Performance Evaluation of Bottle Press Machine MQTT Data Acquisition

Heru Wijanarko, Ahsani Abdun, Diono Diono, Prasaja Wikanta, Hamdani Arif, Asrizal Deri Futra

Depart2ment of Electrical Engineering
Politeknik Negeri Batam Batam, Indonesia
wijanarko@polibatam.ac.id, ahsaniabdun@gmail.com, diono@polibatam.ac.id, antok@polibatam.ac.id, deri@polibatam.ac.id, hamdaniarif@polibatam.ac.id

Abstract. MQTT (Message Queuing Telemetry Transport) is a communication protocol designed for connecting devices in scenarios like machine-to-machine interactions and the Internet of Things (IoT). It excels in resource -constrained environments, slow data connections, and situations with data transmission delays. MQTT ensures data reliability using Quality of Service (QoS) levels, which are divided into three: QoS 0, QoS 1, and QoS 2, covering parameters like Throughput, Packet Loss, Delay, Round-Trip Time (RTT), and Retransmission. This research assessed MQTT's performance in acquiring data from a bottle press machine, employing a PLC TM241CE40R controller developed by PBL (Project Based Learning) at Politeknik Negeri Batam. The results showed that QoS 2 achieved the highest throughput at 1.9 kbps, while QoS 0 had the lowest at 500 bps. Packet loss was highest at QoS 0 (0.81%) but absent in QoS 1 and 2. QoS 2 had the highest delay (approximately 330 ms), whereas QoS 0 had the lowest (around 50 ms). RTT closely matched delay, indicating a well-functioning broker. The 10-meter test exhibited the most retransmissions, revealing unstable internet connectivity, and QoS level 1 was deemed suitable for the Bottle Press Machine system.

Keywords: bottle press machine; data acquisition; MQTT; performance; QoS.

1. Introduction

The era of technology is rapidly advancing. These advancements bring immense benefits to humanity and the industrial sector, where technology advancements, particularly the Internet, are incredibly advantageous. With the advent of the Internet, we can become familiar with the Internet of Things (IoT). IoT is a paradigm in which an object can independently communicate data over the Internet without human-human or physical interactions [1]. IoT has been widely applied in various fields, such as smart homes [2] for monitoring the use of electrical energy, monitoring hinterland areas [3] environmental parameter data acquisition, monitoring water quality [4] leveraging fuzzy classifiers, location detection [5] for the tracking system, and several other examples.

M. U. H. Al Rasyid and M. R. Mufid (eds.), *Proceedings of the International Conference on Applied Science and Technology on Engineering Science 2023 (iCAST-ES 2023)*, Advances in Engineering Research 230,
https://doi.org/10.2991/978-94-6463-364-1_99

MQTT is one of the protocols that can be used for IoT data transmission. Message Queue Telemetry Transport (MQTT) employs the publish and subscribe approach, which allows the user to customize the sending and receiving of data because MQTT sends data according to predefined topics [6]. MQTT messages transmitted to the broker contain topics submitted by publishers; these topics are subsequently forwarded to subscribers in response to user requests. Moreover, Quality of Service (QoS) monitors throughput, latency, jitter, and packet loss. QoS aims to determine the most suitable service capabilities for a particular service or system [7]. The MQTT protocol offers three QoS levels, from the lowest QoS-0 to the highest QoS-2 [8].

Several studies of the MQTT protocol's QoS have been conducted. An example is the MQTT-based IoT system with ten publishers [9]. This investigation contrasted simulation as the initial implementation of an Internet of Things (IoT) system to 10 real publishers. Consistent with the 4.39 milliseconds, the measurement results for QoS-0 and up to ten publishers were between three and four milliseconds. A comparison of representative communications protocols for IoT systems was investigated in [10]. Overall, the Constrained Application Protocol (CoAP) protocol provides the best. However, MQTT has the lowest losses regarding packet loss in that study. Strengthened by another research [11], MQTT demonstrated that these protocols perform similarly to CoAP in the testbed, indicating that these protocols are among the most widely usedin consumer and Industrial IoT environments. In addition, [12] assessed the Raspberry Pi Zero W's performance as an IoT gateway based on the MQTT protocol. The experiment's findings indicate that the chosen QoS level has a minimal impact on the system's performance.

