Remote Water Consumption Metering using MQTT over Sub-gigahertz RF Environment

Fábio Victor Schreiber
CERTI Foundation
Campus da UFSC, C section,
Florianópolis, SC, Brazil

Roberto Alexandre Dias
Federal Institute of Santa Catarina - IFSC
950 Av. Mauro Ramos
Florianópolis, SC, Brazil

Victoria Zanetti Marçal Gallo
Federal Instituteof Santa Catarina – IFSC
950 Av. Mauro Ramos
Florianópolis, SC, Brazil

ABSTRACT
Most remote water metering systemsuse proprietary protocols and technologies that do not communicate with each other in a friendly way and this could be a problem for an enterprise with more than 4.3 million square meters and more than 750 measurement points. The use of equipment that is capable of collecting information and communicatingwith open protocols, based on IoT concepts, can be seen as a solution for such an undertaking.Thus, this article presents questions about the development of an integrated communication gateway with cloud service via MQTT protocol, for a water consumption measurement system in an enterprise using a low power microcontroller system with radio frequency data transmission operating in the sub-gigahertz band.The goal of using a radio frequency system is to maximize the coverage area per gateway installed while minimizing the costs of the equipment for widening the Wi-Fi network.

General Terms
Smart Metering, Water Metering System, Communication Protocols.

Keywords
MQTT, cloud service, water metering system, sub-gigahertz radio frequency,IoT.

1. INTRODUCTION
In small buildings or even small businesses, measuring the water consumption of the units is a relatively easy task because of the small number of measuring points. Nonetheless, in a company with more than 4.3 million square meters, where there will be more than 750 measuring points served by three different sources,this task becomes much more expensive and complex.

In addition, due to the water consumption growth and the reduction of the availability of this resource in Brazil [1], accurate and real-time information is important to identify undesirable situations such as leaks and inadequate consumption.

Brazilian public services are governed by specific rules that do not guarantee, at different stages, the continuity of purchase of equipment of the same manufacturer, thus creating conditions that undermine the interoperability of the equipment.

There is great difficulty in managing a water measurement system in a large territorial space with equipment that does not have interoperability [2], since the efforts used to obtain the information are often directed to make the equipment communicate and not to the service of measurement itself.

For an enterprise of this size, where water bills are under the responsibility of a main office, the numbers involved require quick and accurate responses to charge the values consumed.

In the current IoTscenario, where every day new technologies are available to connect "things", traditional meter reading is no longer an option, specially considering that even today most of the world's utility companies perform data collection manually [3].

IoTprovides the desired information in an easy, quick and accurate way, offering mankind the necessary power to act more appropriately.The potential of IoT is to give objects the power to independently perform actions and generate data, which become part of the world in this context, by configuring new ways of acting [4].This can be very useful in a system of remote measurement for water consumption.

Another important point is that by taking advantage of the granularity and exactitude of information that a remote smart metering system has, significant improvements take place in the management of utilities, which,in turn, encourages consumers to participate in monitoring and managing consumption [5]. In addition, it becomes possible to control the water consumption of a city, to plan maintenance operations of the distribution system, to manage water flow rates, to reduce losses and failures of the distribution system [6], as well as to identify inappropriate consumption and fraud.

Therefore, the objective of this paper is to propose a communication gateway for an intelligent system of remote measurement of water consumption, based on IoT concepts, using MQTT protocol integrated with cloud service.

2. WATER METER
The measurement of water is only possible indirectly and, for this, the measuring equipment uses systems of conversions based on two principles:Fluid velocity across a known cross-sectional area or addition of known volumes at regular time intervals.

Despite the principle of conversion, there are two basic elements classified as primary and secondary, where the first is in contact with the fluid to be measured and the second realizes the transformations of the quantities obtained in analog or digital displays.

There are several types of commercial meters with different technologies, but for the sake of testing, a mechanical meter pulse was used with emission of pulses from the ground to each unit of volume consumed.

To compose the remote metering system, a microcontroller system was installed,which has low power consumption, with radio frequency data transmission in the sub-gigahertz range, programmed to receive and interpret the pulses according to the meter configuration and, along with this meter, to send, via radio frequency, the data collected to an identical microcontroller equipment that functions as a gateway.

The use of the radio frequency system has the goal of maximizing the coverage area per installed gateway, thus minimizing the costs of investing in equipment to expand the Wi-Fi network.

3. HARDWARE AND SOFTWARE

3.1 Evaluation Board CC1200DK

The Texas Instruments CC1200DK development module is a platform developed for hardware performance testing and software development for sub-Giga hertz devices.

