MQTT Quality of Service versus Energy Consumption

Jevgenijus Toldinas[1], Borisas Lozinskis[1], Edgaras Baranauskas[1], Algirdas Dobrovolskis[1,2]
[1]*Department of Computer Science, Kaunas University of Technology, Studentu St. 50, LT-51368 Kaunas, Lithuania*
[2]*Centre of Real Time Computer Systems, Kaunas University of Technology, K. Barsausko St. 59, LT-51423 Kaunas, Lithuania*
eugenijus.toldinas@ktu.lt

***Abstract*—IoT today invented for creating, editing and sharing information and data. Some sources predict the number of tens and hundreds billion connected devices for the year 2020–2025. One of the most popular standard machine-to-machine application layer protocols to interconnect the things and applications to the Internet of Things is Message Queuing Telemetry Transport protocol. Quality of service (QoS) deal with network performance elements like latency, error rate and uptime. MQTT provides three levels of QoS. Usage time of IoT devices is constrained by its most critical resource – battery. This paper deals with the estimation of energy consumption in transferring data using lightweight MQTT protocol over it different QoS levels. For experiments, we implemented client-server architecture and employ MQTT publish/subscribe protocol to transfer data between nodes.**

***Index Terms*—Wireless application protocol; MQTT; Quality of service; Energy consumption; Internet of Things.**

I. Introduction

The term Internet of Things (IoT), defined as smart physical objects being connected to the internet and being able to identify themselves to other devices, was introduced twenty years ago [1]. As the most powerful and popular tool IoT today invented for creating, editing and sharing information and data. Some sources predict the number of 50 billion connected devices for the year 2020 [2]. Another source predicts that in 2025 the number of devices will approach the 200 billion mark [1]. And the next one, predicts that number of connected things would rise up to more than 75 billion at the end of 2025 and there will be more than 9 smart devices per person at the end of 2025 [3].

Several standard machine-to-machine (M2M) application layer protocols are used to interconnect the things and applications to the Internet of Things: Message Queuing Telemetry Transport (MQTT), Constrained Application Protocol (CoAP), Extensible Messaging and Presence Protocol (XMPP), Web Application Messaging Protocol (WAMP) [4]. The Energy consumption of smart objects in complex environments with low energy and battery is the critical point of IoT [2]. MQTT has a low memory footprint, low power consumption, and better distribution of information to recipients [4]. It practically is becoming the standard de-facto protocol in M2M and IoT application, well-known cloud platforms, such as Amazon AWS, Microsoft Azure, and IBM Watson expose their services through MQTT [5]. Simplicity, reliability, three quality of service (QoS) levels, resource-constrained support makes MQTT popular in IoT.

II. Related Work

Because of MQTT popularity and very important role in IoT, many researches were made to investigate it security, reliability, energy consumption and others.

MQTT does not integrate security mechanisms. In [2] authors propose to integrate the concept of the multi agent system at the level of the embedded card for the operation of smart objects improvement in complex environments with low energy consumption. Researchers [4] study and evaluate the famous protocol MQTT latency over different gateways in the industry field.

In [6] IoT Verif presented, an automated tool that learns and identifies SSL/TLS security vulnerabilities of IoT applications that utilize MQTT messaging protocol. Authors [5] demonstrate MQTT+ features that oriented at decreasing the computational load on clients and on the network segment from the broker to the subscribers.

A new approach to the MQTT-SN protocol proposed [7] to publish sensor data, called smart gateway selection method that estimates end-to-end content delay and message loss, during the transmission of content in all levels of QoS.

Authors [8] compared existing application layer IoT and end-user protocols: CoAP, MQTT, XMPP, Representational State Transfer (REST), Advanced Message Queuing Protocol (AMQP), Web sockets. The authors highlight that CoAP, MQTT and AMQP have QoS levels. MQTT is not widely used as HTTP but it has proved to be more efficient for battery-run devices [8].

