A Fog Computing Application Using Knowledge-Based Collaborative Sensors - Noise Annoyance Monitoring and Control of a Sun Tracker Use Cases

Joaquin Canada-Bago, Jose-Angel Fernandez-Prieto
Department of Telecommunication Engineering
University of Jaen
Linares, Spain
e-mails: jcbago@ujaen.es, jan@ujaen.es

Ulrich Birkel
Department of Electrical Engineering and Information Technology
Technische Hochschule Mittelhessen
Gießen, Germany
e-mail: ulrich.birkel@ei.thm.de

***Abstract*—Message Queuing Telemetry Transport (MQTT) and Constrained Application Protocol (CoAP) protocols are widely used allowing communication between Internet of Things (IoT) devices and platforms as well as device-to-device communication. However, when they are used in real applications based on Cloud Computing, different problems are observed, such as data loss, lack of security or long communication times between sensors. In this sense, Fog Computing can improve the performance of these applications. The objective of this work is to propose an application based on Fog Computing using knowledge-based devices for two real scenarios: a) control of a solar tracker and b) noise annoyance monitoring. Several experiments have been carried out in order to verify if the application and MQTT and CoAP protocols are appropriate in the system communications of both use cases. The results show that, in the case of noise annoyance monitoring application, this architecture allows reducing the errors in a satisfactory way. However, in control applications, where a communication time between sensors of less than 10 ms is required, the use of these protocols may not be adequate. For this case, an additional client-server software to the Fog Computing system has been developed to be executed in the IoT devices. Although it has lower performance than the widely used protocols, it allows the transmission time to be reduced, and can be satisfactorily applied to the control of systems, such as the control of a sun tracker.**

Keywords; Knowledge-based sensors; fog computing.

I. INTRODUCTION

The Internet of Things (IoT) [1] [2] concept was introduced by Kevin Ashton in 1999 as a system where the physical world is connected to the Internet through ubiquitous sensors. Nowadays, IoT refers to obtaining data and acting in the real world by means of devices with information processing capacity, communications that allow the storage of data on servers located in the cloud and subsequent analysis of stored data.

IoT typical devices have constrained-resources, sensor and actuator capacity, local information process, and are able to communicate data with servers on the Internet cloud. A first classification divides them into devices without an operating system (e.g., Arduino, WaspMote) or with it (e.g., Raspberry).

Regarding IoT communications, data networks (e.g., IEEE 802.15.4, ZigBee, Sigfox, Wireless Sensor Networks (WSN) [3]), network protocols (e.g., LoRa, LoRaWAN [4]), and application protocols (e.g., Message Queuing Telemetry Transport (MQTT) [5], Constrained Application Protocol (CoAP) [6]) have been specifically designed. Although MQTT is classified as a Machine-to-Machine (M2M) protocol, it uses an intermediate server (broker), which introduces a delay that could not be tolerable in distributed control applications.

Currently, the development of IoT is causing that a large number of devices provides huge amounts of data, which has to be stored in servers located in the cloud. Cloud Computing [7] [8] is a technology that allows large-scale computing with the following advantages: virtualized resources, parallel processing, service integration and data storage. In this context, cloud computing systems such as OpenStack [9] and server virtualization environments such as Proxmox Virtual Environment [10] have been proposed. These cloud servers (also called platforms) provide developers Application Programming Interfaces (APIs) and Software Development Kits (SDKs) so that it is possible to establish communication between the IoT devices and the cloud platform. The protocols most commonly used by the platforms are MQTT and HTTP.

Fog Computing [11][12] is a new paradigm, which is based on the transfer of computer and network services from the cloud to the Internet periphery. In this way, IoT devices will use fog servers located very close to them. After that, data are transmitted to cloud servers by fog servers. Some benefits of fog computing are the following:

- Distributed data storage on fog servers.
- Hierarchical data processing in fog servers.
- Quality of Service (QoS) in the data, in order to prioritize the sending of data from delay sensitive applications.
- Performing complex tasks on servers in the fog.
- Uninterrupted services, intermittent access to the cloud would not affect the application.
- Latency reduction since, on the one hand, communications between devices in the fog are faster and, on the other, the volume of data to be sent to the cloud is smaller.

