*Journal of Network Intelligence*
**Taiwan Ubiquitous Information**
 ISSN 2414-8105 (Online)
Volume **8**, Number **2**, May **2023**

# Message Loss Estimation Based on Probability Models for MQTT

Naiwang Lei

College of Science and Technology
Fujian Open University
Fuzhou Fujian, China
leinw@163.com

Linhao Huang*

College of Science and Technology
Fujian Open University
Fuzhou Fujian, China
hlh@fjrtvu.edu.cn

Hao Fang

College of Computer and Data Science
Fuzhou University
Fuzhou Fujian, China
hfang7@163.com

Chi-Hua Chen*

College of Computer and Data Science
Fuzhou University
Fuzhou Fujian, China
chihua0826@gmail.com

*Corresponding authors: Linhao Huang and Chi-Hua Chen

Received November 12, 2022; revised January 4, 2023; accepted March 28, 2023.

---

Abstract. *Internet of Things (IoT) has significantly changed our lives and the development of technology. Different types of smart terminal devices are connected to the Internet to achieve communications with each other. To establish the connections, communication protocols among different devices should be unified. Message Queue Telemetry Transport (MQTT) as a protocol based on publisher/subscriber and broker requires a small amount of bandwidth and guarantees the message passing process by providing different Quality of Services (QoS) levels. The message loss for the subscriber still exists during the downtime of the broker, which greatly affects service qualities. Currently, the message loss of MQTT is mainly collected by practical experiments which costs extra resources. Therefore, this paper proposed a statistical-based message loss estimation approach to analyze the message loss for the subscriber when MQTT supports different QoS levels. The proposed method can estimate the message loss directly from the derived formula. Numerical analysis is adopted to prove the availability of the message loss estimation formulas.*
**Keywords:** MQTT, QoS, IoT, Probability Model, Message Loss Estimation

---

1. **INTRODUCTION.** With the development of communication technology and sensor technology, the communication from Machine to Machine (M2M) is available which leads to the rise of Internet of Things (IoT). Through the fast data transmission, IoT provides fluent communications among different physical devices which exists everywhere in our daily lives. Amounts of physical devices connected to the Internet makes smart homes [1], smart buildings, smart cities [2], education [3], transportation [4], healthcare [5, 6] and many other fields to be intelligent. IoT greatly improves people's quality of life and impulses the development of the information technology. However, different devices require uniform protocols in each layer of network architecture to communicate with each other. To achieve fast, stable and secure communications among different types of physical entities, reliable IoT protocols should be designed to achieve low-cost message transmission [7] and deal with the message loss and security issues [8, 9, 10]. Several IoT protocols such as Message Queue Telemetry Transport (MQTT), Hyper Text Transfer Protocol (HTTP), Representational State Transfer (REST), and Constrained Application Protocol (CoAP) are proposed to enable communications among devices. In addition, they are supported by Transmission Control Protocol (TCP) except CoAP which is supported by User Datagram Protocol (UDP). HTTP ensures the reliable transmission but causes a large overhead [11]. Both REST and MQTT protocols need fewer network resources to maintain reliable communications among devices. Although CoAP demands less bytes to deliver messages and can achieve low latency [12], MQTT still outperforms on both message delivery delay and packet loss rate [13].

In M2M, MQTT [14] as a protocol within application-level layer is widely used among IoT devices. MQTT is capable to handle message passing among those devices with low hardware performance or under poor network conditions. MQTT is a broker-based protocol with two kinds of clients which play the roles of publisher and subscriber. Broker server acts like an intermediary receiving mes-sages from publisher and transferring to subscriber [15]. During the forwarding process, message loss results in reducing the communication quality. Practical devices are considered as clients, message loss among devices usually happens during the downtime of the broker and the transmission process. MQTT ensures the message delivery quality by providing three different Quality of Services (QoS) levels [16]. The forms of QoS services are represented as at-most-once, at-least-once, and exactly-once deliveries (i.e. QoS Level 0, QoS Level 1, and QoS Level 2)[17]. There exists the most message loss when MQTT supports the first two QoS levels (i.e. QoS Level 0 and QoS Level 1). When broker is crashed, messages from publisher are not able to transfer immediately to subscriber even lost. Message loss estimation is significant for testifying the performance. In addition, message loss of MQTT is usually collected by practical experiments [18]. The collection process usually costs extra resources to calculate the statistics of message loss. The preferable way to estimate the message loss is to derive formulas to obtain the amounts directly.

