

# MQTT Performance as a Message Protocol in an IoT based Chili Crops Greenhouse Prototyping

1<sup>st</sup> Dania Eridani  
dept. of Computer Engineering  
Diponegoro University  
Semarang, Indonesia  
dania@ce.undip.ac.id

2<sup>nd</sup> Kurniawan Teguh Martono  
dept. of Computer Engineering  
Diponegoro University  
Semarang, Indonesia  
k.teguh.m@live.undip.ac.id

3<sup>rd</sup> Amaniyya Addini Hanifah  
dept. of Computer Engineering  
Diponegoro University  
Semarang, Indonesia  
amaniyya@gmail.com

**Abstract**—MQTT is an open source message protocol ideally used in machine-to-machine or Internet of Things. MQTT designed for limited devices, low bandwidth, and high latency system. MQTT provided with transmission assurance by using the QoS level in publish and subscribe mechanism. The aim of this research is to check the MQTT performance as an implementation of monitoring and controlling system. The system used in this research designed to be able to automate a greenhouse prototyping system and also to monitor and control greenhouses remotely. The case of this system is to automatically control and monitor chili crops in the greenhouse prototyping. This system used Wemos D1 R2 board as a control center, SHT11 sensor, YL-69 sensor, 16x2 LCD, RTC, relay, and Level-0 QoS MQTT communication protocol. The parameter used to check the MQTT performance are delay, throughput, and packet loss. The result showed that MQTT protocol is suitable for transmitting data in Internet of Things system. The average QoS parameter result changed based on the transmission speed in each network and the use of level 0 QoS on MQTT protocol.

**Keywords**— Greenhouse Prototyping, Internet of Things, Level 0 QoS, MQTT

## I. INTRODUCTION

IoT or Internet of Things used to connect the real world physical devices. Many systems in various field built based on IoT. Application that often built using IoT is a monitoring and controlling system. IoT also support the monitoring and controlling system in agriculture, like greenhouse [1] [2]. Monitoring the vital parameter of greenhouse now is possibly done by building a monitoring system based on IoT [3] [4]. By using greenhouse, people can create the best environment to have plantation, people can monitoring and control the environment used for the plantation. The common parameters must be monitored in greenhouse system are temperature, humidity, soil moisture, light, and PH level. The parameters used varies depend on the crops planted in the greenhouse.

### A. MQTT (Message Queuing Telemetry Transport)

MQTT or Message Queuing Telemetry Transport is a message protocol that ideal for use in many situations such as for Machine to Machine (M2M) and Internet of Things (IoT) system [5]. MQTT use publish and subscribe mechanism to support data-centric communication. This publish and subscribe mechanism have been widely used due to scalability [6].

MQTT supports several level of Quality of Services (QoS) to ensure that all of the data published received, by using level QoS 0 (at most once), QoS 1 (at least once), and QoS 2 (exactly once). The higher the QoS level the higher of the complexity process [7]. MQTT better used to transfer sensors data, and it is enable to build a system on this protocol. Building a system using MQTT protocol also has advantages

in mobile environment compare to the HTTP protocol. It is really possible to build a system in cellular network using MQTT protocol [8]. MQTT protocol also better than HTTP in QoS, faster response and throughput, security, and lower battery and bandwidth usage [5] [9] [10].

According to [11], there is a correlation between packet loss and delay to the QoS MQTT level used. The use of QoS level 2 produces the biggest delay compared to the other QoS levels, and the smallest delay results generated with the use of QoS level 0. Compare to the packet lost parameter, the smallest packet loss comes from QoS level 2 and the largest comes from QoS level 0.

There is also research that examines MQTT communication using NodeMCU board [12]. The research stated that the MQTT protocol is suitable for all cases defines above but speed of transfer information is less than the serial connection because it depends on the speed of Wi-Fi. It stated that this protocol never lost the data while it connected with Wi-Fi and it saves the data in a queue.