The equipment has a pre-test program installed, which allows rapid evaluations of the RF device with different data packet configurations, transfer rates and transmission power.

Figure 1 portrays the set which is composed of: (a) Two TrxEB plates - SmatRFTransnceiver Evaluation Board, (b) Two CC1200 Evaluation Module Kit 868-930Mhz with micro controllers MSP430F5438A, (c) One MSP-FET430UIF debugger for MSP430 and (d) Two antennas for RF transmission.

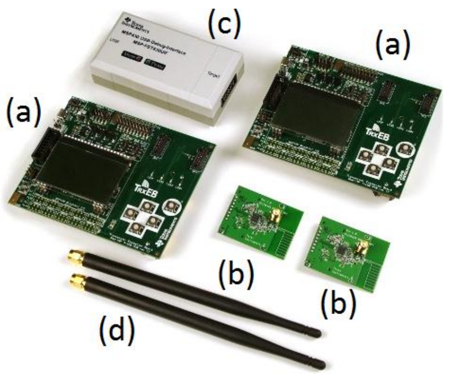

Fig 1: CC1200DK Development Module

The TrxEB board acts as a motherboard for a variety of development kits for low-power integrated circuits from Texas Instruments [7], and it has a number of external connection interfaces that enable rapid prototyping as well as testing of hardware and software.

The MSP430 family of microcontrollers has been developed as ultra low power devices to provide extended battery life.

The MSP430F5438A series has three 16-bit timers, a high-performance 12-bit ADC, four universal serial communication interfaces (USCIs), a direct-memory access hardware multiplier, an alarm-capable RTC module, and up to 87 pins of inputs and outputs. Typical applications include use in analog and digital sensors, remote device controls, and portable meters.

It features a CC1200 radio transceiver designed to operate alongside a micro controller and in ultra low voltage on wireless systems which is primarily intended to meet ISM frequencies and also SRD devices in the range of 164-190MHz, 410-475MHz and 820-950MHz.The CC1200 can automatize a number of common RF-related tasks and can, thus, greatly ease the micro controller.

Some IO pins of the CC1200 are connected to the interrupt ports of the MSP430F5438A and the serial communication port. If the use of a serial peripheralis seen as unnecessary, the pins can be used for another application. Figure 2 shows the connection between the IO pins of the CC1200 and the interrupt ports of the MSP430F5438A.

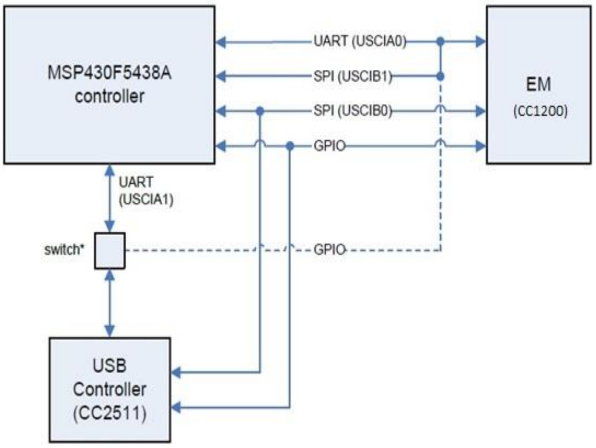

Fig 2: Evaluation module interface

3.2 Wireless Meter-Bus

Wireless Meter-Bus or WM-Bus is the standardapproach, adopted by the European community for smart reading of meters, used by utility operators such as water, electricity and gas.

It uses the 169 MHz, 434 MHz and 868 MHz frequencies, which are license free bands in Europe andwhich, consequently, provide lower costs for public service operators. In addition, they also have better wave propagation in hard-to-reach places compared to the 2.4GHz frequency.

From the several European documents that establish the use of frequency bands, the EN13757 [8] guides the standard WM-Bus. Published by the European Committee for Standardization [9], the EN13757 is divided into six parts as follows:

EN13757-1 - Communication system for meters and remote reading of meters - Part 1: Data exchange;

EN13757-2 - Communication systems for and remote reading of meters - Part 2: Physical and link layer

EN13757 -3 - Communication systems for and remote reading of meters - Part 3: Dedicated application layer

EN13757 -4 - Communication systems for meters and remote reading of meters - Part 4: Wireless meter readout (Radio meter reading for operation in SRD bands)

EN13757 -5 - Communication systems for meters - Part 5: Wireless relaying

EN13757 -6 - Communication systems for meters - Part 6: Local Bus

According to the EN13757-4, which defines the parameters of the physical and link layers for systems that use radio frequency for reading remote meters, focusing on the unlicensed telemetry bands (SRD) from 868 MHZ to 870 MHz, the operating modes for SRD devices are: S, T, C, and R2.

