QuakeSense, a LoRa-compliant Earthquake Monitoring Open System

Pietro Boccadoro*,§, Biagio Montaruli†, Luigi Alfredo Grieco‡,§,
Department of Electrical and Information Engineering (DEI), Politecnico di Bari, Bari, Italy
§CNIT, Consorzio Nazionale Interunivesitario per le Telecomunicazioni, Bari
Email: *pietro.boccadoro@poliba.it, †b.montaruli@studenti.poliba.it, ‡a.grieco@poliba.it,

***Abstract*—Detecting disruptive events, such as earthquakes, using environmental monitoring systems is a particularly promising, but rather challenging, opportunity. The Internet of Things (IoT) can play a significant role in characterizing and predicting seismic events. The present contribution introduces QuakeSense, an open-source earthquake and weather monitoring system. The implemented IoT system is configured as a Long Range (LoRa)-based star topology with a fully energy-autonomous sensor node. The system leverages some of the most useful features of two emerging IoT technologies, e.g., LoRa and Message Queue Telemetry Transport (MQTT), and enables the near real-time monitoring of seismic events through a web-based interface. An experimental campaign has been carried out to verify the current consumption and, therefore, the battery lifetime of the sensor node. Moreover, LoRa parameters have been extensively tested as to evaluate performances in several configurations. The obtained results in terms of latency and Packet Delivery Ratio (PDR) demonstrated the reliability of the proposal.**

***Index Terms*—Internet of Things, Long Range, earthquake, safety.**

I. Introduction

Seismic events can create monstrous economic damage to buildings and people, which turn earthquakes to be among the natural greatest hazards [1]–[3]. An earthquake is a complex phenomenon that consists of rapid vibrations of the ground caused by a sudden release of a significant amount of elastic energy from a subsoil zone defined as hypo-center [4]. Monitoring activities are useful as a predictive form for emergency communications and on-site early warnings. In earthquake engineering, several parameters can be used to evaluate the effects of strong-motion activity, for instance: acceleration amplitude, duration, and spectrum analysis of seismic waves [4].

The IoT is a consolidated evolutionary paradigm for the Internet and, fortunately, one of the most promising fields which can greatly benefit from its introduction is real-time and distributed environmental monitoring, which implies the possibility to gather data and take data-informed decisions.

The QuakeSense project is an open-source earthquake and environmental monitoring system consisting of a low power and low cost IoT network made of energy-autonomous sensor nodes that are powered through an energy harvesting system and are connected to a gateway in a star topology. The project is based on LoRa, one of the most promising Low Power Wide Area Networks (LPWAN) technologies [5] that provides a good compromise between coverage, current consumption, payload length, bandwidth, and data rate. The collected data are offered to users thanks to a dedicated web-based interface, thus allowing real-time monitoring of both seismic events and environmental parameters.

The conducted experimental campaign mainly aimed at verifying the prototype under several conditions, mainly related to the values assumed by three LoRa peculiar parameters, e.g., Bandwidth (BW) (set to 125, 250, and 500 kHz), Spreading Factor (SF) (spanning from 7 to 12), and Coding Rate (CR) (4/5 and 4/8). The results of the experiments demonstrated that power consumption is optimized and energy footprint is lowered in Idle work-shift. The QuakeSense project demonstrated that LoRa compatibility can effectively be reached in terms of both network parameters and performances. In ordinary conditions, the prototype was able to behave as a remote monitoring platform, thus grating the completeness of gathered information. Further, the system was able to promptly react to earthquakes-like solicitations in a matter of a few seconds, thus demonstrating that remote monitoring tasks could be efficiently carried out.

The remainder of this work is as follows: Section II describes related works and provides motivations for the contribution. Section III overviews LPWAN technologies and justifies the adoption of LoRa. A detailed description of the QuakeSense project is given in Section IV. Section V experimentally evaluates the system and discusses achieved results. Finally, Section VI draws conclusions and proposes future works.

II. Related works and motivations

There are several examples of earthquake monitoring systems that have been developed by research institutes and governments departments such as the Rete Accelerometrica Nazionale (RAN)[1] in Italy. For instance, SafeQuake[2] is a real-time seismic buildings monitoring system, the Multi-Parameter Wireless Sensing System (MPwise) project [6], the MEMSCON project described in [1] but also community-based projects such as the Community Seismic Network (CSN) described in [2]. Most of them are based on wireless sensor networks and rely on Micro Electro-Mechanical Systems (MEMS) accelerometers to acquire data. The choice is

[1]http://www.protezionecivile.gov.it/jcms/it/ran.wp
[2]http://www.i-kubed.com/safequake/

motivated by the fact that these are among the most used tools to detect strong-motion activities. Despite the relevance of the cited projects and solutions, none of them is proposed as an open source prototype implementation, unfortunately.

An interesting contribution in the field has been proposed in [7], a work that exploited rapid prototyping to create an open source project. It aimed at demonstrating the effectiveness of the design methodology and criteria to realize a LoRa-compliant system for environmental monitoring. Unfortunately, the modularity and the expandability of the system are limited, and the proposed platform could not be further extended and enhanced by adding external components or implementing sophisticated data analysis.