### B. Wemos D1 R2

Wemos D1 R2 is a microcontroller board based on ESP-8266 that integrated with Wi-Fi module. This board can be used to develop Internet of Things project. Wemos D1 R2 can be seen as in Fig. 1.

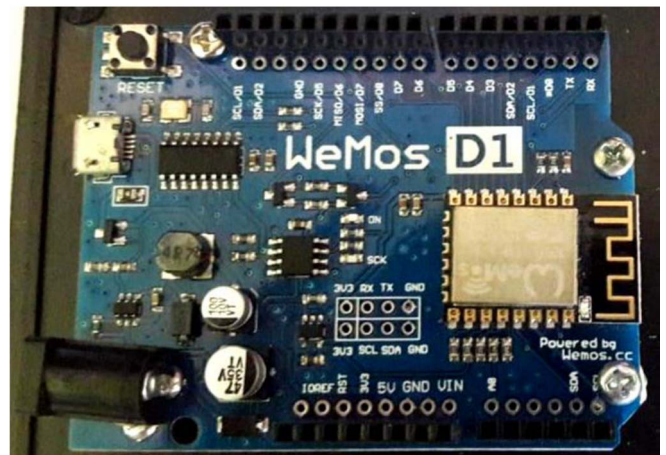


Fig. 1 Wemos D1 R2

Based on [13], Wemos using Tensilica L106 as the microcontroller integrated with ESP8266EX as a Wi-Fi module. It has 11 digital GPIO pin, 1 analog pin, 3.3V operational voltage and 68.6mm x 53.4mm board size. It also supported with flash memory and SRAM. As on [14], there are various others wireless development board that can be used to develop an IoT project as a comparison for Wemos D1 R2. It can be seen that Wemos D1 R2 also suitable to be used as an IoT project.

Based on the previous research, this paper contain the analysist of MQTT performance in the use of greenhouse monitoring and control system in various network bandwidth. The parameters used based on parameters suitable for developing chili crops. The MQTT implemented in Wemos D1 R2 board.

## II. RESEARCH METHOD

This research discuss the performance of MQTT protocol in Wemos D1 R2 board using delay, throughput, and packet loss as the parameters checked. The system used in this research designed to be able to automate a greenhouse prototyping and also to monitor and control greenhouses remotely. The case of this system is to automatically control and monitor chili crops in the greenhouse prototyping.

The method of this research consist of requirement analysist, hardware and software design, hardware and software implementation, and testing and analysist. The requirement analysist result are:

- The greenhouse prototype must be able to read all the parameter monitored.
- The greenhouse prototype must be able to put the data read by the sensors to display.
- The greenhouse prototype must be able to send all the parameters status using MQTT protocol to the MQTT broker.
- The greenhouse prototype must be able to read the message from the MQTT broker.
- The greenhouse prototype must be able automatically control the watering system, the plantation lamp, and the fan using the parameters read by the sensors.

The parameters used as a threshold in this system showed in Table I.

TABLE I. CHILI CROPS PARAMETERS [15]

<i>Parameters</i>	<i>Threshold</i>
Temperature	25-27 °C
Humidity	60-80% RH

Based on the result in [11], MQTT protocol used in this research is Mosquitto for broker and using Level 0 QoS for publish and subscribe mechanism. The QoS level 0 implemented in the system can be seen in Fig. 2. The design of the system used in this research shown in Fig. 3.

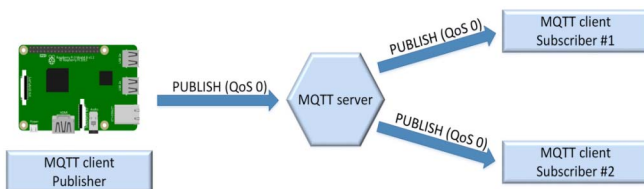


Fig. 2. QoS Level 0 mechanism [16]

The component used as the input to sense the temperature is SHT 11 and the humidity is YL-69. The component used as the output are water pump as the watering component, lamp and fan as the component to control the temperature, and LCD to display the status of the parameters controlled in

the system. This method limited in making the greenhouse prototyping monitor and control system in one node only.