The use of IoT smart devices is widely referenced in the European Commission documents relating to IoT [13] [14] [15]. These documents present devices (smart things) in which algorithms can be executed for intelligent decision based on real-time measurements.

Fuzzy Rule-Based Systems (FRBS) [16] are based on the use of Fuzzy Logic (FL) [17] and express the knowledge through a set of linguistic rules, which are grouped into a Knowledge Base (KB). These systems are correctly adapted to problems in which there is a high degree of uncertainty and imprecision and can be applied to control problems (Fuzzy Logic Controller (FLC)) in which the control algorithm is expressed as rules of action.

A collaborative approach based on FRBS for constrained resource devices is shown in [18][19]. In this approach, each device is able to share its data (e.g., the value of a local variable) with another sensor, with a group of sensors or with all the elements of the system. Therefore, it is conceivable that a device may have local and remote data: local data obtained directly by the device and remote data obtained by other devices and subsequently transmitted to the given device.

The objective of this work is to propose a fog computing collaborative application for the control of a system (sun tracker) and for the noise annoyance monitoring, as well as verify that MQTT and CoAP protocols can be used in the collaborative system communications.

The rest of this paper is organized as follows. Section II describes the fog computing application. Experimental results are provided in Section III. Some conclusions and future work are presented in Section IV. The acknowledgement closes the article.

II. FOG COMPUTING APPLICATION

Fig. 1 shows the proposed system, which is composed of the following components: FRBS collaborative sensors or actuators, direct communication between sensors, communication between sensors and fog computing server, fog computing server, communication between fog and cloud servers and a cloud server.

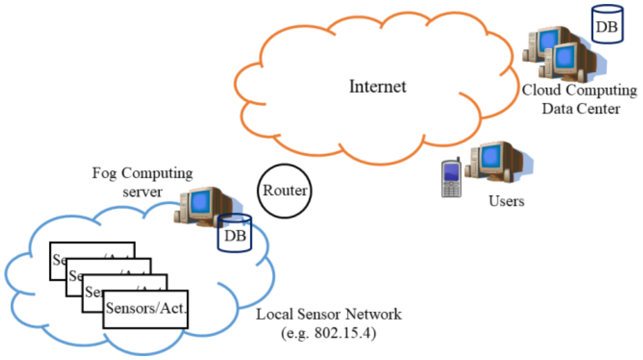

Figure 1. System proposed.

The sensors and actuators used (Fig. 2) are based on a FRBS system and a communications module. The FRBS system allows the sensor to infer the output using a knowledge base.

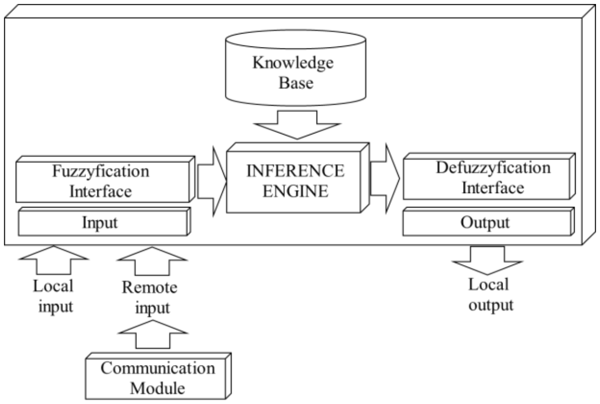

Figure 2. System proposed.

The knowledge-based sensor is based on the structure of the Mandani [20] FRBS, which consists of the following components: a fuzzification interface, a KB, an inference engine and a defuzzification interface.

The fuzzification interface transfers the value of the inputs variable to the fuzzy system. The KB contains the definition of the controller input and output variables, the fuzzy sets defined in the variables, and a set of IF-THEN rules that relate these variables. The inferences engine is responsible for inferring the fuzzy output of the system from the input variables and the knowledge base. Finally, the defuzzification interface adapts the value of the fuzzy output to a real output value.

