Performance Evaluation of MQTT Communication with Heterogeneous Traffic

1st Ryohei Banno
Kogakuin University
Tokyo, JAPAN
banno@computer.org

***Abstract*—One of the protocols gaining popularity for Internet of Things (IoT) systems is MQTT. It is lightweight and has loose coupling nature derived from the publish/subscribe communication model. The capability of handling heterogeneous data is crucial in supporting cooperation among various applications and sensors. In this study, we evaluate the performance of MQTT communication with the combination of different payload sizes. Experimental result shows different tendencies of the hetero data capability by broker products.**

***Index Terms*—MQTT, Publish/subscribe, IoT**

I. Introduction

The Internet of Things (IoT) is rapidly expanding in many industries, including smart factories and warehouses. MQTT [1] is a suitable protocol for exchanging data in such IoT systems. It is based on the publish/subscribe model [2] so that it has loose coupling nature, i.e., it enables relationships among devices and applications to vary flexibly.

To develop a large-scale IoT system, the performance of an MQTT broker is a major concern. Since an MQTT broker needs to handle all messages from publishers to subscribers, it could be a bottleneck. Hence, clarifying the broker performance characteristics has been a practical topic, and various existing studies evaluate MQTT brokers [3]–[7].

However, the existing studies do not adequately consider heterogeneous traffic. IoT data is diversifying; evolving sensor devices enable us to obtain high-quality data like 4k/8k images, whereas some applications require small data such as temperature at short intervals for real-time monitoring. Considering an IoT platform for horizontal data integration among various sensors and applications, an MQTT broker could be a core component. Therefore, its performance for heterogeneous traffic should be clarified.

In this study, we evaluate multiple MQTT broker products to clarify the performance characteristics for heterogeneous traffic.

II. Related work

There are existing studies aiming to confirm MQTT broker performance. Mishra [7] conducted a fundamental evaluation for both public brokers and locally deployed brokers. Ronzani et al. [3] measured how distributed MQTT brokers can scale. Bender et al. [4] built a test framework independent of MQTT implementation and evaluated popular open-source implementations. Gemirter et al. [5] compared MQTT with AMQP and HTTP using a real-time smart city public data set. Different from these studies, we focus on heterogeneous traffic.

This work was supported by JST, PRESTO Grant Number JPMJPR21P8.

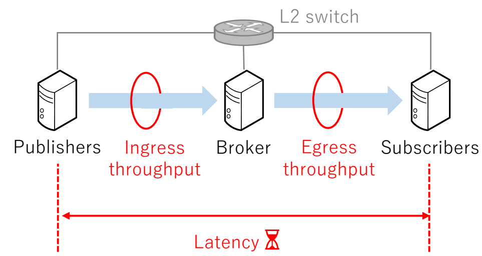

Fig. 1. Experimental environment

TABLE I
Machine information

Item	Specs
CPU	Core i9 10900K 3.7 GHz (10 cores, 20 threads)
Memory	64 GB
Network	1 GbE
OS	Ubuntu 20.04

TABLE II
Software versions

Type	Software	Version
Broker	Mosquitto	1.6.9
	HiveMQ	CE 2022.1
	VerneMQ	1.12.6.2
Client	MQTTLoader	0.8.4

III. Experimental configuration

To conduct experiments, we use three servers, as shown in Figure 1 and Table I. As preliminary testing, we confirmed that the round-trip-time by the ping command is 0.720 to 0.758 milliseconds, and the throughput by the iperf3 command is 913 to 926 Mbps among servers. We use three broker products, Mosquitto, HiveMQ, and VerneMQ, and a load-testing tool MQTTLoader [8], as listed in Table II.

We set three kinds of configurations as follows:

- Small data traffic: $1,024$ bytes payload size and up to $200,000$ msg/sec ingress messages.
- Large data traffic: 20 Mbytes payload size and up to 10 msg/sec ingress messages.