The S or stationary mode is used for unidirectional or bidirectional communication between stationary or mobile devices.

The T-mode or frequent transmit mode allows the meter to work with walk-by or drive-by systems, as it constantly sends the readings at short intervals of time.

The C mode or compact mode is similar to the T mode but it sends more information with the same power.

The last mode is the R2, where the meters work with wake up messages. In this mode, the devices remain inthe receiving state, waiting for the "wake up" signal. After this signal is received, the meter is prepared to accept the data and communication is enabled.

WM-Bus became popular in Europe because of its simplicity to operate in star network format, without the need of an IP and in sub GHz frequencies, offering greater reach, with minimum software stack size and lower battery consumption.

Compared to the OSI model, the stack size of the WM-Bus is somewhat different, since it has fewer requirements in its layers [10].Figure 3 shows a comparison between the WM-Bus stack and the OSI model.

Traditional OSI model	WM-Bus
Application Layer	Application Layer (APL)
Presentation Layer	
Session Layer	
Transport Layer	Extended Data Link Layer (TPL or DLL+)
Network Layer	Data Link Layer (DLL)
Data Link Layer	MCU Hardware Config (HAL)
PHY Layer	PHY (HW)

Fig 3: OSI and WM-Bus stack comparison

Depending on the mode and type of device, the stack can be implemented with less than 32 KB of flash memory, which allows for the usage of a low cost microcontroller.The application layer is defined by the user, who can follow any desired pattern.

3.3 Wi-Fi Module With ESP8266

The ESP8266 is a SoC (System on Chip) device with a built-in 802.11 b / g / n / e / iWiFi network module.With RISC architecture, in the ESP8266 one can find GPIO connectors, serial communication buses, ADC input, PWM output, CPU, flash memory, internal temperature sensor, RAM for data and instructions and ROM for boot.

In addition to the Wi-Fi functionality, some models of the ESP8266 have several other features and a processor that allows integration with external sensors and other devices via the GPIOs.

Because these modelshave low power consumption when in sleep mode, they can be seen as a good option for IoT projects [11].

From the several models available in the market, the ESP-12f model was used for the proposed solution.In this model,there is a 32-bit CPU that works with the frequency of 80 MHz and that supports several security protocols like WEP, WPA and WPA2.

3.4 MQTT and Mosquitto Server

3.4.1 MQTT

The MQTT protocol, an acronym for the term Message Queue Telemetry Transport, is an extremely simple and lightweight messaging protocol that is designed for being used with devices that do not require high bandwidth consumption.These devices interact in publish/subscribe mode, that is, the exchange of information is done by publications addressed to the subscribers in publication environments called brokers.The broker makes the publications available to subscribers who are registered on certain topics.

Because of the simplicity of coding and the need for little information other than the data to be transmitted (without overhead), the MQTT has become ideal for the IoT scenario.

With MQTT, devices do not have to be constantly synchronized to the server. The sensors, when sending their readings, allow the network itself to discover the path and the best synchronization to deliver the information [10].In addition, the MQTT enables the use of devices with low processing power and low memory capacity.

Developed in the late 1990s by Andy Stanford-Clark of IBM and Arlen Nipperda of Arcom (now Eurotech), with the purpose of linking oil pipeline sensors to satellites, it is a protocol for communication that does not require synchronism between the sender and receiver in both space and time [12]. This favors scalability in network environments that are not reliable from a service continuity perspective.

The MQTT became an open OASIS standard in 2014, with support for several programming languages and implementations in free software [12]. Different of other protocols such as HTTP (Hypertext Transfer Protocol) and Extensible Messaging and Present Protocol (XMPP), MQTT manages to be much lighter and more flexible because of the publication and subscription model.

Very popular in IoT, M2M and embedded projects, MQTT is also gaining space in WEB applications and mobile devices that require efficient messaging [13].

The MQTT protocol is defined by two elements, the message broker and the client.

The broker, also called the MQTT server, has the function of transporting the information of the topics acting as the link between publishers and subscribers. Responsible for the authentication and authorization of MQTT clients, the broker receives and makes them available to subscribed clients.

A client can be anything from a simple information-gathering sensor to large servers or data processors or even any user who can interact with the broker through devices such as mobile phones or computers.It can also be simultaneously a message publisher as well as a subscriber. This exchange of publication / subscription messages occurs through the topics.