Researchers [9] in general argue the smart energy management for smart homes and cities. In their research, MQTT QoS-2 excluded from test cases due to its large overhead. Presented experiment with MQTT number of clients 1000 shows: (i) Throughput: QoS0 124.3 (kbps), QoS1 101.5 (kbps); (ii) Latency: QoS0 126 (ms), QoS1 135 (ms); Packets Dropped: QoS0 3.1 %, QoS1 3.8 %.

In [3] authors investigate the side effects of reliability on

Manuscript received 21 January, 2019; accepted 4 April, 2019.

IoT communication protocols and consider MQTT as reliable protocol, since it utilizes acknowledge policy in its transmissions.

Hypertext Transfer Protocol (HTTP), MQT and CoAP were analysed in [10] and experiment results show that both CoAP and MQTT consumes almost equal amounts of energy for sending data.

In [11] authors examine and analyse key security issues in IoT systems and commercial IoT edge-device using MQTT (+TLS) and CoAP (+DTLS) protocols. Experiment results show that securing nodes using TLS/DTLS resulted in only 4.7 % overhead for MQTT with the varying QoS levels and 5 % for CoAP.

Research in the field of the impact of the application layer protocol on energy consumption [12] shows that MQTT has strong advantages compared to REST, because MQTT application layer protocol was specialized for IoT communication while HTTP was extended to address new requirements.

III. MQTT AND QUALITY OF SERVICE

The MQTT is a lightweight asynchronous protocol which model based on Publish/Subscribe sequence flowing over TCP/IP to connect large numbers of remote sensors and control devices [13]; it allows devices to exchange data by using message broker. The broker not only pushes message data to the subscribed customers, but additionally it stores, forwards, filters, and prioritizes publish requests from the publisher to the subscriber. Publisher and subscriber could switch their roles depending on their objectives. Quality of service deal with network performance elements like latency, error rate and uptime. MQTT provides levels of QoS [14]:

At most once (QoS = 0), the message is sent to subscriber only once (at most once), it is not stored and publisher does not receive any information about message delivery. In that case, if the subscriber is not able to receive message or is not connected to the network or in the case of the publisher malfunction the message will be lost. The term "fire and forget" characterizes *At most once* (QoS = 0) level.

At least once (QoS = 1), the message is sent to the subscriber leastwise one time. Acknowledgment procedure is used in that QoS level and if publisher does not get an acknowledgement other publisher attempts to send message to the subscriber will take place. The publisher shall use set DUP flag for duplicated message and every duplicated message is sent setting DUP flag until acknowledgement from the subscriber will be received. To ensure message delivery at least once the message must be stored at the publisher and the subscriber until delivery acknowledgement would be received. *At least once* (QoS = 1) is the default QoS level of the MQTT.

Exactly once (QoS = 2), is used to ensure safety and guarantee the message delivery. During publishing the message, two rounds of transmission are used. The message until it is processed must be stored at the publisher and the subscriber. At the first round, the publisher sends the message to the subscriber and waits for an acknowledgement from the subscriber that the message had been stored. Similarly with *At least once* (QoS = 1) if publisher does not get an acknowledgement other publisher attempts to send message to the subscriber will take place. After acknowledgement from the subscriber is received the second transmission round takes place. At the second round, the publisher by using the PUBREL message informs the subscriber that he can complete processing the message and waits for an acknowledgement for the PUBREL message from the subscriber. If the publisher does not get an acknowledgement of the PUBREL message others attempts to send the PUBREL message to the subscriber will take place until an acknowledgement is received. Only after an acknowledgement of the PUBREL message is received the publisher deletes stored message.

According to an energy consumption and battery life MQTT and QoS domain is depicted in Fig. 1.

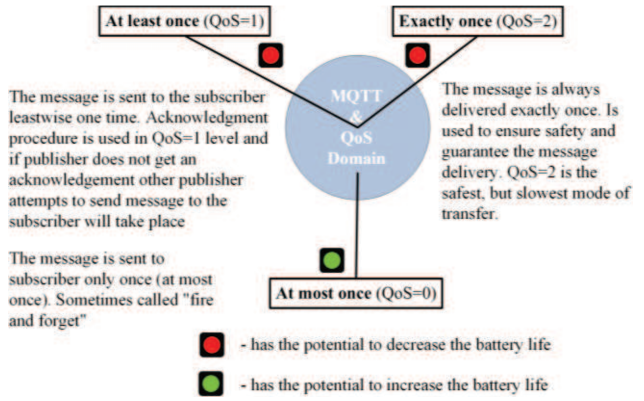

Fig. 1. MQTT and QoS domain.

