRef\]](#)

23. Cousin, P.; Dupont, C.; Fatnassi, S.; Pham, C.; Thiare, O.; Wussah, A.; Koffi, S. IoT, an affordable technology to empower Africans addressing needs in Africa. In Proceedings of the 2017 IST-Africa Week Conference (IST-Africa), Windhoek, Namibia, 30 May–2 June 2017; pp. 1–8.
24. Davcev, D.; Mitreski, K.; Trajkovic, S.; Nikolovski, V.; Koteli, N. IoT agriculture system based on LoRaWAN. In Proceedings of the 2018 14th IEEE International Workshop on Factory Communication Systems (WFCS), Imperia, Italy, 13–15 June 2018; pp. 1–4. doi:10.1109/WFCS.2018.8402368. [CrossRef]
25. Li, Q.; Liu, Z.; Xiao, J. A Data collection collar for vital signs of cows on the grassland based on LoRa. In Proceedings of the 2018 IEEE 15th International Conference on e-Business Engineering (ICEBE), Xi'an, China, 12–14 October 2018; pp. 213–217. doi:10.1109/ICEBE.2018.00041. [CrossRef]
26. Petäjälä, J.; Mikhaylov, K.; Hämmäläinen, M.; Linatti, J. Evaluation of LoRa LPWAN technology for remote health and wellbeing monitoring. In Proceedings of the 2016 10th International Symposium on Medical Information and Communication Technology (ISMICT), Worcester, MA, USA, 20–23 March 2016; pp. 1–5. doi:10.1109/ISMICT.2016.7498898. [CrossRef]
27. Sardar, M.S.; Yi, Y.; Xue-fen, W.; Huang, J.; Zhang, J.; Qin, X.; Zhao, Q.; Wang, Y.; Iqbal, M.A. Experimental analysis of LoRa CSS wireless transmission characteristics for forestry monitoring and sensing. In Proceedings of the 2018 International Symposium in Sensing and Instrumentation in IoT Era (ISSI), Shanghai, China, 6–7 September 2018; pp. 01249–01254. doi:10.1109/ISSI.2018.8538171. [CrossRef]
28. Magrin, D.; Centenaro, M.; Vangelista, L. Performance evaluation of LoRa networks in a smart city scenario. In Proceedings of the 2017 IEEE International Conference on Communications (ICC), Paris, France, 21–25 May 2017; pp. 1–7. doi:10.1109/ICC.2017.7996384. [CrossRef]
29. Rizzi, M.; Ferrari, P.; Sisinni, E.; Gidlund, M. Using LoRa for industrial wireless networks. In Proceedings of the 2017 IEEE 13th International Workshop on Factory Communication Systems (WFCS), Trondheim, Norway, 31 May–2 June 2017; pp. 1–4. doi:10.1109/WFCS.2017.7991972. [CrossRef]
30. Cattle Watch. Available online: <http://www.cattle-watch.com/> (accessed on 18 November 2019).
31. Sodaq. Available online: <https://www.sodaq.com/> (accessed on 18 November 2019).
32. SenseHub Beef. Available online: <http://www.scr-dairy.com/cow-intelligence/sensehub-beef.html> (accessed on 18 November 2019).
33. Moocall. Available online: <https://www.moocall.com> (accessed on 18 November 2019).
34. Van den Abeele, F.; Haxhibeqiri, J.; Moerman, I.; Hoebeke, J. Scalability analysis of large-scale LoRaWAN networks in ns-3. *IEEE Internet Things J.* **2017**, *4*, 2186–2198. [CrossRef]
35. Bor, M.C.; Roedig, U.; Voigt, T.; Alonso, J.M. Do LoRa low-power wide-area networks scale? In Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM '16), Floriana, Malta, 13–17 November 2016; pp. 59–67. doi:10.1145/2988287.2989163. [CrossRef]
36. Dell. DELL PowerEdge R440 Round Rock, Texas, United States. Available online: <https://www.dell.com/en-us/work/shop/povw/poweredge-r440> (accessed on 24 November 2019).
37. Richardson, L.; Ruby, S. *RESTful web services*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2008.
38. Xu, B.; Xu, L.D.; Cai, H.; Xie, C.; Hu, J.; Bu, F. Ubiquitous data accessing method in IoT-based information system for emergency medical services. *IEEE Trans. Ind. Inform.* **2014**, *10*, 1578–1586. doi:10.1109/TII.2014.2306382. [CrossRef]
39. Baruffa, G.; Femminella, M.; Pergolesi, M.; Reali, G. A Big Data architecture for spectrum monitoring in cognitive radio applications. *Ann. Telecommun.* **2018**, *73*, 451–461. doi:10.1007/s12243-018-0642-7. [CrossRef]
40. Adafruit Learning System. *Adafruit Feather M0 radio with LoRa radio module*; 2019. Available online: <https://learn.adafruit.com/adafruit-feather-m0-radio-with-lora-radio-module> (accessed on 25 November 2019).
