International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 / https://doi.org/10.14445/22315381/IJETT-V71I3P223

Volume 71 Issue 3, 227-235, March 2023

Original Article

Evaluation of Delay Parameter of MQTT Protocol

Shital Pawar[1], Nibedita Panigrahi[2], Jyothi A.P.[3], Meghana Lokhande[4], Deepali Godse[5], D. B. Jadhav[6]

[1]*Department of Computer Engineering, Bharati Vidyapeeth's College of Engineering for Women, Pune, Maharashtra, India*
[2]*Department of Information Science and Engineering, RV Institute of Technology and Management, Bengaluru, Karnataka, India*
[3]*Department of Computer Science and Engineering, Ramaiah University of Applied Sciences, Bengaluru, Karnataka, India*
[4]*Department of Computer Engineering, Pimpri Chinchwad College of Engineering, Pune, India*
[5]*Department of Information Technology, Bharati Vidyapeeth's College of Engineering for Women, Pune, Maharashtra, India*
[6]*Department of Mechanical Engineering, Bharati Vidyapeeth (Deemed to be University) College of Engineering, Pune, Maharashtra, India*

[1]*Corresponding Author : shitalp16@gmail.com*

Received: 28 October 2022 Revised: 08 February 2023 Accepted: 22 March 2023 Published: 25 March 2023

Abstract *- In IoT, the number of devices is growing exponentially. These huge numbers of IoT devices are used for providing information from various sources in a variety of applications from different domains. The essential requirement is choosing the best service based on Quality of Service at design time or at run time whenever the presently running service starts to degrade. This is specifically necessary for safety-related applications like healthcare, industrial automation, etc., where the service failure will have a critical impact. The internet of Things consists of a variety of physical objects which are connected to the sensors through the internet. There are many protocols, like MQTT, CoAP, XMPP, etc., available in IoT for exchanging data between devices. The precise evaluation of delay plays a vital role in providing Quality Service in IoT. As MQTT is lightweight, uses a small amount of power and is an ideal protocol for IoT; the MQTT protocol has been selected for Quality-of-Service evaluation. In this paper, the delay parameter of the MQTT protocol has been evaluated for an Industrial IoT application with two different MQTT brokers, namely Mosquitto and Paho.*

Keywords *- IoT application, End to end delay, MQTT protocol, MQTT broker, Quality of Service (QoS).*

1. Introduction

The Technology where the physical entities are endowed with virtual representation, which permits the entities to interchange contextual information to organize actions within, is known as the Internet of Things (IoT). They would possess a prompt response and good reply to the changes in the environment which they would also utilize the resources very efficiently [1]. The industry is one application domain of the stream IoT. Recently the number of IoT devices has been increasing rapidly. The actuators, networks, control production systems, sensors, and services to join would all comprise the Industrial Internet of Things. Supply chains could be observed and optimized through the incorporation of IIoT. In this phenomenon, the detection of machine failures, prevention of production delays which cause revenue loss, injuries to employees, and prevention of equipment loss can be carried out. Apart from this, their history, production process, identity, and documentation can be known by their smart products. They can also gather information and can be utilized by the customers. Further Industrial Internet of Things would also permit individuality, resource-saving production and flexibility [2].

The aim of Industrial IoT is to upgrade the production process by linking the machines with the support of data processing to permit the analytical activities to predict maintenance requirements, thereby potentially causing considerable cost deduction and attaining high production. Detecting machinery failures is essential for controlling injuries and revenue loss to employees. The Industrial Internet of Things probably allows flexibility, resource-saving, individuality, etc. Because of this, the Industrial IoT contains different security attacks like industrial espionage, sabotage, DoS attacks etc.[3][4].

Various protocols like Message Queuing Telemetry Transport (MQTT) are used to transmit the information in the IoT network. From the prevailing pieces of literature, it was found that MQTT is regarded as an effective candidate for its lightweight features and its working ability in constrained power and memory devices [5][6]. Further, the MQTT broker is regarded as the predominant component of MQTT-oriented IIoT since it provides several user services. It was observed that the MQTT protocol is found to flood the broker that, causes the DoS attack.

The hacker teases the broker and transmits the false control or the data messages during the process of the DoS attack. Hence automatic recoverability from the DoS attack, the influence of broker failure, and the time duration for recoverability are the important security constraints in the MQTT protocol. It is necessary to examine and evaluate the QoS parameters of IoT communication protocols like MQTT, AMQP, XMPP, CoAP etc. So, the QoS parameter benchmarking is crucial because these parameters can impact the overall production line performance if any security threat happens [7-9]. Very few researchers have concentrated on the Quality of Service evaluation of IoT applications. The performance of IoT applications can be evaluated by evaluating QoS parameters like delay, latency, throughput etc. In this paper, delay parameters of the MQTT protocol have been evaluated for industrial IoT applications. As a machine to machine communication in MQTT protocol is performed through brokers, in this paper, the performance of paho and mosquitto brokers is examined by evaluating the delay parameter of MQTT protocol communication.

