Internet of Things using Publish and Subscribe Method Cloud-based Application to NFT-based Hydroponic System

Muhammad Agus Triawan#1, Hilwadi Hindersah#2, Desta Yolanda#3, Febrian Hadiatna#4

#*School of Electrical Engineering and Informatics Bandung Institute of Technology*
Ganesha Street 10, Bandung 40132, Indonesia

1magustriawan@student.itb.ac.id
2hilwadihindersah@lskk.ee.itb.ac.id
3destayolanda@student.itb.ac.id
4febrianhadiatna@student.itb.ac.id

***Abstract*— The integration Internet of Things (IoT) generally identical with Cloud computing. IoT aims to connect the heterogeneous objects/devices with each other, in order to communicate or exchange data/information. Development of the Cloud IoT system, have challenges to achieve these objectives, such as middleware, interoperability and scalability of system. In this paper, we will discuss the solution of the challenges, by applying the publish and subscriber method, using MQTT protocol. This paper, proposed of architecture model design, which is implemented in the form of system topology. This topology apply to monitor and control the provision of nutrition hydroponics, accessible as a Cloud SaaS. From the test results, average node response time needed to connect and communicate with IoT Cloud system (online) is 7.21 seconds. The time required to declare offline state of a node is 21.55 seconds. The test results of Cloud IoT (broker), obtained that number of nodes for each level QoS, affecting resource usage of CPU and throughput. The highest resource usage of CPU is 19% of 2.0GHz and throughput TX/RX is 34.68KBps/49.4KBps. This result obtained from 1000 nodes scheme testing with 1 messages/node, by using simulation application, on QoS level 2. The average of minimum bandwidth used is ± 0.90Kbps by 1 publisher, to send 1 message with 115 Bytes of payload size, on each 3 levels of QoS MQTT, and approximately ± 0.04Kbps on subscriber to receive that message.**

Keywords*— *IoT, Cloud, broker, publish, subscribe, middleware, interoperability, MQTT, scalability.

I. INTRODUCTION

Internet of Things (IoT) or known as machine-to-machine refers to various devices (things) like sensors, actuators and other devices (heterogeneous objects/devices). This device which can produce, receive, deliver data or information, connect through a wired or wireless communication between the same or different devices, in order to communicate with each other, without any interaction or interference from human [1,4,8,21]. The use of phrase "Internet of Things" (IoT) was first introduced by Kevin Aston while working at the Auto-ID MIT, in his presentation at P&G 1999, with the simple concept of linking supply chain P&G RFID information to the Internet [2].

IoT devices typically have hardware characteristics indentical with low-power, limited processor capacity, small memory, even the communication capacity/resources and called as resources-constrained or constrained-devices [21]. In 2020, the number of devices to be connected to the Internet, will reach more than 50 billion [22,23]. It is one of the challenges in conductiong development IoT system.

There is some discussion about Cloud IoT such as: the integration Cloud with IoT, 2014 [19]; IoT Cloud-based services, 2016 [16]; the use of IoT for healt monitoring requirement, 2015 [20]; and the last review, which discusses about alternatives to replace the existing broker server on the Cloud, 2015 [18].

According to above reviewed, there are some research questions such as middleware: how heterogeneous objects can be interconnected; interoperability: how heterogeneous objects that connected, have the capability to communicate with each other; and the last about scalability aspect: how the server capabilities with fixed resources able to handling the increased number of clients.

This study is to realize Internet of Things communication, by applying Cloud-based publish and subscribe method. The purpose of combining the Cloud computing with IoT technology is to integrate things, in order to manage the messages of each object/device; to collect and organize data/information through the middleware layer; and through application layer, end-user can monitor and even control/setting of the system IoT in the Cloud, or straightforward can be imagined, Cloud computing as a control center of Internet of Things [3,19]. Cloud IoT systems development, applied to the needs of the monitoring and controlling/settings the parameters of providing nutrition hydroponics NFT-based (Nutrient Film Technique).

II. METHODOLOGY

Publish and subscribe method is a set of autonomous components that interact by events or message from source to destination, based on interest or desired topics. Publish and subscribe is a separator component between the node that serves as a publisher or subscriber. Component that generates

978-1-5090-5089-5/16/$31.00 ©2016 IEEE

the messages known as a publisher, while that consume messages known as subscriber. This means, the publisher and subscriber does not know the existence of each other. There are other components, know as broker who can be identified by the publisher and subscriber, which acts as an intermediary, so the publisher and the subscriber able to connect and communicate.