The message consists of the topic and the data to be transmitted, the latter one known as “payload”.Both publishers and subscribers can be subscribed to more than one topic at the same time, as well as being a publisher on a topic and a subscriber in another one.

A topic is a subject of interest to the clientcreated in the broker. Similar to the Uniform Resource Locator (URL) concept, it is the location of the broker where the information will be available for reading by various registered clients.

The great advantage of MQTT is that neither publishers nor subscribers need to be synchronized to send and receive messages.However, it is possible to send messages to the broker in a synchronous way and only continue execution after confirmation and reception.

Because the publisher and subscriber are decoupled, it becomes impossible to know if there is any subscriber on the topic that will "listen" to the messages that are sent, and depending on the solution requirements and the levels of service quality, it is sometimes necessary to make the subscribers into publishers in order to send a message, indicating that they have received and processed the information [13].

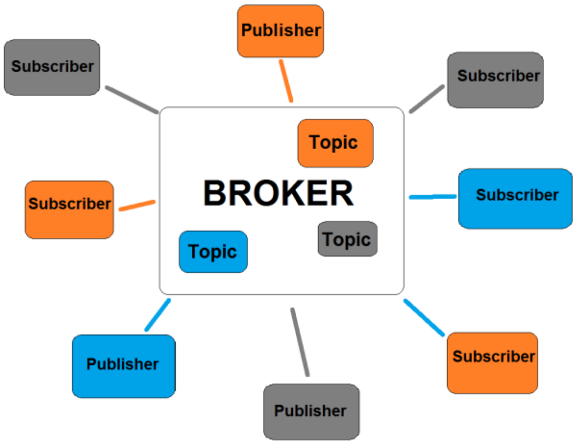

Fig 4: Modelo Publisher/Subscriber

To send messages or access the information of the topics, each client needs to make a connection with the server. There are several MQTT servers available for various platforms, such as LINUX, Windows and MacOS, with free and open source versions, free but proprietary versions or paid versions.The choice of a MQTT server should take into account the specific needs of the applications that are under development as well as the financial resources available for the project.

To make a connection, a client must send a data packet to the server, which must have all the necessary information to start the connection [10]. This includesthe Quality of Service information (QoS).

QoS is the agreement between a sender and a receiver about the guarantees and success of message delivery. According to the chosen QoS, there will be differences on the classification of the delivery's success [13].

There are three quality levels in the MQTT protocol. The lowest level is QoS 0, where messages are delivered once at most and occur according to network resources.No response confirmation is sent by the receiver and no repetition is performed by the sender [14]. The sender does not store or schedule any other messaging and the advantage is in the smallest overhead between others levels of security.

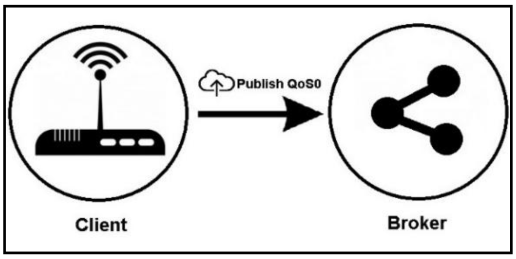

Fig 5: Connection diagram – QoS 0 level

At the QoS 1 quality level, there is a guarantee of delivering the message at least once and confirmation is given through a data packet called PUBACK - Publish Acknowledgment [14].The major disadvantage of this level of quality of service, however, is to generate duplicate messages if the sender does not receive acknowledgment of receipt within the set time.

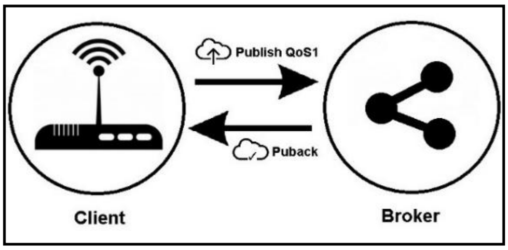

Fig 6: Connection diagram – QoS1 level

At the QoS 2 level, the message is guaranteed to be delivered only once [14]. At this quality level, there are two methods that can be used by the receiver, methods A and B, and they differ fromeach other due to the time when the message is delivered to the final destination.