The research [13] is similar to a bottle press machine system but employs Electro-Pneumatics FESTO with the CoAP protocol and a Raspberry Pi. A comparison of MQTT and CoAP performance has been carried out, although the environment utilized was a virtual setting using a virtual reality system. Furthermore, in the [14] study, the MQTT protocol was also employed, but with a more specific focus on evaluating the system's performance to test and demonstrate its capabilities and advantages.

This research is to validate the reliability and effectiveness of MQTT in a practical setting, specifically in the domain of data acquisition from a bottle press machine system. This research evaluated QoS performance for delay, throughput, packet loss, RTT (Round Trip Time), and retransmission for each level within the MQTT QoS protocol. This analysis was performed using a bottle press machine controlled by a PLC Schneider TM241CE40R controller, with data acquisition sourced from an infrared sensor to detect the presence of bottles in the pressing section. The aim was to determine the optimal QoS level for data acquisition in the bottle press machine system. By analyzing these metrics under various scenarios, we aim to contribute novel findings that go beyond the general acknowledgment of MQTT's lightweight design.

2. Method

MQTT is a protocol that operates on top of TCP/IP at the Application layer. MQTT uses a publish/subscribe concept, allowing subscribers to receive messages from publishers by subscribing to relevant topics without repetitive requests. MQTT is lightweight and well-suited for machine-to-machine (M2M) or IoT applications. It can work effectively on resource-constrained devices with limited bandwidth and power resources. MQTT consists of four main components: Publisher, Broker, Subscriber, and Topic. The sender of data is referred to as the publisher, the receiver of data is the subscriber, and the broker serves as an intermediary for message exchange between publishers and subscribers. Both publishers and subscribers must be connected to the same broker to send or receive data. Data sent must be associated with a specific topic, enabling the broker to identify which publisher's message is being sent and route it to the appropriate subscriber.

In the realm of QoS, levels represent agreements between message senders and receivers that define the assurance of message delivery for specific messages. With QoS levels 1 and 2, we can be assured that the packets we send will reach their destination. The MQTT protocol offers several QoS levels, namely level 0, level 1, and level 2 [15]. QoS level 0 is often referred to as "fire and forget." At this level, a message is sent only once without checking whether it reaches its destination. QoS level 1 sends a message at least once and checks the delivery status using PUBACK (Publish Ack). When PUBACK is lost, there is a possibility that the message may be sent or received more than once. QoS level 2 guarantees that messages are delivered only once to the intended recipient through a 4-way handshake. This level is the most secure as no messages are lost, but it may have longer delays or latency [16].

In the illustrative depiction of data transmission presented in Figure 1, the infrared sensor functions as a publisher, disseminating infrared data under the designated "infr" topic to a central broker. Computers and mobile devices, serving as subscribers, actively subscribe to the "infr" topic through their connection with the broker, ensuring that both classes of devices consistently receive temperature-related data.

2.1 QoS Parameters

1) Throughput

Throughput is the total number of packets successfully observed or sent to a destination during a specific time interval divided by the duration. Throughput is measured in bps (bits per second), quantifying the effective data transfer speed or rate. The throughput calculation is expressed as in (1).

$$Throughput = \frac{Data\ packets\ received}{Observation\ time} \quad (1)$$

2) Packet Loss

Packet Loss is a parameter that describes a condition indicating the number of packets lost, which can occur due to collisions and congestion in the network. The packet loss calculation is articulated as in (2).

$$Packet\ Loss = \frac{(Data\ packets\ sent - Data\ packets\ received)}{Data\ packets\ sent} \times 100\% \tag{2}$$

3) Delay

Delay (Latency) is the time it takes for data to travel from source to destination. Distance, physical media, congestion, or long processing times can influence delay. Delays can manifest due to geographical distance, physical transmission media, network congestion, or protracted processing durations. The average delay is determined through the following formula (3).

