The final publication is available at

https://doi.org/10.1109/FMEC49853.2020.9144776

IoT-WLAN proximity network for Potentiostats

Pedro González[1,2], Jaime Lloret[2], Jesus Tomás[2], Oscar Rodriguez[3], Mikel Hurtado[1]

[1]Departamento de Ingeniería Electrónica, Universidad Central, Bogotá D.C, Colombia

[2]Instituto de Investigación para la Gestión Integrada de zonas Costeras, Universitat Politècnica de València, Valencia, España

[3]Departamento de Química, Facultad de Ciencias, Universidad Nacional, Bogotá D.C, Colombia

pgonzalezr1@ucentral.edu.co, jlloret@dcom.upv.es, jtomas@upv.es, orodriguez@unal.edu.co, mhurtadom1@ucentral.edu.co

Abstract— **The implementation of potentiostats as portable and communicated devices has reached significant progress to benefit research, industry, and education. The Internet of Things (IoT) is a good opportunity to interconnect devices such as the potentiostats together with electronics, communication technologies, and chemistry into a single system. This work proposes a network for potentiostats using machine-to-machine (M2M) protocols, modifying its functioning mechanism in the broker to check the payload of the message that passes through it and synchronize the sensors depending on its content. Although one sensor can be synchronized directly to another, the broker decides which sensor to pair. This modification was made in the M2M protocol algorithm, both in the Broker and in the Client (sensor). In addition to this, the network uses an interconnection architecture of IoT smart networks of proximity with centralized management. The results of the tests carried out showed that the use of a modified M2M such as the one proposed in the architecture allows synchronization and comparison of the measurements of several sensors in real-time.**

Keywords- Internet of Things, Artificial Intelligence, Potentiostats, M2M Protocol, Mobile App, IoT-WLAN, MQTT.

I. INTRODUCTION

Potentiostats are electronic devices that are used to control a three-electrode cell and to perform electroanalytical measurements (mainly electrochemical). A potentiostat works by maintaining the potential of the working electrode at a constant value concerning the reference electrode by adjusting the current at an auxiliary electrode, also, requires adjustments depending on the measurement is doing and therefore, has two working parameters, voltage (V) and current (I) [1].

However, it is also possible to include pressure (P), temperature (T), humidity (H), location (L), and other parameters that can help with the precision of the measurements. Moreover, since its functions are integrated with IoT technologies, more parameters are transmitted than usual. A potentiostat can be converted into a portable and mobile device thanks to its new communication capabilities. In the past, they used to be large devices that laboratories used for electroanalytical experiments and required large spaces. However, they have gradually has adapted to advances in electronics and computing. Nowadays, it is can

found very interesting proposals of its miniaturization. Some of these versions are connecting to smartphones through wireless technologies, such as Bluetooth and WiFi.

In other words, the evolution of these devices has been notorious, and a completely flexible and programmable IoT architecture could increase the possibilities of achieving many more applications. The following work shows a network proposal with a mobile application (App) based on this architecture, with potentiostats as sensors over wireless modules. The general idea is to measure a substance through the sensor and create its footprint (mainly parameters V and I), and then transmit the information to the Broker and visualize it in the App. The sensor implementation can be made using low-cost boards of Arduino [2] and Adafruit [3] since they allow fast and easy hardware development. Nevertheless, there is a large number of open-source boards available for people to develop their software. The objective is to build a prototype node and focus the work on the communication network. This way, each sensor can build the footprint (voltammograms) of a substance and share it through the network throughout the broker.

The rest of the paper is structured as follows. Related works to potentiostat devices, such as small connected sensors, are explained in Section 2. In Section 3, the problem and the proposal, complemented by two sections: the architecture and the mobile application, are indicated. Section 4 presents the performance of the tests and results in the discussion. Finally, Section 5 shows the conclusions and future work.

II. RELATED WORK

Currently, the potentiostat has landed be a small and portable device, but its ability to communicate is still relatively new. The following works show the integration of communications technologies with electroanalytical systems.

