Demonstrating MQTT+: An Advanced Broker for Data Filtering, Processing and Aggregation

Riccardo Giambona
DEIB - Politecnico di Milano
Milano, Italy
riccardo.giambona@mail.polimi.it

Alessandro E.C. Redondi
DEIB - Politecnico di Milano
Milano, Italy
alessandroenrico.redondi@polimi.it

Matteo Cesana
DEIB - Politecnico di Milano
Milano, Italy
matteo.cesana@polimi.it

ABSTRACT

The Message Queueing Telemetry Transport (MQTT) publish/subscribe protocol is the de facto standard at the application layer for IoT, M2M and wireless sensor networks applications. This demonstration showcases MQTT+, an advanced version of MQTT which provides an enhanced protocol syntax and enriches the broker with data filtering, processing and aggregation functionalities. Such features are ideal in all those applications in which edge devices are interested in performing processing operations over the data published by multiple clients, where using the original MQTT protocol would result in unacceptably high network bandwidth usage and energy consumption for the edge devices. MQTT+ is implemented starting from an open source MQTT broker and evaluated in different application scenarios which are demonstrated live using the Node-RED IoT prototyping framework.

CCS CONCEPTS

• **Networks** → **Application layer protocols**; *Network simulations*; • **Computer systems organization** → *Sensor networks*;

KEYWORDS

MQTT, publish/subscribe, data aggregation

ACM Reference Format:
Riccardo Giambona, Alessandro E.C. Redondi, and Matteo Cesana. 2018. Demonstrating MQTT+: An Advanced Broker for Data Filtering, Processing and Aggregation. In *21st ACM International Conference on Modelling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM '18), October 28-November 2, 2018, Montreal, QC, Canada.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3242102.3243317

1 INTRODUCTION

The MQTT protocol has received great attention in the last few years, practically becoming the standard *de-facto* in M2M and IoT applications [1–3]. As a matter of fact, all major cloud platforms (e.g., Amazon AWS, Microsoft Azure, IBM Watson) expose their services through MQTT. The reasons of such popularity derive from MQTT's incredible simplicity client-side, which nicely fits in resource-constrained applications, yet supporting reliability and several degrees of quality of service (QoS). There is however a set of common IoT and M2M applications scenarios where the use of MQTT causes an inefficient use of the available network and computing resources. Those are all cases where data consumers (subscribers) are interested in only a subset of the data produced (published) by sensor devices, while the broker still forwards the entire data available. Examples include clients interested in receiving data only if it respects some condition, clients interested in certain aggregation functions (e.g., cumulative sum, average) over a set of data published, or clients interested in the result of some processing task over such data, rather than the data itself. In all these case, two main drawbacks can be identified: (i), the data forwarded by the broker may potentially be discarded by subscribers, wasting network resources and (ii) subscribers need to perform additional processing operations, consequently decreasing their available computational and energy resources. To mitigate those issues, we showcase in this paper an advanced MQTT broker (MQTT+) able to deal with such situations. MQTT+ allows a client to subscribe to advanced functionalities on the data published, including rule-based data filtering, spatial and temporal data aggregation and data processing. All functionalities are provided reusing as much as possible the original MQTT protocol logical and syntactical rules and minimally modifying the client-side procedures.

MSWIM '18, October 28-November 2, 2018, Montreal, QC, Canada

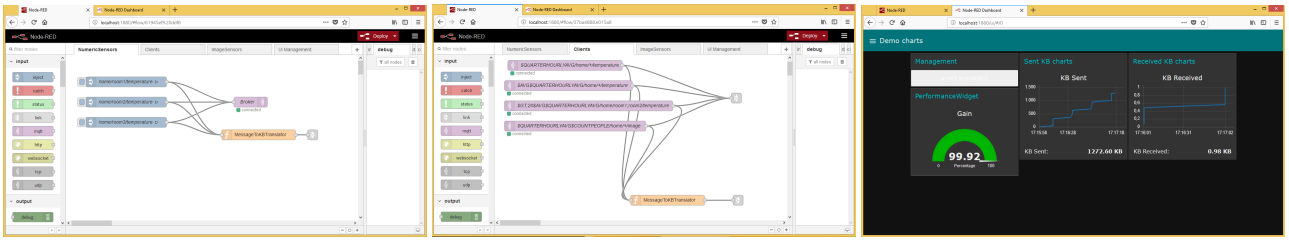
ACM ISBN 978-1-4503-5960-3/18/10…$15.00
https://doi.org/10.1145/3242102.3243317

2 SHOWCASED FEATURES

MQTT follows a traditional publish/subscribe pattern in which a *client* device publishes information relative to a particular *topic*, i.e., a multilevel string describing the data being published (e.g. `kitchen/temp`). Other clients interested in such information subscribe to that topic.