$$Average\ Delay = \frac{Total\ Delay}{Total\ packets\ received} \tag{3}$$

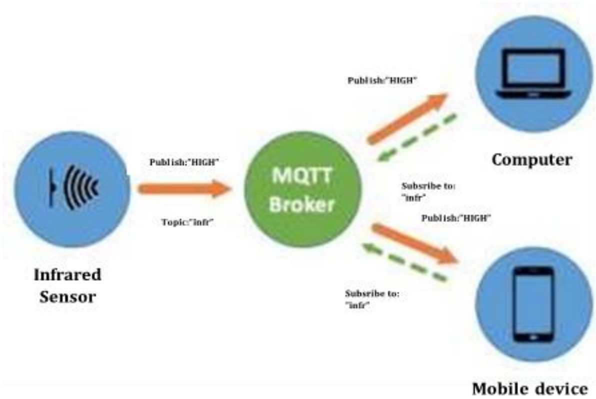


Fig. 1. Infrared sensor MQTT flow

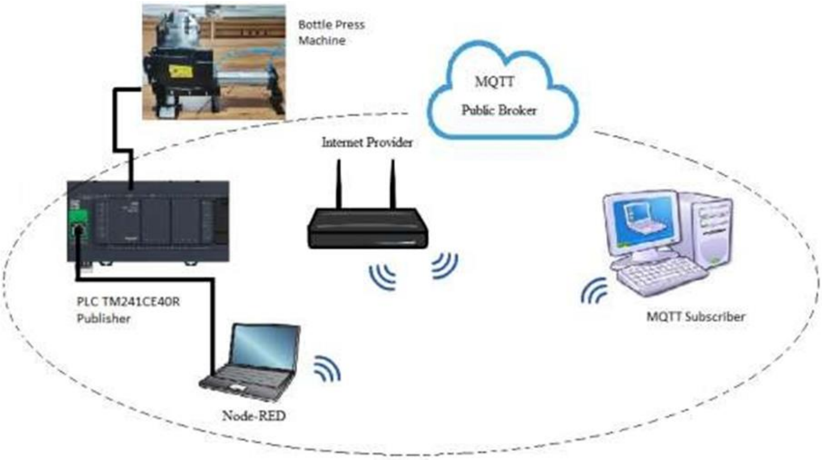


Fig. 2. System topology

4) Round Trip Time (RTT)
Round Time Trip (RTT) is the time it takes for a packet to be sent and acknowledged. Every TCP protocol has an RTT because TCP is reliable and does not allow for lost packets. Any lost packets in TCP will be resent.

5) Retransmission
Retransmission pertains to resending packets within a network due to network anomalies or imperfections. Network irregularities can lead to packet damage or loss, necessitating the retransmission of affected packets for data integrity and reliability.

2.2 The System's Topology

In the context of this research, shown in Fig. 2., when a signal is received from a sensor attached to the press machine, the Programmable Logic Controller (PLC) sends the relevant data or topic to an MQTT broker. Then, the MQTT broker stores this data in its repository. If any client in the system subscribes to the corresponding topic, the MQTT broker forwards the data or topic to the subscribing entity.

For the experimental setup in this investigation, a PLC is combined with a laptop acting as the publisher, and Node-RED is utilized to facilitate this functionality. Concurrently, a public MQTT broker is employed, and a separate laptop is assigned to the subscriber. It is worth noting that the chosen public MQTT broker is HiveMQ.

3. Results And Discussion

Several experiments were conducted to evaluate the performance of the MQTT protocol under varying conditions of distance and time. The assessment involved examining Quality of Service (QoS) parameters at three levels: QoS level 0, QoS level 1, and QoS level 2. The measured throughput values exhibited significant differences among these QoS levels. QoS level 0 had the lowest throughput, while the highest was achieved with QoS level 2. This divergence can be attributed to the underlying QoS mechanisms. QoS level 0 involves a single packet transmission, QoS level 1 entails a two-packet exchange, and QoS level 2 uses a four-way handshake mechanism, involving a minimum of four packets for each publisher-subscriber interaction, resulting in the highest throughput.