2. Background and Related Work

The Internet of Things (IoT) consists of sensors, devices, smart nodes etc., which can communicate with each other without any human interruption. The different objects can function independently with reference to other objects. IoT nodes deliver lightweight data, gather and extract the data from authorized cloud resources and analyze the gathered data for making precise decisions. It is really a challenging task to develop an IoT application because of certain reasons like the great difficulty in distributed computing, the generic guidelines required to manage the lower-level communication and to facilitate the higher-level implementation being insufficient, the use of multiple programming languages and the variety of communication protocols etc. [10][11].

The number of IoT applications and devices is growing rapidly, so cyber-attacks are also increasing and causing severe risks to the safety and security of IoT applications. In IoT, protocol communication is based on request and response, so providing security to these request and response messages is essential. As the number of cyberattacks, information leaks and vulnerabilities is increasing, IoT device manufacturers and researchers are now trying to design systems to provide security to the data flow between the devices, find new vulnerabilities, and provide privacy and security to the devices and users. Even though the researchers are trying to address IoT's privacy and security-related issues, many investigations are in the inceptive phases or lacking in applicability. Most of the security-related issues are still unsolved. In addition, recourse scheduling is becoming difficult, particularly for communication protocols. Underutilization of resources becomes an issue when essential communication protocols are of multiple types [12-14].

Moving towards achieving the service with improved QoS is quite difficult as the Quality of Service consists of various parameters like throughput, latency, jitter, response time, delay etc. In addition to these, there are also a variety of QoS parameters which are tough to measure, for example, security and functional stability of service. It is essential to do the research to find alternative methods for data preprocessing and comparing different users. IoT applications provide a variety of services through the number of IoT devices which are mobile and resource constrained. So, the main research problem is that the IoT devices have limited resources and how to estimate the QoS on the user side, to ensure the ideal selection, conformation, and modification of IoT services [15-17].

Gary White et al. developed the algorithm for QoS prediction and tested it on the QoS dataset. He concluded that it is essential to do the benchmarking of QoS parameters for real-time IoT applications [28]. Dmitrii Dikii et al. monitored the anomalous behavior of IoT devices by using machine learning and concluded that it is essential to evaluate the QoS parameters of IoT applications to analyze the IoT network traffic accurately [19]. Trent N. Ford et al. have carried out the performance evaluation of the MQTT protocol on raspberry pi. He has performed the throughput analysis in a specific tested environment. He has evaluated the data transfer time for three different raspberry pi devices. He has examined the effect of the DoS attack on the data transfer time. He has evaluated the performance of three different Raspberry pi devices. He concluded that the performance of various MQTT brokers could be evaluated further [20]. Fatma Hmiss et al. performed MQTT communication modeling using mist computing and evaluated the QoS parameters like energy consumption, delay, etc. He specified that there is a need to evaluate the various QoS parameters for evaluating the performance of MQTT communication [29]. Minhaj Khan et al. have discussed the security needs, recent threats and their solutions. He observed that the DoS attacks affect the delivery of services provided by IoT devices, which greatly impacts QoS parameters [22].

3. Methodology

In this research, the delay parameter of QoS was considered. The delay parameter becomes influenced by the security issues of an IoT application. MQTT protocol is popularly used in Industrial IoT as it ensures the delivery of messages and implements the Quality of Service. So, in this paper, the delay parameter of the MQTT protocol has been evaluated for IoT applications. In fig. 2, the flowchart represents the procedure of delay evaluation for the MQTT protocol. MQTT protocol provides three levels of Quality of Service. As shown in fig. 2, the delay value of MQTT protocol has been evaluated on a real-time Industrial IoT system. Suppose a delay in normal operating conditions occurs.

In that case, the delay for that specific protocol request is recorded and reported to the control administration panel, along with the reason for the possible threat. If no delay was found in the protocol communication, the QoS value was recorded.

3.1. MQTT Communication

MQTT is a standard IoT protocol which is based on the Publish-Subscribe model. As shown in fig. 1, the Publish–Subscribe model decouples the publisher (client who is sending a message) from the subscriber (client who is receiving the message). In other words, the publisher and subscriber are not communicating with each other directly. The communication between them is handled by the other entity called a broker. Figure 1 illustrates the publish-subscribe communication model of the MQTT protocol. The temperature sensor acts as a publisher and sends the temperature reading to the broker. The client acts as a subscriber and subscribes to these temperature readings from the broker. The special features of the MQTT protocol, like Lightweight, easy to implement, small in size, data packets small in size, and low power usage, make this protocol perfect for IoT applications [23].