Therefore, this paper proposes a novel statistical-based message loss estimation for the subscriber in MQTT providing three QoS levels services. With hypotheses on the probability distribution of the broker crashed and its repair time, the message loss can be calculated directly through close-form formulas. The estimation of message loss is regarded as an performance indicator when MQTT supports different QoS levels. Moreover, numerical analysis is applied to prove the validity of derived message loss estimation formulas. The main contributions of this paper are summarized as follows.

1) A novel message loss estimation method based on statistical inference. Through the calculation of probability of broker crashed and the exception of message loss,

message loss of the subscriber is estimated in theoretical when MQTT provides different QoS level services.

2) Numerical analysis is adopted to verify the derived message loss from the statistical inference. The experiments assume the time of repairing the crashed broker is Normal Distribution (ND) and the broker crash frequency is Poisson Distribution (PD). Under reasonable assumptions, the proposed method effectively estimates message loss for the subscriber in MQTT of different QoS levels.

The rest paper is organized as follows. In Section 2, basic message transmission processes of MQTT supporting different QoS levels are introduced and different scenarios of message loss are also demonstrated. Moreover, the details of hypothesis in statistical inference are illustrated. In Section 3, the specific derivation process of message loss estimation is devised. Numerical analysis is adopted to testify the hypotheses in this chapter. Finally, the conclusions are presented to summarize the work of this paper and discuss future studies.

## 2. MESSAGE LOSS OF QOS LEVELS BASED ON MQTT.

Message transmission process based on MQTT protocol includes two phases: 1) publisher to broker and 2) broker to subscriber. During the process, the stability of broker operation is the most significant. When broker stops ser-vices for some reasons, the subscriber is not able to re-ceive messages from the publisher. Besides, different QoS levels influence the quality of the subscriber receiving messages. In Subsection 2.1, the message forwarding procedure based on MQTT under different QoS levels is introduced. In Subsection 2.2, specific message loss scenarios of the subscriber are demonstrated.

### 2.1. MQTT Messaging Process with Three QoS Levels.

In MQTT, three QoS levels are supported to achieve different qualities of message distribution. In QoS Level 0, both publisher and broker send publish messages only once and they do not confirm whether the subscriber gets the message. As shown in Figure 1, when the publish message is sent, the storage is released. Therefore, this situation is also named at-most-once. The circumstance of QoS Level 1 is called at-least-once, which means the subscriber can receive at least one message from publisher. QoS Level 1 uses status check message PUBACK to confirm that the subscriber receives at least one message, the specific process is shown in Figure 2. When publisher successfully sends a message to broker, the broker will store it and publish to the subscriber who subscribes the topic from the publisher. If the subscriber gets the message, a PUBACK message is returned to broker and the broker will delete the message and inform publisher by sending another PUBACK. In addition, the broker only stores the latest one message from publisher when there is no subscriber online ready to receive a new message. However, the loss of check message PUBACK will cause repeating sending the message because the sender does not get the confirmation. MQTT supporting QoS Level 2 considers the drawbacks of the first two QoS levels and applies four-way handshake mechanism. The message transmitted from publisher to broker first and the broker will return a PUBREC packet to inform publisher packet arriving. Then, the publisher sends another packet PUBREL for re-check. When broker receives PUBREL and send final complete packet PUBCOMP, the full message passing process is successfully done. The same process is applied when broker communicates with the subscriber. The full process is shown in Figure 3. It is obvious that MQTT under QoS Level 2 provides the most reliable message passing but increases overloads and delays.

### 2.2. Message Loss Scenarios.

In QoS Level 0, the subscriber's message loss is corresponding to the broker downtime. Specifically, the publisher will only send the message once to the broker and does not confirm whether the broker get the message. When the


Figure 1. MQTT - QoS Level 0.

