Software Model for a Low-Cost, IoT oriented Energy Monitoring Platform

Elias M. Pinheiro[1], Sérgio D. Correia[2]
School of Technology and Management, Polytechnic Institute of Portalegre
Portalegre, Portugal

Abstract

The depletion of natural resources and increased energy consumption has given greater focus on energy efficiency, with many governments and institutions seeking new technological solutions.IoT has been identified has a promising research field.In the following paper we propose a low-cost open energy monitoring platform based on the MQTT messaging protocol, with great emphasis given to future integration on decentralized IoT networks.

Keywords:*Energy Monitor, Wireless Sensor Network, MQTT, Open Software*

I. INTRODUCTION

Increased consumption of electrical energy and the subsequent increase in demand for fossil fuels has created severe economic challenges, especially in emerging economies. On the other hand, the increase of greenhouse gas emissions has led to a situation of climate change, which, sooner or later, will have to be addresses globally [1]. In response to these challenges, governments have adopted and wide array of measures and policies, at various levels [2]. Of interest is the focus given to energy efficiency and new technological solutions.

In the wake of the 2008 global financial crisis, the European Union (EU) has devised a plan, aptly named "A European Economic Recovery Plan" [3], to overcome the then current crisis. It identifies investment in energy efficiency as a factor of long term competitiveness and singles out inter-connection to obtain it. The link between energy efficiency and technological innovation is further explored in the EU document entitled "Internet of Things - An action plan for Europe" [4]. In it, the EU enumerates a series of lines of action to fully harness the capabilities and potential advantages of the IoT paradigm.

We have previously established the connection between energy efficiency, and energy management and monitoring systems [5]. Expanding on the concept, in the current paper we propose a software architecture of a low-cost energy monitor solution, that, while functioning like a WSN, does make some concessions to the IoT paradigm, to allow an easier and more seamless integration with wider networks.

The rest of this paper is organized as follow: in section II a brief description is given on the most commonly used technologies in WSN and IoT networks; a short overview of the hardware architecture is presented in section III; section IV details the implementation of the proposed platform, while section V provides a discussion on the chosen technologies; finally, conclusions are presented in section VI.

II. STATE OF THE ART

IoT consists of a large number of different devices networked together, and communicating through the use of different interfaces and protocols. Therefore, traditional synchronous point-to-point communication is inadequate for an IoT based solution [6]. It is important to make a clear distinction between an IoT network and a WSN. AnidealIoT network is composed of different nodes, with different purposes and functionalities, connected in an apparent seamless way, despite utilizing a wide array of different technologies. They also do not serve a unique and clearly defined purpose, instead operating in the "good servant" [7] principle: as unassuming as possible while maximalising usefulness [8]. On the other hand, WSN are usually composed of many nodes, similar to each other in composition and function. Unlike IoT networks, WSN do have distinct and clear goals, sensing physical data over significant geographical distances [8] and every component is geared towards it. They are also controlled in a centralized way, through a central device, which is the only point of interaction between the network and the users, with the remaining components being self-manageable [9].

Despite their differences, both types of networks are closely linked. In fact, WSN's are most often used to provide data for IoT systems [6]. While the ultimate purpose of IoT is to become a global infrastructure for interconnected things through interoperable technologies in order to provide advanced services and production of information [10], this is not yet a reality. In truth, currently,IoT consist of loose and disparate purpose-built networks connected together [11]. In other words, currently, IoT is a network of sensor networks.

This poses a problem in terms of communication. Since most IoT solutions require a very large number of connected devices, their design requires a resource-oriented communication interface, such as *Representational State Transfer* (REST), currently the most common one. However, while being used with

great success in web technologies, REST does not lend itself to the limitations of an IoT environment [12]. While REST HTTP is an appropriate solution when using computation devices of reasonable power, it is inadequate in more resource constrained devices [13]. Furthermore, REST, being essentially Service Oriented, in the sense that each device offers a service to another in a request-reply model [9], is appropriate at moving data between different software applications, but not necessarily at moving data between different components. In a fully developed IoT solution, this would not be a limitation, allowing different devices to communicate with each other through services, seamlessly merging different networks into one another, despite greater load in the infrastructure. However, as stated before, currently, IoT consist of a network of sensor networks, which are centralized, with a single device being the sole point of interface. Therefore, REST HTTP is not adequate for device to device communication in an IoT environment, except for communication between central devices in different networks.

