Performance of Sensors Monitoring System using Raspberry Pi through MQTT Protocol

Dania Eridani
dept. of Computer Engineering
Diponegoro University
Semarang, Indonesia
dania@ce.undip.ac.id

Eko Didik Widianto
dept. of Computer Engineering
Diponegoro University
Semarang, Indonesia
didik@live.undip.ac.id

***Abstract*—This paper shows the performance of sensors monitoring system built in Raspberry Pi as implementation of Internet of Things through MQTT protocol. The main focus is to observe the data read by the sensors that showed in the monitoring system over MQTT protocol. The controller used in this research is Raspberry Pi, as it also can directly be used as the transmission board. Temperature sensor and IR Range Sensor used in this research. The result show that the use of QoS parameter in MQTT configuration is the key of the data transfer success. By using the QoS parameter in the MQTT protocol give any packet loss. All of the data from the hardware can be transmitted to the monitoring system. The average delay between the data read by the sensor and the data seen in the HiveMQ websocket client is 0.23s maximum.**

***Keywords*—monitoring system, Raspberry Pi, MQTT protocol, sensors, Internet of Things**

I. INTRODUCTION

Internet of Things is one technology that able to help everyday life. It make the communication between objects (things) possible [1]. By using Internet of Things the monitoring system is way become easier because it can collects data from sensors, connect living and unliving things, and change the communication over IP network [2]. There are a lot of research about monitoring system using Internet of Things i.e. [3], [4], [5], etc. The use of Internet of Things makes the transmission process of the data change to via internet so that it become easy. Based on [6] Raspberry Pi has with various kind of functions that support building an Internet of Things application.

A. Raspberry Pi

Raspberry Pi is a Single Board Computer (SBC) product by Raspberry Pi Foundation [7]. Raspberry Pi support various interfaces and it has a wi-fi module in its board so that user does not have to use additional transmission board.

Raspberry Pi has SD card slot, DSI display connector, micro USB power, CSI connector camera, Broadcom, Ethernet Port, USB 2, GPIO headers, RCA video out, audio out, etc [8]. Raspberry Pi run its own operating system through an SD card. The used of Raspberry Pi in this research is to check the data performance using this Single Board Computer rather than using microcontroller.

As stated in [6] Raspberry Pi can support various kind of Internet of Things application. It can be used as a new tool in embedded project and educational tool.

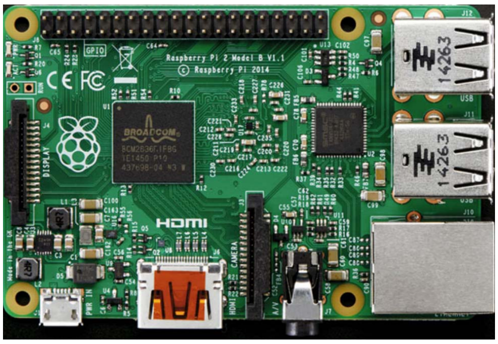

Fig. 1. Raspberry Pi

B. MQTT Protocol

MQTT (Message Queuing Telemetry Transport) is a protocol communication used to transmit data using internet. Based on [9] MQTT performs better than the HTTP. MQTT protocol usually used in embedded system because it can make the sensors communicate until the backend system, lightweight across the wire, and easy to be implemented in embedded system [10].

MQTT protocol work by sending and receiving topic using publish and subscribe mechanism. The publish mechanism usually used to send the data to the broker and the subscribe mechanism used to get the data from the broker [4]. Figure 2 shows the example of the publish and subscribe mechanism in MQTT protocol.

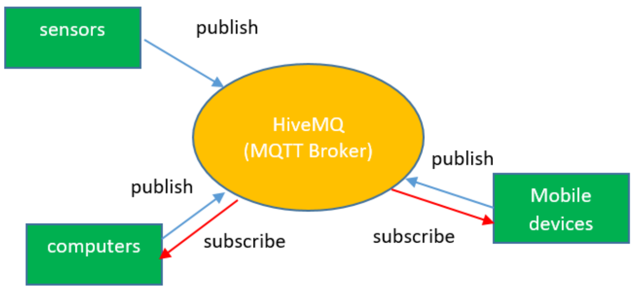

Fig. 2. Publish and subscribe mechanism using HiveMQ [11]

MQTT has three levels of Quality of Service (QoS) to make sure that all the data bring alongside the topic sent between the sender and the receiver [12]. The first QoS is use no confirmation, the second with confirmation required, and the last using four-step handshake.