A. Ainla et al. [4], present the design and characterization of an open-source Universal Wireless Electrochemical Detector (UWED). This device can be connected to mobile devices (smartphones or tablets), via a Bluetooth Low Energy (BLE) connection. The authors developed an App to receive the experimental parameters and visualize the result in real-time. They also added a proxy to store, process, and transmit data and experimental protocols. This design simplifies and reduces the size of the device. The software is

flexible and allows adaptation to different types of analyses. The parameters (V) and (I) performing the tests are similar to those used by commercial desktop devices and sufficient to perform electrochemical analyses in aqueous solutions. In summary, the UWED is presented as a small low-cost wearable and wireless open-source device.

T. Dobbelaere et al. [5], showing the design of a low-cost potentiostat connected by USB. This device measures an applied (V) of ±8V, and (I) from nanoamps up to ±25mA. It has a 20-bit Digital to Analog (D/A) and a 22-bit analog to digital (A/D) converters. The authors commented that the precision of the converters (the ranges of (V) and (I)) and the dual control modes make it especially suitable for battery characterization. For this reason, they presented a use case on a lithium-ion test battery using thin-film anatase (TiO2) as the working electrode. The software was developed with open source tools and implemented in open hardware. The user interface was developed in Python language and everything is freely available for the public. The authors provide all the necessary materials, such as the schematics, PCB design, microcontroller firmware, and software, so that the device can be modified and customized.

In [6] an ABE-Stat is presented as an open-source project about a battery-powered potentiostat device with wireless WiFi connection. This device is capable of performing electrochemical analyses such as Cyclic Voltammetry (CV), Differential Pulse Voltammetry (DPV) and high impedance potentiometric measurements. Its interface was developed on Android and allows its connection to the internet through Wi-Fi. The authors emphasize that it is the first potentiostat capable of Evaluating Electrochemical Impedance Spectroscopy (EIS). It handles a user-selectable wide frequency spectrum (0.1 Hz to 100 kHz). In this work, the authors also provide extensive information about its development including hardware design and source code.

V. Bianchi et al. [7], present a low-cost portable system with a Wi-Fi connection. The device was developed on a board that integrates a microcontroller and a Wi-Fi network processor. The wireless module allows the transmission of protocols directly to a cloud service and so that data can be shared with multiple users. It also stores the data in case the connection is lost and then retransmits it. Since the device was designed to work with a battery to improve portability, a study was conducted on the battery's life. Finally, the authors compared the data acquired with an electrochemical desktop device (Autolab PGSTAT-204) with the results of the device developed to validate it. To do this, ferri/ferrocyanide was selected as a redox test, obtaining the calibration curves for both devices. The results are feasible and reproducible.

In [8], a potentiostat for electrochemical monitoring of 64 channels is presented. The most modern small potentiostat has few parallel channels with limited flexibility due to their architecture. This device is not only new but innovative because it uses a Howland current pump and a digital potential controller. It was manufactured with a matrix of CMOS potentiostats of 64 channels. Each channel has a current range of 120dB with an accuracy of 1.1pA and 100kHz of bandwidth. The applications of this potentiostat

as a monitoring device include industry, microbial electrolysis, corrosion monitoring, health, and biosensors.

J. Dieffenderfer et al. [9], present a wearable potentiostat device with a wireless connection to smartphones. This device works as an electrochemical sensor for the analysis of biomarkers within sweat. For this reason, the authors depicted it as a non-invasive device that can deliver valuable physiological information. This work presents an electronic interface that can be used as a biosensor attached to the skin. In this case, the system is calibrated using cyclic voltammetry and measures lactate concentrations in sweat.

III. PROPOSAL DESCRIPTION

A. Problem

Potentiostats are devices used to make an electrochemical analysis of different substances and determine its composition. Nevertheless, it requires continuous monitoring and data collection to control the reference voltage and observe the changes. Some of the current systems store the different measurements represented in voltammograms that have already been characterized. However, these results cannot be compared in real-time with another device measuring along at the same time. Although the potentiostats are smaller and more portable, they have appeared recently and are not yet networked. Nevertheless, potentiostats are increasingly been used as biosensors, though it is still not possible to have several measurements at the same time in different locations. Therefore, it is not possible to correlate the distance between them and the external factors that may affect the sensor measurements.