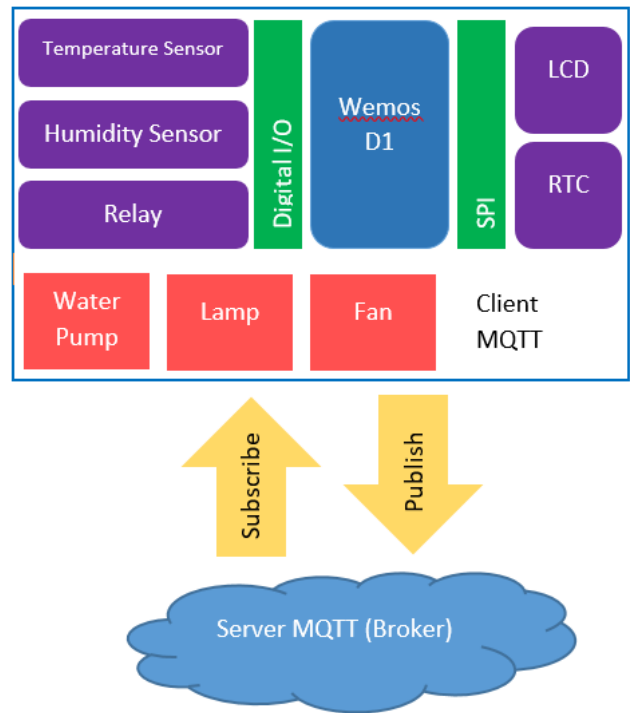


Fig. 3. System Design

Based on [11] to check the performance of MQTT protocol is done by comparing the the QoS parameter like delay and packet loss in the 3 QoS level of MQTT protocol. Here in this research, the testing mechanism method used is by checking the QoS parameter such as delay, throughput, and packet loss using only QoS level 0 on the system network to analyze MQTT performance in the various network bandwidth.

## III. IMPLEMENTATION

The components in the system united in a 68.6 mmX53.4 mm board using Wemos D1 R2 board as shown in Fig. 4. The board then connected to relay, sensors, adaptor, 16x2 LCD, and electric socket. The system then put in 300mmX200mm box as shown in Fig. 5

