

Practical Design of a WSN to Monitor the Crop and its Irrigation System

Laura Garcia^{1,2}, Lorena Parra¹, Jose M. Jimenez^{1,2}, Jaime Lloret¹, Pascal Lorenz²

¹Instituto de Investigación para la Gestión Integrada de zonas Costeras, Universitat Politècnica de València.

C/ Paranimf nº 1, Grao de Gandía – Gandía, Valencia (Spain)

²Network and Telecommunication Research Group, University of Haute Alsace, 34 rue du Grillenbreit, 68008, Colmar. France

E-mail: laugarg2@teleco.upv.es, loparbo@doctor.upv.es, jojiher@dcom.upv.es, jlloret@dcom.upv.es, lorenz@ieee.org

Received: Sep. 30, 2018 Accepted: December 30, 2018 Published: December 31, 2018

DOI: 10.5296/npa.v10i4.14147

URL: <https://doi.org/10.5296/npa.v10i4.14147>

Abstract

Every day it becomes more necessary to control crops because of the environmental problems, such as the lack of water for irrigation. Therefore, the use of precision agriculture becomes more evident. When it comes to making decisions on crops, the need of applying the concept of Smart Agriculture, which focuses on utilizing different sensors and actuators, is evident. As the number of IoT devices used in agriculture grows exponentially, it is necessary to design the implemented network so that the data is transmitted without problems. This paper shows a wireless network design, in which we use the information collected by the sensors of a Wireless Sensor Network (WSN), and a Wireless Mesh Network (WMN) formed by Access Points (AP) to transmit the data to a network that monitors the crops and its irrigation system. In addition, through simulations, we show the maximum number of nodes that should be connected to an AP in order to have an efficient network.

Keywords: nodes, sensors, actuators, algorithm, protocol, wireless, design.

1. Introduction

Precision agriculture is gaining more relevance throughout the years. Countries with large tradition in agriculture are implementing smart agriculture systems. Furthermore, research institutes and companies that focus their activity on smart agriculture and precision agriculture are increasing as well. Smart agriculture focuses on utilizing different sensors and actuators that are part of a system that monitors crops and takes decisions according to the readings. These decisions are reflected on the actions performed by the actuators. Irrigation has been considered in existing precision agriculture systems. However, monitoring the quality of the water that is going to be employed for irrigation has started to be considered for new smart agriculture proposals [1]. The reason for this interest in water quality monitoring for irrigation began as a result of the water scarcity problem that affects several areas belonging to Mediterranean countries. Treated wastewater is apt for crops irrigation and allows reducing water consumption [2]. However, in order to be able to use wastewater for irrigation, some water cleanliness requirements must be met. Therefore, monitoring water quality before performing irrigations tasks allows verifying the correct quality of the water.

In order to perform the aforementioned activities, different sensors must be utilized. These sensors are designed for the field or the water according to its purpose. For field monitoring, there are varied low-cost commercial sensors that allow monitoring soil moisture and humidity, as well as rain and other environmental conditions. However, the available commercial sensors for water quality monitoring present a very high cost which turns out in the resulting increase of the price of deploying a smart agriculture system. Nevertheless, recent researches on water quality monitoring have provided inspiring results on low cost sensors for turbidity [3] and conductivity [4] monitoring. The use of low-cost sensors for smart and precision agriculture allow reducing the price of said systems and open the door to the implementation of these systems in developing countries. Therefore, both the quantity of the production and the quality of the resulting produce would increase which, at the same time, could lead to better nutrition of the people of the area and a reduction in the prices of the produce.

The information from the sensors deployed on the field or in the water is gathered utilizing boards such as those provided by Arduino. Due to the characteristics of agricultural environments, wired communication among the nodes is practically impossible. The employed machinery could sever the wires, and animals and weather conditions would lead to the necessity of installing protections for all the wires of the system. Therefore, wireless communication is the norm for these types of scenarios. Nowadays, after the surge of IoT (Internet of things), several protocols have been designed to meet the characteristics and requirements of these kind of networks. CoAP (Constrained Application Protocol) and MQTT (Message Queuing Telemetry Transport) are an example of recently developed protocols for IoT that are increasing in popularity. Both are application layer protocols but differ on the transport protocol they utilize. While MQTT utilizes TCP, CoAP works on top of UDP. CoAP has been utilized in some implementations of smart environmental monitoring systems [5]. It is a simplified version of HTTP and thus it is possible to bridge both protocols. However, as it is intended for low sized packets, the use of SigFox or LoRA is more considered for CoAP.

This fact has led to few implementation resources for CoAP with WiFi antennas. Furthermore, the observe option, that was further explained on a posterior RFC [6], is not fully implemented on the available libraries for CoAP over WiFi chips such as the ESP8266.

In addition, the number of nodes is an important factor to consider when designing the network and the equipment that is going to be utilized so as to avoid the saturation of the network. In this paper, we propose the design of a WSN (Wireless Sensor Network) for crop monitoring and decision taking on the actions to be taken regarding the irrigation of said crops. CoAP is utilized for the communication employing the observe option by implementing a modification of an existing library so as to allow access to several resources and the possibility of having several observers.

The rest of the paper is organized as follows. Section 2 presents the related work. The architecture overview is depicted in Section 3. The simulation results are discussed in Section 4. Finally, the conclusion and future work is presented in Section 5.

2. Related Work

WSN have been greatly utilized for deploying smart systems for precision agriculture. Marco Mancuso et al. presented in [7] a WSN for environmental monitoring in a tomato greenhouse. Sensicast devices were utilized as the nodes. Their proposed system considered air and soil temperature and humidity. For data transmission, the proprietary SensiNet™ protocol was utilized over ZigBee. Results showed a relation between temperature and humidity and determined that both of them are non-linear. Another WSN for smart agriculture that utilizes ZigBee is proposed in [8]. Authors addressed the necessity of precision agriculture systems so as to reduce water consumption. The monitored variables were soil temperature and moisture. Several modules with ZigBee radios such as TelosB, ATmega256RFR2 and OpenMote-CC2538 were considered. The data was stored each 15 minutes on a SD card for further analysis. Due to some data loss, the resulting efficiency of the system was 94.79%.