Another problem is that most sensors only send information of its measurements, although the potentiostat requires sending and receiving information to adjust its operation. That is, some potentiostats developments operate in client-server configuration but not in a network, which makes automatic synchronization difficult between several clients. Moreover, these configurations do not offer long-distance communications with the device that processes the information.

B. Proposal

Taking advantage of IoT technologies and protocols, we propose a wireless local area network for potentiostats (IoT-WLAN) that will be monitored through an App. This potentiostat network design follows a proposed architecture of the Interconnection of an IoT smart network of proximity with centralized management.

To explain this proposal, we will start presenting the mechanics of the interactions between the defined nodes through Figure 1 and explain the architecture on which the network was designed. Finally, we will present a mobile application. The network measures different substances of an area, identifies them, and reports them to the broker. Whenever the system needs to know whether the same substance is present in all the areas or its distribution; the sensor takes data more precisely and synchronizes it one by one with the other sensors.
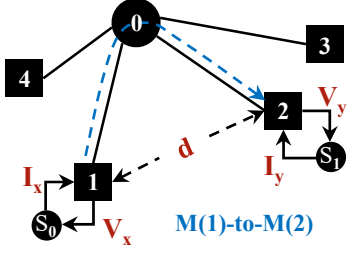
Figure 1.   M2M protocol over Potentiostats Network

The Broker M(0) can be an IoT gateway or a mobile device, depending on the topology and network configuration. Nodes M(1, 2, 3, 4) of Figure 1 represent wireless transmission modules connected to a sensor (Sn). In Figure 1, the sensors S (0, 1, 2, ..., n) are potentiostats (e.g., S0 and S1 concerning its current (I) and voltage (V) parameters). In this example, M(1) measures the current of S0, because these values are a function of the voltage V(t). This voltage is a reference value that controls the measurements through a V(t) ramp function signal. That is, M(1) injects V(t) into S0 to obtain the values of (I).

Through an M2M protocol, a sensor can establish a relation with another sensor to synchronize the same measurement on both sensors. If M(1) obtains measurements in expected or minimum acceptable ranges with Vx, then it is possible to change the values of other sensors making Vx = Vy. This way, if Ix $\neq$ Iy we can conclude that some additional element is interfering or altering the measurement. Therefore, Sn{I, V} will be the set of measurements of node M. However, if more parameters are obtained from Sn, then Sn{I, V, P, T, H, L}.

Let S_i and S_j be the set of measurements of two nodes so that S_i and S_j are the measurements of all possible Sn sensor that meets the condition of the precision factor (f_p).

Then (r) in equation 1 represents the set of all M2M relations established by the Broker in an ordered pair (S_i, S_j).

$$r(S_i, S_j) \; \forall_{i,j} \mid i,j \geq 0 \land i,j < n \land f_p(S_n) \geq 90\% \quad (1)$$

Let M be a set of sensor-nodes and R the set of all relations established by Broker M(0). The sensor-node is the union of two disjoint sets between the node in the Things layer and the sensor in the Sensors layer.

$$M(n) = M(n) + Sn \, \{parameters\} \quad (2)$$

Then the relation between M(i)-to-M(j) will be shown as in equation 3.

$$R(M(i), \; M(j)) \quad (3)$$

This proposal is an application of a network model based on workgroup theory [10]. In this case, each Sn is a directed graph that maintains a reference to its parent node M(n), according to the graph theory [11] and data structure [12], merging both architecture layers into one.

C. Architecture with centralized management

This article presents the proposed architecture in [13] and [14], where the IoT network is organized in five layers called: Internet, Management, Assistants, Things, and Sensors in an ascending hierarchy and tree structure. The network model of this architecture allows grouping things based on M2M protocols, at the level of the management layer and by groups of parameters in the Internet layer.

Moreover, this architecture is flexible and was designed to allow interaction with artificial intelligence (AI) algorithms. The broker device is the node of the management layer that hosts the AI and manages Internet access.