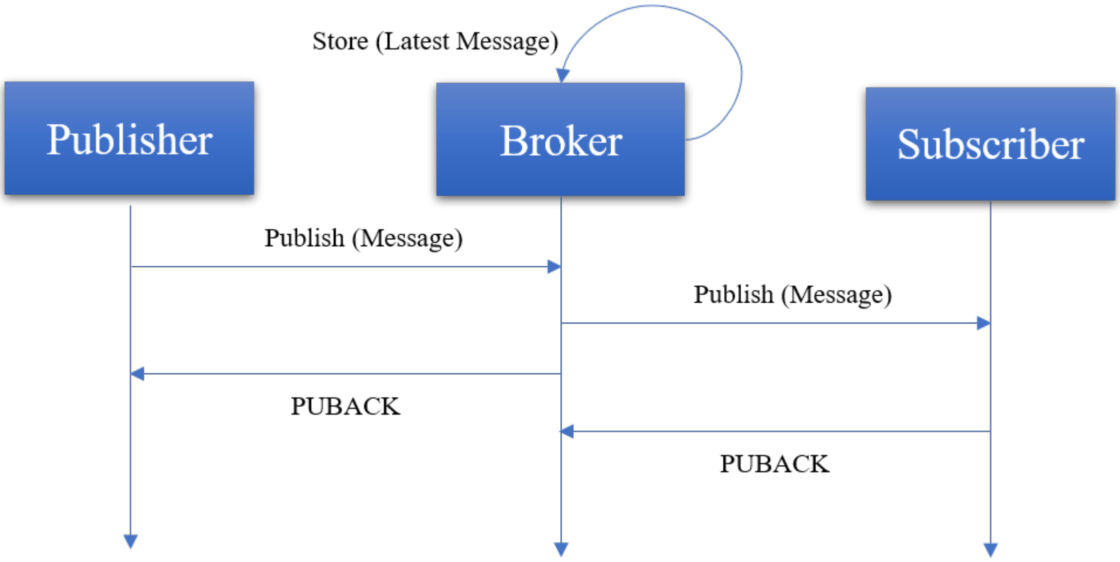


Figure 2. MQTT - QoS Level 1.

broker is offline, the message from publisher is lost and the subscriber cannot receive the message as well. During the downtime of broker, the subscriber will lose all messages from the publisher. When MQTT sup-ports QoS Level 1, the publisher will keep sending mes-sages to the broker if the subscriber does not receive it. The broker will store the latest message from the publisher and send it to the subscriber. Therefore, message loss occurs when the broker or the subscriber is down-time. In the situation of QoS Level 2, the broker stores all messages from the publisher and applies four-way hand-shake, which avoid the message loss.

3. **MESSAGE LOSS ESTIMATION FOR THE SUBSCRIBER IN MQTT.** Message loss estimation for the subscriber in MQTT under different network circumstances is difficult. This pa-per proposes a novel statistical-based message loss estimation for the subscriber. In Subsection 3.1, Some hypotheses in deducing are introduced and the variables in the derived message loss estimation formulas are introduced as well. Subsequently, the specific derivation of message loss estimation equations under three QoS levels supported by MQTT are demonstrated in Subsection 3.2.

3.1. **Hypotheses and Variable Definitions.** There are two main hypotheses in this paper. The frequency of broker crashed is a significant factor in message loss estimation. The first hypothesis is that the prob-ability distribution of broker crashed is assumed as Poisson Distribution (PD). The average frequency of broker crashed in certain time interval is denoted as $\lambda$. The time taken from client to broker depends on different types of IoT devices. Besides, the transmission time from client to broker is assumed as $c$ in this paper. The time distribution of repairing broker is Normal Distribution (ND), which is the second hypothesis. For ND, $\mu$ and $\sigma$ represent the mean and standard deviation, respectively.