Several architectures have also been designed for precision agriculture. J. A. López-Riquelme et al. presented in [9] a FIWARE cloud-based software architecture for precision agriculture. Authors discussed the advantages of FIWARE and its open-source modules and enablers for agriculture. The hardware architecture was performed utilizing the MEWiN Mainboard with a SIM900 Module. Experiments were performed both on the laboratory and on real crops. The average power consumption of the node was 4.29 mA. Another architecture for precision monitoring intended for potato crops and based on Cyber-Physical Systems (CPS) is presented by Ciprian-Radu RAD et al. in [10]. Their proposed architecture was divided in four layers being application, analyzing, networking and sensing. The data obtained from the system has then to be processed so as to perform the decision making and control the system on real-time. The aims of the proposed architecture were the optimization of water and chemical usage, the increase in production and sustainability and the reduction of costs. H. Navarro-Hellín et al. presented in [11] an

architecture for WSN intended for water management. The system utilized ES-2 sensors for water temperature and conductivity, GS3 and HP II sensors for moisture, temperature and conductivity, 10HS sensors for soil moisture, LMK sensors for pressure and MPS-2 sensors for temperature and soil matric potential. The communication was performed over GSM/GPRS and the data was stored both in a database and in SD cards. A web application allowed monitoring the data remotely.

Furthermore, implementations of monitoring systems employing IoT protocols such as CoAP are starting to be performed. Adarsh B. U. et al. performed in [12] an implementation of a water quality monitoring system utilizing CoAP over 6LoWPAN. The Contiki OS cooja simulation environment was utilized to design the system. The monitored parameters were conductivity, pH, temperature and ORP (Oxidation Reduction Potential). A TelosB mote was utilized so as to gather the data from the sensors and transmit it to the database. Actuators were utilized so as to add acid or basic solutions to the water according to the sensed data. The data could be accessed through a web interface. An environmental monitoring system that utilized CoAP as the communication protocol was developed by Tomislav Dimčić et al. in [13]. Authors utilized a Telit GSM-GPRS modem with a Python engine and implemented CoAP on the application layer of the M2M (Machine-to-Machine) system. Comparisons were performed with CoAP and HTTP where CoAP obtained a considerably lower transmission time. Lastly, a comparison of application protocols for IoT systems was performed by Nitin Naik in [14]. CoAP, MQTT, HTTP and AMQP were compared according to message size and overhead, power consumption and resource requirements, bandwidth and latency, contrasting reliability vs. interoperability, security vs. provisioning and IoT usage vs. standardization. CoAP presented the lowest size, power consumption and bandwidth usage followed by MQTT, AMQP and HTTP. However, MQTT was the most reliable and HTTP and CoAP were the ones that presented more interoperability. MQTT was the most secure protocol followed by CoAP and the most utilized for IoT systems. However, it is also the least standardized protocol, being HTTP and CoAP the most standardized ones.

Our proposed architecture for smart agriculture and irrigation considers the usage of CoAP over WiFi due to its reduced overhead and the widespread utilization and availability of WiFi devices.

3. Architecture Overview

The proposed system uses the information collected by the sensors of a WSN. The data from the sensor network will be sent through a Wireless Mesh Network (WMN) formed by Access Points (AP) located in environments where irrigation canals exist in urban and rural areas. In addition, in a centralized location, we will install the network devices that allow the Wide Area Network (WAN) connection to send the collected dataset to an external location, where it will be stored and processed. Also, in that location, we will have equipment to automatically respond to alarms that have been previously detected and where the action that must be carried out in response is known. The communication technology we use is that of device-to-device (D2D) and device-to-machine (D2M).

We define a model that can be used by different technologies and work environments. Our system includes features such as the following ones:

- Independent of communication technology.
- Supports different manufacturers.
- Adaptable to different physical environments.
- Low power consumption. We avoid the need of the nodes of an external power supply.
- Low bandwidth consumption.
- Scalable, supporting the aggregation of new nodes.

The WSN nodes collect information from the sensors connected to each one of them. The collected data will be stored in a SD card that will be included in the node hardware. In addition, the data will be sent periodically, through a WAN connection, to the place where they will be stored and processed, using Artificial Intelligence (AI). The AI uses the processed information to know what is happening and to use it in the decision algorithms. The actions that must be taken, as a result of the application of AI, are sent back to the WSN so that they receive nodes with actuators and carry out the appropriate actions.

Previously, we created a corpus to implement the AI. We define the different parameters and thresholds that allow an optimal functioning of the system. Depending on the type of crop and the types of area, we generate different corpus for different crops. For example, a corpus is defined for growing citrus, another for rice, another for vegetables, etc.

Once the nodes have been identified and the different sensors and actuators labeled, information will begin to be collected. After a training phase, a casuistic series is obtained. Using a decision algorithm and rules that include all the possible cases, we determine the actions that the actuators must carry out. The system will be able to predict actions before receiving specific alarms from the nodes indicating an anomalous situation. Figure 1 shows the proposed architecture.

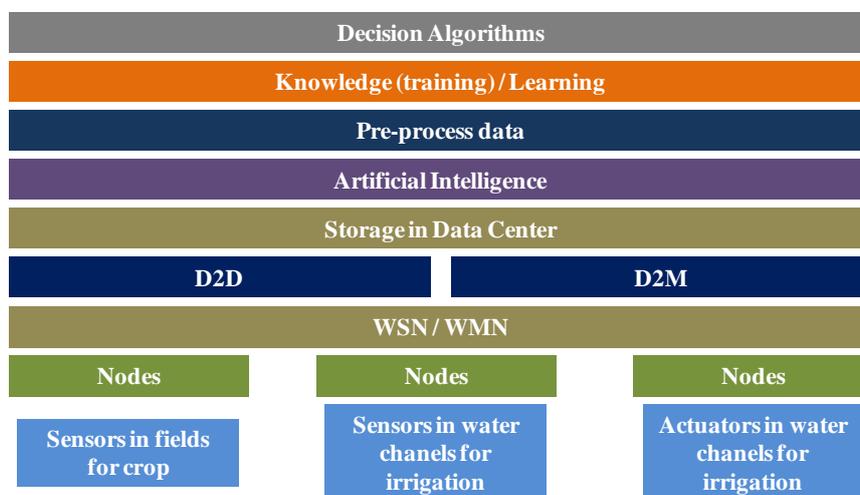


Fig. 1. Proposed Architecture.

In our architecture, each sensor node gathers measurements from each one of the sensors that it has connected. All these data will be used later in the decision algorithms. The sensor nodes are installed in two areas, mainly in the water canals for irrigation and in the crop fields. We have grouped the sensors located in the water canals for irrigation. Each group is formed by three sensor nodes for water control. This is done so as to avoid false positive problems. If a variation of the parameters is not detected in more than one node of a group, it is considered to be an error in the measurement. This redundant implementation allows us to provide high reliability for the taken measures.