MQTT+ features are oriented at decreasing the computational load on clients and on the network segment from the broker to the subscribers, at the cost of a slight increase in the complexity of the broker implementation. In order to leverage such functions, an enhanced syntax is introduced. The new syntax is nicely integrated with the original MQTT syntactic rules by making use of leading `$` characters followed by several specific keywords and requires optional modifications of negligible impact on clients. The features showcased in this demonstration are:

- **Rule-based subscriptions:** in many cases, a client is interested in a topic only if the data published on it respect some condition. MQTT+ allows a client to perform a rule-based subscription using ad-hoc prefix operators. A device may subscribe to `$GT;value/kitchen/temp` and receive only messages whose payload contains a value greater than `value`. Similarly, operators for greater than or equal (`$GTE;value`), less than (`$LT;value`) and less than or equal (`$LTE;value`) are defined. Finally, the



(a) (b) (c)

Figure 1: (a) Publishers (b) Subscribers to the MQTT+ advanced topics; (c) Real-time performance evaluation

`$CONTAINS;text` operator is defined, which searches in the payload of the published message the string `text`.

- **Temporal-data aggregation:** Often, a subscriber wants to retrieve data at a much lower frequency compared to the publication rate. MQTT+ allows a client to subscribe to special topics like `$<TIME><OP>/topic`, with `OP` = {`COUNT`,`SUM`,`AVG`,`MIN`,`MAX`} and `TIME` = {`DAILY`,`HOURLY`,`QUARTERHOURLY`}. This allows a client to obtain e.g., the daily average of a certain sensor value. The MQTT+ broker handles all buffering and aggregation operations internally in a transparent way.
- **Spatial-data aggregation:** In MQTT, a client may subscribe to a specific topic by using the exact topic string, or subscribe to multiple topics at once by using a single-level (+) or multi-level (#) wildcard. The broker will then forward to the client all messages whose topic matches the subscription topic, including the wildcard. However, a client may be interested in aggregating such topics at once: MQTT+ allows for this possibility. In case a client is interested in aggregating all data matching a topic name, it may subscribe to `$<OP>/topic/`, where `OP` can assume the same values defined for temporal aggregation and `/topic/` contains one or more wildcards, according to MQTT rules.
- **Data processing:** One of the main features of MQTT is that practically any type of data can be transferred with publish and subscribe messages, including multimedia data and images. We focus on the case where subscribers to such image topics are interested in the content of such images, rather than in the pixel-domain based representation of the images. As an example, a subscriber may be interested in counting how many people are present in an image, if a certain person is there or what kind of objects are present. All these operations require image analysis algorithms to be run on the subscriber devices after the broker has forwarded the images from the cameras, with a great effort in terms of network resources and computational power on the edge devices. The proposed enhanced MQTT+ allows to overcome such drawbacks by enabling data processing directly at the broker. Specific operators (such as the `$CNTPPL` prefix to count people) trigger the broker to run specific algorithms and to return the result to the subscriber. We focus here only on image processing, although the framework can be extended to any other type of data and processing operations (video and data compression, signal processing, etc.).
- **Composite subscriptions:** One of the strengths of MQTT+ is the capability of allowing composite subscriptions, by properly chaining the operators introduced so far, thus enabling even more advanced functions. Indeed, MQTT+ supports spatio-temporal aggregations, spatio-temporal aggregation of processed data and even rule-based spatio-temporal aggregation.

3 DEMONSTRATION SCENARIO

The demonstration is showcased on a single laptop, where an implementation of the MQTT+ broker is running. In order to showcase the advanced broker operations, the Node-RED graphical framework is used. Node-RED is emerging as one of the most popular IoT frameworks and it allows to easily create operational flows by using pre-defined or user-defined building blocks. As an example, a user may easily create clients publishing data to the MQTT+ broker with specific topics at a certain rate. Similarly, the user can create subscribing clients, and use the proposed MQTT+ syntax to achieve specific goals. The example in Figure 1 shows four nodes publishing numeric data on different topics, while a single subscriber is interested in their spatio-temporal aggregation (the quarter-hourly average of all measurements). To demonstrate the benefits of MQTT+ compared to MQTT, a different visualization shows the bandwidth usage of a traditional MQTT broker and the gain that the proposed MQTT+ broker provides. Another scenario, comprising sensors publishing images and clients subscribing to the number of people contained in such images is showcased. For a better understanding, other scenarios may be created on-the-fly.

4 CONCLUSION

The demonstration showcases MQTT+, an advanced broker providing data filtering, aggregation and processing. The demonstration, based on a real implementation of the broker and on the Node-RED environment, aims at validating the innovative features of the proposed solution, as well as comparing its performance with the state-of-the-art.

REFERENCES

[1] Andrew Banks and Rahul Gupta. 2014. MQTT Version 3.1. 1. *OASIS standard* 29 (2014).

[2] Urs Hunkeler, Hong Linh Truong, and Andy Stanford-Clark. 2008. MQTT-S - A publish/subscribe protocol for Wireless Sensor Networks. In *Communication systems software and middleware and workshops, 2008. comsware 2008. 3rd international conference on*. IEEE, 791–798.

[3] Dinesh Thangavel, Xiaoping Ma, Alvin Valera, Hwee-Xian Tan, and Colin Keng-Yan Tan. 2014. Performance evaluation of MQTT and CoAP via a common middleware. In *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2014 IEEE Ninth International Conference on*. IEEE, 1–6.