Available online at www.sciencedirect.com

ScienceDirect

IFAC-PapersOnLine 49-6 (2016) 150–155

Communication with resource-constrained devices through MQTT for control education

Miguel A. Prada, Perfecto Reguera, Serafín Alonso, Antonio Morán, Juan J. Fuertes, Manuel Domínguez

SUPPRESS research group, University of León, Escuela de Ingenierías - Campus de Vegazana, León, 24007, Spain. (e-mail: ***ma.prada, prega, saloc, a.moran, jj.fuertes, manuel.dominguez@unileon.es****).*

Abstract: The developments on machine-to-machine systems are interesting for control education, not only for the opportunities to apply control and automation solutions to new problems, but also for the availability of new hardware, software and communication platforms. These technologies facilitate a low-cost and easier integration of physical equipment in educational tools such as the remote laboratories. This paper proposes the use of a lightweight protocol for communication with resource-constrained devices, MQTT, as an aid to integrate new devices in educational applications, specifically in those that use web standards such as Javascript to provide interactive user interfaces. To evaluate this approach, an educational application focused on the control of a DC motor position loop, built with EjsS, was developed. This tool uses the MQTT protocol to parametrize and communicate with an Arduino microcontroller that, in turn, controls a physical setup implemented with the Feedback MS150 modular system. The proposed approach enables the easy connection of interactive educational tools to new real equipment, especially those driven by resource-constrained devices.

Keywords: Virtual and remote labs, education, PID, protocols, resource-constrained devices, MQTT, Arduino, EJsS.

1. INTRODUCTION

In the last years, there has been a growing integration of computational devices in physical elements to enable sensing and control, outside of the areas where automation has traditionally been applied. The developments on embedded systems, communication technologies and Internet protocols make it possible to build machine-to-machine (M2M) systems in any application area. Indeed, the growth of physical entities connected to the Internet is expected to be steep (212 billion entities and the 45% of the Internet traffic in the early 2020s) and so is the economic growth related to these services (Al-Fuqaha et al., 2015). For that reason, the research on Internet of Things (IoT) and cyber-physical systems is very active, especially on the topics of communication technologies and data analytics.

This field is interesting for control education, not only for the opportunities to apply control and automation solutions to new problems, but also for the availability of new hardware, software and communication platforms. These technologies facilitate a low-cost and easier integration of physical equipment in educational tools such as the remote laboratories.

It is well known that the observation of the actual behavior and response of physical phenomena is essential for learning technical disciplines. However, the availability and cost of equipment may hinder the setup of hands-on experiences in higher education. The interest on overcoming those difficulties through sharing and scheduling, as well as the ability to provide additional resources and interactivity that enable an easier interpretation of the studied processes, has promoted the development of many internet-based educational tools, especially remote laboratories (Dormido, 2004; Vargas et al., 2011; Santana et al., 2010). Nevertheless, these tools have associated their own problems, being the main one the effort required to create, manage and maintain the experimental setups (Salzmann and Gillet, 2007). The proposed approaches to alleviate the required management effort are focused on the standardization of software components and data acquisition technologies. In this sense, the use of certain tools has been widely adopted: learning/content management systems (Prada et al., 2015; Guinaldo et al., 2013), authoring tools for the rapid development of educational interactive applications such as Easy Java/Javascript Simulations (EjsS) (Esquembre, 2004) or communication standards such as OPC for the integration of industrial-level equipment (Domínguez et al., 2014).

As well as the OPC intercommunication standard is widespread in industrial environments and representational state transfer (REST) web services are widely used to expose resources to the Internet, some application-layer protocols have been proposed for the communication of embedded devices among themselves and with applications

2405-8963 © 2016, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2016.07.169

and middleware. The common features of these protocols is that they are lightweight enough to be used in resource-constrained devices such as those that are powered with batteries or use unreliable low-bandwidth links. They also provide mechanisms for asynchronous communication and are focused on the concept of resource. Among those application-layer protocols, two open standards, MQTT (message queue telemetry transport) and CoAP (constrained application protocol), are being increasingly adopted. Therefore, they are suitable for the communication with readily available *smart sensors* but also with low-cost microcontrollers and computers, such as Arduino and Raspberry. These low-cost hardware platforms have received considerable attention from the control education community and are used as controllers or data acquisition devices in many projects (Valera et al., 2014; Bermúdez-Ortega et al., 2015). Nevertheless, the lower range of these platforms (i.e., the microcontrollers) have been found to be too resource-scarce to implement common strategies for data exchange in educational settings (Reguera et al., 2015). For that reason, it seems to be appropriate to use instead the technologies for lightweight communication.

