Distributed MQTT broker architecture using ring topology and its prototype

Shuichi Ohno\textsuperscript{1, a)}, Keitaro Terada\textsuperscript{2}, Tetsuya Yokotani\textsuperscript{3}, and Koichi Ishibashi\textsuperscript{3}

\textsuperscript{1} Graduate School of Electrical Engineering and Electronics, Kanazawa Institute of Technology
Nonoichi, Ishikawa 921-8812, Japan
\textsuperscript{2} Nagoya Works, Mitsubishi Electric Corporation
Higashi-Ku, Nagoya, Aichi, 461-8670, Japan
\textsuperscript{3} College of Engineering, Kanazawa Institute of Technology
Nonoichi, Ishikawa, Japan
a) b1629552@planet.kanazawa-it.ac.jp

Abstract: Some IoT services require simplified mechanisms in networks in order to communicate reliably and with low latency. This paper proposes a mechanism for connecting MQTT brokers for this purpose. In this proposal, the ring topology for connecting multiple brokers can efficiently process message routing and provide reliable and low latency communication. This paper describes the architecture and detailed operations of multiple ring-based brokers in MQTT. It also reports a prototype of the proposed system and its evaluation. Finally, the results demonstrate that this proposal is a promising and suitable approach.

Keywords: internet of things, IoT, MQTT, low latency, virtual ring, shared memory

Classification: Network

References

[1] H. Kagermann, J. Helbig, A. Hellinger, and W. Wahlster, “Recommendations for implementing the strategic initiative INDUSTRIE 4.0: Securing the future of German manufacturing industry final report of the Industrie 4.0 Working Group,” Forschungsunion, 2013.
[2] MQTT.org. http://mqtt.org/
[3] T. Yokotani, S. Ohno, H. Mukai, and K. Ishibashi, “IoT platform with distributed brokers on MQTT,” International Journal of Future Computer and Communication, vol. 10, no. 1, pp. 7–12, 2021.
[4] P. Schulz, M. Matthe, H. Klessig, M. Simsek, G. Fettweis, J. Ansari, S. Ali Ashraf, B. Almeroth, J. Voigt, I. Riedel, A. Puschmann, A. Mitschele-Thiel, M. Müller, T. Elste, and M. Windisch, “Latency critical IoT applications in 5G: Perspective on the design of radio interface and network architecture,” IEEE Commun. Mag., vol. 55, no. 2, pp. 70–72, 2017. DOI: 10.1109/mcom.2017.1600435cm
[5] ITU-T, “Network performance objectives for IP-based services,” ITU-T Y.1541, 2011.
[6] T. Yokotani, S. Yamamoto, S. Ohno, K. Sasabayashi, and K. Ishibashi, “Survey and comparison of Interworking point routing mechanisms for IoT services in wide area ICNs,” IEICE 2020 International Conference on Emerging Technologies for Communications (ICETC 2020), D2-4, 2020.

[7] R. Bannno, J. Sun, S. Takeuchi, and K. Shudo, “Interworking layer of distributed MQTT brokers,” IEICE Trans. Inf & Syst., vol. E102-D, no. 12, pp. 2281–2294, 2019. DOI: 10.1587/transinf.2019pak0001

[8] S. Abdelwahab, S. Zhang, A. Greenacre, K. Ovesen, K. Bergmam, and B. Hamdaoui, “When clones flock near the fog,” IEEE Internet Things J., vol. 5, no. 3, pp. 1914–1923, 2018. DOI: 10.1109/jiot.2018.2817392

[9] ITU-T, “Ethernet ring protection switching,” ITU-T Recommendation, G.8032/Y.1344, 2020.

[10] D. Buntinas, G. Mercier, and W. Gropp, “Implementation and evaluation of shared-memory communication and synchronization operations in MPICH2 using the Nemesis communication sub-system,” Parallel Computing, vol. 33, no. 9, pp. 634–644, 2007. DOI: 10.1016/j.parco.2007.06.003

[11] https://mosquitto.org/

1 Introduction

R&D activities and real deployments on the IoT have been pursued for several reasons, e.g., INDUSTRIE 4.0 [1]. Some IoT services require low latency communication, e.g., factory automation using sensor networks. For this purpose, simplified network protocols are required. One attractive option is the MQTT protocol [2]. However, the cooperation among brokers, which are the interworking points in an MQTT network, has not been specified. Authors have proposed the basic architecture on the multiple broker system with a ring topology [3]. This paper proposes an implemental approach and operations based on [3]. It also demonstrates its suitability using a prototype.

2 Requirements for low latency services

Reliability and low latency requirements of services in the IoT are summarized in [4]. For example, factory automation requires a packet loss ratio of $10^{-9}$ and a transfer delay of 10–250 ms, which are stricter values than those required by the IP QoS specified in ITU-T Y.1541 [5]. These services are the focus of this study.