Because energy consumption of IoT with low energy and battery level is critical point, QoS levels must deal with it.

IV. MODELLING ENERGY CONSUMPTION IN THE QOS LEVELS OF MQTT

Feature diagram, inheritance, templates and other techniques are used usually for modelling variabilities [15]. In our case, we use feature diagram which model has the graphical notation where the features, their attributes, concepts, relations and constraints are represented (Fig. 2).

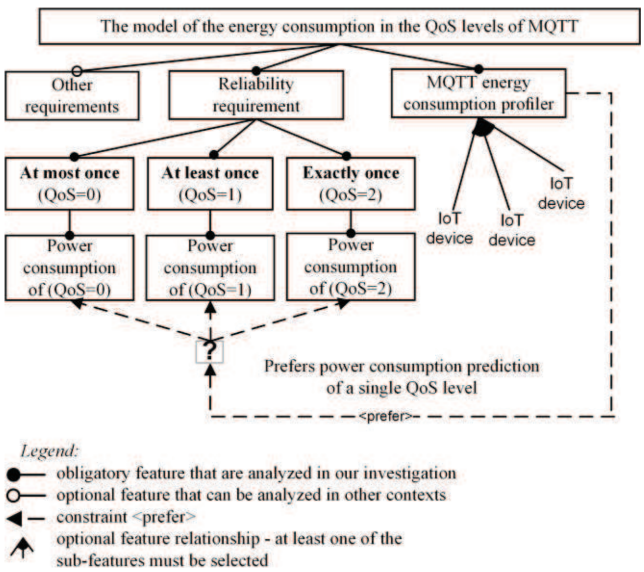

Fig. 2. The model of the energy consumption in the QoS levels of MQTT.

The top-level feature is shown by the root node, in our case that is “The model of the energy consumption in the QoS levels of MQTT”. Variants are depicted using transitional features that are no decomposable into smaller features in our proposed context. Edges mark possible relations, relevancies and constraints between features. Information about the used elements in the feature diagram is shown in the legend (see Fig. 2).

In proposed model (Fig. 2), we evaluate reliability requirements in terms of MQTT QoS levels with energy consumption due to the nature of IoT systems, where energy consumption plays an important role for the system’s lifetime.

V. Case Study and Experimental Results

The modules that have been used for our experiments depicted in Fig. 3.

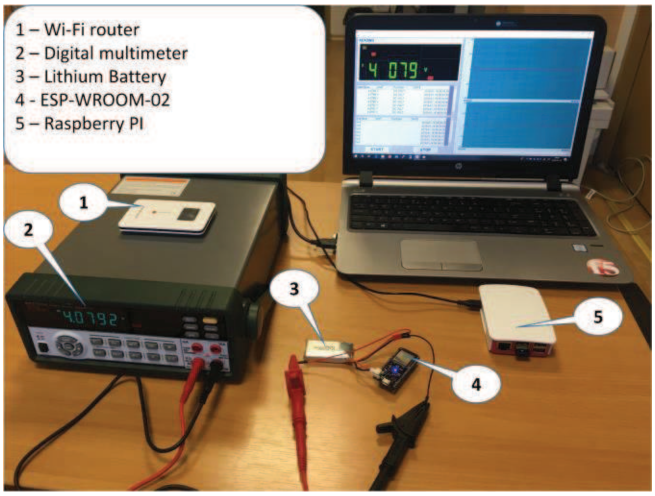

Fig. 3. The modules that have been used for experiments.