Broker responsible for receiving, or filtering entire messages to be forwarded or sent to all subscribers, according to the topic of interest by each node. Broker can be identified by each node via IP Address/Host name and port number used. So from this, node as publisher or subscriber, can be connected to communicate or exchange data/information each other. Each message send by the publisher, must be contain a topic, which will be used by the broker to forward the message to the subscriber through the topic. And usually each message has a payload, which contains information/actual data.

Message queuing telemetry transport protocol (MQTT) is an IoT/ M2M (machine-to-machine) connectivity protocols, using publish and subscribe method, to handle the communication process, shown on Fig. 1. MQTT created by Dr. Andy Standford-Clark from IBM, and Arlen Nipper from Arcom (now Eurotech) in 1991 [6], it is lightweight protocol, designed for constrained devices such as embedded devices(limited processor or memory resources), low bandwidth, high-latency or unreliable network [5,6,7,8,9].

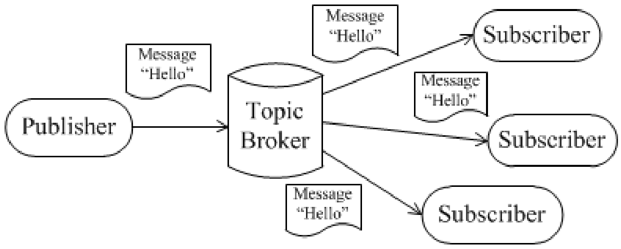


Fig. 1 *Publish* and *subscribe* method of MQTT

MQTT has 14 types of message or packet that is used for the communication process. MQTT protocol supports 3 levels of quality of service (QoS) shown on TABLE I, to handle the application that requiring different QoS level to send the persistent or continuous message, with various communication mechanisms such as one-to-one, one-to-many and vice versa [5,10,8].

TABLE I
3 LEVEL QOS MQTT

QoS Level	*Message Delivery*	*Delivery Semantics*	*Delivery Guarantees*
0	≤ 1	*At most once*	*Best effort no guarantees.*
1	≥ 1	*At lease once*	*Guaranteed delivery and duplicates possible.*
2	≡ 1	*Exactly once*	*Guaranteed delivery and no duplicates.*

There is a middleware layer in the Cloud IoT, intended for either heterogeneous objects nor applications, in order to be interconnected using the agreed protocol, as a middleware protocol, such as using MQTT protocol [19,25]. Through the protocol middleware, heterogeneous objects able to interconnected, but to be able to communicate with each other, heterogeneous objects requires a formatting language data exchange as agreed. This is known as interoperability, which can be handled by implementing a JavaScript Object Notation (JSON) as a formatting language exchange of data/information [17]. And it send or received using the publish/subscribe method, through Cloud computing, using a variety of devices such as smart-phone, tablet or personal computer [8,15,21,24].

Examples of formatting the data sensors and actuator using JSON in this study.

```
{"payload":{"sensor":{"ph":14.00,"ec":10.00,"wt":100
,"ml":100},"actuator":{"ac1":255,"ac2":255,"ac3":255
,"ac4":255,"acv":255}}
```

Cloud computing is a model that provide of various resources of computing, such as: network, server, storage, processing power, software, etc. combined into one (shared pool), or otherwise is called as a method of collective infrastructure, which is massive resources are connected together, based on the Internet as a service that can be accessed in accordance with requirements (on-demand) [11,12,13].

The use of Cloud computing has advantages as compared to traditional computing TABLE II, such as cost-saving aspects: paid according to the resources used, without paying any additional costs, like as infrastructure costs does not cheap and for maintenance cost; reliability aspect: how resources can be accessed via Internet anytime and anywhere, regardless of the availability of power and downtime/high- availability of resources. It can be know through service level agreement (SLA) from the Cloud computing provider, for example SLA reached 99.9%; scalability aspect: which can upscale or downscale of the resources required, and other advantages.

There are 3 models of basic services on the Cloud computing [12,13] are infrastructure as a service (IaaS), platform as a service (PaaS) and software as a service (SaaS).