We have studied the parameters that can influence our system. Among them, we can highlight humidity, temperature, turbidity, conductivity, etc. We have compiled information regarding the relationships that must be established between these parameters for the different situations that can be presented in order to adapt them to the appropriate solution in each case.

When a sensor node generates an alert, due to the exceeding of a threshold value of one or more of the parameters, the measurements of all the parameters of that node are sent at that moment. The information will be sent from a single device to another device. The energy saving is big because only the minimum needed messages are sent.

Figure 2 shows the 4 principal phases of our system, which includes sensor nodes, data storing, AI and actuator nodes. Firstly, the data acquisition from all the distributed sensor nodes is performed. Periodically, the data acquired in the previous phase is stored on the remote storing servers. Then, the different characteristics of the data such as location, date, time, humidity, temperature and turbidity are extracted so as to be processed by applying the decision rules designed by experts. The processed information will be utilized to obtain models and predictions and its results will be stored increasing our database. Lastly, the corresponding actions are performed on the actuator nodes according to the obtained predictions.

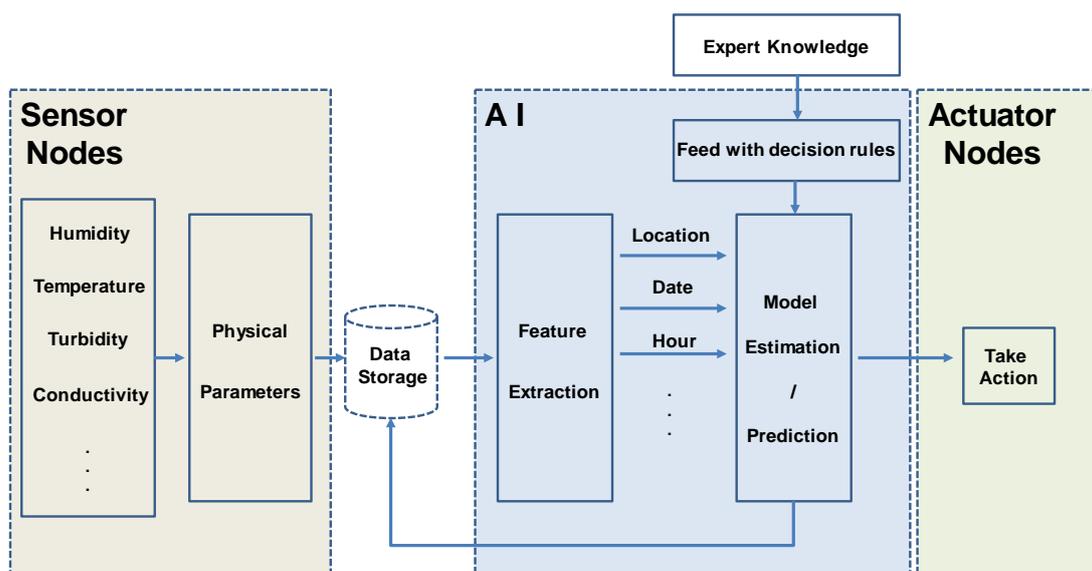


Fig. 2. Phases of our system.

Figure 3 shows the flow of the data. Initially, the data is collected by the physical sensors. The algorithms will process the data and make the appropriate decisions based on the previously created rules. Finally, the action to be carried out will be decided. It will be sent to the actuators so that it is carried out properly.

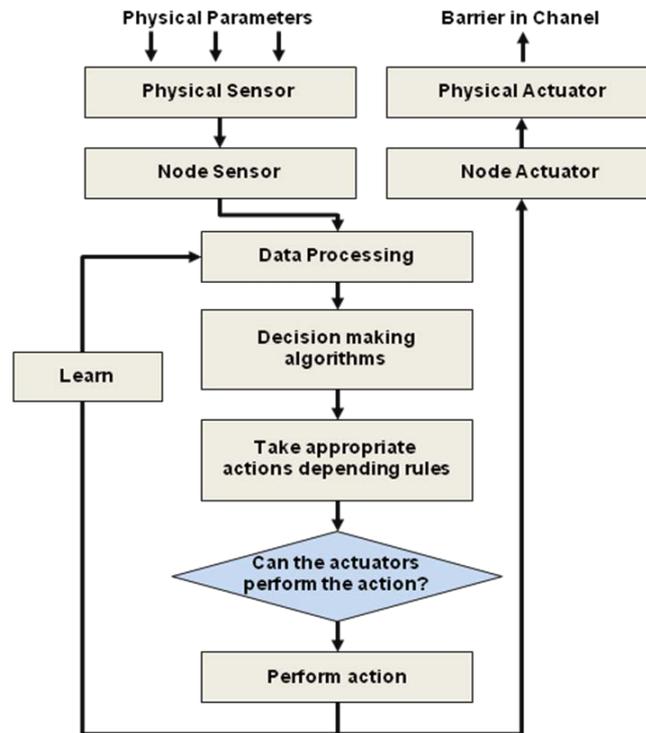


Fig. 3. Data Flow.

3.1 Wireless sensor network and deployment of wireless sensor nodes

The WSN wireless sensor network we propose is based on two types of nodes. One of the types consists of sensor nodes. The other type is formed by actuator nodes. In the sensor nodes, different physical sensors will be connected so as to allow, in some cases, to study the need for irrigation in fields so that the crop develops properly and does not get lost and, in other cases, to detect the quality of the water transported in irrigation water canals and be able to study and control its debugging. The sensor nodes of the water canals will be distributed forming groups of three nodes. We use three nodes for reliability reasons. If any of the measures estimated in a group is not validated by at least two of the group's nodes, it is not considered valid. The sensor nodes used in agriculture will be located individually in each tree that is being observed. The actuator nodes will be connected to the gates to open or close the water passage in a canal.

The nodes are strategically placed, to maximize the impact of the measures taken in the studied area. At the same time, they must be located so that they can be connected correctly through a wireless transmission.

We use IEEE 802.11 technology, although it requires a higher power consumption than other wireless technologies such as Bluetooth or ZigBee, since the nodes can be connected from a greater distance. To communicate the groups of nodes, we will use APs, which will allow us to extend the distance to which the nodes can be connected and, to extend the size of the network as well, since we will locate as many APs as necessary for the transmissions to be carried out correctly. Therefore, we will create a WMN.

In case of being in a land with hills or mountains, the AP will be located in the highest areas so that the coverage will be maximum, while using the least number of devices. Although, tentatively, most of the areas where they are going to be deployed are flat. An example of the location of an AP is shown in Figure 4, it can be seen that the AP is placed in a centralized position from which it makes a circular coverage of 100 meters. As in a plantation of orange trees, the distance between the trees is 4x6 meters, in an area of 200x200 meters that is the one represented in the figure, there are 1600 planted trees. To cover all the trees, we need to use more APs, but in this way, we can theoretically connect with 1250 nodes.