In this network design, the client nodes M(n) are the transmitting devices in the Things layer of the architecture, while the broker node M(0) is in the Management layer. The potentiostats (Sn) are located in the Sensors layer, and its parent node, the wireless modules M(n) are in the Things layer. In any network configuration, any device can assume the role of the broker (a smartphone or a PC or an IoT gateway). It is important to state that the connection is performed in a star topology and routed through the device of layer 3. Therefore, the ideal way to fulfill this condition is to do it on an IoT gateway device located on layer 3 of this architecture.

In this case, the supervision and control of the network are done locally. However, it is possible to do it from the Internet by adding an IoT platform to the Internet layer.

D. Mobile Application (App)

This application was developed in Android Studio so that it can be installed on an android mobile device (smartphone or tablet). The app has two functions. Its main function is to monitor the footprints based on voltammograms sent from the sensors to the broker and visualize them. Its second function is to act as a backup whenever the broker in layer 3, running in the background is not available. Although the M2M protocol passes through the broker with the collected data from a client node to another client node, the algorithm checks the payload every time it passes through. With this information, the modified algorithm of the broker intervenes and decides the most appropriate network sensor. Once selected, it is matched with the source sensor and synchronized with the same parameters (Vx=Vy and Ix=Iy).

The flowchart in Figure 2 shows how the sensors are synchronized to assess whether the measurements are the same or not. Each time a sensor finds a measurement that corresponds to the previous characterization of some substances or materials, it sends the information of its Voltage parameters to the entire network. The broker compares the voltammograms of the measurements of each sensor and evaluates its precision percentage (fp). If any of these percentages is fp $\geq$ 90%, it saves the location of the sensor and evaluates other variables. This way, the algorithm can establish the distance (d) between the nodes and create a map, which could be a contamination map if that was the case.

The algorithms developed for the Broker App or IoT Gateway (Broker) [10], are based on equations 1, 2, and 3. The algorithm comparing the previously stored typical

voltammograms with those obtained from the potentiostats has been developed with equation 1. This comparator is shown in the flowchart and evaluates the factor of precision of each sensor-node as fp(Sn). The algorithm responsible for collecting the parameters sent from each sensor-node was based on equation 2. This algorithm reviews the payloads of the protocol and identifies its source address. The Broker's pairing algorithm based on Equation 3 establishes machine-to-machine relations for sensor pairs to synchronize. Once the acceptable fp values of each sensor are established, this algorithm performs synchronization by sending the parameters (V and I) to the selected destination sensor.

The Broker App has a user interface from which the different voltammograms are been, monitor. Each voltammogram is built with (V) and (I) parameters using a cyclic voltammetry technique [15]. That is, the node applies a ramp voltage signal V(t) between two voltage values to the sensor and reads its current (I) answer. Therefore, the voltammogram will be the footprint of the substance or material that has been analyzed. The sensor-node sends this information through the network on an M2M protocol for the App Broker to process it.
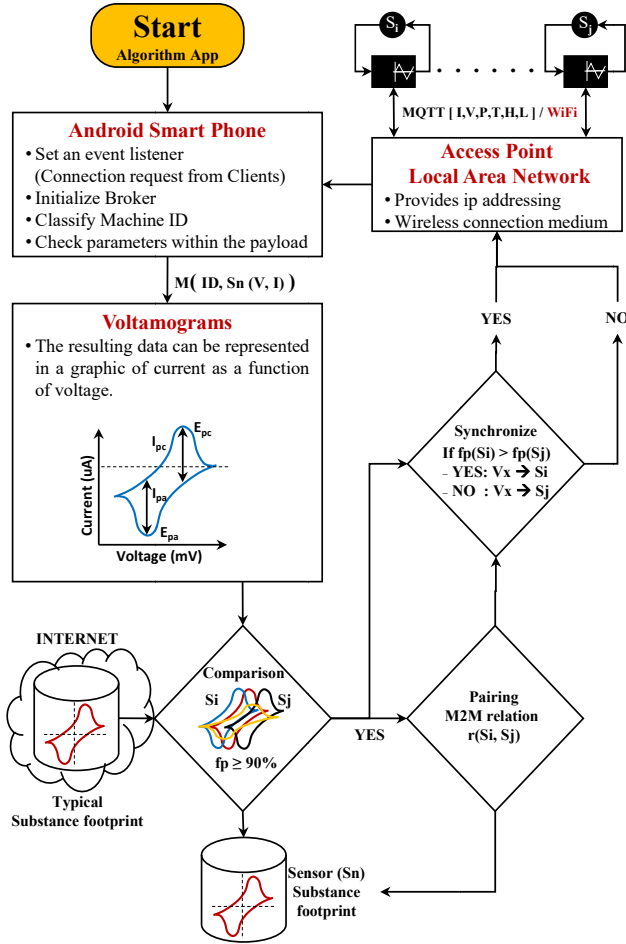