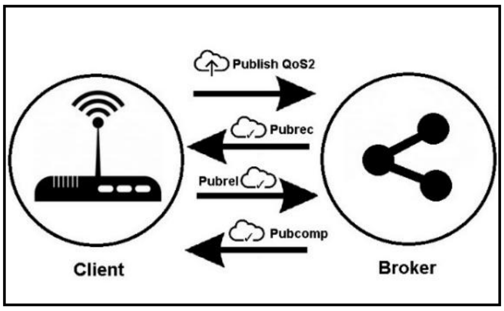

Fig 7: Connection diagram – QoS2 level

The selection of the QoS will depend on the application and the project.It is often necessary that messages be served with the least bandwidth consumption possible in an extremely reliable network, so that, if any data is lost, it will not be critical to the system. In this case, the use of QoS0 level would be the most appropriate [13].On the other hand, if the message is extremely important in an unreliable network, where these messages form commands for devices that cannot be run repeatedly,choosingQoS2 level should be an option to consider.

3.4.2 Mosquitto Server

The Mosquitto server is an open source messaging server (EPL / EDL),licensed by the Eclipse Foundation, that implements the MQTT protocol.

Mosquitto provides a lightweight server implementation with MQTT protocol that is compatible with a variety of situations, ranging from complex machines to low-power and low-processing embedded equipment.In addition to accepting MQTT client application connections, the Mosquitto server has a "bridge" system that allows connections to other MQTT servers in other Mosquitto instances, enabling the server networks to be constructed and allowing to send messages from any network to another, depending on the configuration of these bridges.

The Mosquitto is a high-level Eclipse Foundation project, whichlacks its own code but hosts a wide variety of projects covering a variety of subject areas. These projects tend to have limited lifecycles and eventually some become articles and others are incorporated as a foundation technology for other Eclipse Foundation projects [15].

4. PROBLEM DEFINITION

Sapiens Park is an enterprise located in the city of Florianópolis, capital of the state of Santa Catarina, in the south of Brazil. Its territorial extension corresponds to almost 4.3 million square meters with a prevision of more than 250 business units.

In order tostart this enterprise, studies were conducted with the goal of pointing out some actions to minimize the socio-environmental impact to the area. One of these actions has to do with the limitation on the consumption of drinking water.

According to the Technical Note 02 [16], the consumption of drinking water from public utility services cannot exceed 50% of its total. Hence, so as to meet this demand, alternative sources must be used.

These sources include rainwater harvesting and the reuse of wastewater,which is then treated by the company itself, all in the proportion of 14% and 36% respectively, small percentage variation allowed.

Aiming at both checking the actual consumption and, more importantly, charging upon the supply services, individual meters must be installed in each unit, which means that, at the end of the implantation of this equipment, there will be about 750 meters of consumption to the three types of water.

As a private company, with Santa Catarina state government holding the majority of its shares, Sapiens Park has specific rules for purchasing materials and hiring services. Moreover, if one also considers the long implementation period, the compatibility of measurement equipment will be a problem since there will be no guarantee of continuity of acquisition of proprietary technologies.

In view of the limitations described above and the consideration as a single consumer unit by the water local utility, the internal management of the potable water distribution as well as the respective collection of the consumed values should be carried out by the Sapiens Park team. In this sense, the remote measurement is perceived as an important factor for agility and confidence in the information, since the estimates for the end of the implantation indicate a period of 4 to 7 days for the reading of all the meters, if carried out manually.

The drinking water distribution network will be almost eight kilometers long and it will have three entry points for supply. Moreover, similar to the potable water network, there will be another network in parallel to supply treated wastewater with direct supply from the station treatment of sewage belonging to Sapiens Park. Figure 8 shows the distribution network with the utility's points of supply.

Fig 8: Sapiens Park water supply

5. PROPOSED SOLUTION

The proposed solution used a Texas Instruments CC1200DK development kit, coupled to an ESP8266 WiFi module, which, together, form the data hub for sending information to the server and another CC1200DK kit installed next to a pulsed output water meter for collecting and transmitting data of volumes consumed to the concentrator. Figure 9 shows the communication system.

