

## Research Article

# FROG: A Robust and Green Wireless Sensor Node for Fog Computing Platforms

**Manuel Castillo-Cara** <sup>1</sup>, **Edgar Huaranga-Junco** <sup>1</sup>, **Milner Quispe-Montesinos**,<sup>1</sup>  
**Luis Orozco-Barbosa** <sup>2</sup> and **Enrique Arias Antúnez** <sup>2</sup>

<sup>1</sup>Center of Information and Communication Technologies, Universidad Nacional de Ingeniería, Peru

<sup>2</sup>Albacete Research Institute of Informatics, Universidad de Castilla-La Mancha, Spain

Correspondence should be addressed to Luis Orozco-Barbosa; [luis.orozco@uclm.es](mailto:luis.orozco@uclm.es)

Received 30 September 2017; Revised 9 March 2018; Accepted 25 March 2018; Published 12 April 2018

Academic Editor: Bruno Andò

Copyright © 2018 Manuel Castillo-Cara et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Over the past few years, we have witnessed the widespread deployment of wireless sensor networks and distributed data management facilities: two main building blocks of the Internet of things (IoT) technology. Due to the spectacular increase on the demand for novel information services, the IoT-based infrastructures are more and more characterized by their geographical sparsity and increasing demands giving rise to the need of moving from a cloud to a fog model: a novel deployment paradigm characterized by the provisioning of elastic resources geographically located as close as possible to the end user. Despite the large number of wireless sensor networks already available in the market, there are still many issues to be addressed on the design and deployment of robust network platforms capable of meeting the demand and quality of fog-based systems. In this paper, we undertake the design and development of a wireless sensor node for fog computing platforms addressing two of the main issues towards the development and deployment of robust communication services, namely, energy consumption and network resilience provisioning. Our design is guided by examining the relevant macroarchitecture features and operational constraints to be faced by the network platform. We based our solution on the integration of network hardware platforms already available on the market supplemented by smart power management and network resilience mechanisms.

## 1. Introduction

Over the past years, the development of wireless networks, mobile devices, and computing paradigms have given rise to the introduction of a large amount of information and communication-assisted services. From Smart City applications to the management of rural areas, Internet of things (IoT) technologies are being deployed to monitor or assist the operation of a wide variety of control and production processes. Nowadays, behind city traffic control and food production tasks, IoT systems are being used to monitor the resources and effectiveness of the actions that are manually or automatically being taken in response to the information provided by tiny sensors deployed across the monitored area. For instance, the use of IoT systems have been used to monitor the environmental conditions of vineyards and the growth of grape bunches over a season [1].

Despite the promising results that have been obtained up-to-date, there are still many issues to be addressed on the design and implementation of robust IoT-based systems capable of meeting the end-user application requirements. Besides the issues to be addressed in order to improve the operation of the underlying communication platforms, another major issue has to deal with the design of a scalable and robust data processing architecture. Towards this end, the data processing architecture of IoT systems has moved from a cloud paradigm to a fog paradigm. The latter has been designed bearing in mind that for a large number of applications, the main consumers of the information obtained from the sensor data may be located close to the data sources. Furthermore, many application may have stringent quality-of-service requirements, such as real-time, security, and nonstop operation requirements. Some applications may also be characterized by their high data rates; for example,

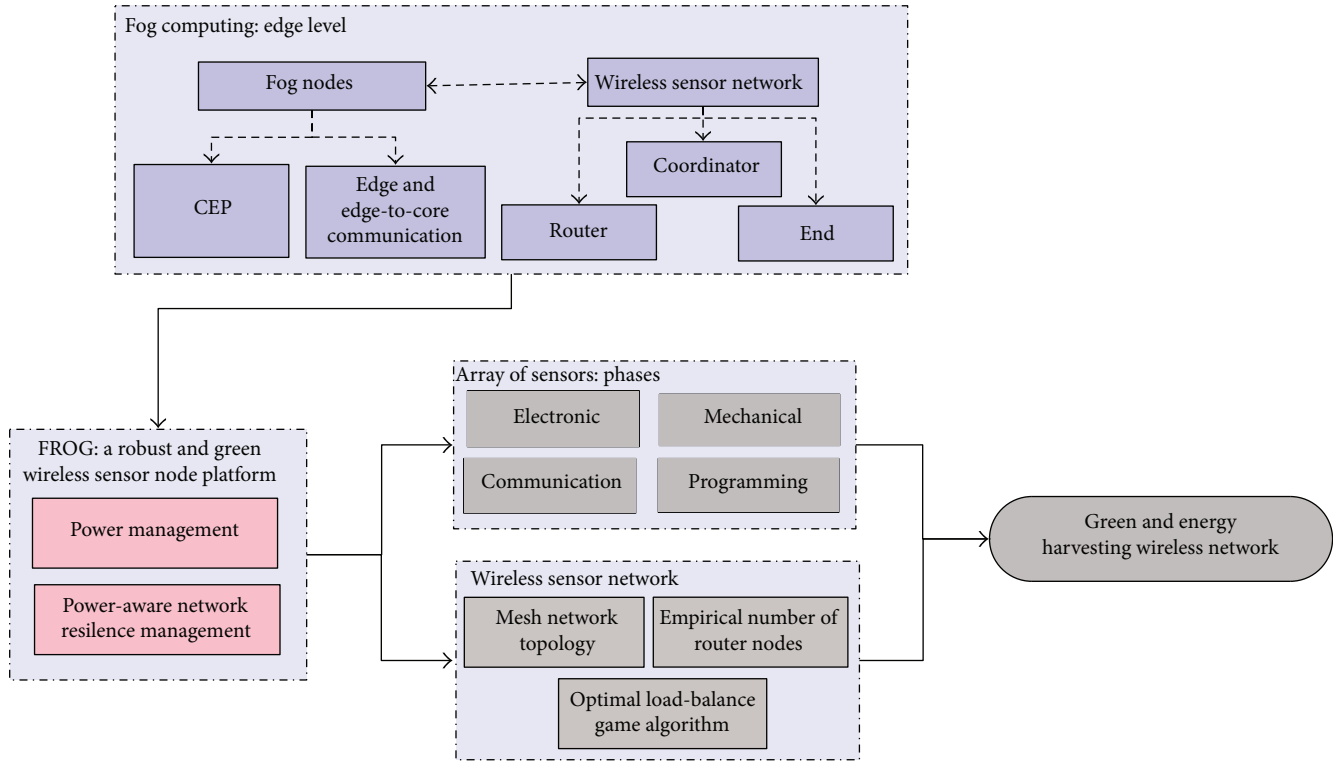


FIGURE 1: Overall schema proposal.

a city traffic monitoring system may have to report on events rising during the peak hours: traffic jams, accidents among others.