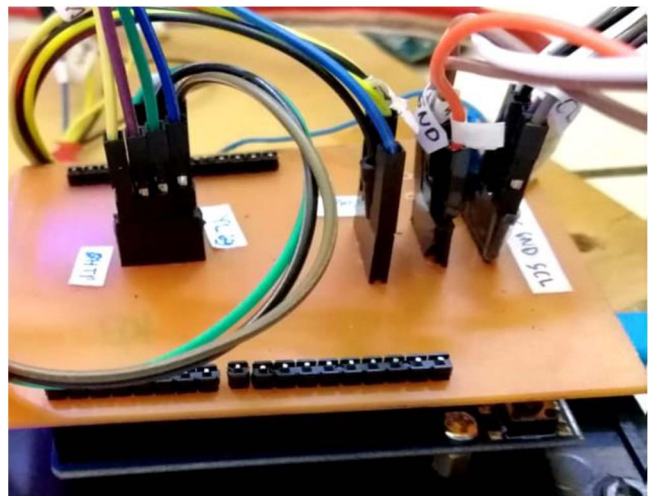


Fig. 4. System Board

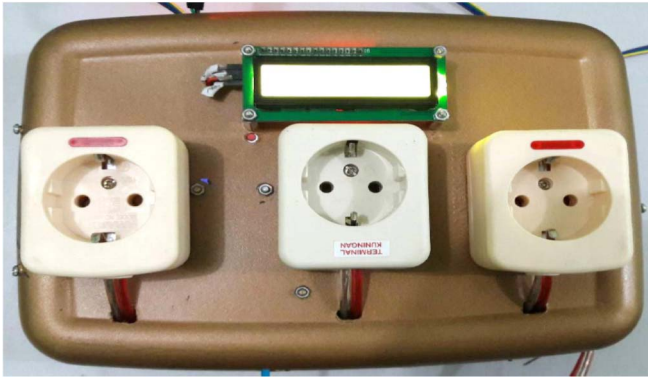


Fig. 5. System Box

After the implementation, the next step of this research is testing the MQTT QoS Level 0 by checking the delay, throughput, and packet loss as the MQTT performance parameters.

#### IV. TESTING AND ANALISYST

This test aims to determine the Quality of Service parameters of the MQTT used can run well. This system itself uses QoS MQTT level 0. This test done by sending and receiving several messages. The test done by using 3 different network specification. Network A has  $\pm 6.64$  Mbps download and  $\pm 7.49$  Mbps upload rates. Network B has  $\pm 1.44$  Mbps download and  $\pm 4.21$  Mbps upload rates. Network C has  $\pm 4.71$  Mbps download and  $\pm 5.53$  Mbps upload rates.

The test are carried out for 1 day per network and 10 sample data are taken for each message sent from the system to the broker (publish) and 10 sample data for each message received from the broker to the system (subscribe). Messages received and sent are 1 character or 8 bits.

##### A. Delay

Delay is the time difference between the sending point to the receiving point. Delay obtained from the difference between the time messages sent and the time it received. Delay obtained from the time difference between the timestamp on the serial monitor and the MQTT server time. Table II-III are the result of the delay testing while publishing messages.

TABLE II. RESULT OF DELAY PARAMETER IN PUBLISH TESTING USING NETWORK A

No	Timestamp	Server	Delay (s)
1	08:08:16.685	08:08:17.509	0,824
2	09:08:47.315	09:08:47.509	0,194
3	10:08:17.976	10:08:18.509	0,533
4	11:08:48.714	11:08:49.509	0,795
5	12:08:19.322	12:08:19.510	0,188
6	13:08:52.667	13:08:53.512	0,845
7	14:08:23.314	14:08:23.510	0,196
8	15:08:53.935	15:08:54.511	0,576
9	16:08:24.614	16:08:25.511	0,897
10	17:08:55.336	17:08:55.511	0,175

TABLE III. RESULT OF DELAY PARAMETER IN PUBLISH TESTING USING NETWORK B

No	Timestamp	Server	Delay (s)
1	08:25:53.389	08:25:53.802	0,413
2	09:25:53.503	09:25:54.267	0,762
3	10:25:53.506	10:25:53.843	0,337
4	11:25:53.522	11:25:53.855	0,333
5	12:25:53.556	12:25:53.878	0,322
6	13:25:53.554	13:25:53.843	0,287
7	14:25:53.584	14:25:53.834	0,25
8	15:25:53.620	15:25:53.869	0,249
9	16:25:53.634	16:25:53.872	0,238
10	17:25:53.623	17:25:53.836	0,213

TABLE IV. RESULT OF DELAY PARAMETER IN PUBLISH TESTING USING NETWORK C

No	Timestamp	Server	Delay (s)
1	08:13:51.719	08:13:51.972	0,253
2	09:13:51.738	09:13:52.051	0,313
3	10:13:51.775	10:13:52.044	0,269
4	11:13:51.811	11:13:52.045	0,234
5	12:13:51.831	12:13:52.117	0,281
6	13:13:51.857	13:13:52.137	0,28
7	14:13:51.892	14:13:52.171	0,279
8	15:13:51.927	15:13:52.289	0,362
9	16:13:51.960	16:13:52.298	0,338
10	17:13:51.991	17:13:52.262	0,271

The average delay generated during transmission (publish) on network A is 0.5223 seconds, on network B is 0.304 seconds, and on network C is 0.288 seconds. The delay comparison of publish testing between the 3 networks shown in Fig. 6.