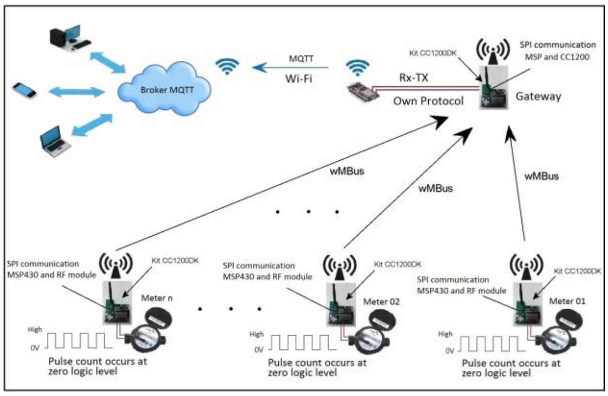

Fig 9: Communication system

In the solution's design, the reading process begins with the measurement done by a single-jet, which has a small magnet on one of its needle.At each turn performed by this pointer (which represents the measurement of a liter value) an interaction occurs with a switch sensor coupled to the outside of the meter.This interaction provides a short circuit condition at the terminals of the cable that is connected to one of the GPIO ports of the CC1200DK module and another one to the ground. The condition of the data IO port of the CC1200DK module has been programmed with a reading code to accumulate the record of each liter consumed the moment the voltage at the terminals returns to the high logic level.To monitor the voltage signals from the switch sensor installed on the meter, the P2.5 port was used.The choice of this port provided an alternative path for consumption simulations because it is associated with the down button and it is possible to simulate the switch sensor without necessarily having a water meter attached to the electronic board.This process keeps accumulating the data obtained in the memory of the CC1200DK and, at every hour, this amount is transferred first between the MSP430 and CC1200 boards via SPI communication and the latter is sent to the hub module by WMBus protocol via radio frequency.

In the concentrator module the process is inverse: when receiving the data through the WMBusprotocol, the CC1200 transfers the data to the MSP430 microcontroller (via SPI).The RX / TX serial communication is used with a proprietary protocol developed for the application to transfer information to the ESP8266.

In ESP8266, through its own protocol, the data is sent via Wi-Fi to the broker and in the latter the information is treated appropriately for the visualization in a friendly way for users.

The protocol used in the assembly of the packages and data transfer can be visualized in table 1.

Table 1: Código doprotocoloproprietário

Byte	Data
1	Meter ID
2	Functiontobemeasured
3	Day
4	Month
5	Year (thousand and a hundred)
6	Year (tens andunity)
7	Hours
8	Minutes
9	Seconds
10	Volume in liter (thousand and a hundred)
11	Volume in liter (tens and unity)
12	Sequential number of the measurement
13	CRC - 1º byte
14	CRC - 2º byte

6. RESULTS

6.1 RF range distance

The CC1200DK has a pre-test program for checking the transmission range that is already installed with the equipment.

It is possible to simulate transmissions of 100, 1,000, 10,000 and 65,000 data packets from 3 to 60 Bytes, with transmission rates of 1,200, 50,000 and 200,000 bps.

It is possible to work even by changing the output powers (Tx) in 14dB, 10dB, 5dB, 0dB, -5dB, -10dB and -16dB.

For the test, 100 packets of 14 Bytes were used, because it is the amount of data that will be transmitted during the measurements of the water consumption of the units of Sapiens Park, in all possible configurations allowed by the equipment until the occurrence of 100% of packet loss.

The choice of test site took into account the greater obstacle-free distance that could be obtained within the site of the project and the available buildings and units in a state of negotiation and construction.

The location and distance of the tests can be seen in figure 10 [17].

Fig 10: Local and test distance

The values obtained in the different power bands with a transmission rate of 1.2 Kbps, 50 Kbps and 200 Kbps can be visualized in table 2, 3 and 4 respectively.

Table 2: Received signal (dBm) at 1.2 kbps x distance (m)

1,2 Kbits/s								
Link M (dBm)		(db)						
		14	10	5	0	-5	-10	-16
Distance (m)	0	72	67	62	57	52	47	42
	5	32	28	23	20	15	12	6
	25	27	23	18	13	9	5	0
	50	21	19	13	10	2	2	-3
	100	25	23	19	13	7	1	*
	200	21	7	6	-4	-4	*	*
	300	23	7	-3	*	*	*	*
	400	22	10	*	*	*	*	*
	500	21	4	*	*	*	*	*
	600	15	-3	*	*	*	*	*
	700	15	*	*	*	*	*	*
	800	15	*	*	*	*	*	*
	900	15	*	*	*	*	*	*
	1000	14	*	*	*	*	*	*
	1100	12	*	*	*	*	*	*

* Not measured - distance before packet loss was already 100%

Loss of 100% of packages

Table 3: Received signal (dBm) at 50 kbps x distance (m)

50 Kbits/s								
Link M (dBm)		(db)						
		14	10	5	0	-5	-10	-16
Distance (m)	0	60	55	51	45	41	38	32
	5	33	29	24	18	15	10	5
	25	17	15	11	8	5	-5	-6
	50	15	12	5	3	-5	*	*
	100	10	6	-6	-5	*	*	*
	200	-5	-6	*	*	*	*	*
	300	*	*	*	*	*	*	*