Interestingly, the analysis revealed that the distance between nodes had a minimal impact on throughput. Instead, internet stability emerged as the primary factor influencing throughput performance. At a 10-meter distance, there was an increased rate of retransmissions, leading to higher data transmission and throughput compared to distances of 5 and 15 meters.

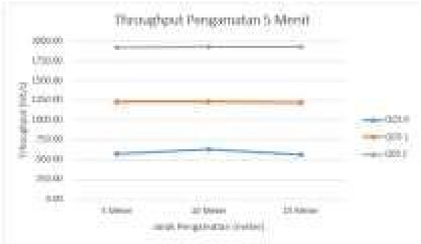
Packet loss, a critical metric in Fig. 4., showed different behaviour across QoS levels. The highest packet loss rate occurred at QoS level 0, reaching 0.81% of lost packets. In contrast, QoS levels 1 and 2 exhibited zero packet loss, highlighting their effectiveness in managing data transmission on the press machines under research.

Further examination of packet loss patterns revealed that the most significant packet loss occurred during QoS level 0 testing at a 10-meter distance, with a reduction in packet loss at the 15-meter distance. This observation underscores the influence of internet stability, as the 10-meter distance experienced a less stable connection due to using a smartphone hotspot, which is sensitive to factors such as service provider, smartphone specifications, and geographical location.

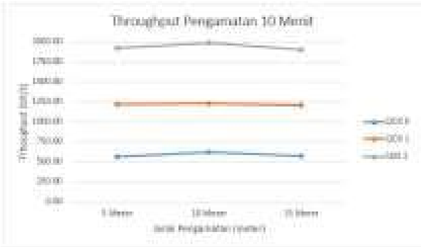
Regarding delay, shown in Fig. 5., QoS level 0 had the shortest delay, approximately 55ms, due to its "fire-and-forget" approach. In contrast, QoS level 1 incurred a delay of roughly 300ms waited for acknowledgment from the broker. Meanwhile, QoS level 2 introduced an additional delay component, reaching around 330ms, ensuring message delivery through the PUBREC process.

Additionally, the investigation indicated that longer testing distances moderately increased delay, with minimal discrepancies observed. Round Trip Time (RTT) presented in Fig. 6. closely mirrored delay, showing minor differences. At QoS level 0, RTT exhibited only a 2ms difference from delay, while QoS level 1 had a 3ms difference, and QoS level 2 showed only a 1ms variance from delay. This alignment suggests the effective operation of the public broker server, HiveMQ.

As shown in Fig. 7., retransmissions, a recurring phenomenon across all QoS levels, displayed a notable dependence on testing distance. Specifically, the 10-meter test distance consistently exhibited higher retransmission rates than distances of 5 and 15 meters. This finding underscores the unstable nature of the internet connection at the 10-meter distance, resulting in an elevated occurrence of damaged or faulty packets and, consequently, an increased frequency of retransmissions. This trend closely aligns with the observed packet loss patterns at the same 10-meter distance, reaffirming the critical role of internet stability in network performance.



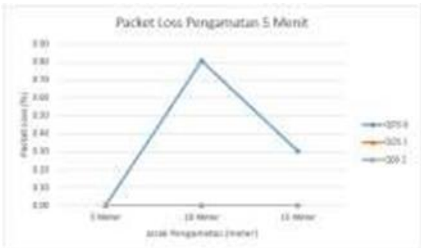
(a)



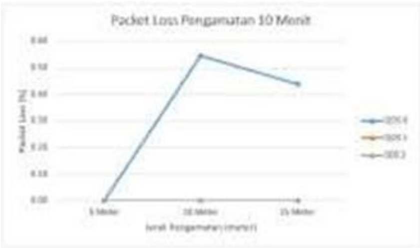
(b)

(c)

Fig. 3. Throughput results on (a) 5 minutes; (b) 10 minutes; (c) 15 -minute scenarios



(a)