One way to use MQTT protocol is by web socket client. One broker that support MQTT protocol is HiveMQ [13]. By using the web socket provides bi-directional communication between a browser and a web server. The web socket usually used to perform testing and debugging.

ISBN 978-1-5386-7421-5/18/USB ©2018 IEEE

This research focusing on analyzation of the transmission data through MQTT protocol by adapting the QoS parameter.

II. RESEARCH METHOD

The method of this research consist of five steps, i.e. study literature, designing the system, implementation, testing and analysis. The study literature used to find the functional and non-functional need used in this research. Besides, it also used as an early study about the transmission board and MQTT protocol.

By the study literature, this research uses Raspberry Pi 3 as the main controller, The HiveMQ used as the MQTT protocol, the sensors used as the data collection are IR range sensor and temperature sensor and read in voltage. The IDE used to develop the python code is Thonny IDE. The data collected are monitored using computer by accessing the HiveMQ web socket client.

Designing the system consist of designing the hardware and the software used in this research. The testing mechanism done by analyzed the data collected from the IR range sensor and the temperature sensor. The data are set to be read in three seconds delay for the temperature sensor and five seconds delay for the IR range sensor. The data collected in 30 seconds and 60 seconds for three times. Delay and packet lost are the parameter analyzed in this research.

III. DESIGN AND IMPLEMENTATION

The design can be seen in Figure 3. The Raspberry Pi used as the main controller in this system. This module connected to several sensors, in this case is two sensors, which are IR sensor and temperature sensor. Both of the sensor is read in voltage. The hardware then set to transmit the data to the cloud server through MQTT protocol. The user use browser in computer or laptop to monitor the transmitted data. The broker used is HiveMQ broker accessed in HiveMQ web socket client through internet.

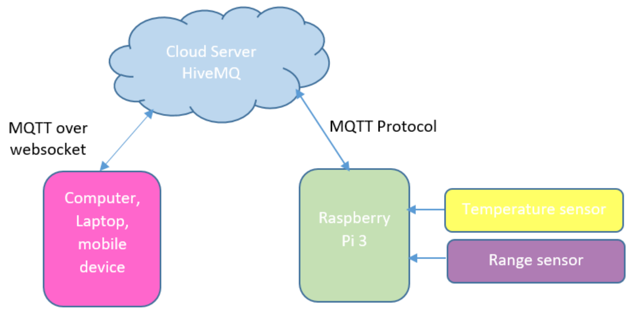

Fig. 3. System design

The QoS parameter set to the program is 2. Figure 4 shows the implementation of the hardware design. Figure 5 shows the monitoring through HiveMQ websocket client. The project name is DaniaPi and it is used to connect the board and the monitoring system.

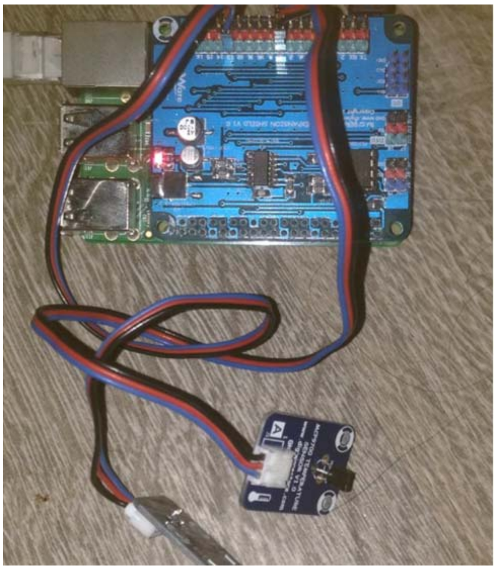

Fig. 4. Hardware impementation

Fig. 5. Data monitoring through HiveMQ websocket client

The monitoring system also uses the publish and subscribe mechanism in MQTT protocol. The monitoring system requesting subscribtion to the broker to get the sensor data.

The implementation procedure first start with the sensors as input devices read the environment data. The sensors connected to the Raspberry Pi so that it can tranfer the data to the HiveMQ server through MQTT protocol with QoS parameter set to 2. To monitor the data read by the sensor, the monitoring devices susch as computer, lapotop or mobile phone access the browser to HiveMQ websocket client to check the data read by the sensor before.

IV. RESULT AND ANALYSIS

The testing mechanism is analyzing the data transmitted from the hardware to the monitoring system through MQTT protocol. The temperature data is set to have three seconds delay and the IR range data is set to have five seconds delay. The first test is to check the delay and packet loss from the collected temperature sensor data in thirty second. The result can be seen in Figure 6.