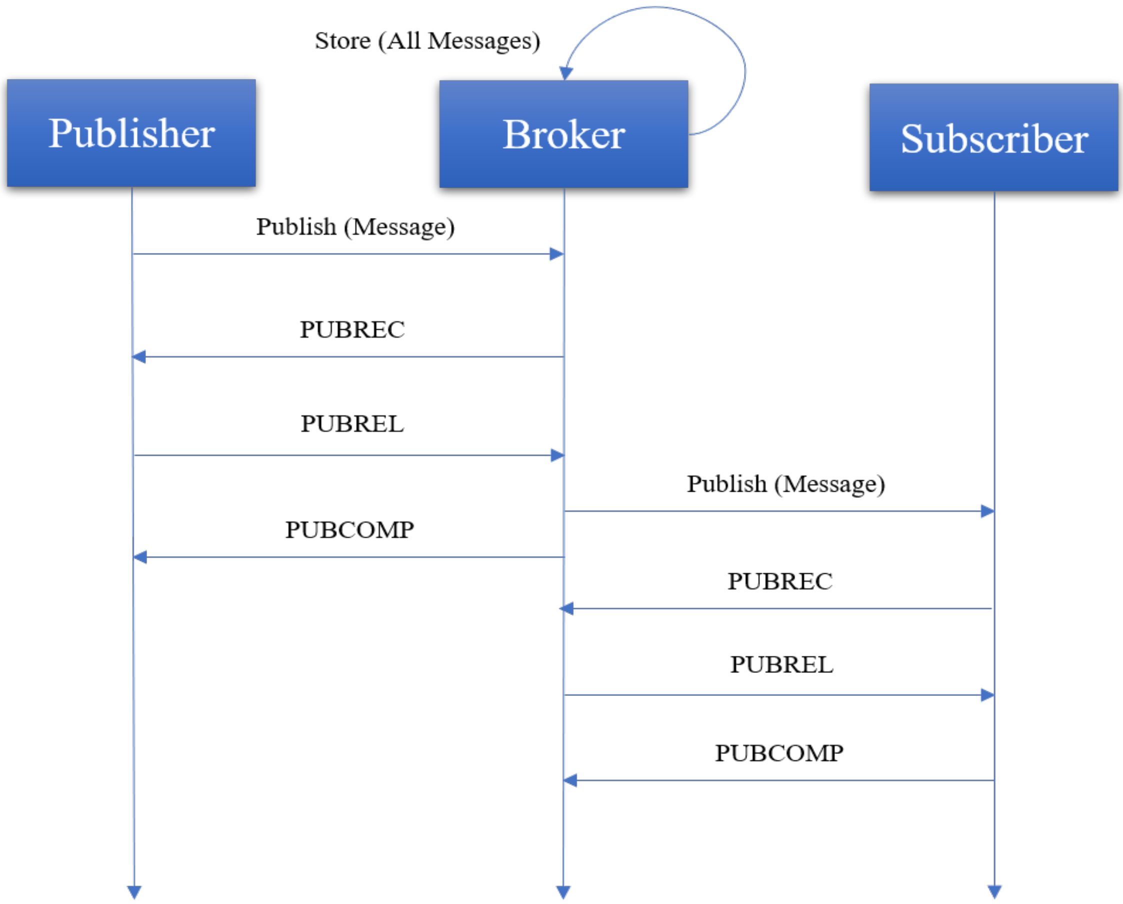


FIGURE 3. MQTT - QoS Level 2.

3.2. **Message Loss Estimation Based on Probability Models.** MQTT supports three QoS levels to achieve different quality of message transmission. Under different QoS levels, the amount of message loss is distinct. With the increase of QoS level, message loss declines for the sub-scriber. The statistical inference of message loss for the subscriber in three QoS levels is illustrated as follows.