As an alternative, instead of a request-reply model based protocol, we could use a publish-subscribe (pubsub) based one, the most prominent of those being *MessageQueuing Telemetry Transport* (MQTT).

This model utilizes two different classes of clients, publishers and subscribers and a central piece called the broker. A client interested in receiving data (subscriber), instead of making a request to the server, subscribes to a particular topic. On the other hand, being an event based architecture, when certain predetermined conditions are met on a specific topic, the publisher sends its data to the broker, which in turn, filters it and routs it to the appropriate subscribers [13]. Thus, in anIoT environment, this model allows for increased scalability and easier interconnection between devices, by allowing dynamic, n-to-n and asynchronous communications [14].

Furthermore, performance comparison between the two protocols has demonstrated that MQTT shows less latency, less bandwidth usage and less energy consumption when compared to REST HTTP, all critical factors in resource deprived IoT infrastructures [13].

III. HARDWARE OVERVIEW

In previous work we have proposed a low-cost energy monitor solution based on open hardware, with a focus on scalability. Electric voltage and current sensors communicate with an Arduino board through an I2C bus. The collected data is then processed and send through wireless LAN to a Raspberry Pi, which operates as the central point of the network and is its only point of interface. A detailed hardware implementation can be found in [5].

IV.IMPLEMENTATION

Unlike a traditional WSN, an energy monitor solution is generally not required to be implemented over a large area, usually only needing to cover a single building. For similar reasons, sensors nodes often have access to a permanent power source, thus disregarding the need for a battery. Therefore, limitations are restricted to the node's processing power and available bandwidth. For ease of presentation, we have divided the solution in three distinct layers: Sensing layer, Communication Layer and Services Layer.

A. Sensing Layer

The sensing layer corresponds to the sensor nodes. These are composed of three different sub-systems: processing subsystem, responsible for processing data;sensing subsystem, responsible for acquiring data; and communications subsystem, responsible for transmitting data (Fig. 1).

1) Sensing Subsystem:

The sensing subsystem consists in a variable number of sensors, depending on implementation needs, connected to an I2C interface board, to improve scalability while diminishing cost. The I2C interface allows the processing subsystem to function as master to several slaves.

2) Communication Subsystem:

Allows the node to communicate with the remaining network.Consists of an ESP8266 module connected to an Arduino through a serial connection. Because ESP8266 can host its own application, it is possible to use a network manager. Upon start-up, if a network is not available, the module will act as an access point, hosting a rudimentary web server, providing the user with a graphical user interface. Upon configuration, it connects to the chosen Wi-Fi network, and stores the necessary parameters in EPROM.

3) Processing subsystem:

Consisting in an Atmel microcontroller featured in an Arduino board, it is responsible for all computation required at the sensing layer level. It reads data sent from the sensors from an I2C bus, performs the necessary calculations to convert the input values into electric voltage and current, and transmits it to the central server. Upon start-up the Arduino connects to the network whose credentials are stored in the ESP8266 module. Afterwards it communicates to the server its ready state and awaits response with configuration parameters. It periodically reads the values from the I2C addresses received from the server, and makes the necessary computations. It then transmits the data to the server. Between readings, it sweeps the available range of I2C addresses in search for newly plugged-in devices, thus attaining plug and play capabilities. If a new device is found, its address is communicated back to the server, where it is stored and awaits user configuration.