Figure 2. Flowchart of the App

The typical substance footprint (TSF) is stored in the cloud and the Broker App or IoT-Gateway, and the query via

the Internet whenever it needs to compare it with the one obtained by a sensor. The footprint is also temporarily stored in the internal memory of the wireless module (node).

IV. PERFORMANCE TEST

In this section, the tests of the proposed system are explained. These tests seek to demonstrate that the broker algorithm can create new M2M relations so that the sensors synchronize with the most precise measurement given by another sensor. Following the relational condition explained in Figure 1 and the flowchart of Figure 2, an IoT-WLAN network was implemented. This network has three parts. The first part is the set of hardware devices used on the implementation. The second one is the design of the wireless network on Wi-Fi technology in a star topology and with MQTT protocol. The third one is the resulting analysis of the operating network.

Although the architecture supports any M2M protocol to perform these tests, MQTT was selected because of its adaptability and extensive documentation.

A. Implementation of the devices

The development of this proposal involves the use of ESP8266-01 board [2] with a Programmable System-on-Chip (PSoC), integrated with a microcontroller and a Wi-Fi network module. The transmitter module of the Things layer sends the parameters (V) and (I) from the sensor to the broker. These parameters are obtained from the LMP91000 potentiostat [16] and its 3-Lead Electrochemical Cell. The rest of the parameters (P, T, H, L) were added to the test and demonstrated the versatility and scalability of the system. The program in the ESP8266-01 microcontroller provides a ramp signal in voltage to the potentiostat while recording the currents produced during the reaction. The resulting data is sent to the mobile application and is represented by a plot current vs. voltage called voltammogram.

The ADS1115 module is an ADC with four channels of 15-bits + 1-bit of precision. Therefore, each port can be used to collect more parameters. The port A0 of the ADC measures (I) on the potentiostat and the module MCP4725 12-Bits DAC applies V(t). The modules BME280 measure (P, T, H), and the Ultimate GPS Breakout v3 measures (L). Figure 3 shows the sensor node circuit and the set of parameters of each module as a single sensor.
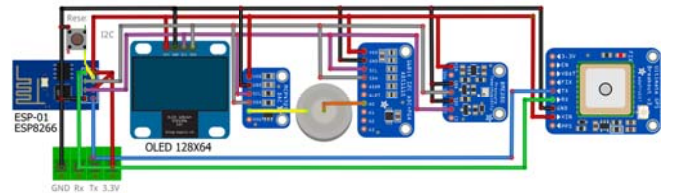

Figure 3. Sensor-Node circuit (M(1) + S_0)

The Wi-Fi transmission module is the node M(n), and the potentiostat and the other measuring circuits are the sensors Sn{I, V, P, T, H, L}. Both sets are a sensor-node. The entire circuit is shown in Figure 3 and was performed with the fritzing tool [17].

The ESP8266 was programmed to collect all sensor information using the Inter-Integrated Circuit (I2C) protocol. Then the ESP8266 puts this data on a PUBLISH message of the MQTT-Client protocol and sends it to the network with the Broker's destination address.

B. Testbed

The technical implementation of this proposal was possible through the mentioned modules. The target was using a low cost, fast and easy implementation to focus the tests on the communication system.