3.2.1. *QoS Level 0.* QoS Level 0 supported by MQTT is the most unreliable when the message is transferred from the publisher to the subscriber. As shown in Equation (1), $i$ represents the times of broker crashed in the unit time, and the second term denotes its probability under PD with $\lambda$. The third term is the expectation of message loss during the period $t$ of broker crashed. Since QoS Level 0 does not require the confirmation of message arriving, messages sent by the publisher are all lost in $t$. Besides, the number of message loss is denoted as $\frac{t}{c}$. Consequently, the total message loss for the subscriber is the summation of $i$ from 0 to the infinity, which includes all possible situations.

$$\begin{aligned}
&\sum_{i=0}^{\infty}(i \times Pr_p(i;\lambda) \times (\int_{t=-\infty}^{\infty} \frac{t}{c} \times Pr_p(t;\mu,\sigma^2)dt)) \\
=&\sum_{i=0}^{\infty}(i \times (\frac{\lambda^i e^{-\lambda}}{i!}) \times (\frac{1}{c} \times \int_{t=-\infty}^{\infty} t \times (\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2}(\frac{t-\mu}{\sigma})^2})dt)) \\
=&\frac{1}{c} \times (\int_{t=-\infty}^{\infty} t \times \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2}(\frac{t-\mu}{\sigma})^2}dt) \times (\sum_{i=0}^{\infty} i \times (\frac{\lambda^i e^{-\lambda}}{i!})) \\
=&\frac{1}{c} \times \mu \times \lambda = \frac{\lambda\mu}{c}
\end{aligned} \tag{1}$$

3.2.2. *QoS Level 1.* Compared to MQTT in QoS Level 0, the message loss decreases in QoS Level 1. MQTT utilizes packet PUBACK to confirm message passing successfully from the publisher to the subscriber. When the subscriber is offline, the publisher will continue sending messages to the broker. However, there is only one latest message from the publisher will be stored in the broker. Moreover, if the broker is down, messages from the publisher are all lost except the latest one. The subscriber will receive only one message from the broker after broker restart. In the Equation (2), the only difference from Equation (1) is in the third integral term. Since the broker will store one message for the subscriber in any situations, the number of message loss is $\frac{t}{c} - 1$. The derived message loss estimation for the subscriber is shown in Equation (2).

$$\begin{aligned}
&\sum_{i=0}^{\infty}(i \times Pr_p(i;\lambda) \times (\int_{t=-\infty}^{\infty} (\frac{t}{c} - 1) \times Pr_p(t;\mu,\sigma^2)dt)) \\
=&\sum_{i=0}^{\infty}(i \times (\frac{\lambda^i e^{-\lambda}}{i!}) \times (\frac{1}{c} \times \int_{t=-\infty}^{\infty} (t-c) \times (\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2}(\frac{t-\mu}{\sigma})^2})dt)) \\
=&\frac{1}{c} \times (\int_{t=-\infty}^{\infty} (t-c) \times (\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2}(\frac{t-\mu}{\sigma})^2})dt) \times (\sum_{i=0}^{\infty} i \times (\frac{\lambda^i e^{-\lambda}}{i!})) \\
=&\frac{1}{c} \times (\mu - c) \times \lambda = \frac{\lambda(\mu - c)}{c}
\end{aligned} \tag{2}$$

3.2.3. *QoS Level 2.* To ensure the transmission quality, MQTT adopting QoS Level 2 avoids message loss during the transmission process. The broker will store all messages from the publisher when the subscriber is not ready to receive the message. According to the strict four-way handshake confirmation mechanism, the publisher will check whether the message arrived at the broker. Therefore, during the downtime of the broker, the publisher keeps sending messages which are not received at the broker. In QoS Level 2, MQTT ensures the reliability of message transmission. The estimation of message loss of MQTT in QoS Level 2 is shown in Equation (3).

$$\sum_{i=0}^{\infty}(i \times Pr_p(i;\lambda) \times (\int_{t=-\infty}^{\infty} 0 \times Pr_p(t;\mu,\sigma^2)dt)) = 0 \tag{3}$$

3.3. **Numerical Analyses.** To prove the validity of derived equations, numerical analysis is adopted to analyze the relationships of assumed variables. There are four variables in proposed message loss estimation approach, which are $\lambda$, $\mu$, $\sigma$ and $c$. Besides, the time of repairing the crashed broker is Normal Distribution (ND) and the broker crash frequency is Poisson Distribution (PD). The frequency of broker crashed in unit time is

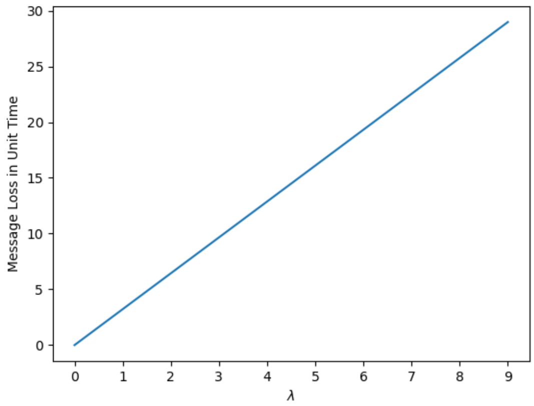


FIGURE 4. Message Loss with different $\lambda$ - QoS Level 0.