From the above, it is clear that the services to be provided by the IoT platforms will heavily depend on the underlying communication facilities. In this paper, we focus at two main network design parameters: power management and network resilience mechanisms. The former plays a key role on the robustness of a wireless sensor network deployed in areas of difficult access and often exclusively powered by batteries. Our solution takes into account the application profile as a means to define an application-aware power management mechanism [2]. As for the second issue, network resilience, it is well known that such networks will often have to self-heal based on events, such as the energy depletion of a node providing relay services to other nodes; or due to upgrade or architectural changes, such as a change on the number of nodes covering a certain area or the change of the geographical location of a data processing server. It should be clear that the nodes should collaborate in order to be able to deliver the data service as required by the end-user application. Both of these two design parameters are clearly interrelated. The power consumption of a given node heavily depends on the network mechanisms to be performed by the nodes; that is, a node serving as a router will be required to forward the packets generated by their neighbors while a leaf node only has to take care of its own traffic. As for the self-healing network mechanisms, they will require to get to know the power resources at the time of deciding the role to be assigned to a node; for example, an end node may be called to operate as a router.

Since one of our goals is to develop a robust and self-healing fog platform to be used in various Smart City and Smart Farming applications, we take as basis the open-source communication hardware platform. This decision has been motivated by the large number of different sensors available in the market offering larger coverage range: two main features towards the deployment of an ever increasing number of citywide applications.

Numerous works have been already reported on the use of open-source hardware and software technologies in Smart City and Smart Farming applications [3–6], implementation of protocol mechanisms [7], or using data fusion to reduce the power consumption [8].

Due to the increasing number of parameters to be monitored, we also address the design of a configurable sensor node platform: a platform consisting of a configurable array of sensors, from now on sensor node. This is an important issue to be taken into account when developing platforms to be deployed in the fields, for example, Smart Farming applications [9] as well as in numerous Smart City applications and weather and pollution monitoring, [10–12]. Our work therefore undertakes a holistic approach towards the design of configurable wireless node platforms integrating power management and network resilience mechanisms. Figure 1 shows the overall proposed methodology. The ultimate goal is to develop smart fog computing platforms capable of providing a wide range of end-user applications including the monitoring of complex event processing (CEP) and the provisioning of edge-to-core communications.

In this paper, we focus on the design and integration of the three building blocks, bottom boxes in Figure 1.

We start by designing a configurable node platform integrating off-the-shelf components, that is, sensors and wireless network platforms. We then incorporate into our design the aforementioned power management and network resilience mechanisms.

In the following, the paper is organized as follows. Section 2 reviews the related work and describes the main contribution of our work. Section 3 reviews the principles and architecture of the fog computing paradigm. We also explain the role played by each node making a part of the edge network infrastructure. Section 4 details our proposal. We first introduce the power management mechanism and explain its implementation based on the functionalities of the Arduino platform. In the second part of this section, we provide the rationale behind the design of the power-aware network resilience mechanism. Based on our discussion, we specify the operation of our proposal.

In Section 5, we first evaluate the benefits of an onboard power management mechanism. This first evaluation shows the benefits of our application-aware power management mechanism. We then evaluate the power-aware resilience mechanism. We finally spell out our conclusions and future research plans in Section 6.

## 2. Related Work

Power consumption management of wireless sensor networks has long attracted the attention of many research groups. While many studies have looked into solutions ranging from the use of alternative power sources, that is, solar energy [13], other studies have centered on the design of power-aware protocol mechanisms [14, 15]. With the main goal of improving the overall network operation, fog computing has emerged to extend the cloud computing paradigm to the edge of the network [16]. Fog computing has been created to address, among others, the mobility, geographical, and latency requirements of most IoT applications [17]. Due to the requirement when deploying an increasing number of interconnected devices, power management and self-healing mechanisms have become two of the key design parameters of fog computing platforms [18–20]. In [18], the authors have identified the main features of the underlying network mechanisms of a fog computing platform, namely, location, distribution, scalability, density of devices, mobility support, real-time, and standardization. Moreover, the network infrastructure in a fog computing platform must be autonomous and efficient in terms of the energy consumption and network resilience [21].

As for the available communication technologies, the IEEE 802.15.4 ZigBee standard, from now on ZigBee, has attracted the attention of many designers. Various studies have shown the benefits of ZigBee in the context of various Smart City and Smart Farming applications. In [22], the authors have evaluated the connectivity, packet loss rate, and transmission throughput in an indoor experimental area. A hybrid network approach, ZigBee/5G, has been described in [23]. The authors have shown the effectiveness of integrating various protocol mechanisms aiming at reducing the power consumption. On the other hand, the authors from

[24] have evaluated and compared the power consumption of ZigBee versus the Advanced and Adaptive Network Technology (ANT). According to their analysis, ANT outperforms ZigBee in terms of the energy consumption implemented through a sleep/wake algorithm. However, their study has also revealed that by properly managing the overall platform, radio, and sensors, the power consumption reported by the ZigBee systems can be considerably reduced. These results make ZigBee an excellent experimental platform to study the benefits of integrating power consumption management mechanisms. Furthermore, the wider coverage range of ZigBee makes it an excellent development platform for the development of Smart City and Smart Farming applications.

In a recent work, Piromalis and Arvantis have introduced a scalable hardware architecture for wireless sensor and actuator networks to be used in Smart Farming applications [21]. While our work follows a similar approach on the design of the sensor node platform, our work focuses on the power consumption management and network resilience mechanisms taking into account the latest trends on the fog computing paradigm. That is to say, our approach brings into the design of the sensor node platforms, some of the major features of the fog computing paradigm, mainly, geographical location, latency, and power management.

## 3. Fog Computing: Principles and Technologies

The fog computing paradigm can be simply defined as a natural extension of the cloud computing paradigm. Under the fog computing paradigm, the data processing and user services are performed by smart devices, called fog nodes, closely located to the end users.

In this context, the fog computing architecture is divided into two levels, core and edge [19]. The components and technologies of each level are adapted to the services to be provided by each one of them, see Figure 2. The core comprises the main computing components of the system, for example, the main database and broker, among others [18], while the edge level is comprised by other components: wireless sensor and actuators networks and the fog nodes. The latter comprises a local broker, local databases, and a processing and visualization server [18, 25, 26]. The fog nodes are in charge of processing the time-sensitive data while making a better use of the underlying network infrastructure.

In this work, we focus on the edge level of fog computing architectures and in particular on the features and operation of each one of the edge elements. In fact, this work contributes on enhancing the intelligence of the network nodes and the autonomy, including the self-healing mechanisms, of the wireless network infrastructure. In order to fully justify our contribution, we start by briefly describing the main processing tasks of the two main components of an edge infrastructure: the fog nodes and the wireless sensor network. We will also highlight the features and processes to benefit from our proposal.