TABLE II
TRADITIONAL COMPUTING VS CLOUD COMPUTING

Manage Entity	***Traditional Computing***	***Cloud Computing***		
		IaaS	**PaaS**	**SaaS**
Applications	✓	✓	×	×
Data	✓	✓	×	×
Runtimes	✓	✓	✓	×
Integration/Middleware	✓	✓	✓	×
O/S	✓	✓	✓	×
Virtualization	✓	×	✓	×
Server Hardware	✓	×	✓	×
Storage	✓	×	✓	×
Networking	✓	×	✓	×
(✓) *You manage.* (×) *Manage by vendor.*				

Service models of Cloud computing that used in this study. IaaS model: infrastructure that given, can be accessed remotely. So that we can monitor the use of physical resources (CPU, bandwidth, etc), perform the installation and configuration of various applications. such as operating systems, middleware, and others, which aims to connect and communicate various things. on PaaS and SaaS service model, are more likely to end-users in utilizing or using CloudIoT applications.

III. DESIGN SYSTEM

The architecture design proposed in this study, to address the issues of middleware, interoperability and scalability. The illustrated in Fig. 2 bellow.

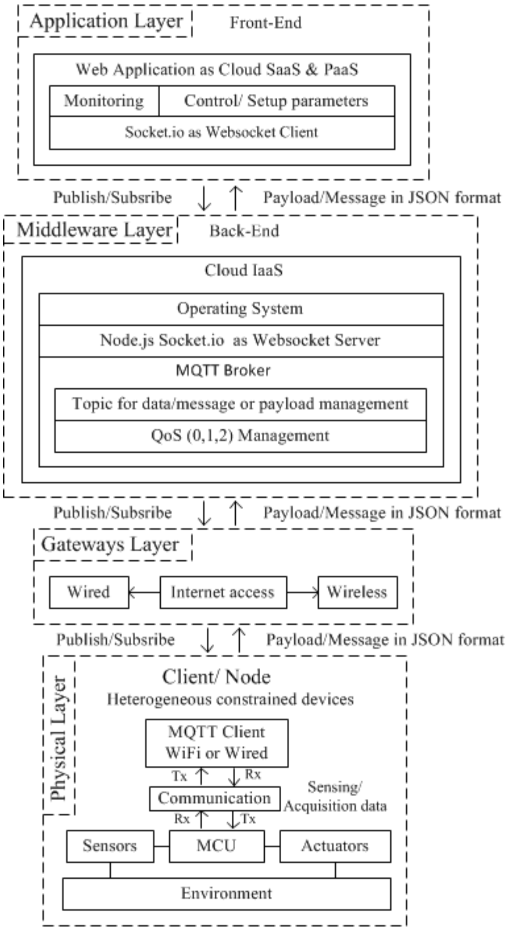


Fig. 2 Design of architecture system

1) Application layer: this layer at the top of the stack, responsible to handling end-user, in order to perform monitoring/controlling of Cloud IoT system, using various devices such as smartphones, tablet, or personal computer.

2) Middleware layer: this is one of the most critical layers that operates in bidirectional mode, to handling the complexity of the distribution data received or forwarded from heterogeneous objects [9,14,15,25]. It acts as an interface between the physical layer at the bottom and the application layer at the top. This layer provides a set of programming abstraction to facilitate the process of integration and communication of heterogeneous objects.

3) Gateway layer: the first stage to sending/receiving data, occurred in this layer. This layer has responsible to handling the routing mechanism such as how the data/message will be sent to/received by Cloud IoT.

4) Physical layer: allowing various clients/nodes that can be monitor or controlled remotely from Internet, through communication devices used in client/node. This layer has a sensing and data acquisition processes witch automatically obtained from its environment.

The following is topology Cloud IoT system in Fig. 3, designed base on the architecture of the IoT system in Fig. 2. Applied to NFT-based hydroponics system for monitoring and controlling the parameters nutrition.