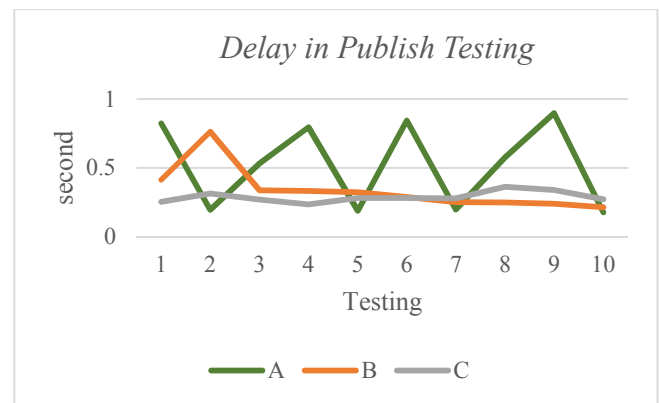


Fig. 6. Delay in Publish Testing

Table V-VII are the result of the delay testing while subscribing messages. The comparison of delay parameters on subscribe testing between the 3 networks shown in Fig. 7.

TABLE V. RESULT OF DELAY PARAMETER IN SUBSCRIBE TESTING USING NETWORK A

No	Timestamp	Server	Delay (s)
1	08:49:57.305	08:49:57.007	0,343
2	09:04:23.461	09:04:23.143	0,318
3	10:19:12.020	10:19:11.438	0,582
4	11:34:44.126	11:34:43.779	0,347
5	12:49:39.999	12:49:39.680	0,319
6	13:04:40.941	13:04:40.608	0,333
7	14:19:46.005	14:19:45.398	0,607
8	15:34:04.223	15:34:03.892	0,331
9	16:49:29.162	16:49:28.832	0,330
10	17:04:28.337	17:04:28.025	0,312

TABLE VI. RESULT OF DELAY PARAMETER IN SUBSCRIBE TESTING USING NETWORK B

No	Timestamp	Server	Delay (s)
1	06:26:14.029	06:26:13.678	0,351
2	07:34:35.036	07:34:34.699	0,337
3	08:38:36.457	08:38:36.112	0,345
4	09:44:51.613	09:44:51.335	0,278
5	10:04:41.233	10:04:40.875	0,385
6	11:06:58.266	11:06:58.008	0,258
7	12:08:18.023	12:08:17.745	0,278
8	13:13:08.859	13:13:08.587	0,272
9	14:13:13.624	14:13:13.332	0,292
10	15:14:05.707	15:14:05.340	0,367

TABLE VII. RESULT OF DELAY PARAMETER IN SUBSCRIBE TESTING USING NETWORK C

No	Timestamp	Server	Delay (s)
1	06:00:32.734	06:00:32.436	0,298
2	07:04:07.776	07:04:07.498	0,278
3	08:04:08.898	08:04:08.628	0,27
4	09:13:52.643	09:13:52.034	0,509
5	10:25:44.903	10:25:44.619	0,284
6	11:13:08.469	11:13:08.255	0,214
7	12:20:16.835	12:20:16.479	0,356
8	13:52:49.271	13:52:48.978	0,293
9	14:57:58.722	14:57:58.384	0,338
10	15:59:21.830	15:59:21.566	0,264

The average delay generated during transmission (subscribe) on network A is 0.3822 seconds, on network B is 0.3163 seconds, and on network C is 0.3104 seconds.