*3.1. Fog Nodes.* The fog nodes are the main processing units located closely to the data nodes. They process the data

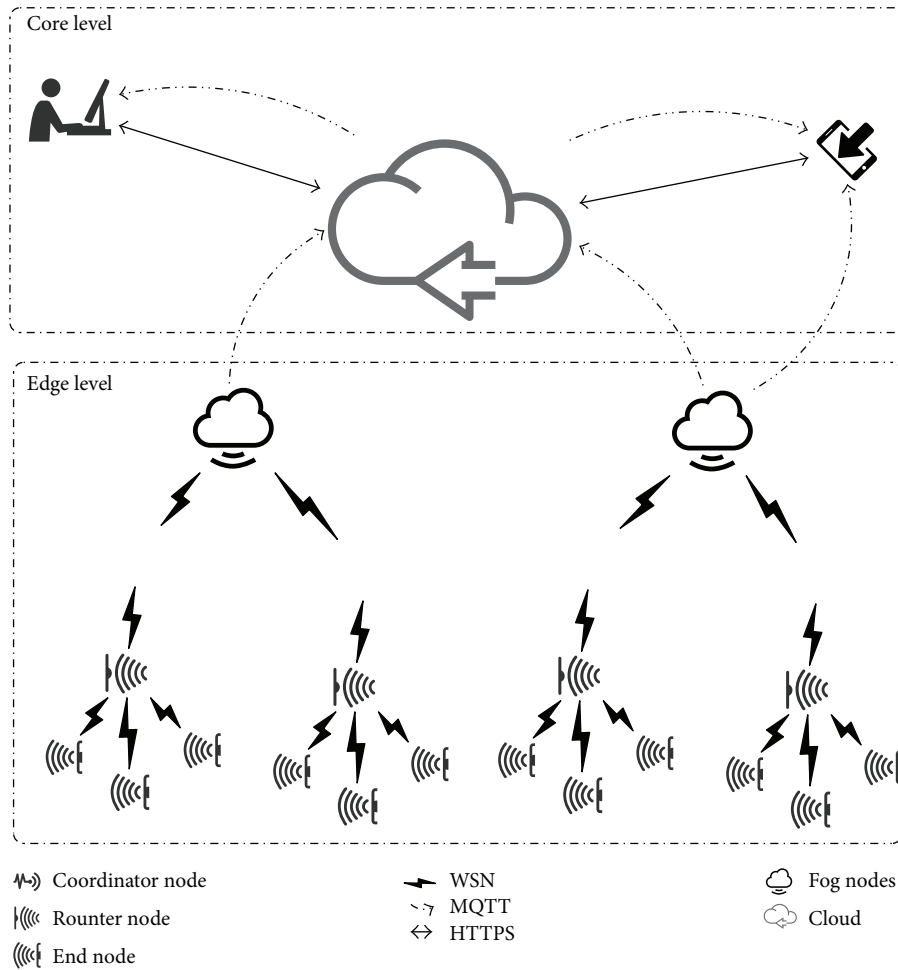


FIGURE 2: General fog computing architecture.

captured by the sensor node and are also responsible for filtering and delivering the relevant data to be stored in the cloud. Their two main tasks can be described as follows:

- (i) Complex event processing (CEP). This task refers to the processing fusion of the data collected by the sensor nodes. The main outcome of this task is to notify stakeholders of patterns derived from events of the lower level [25]. Tools, such as Apache Flink, are being used as a CEP motor in the fog node for local events and in the cloud level for general events. A fog node processes the different events based on the information extracted from the data collected by the sensor nodes. At the edge level, an initial analysis is carried out. As the second step, the fog node sends the data and events to the core level via the Internet reducing the processing and analysis to be done at the core level.
- (ii) Edge and edge-to-core communications. Fog nodes are in charge of sending (1) local alerts to the subscribers and (2) the sensed data and events to the core level via the Internet [26]. Nowadays, many platforms use the MQTT protocol: a lightweight protocol where the devices send (publish)

information with a label (topic) to a server that works as a broker. The broker sends information to all subscribers to the referred topic [19]; that is, (i) the communication mechanisms are implemented in a local broker implemented at the fog node and (ii) a global broker built into the cloud facilities [26].

**3.2. Wireless Sensor Network.** In the following examples, we based our design on the ZigBee standard. This standard provides the following advantages for our proposal: (i) long distances between sensors because of the low frequency; (ii) a hierarchical topology that can be deployed in environments with high dimensions such as a Smart City or Smart Farming; and (iii) a hierarchical definition of the roles played by the various node elements enabling the natural interconnection with the fog nodes. The three node roles can be simply described as follows:

- (i) Coordinator node. Every ZigBee network needs to have a single coordinator node. This device has the following features whose main functions are network initialization, distribution of addresses, and management operations.

- (ii) Router node. Router nodes receive the data packets generated by the sensor nodes and deliver them to the coordinator node. In order to ensure the reliable delivery of the data generated by the end nodes, every end node should be able to communicate with at least one router node. In our configuration, the router node has been implemented using an off-the-shelf hardware platform equipped with various sensor devices and a long-range RF interface. A customized programmable *on/off* switch has been designed in order to implement the power management mechanism. The implementation of this latter switch was deemed necessary to enable the use of analog sensor devices. The routers also implement the self-healing network mechanism.
- (iii) End node. The main mission of an end node is to monitor its environment. An end node always relies on a router node or coordinator node to make part of the network. The end nodes have also been implemented using off-the-shelf hardware platforms. Similarly to the router nodes, the end nodes also implement the dynamic power management mechanism.

Based on the above discussion, we will proceed to explain the two main contributions of our proposal, namely, the power management mechanisms and the self-healing protocol.

#### 4. FROG: A Robust and Green Wireless Sensor Platform

In this section, we first introduce the power management mechanism of our proposal. Our design also includes the implementation of the required electronics to properly couple the sensor nodes to the platform. In the second part, we describe the network resilience-provisioning mechanism.

**4.1. Power Management.** A large number of works have been reported in the literature aiming to manage the power consumption of sensor nodes. Many of such works have defined mechanisms to turn *on/off* the radio system, see for instance [27]. Many works were initiated at the time when the sensor nodes comprised a limited number of sensors, typically ranging between two or three sensors. In the last few years, the number of sensors that can be integrated into one node platform has considerably increased. The main reason behind this trend has been motivated by the development of novel applications in sectors, such as Smart Cities and Smart Farming. This increase has resulted in a higher energy demand to power the onboard processing and monitoring functions [2]. Furthermore, the data acquisition requirements, time, and data amount to be processed, greatly vary from one application to another.

In this work, we follow the latest trends on developing a multiple-purpose sensor platform capable of dynamically adapting its duty cycle to the application [28]. Different from the early proposals introduced in the literature, the proposed power management mechanism consists basically on turning *on/off* the power to be delivered to the sensors, rather than

dealing with the radio system. This choice has also been made taking into account the latest development on low power radio transceivers.

In order to illustrate the benefits of the proposed mechanism, we will take the case depicted in Figure 3 where we have plotted the level of CO<sub>2</sub> reported by the sensors placed in our premises using three different sampling periods, 5 s, 10 s, and 20 s. As seen from Figure 3, the trend of the data is for the most predictable; that is, from the data reported with the lowest sampling resolution, we could easily predict other values. We further notice that there are few exceptions where there is a significant change on the values reported by the sensors. Based on the previous analysis and bearing in mind the wide spectrum of applications, including the monitoring of sensitive data, we proceed to define a dynamic power management scheme meeting the requirements of a large number of potential applications. In other words, the design should take into account all the safety requirements by properly fixing the sampling rate and transmission periods. Accordingly, the management mechanism only applies to the sensor nodes; the main controllers should always be powered up. In this way, we should be able to integrate into our platform both paradigms: synchronous and asynchronous event scheduling.