3.2. MQTT QoS levels

It offers the following three levels of QoS. [23]:

- QoS0: At most once – This is the fastest way to transfer messages. The message delivered from the client through the broker will not be acknowledged.
- QoS1: At least once – The client will receive an acknowledgement from the broker for every message receipt. If an expected acknowledgement is not received within the specified time, then the client will resend the message.
- QoS2: Exactly once – It is the slowest and safest way to transfer messages. It guarantees that every message is received exactly once by the receiver. It is generally preferred for banking applications.

QoS is the key feature of the MQTT protocol, which gives the client the power to select the service level based on network reliability and application logic. Example: The mobile application will use the level QoS0 when connected to a reliable wireless network, while it will choose QoS1 when it connects to the mobile network [25,26,30]. The evaluation of QoS parameters is useful to ensure the performance of IoT applications.

4. IoT Application

As hazardous chemical plants are risky, Industrial IoT deployment is suitable for a few plants where continuous fumes and gas evaporation is occurring. The Following fig. 3 shows the processing flow of a hazardous chemical mixer plant, including various sensors such as thermal sensor, cooling sensor, fume sensor and density sensor for specific operations. The MQTT-activated communication flow for publish–subscribe mode is shown schematically in fig. 3. The details of the sensors are provided. As any delay harms the production line, the client conducted tests to evaluate delay parameters per the testing strategy, and results for benchmarking was produced. The delay evaluation results are discussed in the Result section.