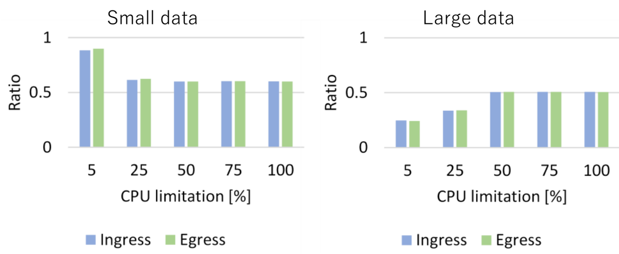

Fig. 2. Throughput ratio with Mosquitto

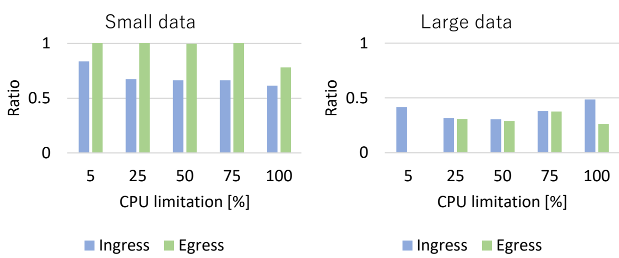

Fig. 3. Throughput ratio with HiveMQ

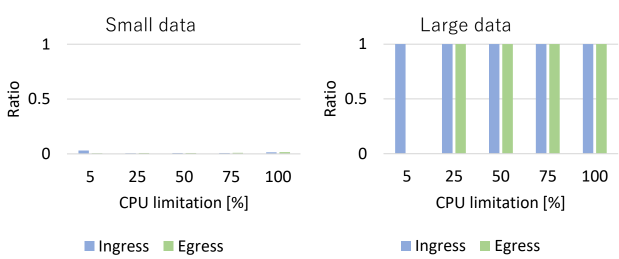

Fig. 4. Throughput ratio with VerneMQ

- Hetero data traffic: the combination of the above two.

In each of the small and large data configurations, the number of subscribers is five, where they compose a shared-subscription group to receive messages in parallel. Hetero data configuration combines these.

Regarding MQTT parameters, we set the protocol version to 5.0, the QoS level to zero, and the retain flag to false. The measurement metrics are ingress throughput, egress throughput, and latency from the publishers to the subscribers. Measurement time for each experiment is 70 seconds, including five seconds ramp-up/ramp-down times.

IV. Result and Discussion

We measured throughput with changing the available CPU resource of brokers by the cpulimit command. Note that only one core is enabled in this experiment. Figures 2, 3, and 4 show how the average throughput of hetero data changes compared to the homogeneous data cases, i.e., small and large data. For example, in the case of Mosquitto with a five percent CPU limitation, the small data ingress throughput is about 88 percent of the homogeneous small data case, whereas the large data throughput is about 25 percent of the homogeneous large

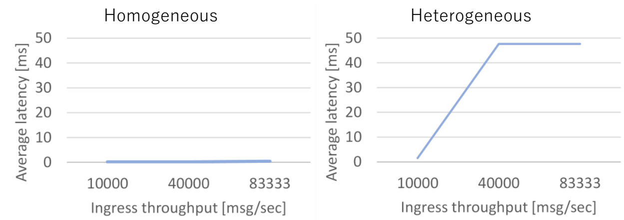

Fig. 5. Latency for small data

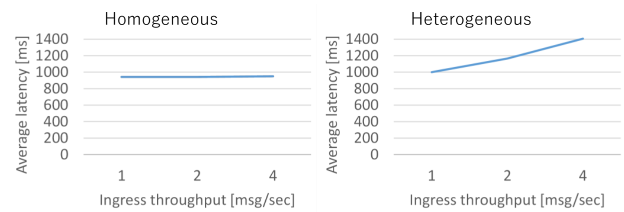

Fig. 6. Latency for large data

data case. From these results, Mosquitto and HiveMQ tend to suppress large data, whereas VerneMQ suppresses small data.

We also measured latency without the CPU resource limitation. We use the following ingress traffic patterns:

- $10,000$ msg/sec small data & 1 msg/sec large data
- $40,000$ msg/sec small data & 2 msg/sec large data
- $83,333$ msg/sec small data & 4 msg/sec large data

Figures 5 and 6 show the results of latency for Mosquitto. The average latency increases for both small and large data compared to the homogeneous data cases.

V. Conclusion

In this study, we conducted experiments to clarify the characteristics of MQTT brokers for heterogeneous traffic. Future work includes additional experiments with practical IoT traffic patterns and client configurations.

References

[1] MQTT, https://mqtt.org/ (accessed Mar. 29, 2023).
[2] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec, "The many faces of publish/subscribe," *ACM Computing Surveys*, vol. 35, no. 2, pp. 114–131, 2003.
[3] D. Ronzani, C. E. Palazzi, and P. Manzoni, "Bringing mqtt brokers to the edge: A preliminary evaluation," in *IEEE Annual Consumer Communications and Networking Conference*, 2022, pp. 695–698.
[4] M. Bender, E. Kirdan, M.-O. Pahl, and G. Carle, "Open-source mqtt evaluation," in *IEEE Annual Consumer Communications and Networking Conference*, 2021, pp. 1–4.
[5] C. B. Gemirter, C. Senturca, and S. Baydere, "A comparative evaluation of amqp, mqtt and http protocols using real-time public smart city data," in *International Conference on Computer Science and Engineering*, 2021, pp. 542–547.
[6] E. Gamess, T. N. Ford, and M. Trifas, "Performance evaluation of a widely used implementation of the mqtt protocol with large payloads in normal operation and under a dos attack," in *ACM Southeast Conference*, 2021, p. 154–162.
[7] B. Mishra, "Performance evaluation of mqtt broker servers," in *International Conference on Computational Science and Its Applications*, 2018, pp. 599–609.
[8] R. Banno, K. Ohsawa, Y. Kitagawa, T. Takada, and T. Yoshizawa, "Measuring Performance of MQTT v5.0 Brokers with MQTTLoader," in *Proc. IEEE Consumer Communications & Networking Conference*, 2021, pp. 1–2.