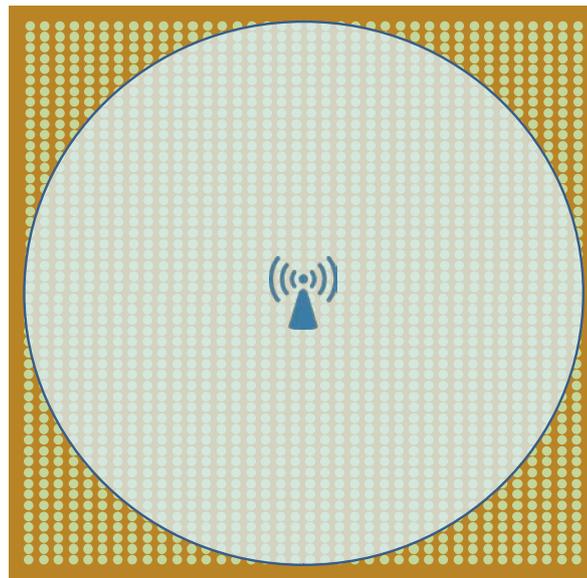


Fig. 4. AP coverage in field of oranges of 200 x 200 meters.

Due to the increased power consumption of IEEE 802.11 technology, we take into account the published recommendations for WLAN in [15].

All groups of sensor nodes will connect wirelessly with a group of sensors located in an urban area, which we will call the "Urban area 0" group. To achieve this purpose, we will implement APs with DD-WRT firmware. Thanks to the firmware, we are able to activate a Wireless Distribution System (WDS) that, working in repeater mode, will allow the APs to communicate with other APs (forming the WMN) and with nodes simultaneously.

The group "Urban area 0", has as a distinguishing feature from the rest of the nodes that, being in an urban area, it can receive power supply and establish a permanent wired or wireless WAN connection. The rest of the groups of nodes will send the measures to this

special group, so that they are forwarded to a remote location where our data storage systems and the equipment necessary to perform their treatment through AI will be located.

Once the AI system makes a decision regarding the action that must be carried out, it will store the adopted measure and send it at the same time to the "Urban area 0" group using the WAN connection. Together with this group of nodes and the network devices that allow us to connect to the WAN network, a team will be implemented that will allow us to store all the actions that the AI has already dealt with, with the appropriate response. So, when an alarm is generated in any group of sensors, when exceeding the thresholds allowed in any of the parameters studied, if the same event was previously produced, the associated action will be carried out immediately employing fog computing so as to avoid unnecessary latencies. Lastly, from the central node of the "Urban area 0" group, the order of the action will be sent to the affected node or actuating nodes, so that they carry out the recommended action.

Figure 5 shows an example of areas where we locate the different nodes and equipment.

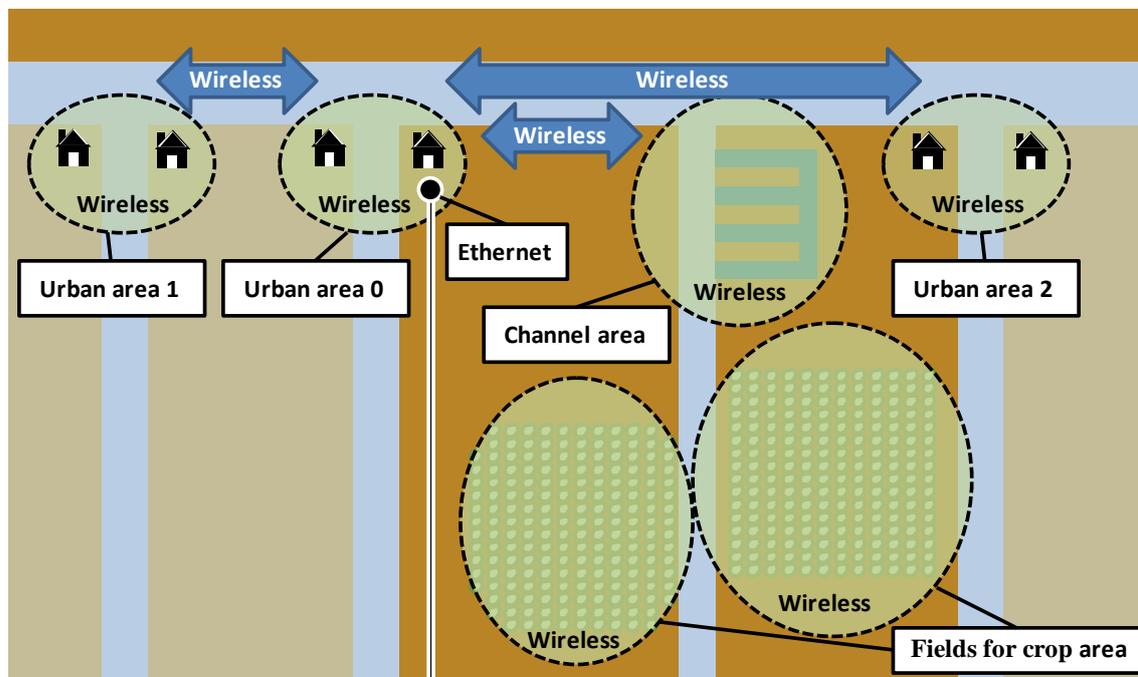


Fig. 5. Areas where different nodes and equipment are located.

3.2 Wireless Sensor Node

To design the sensor nodes, we must consider several factors. The sensor node must connect different types of sensors to measure the required parameters. It must have a storage system that in our case will be an SD card. It must have the necessary hardware to transmit the measurements in the IEEE 802.11 technology. Moreover, for its Operation, it must have a self-feeding system for power generation. The energy system consists of rechargeable batteries and small solar panels which are responsible for recharging the batteries. In this way, we ensure a long service life of the devices in operation. It will also have a USB port so that administrators can connect directly to the node.

In the same way, to carry out the design of the actuator nodes, we must also consider several factors. The actuator node must have a linear actuator, this allows us to operate a hydraulic or pneumatic actuator that will move the gates to open or close the canals, and like the previously described sensor nodes a storage system (SD card) will be needed as well. The hardware necessary to receive and transmit the actions using IEEE 802.11 technology must be arranged. Furthermore, for its operation, it must have a self-powered system for power generation. It will also have a USB port so that administrators can connect directly to the node.

