Proceedings of the 2017 Winter Simulation Conference
W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, eds.

PERFORMANCE EVALUATION OF MQTT-BASED INTERNET OF THINGS SYSTEMS

Mohamed Handosa

Department of Computer Science
Virginia Tech
Blacksburg, VA 24060, USA

Denis Gračanin

Department of Computer Science
Virginia Tech
Blacksburg, VA 24060, USA

Hicham G. Elmongui

Department of Computer and Systems Engineering
Alexandria University
Alexandria, 21544, Egypt

ABSTRACT

The Internet of Things (IoT) systems usually use constrained devices with limited computation and communication resources facilitating the use of lightweight communication protocols. Message Queue Telemetry Transport (MQTT) is a lightweight publish-subscribe-based messaging protocol that works on top of the TCP/IP protocol. We present our progress towards building a simulation tool for evaluating the Quality of Service (QoS) in MQTT-based IoT systems. This tool can facilitate the design of IoT systems that need to meet certain QoS requirements.

1 INTRODUCTION

Internet of Things is fast becoming integrated in our everyday life (Sinclair 2017). Implementing an IoT system involves interconnecting a variety of devices, some of which with limited resources, through a communication network. Those devices rely on some communication protocol to exchange messages over the network. Different IoT applications tend to have different requirements, which in turn imply different QoS requirements that need to be met by the communicating devices. Meeting those requirements can be crucial, especially for IoT systems with actuation capabilities that can change the state of a physical space. In some applications, failure to meet QoS requirements may result in physical harm. Several lightweight communication protocols have been proposed such as the Constrained Application (CoAP) Protocol and the MQTT Protocol (Thangavel, Ma, Valera, Tan, and Tan 2014). We present initial work on a simulation tool for MQTT-based IoT systems where a message broker receives messages from publishers and distributes them to the subscribers based on the topic of the message.

2 APPROACH

Designing an IoT system is a complex process with many decisions to be taken to ensure a successful system. Prototyping is one way to evaluate different aspects of an IoT system before the actual implementation (Gračanin, Matković, and Wheeler 2015). However, besides the cost involved, it could be difficult to evaluate a system based on prototypes. Fortunately, simulating IoT systems can offer a cheaper solution that also offers the flexibility of changing a variety of parameters and evaluating different candidate designs. Simulating an IoT system requires simulating several entities including the devices and the communication network as well as the context and the behavior of the users.

We rely on the ns-3 network simulator to model MQTT network component of an IoT system. Each node is emulated by a lightweight virtual machine, namely a Linux container. A broker node hosts an implementation of the MQTT broker (Mosquitto), while a client node hosts an application that can publish messages and/or subscribe to topics. The virtual machines are interconnected through the simulated network using a set of virtual bridges and tap devices (Figure 1 left). The tool can automatically create the nodes along with the simulated network that connects them according to a set of parameters that can be adjusted by the user such as the number of publishers, the number of subscribers, the number of topics, the frequency of messages, the required QoS, etc. Figure 1 right shows the average delay for a simulated CSMA network with different number of nodes and different QoS levels. The ability to predict the delay based on the number of publishers (e.g., sensor data sources) is critical to estimate the overall performance of an IoT systems and the trade-off between speed and accuracy.

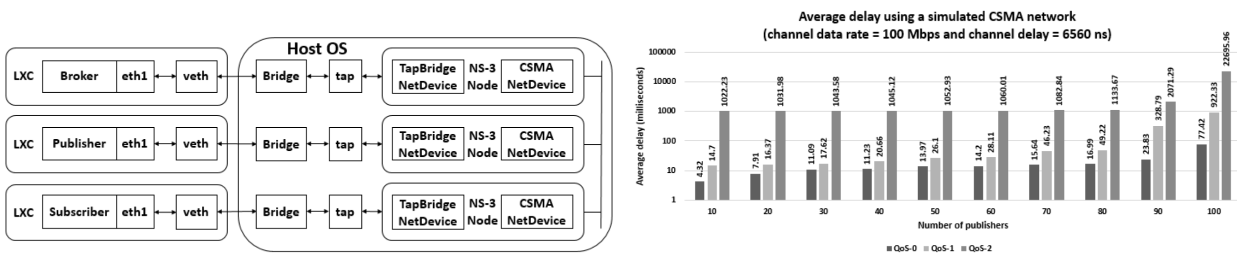

Figure 1: **Left:** An example of modeling a MQTT network component of an IoT system. **Right:** Number of publishers (data sources) versus average delays for QoS-0, QoS-1, and QoS-2.

The simulation results were in agreement with the measurements of the initial implementation of an IoT system for a smart built environment (Gračanin, Handosa, and Elmongui 2017). The measured delay for QoS-0 and up to ten publishers is between three and four milliseconds, consistent with the 4.39 milliseconds delay determined by the simulation.

3 CONCLUSION

The simulation tool allows for simulating the MQTT-based communication traffic in an IoT system. Analyzing the simulated traffic can reveal valuable information that can support the design of an IoT system and help to evaluate ahead its ability to meet certain QoS requirements. For the future work, we are planning to extend the tool to incorporate more adjustable parameters to support a variety of scenarios in terms of IoT system structure and deployment.

REFERENCES

Gračanin, D., M. Handosa, and H. Elmongui. 2017, 9–14 July. "A Service Infrastructure for Human-Centered IoT-based Smart Built Environments". In *Proceedings of the 5th International Conference on Distributed, Ambient and Pervasive Interactions, DAPI 2017*, edited by N. Streitz and P. Markopoulos, Volume 10291 of *Lecture Notes in Computer Science*, 262–274. Cham: Springer International Publishing.

Gračanin, D., K. Matković, and J. Wheeler. 2015, 6–9 December. "An Approach to Modeling Internet of Things Based Smart Built Environments". In *Proceedings of the 2015 Winter Simulation Conference (WSC)*, edited by L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. Rosetti, 3208–3209.

Sinclair, B. 2017. *IoT Inc.: How Your Company Can Use the Internet of Things to Win in the Outcome Economy*. New York: McGraw-Hill Education.

Thangavel, D., X. Ma, A. Valera, H.-X. Tan, and C. K.-Y. Tan. 2014, 21–24 April. "Performance evaluation of MQTT and CoAP via a common middleware". In *Proceedings of the 2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, 1–6. IEEE.