The network design was adapted to this architecture to convert a conventional network into an IoT network. The idea with this architecture is to take advantage of the access points of a home network and reuse it. Therefore, Figure 4 presents an IoT network design using a conventional Wi-Fi Access Point (AP) and a mobile device such as a Broker.
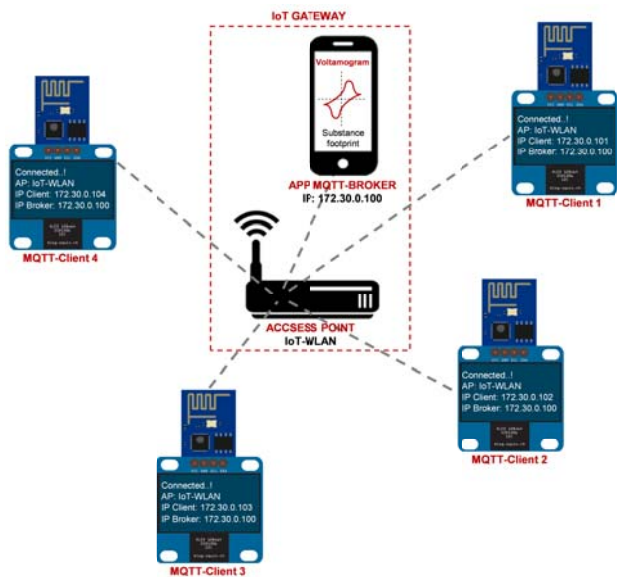

Figure 4.   IoT-WLAN proximity network design

For the MQTT-Broker protocol to work on the Mobile App, it was necessary to install Mosquitto on the smartphone [18]. The Arduino code of the ESP8266 uses a library called <PubSubClient.h> to implement the MQTT-Client protocol. The Arduino code of the ESP8266 uses a library called <PubSubClient.h> to implement the MQTT-Client protocol. In both cases, the Broker and the Client are connected to the network through the AP with Service Set Identifier (SSID: IoT-WLAN). MQTT is an open-source M2M protocol [19], and therefore, it is possible to modify its operation mechanics. This advantage allows new proposals by modifying the already developed code that has been shared through communities of programmers.

The tests were performed following the design of Figure 4. Additionally, a personal computer (PC) connected by Ethernet cable to the AP was included. On the PC, the Wireshark tool was used to capture the sent data from the sensor nodes to the Broker.

The captured data is used to analyze the Broker's performance after modifying the protocol algorithm of its usual operation. That is, the speed of the data may be affected by the intervention of the Broker in the processing of MQTT payloads. Therefore, the following measures were made: first with standard MQTT packages to have a reference and observe the system changes, and then, the MQTT packets were measured with the algorithm proposed one-to-one to know how the transmission speed was affected when the number of sensor-nodes increased.

Figure 5 represents a reference information flow of a M2M synchronization. The test was performed by initially transmitting a standard MQTT establishment process, including TCP sessions of connection initiation and termination (filter: tcp.stream eq 0). The transmission was made from the source address 172.30.0.101 with the messages Connect, Subscribe, Publish, Unsubscribe, and Disconnect with destination 172.30.0.102. Where the Connect Message is IP-Broker: 172.30.0.100, the Subscribe Message is Topic:/S1, and Publish Message is Topic:/S1, Payload: V, I. Figure 5 shows the number of packets per second TCP between a M2M synchronization after the broker pairs it. The filter used in the test to capture the packets was as follows: (ip.addr eq 172.30.0.101 and ip.addr eq 172.30.0.102) and (tcp.port eq 52620 and tcp.port eq 1883). The maximum peak value was 83 Bytes, while the minimum value was 54 Bytes, which belongs to an Unsubscribe Message. The average number of Bytes transferred was 63 Bytes.

In Figure 6, the broker intervention process is more evident. The values sent of V and I are longer data strings. This Figure shows the relation between the numbers of packets transmitted per second. The average duration of each packet sent through the network was 77s.
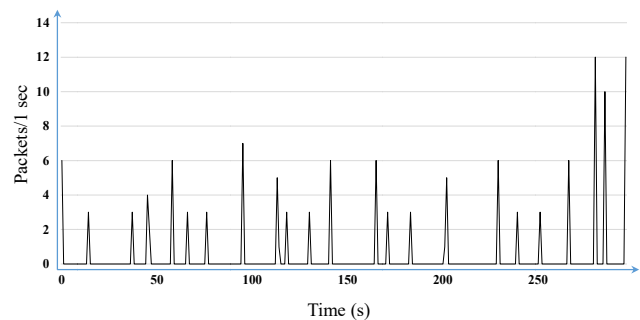