In order to consume the minimum amount of energy during the transmission of the data in our sensor network, we will use the Constrained Application Protocol (COAP) [16]. The protocol was developed as an Internet Standards Document, described in RFC 7252 [17]. It is a protocol similar to HTTP, but it is specialized for restricted use in nodes and networks for IoT. It is a protocol designed for machine-to-machine (M2M) applications such as smart energy and building automation.

3.3 Association problem of the nodes to the AP

The problem that arises when trying to connect a large number of nodes to the "Urban area 0" is the distance at which it can be found. To solve this, we use a WMN, as shown in Figure 6. Here, we can see the nodes that will connect as clients to the APs. The whole set of APs will form the WMN that in the "Urban area 0" will be connected with cable to a switch that will allow reaching a server. The server is responsible for managing the number of users that are associated with each access point. When the nodes associated with an AP reach a threshold, the server ensures that the nodes that exceed this threshold are associated with another AP.

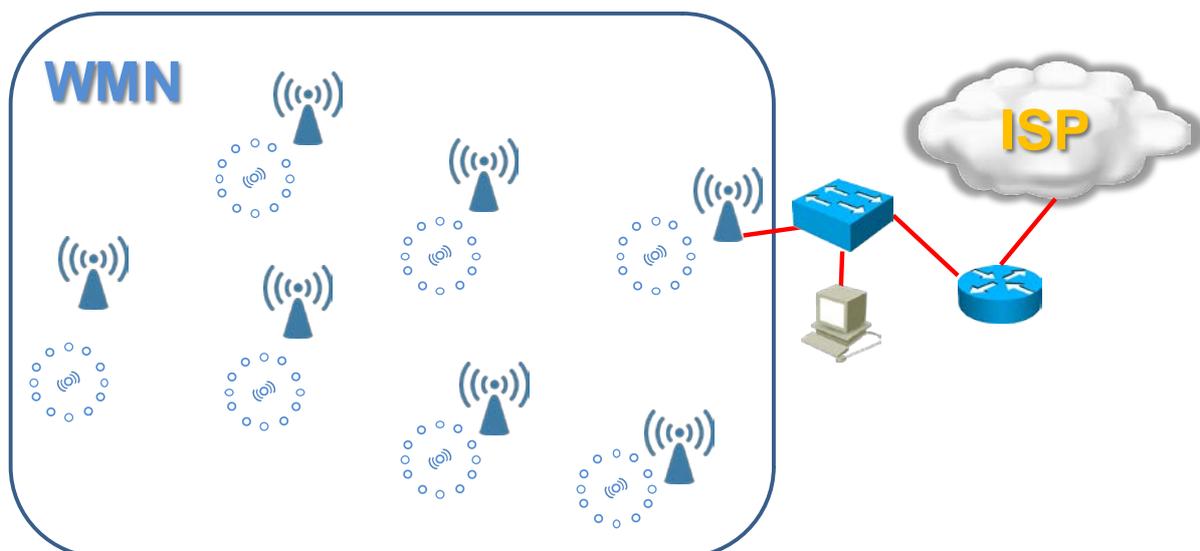


Fig. 6. Network architecture of our nodes-balanced system.

One of the main obstacles, historically speaking, at the time of using wireless technology if we compared it with the wired, was the shortage of bandwidth. Since the appearance of this technology, the transmission speed has increased remarkably. The initial 1Mbps of the IEEE

802.11 standard has passed to the theoretical 7 Gbps that can reach the IEEE 802.11ac standard, including the future standard IEEE 802.11ax (its implantation is foreseen in 2019) that will provide speeds of up to 11 Gbps.

With these transmission speeds, the problem of bandwidth is beginning to disappear in certain environments. For example, in practice, if we use the IEEE 802.11ac standard, we can transfer 900 Megabytes per second. It is achieved by using eight 256 QAM channels of 160 MHz each. These channels can transfer a maximum of 870 Mbps, although it is usual to use no more than 2 or 3 of the eight channels.

Our network is not a usual network in which we find very varied types of traffic and with different requirements, such as HTTP, multimedia streaming, SMTP, POP3, FTP, VoIP, etc., which may require large bandwidths. Although the bandwidth of an AP is shared for the transmission of different groups of nodes, because the bandwidth required by CoAP for the transmission of the information of the IoT devices is very low, we will not find limitation from the point of view of bandwidth. Actually, our frames will have a maximum size of 1077 bytes, which is lower than the maximum size of a CoAP packet (1125 Bytes). Figure 7 shows an IPv4 CoAP frame, with the different fields that form it and the size of each of them. In our case, the token has a length of 1 Byte and the Observe option is utilized.



Fig. 7. Frame COAP.

If we use APs that meets the IEEE 802.11n standard, using the 20 MHz channel, we can theoretically transmit at 72.2 Mbps, but the real transmission rate is approximately 50 Mbps. Since the maximum size of a COAP datagram is 8616 bits, at a speed of 50 Mbps the transmission is performed in 0.000172 seconds. Theoretically we can get to transmit approximately 5800 datagrams in a second.

The main challenge of our proposal, when associating groups of nodes with APs, is to know the maximum number of simultaneous connections that APs can establish, together with a design that covers all the observed space with the minimum amount of AP needed. Generally, the limitation is given by the manufacturer. For example, if we use the Linksys WRT600N v1.1 [18] APs that work in the 2.4 and 5 GHz bands with the a / b / g / n standards, it supports a theoretical maximum of 253 users that are assigned by DHCP, although by default only 50 are allowed.

In response to our initial proposal, for example, when the number of nodes associated with an AP exceeds 50, from the server located in "Urban area 0" through the WMN, it will be indicated to the surplus nodes that they must be associated with another AP.

3.4 COAP

CoAP is an application protocol for IoT systems that works over UDP. It is a simplified version of HTTP so it provides interoperability with it. However, CoAP has not been widely

implemented utilizing WiFi. In fact, the available libraries for WiFi chips are scarce. Particularly, for the ESP8266 chip there is the ESP-CoAP library [19] that allows implementing CoAP and provides the biggest number of functionalities compared to other available libraries. However, one of the things to consider with CoAP is that the client has to make a request to obtain the data. Therefore, clients could be requesting data when it would not be necessary, making the system forward more messages than necessary and increasing the energy consumption. For that reason, CoAP provides an “observer” option that would allow the system to forward the information regularly to the client once the client subscribes to a resource.

Although the ESP-CoAP library allowed observing one resource, we found necessary the ability to observe more than one resource. Therefore, the library for the server was modified so as to allow observing several resources as well as allowing several observers to one resource and cancelling the observe if wanted.