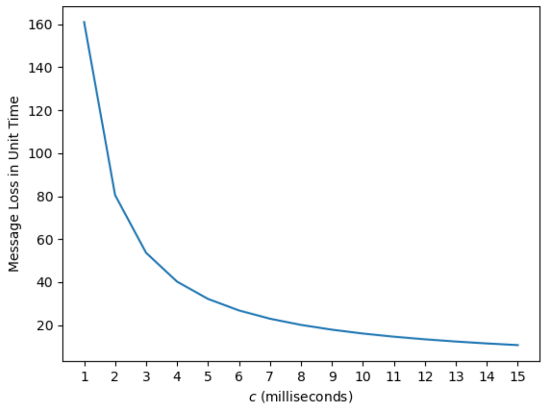


FIGURE 5. Message Loss with different $c$ - QoS Level 0.

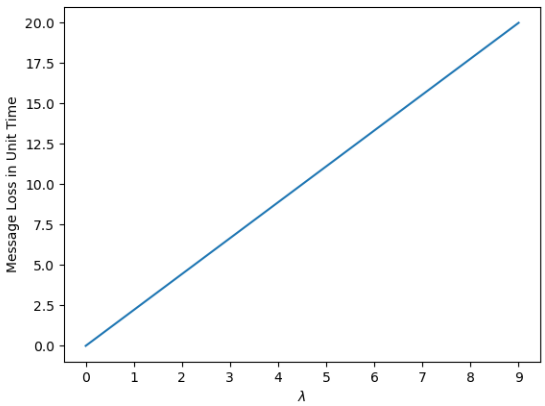


FIGURE 6. Message Loss with different $\lambda$ - QoS Level 1.

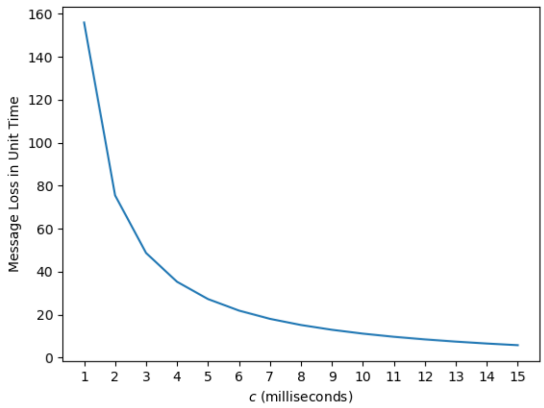


FIGURE 7. Message Loss with different $c$ - QoS Level 1.

denoted as $\lambda$. $\mu$ and $\sigma$ represent the mean and standard deviation of the ND. The message transmission time from physical devices to the broker is represented as $c$. In practical experiments, a Mosca broker is built to collect the repairing time of broker. Scenarios of broker crashed have been simulated for 19285 times. Therefore, the practical $\mu = 32.2008$ milliseconds and $\sigma = 0.7975$ milliseconds are counted based on the collected data.

In QoS Level 0, $c$ is fixed as 10 milliseconds at first and Figure 4 shows that the message loss increases when $\lambda$ rises from 0 to 9. With the grow of the frequency of broker crashed, message loss for the subscriber naturally increases. The message estimation is reasonable. On the contrary, the message loss declines when $c$ increases and $\lambda = 5$ is unchanged (Shown in Figure 5). When the transmission process takes more time, the message from the client to the broker is less in unit time and the message loss is less as well. Based on the practical $\mu$ and $\sigma$, the message loss for the subscriber in MQTT with QoS Level 0 is represented as $32.2008 \times \frac{\lambda}{c}$.