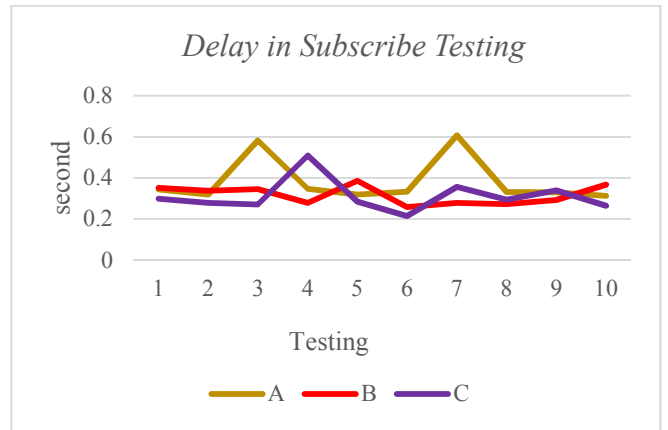


Fig. 7. Delay in Subscribe Testing

The test results found that the delay influenced by internet connections like download and upload speeds, it seen from the comparison of three different network speeds and the presence of outside interference can inhibit network connections in accordance with previous studies [12]. Delay time also influenced by the presence or absence of data that collides into or out of the broker or into the broker.

### B. Throughput

Throughput is the average speed of data received by a node in a certain time interval. This throughput test conducted at the time of sending data and receiving data. Test sending messages (publish) and receiving data (subscribe) by sending each of 1 characters or 8 bits of message. Table VIII-X are the results of throughput testing while publishing messages.

TABLE VIII. RESULT OF THROUGHPUT PARAMETER IN PUBLISH TESTING USING NETWORK A

No	Message	Delivery Time (s)	Troughput (bit/s)
1	8 bit	0,824	9,71
2	8 bit	0,194	41,24
3	8 bit	0,533	15,01
4	8 bit	0,795	10,06
5	8 bit	0,188	42,55
6	8 bit	0,845	9,47
7	8 bit	0,196	40,82
8	8 bit	0,576	13,89
9	8 bit	0,897	8,92
10	8 bit	0,175	45,71

TABLE IX. RESULT OF THROUGHPUT PARAMETER IN PUBLISH TESTING USING NETWORK B

No	Message	Delivery Time (s)	Troughput (bit/s)
1	8 bit	0,413	19,37
2	8 bit	0,762	10,50
3	8 bit	0,337	23,74
4	8 bit	0,333	24,02
5	8 bit	0,322	24,84
6	8 bit	0,287	27,87

7	8 bit	0,25	32,00
8	8 bit	0,249	32,13
9	8 bit	0,238	33,61
10	8 bit	0,213	37,56

TABLE X. RESULT OF THROUGHPUT PARAMETER IN PUBLISH TESTING USING NETWORK C

No	Message	Delivery Time (s)	Troughput (bit/s)
1	8 bit	0,253	31,62
2	8 bit	0,313	25,56
3	8 bit	0,269	29,74
4	8 bit	0,234	34,19
5	8 bit	0,281	28,47
6	8 bit	0,28	28,57
7	8 bit	0,279	28,67
8	8 bit	0,362	22,10
9	8 bit	0,338	23,67
10	8 bit	0,271	29,52

From the results of throughput testing by sending data of 1 Byte or 8 bits, the average value of throughput on network A is 23.74 bps. The average throughput on network B is 26.57 bps. While on the network C, the average throughput value is 28.21 bps. The throughput comparison in publish testing shown in Fig. 8.

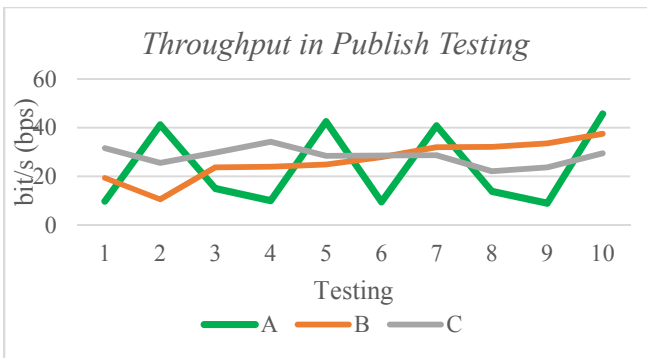


Fig. 8. Throughput in Publish Testing

Table XI-XIII are the results of throughput testing while subscribing messages.