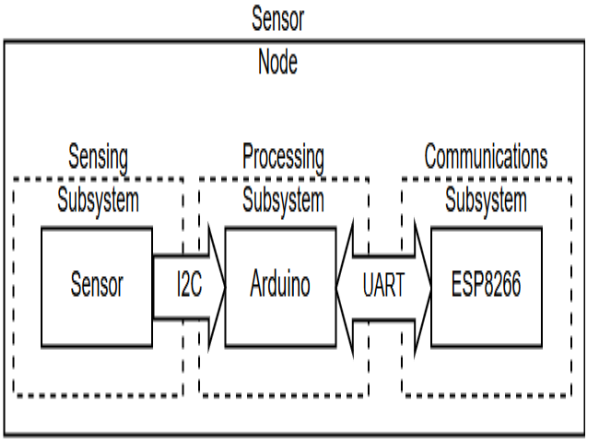

Fig 1: Block diagram of the sensor node

B. Communication Layer

The communication layer allows communication between all the devices in the network. However, due to the star configuration of the specific network of the proposed solution, it serves only for communication between each of the sensor nodes and the central server. While typical WSN's use a non-IP network for the sensor nodes, because of the mentioned characteristics of the implementation, namely, relatively small areas with a comparatively small number of devices, such is not required. The use of an IP network for the sensor nodes is a cost reducing measure, since it excludes the need for a transparent gateway to bridge an IP network with a non-IP one. Furthermore, if one so desires, allows the entire solution to function on HTTP based messaging protocols, although that would be inefficient due to the sensor nodes limited processing capabilities.

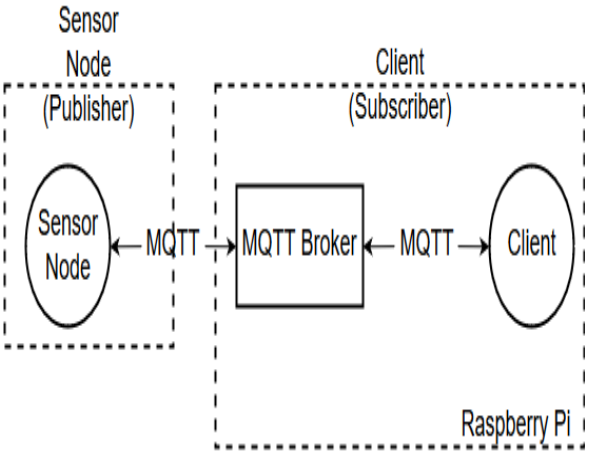

Fig 2: MQTT network topology during regular functioning

Messaging is done through MQTT protocol (Fig. 2). Each sensor node functions as both publisher and subscriber. When first connect to the wireless network, it publishes a ready message to a dedicated topic and then subscribes to the same topic for a response, which contains configuration parameters. It then continues its operating cycle. The Raspberry Pi also functions as a publisher and subscriber, as well as the MQTT broker. It subscribes to sensor readings topics, and publishes configuration parameters. Communication is made through a dedicated wireless LAN originating from the Raspberry Pi, although a general-purpose LAN can be used if the need arises.

The Raspberry Pifunctions both as Broker and Client. The reason for this is database integration. Further details are given in the Services Layer.

To provide each sensor node with a unique IP and avoiding the need for manual configuration, the Raspberry Pi serves as a DHCP server (Fig. 3).

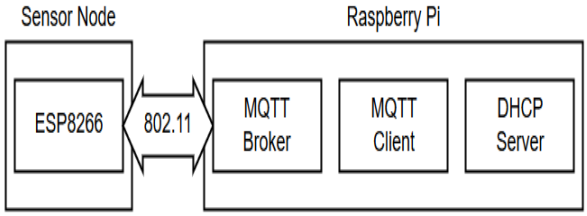

Fig 3: Block diagram of the communication layer

C. Services Layer

The services layer is mainly responsible for managing the entire network and user interfaces. It is centred in the Raspberry Pi and is composed of several different elements (Fig. 4).

1) MQTT Broker:

As mentioned above, MQTT networks require a broker to filter and rout messages. In the proposed solution, the Raspberry Pi is the most advisable device to operate as a broker due to its robust processing power. However, a dedicated device might be used. In this implementation Eclipse Mosquitto MQTT Broker is used.

2) MQTT Client:

A MQTT client is required to store the received messages in a database, since no open source broker supports database integration. EclipsePaho MQTT client is an open source python based MQTT client which easily integrates MQTT functionalities with a database connector. When a message is received in the subscribed topic, a call-back is triggered that parses and insert the relevant data into the database. Similarly, when data must be transmitted, a python script is called that publishes to the appropriate topic.

3) Database:

To store data and configuration parameters, a database is required. While data might be sent directly to the user using a subscribed MQTT client, the use of a database extends the range of functionalities. It allows for statistical analysis and integration with decision support systems, thus extending the solution abilities to generate information from the acquired data. A mySql server is used.