The QuakeSense project can be considered as a step ahead in the described landscape, since it aims at solving the downsides of the related works. Moreover, QuakeSense pioneers the LPWAN technologies as to address the peculiar IoT design criteria and typical constraints in the field of earthquakes monitoring systems.

III. Low Power Wide Area Networks

LPWAN can be conceived as an IoT telecommunication protocol family offering wide-area coverage solutions that jointly meet the low power design criteria at the expense of low data rate while trading off on latency. They have been designed for different delay-tolerant applications and use cases that do not require high data rates, and typical LPWAN solutions and systems usually transmit between 10 and 1000 B, as in precision farming, smart metering, industrial assets monitoring, smart city applications, wildlife monitoring, logistic and asset tracking [8]–[13].

The main features of some of the most representative LPWAN technologies are: *Long Range* and *Low power* operation. As for the former, LPWANs adopt different modulation schemes and frequency bands. In terms of modulation schemes, some LPWAN technologies, such as SigFox, Weightless-N, and Telensa, adopt the Ultra Narrow Band (UNB) modulation, while LoRa and Ingenu RPMA rely on Chirp Spread Spectrum (CSS) and Direct Sequence Spread Spectrum (DSSS), respectively. As for frequency bands, instead, LPWAN technologies such as Long Range Wide Area Network (LoRaWAN), SigFox, Telensa, Dash7, and Weightless-M and -P use the sub-GHz Industrial Scientific Medical (ISM) band. The choice is motivated by the fact that this solution offers robust and reliable communications at low power budgets and avoid licensing costs.

The latter, instead, is one of the peculiar IoT constraints, in the LPWAN framework the energy footprint is lowered thanks to: (i) simplified network topologies, (ii) duty cycling policies, and (iii) light-weight Medium Access Control (MAC). For example, the adoption of simple star topologies avoids dense, complex and expensive deployments. As opposed to mesh topologies, this has a significant effect on energy consumed by each end-nodes in the network, which are not responsible for relaying traffic coming from other nodes in the network. Many LPWAN technologies offload complex tasks to the base station or the back-end system in order to maintain the design of end-devices as simple as possible. Duty Cycling is a well-known solution to allow selective and scheduled radio transceivers activations for both uplink and downlink transmissions. LPWAN technologies provide different synchronization methods between the base station and the end-devices. Some LPWAN technologies, such as LoRaWAN and Sigfox, lower the complexity of the design avoiding the usage of complex MAC in favor of simpler solutions. This, once again, demonstrates how LPWANs trade-off between network scalability and simplicity of low cost end-devices.

Almost each of the aforementioned figures of merit and design criteria lead to low per-unit cost, thus affecting the overall cost of the systems in their final version.

A. LoRa Technology

LoRa is a wireless technology that belongs to LPWANs and offers a good compromise between power consumption and performances in terms of latency, bandwidth and coverage. The LoRa protocol stack is divided into three layers: LoRa PHYsical layer (PHY), LoRaWAN (MAC and network layer) and Application layer.

1) LoRa PHY: It is based on a proprietary CSS modulation scheme that uses frequency chirps with a continuously linear variation over time as to encode information and spread the resulting signal over the entire available bandwidth. The solution makes the transmitted signal robust to side effects such as Doppler, multipath and fading. The main communication parameters of LoRa PHY such as throughput, data rate, range and payload size depend on three main settings: BW, SF and CR [5], [14]. BW is the range of frequencies in the transmission bands, and may assume the value of 125, 250, or 500 kHz. As BW increases, data rate increases as well, therefore the transmission time is reduced, but the receiver sensitivity may decrease, thus resulting in worst coverage. The SF, instead, is defined as the ratio between the symbol rate and chip rate, and ranges from 7 to 12. A higher spreading factor may increase the Signal to Noise Ratio (SNR), sensitivity and range. On the other hand it may also increase the time on air. The CR represents the rate of the Forward Error Correction (FEC) technique and offers protection against short bursts of interference. It can be calculated as $CR = 4/(4+N)$ with $N \in 1, \ldots, 4$. Decreasing the coding rate may help reducing the Packet Loss Ratio (PLR), but may increase the time on air. The data rate depends on the aforedescribed parameters, for instance SF, BW, and CR, and can be expressed, in [bps], as:

$$DR = SF \times \frac{BW}{2^{SF}} \times CR \tag{1}$$

2) LoRaWAN: The LoRaWAN network architecture consists of four fundamental components:

- *End-devices*: connected to one, or more, gateways through a single-hop connection.
- *Gateways*: used to communicate with the Network Server to forward packets coming from end-devices as JSON-encoded messages via the Gateway Message Protocol

(GWMP) over a User Datagram Protocol (UDP)/Internet Protocol (IP) stack and a backhaul channel.