TABLE XI. RESULT OF THROUGHPUT PARAMETER IN SUBSCRIBE TESTING USING NETWORK A

No	Message	Delivery Time (s)	Troughput (bit/s)
1	8 bit	0,343	23,32
2	8 bit	0,318	25,16
3	8 bit	0,582	13,75
4	8 bit	0,347	23,05
5	8 bit	0,319	25,08
6	8 bit	0,333	24,02
7	8 bit	0,607	13,18
8	8 bit	0,331	24,17

9	8 bit	0,33	24,24
10	8 bit	0,312	25,64

TABLE XII. RESULT OF THROUGHPUT PARAMETER IN SUBSCRIBE TESTING USING NETWORK B

No	Message	Delivery Time (s)	Troughput (bit/s)
1	8 bit	0,351	22,79
2	8 bit	0,337	23,74
3	8 bit	0,345	23,19
4	8 bit	0,278	28,78
5	8 bit	0,385	20,78
6	8 bit	0,258	31,01
7	8 bit	0,278	28,78
8	8 bit	0,272	29,41
9	8 bit	0,292	27,40
10	8 bit	0,367	21,80

TABLE XIII. RESULT OF THROUGHPUT PARAMETER IN SUBSCRIBE TESTING USING NETWORK C

No	Message	Delivery Time (s)	Troughput (bit/s)
1	8 bit	0,298	26,85
2	8 bit	0,278	28,78
3	8 bit	0,27	29,63
4	8 bit	0,509	15,72
5	8 bit	0,284	28,17
6	8 bit	0,214	371,38
7	8 bit	0,356	22,47
8	8 bit	0,293	27,30
9	8 bit	0,338	23,67
10	8 bit	0,264	30,30

From the results of throughput testing by receiving 1 Byte or 8 bits data from MQTT broker, the average value of throughput on network A is 22.6 bps. Network B throughput is 25.77 bps. While on the network C, is 27.03 bps. The throughput comparison in Subscribe testing shown in Fig.9. The difference in the value of throughput at the time of publish and subscribe is due to the bandwidth of the download and upload speeds by the network connection and the amount of messages sent or received.

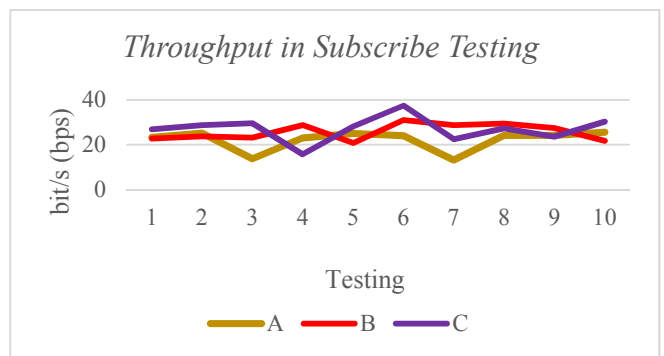


Fig. 9. Throughput in Subscribe Testing

### C. Packet Loss

Packet loss is the number of packets lost in a data packet shipment. This time the test is done by sending a message to the broker (publish) and receive messages from the broker (subscribe) for 60 seconds in a row to flood the server. Table XIV is the result of packet loss testing.

TABLE XIV. RESULT OF PACKET LOSS PARAMETER IN PUBLISH AND SUBSCRIBE TESTING

No	Type	Network	Duration (s)	Packet Send	Packet Received	Packet loss (%)
1.	Publish	A	60	208	206	3,33
2.	Subscribe	A	60	129	124	8,33
3.	Publish	B	60	356	352	6,67
4.	Subscribe	B	60	486	481	8,33
5.	Publish	C	60	449	448	1,67
6.	Subscribe	C	60	171	162	15