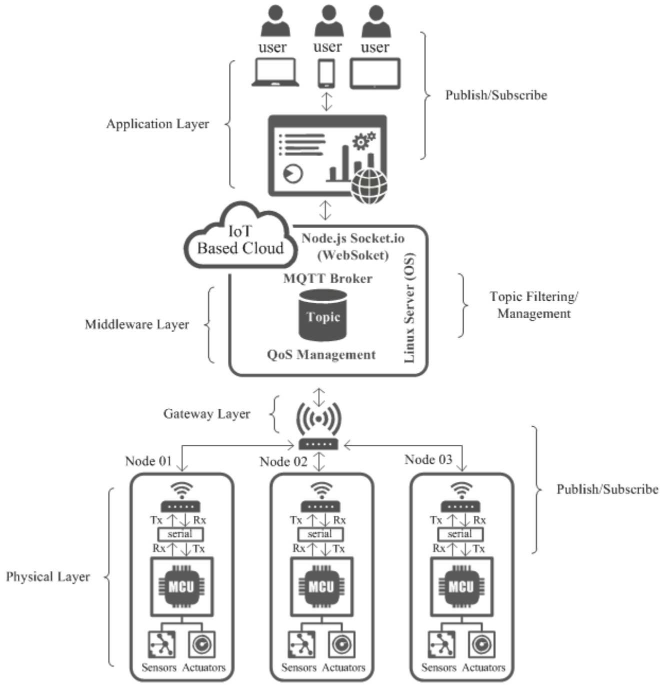
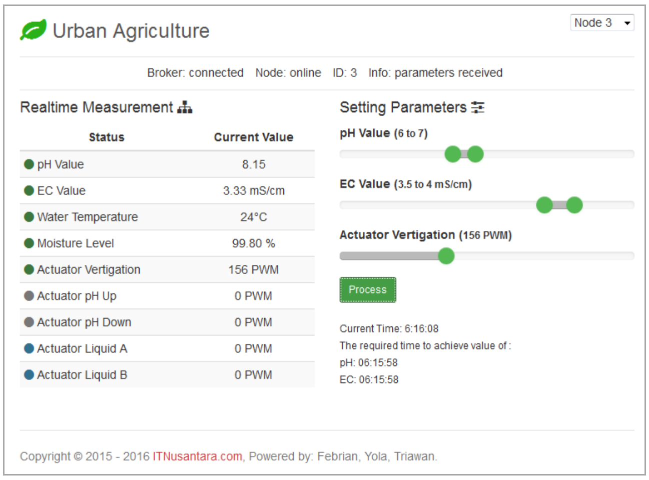


Fig. 3 Design of topology system

1) Application layer: There is Web application as a user interface in Fig. 4, which can be accessed from the Internet. This layer as an intermediary, that allows end-users to interact with the various nodes, such as to monitor and control/setting the parameters of hydroponic nutrients.



Urban Agriculture

Node 3

Broker: connected Node: online ID: 3 Info: parameters received

Realtime Measurement

Status	Current Value
● pH Value	8.15
● EC Value	3.33 mS/cm
● Water Temperature	24°C
● Moisture Level	99.80 %
● Actuator Vertigation	156 PWM
● Actuator pH Up	0 PWM
● Actuator pH Down	0 PWM
● Actuator Liquid A	0 PWM
● Actuator Liquid B	0 PWM

Setting Parameters

pH Value (6 to 7)

EC Value (3.5 to 4 mS/cm)

Actuator Vertigation (156 PWM)

Process

Current Time: 6:16:08

The required time to achieve value of :

pH: 06:15:58

EC: 06:15:58

Copyright © 2015 - 2016 ITNusantara.com, Powered by: Febrian, Yola, Triawan.

Fig. 4 User interface application for monitoring and controlling/settings

2) Middleware layer: MQTT protocol acts as middleware protocol to connect the device. From this layer can handle interoperability using JSON, so the devices able to communicate with each other. There are event-loop mechanism and non-blocking operation, using Node.js applications shown in Fig.5. The function is to execute and run commands/programs simultaneously, without must to wait the previous operation is completed. This mechanism is used to handle communication of 1-1, n-n, 1-n or vice versa. Example: one node publish a message with a payload (sensors and actuators) to many end-users (subscribe).

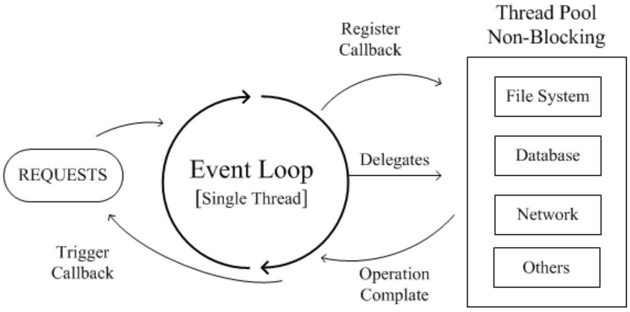


Fig. 5 *Event-Loop* and *Non-Blocking operation*

3) Gateway layer: there is a device that has capability of transmitting data from low-power link to the Internet. Such as using ADSL/ONT modem, or using mobile broadband.