The notification class was modified so as to allow observing several resources as showed in Figure 8.

```

void coapServer::notification(String url, char *payload)
{
    response->version=request->version;
    response->type=COAP_ACK;

    response->messageid=messid++;

    response->code = COAP_CONTENT;
    response->payload = (uint8_t *)payload;
    response->payloadlen = strlen(payload);
    response->optionnum = 0;

    obsstate=obsstate+1;

    response->options[response->optionnum].buffer =8obsstate;
    response->options[response->optionnum].length = 1;
    response->options[response->optionnum].number = COAP_OBSERVE;
    response->optionnum++;

    char optionBuffer2[2];
    optionBuffer2[0] = ((uint16_t)COAP_TEXT_PLAIN & 0xFF00) >> 8;
    optionBuffer2[1] = ((uint16_t)COAP_TEXT_PLAIN & 0x00FF) ;
    response->options[response->optionnum].buffer = (uint8_t *)optionBuffer2;
    response->options[response->optionnum].length = 2;
    response->options[response->optionnum].number = COAP_CONTENT_FORMAT;
    response->optionnum++;

    for(uint8_t i=0;i<obscount;i++){
        for (int j = 0; j < rcount; j++){
            //send notification
            if(observer[i].observer_url==resource[j].rt && url==resource[j].rt){
                response->token=observer[i].observer_token;
                response->tokenlen=observer[i].observer_tokenlen;
                sendPacket(response, observer[i].observer_clientip, observer[i].observer_clientport);
            }
        }
    }
    if(messid==5000)
        messid=0;
}

```

Figure 8. Code for observing several resources.

ParseOption was modified as well so as to enable the possibility of cancelling observing a resource (see Figure 9).

```

if(request->options[num].number==COAP_OBSERVE){
    if(*(request->options[num].buffer)==1){
        for(uint8_t i=0;i<obscount;i++){
            if(observer[i].observer_clientip==Udp.remoteIP() && observer[i].observer_url==url){
                if(obscount==1){
                    observer[i]={0};
                    obscount=0;
                }else{
                    observer[i]=observer[i+1];
                    observer[i+1]={0};
                    obscount=obscount-1;
                }
            }
        }
    }
}

```

Fig 9. Code for cancelling the observation with a GET message.

4. Results and discussion

In this section, we present the obtained results after simulating the aforementioned scenario. For our simulation, we consider a different number of nodes that send data to a single AP. Firstly, the number of nodes that are unable to send their data because another node is sending data is calculated. Then, we calculate the number of nodes that, after applying the CSMA/CA, are still unable to communicate their data to the AP. Finally, we show the number of attempts of sending data which failed during 1 second in the network.

The number of nodes which are unable to send the data when it is expected is shown in Fig. 10. We consider that each node has a different clock time and they have to send the data every second, as long as we are considering the worst case when the orange trees are being irrigated. Due to the different clock times, each sensor sends the data in a different moment. Our purpose with this graphic is to evaluate the effect of increasing the number of nodes in the number of nodes which are unable to send the data.

First of all, the number of nodes that are unable to send their data when it is expected because the network is currently being used by another node is evaluated. As it is expected, the higher the number of nodes, the higher the nodes unable to send the data. This is because as more nodes are in the network for more time the channel is used. In blue bars, the mean value after 30 different simulations (with different clock times) is represented and the line represents the minimum and maximum simulation results. The number of nodes unable to send the data increases following a second order polynomial relation. If we consider the percentage of nodes unable to send the data, the relation is still positive and increases following a linear relation. In the first case, with only 25 nodes, our simulation indicates that 4% of the nodes are unable to send the data when it is expected. On the most extreme case, with 250 nodes, 32% of the nodes detect a message when they try to send their data

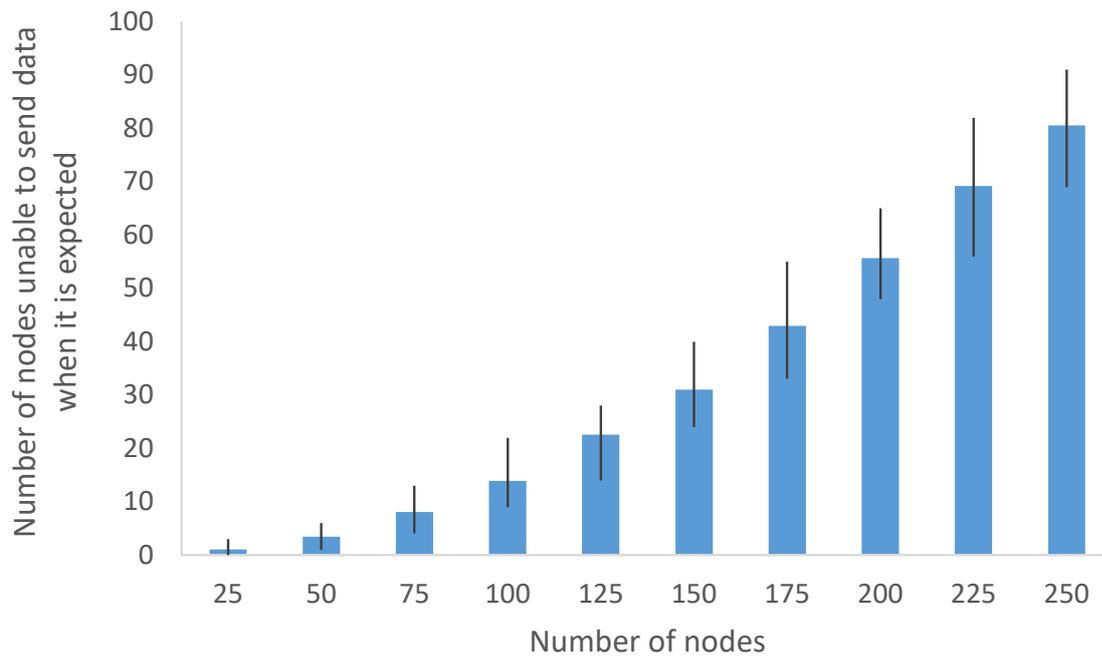


Figure 10. Number of nodes unable to send data when it is expected

Next, we present the number of nodes that are unable to send the data after the application of the CSMA/CA algorithm using an exponential Backoff algorithm to calculate the waiting time, see Fig. 11. The used Backoff algorithm gives 5 attempts before considering that the connection is unreliable. Again, we present in the graphic the mean, the minimum, and the maximum of 30 simulations. The results show that when there are less than 75 nodes (25 or 50) no node was unable to connect to the AP and send the data. However, when we connected up to 75 nodes to a single AP with the network parameters set in the previous section, some nodes were unable to connect and forward their data. As it is expected, the higher the number of nodes connected to the same AP, the higher the number of nodes unable to connect after applying the CSMA/CA. The relation between the number of nodes and the number of nodes unable to connect follows a potential relation. The number of nodes unable to send the data is up to 23 for a network with 250 nodes. If we consider the percentage of nodes instead of the number of nodes, the values are 0%, 0.04%, 0.33%, 1.07%, 1.60%, 3.03%, 4.95%, 7.60% and 9,52%.