41. Fiedler, A.; Müller, H.J. Emissions of ammonia and methane from a livestock building natural cross ventilation. *Meteorol. Z.* **2011**, *20*, 59–65. doi:10.1127/0941-2948/2011/0490. [CrossRef]
42. Hristov, A.N.; Hanigan, M.; Cole, A.; Todd, R.; McAllister, T.A.; Ndegwa, P.M.; Rotz, A. Review: Ammonia emissions from dairy farms and beef feedlots. *Can. J. Anim. Sci.* **2011**, *91*, 1–35. doi:10.4141/CJAS10034. [CrossRef]
43. Mattachini, G.; Riva, E.; Provolo, G. The lying and standing activity indices of dairy cows in free-stall housing. *Appl. Anim. Behav. Sci.* **2011**, *129*, 18–27. doi:10.1016/j.applanim.2010.10.003. [CrossRef]

44. Aosong. *Digital Temperature and Humidity Sensor AM2321 Product Manual*; Aosong Guangzhou Electronics Co., Ltd.: Guangzhou, China, 2014.
45. Adafruit Learning System. *TSL2561 Luminosity Sensor*; 2018. Available online: <https://learn.adafruit.com/tsl2561> (accessed on 25 November 2019).
46. DFRobot. *Gravity: Analog infrared CO2 sensor for Arduino SKU: SEN0219*; 2017. Available online: https://wiki.dfrobot.com/Gravity__Analog_Infrared_CO2_Sensor_For_Arduino_SKU_SEN0219 (accessed on 25 November 2019).
47. Winsen. *Ammonia Gas Sensor Model: MQ137*; Zhengzhou Winsen Electronics Technology Co., Ltd.: Zhengzhou, China, 2015.
48. Adafruit Learning System. *Adafruit INA219 current sensor breakout*; 2019. Available online: <https://learn.adafruit.com/adafruit-ina219-current-sensor-breakout/> (accessed on 25 November 2019).
49. u-blox. *NEO-6 u-Blox 6 GPS Modules*; u-blox AG: Thalwil, Switzerland, 2011.
50. InvenSense Inc. *MPU-6000 and MPU-6050 Product Specification Revision 3.4*; InvenSense Inc.: Sunnyvale, CA, USA, 2013.
51. Maxim. *DS18B20-PAR 1-Wire Parasite-Power Digital Thermometer*. Available online: <https://www.maximintegrated.com/en/products/sensors/DS18B20-PAR.html> (accessed on 25 November 2019).
52. Sklar, B. Rayleigh fading channels in mobile digital communication systems. I. Characterization. *IEEE Commun. Mag.* **1997**, *35*, 90–100. doi:10.1109/35.601747. [CrossRef]
53. Ortín, J.; Cesana, M.; Redondi, A. How do ALOHA and Listen Before Talk coexist in LoRaWAN? In Proceedings of the 2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Bologna, Italy, 9–12 September 2018; pp. 1–7. doi:10.1109/PIMRC.2018.8580906. [CrossRef]
54. Iannello, F.; Simeone, O.; Spagnolini, U. Medium access control protocols for wireless sensor networks with energy harvesting. *IEEE Trans. Commun.* **2012**, *60*, 1381–1389. doi:10.1109/TCOMM.2012.030712.110089. [CrossRef]
55. Tukey, J.W. *Exploratory data analysis*; Addison-Wesley Series in Behavioral Science; Addison-Wesley: Reading, MA, USA, 1977.
56. Seabold, S.; Perktold, J. statsmodels: Econometric and statistical modeling with python. In Proceedings of the 9th Python in Science Conference, Austin, TX, USA, 28 June–3 July 2010.
57. Waskom, M.; Botvinnik, O.; O’Kane, D.; Hobson, P.; Ostblom, J.; Lukauskas, S.; Gemperline, D.C.; Augspurger, T.; Halchenko, Y.; Cole, J.B.; et al. *mwaskom/seaborn: v0.9.0 (July 2018)*; 2018. doi:10.5281/zenodo.1313201. Available online: <http://seaborn.pydata.org/> (accessed on 25 November 2019). [CrossRef]
58. Silverman, B.W. *Density Estimation for Statistics and Data Analysis*; Routledge: New York, NY, USA, 1998.
59. Fan, J. *Local Polynomial Modelling and Its Applications: Monographs on Statistics and Applied Probability 66*; Routledge: New York, NY, USA, 1996.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).