4) Physical layer: data or information obtained from the Arduino Nano 3.0 with ATMEGA328 as node using sensors (pH, EC, Water temperatur and soil moisture or water level) and 4 actuators (pump for controlled tanks of acid solution, alkaline solution, nutrient A solution, nutrient B solution). The data/information send via low power communication devices, such as ESP8266 wireless based.

Some important functions/features of MQTT that used in this study.

1) KeepAlive feature: used to ensure the connection between broker and client keep establish and opened.

2) The function of the retained message is a normal message that has retained flag, so that the most recent message to be stored or retained by the broker.

3) Last will and testament (LWT): to say what would happen when a node enters offline or not connected to the broker. Due to network conditions such as unstable or disconnects, and is triggered by the expiration of the time interval feature "keepalive", and ideally the use LWT flag often combined with the retained messages.

IV. Implementation and Result

The testing of scalability aspect to evaluate the competence of server Cloud IoT as broker, with fixed-resources in handling additional number of client as node or end-user, based on CPU, throughput send-receive, and also examine on node as publisher and subscriber to evaluate the throughput, by using application simulation “mqtt-malaria” that can to demonstrate amount of node as *publishers* and *subscribers*.

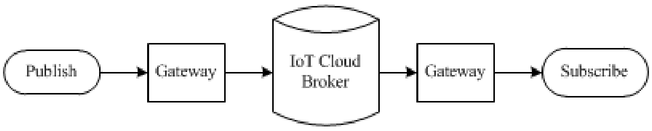


Fig. 6 Environment testing of Cloud IoT

TABLE III
Spesification of Hardware and Software in the Cloud IoT

No	*Cloud Hardware*	*Software*
1	*Processor* 2.0 GHz	*Operating System*: Ubuntu 14.04 32bit
2	*Memory* 512MB	Broker: Mosquitto 1.4.9/ MQTT 3.1/3.1.1
3	*Storage* SSD 20GB	Node.js 4.4.7 *with packet* Socket.io, MQTT Client, *and* Express.

The environment testing scheme of IoT system based on Cloud computing shown on Fig. 6, and specification hardware and software that used describes on TABLE III.

A. Testing of Bandwidth Publisher and Subscriber

The testing result of bandwidth, which is collected through gateway from each node as publisher/subscriber to the server Cloud IoT by using “mini speedtets” that installed on the server.

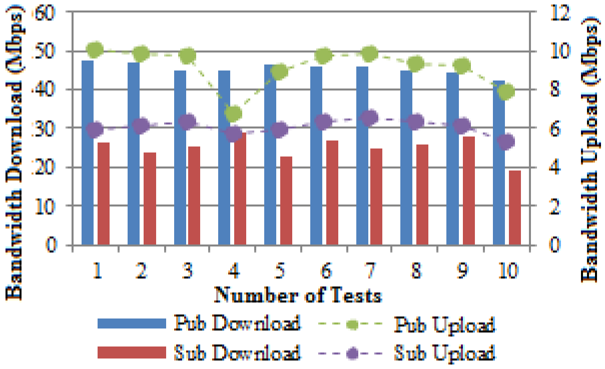


Fig. 7 Bandwidth (up/down) obtained by publisher and subscriber

From the testing result in Fig. 7, we found the average configuration of *bandwidth upload* and *download* from *publisher is* 9.14 Mbps and 45.53 Mbps and on *subscriber* is 6.08 Mbps and 9.14 Mbps.

B. *Testing of Online and Offline Node Response Time*

Configuration testing by giving setting number of interval “keep Alive” 15 second and data sending interval payload (sensor and actuator) 1 second, to determine where the node device signing online or offline status. It is can be seen on web.

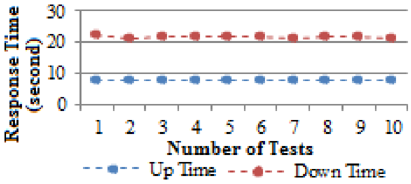


Fig. 8 Online and offline node response time

Based on the testing result on Fig. 8, the average of response time to show of node status on online condition 7.67 second, and to show of node status on offline condition 21.55 second.