The aim of this paper is twofold. On one hand, we propose the use of one of the dominant lightweight protocols for communication with resource-constrained devices as an aid to integrate low-cost hardware platforms, wireless sensor networks or any other IoT devices in educational applications. On the other hand, we propose an approach to integrate the aforementioned communication in interactive user interfaces that use web standards such as Javascript for multi-platform compatibility. For that purpose, the paper is organized as follows. Section 2 discusses the currently available lightweight protocols and their suitability. Section 3 discusses the alternatives for the integration of the stated objectives using an existing physical system as example. Section 4 presents the experiment prepared with the selected approach. Finally, section 5 discusses the conclusions and future work.

2. PROTOCOLS FOR RESOURCE-CONSTRAINED DEVICES

The application-level protocols are one of the most important building blocks to achieve the requirements expected in device-to-device environments. Most of those requirements (scalability, performance,...) are shared with the traditional Internet protocols such as HTTP, but there are some differences. First of all, the protocols suitable for M2M communication must be lightweight, to allow battery-operated devices to communicate with low power consumption. Since the resource-constrained devices might need to use unreliable low-bandwidth links, there is also stress on reliability. Due to the requirements of mobility and interoperability, the proposed protocols can sometimes work on different protocol stacks (i.e., over different physical and link layers), although the standard setting is generally TCP/IP over IEEE 802.15.4 networks. Finally, most of the proposed protocols are centered on the concept of *resource* (although it might receive other names such as *topic*), which is addressable and observable.

Among the open standards, the existing dominant application level protocols are MQTT (message queue telemetry transport) and CoAP (constrained application protocol). For that reason, this section will focus in the features of those protocols. There are, however, other choices such as XMPP (extensible messaging and presence protocol), AMQP (advanced message queuing protocol) or DDS (data distribution service) (Al-Fuqaha et al., 2015).

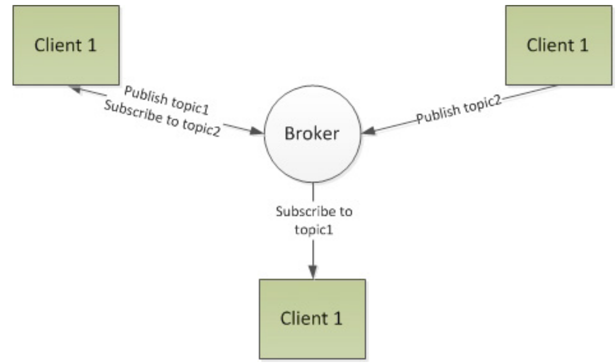

Fig. 1. MQTT publisher-subscriber architecture.

The **Message Queue Telemetry Transport** (MQTT) is a publish/subscribe messaging protocol, originally designed by Andy Stanford-Clark and Arlen Nipper, that is currently an OASIS (Organization for the Advancement of Structured Information Standards) standard (Banks and Gupta, 2014). Its operation is based on a client/server model where a central server, known as broker, receives messages from the clients, which are essentially all the nodes involved in the communication. Those messages can be topic publications or subscriptions. This structure enables many-to-many communication and decouples producer and consumer (see Figure 1). Its resulting flexibility and simplicity facilitates the connection of embedded devices to middleware and applications.

Although any device can be publisher and subscriber, a typical MQTT architecture might include a set of sensors that periodically publish their measurements (payload) to a topic address. Any interested device registers as a subscriber of any specific topic, in order to receive a message from the broker every time the topic is updated. The MQTT topics are hierarchical (e.g., $building/leftwing/temperature$) and wildcards are allowed when registering a subscription.

The MQTT packets have a very low overhead (starting from two bytes) and are generally sent over TCP, although UDP can be used with the MQTT-SN specification (intended specifically for sensor networks). The TCP/IP ports 1883 and 8883 are reserved for this protocol. It allows three levels of quality of service (at most once, at least once, and exactly once), admits authentication through username and password and encryption with TLS/SSL.

The specification also considers the transport of MQTT over a Websocket connection, stating that MQTT control packets must be sent in WebSocket binary data frames although a single data frame can contain any number of control packets.

The **Constrained Application Protocol** (CoAP) is a client/server protocol defined by the Internet Engineering Task Force (IETF) Constrained RESTful environments (CoRE) Working Group that follows the RFC7252 spec-