One of the main aspects of IoT infrastructures is wireless communication and we setup the experimental environment through the Wi-Fi technology. We have used (see Fig. 3) Wi-Fi router (1), Raspberry Pi, (5) and Wi-Fi Internet of Things Module ESP-WROOM-02 (4). The module has been adjusted to get the best RF performance and it integrates ESP8266EX. It is recommended for tests or for further development. Table I and Table II show the technical specifications of the experimental setup.

TABLE I. TECHNICAL SPECIFICATIONS OF ESP-WROOM-02.

Items	**Specifications**
Operating voltage/Power supply	2.7 V ~ 3.6 V
Processor	Tensilica L106, 32-bit, RISC
Processor clock speed	160 MHz
Wi-Fi frequency range	2.4 GHz ~ 2.5 GHz
Wi-Fi communication protocols	802.11 b/g/n
Operating current	Average: 80 mA

TABLE II. TECHNICAL SPECIFICATIONS OF RASPBERRY PI.

Items	**Specifications**
CPU	Broadcom BCM2837 Arm7 Quad Core
CPU clock frequency	900 MHz
RAM	1 GB
Wi-Fi communication protocols	802.11 b/g/n
GPU	Broadcom VideoCore IV
CPU clock frequency	250 MHz

We use digital multimeter MASTECH MS8050 (2) for measurements and Lithium battery (3) LS903052, 3.7 V, 1200 mAh as power supply for ESP-WROOM-02 (see Fig. 3).

To investigate the energy consumption of MQTT QoS levels, we have established a simple scenario using the experimental environment (see Fig. 3). We have employed Raspberry Pi and ESP-WROOM-02 and connected them through the Vodafone Wi-Fi router (1). Raspberry Pi and ESP-WROOM-02 acts as the two end-nodes of the network - that simple scenario could be extended to accommodate multiple nodes. Two mentioned above end-nodes are located at a distance less than 1m from each other and Vodafone Wi-Fi router. We conduct our experiments in standard room conditions. In case of our experiment scenario, ESP-WROOM-02 is generating data packets as a publisher. Simultaneously ESP-WROOM-02 is receiving data packets as a subscriber. Raspberry Pi is playing broker role, it receives the data packets from the publisher and forwards received data packets to the subscriber.

The results of measurements are presented in Fig. 4–Fig. 6. Figure 4 shows the battery voltage level for MQTT QoS *At most once* (QoS = 0), Fig. 5 shows the battery voltage level for MQTT QoS *At least once* (QoS = 1) and Fig. 6 shows the battery voltage level for MQTT QoS *Exactly once* (QoS = 2).

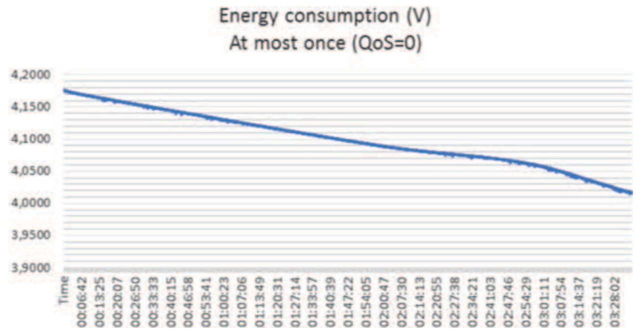

Fig. 4. Battery voltage level for MQTT QoS *At most once* (QoS = 0).

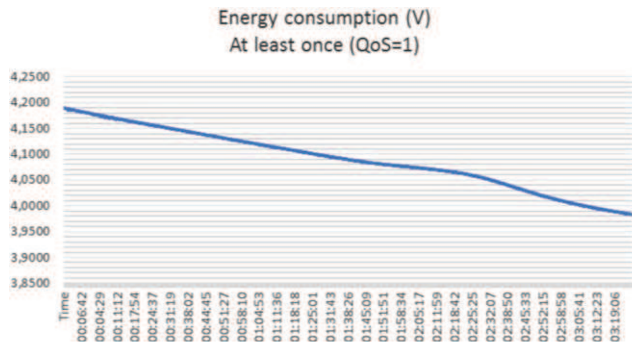

Fig. 5. Battery voltage level for MQTT QoS *At least once* (QoS = 1).