On the other hand, the communications module allows the sensor to use remote variables obtained by other sensors. Two types of communications are used in this application: direct communication between sensors (Fig. 3) and communication through the fog computing server. Direct communication between sensors allows the sensors to request and obtain the value of a variable obtained by a remote sensor in the fastest way. In this sense, each sensor has a small server that allows the collaborative approach.

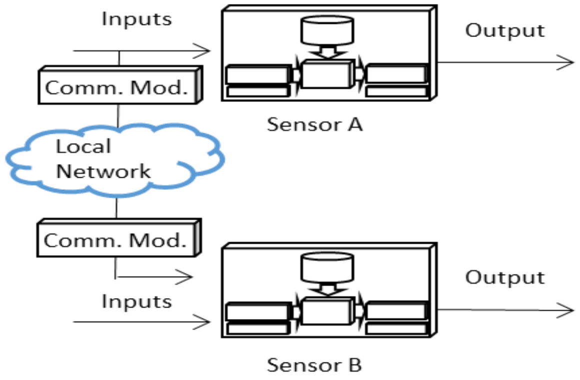

Figure 3. Sensor direct communication.

The main functions of the fog server are: a) local storage of data obtained by all the sensors, b) data communication to sensors, c) data communication to the cloud platform, and d) increased security of the sensor network. In this way, the measurements and inferences made by the sensors are sent to the fog server and stored in a database. Sensors that need these values can request them from the fog server, although the procedure is slower. Finally, a cloud platform is used for data storage and for Internet users to view and analyze the data obtained.

Note that although the fog computing platform could perform preliminary data analysis processes, in this work, it has not been developed.

III. RESULTS

Several experiments have been carried out in order to verify that the application can be used in both use cases. In the case of the control of the solar tracker, which have the following requirements: a) controllers reaction time must be less than 10 ms; b) five sensors (of which two must be collaborative FRBS controllers) are necessary; c) knowledge bases of up to five variables and d) 15 rules of action per KB.

Fig. 4 shows the test bed installed for the experiments. It consists of a set of Arduino DUE devices with Ethernet shield, an IEEE 802.3 local area network based on a QoS switch, a fog server based on a Raspberry Pi 3B, MySQL database and a cloud platform with HTTP protocol. The software of the Collaborative FRBS sensors, TCP server in sensors, fog server and communications have been developed in C language.

First, the computation time of the tracker control based on a FRBS has been calculated. Using a test KB composed of 5 variables and 15 rules, the microcontroller computes more than 500 inferences per second, which is equivalent to a reaction time of 2 ms. The KB has been modeled through expert knowledge. One of our previous works [21] provides more details about the FRBS system and some inference examples can be observed.

Figure 4. Photograph test bed.

Secondly, the communication time of variables between sensors has been measured using the MQTT and CoAP protocols. Regarding MQTT, the “Arduino Client for MQTT” library and the Mosquitto broker on Raspberry PI have been used. Several tests have been carried out with five devices as QoS1 clients of the broker transmitting simultaneously in two topics. The shortest communication time between sensors has been 67ms. In case of CoAP protocol, the CoAP-simple-library has been used. When performing the experiment under similar conditions to the previous one, a communication time of 30 ms was obtained.

Subsequently, various experiments have been performed using a small TCP server on each device to carry out direct communications between sensors. The server has less functionality than the MQTT and CoAP protocols, allowing exclusively the communication of variables between sensors. In this case, the measured communication times are 5 ms.

Regarding the use of the fog server for the communication of data between sensors, the measured communication times have been much longer due mainly to access to the database.

In the case of the noise annoyance monitoring, in a previous work [22] we presented the design and implementation of a complete low-cost system, composed of nine sensors nodes, for a Wireless Acoustic Sensor Network (WASN) deployed in the city of Linares (Jaén), Spain. The network topology for the proposed complete system is shown in Fig. 5.