* Not measured - distance before packet loss was already 100%

Loss of 100% of packages

Table 4: Received signal (dBm) at 200 kbps x distance (m)

200 Kbits/s								
Link M (dBm)		(db)						
		14	10	5	0	-5	-10	-16
Distance (m)	0	53	48	44	39	34	29	24
	5	24	18	15	8	3	-2	-8
	25	15	7	5	-1	-8	-8	*
	50	7	4	-8	-8	*	*	*
	100	4	-8	*	*	*	*	*
	200	-9	*	*	*	*	*	*
	300	*	*	*	*	*	*	*

* Not measured - distance before packet loss was already 100%

Loss of 100% of packages

With the test, it was possible to notice a strong signal loss as the amount of bits in the package increased. Still, for the application, the values of up to 1.2 Kbps are sufficient.

6.2 Network performance with MQTT

In order to analyze the performance of the MQTT protocol in relation to the size of the transmitted payload, two hardware configurations were implemented. The first one used two Beagle Bones Black boards with Ethernet network and a Mosquitto broker. In the second one, one of the Beagle Bones was replaced by an ESP8266-12F to make the connection with the network through the Wi-Fi module.The assembly diagrams can be seen in figures 11 and 12 respectively.

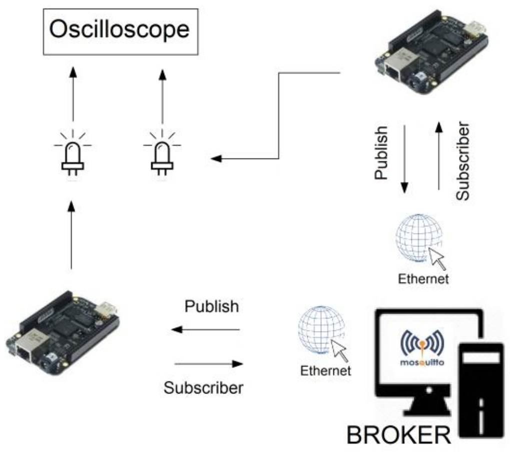

Fig 11: Ethernet – Ethernet Network Test

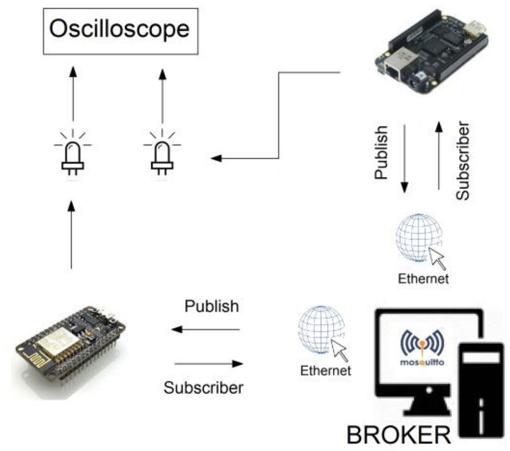

Fig 12: Ethernet – Wi-Fi Network Test

The proposal was to measure the time difference between Ethernet and Wi-Fi networks for data transmission. In the first case, one of the boards worked as a Publisher and the second as a Subscriber. In the second case, the ESP8266 board replaced the Beagle bone Subscriber.

The results showed that, with Ethernet access, the transmission time remained stable. On the other hand, the wireless solution presented higher latency for larger data packets, which is not perceived as a problem for the proposed solution, since the amount of data expected to be sent by the system keeps the latency time constant. The graph of the measured times can be seen in figure 13.

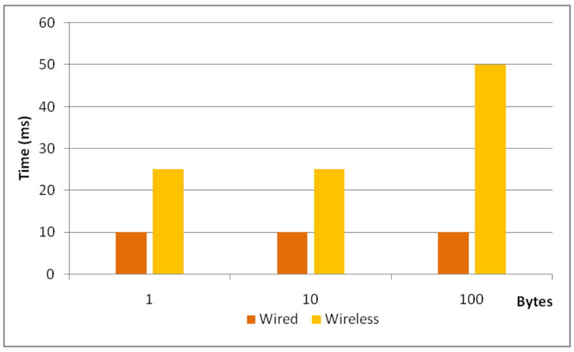

Fig 13: Network performancegraph

6.3 Dashboards

The measurement system was designed to make the data available to users in full time and for this we use the Grafana dashboard [18].In Grafana, it is possible to consolidate data sources and present them graphically and interactively, since the user can access historical information, change graphical forms, build tables, insert visualization panels, etc.