From the test results obtained, the packet loss when publishing messages on network A has a ratio of 3.33%, on network B 8.33%, and on network C 1.67%. Whereas when subscribing to a message, packet loss is 8.33% on network A, on network B has a 8.33% packet loss ratio, and on network C is 15%. It can be seen from the results of research that the value of packet loss at the time of subscribe mechanism is greater than publish one. The result is quite different comparing to [12], which showed that there is any packet lost in the publish and subscribe mechanism done in MQTT protocol using HiveMQ that connected to WiFi. This possibly because of the system or microcontroller is executing other programs, so that there are packets that do not reach the system. This can also be due to network congestion. and at the using of QoS level 0, there is no guarantee the message will actually be sent.

### V. CONCLUSION

The use of MQTT protocol in chili crops greenhouse prototyping is able to send sensor readings to the MQTT broker so that they can be monitored remotely. It also able to receive control messages from the MQTT broker to activate the pump, fan and lights properly. Things that affect delay, throughput, and packet loss are the differences of internet connection, data transmission speed of each network and the characteristics of QoS level 0 on MQTT.

### REFERENCES

[1] R. Koshy, M. D. Yaseen, K. Fayis, N. Shaji, N. J. Harish, and M. Ajay, "Greenhouse Monitoring and Control Based on IOT Using WSN," pp. 59–64, 2016.

[2] S. Shelvane, M. Shedage, and A. Phadtare, "Greenhouse monitoring using Raspberry Pi," pp. 5030–5035, 2019.

[3] V. N. S, P. Rajesh, and M. N. B, "IOT BASED GREEN HOUSE MONITORING SYSTEM," vol. 9958, pp. 45–47, 2015.

[4] T. A. Singh and J. Chandra, "IOT Based Green House Monitoring System," *J. Comput. Sci.*, vol. 14, no. 2015, 2018.

[5] 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.

[6] A. Rodriguez, L. M. Kristensen, and A. Rutle, "On Modelling and Validation of the MQTT IoT Protocol for M2M Communication."

[7] O. Deschambault, A. Gherbi, and C. Légaré, "Efficient Implementation of the MQTT Protocol for Embedded System," vol. 13, no. 1, pp. 26–39, 2017.

[8] X. Ma, A. Valera, H. Tan, and C. K. Tan, "Performance Evaluation of MQTT and CoAP via a Common Middleware."

[9] P. G. Krishna, K. S. Ravi, S. Kumar, V. S. S, and M. V. S. N. S. Kumar, "Implementation OF MQTT Protocol on Low Resourced Embedded Network," *Int. J. Pure Appl. Math.*, vol. 116, no. 6, pp. 161–166, 2017.

[10] J. M. K. Sri and N. Vg, "Implementing and Testing of IoT Technology in Agriculture," no. 6, pp. 848–852, 2019.

[11] S. Lee, H. Kim, D. K. Hong, and H. Ju, "Correlation analysis of MQTT loss and delay according to QoS level," *Int. Conf. Inf. Netw.*, pp. 714–717, 2013.

[12] M. Kashyap, V. Sharma, and N. Gupta, "Taking MQTT and NodeMcu to IOT: Communication in Internet of Things," *Procedia Comput. Sci.*, vol. 132, no. Iccids, pp. 1611–1618, 2018.

[13] Wemos, "D1 | WEMOS Electronics," *Products*. 2015.

[14] P. Plaza, E. Sancristobal, G. Carro, and M. Castro, "Wireless Development Boards to Connect the World," vol. 22, 2018.

[15] Suhendri, B. Irawan, and T. Rismawan, "Soil Humidity ontrol SystemSistem Pengontrolan Kelembaban Tanah Pada Media Tanam Cabai Rawit Menggunakan Mikrokontroler Atmega16 Dengan Metode Pd (Proportional & Derivative)," *J. Coding*, vol. 03, no. 3, pp. 11–22, 2015.

[16] G. C. Hillar, *MQTT Essentials - A Lightweight IoT Protocol*. 2017.