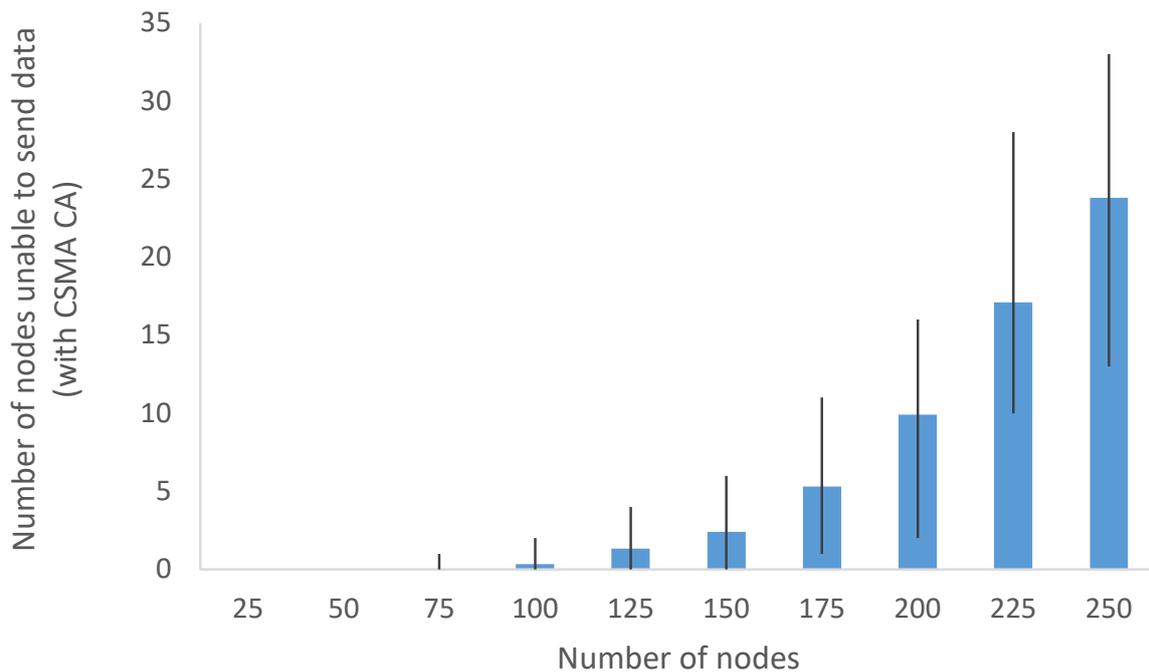


Figure 11. Number of nodes unable to send data after the CSMA/CA

Finally, we present the data of the number of failed attempts in all the network during 1 second. This value is important because it represents the number of times that a node must apply CSMA/CA and the Backoff algorithm. This process not only consumes time, but it also consumes energy. Even it is performed in the microprocessor and they are energy efficient, some energy is consumed and if this process should be repeated many times in one second (a maximum of 5 times) it will have an effect on the battery lifetime. Moreover, the time that the microprocessor remains applying the algorithm is time that cannot be utilized to perform other actions. Therefore, it is important to consider the number of failed attempts to send the data in the network, see Fig. 12. The number of attempts that fails, increases with the number of nodes in the network. With 250 nodes a total of 267 attempts failed during 1 second (mean value). It means that to send 250 packets with data 267 attempts failed and 226 attempts succeeded (250 less the number of nodes unable to send the data, see Fig. 11). For the best case, with the lowest number of nodes, the mean number of failed attempts is almost 2, the maximum and minimum are 6 and 0.

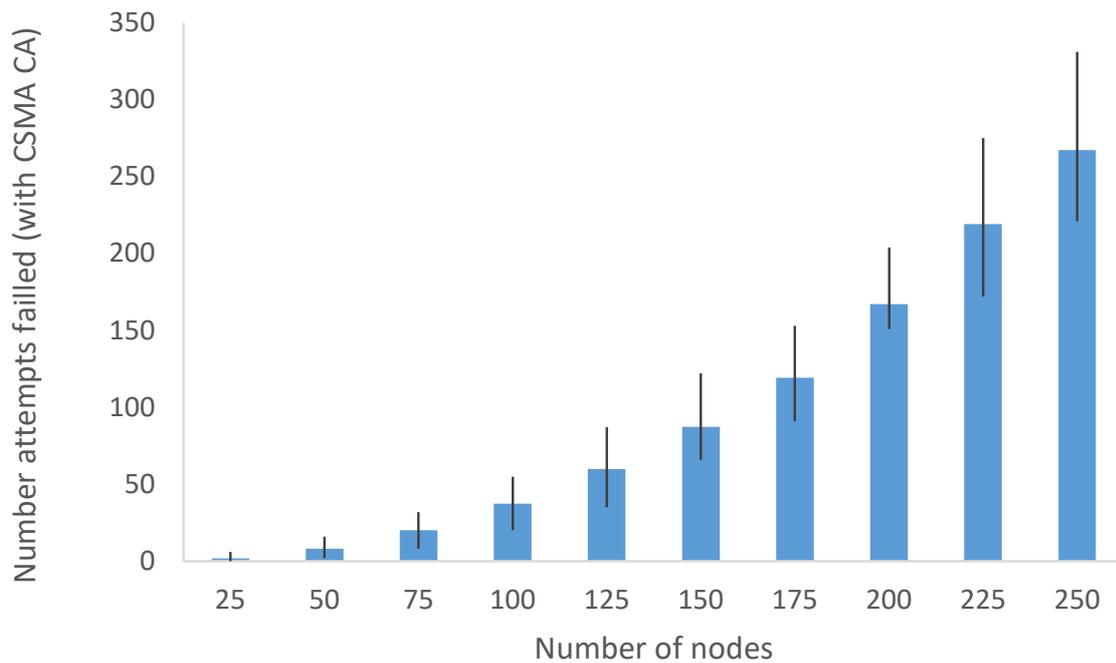


Figure 12. Number of failed attempts of sending data to avoid collisions

Observing the obtained results, we propose the connection of a maximum of 50 nodes to each one of the AP, so that no collisions appear, and the operation of the network is optimal.