4) DNS Server:

To simplify the integration of new nodes into the network, a DHCP server is configured in the Raspberry Pi. Any new node connecting to the shared Wireless LAN is automatically attributed a new IP.

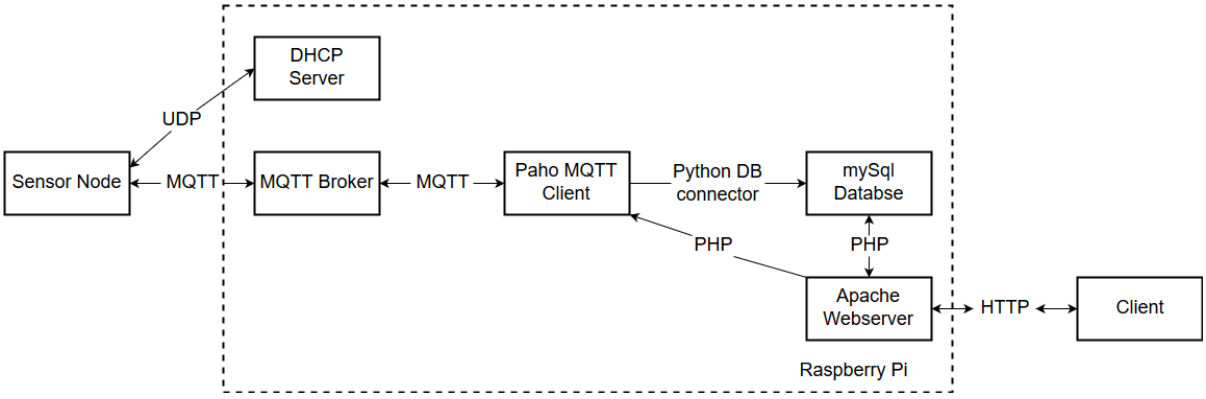

Fig 4: Services layer block diagram

Therefore, since IP is dynamic, each node MAC address is used for identification purposes.

5) Web Server:

An Apache Webserver is used to provide a graphical user interface. This interface allows the user to configure specific devices, as well as consult the stored and processed graphical data.

V. DISCUSSION

There is currently a strong institutional and corporate push towards the IoT paradigm. However, its full realization is not yet possible, due to hardware limitations and lack of standards [10]. Nonetheless, we propose a solution, which, while not strictly an IoT application, does make concessions towards its future.

The focus is on low cost components and high potential of scalability. The use of open hardware and software significantly reduces the overall cost of installation. Scalability is guaranteed by its choice of communication protocols. The use of I2C protocol in sensor nodes enables monitoring of many circuits with a single node, therefore reducing cost in processing units, namely Arduinos, and communications modules. The use of an IP network, allied with the characteristics of implementation, allows for the use of an IP sensor network, which further decreases cost by relinquishing the need for dedicated gateways.

The use of the MQTT messaging protocol is of paramount importance to the success of our purposed goals. As established before, HTTP based request-reply protocols are not ideal in an IoT environment. While the proposed solution is not an IoT application, we consider the ability to simplify integration with other networks, or, in the case of a fully developed decentralizedIoT solution, the ability for device-to-device communication, added value.Since MQTT operates on the TCP/IP protocol, the network follows the Open Systems Interconnection model (OSI model). This provides many advantages. It avoids the problem of small MTU's common to many non-IP protocols; due to the publish-subscribe model, mesh network routing is simplified; being single-hop architecture mitigates the problem of multicast efficiency; and it bypasses the need for resource discovery, since the publishers and subscribers do not need to know each other. However, the use of TCP/IP based protocols also presents some disadvantages, particularly at the transport layer level: due to energy economy, devices in IoT networks often fall into sleep mode, thus invalidation TCP long lived connections; the small amounts of data transmitted are often offset by the large overhead when establishing connections; and finally, critical applications might require low latency data transfer, and TCP handshake might be outside the bounds of its tolerance [13]. However, none of these problems invalidates our proposed solution, since it does not require significant energy economy, uses a dedicated Wi-Fi network and has high tolerance for latency.

