

Instituto Tecnológico y de Estudios Superiores de Occidente

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

Departamento de Electrónica, Sistemas e Informática
Especialidad en Sistemas Embebidos



Considerations in the implementation of an IoT Irrigation System

TRABAJO RECEPCIONAL que para obtener el **DIPLOMA** de
ESPECIALISTA EN SISTEMAS EMBEBIDOS

Presenta: **LIZBETH RUZ ARMAS**

Asesor **JORGE A. PARDIÑAS-MIR**

Tlaquepaque, Jalisco. 27 de septiembre de 2024.

Considerations in the implementation of an IoT Irrigation System

Lizbeth Ruz-Armas
Western Institute of Technology
and Higher Education (ITESO)
Tlaquepaque, Mexico
lizbeth.ruz@iteso.mx

Jorge A. Pardiñas-Mir
Electronic, Systems and
Informatics Department
Western Institute of Technology
and Higher Education (ITESO)
Tlaquepaque, Mexico
jpardinas@iteso.mx

Lizbeth Ruz-Armas

Abstract— The use of electronics in every-day devices has considerably increased over the last few years, allowing the Internet of Things (IoT) to emerge. This results in devices with a wide range of features such as monitoring and controlling the environment of a house or a building. They are based on embedded systems that gather and process information. In this paper, we present the optimal considerations for the implementation of an IoT system in an irrigation application. The application includes Thread network protocol for low power consumption and commonly used standards, and the CoAP communication protocol for increased security and extended battery life. An overview of the types of architectures, protocols and technologies typically used in a IoT system are also described.

Keywords—IoT, communication protocols, network protocols, IPv6.

I. INTRODUCTION

The development of technology creates the need to induce autonomy and include connectivity in devices we use every day. The Internet of Things refers to a type of network where day-to-day devices are connected and have access to the Internet according to standard protocols. This is based on environment detection components that communicate status and control data to achieve recognition, positioning, tracking, monitoring, and handling of information. The main characteristics found in day-to-day devices that make them ideal for an IoT implementation include the need for connectivity, heterogeneity, dynamic operation, scaling, and security [1].

Previous works show the first applications of IoT and their basic architecture using 3 to 5 stack layers [2]. Also, thorough studies have been carried out on the types of protocols and technologies used in IoT applications [1,2]. However, they do not present the integration in an operational system where the architectures, protocols, and technologies are included.

In this work, we discuss the main considerations in the implementation of an IoT in a water irrigation system using a wireless network Thread and CoAP as communication protocol, as well as its advantages and disadvantages. This also

includes an analysis of the protocols and technologies commonly used in wireless networks.

The rest of the paper is organized as follows: Section II describes architectures, protocols, and technologies used in a IoT system. Section III explains the irrigation system prototype. In Section IV the system implementation in an irrigation system is described, and finally, Section V concludes this work.

II. IOT PROTOCOLS AND TECHNOLOGIES

To implement an IoT system, three important aspects need to be addressed: 1) the architecture of the system, 2) the communication network, and 3) the communications protocol.

Regarding the architecture of the IoT, a specific standard has not been defined [1]. Typically, the stack communication model based on the IOS OSI 7-layer is used. A basic architecture consists of 3 layers: the perception or physical layer, transport, and processing or application [1]; the work we present here is based on a similar topology commonly used, known as TCP/IP web Stack. Figure 1 shows a comparison between 3 different architecture models.

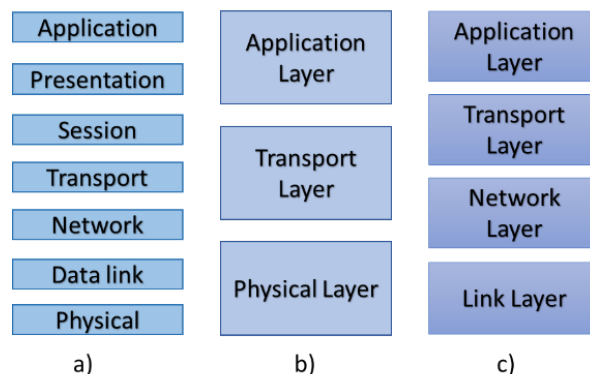


Figure 1. Protocol stacks: a) OSI model, b) 3-layer IoT basic architecture, c) Typical TCP/IP web stack

Given that communication needs are different for a house, a building, or a city, the right option in terms of power consumption, bandwidth, and range may be determined

following the specifications of the IoT that will be implemented [3].