Another format available to view the data was through the IoT MQTT dashboard smartphones application.

To test the solution, charts and tables were generated on the dashboards through data collection during a day of consumption measurement. The values collected can be seen in the figures that follow.

Fig 14: Grafanadashboard

Fig 15: IoT MQTT Dashboard

7. CONCLUSION

The tests carried out for data transmission show that the system can operate at reasonable distances, guaranteeing a large area of coverage. However, special attention must be given to the interference caused by the buildings.Even foreseeing the existence of a data network within the enterprise, improperly installed equipment without a directed line of sight between radio antennas may be a problem for data transmission.

It was possible to identify that the use of a communication gateway integrated in the cloud via MQTT protocol, with its own telemetry measurement system that uses the radio frequency in environments similar toSapiens Park, is technically feasible within the contexts of reduced reading time and agility of access information as well aswithin the flexibility to connect several types of different equipment, since the technology used is not proprietary.

Particular attention should be paid to the availability of broker services, since, in many situations, the connection with the broker was most likely to be difficult due to the use of broker free of cost and without specific contractual guarantees.

8. ACKNOWLEDGMENT

We thank the Federal Institute of Technology of Santa Catarina for technical and financial support, especially the master's degree in Industrial Mechatronics, for the availability of equipment and laboratories. We also thank the financial support of the Foundation for the Research and Innovation of the Santa Catarina State (FAPESC) and CERTI Foundation. Thanks to AQTech Company that technically supported the implementation of this project. Finally, we thank the NERsD Laboratory that support in IoT Project.

9. REFERENCES

[1] FAO – Food and Agriculture Organization of the United Nation. AquastatDatabase.http://www.fao.org/nr/water/aquastat/data/query/index.html?lang=en. Accessed in 01/15/2018.

[2] Turcu, C., Turcu, C. and Gaitan,V. 2012. An Internet of Things Oriented Approach for Water Utility Monitoring and Control. Advances in Computer Science, Praga, 24-26 set. Recent Advances in Computer Enginnering Series. 5. P.175-180.

[3] Nayaka, R. J., Biradar, R. C. 2015. A Survey on wireless network applications in automated public utilities control and management. Journal of Telecommunication and information Technology, Warsaw, n.3, p.13-24.

[4] Lacerda, F. Arquitetura da Informação Pervasiva: Projetos de ecossistemas de Informação da Internet das Coisas. 2015. 226p. Universidade de Brasília.

[5] Živić, N.S., Ur-Rehman, O. and Ruland, C. 2015. Evolution of Smart Metering Systems. 23rd Telecommunications forum TELFOR. Serbia, p.24-26.

[6] Spinsante, S., et al. 2014. Wireless M-Bus Sensor Networks for Smart Grids: Analysis and Results. International Journal of Distributed Sensor Networks. 16 pages.

[7] TI – Texas Instruments User´s Guide. 2016. MSP430x5xx and MSP430x6xx Family. 1189p.

[8] Standards: ICT and communication – Data – EN13757 https://ec.europa.eu/eip/ageing/standards/ict-and-communication/data/en-13757_en. Accessed in11/13/2018.

[9] European Committee for Standardization https://www.cen.eu/Pages/default.aspx Accessed in04/04/2018.

[10] Understanding Wireless M-Bus for the European Smart Meter Market. http://www.epd-ee.eu/print.php?id=9197. Accessed in04/15/2018.

[11] ESP 8266 data sheet. https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf Accessed in 06/06/2017.

[12] Yuan, M. Getting to know MQTT. Why MQTT is one of the best network protocols for the Internet of Things. 2017. https://www.ibm.com/developerworks/library/iot-mqtt-why-good-for-iot/index.html

[13] Hillar, G. C. MQTT Essentials – A lightweight IoT Protocol. 1 ed. Birmingham: Packt, 2017. 263p.

[14] OASIS – Advancing Open Standard for the Information Society. MQTT version 3.3.1. 2014. http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html#_Toc398718043 .

[15] Eclipse Foundation web page http://www.eclipse.org/org Accessed in11/15/2018.

[16] SAPIENS PARQUE. Technical Note nº 02, 2014. EvaluationofConsumptionWaterReservation Volumes ConsideringVariousSources. http://www.sapiensparque.com.br/en/home/

[17] Vision of Sapiens Park by google maps https://www.google.com/maps/@-27.4344032,-48.450146,900m/data=!3m1!1e3. Accessed in11/13/2017.

[18] Grafana Labs Dashboards https://grafana.com/dashboards .Accessed in01/14/2018.