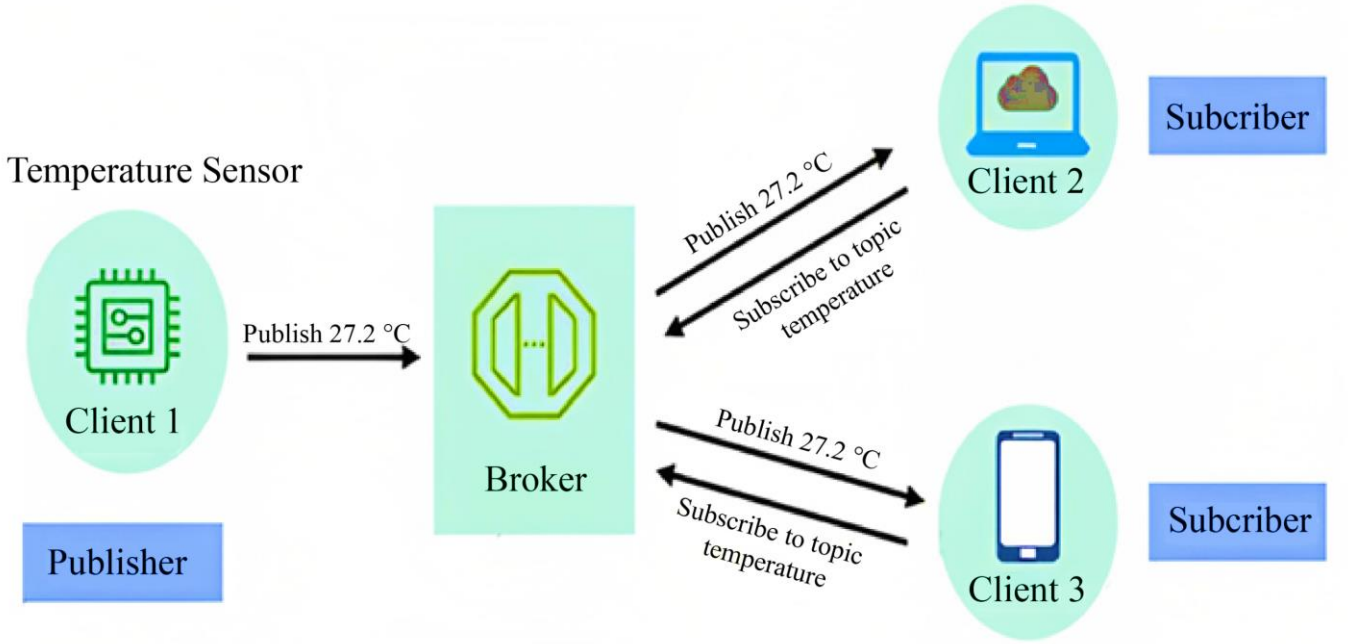

Fig. 1 Communication in MQTT with Publish-subscribe model

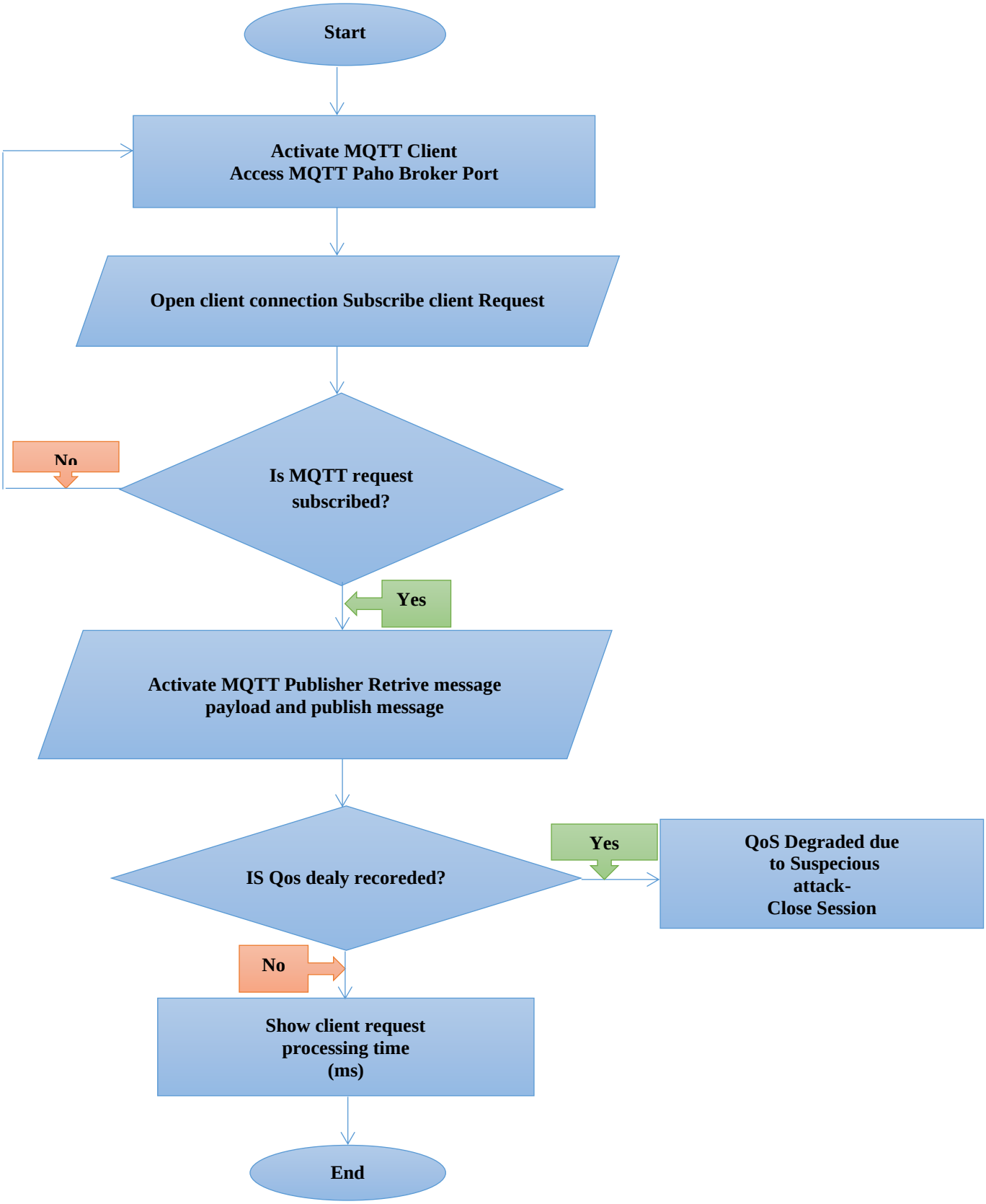

Fig. 2 Flowchart for evaluation of delay parameter for MQTT protocol

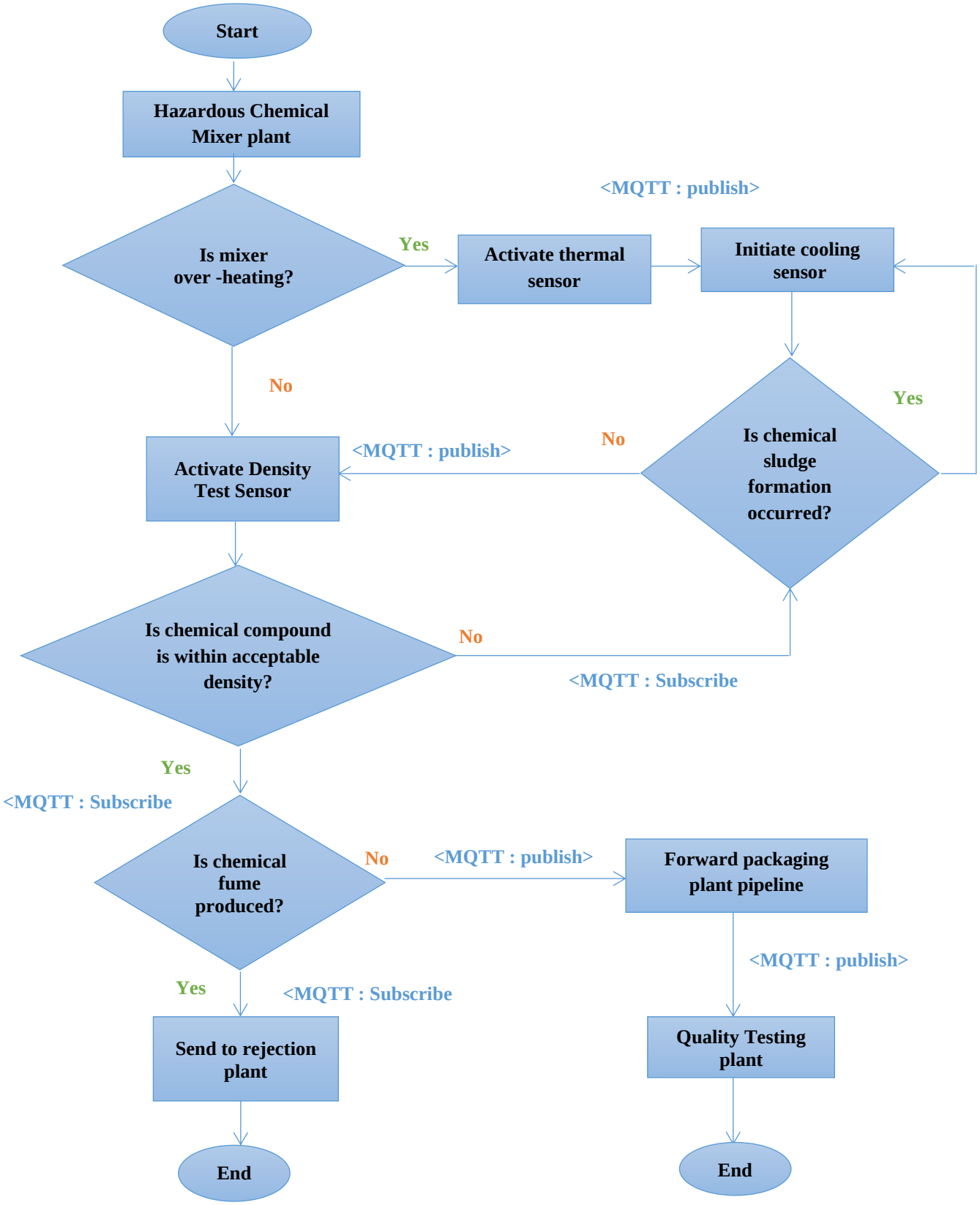

Fig. 3 MQTT communication flow for Chemical Mixer Plant Structure

5. Results and Discussion

The MQTT protocol uses the publish-subscribe model for communication where the publisher and subscriber are clients, and these clients are connected to the MQTT broker over the network. MQTT broker handles the communication between MQTT clients. In this paper, two MQTT brokers, namely, Mosquitto and paho, were considered for testing. The incremental load tests for these two brokers were conducted. In this case, a threshold delay value of 2 milliseconds was considered. The tests were conducted on Industrial IoT applications for both brokers. The benchmarking results of delay parameters on Industrial IoT applications for Mosquitto and Paho broker are shown in table1.