- *Network Server*: responsible for decoding and routing packets from each device to the associated application server. It also deals with end-devices management and duplicate messages filtering.
- *Application Server*: responsible for managing join-requests and the handling and encryption of application payload.

A LoRaWAN network adapts both date rate and frequency through a mechanism called Adaptive Data Rate (ADR) as to maximize both battery lifetime of the end-devices and the overall network capacity as a function of channel conditions.

LoRaWAN devices belong to different classes. *Class A* devices implement bidirectional communication on the basis of the Aloha-type MAC. These devices use dedicated receive windows to listen for messages coming from the Network Server. *Class B* supports bi-directional communication, as in Class A, but are also able to open extra receiving windows at scheduled times on the basis of a synchronized beacon received from the gateway. *Class C* devices are always in receive mode, except when they have data to be transmitted. Hence, they have the maximum power consumption while providing minimum latencies. In order to participate in a LoRaWAN network, an end-device must be activated using either Over-The-Air Activation (OTAA) or Activation By Personalization (ABP) [5].

3) Application Layer: The communication primitives between the Network Server and the Application Server are defined by the Application Server interfaces, which typically use common IoT application messaging protocols and Web standards, such as MQTT, Advanced Message Queuing Protocol (AMQP) and HyperText Transfer Protocol (HTTP). Moreover, the Application Servers usually offer web interfaces through which both users and organizations can manage applications and devices thanks to a set of Application Programming Interface (API) to integrate external services, if needed.

IV. The QuakeSense project

Figure 1 graphically describes the QuakeSense system, specifying the envisioned components for each functional block. The QuakeSense project [16] is implemented through an IoT network consisting of autonomous devices called sensor nodes connected to a gateway relying on the LoRa communication technology. Sensor nodes deal with earthquakes detection and environmental monitoring, in fact they send earthquake parameters and environmental data to the gateway. The gateway is responsible for collecting, processing and relaying of messages sent by sensor nodes to an IoT platform called Adafruit IO using the MQTT protocol.

A. Hardware implementation

Figure 2 gives an overview of the project implementation. The hardware perspective of the QuakeSense project mainly addresses the sensor node and the gateway. While this section provides an overview on the main hardware components that

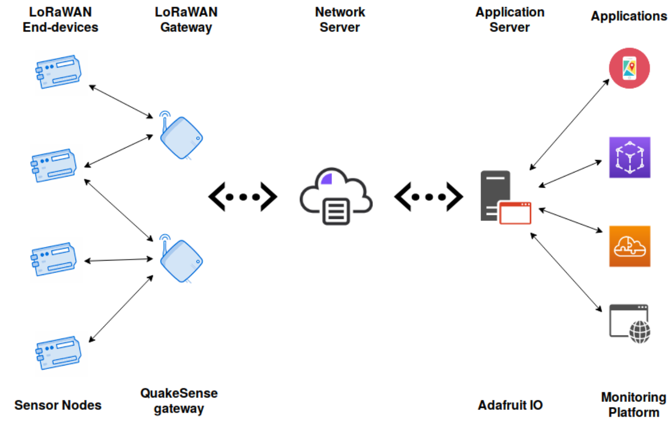


Fig. 1: The QuakeSense project: Network Architecture.

made up the sensor node and the gateway, the complete list of their datasheets is available in [16]. As for the sensor node, that is in charge of collecting environmental data, and communicate them, the main components are:

- STM32 Nucleo-F401RE, a development board based on the ARM Cortex-M4 MicroController Unit (MCU).
- Dragino LoRa/Global Positioning System (GPS) shield, consisting of a low-power, LoRa-compliant radio module (namely RFM95W) and the Quectel L80 GPS module.
- X-NUCLEO-IKS01A2 expansion board.

The last one embeds several MEMS and environmental sensing units, for instance: a 3-axis accelerometer/gyroscope (LSM6DSL), a 3-axis accelerometer/magnetometer (LSM303AGR), a pressure sensor (LPS22HB) and a temperature/humidity sensor (HTS221). The sensor node also includes the Seed Studio Solar Charger Shield v2.2, which is connected to a 2000 mAh rechargeable LiPo battery and a 1.5 W solar panel. Overall, the device is 10 cm (length) x 7 cm (width) x 13 cm (height, with the antenna; 7 cm without it). The overall cost of the hardware was as cheap as 90 €.

The gateway, instead, consists of:

- STM32L4 Discovery Kit development board based on the ultra low-power ARM Cortex-M4 MCU.
- Dragino LoRa Shield that includes a RFM95W radio module based on SX1276 low-power and long-range LoRa transceiver.