3 Cooperation among distributed MQTT brokers

Approaches to operational protocols for distributed MQTT brokers are classified in [6]. In previous research, Bannno [7] assumed a tree topology and flooding to transfer data among brokers. In this approach, if branching points fail, downstream communication past these points is not possible. Abdelwahab [8] proposed a centralized architecture in the cloud and a mesh topology with flooding. This approach will increase traffic volume. Therefore, it can be concluded that new approaches should be proposed.
4 Distributed broker architecture and operations

Authors have proposed a new architecture for cooperation among MQTT brokers on the virtual ring [3]. This research has provided the basic architecture for interconnection among MQTT brokers on the virtual ring. It has also indicated performance evaluation using logical multiple queueing analysis. However, it has not described the detailed operations and implementation of this architecture. Therefore, this paper provides the detailed operations as shown in Figure 1.

![Fig. 1. Operations based on distributed broker architecture](image)

Distributed MQTT brokers are connected on a virtual ring configured as a VLAN. If a failure occurs on this ring, the virtual ring can be recovered using the layer 2 protection standardized in ITU-T, e.g., [9]. As shown in Figure 1, each broker contains an Access Control block and a Shared Memory block. The Access Control block recognizes and transfers MQTT topics to the next broker on the ring. The Shared Memory block consists of Read and Write areas. When topics are created by a publisher, such as an end point that is connected to this broker, these topics are routed to a Write area. When topics are transferred from other brokers, they are routed to the Read areas, which exist in every broker. This mechanism is expanded based on [10]. Figure 1 shows the transfer sequence among the Shared Memory blocks.

This operational sequence is as follows.

1. When broker #B receives some topics from a publisher that is connected to it, these topics are stored in the Write area of its shared memory.
2. These topics are then transferred to next broker, i.e., broker #C, who stores these topics in the Read area of its shared memory.
3. The topics can then be shared to the shared memory of all of brokers. Then, when a subscriber requests some topics from a broker, that broker provides the requested topics to the subscriber.

In these operations, infinite looping should be avoided. Therefore, on this ring, a VLAN initiated by a broker is blocked at the opposite ingress point. In this figure,
5 Performance evaluation

In the previous section, the proposed mechanism was described. To evaluate the latency between a publisher and a subscriber, emulation by virtualization on a PC with the following specifications was used.

- CPU: Intel® Core™ i3-8100
- RAM: 4.00 GB
- Clock Rate: 3.60 GHz

5.1 System configuration

The system consists of 100 brokers on a ring. A broker accommodates one publisher and one subscriber. Each publisher transfers 10 topics, i.e., a total of 1,000 topics, to the broker. Then, average latency between the publisher and all of subscribers is collected.

5.2 Evaluation results

Figure 2 shows the average latency results with 99% confidence intervals. The upper confidence interval is 44.51 ms and the lower one is 44.39 ms. As a result, the variation is very small. Hence, the proposed architecture is suitable for real time IoT services with severe time constraints.

![Fig. 2. The result of the experiment](image)

6 Prototype system

A part of the proposed system was prototyped to verify the feasibility of the implementation. Figure 3 shows the prototype system, consisting of four brokers on a ring. A publisher is registered to one broker and two subscribers are registered to other brokers. These brokers, publishers, and subscribers were implemented on Raspberry Pis installed with Mosquitto brokers [11].
In this configuration, two subscribers identified by 192.168.100.3 and 192.168.100.4 were registered on each broker, as shown in Fig. 3(a). Then, a publisher identified by 192.168.100.2 transfers a message with topic 16170670654795659, as shown in Fig. 3(b). At this time, this topic is shared with each subscriber on this ring. Then, each subscriber can receive this topic, as shown in Fig. 3(c).

Fig. 3. Prototype system using Raspberry Pi

Figures 3(d) and (e) show results of this experiment. These figures display data included in topic 1617067065 and 4795659.

7 Conclusion

This paper proposed detailed operations for cooperation among distributed MQTT brokers. These operations provide simplified operation, high reliability, and little variation in the transfer delay. The possibility of real deployment using this architecture was demonstrated by the prototype of the system. It can be concluded that the proposed architecture can be applied to time-constrained IoT services, especially, various IoT services in smart cities.

Acknowledgments

The research leading to these results has been supported by the EU-JAPAN initiative by the EC Horizon 2020 Work Programme (2018-2020) Grant Agreement No.814918 and Ministry of Internal Affairs and Communications “Federating IoT and cloud infrastructures to provide scalable and interoperable Smart Cities applications, by introducing novel IoT virtualization technologies (Fed4IoT) (JPJ000595)”. 