Table 1. Benchmarking results for the delay parameter

<table>
<tr><td colspan="2">Category of Test</td><td colspan="2">Performance</td><td>Type of Test: Load Test</td></tr>
<tr><td>Objective of Test</td><td colspan="4">“To check whether the MQTT broker can handle the given additional load for the specified time duration without overstepping the threshold value of delay parameter in the given tolerable message loss rate.”</td></tr>
<tr><td>Test Details</td><td>Testing Scenario 1: Testing against Mosquitto Server</td><td colspan="2">Testing Scenario 2: Testing against Paho Server</td><td>Considered time duration 2ms</td></tr>
<tr><td>Expected Performance</td><td colspan="4" rowspan="2">check
when {
at this specific time, T1:The tester sends many PUBLISH messages and
verify the rate of increase and
during the Meantime, after T1:
an entity receives many PUBLISH messages along with
topic_name identical to TOPIC and
payload identical to Retained__message
}
then {
The meantime, after T1:
an entity ensures and sends the PUBACK messages and
an entity ensures the Packet Loss Limit as well, as an entity ensures DELAY
}</td></tr>
<tr><td></td></tr>
<tr><td>Output</td><td colspan="4">Average PUBLISH/PUBACK delay in milliseconds (KPIx)</td></tr>
<tr><td>Results</td><td>Rate of Success for publishing the messages</td><td>Delay (Delay1) for a single message</td><td>Delay (Delay2) for multiple messages</td><td>Time considered</td></tr>
<tr><td>Values TC1- with Mosquitto broker</td><td>100%</td><td>0.998 milliseconds</td><td>0.999 milliseconds</td><td>2 milliseconds</td></tr>
<tr><td>Values TC2-with Paho broker</td><td>100%</td><td>0.92 milliseconds</td><td>0.93 milliseconds</td><td>2 milliseconds</td></tr>
<tr><td>Test (Pass/Fail)</td><td></td><td colspan="3">Pass</td></tr>
</table>

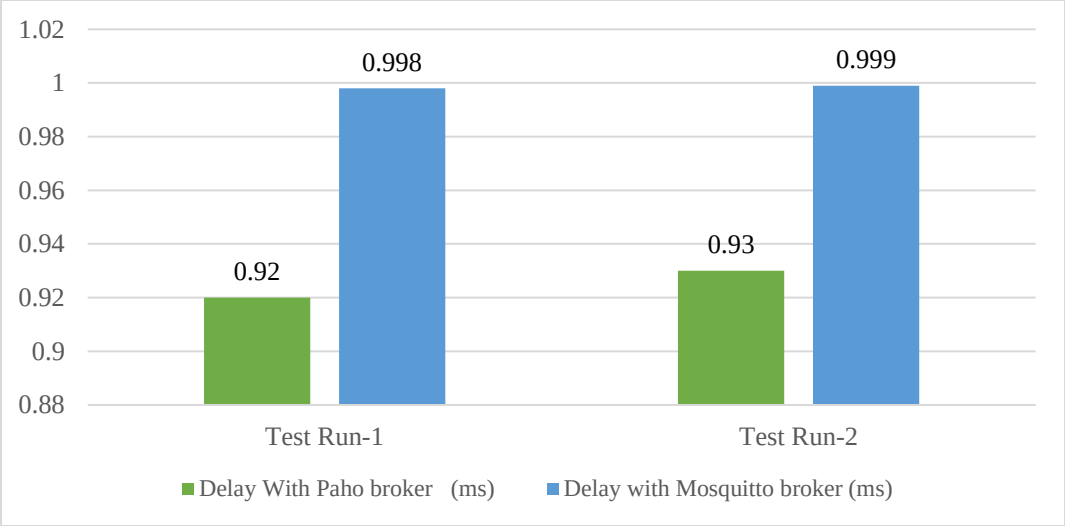

Fig. 4 Performance analysis for delay [Test Run-1→ for single request, Test Run-2→ for multiple requests]

The result of test run 1(for a single request) and test run 2 (for multiple requests) is represented in figure 4. Initially, only one request was transmitted through both brokers, and the delay value was recorded. In the case of the Mosquitto broker delay value was 0.998ms, and for the Paho broker, the delay value was 0.92ms. As both the delay values were less than the threshold delay value, this test case passed for both brokers. In the next phase, the multiple requests were transmitted through brokers, and the delay value was recorded. In this case, the delay value for the Mosquitto broker was 0.999ms, and for the paho broker was 0.93ms. Again, the delay values were less than the threshold value, so the test case passed for both brokers. From this experimentation, it was observed that the delay value for the paho broker was less than the mosquito broker in both test runs. Hence it was concluded that the delay in processing the single and multiple requests with the paho broker is less as compared to the mosquito broker. Many authors have evaluated the performance of mosquitto, Active MQ, Verne MQ, and Hive MQ brokers. In this paper, the performance of Mosquitto and Paho brokers has been evaluated through delay parameters.