Next, the options for an IoT system are briefly described, which can be divided in 3 categories depending on their operational range. Figure 2 shows the scheme that describes long-to-short range protocols for communication coverage.

LTE-CAT M1	NB-IoT	Wi-Fi	Thread	Bluetooth
LoRaWAN	Sigfox	ZigBee		RFID
LONG RANGE NETWORK		MEDIUM RANGE NETWORK		SHORT RANGE NETWORK

Figure 2. Range network protocols.

The description of these protocols is shown next.

A. Long range networks:

- NB-IoT: is a narrowband IoT that offers very low power consumption and may use the existing network infrastructure, which makes it a very low-cost implementation.
- Sigfox: it uses Ultra Narrowband (UNB), which allows connectivity of many devices using small quantities of information exchange, which results in low power consumption. It is frequently used in M2M applications.
- LTE-CAT M1: it may connect devices with data requirements in a medium range and has become a standard in low-power wide-area implementations.
- LoRaWAN: it was created to optimize low-power consumption in Wide-Area Networks (WAN), and it may handle millions of devices with low cost, i.e., M2M, industrial applications, and smart cities.

B. Medium range networks:

- Zigbee: it is used in communications for industrial machinery, home appliances, and traffic systems. It uses mesh, star, or tree topologies, and is characterized by its low cost, low power consumption, and reliability.
- Thread: it uses IPv6, which can handle 128-bit IP addresses. It is designed for IoT systems implemented in home appliances.

C. Short range networks:

- Bluetooth: it is typically used in wireless devices such as headphones and speakers, although, it is commonly found in laptops, smartphones, and tablets.
- RFID: stands for Radio Frequency Identification and was introduced in the first IoT implementations. It is commonly used to locate the position of a given object in a close-range area, such as a house or a building.

Other aspects to be considered in the implementation are the communication protocols. There is a wide variety of them used in IoT devices; this is due to the different types of information that can be handled and monitored in different environments

and implementations [4], [5]. Next, we briefly describe some of the most common protocols:

- MQTT: it stands for Message Queuing Telemetry Transport, and it is based on Client/Server message transportation. It operates over TCP/IP or other protocols that support bidirectional communication. It is ideal for IoT applications where small quantities of information are transmitted and a high bandwidth is not required.
- CoAP: it stands for Constrained Application Protocol; it was created to provide compatibility with HTTP but with smaller information packets. It is a client/server protocol which uses User Datagram Protocol (UDP) instead of TCP, which, not only does it simplify the header of the package, but it also reduces the size of every requirement. At the same time, CoAP uses Acknowledge (ACK) responses, CONFIRMABLE which is a confirmation request, and a NONCONFIRMABLE, which is a non-confirmation of messages.
- XMPP: Extensible Messaging and Presence Protocol, it was created initially for Instant Message (IM), based on the Extensible Markup Language (XML). This protocol allows real time exchange of data. Since its creation it has been used in communication protocols, thus, over time a specific version was created for IoT, denoted as XMPP-IoT.

III. THREAD PROTOCOL

We started the implementation by selecting Thread as the network protocol, which already operates with IP addresses and has the advantage of easily connect to a node in a Thread network, has at least 250 nodes to operate, and it has low power consumption, it may even operate in battery-based devices. It can communicate with other long-range devices, making it ideal for home and buildings.

Thread has a mesh type of network structure, which is a node-interconnected setting very similar to that of a Wide Area Network (WAN). It offers good coverage, and it can be scaled up to hundreds of devices without failure spots. Information can travel through different paths so that, if a node fails, information can still be transferred seamlessly among nodes [6].

Figure 3 shows the architecture of a Thread network and its parts are described as follows: the Router Eligible End Device (REED) is a device that may operate as a router in case no other router is present. It may also operate as an End Device (ED), namely, as a sensor, which would need to communicate through a router. The Thread Router manages the traffic through the devices in the network. The Leader defines the network parameters, coordinates commissioners, and routes traffic among devices. The Border Router (BR) provides a bridge between a Thread network and home LAN or other IP infrastructure such as Wi-fi, Ethernet and cellular.

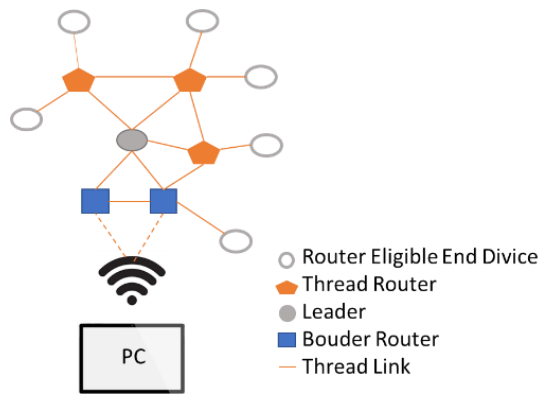


Figure 3. Thread network architecture.