B. Software implementation

QuakeSense is, also, an open software project and relies on open technologies and standards such as the STM32duino framework[3] and MQTT. As for the IoT platform, Adafruit IO has been chosen in order to visualize and analyze messages sent by sensor nodes on a rich user interface. MQTT [17] is an application messaging protocol based on the Publish/Subscribe paradigm [17], designed with the purpose of minimizing bandwidth and power consumption, while improving scalability. In MQTT, after a client establishes a connection with the

[3]https://github.com/stm32duino

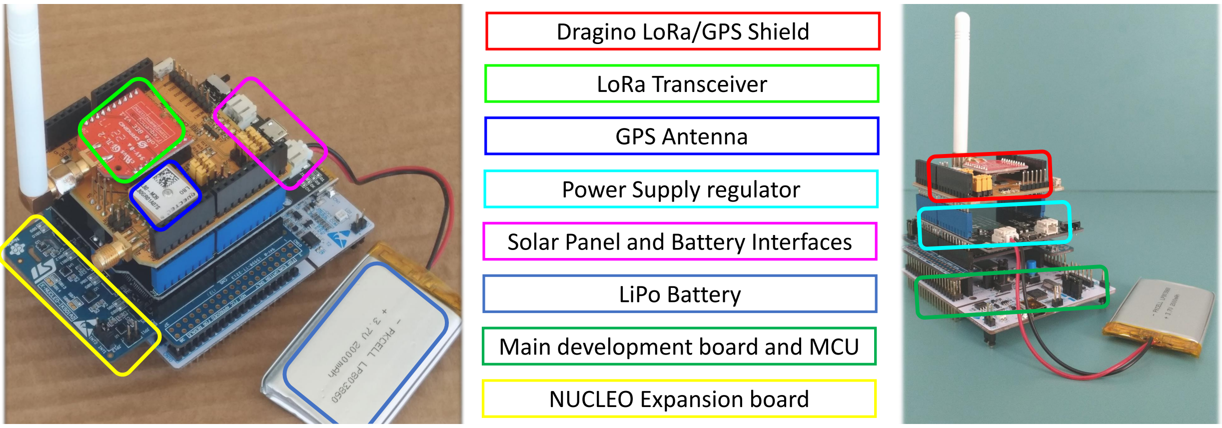


Fig. 2: Detailed overview of the QuakeSense node.

so called broker, it can send Publish messages specifying the topic of interest. In addition, a client can also register to the broker as subscriber to receive messages on topics of interest sent by publishers.

C. *QuakeSense workflow*

By default, the end node runs in Low-Power mode. In this configuration, the STM32 MCU runs in Stop mode, the GPS module is in AlwaysLocate mode, while the LoRa module in Standby mode. In case of emergency, the accelerometer generates an interrupt associated to the wake-up event and the node starts running in Run mode, which results in waking up the STM32 MCU in Full On mode. The wake-up event happens if at least one of the 3 acceleration components exceeds the reference threshold (50 mg for the horizontal components, 1120 mg for the vertical one). Once in Run mode, the node starts reading and recording data to compute the bracketed duration and the 3 components of the Peak Ground Acceleration (PGA) solicitations. The strong-motion activity turns the GPS module in Full On mode so that data are geo-referenced thanks to the GPS module, and include latitude, longitude, altitude, date and time. The Full On mode turns the LoRa module in Transmit mode to send the computed values to the gateway. The baseline behavior of the node foresees a periodical (by default, every 15 minutes) monitoring of the environmental parameters, e.g., temperature, relative humidity and pressure. Each LoRa message is composed by 6 fields:

1) Destination Address (1 B): receiver identifier.
2) Source Address (1 B): sender identifier.
3) Message ID (1 B): message identifier.
4) Payload Length (1 B): message length identifier.
5) Payload (0 - 255 B).
6) Checksum (1 B).

The payload is a variable-length array of two types: Earthquake Alert Message (EAM)[4] (see Figure 3b), and Environmental Data Packet (EDP)[5] (see Figure 3a).

[4]105 B long on average, containing parameters related to a seismic event such as duration, PGA, location, date and timestamp.

[5]33 B at maximum, contains environmental parameters.

temperature	humidity	pressure
11 B	11 B	11 B

(a) EDP payload format

PGHAX	PGHAY	PGVA	duration	latitude	longitude	altitude	date	time
13 B	13 B	13 B	9 B	10 B	10 B	10 B	14 B	13 B

(b) EAM payload format

Fig. 3: Payload compositions.

The gateway is functionally in charge of receiving packets sent by sensor nodes, parsing the encapsulated values and forwarding them to the Adafruit IO platform via the MQTT protocol [17]. The gateway also deals with packets integrity: every time a new packet is received, the gateway computes the checksum and compares it with the one within the received message. In case of a mismatch, the packet is dropped and an error message is sent to the Adafruit IO platform.

V. Tests and Evaluations

This section provides a detailed description of the experimental verification of the QuakeSense project. In a nutshell, the prototype has been verified in terms of current drawn and, hence, energy consumption. The measurements have been done in different functional configurations. The network performances have been evaluated in terms of both network latency and PDR.

As a preliminary consideration of the experimental evaluation, the deployment had to be done in a way that could favor both radio propagation and earthquake detection. The optimal deployment for the device could be done granting direct exposure to sunlight, so to take advantage of its energy harvesting capabilities. Moreover, the deployment point had to be at a certain height from the ground, so to improve seismic phenomena detection. Further, the deployment point had to be sheltered from environmental noise (i.e., wind). To meet all these requirements, rooftops are considered as good candidates [6] [7].