6. Conclusion

IoT has now changed the traditional approach of living to an automated lifestyle. The various IoT applications like health monitoring, home automation, smart city, smart grid, Industrial IoT etc., use trillions of sensors and billions of devices that produce huge amounts of data daily. In IoT applications, communication occurs through the internet, which generates high demand for fundamental communication, greatly impacting the quality of Service. The QoS is an essential element for IoT systems that could be used to evaluate the performance and quality of IoT systems and devices. In connection with the security and performance of IoT applications, the QoS parameters like delay, jitter, packet loss etc., are most important. Recently, the number of IoT devices has been increasing tremendously on the network. Hence, it becomes essential to focus on the IoT system's QoS and smooth data transformation over the network. Hence, it is necessary to measure the QoS values for IoT applications. Any attack request in an IoT application causes a delay in processing the actual request. At the security level, any delay in request processing is risky. In this paper, the delay parameter has been evaluated for IoT applications with single and multiple requests on two different MQTT brokers. From the results, it is concluded that, as the delay value is less for the Paho broker than the Mosquitto broker, the Paho broker processes the requests faster than the Mosquitto broker. In future, the other QoS parameters like data loss, jitter, throughput, latency etc., can be evaluated to estimate the performance and security of IoT applications. An IoT system's performance may become unstable, leading to request response failure in streamlined protocol communication. Hence, the evaluation of QoS parameters can identify performance degradation by observing the variations in QoS parameter evaluation.

References

[1] Mauro A.A. da Cruz et al., "Performance Evaluation of IoT Middleware," *Journal of Network and Computer Applications*, vol. 109, pp. 53-65, 2018. [CrossRef] [Google Scholar] [Publisher link]

[2] Shital Pawar, and Suhas Patil, "A Novel Approach for Enhancement of Security through Evaluation of Quality-of-Service Parameters in Industrial Internet of Things," *2021 International Conference on Intelligent Technologies (CONIT)*, pp. 1-6, 2021. [CrossRef] [Google Scholar] [Publisher link]

[3] Aghabi N. Abosaif, and Haitham S. Hamza, "Quality of Service-Aware Service Selection Algorithms for the Internet of Things Environment: A Review Paper," *Array*, vol. 8, 2020. [CrossRef] [Google Scholar] [Publisher link]