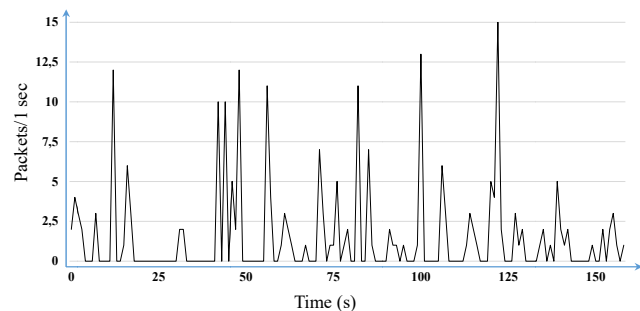Figure 5.   Machine-to-Machine synchronization M(S0)-M(S1)

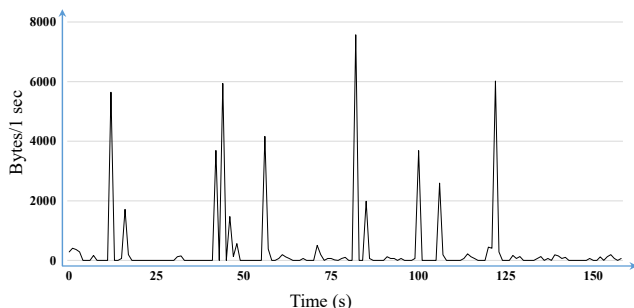Figure 6.   Datastream M(S0)-M(S1) with values payload of V and I

Figure 7. Pair M(S0)-M(S1)

In Figure 7, the maximum values are peaks of 1392 Bytes, while the minimum values were 42 Bytes and the average values were 266 Bytes.

V. CONCLUSION

Currently, there are some experimental implementations of potentiostats on small boards that are very efficient in comparison to the traditional desktop stations used in laboratories. However, its connectivity is limited to the monitoring of information. With this contribution, the possibility of using the IoT concept to configure proximity wireless networks and transmit its information in a collaborative environment is feasible.

The architecture on which the network was designed allows flexibility in adapting existing networks to the IoT. Therefore, it can be used in any application such as the electrochemical measurement and analysis system (potentiostats) of this work.

The algorithms of the App Broker, built from the deduced equations from the architecture, improve the information processing. As a result, the tests showed that the MQTT packages are not affected in its speed due to the intervention of the Broker in the payloads.

In the network tests a conventional Wi-Fi AP was used and, although outdoor distance reaches an approximate coverage of 100 to 150 meters, it is possible to use GPS as an optional parameter (L). There are many applications and where this solution may be necessary such as, for example, whenever the coverage is extended with repeaters or a low-power wide-area network (LPWAN) is used.

We expect to do the same experiment with more client sessions (at least 10 to 15 clients) to see the performance. In the same way, we would like to try other M2M protocols such as CoAP and HTTP Restful. Moreover, it would be interesting to implement the Broker on an ESP8266 or replace the AP with an IoT Gateway (Broker) based on a single-board computer (SBC) such as the Raspberry Pi 3 Model B + (RPi3).

In future works, we will perform more tests that measure the bandwidth based on the size of the parameters, including not only transmitted parameters such as (V, I), but also the rest of parameters (P, T, H, L).

REFERENCES

[1] A. Charles, *Electrochemical techniques in corrosion science and engineering, Hardback*, vol. 48, no. 8. 2003.

[2] Arduino, "Accessories and devices of reduced board." [Online]. Available: https://www.arduino.cc/. [Accessed: 10-Oct-2019].

[3] Adafruit, "Adafruit Industries, Unique & fun DIY electronics and kits," 2015. [Online]. Available: https://www.adafruit.com/. [Accessed: 10-Oct-2019].

[4] A. Ainla *et al.*, "Open-Source Potentiostat for Wireless Electrochemical Detection with Smartphones," *Anal. Chem.*, vol. 90, no. 10, pp. 6240–6246, May 2018, doi: 10.1021/acs.analchem.8b00850.