6. Conclusion and Future Work

We propose an architecture design for WSN networks to monitor the crops and its irrigation system. The proposed architecture is divided into multiple layers. From the bottom where we have located the different sensors and actuators up to the upper layer where we obtain the result of the application of AI to the measurements observed in the field through the sensors. As the observed measurements must be sent over long distances, we propose the superposition of a WMN over the WSN. In this way, remote locations can be reached with wireless technology for the storage and processing of data through other systems. In addition, given the results obtained through simulations, we propose an association of a maximum of 50 nodes to each AP so as to avoid collisions, making our network work optimally.

In the future we hope to implement the proposed networks with real devices and see if our proposal can be improved, either due to the type of sensors, actuators and nodes used, or to the AP that can be used.

Acknowledgement

This work has been partially supported by the "Ministerio de Economía y Competitividad" in the "Programa Estatal de Fomento de la Investigación Científica y

Técnica de Excelencia, Subprograma Estatal de Generación de Conocimiento" within the project under Grant TIN2017-84802-C2-1-P. This work has also been partially supported by European Union through the ERANETMED (Euromediterranean Cooperation through ERANET joint activities and beyond) project ERANETMED3-227 SMARTWATIR.

References

- [1] Parra L., Rocher J., García L., Lloret J., Tomás J., Romero O., Rodilla M., Falco S., Sebastián M. T., Mengual J., González J. A., Roig B. Design of a WSN for smart irrigation in citrus plots with fault-tolerance and energy-saving algorithms. *Network Protocols and Algorithms* 10 (2), 95-11. 2018. <https://doi.org/10.5296/npa.v10i2.13205>
- [2] Pedrero F., Kalavrouziotis I., Alarcón J. J., Koukoulakis P., and Asano T., "Use of treated municipal wastewater in irrigated agriculture-Review of some practices in Spain and Greece", *Agricultural Water Management*, Vol. 97, Issue. 9, pp. 1233–1241, September 2010. DOI: <https://doi.org/10.1016/j.agwat.2010.03.003>
- [3] Parra L., Rocher J., Escrivá J., Lloret J., "Design and development of low cost smart turbidity sensor for water quality monitoring in fish farms", *Aquacultural Engineering*, Vol. 81, Issue February, pp. 10–18, May 2018. <https://doi.org/10.1016/j.aquaeng.2018.01.004>
- [4] Parra L., Ortuño V., Sendra S., Lloret J., "Low-cost conductivity sensor based on two coils", In *Proceedings of the First International Conference on Computational Science and Engineering (CSE 2013)*, Valencia, Spain, August 25-31, 2013.
- [5] Kim D. S., Chung B. J., Son S. Y. Implementation of a Low-Cost Energy and Environment Monitoring System Based on a Hybrid Wireless Sensor Network. *Journal of Sensors* 2017, 1-11. <https://doi.org/10.1155/2017/5957082>
- [6] RFC 7641: Observing Resources in the Constrained Application Protocol (CoAP), September 2015. Available online: <https://tools.ietf.org/html/rfc7641>
- [7] Mancuso M., Bustaffa F. A wireless Sensor Network For Monitoring Environmental Variables in a Tomato Greenhouse. *IEEE International Workshop on Factory Communication Systems*, Torino, Italy, 28-30 June 2006
- [8] Vinicius K., Esgalha H., José S., Prado T. Wireless Sensor Network for Smart Agriculture using ZigBee Protocol. *IEEE First Summer School on Smart Cities*, Natal, Brazil, 6-11 August 2017, 61-66. <https://doi.org/10.1109/S3C.2017.8501379>
- [9] López-Riquelme J. A., Pavón-Pulido N., Navarro-Hellín H., Soto-Valles F., Torres-Sánchez R. A software architecture base don FIWARE cloud for Precision Agriculture. *Agricultural Water Management* (183), 123-135, 2017. <https://doi.org/10.1016/j.agwat.2016.10.020>
- [10] Rad C., Hancu O. Takacs I. Oltenau G. Smart Monitoring of Potato Crop: A Cyber-Physical System Architecture Model in the Field of Precision Agriculture. *Agriculture*

and Agricultural Science Procedia 6 (2015), 73-79.
<https://doi.org/10.1016/j.aaspro.2015.08.041>

[11] Navarro-Hellín H., Torres-Sánchez R., Soto-Valles F., Albadalejo-Pérez C., López-Riquelme J. A., Domingo-Miguel R. A Wireless sensors architecture for efficient irrigation water management. *Agricultural Water Management* 151 (2015), 64-74.
<https://doi.org/10.1016/j.agwat.2014.10.022>

[12] Adarsh. B. U., Divya Darshini B., Shobha K. R. Natrajan K., Paventhan A., Allu S. K. Pahuja N. Design of 6LoWPAN enabled Real Time Water Quality Monitoring System using CoAP. *Proceedings of the Asia-Pacific Advanced Network 2014*, 38, 42-54.
<https://doi.org/10.7125/APAN.38.7>

[13] Dimčić T., Krčo S., Gligorić N. CoAP (Constrained Application Protocol) implementation in M2M Environmental Monitoring System. *E-society Journal: research and applications* 3 (1), 21-33, 2012.

[14] Naik N. Choice of Effective Messaging Protocols for IoT Systems: MQTT, CoAP, AMQP and HTTP. *IEEE International Systems Engineering Symposium, Vienna, Austria, 11-13 October 2017*, 1-7. <https://doi.org/10.1109/SysEng.2017.8088251>

[15] Lloret, J.; Sendra, S.; Coll, H.; Garcia, M. Saving energy in wireless local area sensor networks. *Comput. J.* 2009, 53, 1658-1673. <https://doi.org/10.1093/comjnl/bxp112>

[16] COAP. Available at: <http://coap.technology/> (Last accessed on 21/11/2018).

[17] The Constrained Application Protocol (CoAP). Available at: <https://tools.ietf.org/html/rfc7252> (Last accessed on 21/11/2018).

[18] Dual-Band Wireless-N Gigabit Router with Storage Link SKU WRT600N. Available at: http://downloads.linksys.com/downloads/userguide/1224638362365/WRT600N_ug.pdf (Last accessed on 21/11/2018).

[19] ESP-CoAP library. Available online: <https://github.com/automote/ESP-CoAP> (Last accessed on 21/11/2018).

Copyright Disclaimer

Copyright reserved by the author(s).

This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).