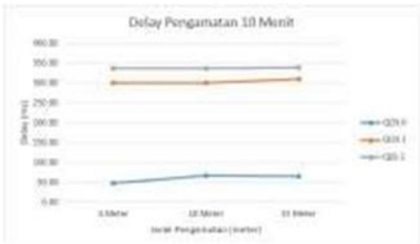


(b)

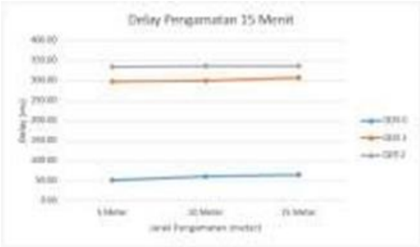
(c)

Fig. 4. Packet Loss results on (a) 5 minutes; (b) 10 minutes; (c) 15 minutes scenarios

(a)

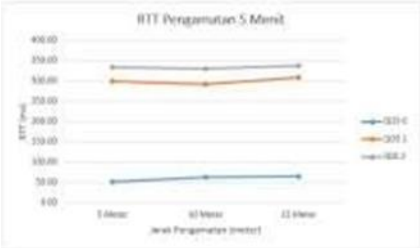


(b)



(c)

Fig. 5. Delay results on (a) 5 minutes; (b) 10 minutes; (c) 15 minutes scenarios



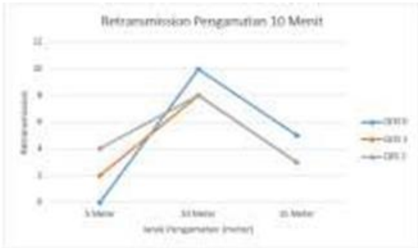
(a)

(b)

(c)

Fig. 6. RTT results on (a) 5 minutes; (b) 10 minutes; (c) 15 minutes scenarios

(a)



(b)

(c)

Fig. 7. Retransmission results on (a) 5 minutes; (b) 10 minutes; (c) 15 minutes scenarios

4. Conclusion

In summary, this research successfully implemented MQTT for a machine controlled by the Schneider TM241CE40R PLC, with the assistance of additional applications like Node-RED. Wireshark proved helpful for managing MQTT protocol parameters and variables. The study revealed varying performance results across different QoS levels. The highest data transfer speed, or throughput, was achieved with QoS 2, while the lowest was observed with QoS 0. Packet loss was most pronounced in QoS 0 but negligible in QoS 1 and QoS 2. Delay times were shortest with QoS 0, followed by QoS 1 and QoS 2, with round-trip times closely tracking these delays, indicating the public broker HiveMQ's effective operation. Retransmissions occurred across all QoS levels, with more frequent instances at a 10-meter testing distance, emphasizing the importance of internet stability in MQTT network performance.

In practical terms, the choice of QoS level in MQTT should be carefully tailored to specific system needs. For clients valuing speed and less concerned about data accuracy, QoS 0 may suffice. Those seeking resource optimization and not bothered by occasional duplications might prefer QoS 1. Conversely, clients prioritizing data accuracy, avoiding data loss, and eliminating message duplication should opt for QoS 2. In the case of the data acquisition system for bottle press machines in this research, QoS level 1 proved suitable for monitoring purposes, ensuring prompt data reception and no data loss. Ultimately, internet speed and stability emerged as key influencers in MQTT protocol performance, impacting publishers, subscribers, and brokers alike.

Acknowledgment

This work is part of the Project Based Learning (PBL) in Politeknik Negeri Batam, conducted by the Mechatronics Engineering Study Program. We gratefully acknowledge and address the Industrial Smart Factory & Automation Laboratory (ISFAL) for providing the necessary support and guaranteeing all the facilities for this research.