In the implementation we used CoAP as the communication protocol, and Thread as the network protocol, which typically has a 5-layer architecture. The integration of the system is shown in Figure 4.

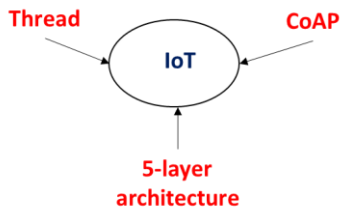


Figure 4. IoT system using Thread and CoAP with 5-layer architecture.

Thread wireless network protocol is based on IP and it was specifically designed to communicate IoT devices. It consists of several open-source protocols and standards [7], and its architecture is described in Figure 5.

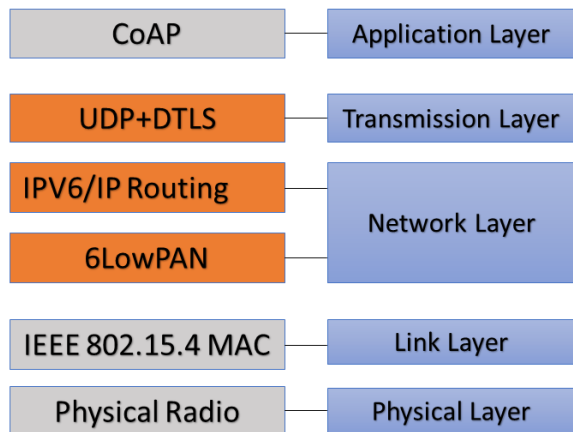


Figure 5. Thread and TCP/IP stack comparison.

Thread is a stack of protocols; in the lower layers it uses the IEEE 802.15.4 standard with MAC security. It was developed for portable devices powered by a battery, and thus, it only requires sending sporadic and short “flashes” of information.

This makes it more efficient due to a lower power consumption compared to RF communication used in WLAN and Bluetooth.

The transmission layer in Thread consists of IPv6, 6LoWPAN, and UDP (User Datagram Protocol) standards. When the standard 802.15.4 is used in the link layer, it allows packet lengths of only 127 bytes, whereas the size in IPv6 is 1280 bytes [8]. Therefore, an additional layer is needed to make the size conversion. To achieve this, an intermediate layer is included, known as 6LoWPAN, which is a header compression of IPv6 packets and it can solve this problem by separating the packets in the transmitter [7].

The UDP allows data transmission without a previous connection, which makes it possible to send information at high speed as it does not require a connection confirmation, nor does it wait for acknowledgement of the packets being received. For this reason, the delivery of packets is not guaranteed. To compensate this, the Datagram Transport Layer Security (DTLS) is included, which is a communication protocol designed to protect the privacy of the information, prevent its interception, and avoid its manipulation.

IV. IOT IRRIGATION SYSTEM IMPLEMENTATION

The system has a set of sprinklers to control the irrigation and a set of sensors to control temperature and humidity. The latter are monitored to determine the activation of the irrigation based upon predefined specifications.

The irrigation system consists of 3 parts: the irrigation control, the command console, and the sensor control center, as shown in Figure 6.

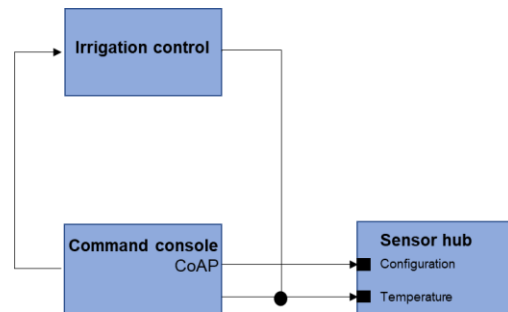


Figure 6. Block diagram of the irrigation system

The irrigation system has a Thread connection, which is used to control a mechanic actuator that may operate in automatic or manual mode. In automatic mode, temperature may trigger the operation of the sprinkler. It may use Thread to read the temperature from the sensor Hub. It is connected to ethernet where it is power supplied with POE (Power Over Ethernet) and is connected to the house’s network. The irrigation control in a Thread configuration is coined Border Router.

In the sensor hub, the sampling frequency may be configured to measure the temperature variable through the Thread connection. It operates as a Router Eligible End Device.