C. *Testing of Resources use on Server Cloud IoT and Node*

1) Configuration testing with 100 *node* and 100 *messages/node* examined for 10 times by using different QoS level and payload *size* 115Byte.

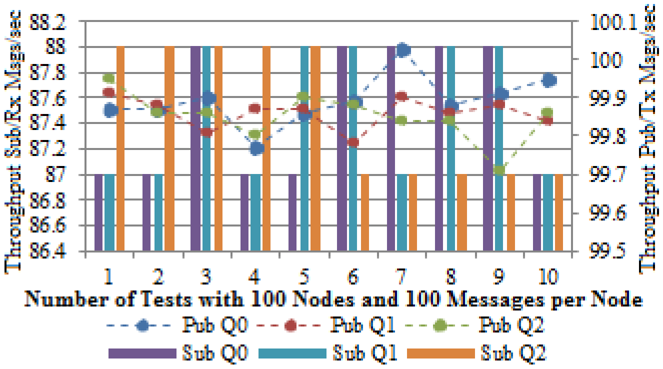


Fig. 9 Throughput used on the publisher and subscribe tested for 10 times with constant values

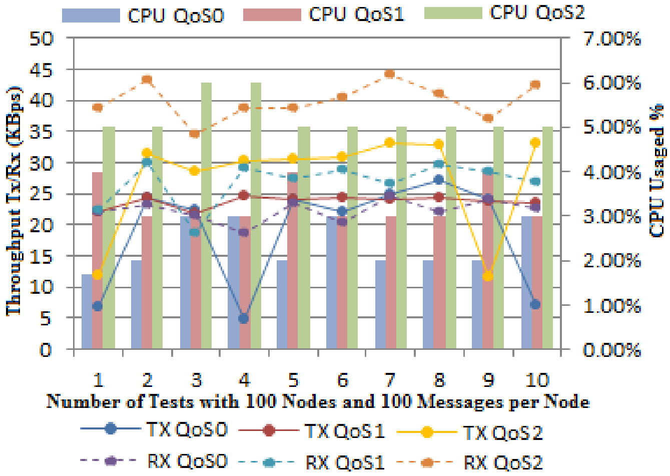


Fig. 10 CPU and throughput Tx/Rx used on the server Cloud IoT tested for 10 times with constant values

Based on the testing result in Fig. 9 and Fig. 10, the average used of throughput for sending, in publisher to each QoS level is QoS0 99.89 messages/second, QoS1 99.86 messages/second and QoS2 99.85 messages/second. The throughput receiving in subscriber used QoS0 87.5 messages/ second, QoS1 87.5 messages/second and QoS2 87.5 messages/second.

In server, the average of CPU that used is QoS0 2.37%, QoS1 3.30%, QoS2 5.20%; the average of throughput sending to QoS0 18.78KBps, QoS1 23.66KBps, and QoS2 27.43KBps; and the average of throughput receiving to QoS0 22.313KBps, QoS1 26.835KBps, and QoS2 39.965Kbps.

2) Configuration testing from 100 to 1000 nodes with the increasing of node is 100 nodes and one message per node, with using different level QoS and payload size is 115 Bytes. Testing result that can be seen in Fig. 11 dan Fig.12

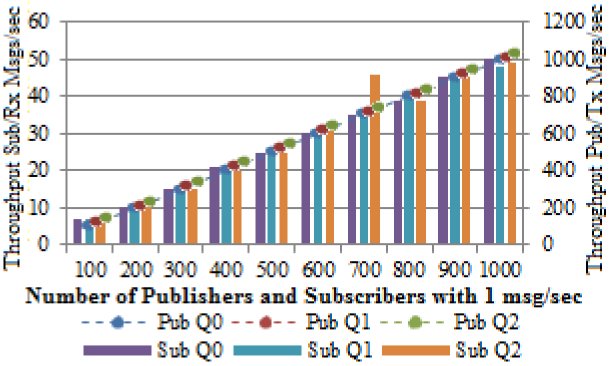


Fig. 11 Throughput used on the publisher and subscribe tested with increasing values