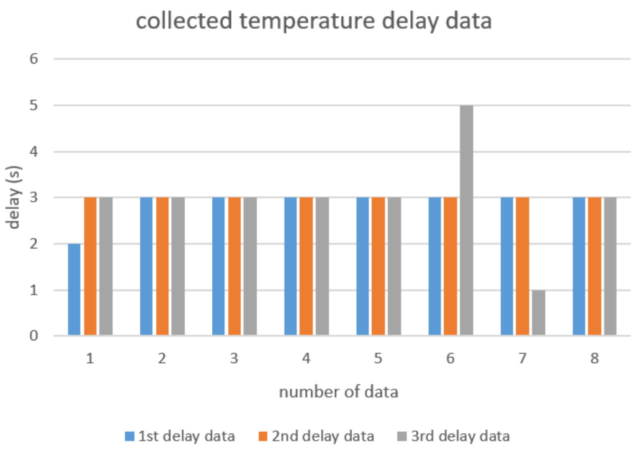

Fig. 6. Collected temperature delay data

The data showed that the first average delay is 2.875s and the second and third average delay is 3 s, meanwhile the delay given to the system is 3s delay. Based on the result there are -0.04s average delay between the hardware and the monitoring system. The data sent and the data received doesn't lost at all, it is 100% received. The second test is to check the delay and packet loss from the collected IR range sensor data in thirty second. The result is seen in Figure 7.

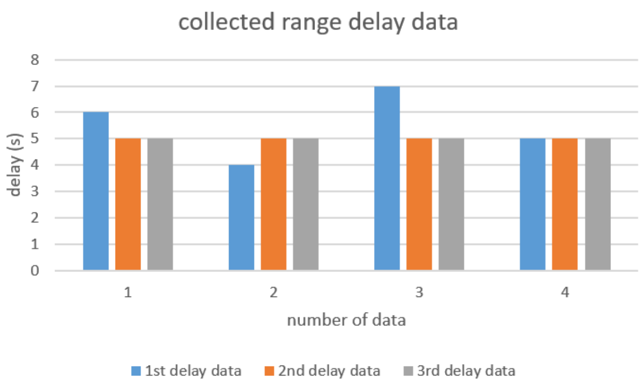

Fig. 7. Collected IR range delay data

The data showed that the first average delay is 5.5 and the second and third average delay is 5s, meanwhile the delay given to the system is 5s delay. Based on the result there are 0.17s average delay between the hardware and the monitoring system. The data sent and the data received doesn't lost at all, it is 100% received. The third test is to check the delay and packet loss from the collected temperature sensor data in sixty second. The result is seen in Figure 8.

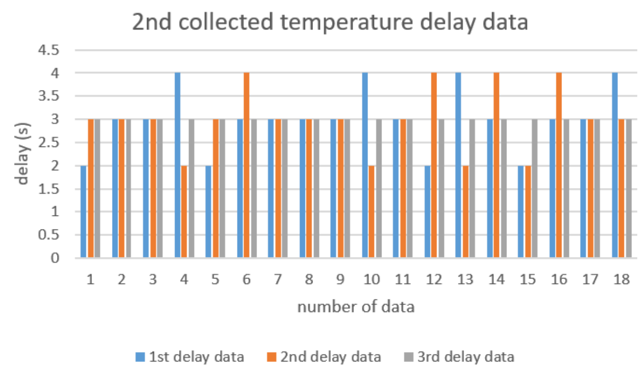

Fig. 8. The second collected temperature delay data

The data showed that the all average delay is 3s, meanwhile the delay given to the system is 3s delay. Based on the result there are no delay between the hardware and the monitoring system. The data sent and the data received doesn't lost at all, it is 100% received. The forth test is to check the delay and packet loss from the collected IR range sensor data in sixty second. The result is seen in Figure 9.

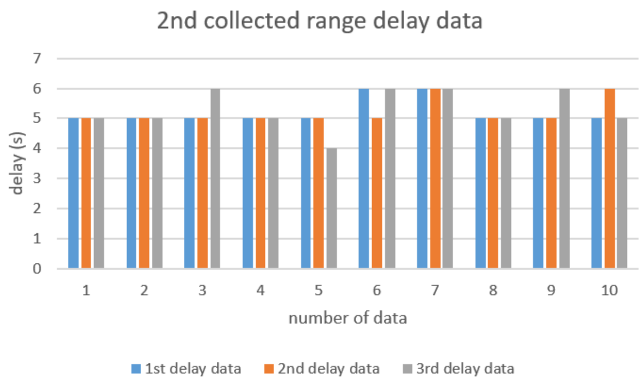

Fig. 9. The second collected IR range delay data

The data showed that the first and second average delay is 5.2s and the third average delay is 5.3s, meanwhile the delay given to the system is 5s delay. Based on the result there are 0.23s average delay between the hardware and the monitoring system. The data sent and the data received doesn't lost at all, it is 100% received. The result for all of the testing can be seen in Table 1.