[5] T. Dobbelaere, P. M. Vereecken, and C. Detavernier, "A USB-controlled potentiostat/galvanostat for thin-film battery characterization," *HardwareX*, vol. 2, pp. 34–49, Oct. 2017, doi: 10.1016/j.ohx.2017.08.001.

[6] D. M. Jenkins, B. E. Lee, S. Jun, J. Reyes-De-Corcuera, and E. S. McLamore, "ABE-STAT, a fully open-source and versatile wireless potentiostat project including electrochemical impedance spectroscopy," *J. Electrochem. Soc.*, vol. 166, no. 9, pp. B3056–B3065, 2019, doi: 10.1149/2.0061909jes.

[7] V. Bianchi, A. Boni, S. Fortunati, M. Giannetto, M. Careri, and I. De Munari, "A Wi-Fi cloud-based portable potentiostat for electrochemical biosensors," *IEEE Trans. Instrum. Meas.*, pp. 1–1, Jul. 2019, doi: 10.1109/tim.2019.2928533.

[8] P. Li, T. R. Molderez, F. Ceyssens, K. Rabaey, and M. Verhelst, "A 64-channel, 1.1-pA-accurate On-chip Potentiostat for Parallel Electrochemical Monitoring," *ESSCIRC 2019 - IEEE 45th Eur. Solid State Circuits Conf.*, pp. 317–320, 2019, doi: 10.1109/ESSCIRC.2019.8902687.

[9] J. Dieffenderfer, M. Wilkins, C. Hood, E. Beppler, M. A. Daniele, and A. Bozkurt, "Towards a sweat-based wireless and wearable electrochemical sensor," in *Proceedings of IEEE Sensors*, 2017, doi: 10.1109/ICSENS.2016.7808470.

[10] P. L. G. Ramirez, M. Taha, J. Lloret, and J. Tomas, "An Intelligent Algorithm for Resource Sharing and Self-Management of Wireless-IoT-Gateway," *IEEE Access*, pp. 1–1, 2019, doi: 10.1109/ACCESS.2019.2960508.

[11] R. Diestel, *Graph Theory*. Springer-Verlag New York, 2000.

[12] B. A. Galler and M. J. Fisher, "An improved equivalence algorithm," *Commun. ACM*, vol. 7, no. 5, pp. 301–303, 1964, doi: 10.1145/364099.364331.

[13] J. Lloret, S. Sendra, P. L. González Ramírez, and L. Parra, "An IoT Group-Based Protocol for Smart City Interconnection," in *Smart Cities*, Sergio Nesmachnow and L. H. Callejo, Eds. Soria, Spain, September 26–27,: Communications in Computer and Information Science, vol 978. Springer, Cham, 2019, pp. 164–178, doi: 10.1007/978-3-030-12804-3_13

[14] P. L. González Ramírez, J. Lloret, J. Tomás, and M. Taha, "Architecture to Integrate IoT Networks using Artificial Intelligence in the Cloud," *IEEE Xplore*, pp. 996–1001, 2019, doi: 10.1109/CSCI.2018.00193.

[15] "Linear Sweep and Cyclic Voltametry: The Principles | Department of Chemical Engineering and Biotechnology." [Online]. Available: https://www.ceb.cam.ac.uk/research/groups/rg-eme/Edu/linear-sweep-and-cyclic-voltametry-the-principles. [Accessed: 16-Dec-2019].

[16] T. Instruments, "LMP91000 Sensor AFE System: Configurable AFE Potentiostat for Low-Power Chemical-Sensing Applications," p. 36, 2014.

[17] Friends-of-Fritzing foundation, "Fritzing Fritzing," *fritzing.org*, 2017. [Online]. Available: https://fritzing.org/home/%0Ahttp://fritzing.org/home/.

[18] "MQTT Broker on Android | How To Run MQTT Broker in Android." [Online]. Available: https://iotbyhvm.ooo/mqtt-broker-on-android-how-to-run-mqtt-broker-in-android/. [Accessed: 01-Oct-2019].

[19] IBM and Eurotech, "MQTT V3.1 Protocol Specification," 2010. [Online]. Available: https://public.dhe.ibm.com/software/dw/webservices/ws-mqtt/mqtt-v3r1.html. [Accessed: 08-May-2019].