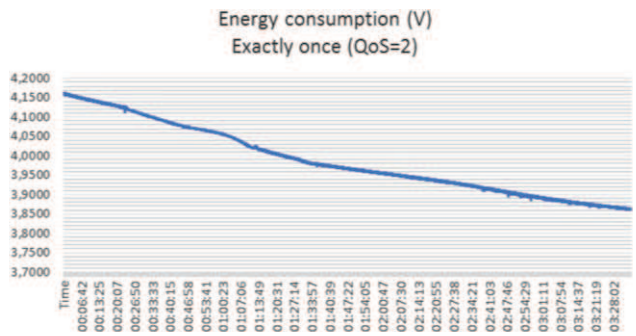

Fig. 6. Battery voltage level for MQTT QoS *Exactly once* (QoS = 2).

We evaluate the difference in energy consumption that was expected because in QoS *At most once* (QoS = 0) message delivery is not acknowledged. In QoS *At least once* (QoS = 1) acknowledgment for confirmed message reception is sent, which adds to the latency of message transmission in QoS-1. In QoS *Exactly once* (QoS = 2), the message is always delivered exactly once and it is the safest, but slowest mode of transfer.

The experimental results graphically represented in Fig 7.

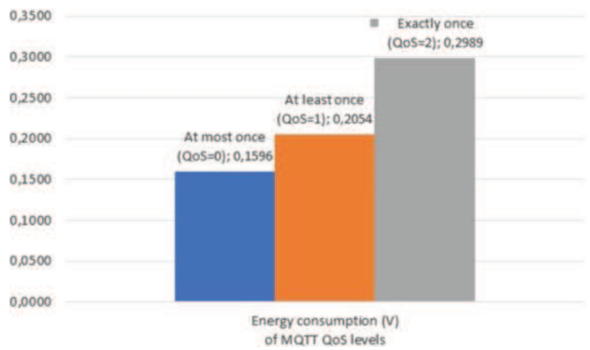

Fig. 7. Energy consumption of MQTT QoS levels.

The experimental results are summarized in Table III.

TABLE III. EXPERIMENTAL RESULTS FOR MQTT QOS LEVELS.

MQTT QoS level	Battery energy consumed (V)
At most once (QoS = 0)	0.1596
At least once (QoS = 1)	0.2054
Exactly once (QoS = 2)	0.2989

On the basis of these results we can evaluate the difference in energy consumption of three MQTT protocol QoS levels. QoS *At most once* (QoS = 0) consumes 29 % less energy than QoS *At least once* (QoS = 1), and 87 % less energy than QoS *Exactly once* (QoS = 2). QoS *At least once* (QoS = 1) consumes 45 % less energy than QoS *Exactly once* (QoS = 2).

VI. CONCLUSIONS

The energy consumption of MQTT protocol is highly different in case of using various QoS levels. Therefore, the developers of IoT systems should choose the most appropriate QoS level by taking into account the requirement of reliability, latency and energy needed to perform communication with respect to the battery lifetime.

The main results of this paper are as follows:

1) In real time measured values for energy consumption of MQTT protocol obtained, using various QoS levels.

2) Our proposed model of the energy consumption in the QoS levels of MQTT can be used for energy consumption profiling in IoT systems

3) Using MQTT protocol for IoT communications the results of our experiment can be used for energy consumption prediction of three QoS levels:

– QoS *At least once* (QoS = 1) consumes 29 % more energy than QoS *At most once* (QoS = 0),

– QoS *Exactly once* (QoS = 2) consumes 87 % more energy than QoS *At most once* (QoS = 0),

– QoS *Exactly once* (QoS = 2) consumes 45 % more energy than QoS *At least once* (QoS = 1).

REFERENCES

[1] Philip N. Howard, "Sketching out the Internet of Things trendline", 2015. [Online]. Available: https://www.brookings.edu/blog/techtank/2015/06/09/sketching-out-the-internet-of-things-trendline

[2] F. Z. Chafi, Y. Fakhri, "The integration of Multi Agent System within the Internet of Things", in *Proc. 3rd Int. Conf. Smart City Applications (SCA 2018)*, New York, New York, USA, 2018, pp. 1–8. DOI: 10.1145/3286606.3286835.