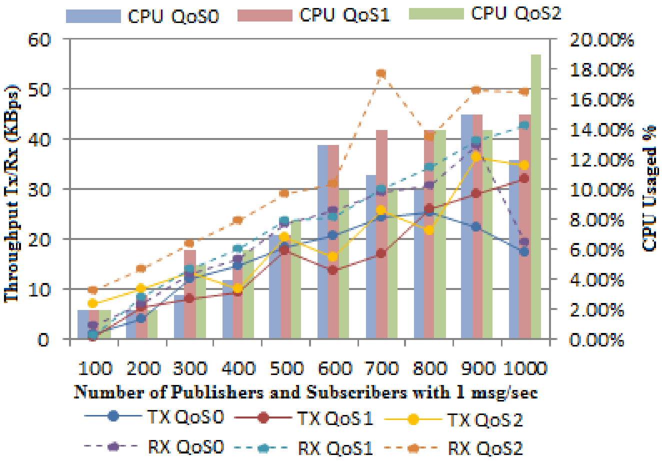


Fig. 12 CPU and throughput Tx/Rx used on the Server Cloud IoT tested with increasing values

Based on the testing result in Fig. 11 and Fig. 12, obtained the average value of difference by increasing number of publisher to each QoS level. The throughput sending using QoS0 100.17 messages/second, QoS1 99.21 messages/ second, and QoS2 100.01 messages/ second. In additional of subscriber, the throughput receiving using QoS0 4.78 messages/ second, QoS1 4.56 messages/ second, and QoS2 4.67 messages/ second.

The CPU server that used in the additional of node is QoS0 1.11%, QoS1 1.44%, and QoS2 1.89%. The average value of difference by increasing number of node on sending throughput in server to each QoS level is QoS0 1.79KBps, QoS1 3.50KBps and QoS2 3.07KBps. The average value of difference by increasing number of node on receive throughput in server to each QoS level is QoS0 1.84KBps, QoS1 4.66KBps and QoS2 4.41KBps.

V. CONCLUSION

Based on the Cloud IoT architecture system proposed on Fig. 2, and implemented to NFT-based hydroponic system for monitoring and controlling the parameters nutrition, on Fig. 3, obtained that MQTT as a middleware protocol can connect nodes and end-users to each other, using publish and subscribe method. On the middleware layer, interoperability aspect can be solved by using JSON. So nodes and end-users can communicate or exchange data/information. On the physical layer, the size of maximum data is 115 Bytes, the result is obtained from sensing/data acquisition and formatted into JSON. Through gateway layer end-user and node can connect to the Cloud IoT that created. The average time required by node in order to be connected and communicated (online) is 7.67 second, with the sending data interval of payload (sensors and actuators) 1 second. The average time which is needed to know node is entering offline status is 21.55 second, can be identified via Web. On the application layer, end-user able to control and monitor the parameters of nutrition hydroponic system NFT-based, via Web application with responsive design shown on Fig. 4.

The testing scalability of Cloud IoT server, obtained the increasing numbers of node with different QoS will affected CPU usaged, and throughput sending/receiving (QoS2 > CPU QoS1 > CPU QoS0). The highest resource usage of CPU is 19% of 2.0GHz and throughput TX/RX is 34.68KBps/49.4KBps. This result obtained from 1000 nodes scheme testing with 1 messages/node, by using simulation application, on QoS level 2. So from the highest test results of CPU resource and throughput TX/RX usaged, we can upscale or downscale of Cloud IoT server resources if necessary.

REFERENCES

[1] Honbo Zhou, The Internet of Things in the Cloud A Middleware Perpective, CRC Press.

[2] Kevin Ashton, “That ‘Internet of Things’ Thing,” RFID Journal, 22, June 2009.

[3] J. Rui and S. Danpeng, "Architecture Design of the Internet of Things Based on Cloud Computing," 2015 Seventh International Conference on Measuring Technology and Mechatronics Automation, Nanchang, 2015, pp. 206-209.

[4] Y. K. Chen, "Challenges and opportunities of internet of things," 17th Asia and South Pacific Design Automation Conference, Sydney, NSW, 2012, pp. 383-388.

[5] ______, 10 December 2015, MQTT Version 3.1.1 Plus Errata 01. Edited by Andrew Banks and Rahul Gupta. OASIS Standard Incorporating Approved Errata 01. http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/errata01/os/mqtt-v3.1.1-errata01-os-complete.html. Latest version: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html, 7 Juni 2016.

[6] ______, 2015, Frequently Asked Questions, http://mqtt.org/faq, 7 Juni 2016.

[7] ______, OASIS Message Queuing Telemetry Transport (MQTT) TC, https://www.oasis open.org/committees/tc_home.php?wg_abbrev=mqtt, 9 Juni 2016.