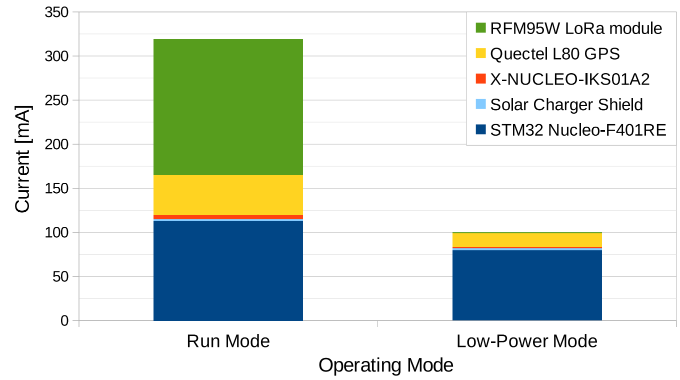
A. Power consumption

The analysis is two-folded, since it demonstrates both the aggregated value of the end device and detailed information about each component of the IoT node. The measurements have been conducted with a probe on the LiPo battery and an INA219[6] as DC current sensor.

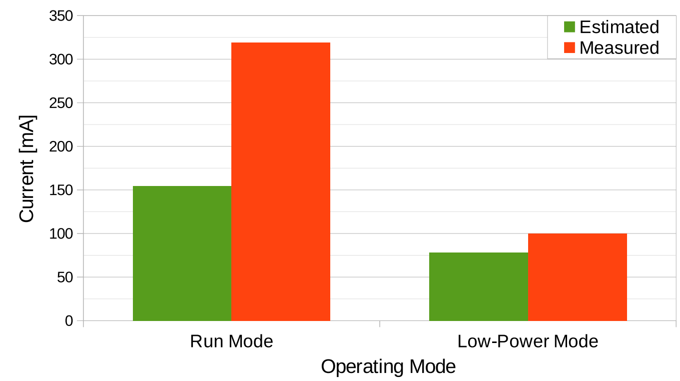
1) Low-Power mode: In this mode, the sensor node accounts for the lowest value in terms of current drawn. For instance, as shown in Fig. 4a, the components of the IoT device contribute in a means of 80 mA, as for the STM32 Nucleo board, while considering the LoRa module, the value resulted to be 1 mA. As for the GPS module, it accounts for 15 mA, whereas the expansion board and the solar charger shield are both characterized by a 2 mA current consumption. Overall, the reference value for this configuration is 100 mA.

2) Run mode: In this mode, all the hardware components of the IoT node are fully operational. Therefore, the overall value of current drawn considering both monitoring and transmission activity resulted to be 319 mA. To such value, the components of the node contributed in a different way. As for the STM32 Nucleo board, its contribution is 113 mA, while the RFM95W and GPS modules account for 154 mA and 45 mA, respectively. The Solar Shield still accounts for a 2 mA current consumption even in Run mode, while the measured current related to the X-NUCLEO-IKS01A2 is about 5 mA.

By combining the measured current consumption values in a reference time interval equal to 1 hour, it has been possible to estimate the battery lifetime. In order to obtain a detailed evaluation, it has been computed a weighted average of 4 current values obtained by subdividing a typical operating cycle of the sensor node into 4 states: (i) Initialization, (ii) Sensing, (iii) Idle and (iv) Active. During initialization, the sensor nodes configures the hardware components. In particular, it initializes the sensors, the RFM95W transceiver and waits 20 seconds to allow the GPS module to fix position. During this phase, both the STM32 MCU and the GPS module are in Full On mode, and thus the current consumption of the sensor node is about 162 mA. The contribution to the weighted average (let it be C_1) is 162 mA $\times$ 0.006 h $=$ 0.972 mAh. After the initialization, the state of the sensor node becomes Idle. In this state, the node operates in Low-Power mode, thus contributing with 100 mA $\times$ 0.977 h $=$ 97.7 mAh (let it be C_2) to the weighted average. As described in the Section IV, by default every 15 minutes the sensor node collects and sends an EDP to the gateway. This is the so-called Sensing status. In particular, the STM32 MCU is woken up to Full On mode, the LoRa module operates in Transmit mode, while the GPS module remains in AlwaysLocate mode. The IoT node transmits 4 packets in the reference time interval (1 hour) and the overall current consumption is 289 mA. Known that the transmission of an EDP takes 0.452 s[7], the contribution to the weighted average (referred to as C_3) is 289 mA $\times$ 0.000125 h $\times$ 4 $=$ 0.145 mAh. Finally, during the Active state, the sensor node deals with monitoring of seismic events. The contribution to the weighted average is (from now on, C_4) $(120 \text{ mA} \times 0.016 \text{ h}) + (319 \text{ mA} \times 0.0003 \text{ h}) = (1.920 + 0.096) \text{ mAh} = 2.016$ mAh. This result is obtained by considering that, first, the sensor node reads acceleration samples from LSM6DSL sensor for 60 seconds (0.016 h) and the STM32 MCU operates in the highest power mode (Full On mode), thus resulting in a partial current consumption of 120 mA. Then, the IoT node runs in Run mode and sends an EAM packet. So, the second term of the sum is obtained by considering the current drawn in Run mode (319 mA) and a time transmission interval of 1.067 s[8]. Given the previous values, it is possible to evaluate the average weighted current consumption with the following equation:



(a) Details on current consumption.