[3] B. Safaei, A. M. H. Monazzah, M. B. Bafroei, A. Ejlali, "Reliability side-effects in Internet of Things application layer protocols", in *2nd Int. Conf. System Reliability and Safety (ICSRS 2017)*, Milan, Italy, 2017, pp. 207–212. DOI: 10.1109/ICSRS.2017.8272822.

[4] S. B. Kenitar, S. Marouane, A. Mounir, A. Younes, A. G. Gonzalez, "Evaluation of the MQTT protocol latency over different gateways", in *Proc. 3rd Int. Conf. Smart City Applications (SCA 2018)*, New York, New York, USA, 2018, pp. 1–5. DOI: 10.1145/3286606.3286864.

[5] R. Giambona, A. E. C. Redondi, M. Cesana, "Demonstrating MQTT+: an advanced broker for data filtering, processing and aggregation", in *Proc. 21st ACM Int. Conf. Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM 2018)*, New York, New York, USA, 2018, pp. 357–358. DOI: 10.1145/3242102.3243317.

[6] K. Alghamdi, A. Alqazzaz, A. Liu, H. Ming, "IoTVerif: an automated tool to verify SSL/TLS certificate validation in android MQTT client applications", in *Proc. Eighth ACM Conf. Data and Application Security and Privacy (CODASPY 2018)*, New York, New York, USA, 2018, pp. 95–102. DOI: 10.1145/3176258.3176334.

[7] D. Guha Roy, B. Mahato, D. De, R. Buyya, "Application-aware end-to-end delay and message loss estimation in Internet of Things (IoT) — MQTT-SN protocols", *Future Generation Computer Systems*, vol. 89, pp. 300–316, 2018. DOI: 10.1016/j.future.2018.06.040.

[8] V. Karagiannis, P. Chatzimisios, F. Vazquez-gallego, J. Alonso-zarate, "Application Layer protocols for the Internet of Things research motivation", *Trans. IoT and Cloud Computing*, pp. 1–10, 2015.

[9] A. R. Al-Ali, I. A. Zualkernan, M. Rashid, R. Gupta, M. Alikarar, "A smart home energy management system using IoT and big data analytics approach", *IEEE Trans. Consumer Electronics*, vol. 63, no. 4, pp. 426–434, 2017. DOI: 10.1109/TCE.2017.015014.

[10] M. Pavelic, V. Bajt, M. Kusek, "Energy efficiency of machine-to-machine protocols", in *41st Int. Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO 2018)*, 2018, pp. 0361–0366. DOI: 10.23919/MIPRO.2018.8400069.

[11] S. Shapsough, F. Aloul, I. A. Zualkernan, "Securing low-resource edge devices for IoT systems", in *Int. Symposium in Sensing and Instrumentation in IoT Era (ISSI 2018)*, Shanghai, China, 2018, pp. 1–4. DOI: 10.1109/ISSI.2018.8538135.

[12] J. Hofer, S. Pawaskar, "Impact of the application layer protocol on energy consumption, 4G utilization and performance: How the application layer protocol strengthens green IoT", *3rd Cloudification of the Internet of Things Conf. (CIoT 2018)*, 2019.

[13] IBM, "IBM MQTT protocol", 2019. [Online]. Available: https://www.ibm.com/support/knowledgecenter/en/SSFKSJ_8.0.0/com.ibm.mq.pro.doc/q002870.html

[14] IBM, "Qualities of service provided by an MQTT client", 2019. [Online]. Available: https://www.ibm.com/support/knowledgecenter/en/SSFKSJ_8.0.0/com.ibm.mq.dev.doc/q029090.html.

[15] S. Ripon, M. M. Piash, S. M. A. Hossain, M. S. Uddin, "Semantic WebBased analysis of product line variant model", *International Journal of Computer and Electrical Engineering*, pp. 1–6, 2014. DOI: 10.7763/IJCEE.2014.V6.783.