TABLE I. AVERAGE DATA RESULT

Number	Temp in 30s	IR in 30s	Temp in 60s	IR in 60s
1	-0.12s	0.5s	0s	0.2s
2	0s	0s	0s	0.2s
3	0s	0s	0s	0.3s
Average	-0.04s	0.17s	0s	0.23s

From the result can be concluded that the use of QoS give 0% packet loss because all of the data from the hardware can be transmitted to the monitoring system. It is also can be concluded that the maximum average delay from the test is 0.23s came from the average IR data sensor read in 60 seconds.

V. Conclusion

The use of MQTT protocol make the sensor can fully transmit all of the data to the server and to the monitoring system. By using the QoS parameter in the MQTT protocol give any packet loss. All of the data from the hardware can be transmitted to the monitoring system. QoS parameter in MQTT configuration is the key of the data transfer success. The average delay between the data read by the sensor and the data seen in the HiveMQ websocket client is 0.23s maximum.

Acknowledgements

This research is part of the strategic research supported by Faculty of Engineering Diponegoro University Semarang. This research is aim to increase the Internet of Things knowledge. We would like to say thank you for Faculty of Engineering Diponegoro University for the support.

References

[1] Z. K. A. Mohammed and E. S. A. Ahmed, “Internet of Things Applications , Challenges and Related Future Technologies,” *World Sci. News*, vol. 67, no. 2, pp. 126–148, 2017.

[2] S. Pal, S. Ghosh, and S. Bhattacharya, “Study and implementation of environment monitoring system based on MQTT,” *Environ. Earth Sci. Res. J.*, vol. 4, no. 1, pp. 23–28, 2017.

[3] L. M. Varalakshmi, R. Kurinjimalar, and A. Vinjayalakshmi, “REMOTE MONITORING SYSTEM USING MQTT PROTOCOL,” vol. 119, no. 14, pp. 1633–1639, 2018.

[4] D. J. Vyas, N. N. Rudani, and M. T. Student, “MQTT & IOT Based Control and Monitoring of Smart Green House,” *IJEDR Int. J. Eng. Dev. Res.*, vol. 6, no. 1, pp. 2321–9939, 2018.

[5] M. Giridhar and S. Kayalvizhi, “IoT Based Industrial Smart Grid Monitorin Using MQTT Protocol,” vol. 118, no. 22, pp. 41–49, 2018.

[6] C. W. Zhao, J. Jegatheesan, and S. C. Loon, “Exploring IOT Application Using Raspberry Pi,” *Int. J. Comput. Networks Appl. Vol.*, vol. 2, no. 1, pp. 27–34, 2015.

[7] J. C. Shovic, *Raspberry Pi IoT Projects*. 2016.

[8] G. Loyse, “raspberry-pi Documentation,” *Rapberry Pi User Doc.*, p. 37, 2017.

[9] T. Yokotani and Y. Sasaki, “Comparison with HTTP and MQTT on required network resources for IoT,” *ICCEREC 2016 - Int. Conf. Control. Electron. Renew. Energy, Commun. 2016, Conf. Proc.*, pp. 1–6, 2017.

[10] B. S. Ullas, S. Anush, J. Roopa, and G. R. M, “Machine to Machine Communication for Smart Systems using MQTT,” *Int. J. Adv. Res. Electr. Electron. Instrum. Eng.*, vol. 3, no. 3, pp. 8242–8248, 2014.

[11] H. M. B. Enterprise, “MQTT 101 – How to Get Started with the lightweight IoT Protocol.” [Online]. Available: https://www.hivemq.com/blog/how-to-get-started-with-mqtt. [Accessed: 20-Sep-2018].

[12] S. A. Shinde, P. A. Nimkar, S. P. Singh, V. D. Salpe, and Y. R. Jadhav, “MQTT - Message queuing telemetry transport,” *Int. J. Res.*, vol. 3, no. 3, pp. 240–244, 2016.

[13] H. M. B. Enterprise, “MQTT over WebSockets.” [Online]. Available: https://www.hivemq.com/blog/mqtt-essentials-special-mqtt-over-websockets. [Accessed: 20-Sep-2018].