The console is connected to the house's network through ethernet. It may connect to the Thread network to configure the parameter of the irrigation control and then read the temperature.

Figure 7 shows the implementation of the irrigation system based on a Thread network architecture. Here, we used two microcontroller units FRDM64f and two wireless transceiver boards FRDM CR20A that were plugged into the platform. One router was connected to the internet, and one computer. The system will be described in the next section.

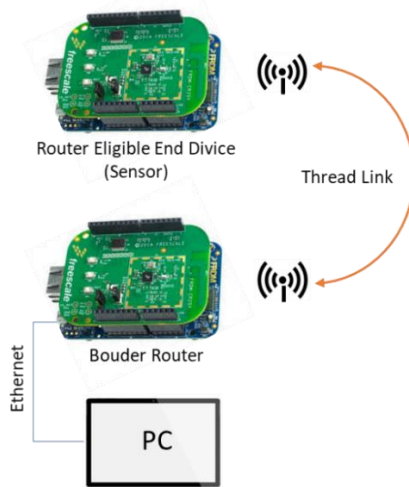


Figure 7. Irrigation Thread network.

Two Putty terminals were used, one for the irrigation control (Border Router) and the other for the sensors (Router Eligible End Device). A Thread network was created by means of its Unique Local Addresses (ULA), as shown in Figure 8.

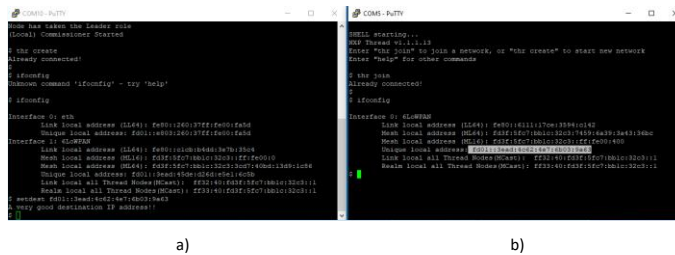


Figure 8. Irrigation terminals. a) Border router. b) Router Eligible End Device.

The following addresses were used in the CoAP commands following the next format:

`python coapclient.py -operation GET -p coap://[fd01::3ead:b40e:88c4:e112:8504]:5683/temp -P ""`
 An example for temperature measurement is shown in Figure 9.

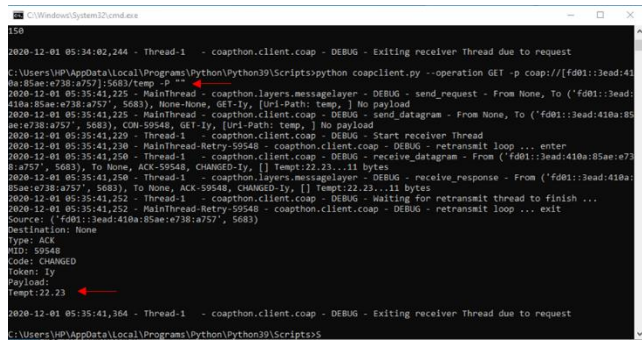


Figure 9. Temperature reading in the irrigation system.

The system may be configured to set the sampling frequency as shown in Figure 10. Also displays the state of the sprinklers as in Figure 11.

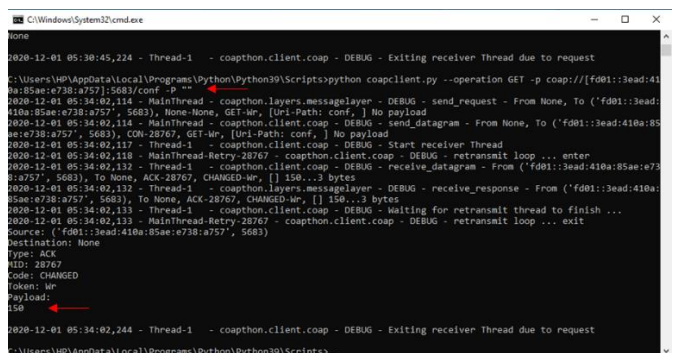


Figure 10. Set configuration at 150 s.