(b) Comparison between estimated and measured current consumption.

Fig. 4: Current consumption overview.

$$\begin{aligned} AvgCurrent &= \frac{C_1 + C_2 + C_3 + C_4}{testDuration} \\ &= \frac{(0.972 + 97.7 + 0.145 + 2.016) \text{ mAh}}{1 \text{ h}} \\ &= 100.833 \text{ mA} \end{aligned} \quad (2)$$

Moreover, considering that the sensor node operates at 3.3 V, it is possible to estimate the average power consumption

[6]http://www.adafruit.com/datasheets/ina219.pdf

[7]Computed using the Semtech LoRa calculator [15].

[8]Computed using the Semtech LoRa calculator [15].

as 0.1 Ah $\times$ 3.3 V = 0.33 Wh. Finally, knowing the battery capacity (i.e., 2000 mAh), its estimated duration results to be:

$$BatteryDuration = \frac{2000 \text{ mAh}}{100.833 \text{ mA}} = 19.83 \text{ h} \tag{3}$$

Battery lifetime has also been evaluated when the system effectively uses the solar panel. Results demonstrated that after a 26 hours work-shift, the battery had a remaining charge of 32% (corresponding to 640 mAh). Therefore, the prototype could still operate up to a maximum of 38 hours. Overall, the ratio between the Run mode configuration and the Low-Power one in terms of maximum current consumption is about 3.19%. It is worth noting (see Fig. 4b) that measured values differ from theoretical ones of about 100% more in Run mode and about 30% more in Low-Power mode.

B. Network performances

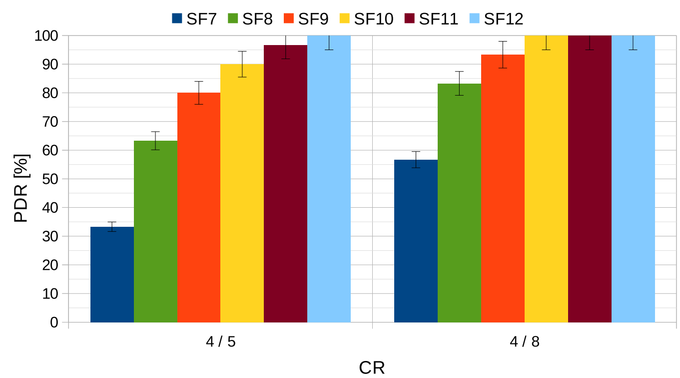
The experimental analysis on network performances has been carried out considering two of the typical network parameters, for instance PDR and network latency. As for the former, it is defined as the ratio between the number of packets received correctly (without errors) and the total number of packets transmitted. The latter, instead, has been evaluated as the time interval between the sending of a LoRa packet by the sensor node and its reception on the Adafruit IO platform. Further, the LoRa communication technology has been functionally evaluated over a 24-hours long experimental campaign during which a total of 2160 packets were sent at a fixed pace of 40 seconds. The LoRa settings adopted to perform the tests are shown in Table I. For each configuration of the LoRa settings (SF, BW and CR), the sensor node sends 6 packets (3 EDPs and 3 EAMs) and each set of transmissions is repeated 10 times. When the gateway receives a packet, it checks its integrity using the checksum contained in the packet's trailer and, then, sends an MQTT message to the Adafruit IO platform containing information regarding the status of the received packet. Tests carried out have shown

Distance node-gateway [m]		822
BW [kHz]		125, 250, 500
SF		[7 - 12]
CR		4/5 and 4/8
TX power [dBm]		+14
Frequency Band [MHz]		868
Packet length [B]	EDP	33
	EAM	105