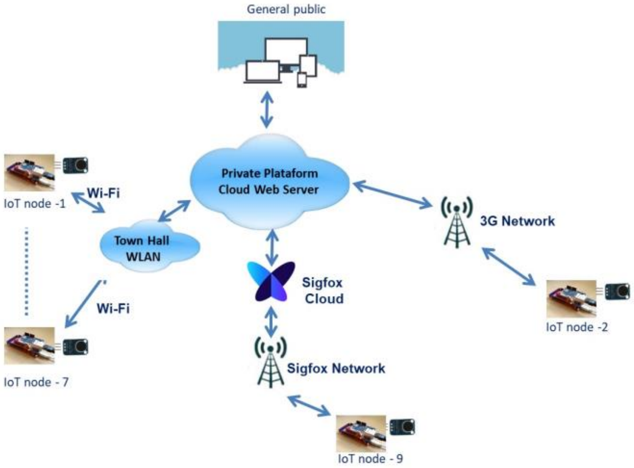

Figure 5. Network topology used for the WASN deployed in the city.

In this real system, a cloud system was used and several problems were encountered, such as: data loss due to the lack of a connection to the cloud, processing of a large volume of data on the server, inability to collect data at certain locations, and inability to use complex knowledge-based systems. To solve these kinds of problems, we have designed and incorporated a fog computing platform between the sensor nodes and the cloud. The fog servers are only in charge of sensed data store and retransmission to the cloud server. A previous work [23] provides more details on how the KB and the FRBS are implemented in the sensors. The final network topology used is shown in Fig. 6.

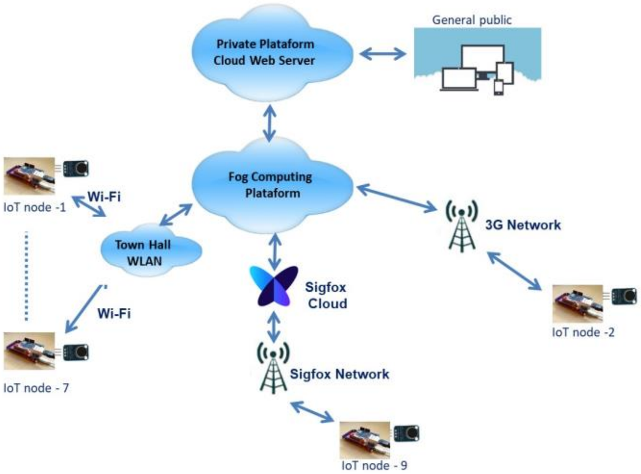

Figure 6. Network topology incorporating the fog server in the WASN deployed in the city.

The fog server has stored the data received from the sensors and retransmitted it to the cloud server. Various tests have been performed with intermittent internet access. In this sense, when recovering communications over the Internet, the data are retransmitted without any loss.

IV. CONCLUSIONS

The device used has sufficient processing capacity to infer the output values using a typical KB to control a sun tracker. The use of the implementations of MQTT protocols and CoAP exceed the maximum reaction time (10 ms) for sun tracker control. Nevertheless, the reaction time is shorter than the maximum by means of the small TCP server and direct communications between sensors.

In the case of the noise annoyance monitoring, since the sensor nodes calculate the sound pressure level every second, and every 30s they send the noise annoyance to the fog server, communication times are not critical in this scenario.

With regard to future work, the following actions are proposed: a) implement the application for tracker control, b) perform complex tasks (which IoT devices cannot execute) on servers in the fog, which increases the capabilities of these applications.

ACKNOWLEDGMENT

This work was partially funded under the Programa Operativo FEDER-Consejería de Economía y Conocimiento de la Junta de Andalucía, Spain, Project Ref. 1380677.

REFERENCES

[1] S. Madakam, R. Ramaswamy, and S. Tripathi, “Internet of Things (IoT): A Literature Review” Journal of Computer and Communicatios, vol. 3, pp. 164-173, 2015.

[2] R. Buyya and A. Dastjerdi, Internet of Thing. Principles and Paradigms. Elsevier, 2016.

[3] H. Karl and A. Willing, Protocols and Architectures for Wireless Sensor Networks. Chichester, West Sussex: Willey, 2007.