References

1. K. Rose, S. Eldridge, and L. Chapin.: "The internet of things: An overview," *The internet society (ISOC)*, no. 80, pp. 1–50, (2015).
2. R. Rohmiyati, M. S. Gozali, and H. Wijanarko.: " Sistem Pemantauan Energi Listrik Rumah Pintar Berbiaya Rendah," *Journal of Applied Sciences, Electrical Engineering and Computer Technology*, vol. 1, no. 1, Art. no. 1, Apr. (2020).
3. S. Aisyah, M. L. Prasetyo Nugroho, Kamarudin, and H. Wijanarko.: "Image Transmission Performance Over UHF Channel Using Real Time Processing in LOSand NLOSTransmission," in *2019 2nd International Conference on Applied Engineering (ICAE)*, Oct. 2019, pp. 1–6(2019)
4. M. H. Ramadhan, G. Dewantoro, and F. D. Setiaji.: " Rancang Bangun Sistem Pakar Pemantau Kualitas Air Berbasis IoT Menggunakan Fuzzy Classifier," *Jurnal Teknik Elektro*, vol. 12, no. 2, Art. no. 2, Dec. (2020).
5. S. Aisyah, A. F. Daulay, H. Wijanarko, D. S. Pamungkas, and K. Kamarudin.: " IoT-Based Tracking System of Transceiver Location," *Jurnal Rekayasa Elektrika*, vol. 17, no. 4, Art. no. 4, Dec. (2021).
6. H. A. Rochman, R. Primananda, and H. Nurwasito.: " Sistem Kendali Berbasis Mikrokontroler Menggunakan Protokol MQTT pada Smarthome," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 1, no. 6, pp. 445–455, (2017).
7. I. Iskandar and A. Hidayat.: "Analisa Quality of Service (QoS) Jaringan Internet Kampus (Studi Kasus: UIN Suska Riau)," *Jurnal CoreIT: Jurnal Hasil Penelitian Ilmu Komputer dan Teknologi Informasi*, vol. 1, no. 2, Art. no. 2, (2015).
8. F. Ilham, A. G. Putrada, and S. Prabowo.: "Analisis Performansi Qos Mqtt Pada Sistem Monitoring Sungai," *eProceedings of Engineering*, vol. 6, no. 1, Apr. 2019, Accessed: Aug. 12, (2022). [Online]. Available: https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/8482

9. M. Handosa, D. Gračanin, and H. G. Elmongui.: "Performance evaluation of MQTT-based internet of things systems," in *2017 Winter Simulation Conference (WSC)*, Dec. 2017, pp. 4544–4545.(2017)

10. T. Moraes, B. Nogueira, V. Lira, and E. Tavares.: "Performance Comparison of IoT Communication Protocols," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, Oct. 2019, pp. 3249–3254.(2019)

11. D. Silva, L. I. Carvalho, J. Soares, and R. C. Sofia.: "A Performance Analysis of Internet of Things Networking Protocols: Evaluating MQTT, CoAP, OPC UA," *Applied Sciences*, vol. 11, no. 11, Art. no. 11, Jan. (2021).

12. D. B. C. Lima, R. M. B. da Silva Lima, D. de Farias Medeiros, R. I. S. Pereira, C. P. de Souza, and O. Baiocchi.: "A Performance Evaluation of Raspberry Pi Zero W Based Gateway Running MQTT Broker for IoT," in *2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, Oct. 2019, pp. 0076– 0081.(2019)

13. W. Montalvo-Lopez, P. Catota, C. A. Garcia, and M. V. Garcia.: "Development of a Virtual Reality Environment Based on the CoAP Protocol for Teaching Pneumatic Systems," in *Augmented Reality, Virtual Reality, and Computer Graphics*, L. T. De Paolis, P. Arpaia, and P. Bourdot, Eds., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2021, pp. 528 –543.(2021)

14. R. Rákay, A. Galajdova, and M. Vagaš.: "Condition Monitoring Of Pneumatic Machinery Without Operation Interruption," *MM Science Journal*, vol. 2022, pp. 6026–6032, Nov. (2022).

15. A. P. Segara, R. Primananda, and S. R. Akbar.: " Implementasi MQTT (Message Queuing Telemetry Transport) pada Sistem Monitoring Jaringan berbasis SNMP (Simple Network Management Protocol)," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 2, Art. no. 2, (2018).