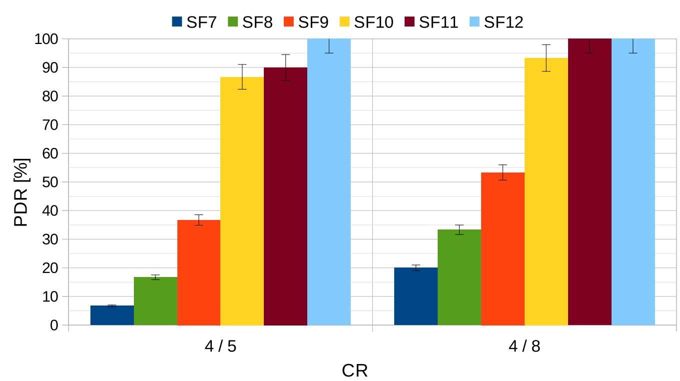
TABLE I: LoRa configuration settings

that setting BW to 125 or 250 kHz leads to a PDR greater than 90% for EDP packets, while in case of EAM packets, it is not less than 73%. In particular, another significant result is that setting BW = 250 kHz, if CR = 4/8, PDR is equal to 100% for both EAM and EDP packets independently to the SF, else if CR = 4/5 only SF = 7 makes difference since, for all the other values of SF, the loss ratio remains 0%. The main interesting results have been obtained for BW = 500 kHz and are described in Fig. 5 which shows the trend of PDR at varying of the SF and CR. It confirms that, when setting BW to the maximum admissible value (500 kHz), in order to increase the PDR it is necessary to increase the SF and/or the CR especially when also the number of transmitted symbols increases too. In fact, the experimented PDR related to EDPs is higher than EAMs since the former have a smaller payload size than the latter.

In terms of latency, the system demonstrated a linear trend with values of about 4 s, with the sole exception of the configuration with BW = 125 kHz (see Fig. 6). In particular, it has been measured that when the SF increases, the latency is increased too in a measure of the 50%, independently from the CR. In addition, the maximum measured value for the latency is about 12 s, while the average value is equal to 5.47 s. Finally, it is possible to state that the prototype offers constant performance in term of latency, except for isolated configurations of LoRa parameters.



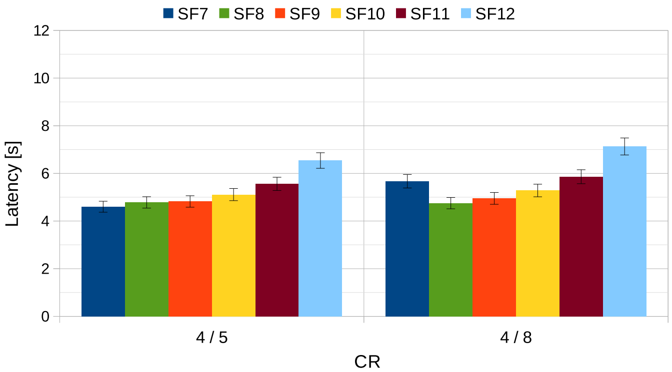
(a) EDP



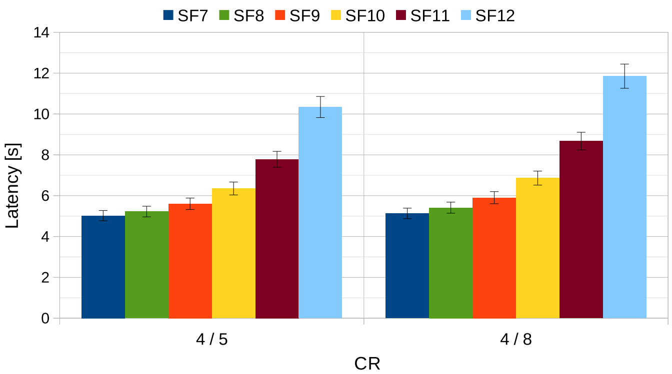
(b) EAM

Fig. 5: PDR for BW = 500 kHz

The best values for PDR have been obtained with BW = 250 kHz; with almost all the SF values (8-12), PDR is equal to 100%. Furthermore, considering the experimented values obtained for BW = 125 kHz, the minimum value assumed by the PDR is 81.7%. This means that, for distances lower than 1 km, the system is very suitable for applications that use the LoRaWAN protocol in the EU863-870MHz frequency band. On the other hand, the results regarding latency tests confirm that the higher the BW, the lower the latency. In addition, setting the BW to 250 or 500 kHz results the latency



(a) EDP



(b) EAM

Fig. 6: Latency for BW = 125 kHz

at varying constantly with respect to the SF. In summary, since the system must be as fast and reliable as possible, especially in the presence of seismic events, it is reasonable to consider a configuration of LoRa parameters that guarantees both low latency and high PDR. Consequently, combining the results obtained for latency and PDR, the following configuration: BW = 250 kHz, SF = 10 and CR = 4/5, which represents a good trade-off between reliability (PDR is 100%), range (822 m) and latency (5.165 s). For the sake of completeness, in optimal radio propagation conditions, such a communication range (i.e., radius) implies that the overall coverage area is about 2 km^2.

VI. Conclusions

This work presented the QuakeSense project, an open-hardware and open-software implementation of a LoRa-compliant environmental monitoring system specifically conceived for monitoring seismic events. The platform leverages the main characteristics of IoT technologies in terms of low power consumption, payload size and extended coverage capabilities. The LoRa end-node communicates gathered data to a single-channel LoRa gateway, which is responsible to forward data to the Adafruit IO platform using the MQTT protocol. An experimental evaluation has been conducted to verify the energy footprint and LoRa performances, in terms of PDR and latency. As for the former aspect, the end-node demonstrated an average current consumption of about 100 mA while transmitting critical data. As for the latter, instead, the best trade-off on network performance have been achieved setting the LoRa parameters as: BW = 250 kHz, SF = 10 and CR = 4/5, which resulted in a PDR = 100% and a latency of about 5 s at more than 800 m.