In this context, many commercial sensors already count with the logic to program the *on/off* periods. For instance, Figure 4 depicts the connection diagram of a digital sensor based on the well-known I2C protocol. However, the connection of simpler sensor devices require the development of a programmable digital switch, see Figure 5.

The switch is simply controlled by the programmable power management mechanism. The duty cycle is commanded using the digital signal generated by the main onboard processor connected to the base of the NPN (negative-positive-negative) transistor Q1. When the pulse is generated, the circuit is activated. The sensors are powered by connecting the power input of the sensors to the collector of Q2 and to the common ground, GND. It is important to mention that the collector voltage decreases although transistor emitter Q2 is connected to the 5 V source. This fact occurs because when the transistor switches to the *on* mode, there is a potential drop between the emitter and the collector of Q2 whose value is  $V_{EC} = 0.7$  V approximately. Nevertheless, the sensor operation is not affected by this slight potential drop [29]. In fact, multiple sensors can be connected to the transistor collector Q2 and a common ground.

Finally, Figure 6 shows the sensor node platform developed in this work. The same platform may be used to implement the router nodes and end nodes. As for the packaging, we have designed a box facilitating the installation of various sensors. We have also designed the chambers to store the volume of air required to determine the level of concentration meeting the specifications of most gas sensors. Moreover, we have also provisioned the platform with a battery charger to be integrated with the follow-up of our project.

**4.2. Power-Aware Network Resilience Management Mechanism.** Based on the underlying programmable power management facility presented in Section 4.1, we developed herein

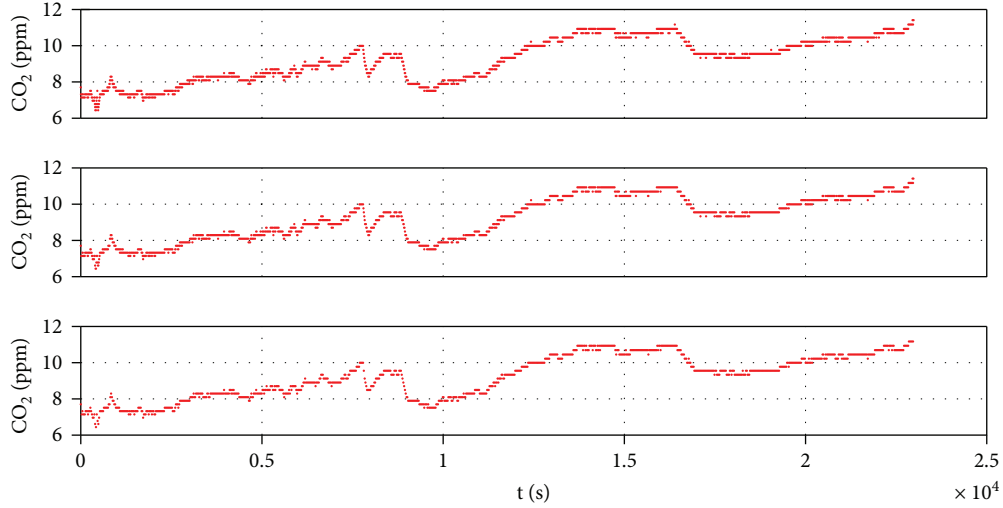


FIGURE 3: Data traces for CO<sub>2</sub> concentration using three different sampling periods, from top to bottom: 5, 10, and 20 seconds.

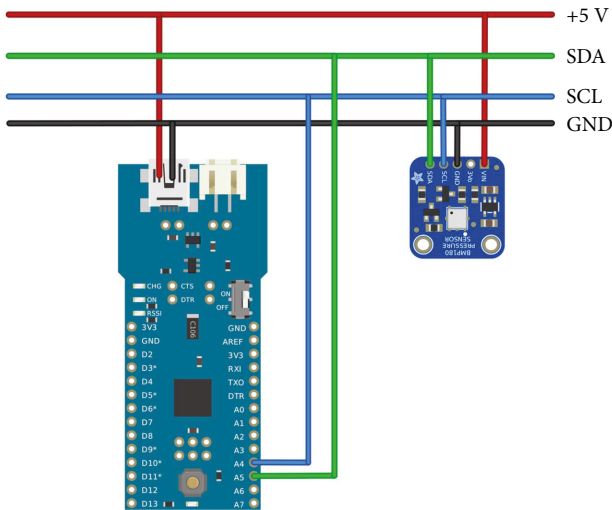


FIGURE 4: Physical connection of a digital sensor using the I2C protocol.

a power-aware network-wide resilience mechanism. The main goal of this latter mechanism is to provide the means to enhance the robustness of the wireless sensor network.

Figure 7 depicts the general topology of the wireless sensor network on which we base the description of our proposal. As seen in Figure 7, the network is composed of three types of sensor nodes, namely, the coordinator, router, and end nodes. The coordinator node is responsible of initializing the sensor network and ensuring its connectivity to the fog premises. This type of nodes is normally implemented by a highly reliable node. The other two types of nodes, routers and end nodes, are the subject of our proposal.

Since the router nodes are in charge of forwarding the data collected by the end nodes and monitoring the environment, it is obvious that they are exposed to higher load demands than the ones handled by the end nodes. We then propose to replicate the number of router nodes providing service to a given number of end nodes. At all times and in

order to get the most of the available resources, the router and end nodes will monitor the environment making use of their onboard sensors.

In order to show the benefits and gains in terms of power consumption, let us define the following relation assuming that the sensor nodes are initially fully charged:

$$\begin{aligned} E_r(t) &= 100 - K_r \times t, \\ E_e(t) &= 100 - K_e \times t, \end{aligned} \quad (1)$$

where  $E_r$  is the energy level of the router node at time  $t$ ,  $E_e$  is the energy level of the end node at time  $t$ ,  $K_r$  is the discharge rate as router node, and  $K_e$  is the discharge rate as end node.

Since the discharge rate depends on the task performed by the sensor nodes, it is clear that given that a router node provides service to one or more end nodes, the discharge rate of a router node is higher than the discharge rate of an end node. The relation between the discharge rate of a router node and an end node can be simply expressed as follows:

$$K_r = K_e + N_e \times M, \quad (2)$$

where  $N_e$  is the number of end nodes connected to the router node and  $M$  denotes the average energy consumed at the router node by each end node connected to it up to time  $t$ .

As seen in Figure 7, three sensor nodes may play the role of router nodes, while six sensor nodes always operate as end nodes. Our goal is to find the number of routers required to guarantee the robustness of the network without leaving out of service any end node. Let us take the following three scenarios. Notice that in our analysis, we do not include the energy consumed by the router node(s) to serve the six end nodes, since this energy consumption will be the same for all the three scenarios under study.