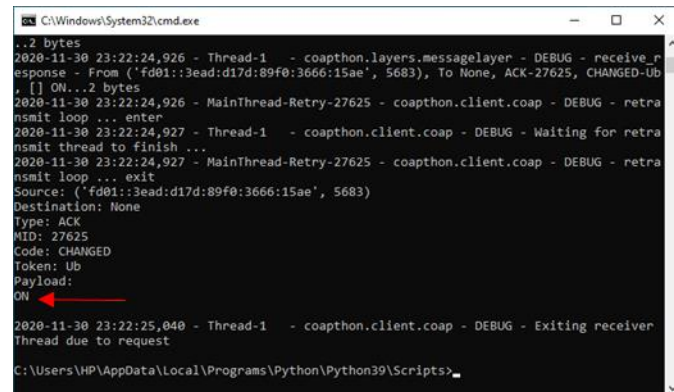


Figure 11. Checking the sprinkles state.

Figure 12 shows the sequence of turning the sprinklers off and on manually, and a photograph of the operation.

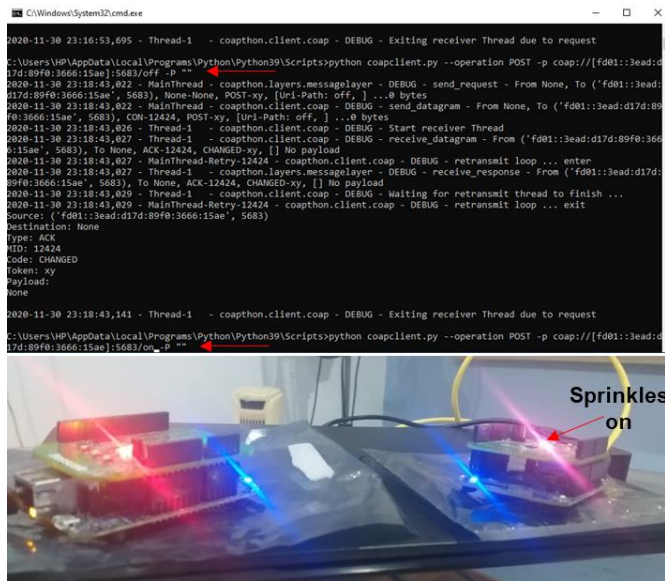


Figure 12. Turning the sprinklers off and on manually

This prototype allows the validation of proposed structure under the considered specifications. Based on these results, an operational prototype may be implemented in a real application.

V. CONCLUSION

We presented the most important considerations to implement an IoT system consisting of the type of application and the operational range, which determine the power consumption and the bandwidth. The architecture, the network and the communication protocols are defined based on these parameters.

It was seen that due to the wide diversity of IoT applications and the continuous development of technology and protocols, it is not possible to define a general structure to create a system. However, a very specific system may be implemented for a particular application, and it may be designed as robust as the customer's needs.

REFERENCES

- [1] J. D. Alvarado Moreno, L. C. Luis Garcia, W. C. Hernandez, and A. M. Barrera Obando, "Embedded Systems for Internet of Things (IoT) Applications: A Review Study," *2018 Congreso Internacional de Innovación y Tendencias en Ingeniería (CONITI)*, Oct. 2018, pp. 1–6. doi: 10.1109/CONITI.2018.8587092.
- [2] W. Rzepecki, Ł. Iwanecki, and P. Ryba, "IEEE 802.15.4 Thread Mesh Network – Data Transmission in Harsh Environment," *2018 6th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*, Aug. 2018, pp. 42–47. doi: 10.1109/W-FiCloud.2018.00013.
- [3] F. Samie, L. Bauer, and J. Henkel, "IoT technologies for embedded computing: A survey," *2016 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, Oct. 2016, pp. 1–10.

- [4] N. Nikolov, "Research of MQTT, CoAP, HTTP and XMPP IoT Communication protocols for Embedded Systems," *2020 XXIX International Scientific Conference Electronics (ET)*, Sep. 2020, pp. 1–4. doi: 10.1109/ET50336.2020.9238208.
- [5] N. Nikolov, "Research of the Communication Protocols between the IoT Embedded System and the Cloud Structure," *2018 IEEE XXVII International Scientific Conference Electronics - ET*, Sep. 2018, pp. 1–4. doi: 10.1109/ET.2018.8549604.
- [6] W. Rzepecki, Ł. Iwanecki and P. Ryba, "IEEE 802.15.4 Thread Mesh Network – Data Transmission in Harsh Environment," *2018 6th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*, 2018, pp. 42–47, doi: 10.1109/W-FiCloud.2018.00013.
- [7] Baccelli, Emmanuel & Hahm, Oliver & Wählich, Matthias & Günes, Mesut & Schmidt, Thomas. (2012). *RIOT: One OS to Rule Them All in the IoT*.
- [8] S. Deering, R. Hinden, RFC2460 – "Internet Protocol, Version 6 (IPv6) Specification", Dec 1998