Despite the results, the prototype should be further tested, first of all over longer distances and under different conditions. A multiple node deployment is certainly among the future work possibilities. In the unfortunate case the accelerometer breaks up, the sensor node will not be able to reveal earthquakes. In the future, it will be evaluated a timer-based routine that periodically checks the status of the accelerometer. The check could also be done during the monitoring of the environmental parameters. Such functionality would imply a new assessment of energy consumption. From a software perspective, it could be enhanced relying on some of the most well-known operating systems specifically designed for supporting IoT devices, for instance RIOT-OS[9] and Mbed OS[10].

Acknowledgment

This work was partially founded by Italian MIUR PON projects Pico&Pro (ARS01 01061) AGREED (ARS01 00254), FURTHER (ARS01 01283), RAFAEL (ARS01 00305) and by Apulia Region (Italy) Research Projects E-SHELF (OSW3NO1) and INTENTO (36A49H6).

References

[1] T. Torfs, T. Sterken, S. Brebels, J. Santana-Corte, R. Van den Hoven, N. Saillen, N. Bertsch, D. Trapani, and D. Zonta, "Low power wireless sensor network for structural health monitoring of buildings using mems strain sensors and accelerometers," in *2nd MEMSCON Workshop: Towards Intelligent Civil Infrastructure*. Ghent University, Department of Electronics and information systems, 2012, pp. 37–44.

[2] M. Faulkner, M. Olson, R. Chandy, J. Krause, K. M. Chandy, and A. Krause, "The next big one: Detecting earthquakes and other rare events from community-based sensors," in *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks*, 2011.

[3] J. Goulet, C. Michel, and A. D. Kiureghian, "Data-driven post-earthquake rapid structural safety assessment," *Earthquake Engineering & Structural Dynamics*, vol. 44, no. 4, pp. 549–562, 2015.

[4] A. Elnashai and L. Di Sarno, *Fundamental Concepts of Earthquake Engineering*. John Wiley & Sons, 2008.

[5] A. Augustin, J. Yi, T. Clausen, and W. Townsley, "A study of lora: Long range & low power networks for the internet of things," *Sensors*, vol. 16, no. 9, 2016.

[6] T. Boxberger, K. Fleming, M. Pittore, S. Parolai, M. Pilz, and S. Mikulla, "The multi-parameter wireless sensing system (mpwise)," *Sensors*, vol. 17, no. 10, 2017. [Online]. Available: http://www.mdpi.com/1424-8220/17/10/2400

[7] M. Pulpito, P. Fornarelli, C. Pomo, P. Boccadoro, and L. A. Grieco, "On fast prototyping LoRaWAN: a cheap and open platform for daily experiments," *IET Wireless Sensor Systems*, vol. 8, no. 5, pp. 237–245, 2018.

[8] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low power wide area networks: An overview," *IEEE Communications Surveys Tutorials*, vol. 19, no. 2, pp. 855–873, 2017.

[9] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "A comparative study of lpwan technologies for large-scale iot deployment," *ICT Express*, vol. 5, no. 1, pp. 1–7, 2019.

[9] https://www.riot-os.org/

[10] https://www.mbed.com/en/platform/mbed-os/

[10] U. Raza, P. Kulkarni, and M. Sooriyabandara, “Low power wide area networks: An overview,” *IEEE Communications Surveys Tutorials*, vol. 19, no. 2, pp. 855–873, 2017.
[11] Q. M. Qadir, T. A. Rashid, N. K. Al-Salihi, B. Ismael, A. A. Kist, and Z. Zhang, “Low power wide area networks: A survey of enabling technologies, applications and interoperability needs,” *IEEE Access*, vol. 6, pp. 77 454–77 473, 2018.
[12] R. S. Sinha, Y. Wei, and S.-H. Hwang, “A survey on LPWA technology: LoRa and NB-IoT,” *ICT Express*, vol. 3, no. 1, pp. 14 – 21, 2017.
[13] A. Ikpehai, B. Adebisi, K. M. Rabie, K. Anoh, R. E. Ande, M. Hammoudeh, H. Gacanin, and U. M. Mbanaso, “Low-Power Wide Area Network Technologies for Internet-of-Things: A Comparative Review,” *IEEE Internet of Things Journal*, pp. 1–1, 2019.
[14] Semtech, “Lora^{TM} modulation basics,” Tech. Rep., 2015, https://www.semtech.com/uploads/documents/an1200.22.pdf.
[15] *SX1272/3/6/7/8 LoRa Modem Design Guide*, Semtech, available online: https://www.semtech.com/uploads/documents/LoraDesignGuide_STD.pdf (accessed on 26 April 2019).
[16] *List of components datasheets and software libraries.*, The QuakeSense project, 2019, available online: https://github.com/biagiom/QuakeSense/blob/thesis/README.md.
[17] *MQTT Version 3.1.1.*, edited by Andrew Banks and Rahul Gupta. 29 October 2014. OASIS Standard. Latest version: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html.