- (i) One router node and two end nodes: only one of the three sensor nodes located at the group of router/end nodes operates as a router node. The other two end

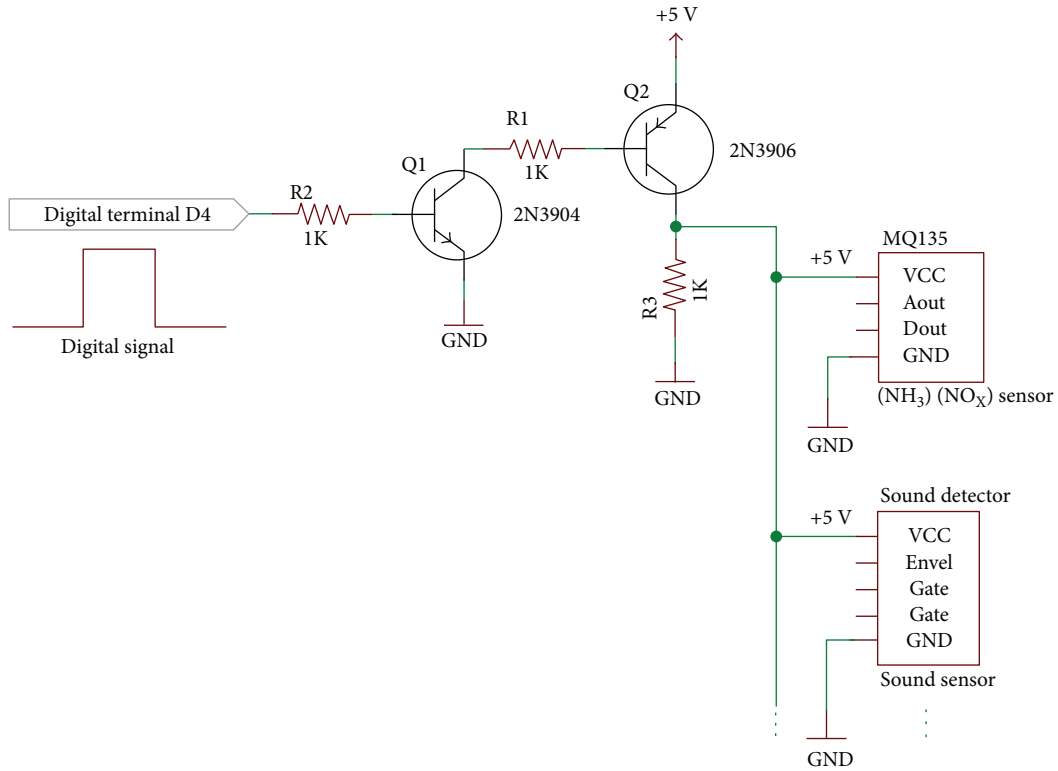


FIGURE 5: Digital on/off switch.

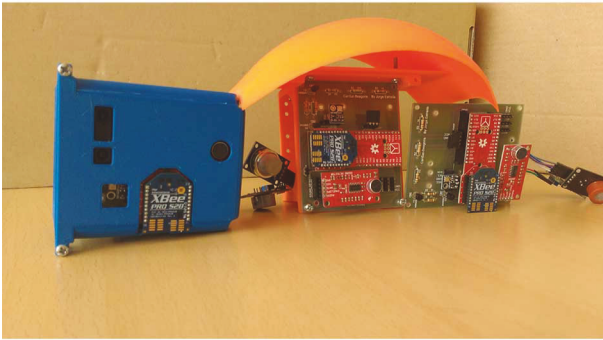


FIGURE 6: FROG node platform.

nodes are connected to the active router node. The power consumption for this case can be simply stated as follows:

$$E_{\text{level2}}^1 \text{ router}(t) = 100 - K_r \times t + 2 \times (100 - (K_e \times t)) = 300 - (K_r + 2K_e)t. \quad (3)$$

- (ii) Two router nodes and one end node:  $L$  of the end nodes located at the group of nodes are connected to one of the two router nodes, while  $(8 - L)$  sensor nodes are connected to the second router node. The third sensor node located at the group of the router/end node is connected to one of the two

active router nodes. The overall energy demand can be simply expressed as follows:

$$E_{\text{level2}}^2 \text{ routers}(t) = 2 \times (100 - K_r \times t) + 100 - K_e \times t = 300 - (2K_r + K_e)t. \quad (4)$$

- (iii) Three router nodes:  $L_1$  end nodes are connected to the first router node,  $L_2$  to the second router node, and  $L_3$  to the third router node. Then, the overall energy demand is expressed as follows:

$$E_{\text{level2}}^3 \text{ routers}(t) = 3 \times (100 - K_r \times t) = 300 - (3K_r)t. \quad (5)$$

Summarizing, for every possible configuration on the second level, we have the following power consumption:

- (i) One router node + two end nodes =  $(K_r + 2K_e)t$   
(ii) Two router nodes + one end node =  $(2K_r + K_e)t$   
(iii) Three router nodes =  $(3K_r)t$

Since  $K_r > K_e$ , we have the following power consumption:

$$K_r + 2K_e < 2K_r + K_e < 3K_r. \quad (6)$$

From (6), it is clear that the use of one router node exhibits the best results in terms of the power requirements. The implementation of a lightweight protocol should provide the means to take over the role of the router node by another

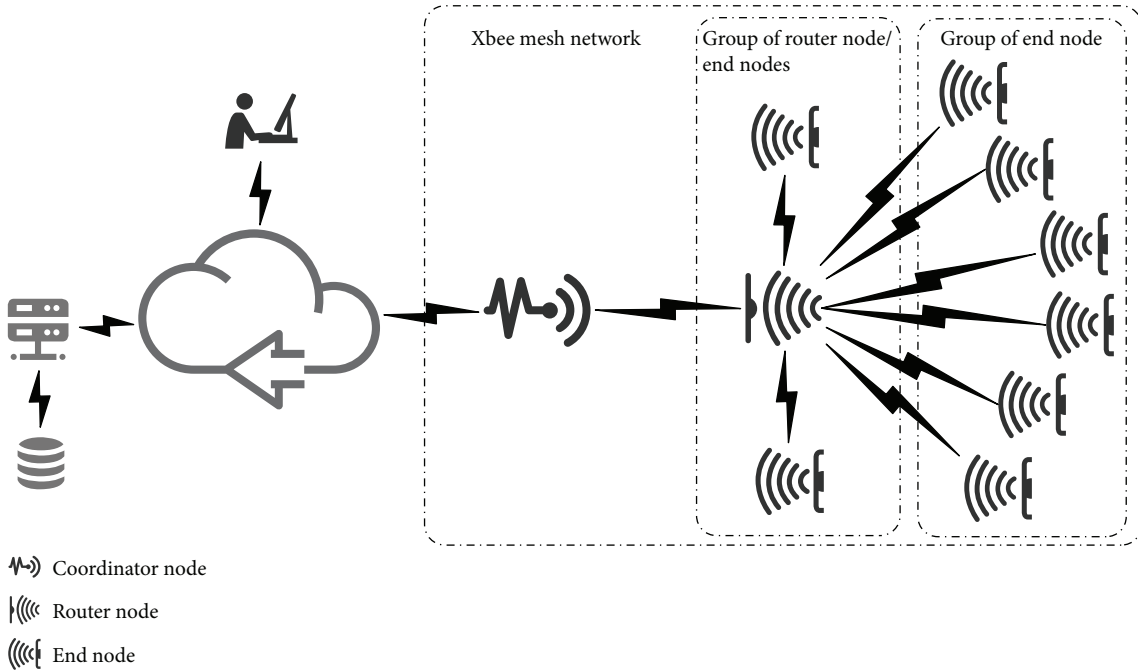


FIGURE 7: Wireless sensor network topology.

sensor node located within the coverage range of the current router node.