[8] Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," in IEEE Communications Surveys & Tutorials, vol. 17, no. 4, pp. 2347-2376, Fourthquarter 2015.

[9] M. H. Asghar and N. Mohammadzadeh, "Design and simulation of energy efficiency in node based on MQTT protocol in Internet of Things," Green Computing and Internet of Things (ICGCIoT), 2015 International Conference on, Noida, 2015, pp. 1413-1417.

[10] Ngo Manh Khoi, S. Saguna, K. Mitra and C. Åhlund, "IReHMo: An efficient IoT-based remote health monitoring system for smart regions," 2015 17th International Conference on E-health Networking, Application & Services (HealthCom), Boston, MA, 2015, pp. 563-568.

[11] B. P. Rao, P. Saluia, N. Sharma, A. Mittal and S. V. Sharma, "Cloud computing for Internet of Things & sensing based applications," Sensing Technology (ICST), 2012 Sixth International Conference on, Kolkata, 2012, pp. 374-380.

[12] Peter Mell, Timothy Grance, The NIST Definition of Cloud Computing, september 2011, National Institute of Standards and Technology, U.S. Department of Commerce.

[13] Y. Yuan, L. Ma and J. Zhang, "Patented Network Analysis on Cloud Computing Technology in Internet of Things," Identification, Information and Knowledge in the Internet of Things (IIKI), 2014 International Conference on, Beijing, 2014, pp. 248-251.

[14] T. Campbell, G. Coulson and M. E. Kounavis, "Managing complexity: middleware explained," in IT Professional, vol. 1, no. 5, pp. 22-28, Sep/Oct 1999.

[15] Mangal Sain, HoonJae Lee and Wan-Young Chung, "Middleware for ubiquitous Healthcare Information system," Advanced Communication Technology, 2009. ICACT 2009. 11th International Conference on, Phoenix Park, 2009, pp. 2325-2328.

[16] Taherkordi and F. Eliassen, "Scalable modeling of cloud-based IoT services for smart cities," 2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops), Sydney, NSW, 2016, pp. 1-6.

[17] Lin, Y. Chen, X. Chen and Y. Yu, "Comparison between JSON and XML in Applications Based on AJAX," Computer Science & Service System (CSSS), 2012 International Conference on, Nanjing, 2012, pp. 1174-1177.

[18] J. L. Espinosa-Aranda, N. Vallez, C. Sanchez-Bueno, D. Aguado-Araujo, G. Bueno and O. Deniz, "Pulga, a tiny open-source MQTT broker for flexible and secure IoT deployments," Communications and Network Security (CNS), 2015 IEEE Conference on, Florence, 2015, pp. 690-694.

[19] Botta, W. de Donato, V. Persico and A. Pescapé, "On the Integration of Cloud Computing and Internet of Things," Future Internet of Things and Cloud (FiCloud), 2014 International Conference on, Barcelona, 2014, pp. 23-30.

[20] M. Hassanalieragh et al., "Health Monitoring and Management Using Internet-of-Things (IoT) Sensing with Cloud-Based Processing: Opportunities and Challenges," Services Computing (SCC), 2015 IEEE International Conference on, New York, NY, 2015, pp. 285-292.

[21] M. A. Razzaque, M. Milojevic-Jevric, A. Palade and S. Clarke, "Middleware for Internet of Things: A Survey," in IEEE Internet of Things Journal, vol. 3, no. 1, pp. 70-95, Feb. 2016.

[22] Carrillo, V. Benitez, C. Mendoza and J. Pacheco, "IoT framework for smart buildings with cloud computing," Smart Cities Conference (ISC2), 2015 IEEE First International, Guadalajara, 2015, pp. 1-6.

[23] David Bradley, David Russell, Ian Ferguson, John Isaacs, Allan MacLeod, Roger White, The Internet of Things – The future or the end of mechatronics, Mechatronics, Volume 27, April 2015, Pages 57-74, ISSN 0957-4158, http://dx.doi.org/10.1016/j.mechatronics.2015.02.005.

[24] J. Yang, L. Zhang and X. A. Wang, "On Cloud Computing Middleware Architecture," 2015 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), Krakow, 2015, pp. 832-835.

[25] G. Fersi, "Middleware for Internet of Things: A Study," 2015 International Conference on Distributed Computing in Sensor Systems, Fortaleza, 2015, pp. 230-235.