In QoS Level 1, the message loss has similar variation tendency when $\lambda$ and $c$ change respectively, which are shown in Figures 6 and 7. When $\lambda$ increases, the message loss in unit time becomes larger. The growth of $c$ incurs the decreases of message loss in QoS level 1. The settings of $\lambda$ and $c$ are the same as MQTT in QoS Level 0. Furthermore,

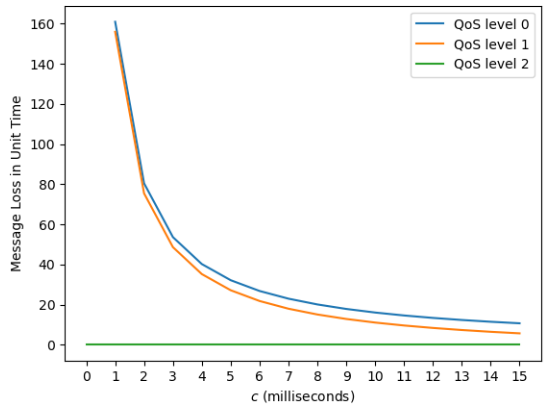


FIGURE 8. Message Loss with different $\lambda$ - MQTT.

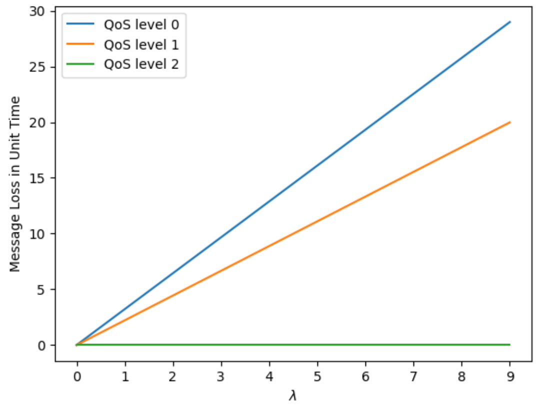


FIGURE 9. Message Loss with different $c$ - MQTT.

$\frac{\lambda(32.2008-c)}{c}$ is the message loss for QoS Level 1. The experimental results demonstrate that the message loss is direct ratio to $\lambda$ and inversely proportional to the value of $c$. In addition, $\mu$ should be greater than $c$ which promises the non-negative of the message loss.

In QoS Level 2, the four-way handshake mechanism has proved no message loss in the transmission process. Therefore, the statistical estimation result is 0. According to the same settings of $\mu$ and $\sigma$, the results of estimated message loss variation in QoS Level 0, 1 and 2 are displayed in Figures 8 and 9. The message loss is 0 when QoS Level 2 is supported by MQTT. Moreover, it is obvious that the message loss is less when MQTT supports QoS Level 1. In general, the proposed method is capable to reasonably estimate message loss for the subscriber in MQTT of different QoS levels.

## 4. CONCLUSIONS.

This study proposes a novel statistical-based message loss estimation approach to logically analysis the mes-sage loss of the subscriber in MQTT. With the increasing of QoS level provided by MQTT, the message loss during the transmission process is reduced. The proposed method makes hypotheses on the probability distribution of the broker crashed and its repair time, which are PD and ND, respectively. In different QoS levels, the circumstances of message loss are different. The proposed method adopts the expectation and probability of the message loss to calculate the total message loss estimation for the subscriber. The derived message loss estimation results adopt numerical analysis to testify the correctness. The analysis shows that the estimated message loss of the subscriber increases with the rise of the broker crashed frequency and declines when MQTT providing higher level QoS services. Besides, the longer time of message transmission results in less message loss. In the future work, this study can be a theoretical reference of the mes-sage loss estimation for the subscriber when MQTT supporting different levels of QoS. The hypotheses can be replaced with other probability distributions for different application scenarios.