In order to design and develop the network resilience algorithm, we must take into account that the coordinator node is the one in charge of controlling the network. In order to take the decision of replacing a router node, we must verify the status of the batteries of the current sensor node and the one of all those sensor nodes that may take over the routing task. Algorithm 1 specifies the operation of our network resilience mechanism.

One of the main characteristics of our proposal is its simplicity. Since the number of sensor nodes involved on the role-changing task is fixed to three sensor nodes, the reliability of the mechanism is practically guaranteed. In fact, such configuration allows us to schedule maintenance tasks while ensuring the presence of the backup node at all times. That is to say, the sensor node taking over the routing task will be the one in charge of taking over the in-transit traffic ensuring that no packets get lost. Furthermore, our algorithm prevents the presence of potential loops involving end nodes and router/end nodes.

## 5. Experimental Evaluation

In this section, we start by describing our experimental setup. We first conduct the evaluation of the sensor node platform in terms of onboard power savings, and then, the evaluation of the power-aware network resilience mechanism.

*5.1. Experimental Setting.* Our experimental setting counted with the following elements:

- (i) FROG platforms. We have assembled nine FROG units. Each FROG unit comprises a set of sensors

```

1: procedure BALANCEGAME
2:   timeLoop ← 1
3:   lookForRouter ← FALSE
4:   while TRUE do
5:     secondLevelDevices ← coordinatorDevice.
findConnectedDevices()
6:     indexDevice ← 0
7:     for indexDevice < secondLevelDevices.size() do
8:       device ← secondLevelDevices[indexDevice]
9:       if device.getRole() == ROUTER-NODE then
10:        if device.getBatteryLevel() <= 50 then
11:          device.setRole(END-NODE)
12:          lookForRouter ← TRUE
13:        end if
14:      else
15:        if lookForRouter == TRUE then
16:          if device.getBatteryLevel() >= 70 then
17:            device.setRole(ROUTER-NODE)
18:            lookForRouter ← FALSE
19:          end if
20:        end if
21:      end if
22:    end for
23:    sleep(timeLoop)
24:  end while
25: end procedure

```

ALGORITHM 1: (Network Resilience Algorithm).

with a total current consumption requirement of 475.48 mA. We implemented the processor and communications platform using an Arduino Fio V3 and an XBee-PRO XSC radio system whose current consumption requirements add up to 266.5 mA.



The total FROG node consumption is, processor plus sensors, 741.98 mA. Following the baseline topology depicted in Figure 7, we have placed three FROG platforms within the router/end node sector. Throughout our experiments and based on our previous analysis, we have considered the case where only one of the three sensor nodes will act as a router node at a given time. In the case when the power management mechanism is enabled, the duty cycle is set with an *off* period of 5 s and an *on* period of 1 s. We have further implemented a coordinator node using a single-board computer, Raspberry Pi 3 Model B. This latter system has been plugged into the power line. As already mentioned, we will add a battery charger as part of our future plans.

- (ii) Batteries. Table 1 summarizes the main characteristics of the Li-Po batteries used throughout our trials.
- (iii) Voltage level monitoring circuit. An Arduino UNO shield was used to monitor the voltage level of the batteries during our experiments. In order to be able monitor the voltage values in the range of 0 to 5 V using the 12 V batteries, we implemented a voltage divider consisting of a 2 k $\Omega$  resistor and a potentiometer,  $R_2$ , see Figure 8.

Since the battery voltage diminishes during the experiments, the setting of the potentiometer had to be adjusted accordingly. A Matlab script was used to dynamically change the setting of  $R_2$ . A second Matlab script was used to monitor and plot the battery voltage as a function of time.

**5.2. Onboard Power Consumption Evaluation.** In the first set of experiments, we monitored the discharge of the batteries by first disabling the power management mechanism, and then enabling the power management mechanism of FROG. We repeated our experiments ten times. It is worth mentioning that the results were very similar in all ten trials.

Figures 9(a) and 9(b) show the results for both platform configurations using Li-Po batteries. In the case when the platform does not implement the proposed power management mechanism, the first deep voltage drop is reported after  $3.5 \times 10^4$  s from the beginning of the test. As seen in Figure 9(a), the discharge time for this type of battery is approximately  $9 \times 10^4$  s. This result confirms the theoretical bound given by the ratio of the battery rate being used, 20C, divided by the current consumption of our platform, 741.98 mA [30]. In our case, this ratio is approximately equal to  $9.7 \times 10^4$  s (26 hours).

Figure 9(b) shows the results for the FROG platform. As seen from Figure 9(b), the battery voltage slowly diminishes and no deep drop is reported during the evaluation period. These results clearly show the benefits of our proposal. However, we should realize that the optimal setting of the power management mechanism depends on various factors, such as the sampling rate (latency) of the actual data to be reported and number of sensors integrated into the platform for a given target application. Furthermore, the integration of asynchronous handling of events, such as the CEP to be

TABLE 1: Technical specifications of the batteries.

Specification	Li-Po battery
Nominal voltage	11.1 V
Voltage per cell	3.7 V
Number of cells	3
C rate	20C
Capacity	6400 mAh

integrated into our system, will play a major role on the actual duty cycle and more specifically on the sensors being powered during a given period of time. Take for instance, the general case where each sensor may be turned on at different time instances and for periods of variable length. Following the discussion similar to the one in [30], the power management strategy will prove beneficial if the following condition holds for the period of operation of the system,  $t$ :

$$\sum_{i=1}^n P_{on}^i \cdot t_{on}^i + \sum_{i=1}^n P_{off,on}^i \cdot t_{off,on}^i \leq \sum_{i=1}^n P_{on}^i \cdot t, \quad (7)$$

where  $t_{on}^i$  denotes the time during which the sensor,  $i$ , is powered up; the corresponding power consumed by sensor  $i$  during this period of time by  $P_{on}^i$ , and the cost of the transition and time delay from *off* to *on* is expressed by  $P_{off/on}^i$  and  $t_{off/on}^i$ , respectively. Finally,  $t$  denotes the total running time. Our experimental results have shown that considerable power saving can be achieved by simultaneously turning *on/off* all the sensor devices. In fact, several of the sensor devices we have used comprise numerous sensors. Furthermore, the power consumed during the *off/on* transition is due to the electronic switch we have developed. We can therefore rewrite (7) to reflect our platform operation as follows:

$$t_{on} \sum_{i=1}^n P_{on}^i + t_{off,on} \sum_{i=1}^n P_{off,on}^i \leq \sum_{i=1}^n P_{on}^i \cdot t. \quad (8)$$

As seen from the last expression, the fact of turning on all the sensors at the same time reduces considerably the power consumption. The second term of the left-hand side of (8) only contributes once during a duty cycle.