[4] Lora Aliance. [Online]. Available from: https://lora-alliance.org/ 2022.08.29

[5] MQTT. [Online]. Available from: http://mqtt.org/ 2022.08.29

[6] IETF. *The Constrained Application Protocol (CoAP)*. [Online]. Available from: https://tools.ietf.org/html/rfc7252 2022.08.29

[7] W. Stallings, Foundations of Modern Networking: SDN, NFV; QoE, IoT and Cloud. Addison-Wesley, 2015.

[8] I. Targio Hashem et al., “The rise of big data on cloud computing: review and open research issues," Information Systems, vol. 47, pp. 98-115, 2015, doi: 10.1016/j.is.2014.07.006.

[9] Openstack. [Online]. Available from: https://www.openstack.org/ 2022.08.29

[10] Proxmox. [Online]. Available from: https://www.proxmox.com/en/ 2022.08.29

[11] H. Freeman and T. Zhang, “The Emerging Era of Fog Computing and Networking,” IEEE Communications Magazine, vol. 4, pp. 4-5, June 2016, doi: 10.1109/MCOM.2016.7497757.

[12] Z. Hao, E. Novak, S. Yi, and Q. Li, “Challenges and Software Architecture for Fog Computing,” IEEE Internet Computing, vol. 21, pp. 44-53, Apr. 2017, doi: 10.1109/MIC.2017.26.

[13] O. Vermesan and P. Friess, Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems. River Publishers, 2013.

[14] O. Vermesan et al. *Internet of Things. Strategic Research Roadmap*. [Online]. Available from: http://www.internet-of-things-research.eu/pdf/IoT_Cluster_Strategic_Research_Agenda_2009.pdf 2022.08.29

[15] CERP-IoT. *Vision and Challenges for Realising the Internet of Things*. [Online]. Available from: http://www.internet-of-things-research.eu/pdf/IoT_Clusterbook_March_2010.pdf 2022.08.29

[16] O. Cordón, F. Herrera, F. Hoffmann, and L. Magdalena, Genetic Fuzzy Systems: Evolutionary Tunning and Learning of Fuzzy Knowledge Bases, Advances in Fuzzy Systems — Applications and Theory: Volume 19. World Scientific, 2001.

[17] A. Zadeh, “Fuzzy Sets,” Inf. Control, no. 8, pp. 338-353, 1965.

[18] J. Canada-Bago, J. A. Fernandez-Prieto, M. Gadeo-Martos, and J. Velasco, “A New Collaborative Knowledge-Based Approach for Wireless Sensor Networks,” Sensors, vol. 10, no. 6, pp. 6044-6062, 2010, doi:10.3390/s100606044.

[19] M. Gadeo-Martos, J. Fernandez-Prieto, J. Canada-Bago, and J. Velasco, “An Architecture for Performance Optimization in a Collaborative Knowledge-Based Approach for Wireless Sensor Networks," Sensors, vol. 11, pp. 9136-9159, 2011, doi:10.3390/s111009136.

[20] E. Mamdani and S. Assilian, “An experiment in linguistic synthesis with a fuzzy logic controller,” International Journal of Man-Machine Studies, vol. 7, pp. 1–13, 1975, doi:10.1016/S0020-7373(75)80002-2.

[21] J. Canada-Bago, J. A. Fernandez-Prieto, M. A. Gadeo-Martos, and P. Perez-Higueras, “Knowledge-Based Sensors for Controlling A High-Concentration Photovoltaic Tracker,” Sensors, no. 5: 1315, Feb 2020, doi:10.3390/s20051315.

[22] J. A. Fernandez-Prieto, J. Canada-Bago, and M. A. Gadeo-Martos, “Wireless Acoustic Sensor Nodes for Noise Monitoring in the City of Linares (Jaén),” Sensors, no. 1: 124, Dec 2019, doi:10.3390/s20010124.

[23] J. A. Mariscal-Ramirez, J. A. Fernandez-Prieto, M. A. Gadeo-Martos, and J. Canada-Bago, “Knowledge-based wireless sensors using sound pressure level for noise pollution monitoring,” 11th International Conference on Intelligent Systems Design and Applications, IEEE Press, Nov. 2011, pp. 1032-1037, doi:.10.1109/ISDA.2011.6121794.