## REFERENCES

[1] T.-Y. Wu, F. Kong, L. Wang, Y.-C. Chen, S. Kumari, and J.-S. Pan, "Toward smart home authentication using puf and edge-computing paradigm," *Sensors*, vol. 22, no. 23, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/23/9174

[2] S. Nižetić, P. Šolić, D. López-de-Ipiña González-de-Artaza, and L. Patrono, "Internet of things (iot): Opportunities, issues and challenges towards a smart and sustainable future," *Journal of Cleaner Production*, vol. 274, p. 122877, 2020.

[3] M. Al-Emran, S. I. Malik, and M. N. Al-Kabi, “A survey of internet of things (iot) in education: Opportunities and challenges,” *Toward social internet of things (SIoT): enabling technologies, architectures and applications*, pp. 197–209, 2020.

[4] F. Zantalis, G. Koulouras, S. Karabetsos, and D. Kandris, “A review of machine learning and iot in smart transportation,” *Future Internet*, vol. 11, no. 4, p. 94, 2019.

[5] C.-M. Chen, Z. Chen, S. Kumari, and M.-C. Lin, “Lap-ioht: A lightweight authentication protocol for the internet of health things,” *Sensors*, vol. 22, no. 14, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/14/5401

[6] C.-M. Chen, S. Liu, S. A. Chaudhry, Y.-C. Chen, and M. A. khan, “A lightweight and robust user authentication protocol with user anonymity for iot-based healthcare,” *Computer Modeling in Engineering & Sciences*, vol. 4, pp. 307–329, 2022.

[7] K. Ponnusamy and N. Rajagopalan, “Internet of things: a survey on iot protocol standards,” *Progress in Advanced Computing and Intelligent Engineering*, pp. 651–663, 2018.

[8] T.-Y. Wu, L. Yang, Z. Lee, S.-C. Chu, S. Kumari, and S. Kumar, “A provably secure three-factor authentication protocol for wireless sensor networks,” *Wireless Communications and Mobile Computing*, vol. 2021, 2021.

[9] C.-M. Chen, X. Deng, W. Gan, J. Chen, and S. Islam, “A secure blockchain-based group key agreement protocol for iot,” *The Journal of Supercomputing*, vol. 77, no. 8, pp. 9046–9068, 2021.

[10] T.-Y. Wu, L. Wang, X. Guo, Y.-C. Chen, and S.-C. Chu, “Sakap: Sgx-based authentication key agreement protocol in iot-enabled cloud computing,” *Sustainability*, vol. 14, no. 17, p. 11054, 2022.

[11] T. Yokotani and Y. Sasaki, “Comparison with http and mqtt on required network resources for iot,” in *2016 international conference on control, electronics, renewable energy and communications (ICCEREC)*. IEEE, 2016, pp. 1–6.

[12] 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, p. 4879, 2021.

[13] D. Thangavel, X. Ma, A. Valera, H.-X. Tan, and C. K.-Y. Tan, “Performance evaluation of mqtt and coap via a common middleware,” in *2014 IEEE ninth international conference on intelligent sensors, sensor networks and information processing (ISSNIP)*. IEEE, 2014, pp. 1–6.

[14] N. De Caro, W. Colitti, K. Steenhaut, G. Mangino, and G. Reali, “Comparison of two lightweight protocols for smartphone-based sensing,” in *2013 IEEE 20th Symposium on Communications and Vehicular Technology in the Benelux (SCVT)*. IEEE, 2013, pp. 1–6.

[15] M. B. Yassein, M. Q. Shatnawi, S. Aljwarneh, and R. Al-Hatmi, “Internet of things: Survey and open issues of mqtt protocol,” in *2017 international conference on engineering & MIS (ICEMIS)*. Ieee, 2017, pp. 1–6.

[16] A. S. Sadeq, R. Hassan, and A. Mahdi, “Enhanced mqtt for providing qos in internet of things (iot): a study,” *Advanced Science Letters*, vol. 24, no. 7, pp. 5199–5203, 2018.

[17] A. Al-Fuqaha, A. Khreishah, M. Guizani, A. Rayes, and M. Mohammadi, “Toward better horizontal integration among iot services,” *IEEE Communications Magazine*, vol. 53, no. 9, pp. 72–79, 2015.

[18] D. G. Roy, B. Mahato, D. De, and 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.