From the above analysis, it is clear that the proper scheduling of the data gathering process and setting of the duty cycle play a major role on the power consumption. However, the solution best fitting the needs of a specific application will have to take into account other requirements, such as the latency and sampling rate. Furthermore, the implementation of a CEP engine will require the handling of both synchronous and asynchronous events.

**5.3. Network Resilience Mechanism Evaluation.** In this section, we evaluate the performance, in terms of the energy consumed, of the FROG platform operating in each one of the two roles: router node and end node. In this case, we placed the nine FROG nodes throughout the hall of our Faculty department. The duty cycle was set with an *on* period of 1 s and an *off* period of 5 s. All the nodes transmit a packet

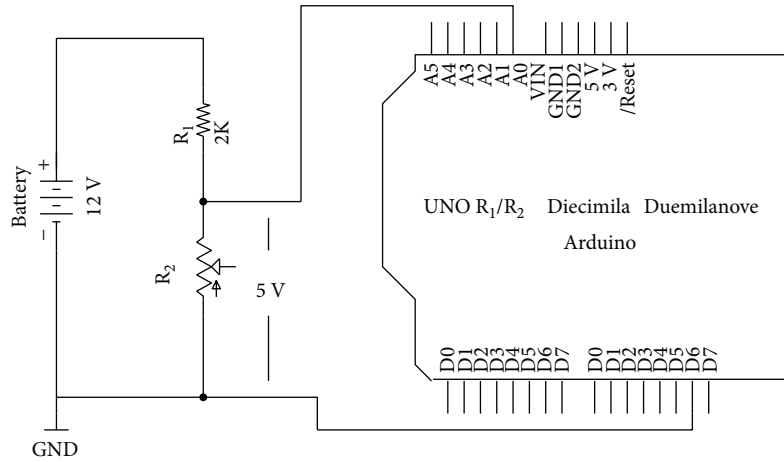
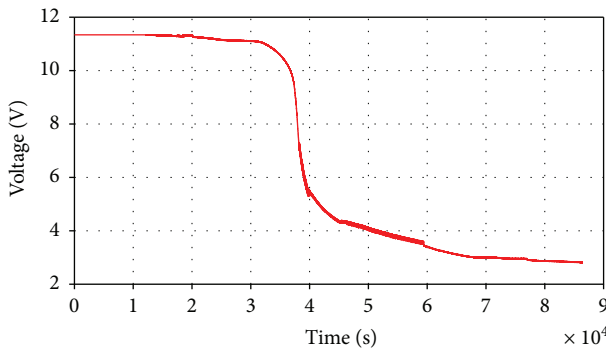
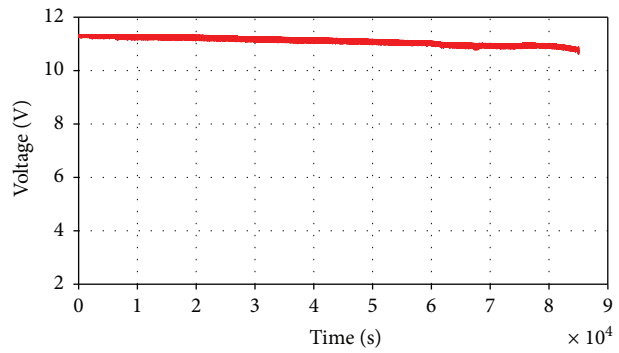


FIGURE 8: Voltage monitoring circuit.

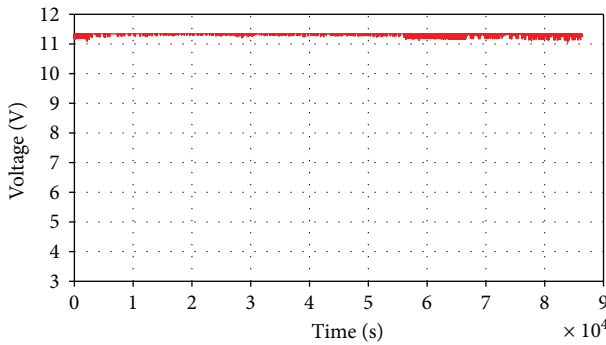


(a) Platform without the power management mechanism

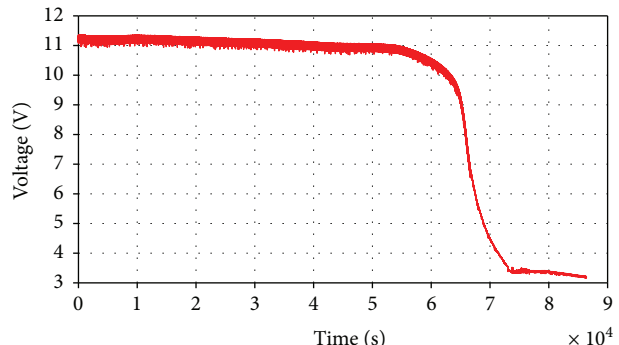


(b) FROG

FIGURE 9: Voltage Vs. Time results for the Li-Po battery.



(a) End node



(b) Router node

FIGURE 10: Voltage Vs. Time results for the Li-Po battery.

conveying the data generated by the sixteen sensors: one sample for each one of the nine sensor devices. Similarly to the previous evaluation trials, we repeated ten times the trials under a given setup. At the beginning of each trial, the battery was replaced by brand new ones.

The experiment ran for a period of  $8.5 \times 10^4$  s (23.61 hours). Figure 10(a) shows the energy consumption

for the end node and Figure 10(b) shows the results for the router node. As seen from the results, the router node consumes much more energy than the end node. We also monitored the number of packets having been successfully received. No losses were reported despite the handover performed as the battery of the router crosses the threshold defined in the algorithm. These results confirm once again

the need to dynamically change the role played by the sensor nodes at times, as specified in Algorithm 1, when they encounter power provisioning problems.

## 6. Conclusions and Future Plans

In this paper, we have developed FROG: a robust and green wireless sensor network platform to be integrated into the fog architecture. Our design has been fully justified after a review of the current trends on the use of IoT technology in various types of Smart City and Smart Farming applications.

We based our design on the use of open software and open hardware platforms. We therefore made use of some of the facilities and functionalities already available on the market. We then design and develop a power management mechanism and a network resilience algorithm. We show that the use of a reduced number of sensor nodes capable of acting as a relay or end node as required will provide the means to considerably improve the network operation while lowering the power required for the overall network operation.

Our immediate research will focus on developing an experimental platform including Smart City applications. The system will include various end-user services based on a complex event processing (CEP) system and the fog computing paradigm. We will also explore the use of software defined networks (SDNs).

## Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this article.

## Acknowledgments

This work has been partially funded by “Cienciaactiva - CONCYTEC” of the Peruvian government, under Grant no. 138-2017-FONDECYT and by the Spanish Ministry of Economy and Competitiveness under Grant no. TIN2015-66972-C5-2-R.