[4] Rajat Verma, Namrata Dhanda, and Vishal Nagar, "Enhancing & Optimizing Security of IoT Systems using Different Components of Industry 4.0," *International Journal of Engineering Trends and Technology*, vol. 70, no. 7, pp. 147-157, 2022. [CrossRef] [Google Scholar] [Publisher link]
[5] Biswajeeban Mishra, "Performance Evaluation of MQTT Broker Servers," *Computational Science and Its Applications – ICCSA 2018*, pp. 599–609, 2018. [CrossRef] [Google Scholar] [Publisher link]
[6] Ivan Vaccari, Maurizio Aiello, and Enrico Cambiaso, "SlowITe, a Novel Denial of Service Attack Affecting MQTT," *Sensors*, vol. 20, no. 10, p. 2932, 2020. [CrossRef] [Google Scholar] [Publisher link]
[7] Rohini Temkar, and Anand Bhaskar, "Quality Assurance of IoT based Home Automation Application using Modified ISO/IEC 25010," *International Journal of Engineering Trends and Technology*, vol. 69, no. 2, pp. 92-101, 2021. [CrossRef] [Publisher link]
[8] Wasswa Shafik et al., "A Study on Internet of Things Performance Evaluation," *Journal of Communications Technology, Electronics and Computer Science*, no. 28, pp. 1-19, 2020. [Google Scholar]
[9] Daniel Silva et al., "A Performance Analysis of Internet of Things Networking Protocols: Evaluating MQTT, CoAP, OPC UA," *Applied Science*, vol. 11, no. 11, p. 4879, 2021. [CrossRef] [Google Scholar] [Publisher link]
[10] Talha Naeem Qureshi et al., "Enhanced Robustness Strategy for IoT in Smart Cities Based on Data Driven Approach," *Workshops of the International Conference on Advanced Information Networking and Applications, Springer, Cham*, pp. 1084-1096, 2019. [CrossRef] [Google Scholar] [Publisher link]
[11] Mónica Martí, Carlos Garcia-Rubio, and Celeste Campo, "Performance Evaluation of CoAP and MQTT_SN in an IoT Environment," *Proceedings 2019*, vol. 31, no. 1, p. 49, 2019. [CrossRef] [Google Scholar] [Publisher link]
[12] Biswajeeban Mishra, "Performance Evaluation of MQTT Broker Servers," *Computational Science and Its Applications – ICCSA 2018*, vol. 10963, pp. 599–609, 2018. [CrossRef] [Google Scholar] [Publisher link]
[13] Eric Gamess, Trent N. Ford, and Monica Trifas, "Performance Evaluation of a Widely Used Implementation of the MQTT Protocol with Large Payloads in Normal Operation and under a DoS Attack," *Proceedings of the 2021 ACM Southeast Conference*, pp. 154-162, 2021. [CrossRef] [Google Scholar] [Publisher link]
[14] Yuang Chen, and Thomas Kunz, "Performance Evaluation of IoT Protocols Under a Constrained Wireless Access Network," *International Conference on Selected Topics in Mobile & Wireless Networking (MoWNeT), IEEE*, 2016, pp. 1-7, 2016. [CrossRef] [Google Scholar] [Publisher link]
[15] Nogaye Lo, and Ibrahima Niang, "A Survey on QoS-Based Communication Protocols for IoT Systems," *Proceedings of the 3rd International Conference of Networking, Information Systems and Security*, 2020. [CrossRef] [Google Scholar] [Publisher link]
[16] M. Nasrul Aziz, Irit Maulana Sapta, and Siti Rochimah, "Security Characteristic Evaluation Based on ISO/IEC 25023 Quality Model, Case Study: Laboratory Management Information System," *Electrical Power, Electronics, Communications, Controls, and Informatics Seminar, IEEE*, 2018. [CrossRef] [Google Scholar] [Publisher link]
[17] Ajay Chaudhary, Sateesh K. Peddoju, and Kavitha Kadarla, "Study of Internet-of-Things Messaging Protocols used for Exchanging Data with External Sources," *14th International Conference on Mobile Ad Hoc and Sensor Systems, IEEE*, pp. 666-671, 2017. [CrossRef] [Google Scholar] [Publisher link]
[18] M. Kamalahhasan, "Applications of Neural Networks for Ranking of Web Services using QoS Metrics," *SSRG International Journal of Electronics and Communication Engineering*, vol. 1, no. 1, pp. 4-7, 2014. [CrossRef] [Google Scholar] [Publisher link]
[19] Dmitrii Dikii, Sergey Arustamov, and Aleksey Grishentsev, "DoS Attacks Detection in MQTT Networks," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 21, no. 1, pp. 601-608, 2021. [CrossRef] [Google Scholar] [Publisher link]
[20] Trent N. Ford, Eric Gamess, and Christopher Ogden, "Performance Evaluation of Different Raspberry Pi Models as MQTT Servers and Clients," *International Journal of Computer Networks and Communications*, vol. 14, no. 2, 2022. [CrossRef] [Google Scholar] [Publisher link]
[21] S.T. Akhilesh, and B Lalit Pattel, "Performance Analysis of WLAN Criterions for Video Conferencing Applications," *SSRG International Journal of Mobile Computing and Application*, vol. 3, no. 2, pp. 1-4, 2016. [CrossRef] [Publisher link]
[22] Minhaj Khan, and Khaled Salah, "IoT Security: Review, Blockchain Solutions, and Open Challenges," *Future Generation Computer Systems*, vol. 82, pp. 395-411, 2018. [CrossRef] [Google Scholar] [Publisher link]
[23] Victor Seoane et al., "Performance Evaluation of CoAP and MQTT with Security Support for IoT Environments," *Computer Networks*, vol. 197, p. 108338, 2021. [CrossRef] [Google Scholar] [Publisher link]
[24] L.H. Patil et al., "Voip Based Wifi Calling System," *SSRG International Journal of Computer Science and Engineering*, vol. 6, no. 9, pp. 7-9, 2019. [CrossRef] [Publisher link]
[25] Asaad Althoubi, Reem Alshahrani, and Hassan Peyravi, "Delay Analysis in IoT Sensor Networks," *Sensors*, vol. 21, no. 11, p. 3876, 2021. [CrossRef] [Google Scholar] [Publisher link]
[26] Murugan Sivaram, "IOT-Pattern-as-a-Service Model for Delay Sensitive IOT Integrated Applications," *The International Arab Journal of Information Technology*, vol. 18, no. 4, 2021. [CrossRef] [Google Scholar] [Publisher link]
[27] Elias M. Pinheiro, and Sérgio D. Correia, "Software Model for a Low-Cost, IoT oriented Energy Monitoring Platform," *SSRG International Journal of Computer Science and Engineering*, vol. 5, no. 7, pp. 1-5, 2018. [CrossRef] [Google Scholar] [Publisher link]
[28] Gary White et al., "IoT Predict: Collaborative QoS Prediction in IoT," *IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2018. [CrossRef] [Google Scholar] [Publisher link]

[29] Fatma Hmiss, and Soane Ouni, "An MQTT Brokers Distribution Based on Mist Computing for Real-Time IoT Communications," *Wireless Personal Communication*, 2021. [CrossRef] [Google Scholar] [Publisher link]

[30] Shital Pawar, and Dr. Suhas Patil, "Development of QoS Evaluation Algorithm for MQTT Protocol with Reference to Threat Model," *International Journal of Engineering and Advanced Technology*, vol. 8, no. 6, pp. 1557-1562, 2019. [CrossRef] [Google Scholar] [Publisher link]