VI.CONCLUSION

The emerging IoT paradigm, while not yet fully realized, uses WSN as its building blocks, by interconnecting previously contained and purpose-built networks. We propose an energy monitoring platform that, while operating like a typical WSN, is engineered to facilitate migration to an IoT environment. Its main advantage is the ability to perform simultaneously as a self-contained network with a single interface-point, or as a part of a larger decentralized network. The use of the MQTT protocol provides device-to-device communication without changes to the existing infrastructure, by subscribing new devices to the same topics, thus allowing integration with a fully developed IoT application

The concurrent use of a Raspberry Pi as a central device has the added advantages of allowing for a more focused GUI and integration with decision support systems. As showed, MQTT provides better performance in limited hardware, such as sensor nodes, while the use of REST HTTP brings the advantages of SOA, at the expense of higher resource consumption. Our solution benefits from both technologies, by focusing the services layer on the Raspberry Pi, a

more robust platform, thus able to better support REST services, while reducing load on the sensor nodes, by using MQTT.

ACKNOWLEDGMENT

The present research was developed as a final graduation project of a Computer Sciences degree at the School of Technology and Management of the Polytechnic Institute of Portalegre, Portugal.

REFERENCES

[1] S. F. Hafeez and S. D. Dalvi, "Global trends of energy efficiency programs: 2010 to 2025" in2017 Asian Conference on Energy, Power and Transportation Electrification (ACEPT), 2017, pp. 1-8.

[2] L. Hurtado, P. Nguyen, W. L. Kling, W.Zeiler, (2013). "Building Energy Management Systems — Optimization of comfort and energy use." InProceedings of the Universities Power Engineering Conference, 2003, pp. 1-6

[3] Commission ofthe European Communities, "A European Economic Recovery Plan", Brussels, 26.11.2008, COM(2008) 800 final

[4] Commission ofthe European Communities, "Internet of Things – An Action plan for Europe", Brussels, 18.6.2009, COM(2009) 278 final

[5] E. Pinheiro, S. Correia, "Hardware Architecture of a Low-Cost Scalable Energy Monitor System", International Journal of Engineering Trends and Technology(IJETT), Volume 61 Number 1, pp. 1-5, July 2018

[6] S. R. Akbar, K. Amron, H. Mulya, S. Hanifah, "Message Queue Telemetry Transport Protocols Implementation for Wireless Sensor Networks Communication - A Performance Review" in 2017 International Conference on Sustainable Information Engineering and Technology (SIET), 2017, pp. 107-112

[7] M. Weiser, "The computer for the 21st century", *Scientific American*, vol. 265, pp. 94-104, 1991

[8] K. K. Kumar, K. S. Rao, "An Efficient users Authentication and Secure Data Transmission of Cluster Based Wireless Sensor Networks", International Journal of Computer Science and Engineering(SSRG-IJCSE), Volume5,Number 1,pp. 1-5, January 2018

[9] J. Rykaowski, D. Wilusz, "Comparison of architectures for service management in IoT and sensor networks by means of OSGi and REST services" in Proceedings of the 2014 Federated Conference on Computer Science and Information Systems, 2014, pp. 1207–1214

[10] F. Wortmann,K.Flüchter, "Internet of Things: Technology and Value Added",Business & Information Systems Engineering, Volume 57,pp. 221-224. March 2015

[11] D. Evans, "The Internet of Things: How the Next Evolution of the Internet is Changing Everything" [White Paper], Cisco Internet Business Solutions Group, 2001

[12] W. Shang, Y. Yu, L. Zhang, R. Droms, "Challenges in IoT Networking via TCP/IP Architecture", UCLA, CA, Cambridge, MA, NDN Technical ReportNDN-0038, 2016

[13] J. Dizdarević,F. Carpio,A. Jukan, X. Masip-Bruin, "ASurveyofCommunication Protocols for Internet-of-Things and Related Challenges of Fog and Cloud Computing Integration."ACM Computer. Surveys, Volume 1, Number 1, pp. 1-27, April2018

[14] S. Patel,S. Jardosh,A. Makwana, A. Thakkar,"Publish/SubscribeMechanismforIoT:A Survey of Event Matching Algorithms and Open Research Challenges."in Proceedings of International Conference on Communication and Networks, 2017, pp. 287–294