## References

- [1] L. Orozco-Barbosa, F. M. García, A. B. Ramos, and F. M. Riquelme, *An Experimental Evaluation of IoT Technologies in Precision Viticulture*, Springer International Publishing, Cham, 2017.
- [2] N. Vallina-Rodríguez and J. Crowcroft, “Energy management techniques in modern mobile handsets,” *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 179–198, 2013.
- [3] A. Kumar, K. Kamal, M. O. Arshad, S. Mathavan, and T. Vadamala, “Smart irrigation using low-cost moisture sensors and XBee-based communication,” in *IEEE Global Humanitarian Technology Conference (GHTC 2014)*, pp. 333–337, San Jose, CA, USA, 2014.
- [4] H.-H. Lin, H.-Y. Tsai, T.-C. Chan et al., “An open-source wireless mesh networking module for environmental monitoring,” in *2015 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*, pp. 1002–1007, Pisa, Italy, 2015.
- [5] V. Jeličić, T. Ražov, D. Oletić, M. Kuri, and V. Bilas, “MasliNET: a wireless sensor network based environmental monitoring system,” in *2011 Proceedings of the 34th International Convention, MIPRO*, pp. 150–155, Opatija, Croatia, 2011.
- [6] B. Jolly, A. Willig, A. McDonald, M. Pannell, and G. Plank, “SNOWWEB—wirelessly connected weather stations in Antarctica,” in *38th Annual IEEE Conference on Local Computer Networks—Workshops*, pp. 194–202, Sydney, NSW, Australia, 2013.
- [7] T. U. Sane, S. L. Shue, and J. M. Conrad, “Implementation of dynamic source routing using 802.15.4 on XBee series 1 modules,” in *SoutheastCon 2015*, pp. 1–8, Fort Lauderdale, FL, USA, 2015.
- [8] Y. Solahuddin and W. Ismail, “Data fusion for reducing power consumption in Arduino-XBee wireless sensor network platform,” in *2014 International Conference on Computer and Information Sciences (ICCOINS)*, pp. 1–6, Kuala Lumpur, Malaysia, 2014.
- [9] F. R. J. López and A. J. López, “Field variables monitoring in real time (GPS, soil moisture, temperature) with precision farming applications,” in *EATIS '12 Proceedings of the 6th Euro American Conference on Telematics and Information Systems*, pp. 1–5, Valencia, Spain, 2012.
- [10] P. A. L. Besari, M. Abdurrohman, and A. Rakhmatsyah, “Application of M2M to detect the air pollution,” in *2015 3rd International Conference on Information and Communication Technology (ICoICT)*, pp. 304–309, Nusa Dua, Bali, 2015.
- [11] M. Noushad, B. Tauheed, S. A. Khan, and M. A. Khan, “Wireless monitoring of temperature and humidity using sensor array,” in *2015 Annual IEEE India Conference (INDICON)*, pp. 1–5, New Delhi, India, 2015.
- [12] M. Mitoi, A. Vulpe, R. Craciunescu, G. Suci, and O. Fratu, “Approaches for environmental monitoring sensor networks,” in *2015 International Symposium on Signals, Circuits and Systems (ISSCS)*, pp. 1–4, Iasi, Romania, 2015.
- [13] W. B. Pramono, P. Setiawan, and Firdaus, “Solar power supply for ZigBee wireless sensor network,” in *2016 International Seminar on Application for Technology of Information and Communication (ISemantic)*, pp. 336–340, Semarang, Indonesia, 2016.
- [14] T. Matsui and H. Nishi, “ECORS: energy consumption-oriented route selection for wireless sensor network,” in *2016 IEEE 14th International Conference on Industrial Informatics (INDIN)*, pp. 1044–1049, Poitiers, France, 2016.
- [15] C. Pérez-Garrido, F. González-Castaño, D. Chaves-Dieguez, and P. Rodríguez-Hernández, “Wireless remote monitoring of toxic gases in shipbuilding,” *Sensors*, vol. 14, no. 2, pp. 2981–3000, 2014.
- [16] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, “Fog computing and its role in the internet of things,” in *MCC '12 Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, pp. 13–16, New York, NY, USA, 2012.
- [17] D. Spirjakin, A. Baranov, A. Karelin, and A. Somov, “Wireless multi-sensor gas platform for environmental monitoring,” in *2015 IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems (EESMS) Proceedings*, pp. 232–237, Trento, Italy, 2015.
- [18] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of things: a survey on enabling technologies, protocols, and applications,” *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

- [19] T. H. Luan, L. Gao, Z. Li, Y. Xiang, G. Wei, and L. Sun, "Fog computing: focusing on mobile users at the edge," <http://arxiv.org/abs/1502.01815>.
- [20] K. C. Okafor, I. E. Achumba, G. A. Chukwudebe, and G. C. Ononiwu, "Leveraging fog computing for scalable IoT data-center using spine-leaf network topology," *Journal of Electrical and Computer Engineering*, vol. 2017, Article ID 2363240, 11 pages, 2017.
- [21] D. Piromalis and K. Arvanitis, "SensoTube: a scalable hardware design architecture for wireless sensors and actuators networks nodes in the agricultural domain," *Sensors*, vol. 16, no. 8, p. 1227, 2016.
- [22] J.-S. Lee and Y.-M. Wang, "Experimental evaluation of ZigBee-based wireless networks in indoor environments," *Journal of Engineering*, vol. 2013, Article ID 286367, 9 pages, 2013.
- [23] J. Mu, "An improved AODV routing for the ZigBee heterogeneous networks in 5G environment," *Ad Hoc Networks*, vol. 58, pp. 13–24, 2017.
- [24] S. Gharghan, R. Nordin, and M. Ismail, "An ultra-low power wireless sensor network for bicycle torque performance measurements," *Sensors*, vol. 15, no. 5, pp. 11741–11768, 2015.
- [25] M. B. A. P. Madumal, D. A. S. Atukorale, and T. M. H. A. Usoof, "Adaptive event tree-based hybrid CEP computational model for fog computing architecture," in *2016 Sixteenth International Conference on Advances in ICT for Emerging Regions (ICTer)*, pp. 5–12, Negombo, Sri Lanka, 2016.
- [26] A. Garcia-de-Prado, G. Ortiz, and J. Boubeta-Puig, "COLLECT: COLLaborativE ConText-aware service oriented architecture for intelligent decision-making in the internet of things," *Expert Systems with Applications*, vol. 85, pp. 231–248, 2017.
- [27] P. Díaz, F. Royo, T. Olivares, F. Ramirez-Mireles, and L. Orozco-Barbosa, "A case study on the power-aware protocol framework for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 9, no. 12, Article ID 718252, 2013.
- [28] U. Raza, A. Camerra, A. L. Murphy, T. Palpanas, and G. P. Picco, "Practical data prediction for real-world wireless sensor networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 8, pp. 2231–2244, 2015.
- [29] R. L. Boylestad and L. Nashelsky, *Electronic Devices and Circuit Theory*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 11th edition, 2013.
- [30] W. Dargie, "Dynamic power management in wireless sensor networks: state-of-the-art," *IEEE Sensors Journal*, vol. 12, no. 5, pp. 1518–1528, 2012.



**Hindawi**

Submit your manuscripts at  
[www.hindawi.com](http://www.hindawi.com)

