Implementation MQTT-SN Protocol on Smart City Application based Wireless Sensor Network

M.Udin Harun Al Rasyid, Ferry Astika, Fadlul Fikri
Politeknik Elektronika Negeri Surabaya (PENS), Indonesia
udinharun@pens.ac.id, ferryas@pens.ac.id, fikrifadlul@gmail.com

***Abstract*— Many smart city applications have implemented the Wireless Sensor Network (WSN) as a solution to solve urban problems. WSN has characteristics with an energy source that must remain lit, the ability to overcome node failures, node heterogeneity, ability to withstand extreme environmental conditions, mobility of nodes and dynamic network topology, scalability for large-scale deployment. However, there are various kinds of machine to machine (M2M) protocol that are used on Smart City based on WSN to connect between sensor devices or the internet. To choose the appropriate protocol and small packet loss probability is not easy. This research aims to implement the Message Queuing Telemetry Transport-Sensor Network (MQTT-SN) Protocol to overcome the limitations of the WSN-based protocol on Smart City sensors. MQTT-SN is an M2M connectivity protocol that is designed for sensor networks that are simple, lightweight, and easy to implement, so they can support and monitor small devices with wider data transmission. Implementation was built on the waspmote hardware using the MQTT-SN protocol for sensor communication with the server. Data transmission uses the XBee module to optimize and still be done in real time. The implementation uses a star topology and analyzes the Quality of Service (QoS) performance and power consumption used on the sensor nodes. Based on the analysis results, there is no packet loss when using the MQTT-SN, HTTP, and CoAP protocols. Temperature sensor, light sensor, and dust sensor data can be monitored remotely using the MQTT-SN protocol.**

Keywords—Smart city; Protocol; MQTT; MQTT-SN; Sensor;

I. INTRODUCTION

The development of technology and the increasing complexity of problems in various cities has become a first step for the implementation of smart cities concept in Indonesia. The development of technology and the increasing complexity of problems in various cities has become a first step for the implementation of smart cities concept in Indonesia. Smart City is a concept of urban development and management with the use of Information and Communication Technology (ICT) to connect, monitor and control various resources within the city more effectively and efficiently to maximize services to its citizens and support sustainable development [1].

Wireless Sensor Network (WSN) is the computer network consisting of spatially distributed sensors to monitor environmental or physical conditions, such as pressure , temperature, health, sound, and to cooperatively pass their data through the wireless network to a base station [2]. WSN is usually placed in different places to monitor the condition of the surrounding environment. Smart City-based WSN is a concept of urban development by integrating autonomous sensors that work wirelessly and used to detect physical phenomena, perform data processing and transmit data from urban environments.

WSN has characteristics that include the energy resource of each sensor node must remain life, the ability to handle node failure, node heterogeneity, the ability to survive extreme environmental conditions, the mobility of nodes and dynamic network topology, scalability for large-scale deployment. Therefore, manage performance on WSN is essential for the sensor network nodes to work well. Each sensor node has its limitations in terms of resources, processing, memory capacity and communication bandwidth. With the operation and lifetime of the sensor network directly dependent on the availability of power in the power unit only supplied from the battery and the ability to overcome node failure. This limitation is one of the important factors in the selection of protocols on Smart City-based WSN.

Message Queuing Telemetry Transport-Sensor Network (MQTT-SN) protocol is a connectivity protocol machine-to-machine (M2M) designed for network sensors that are simple, lightweight, and easy to apply, which can support and monitor small devices with transmission data widely [3]. This protocol uses the data centric communication publish/subscribe, in which information transmitted to the recipient is not based on the IP Address but through the function of the content and the information content delivered.

In this research, we propose implementing the MQTT-SN protocol on the WSN-based Smart City application. The main objective of this research is to analyze the data received performance of smart city networks when implementing the MQTT-SN protocol on WSN-based Smart City Applications. By implementing the MQTT-SN protocol, the sensor can publish data and allows the subscriber to receive data, and the subscriber does not need to be connected simultaneously with the publisher.

In order to study the suitable protocol in smart city applications, many researchers have already shown up with numerous researches related to MQTT-SN protocols for WSN. Antonic et al. [4] proposed the publish/subscribe middleware implementation for building smart city service. This is using taxonomy to identify vital features of IoT publish/subscribe middleware which are compare two protocol, there are MQTT protocol implemented by the Mosquitto broker and CUPUS middleware. MQTT supports for various network communication protocols and simple security mechanism. Thus CUPUS middleware offers to discover data sources and control data acquisition. The CUPUS middleware and Mosquitto broker have similar performances for the at least once delivery semantics. Kodali et al. [5] proposed the easily integrated to make a low cost, low power consumption system using MQTT protocol. The system used for monitoring in the smart home appliance, street light system for the smart cities, fire alert systems. The system is connected ESP8266 WIFI to read and collect data from DHT11 sensor and LDR sensor. The visualization is provided for client using smartphone and laptop to get data from MQTT server.

Sahadevan et al. [6] proposed the communication made power efficient and low cost for computation on the IoT application. The system requires networking to be strong synchronization and then the system is connected intel galileo to MQTT protocol for providing to synchronize data with cloud. The system online-offline extension reduced network utilization and network availability requirements for application. Rahman et al. [7] proposed secure communication for other devices on IoT ecosystem and require security along with cryptosystems. The system use resources with limited computation and transmitting data using MQTT protocol. Thus the system is less power, efficient of transmission and more performance to communication for low-end device than using traditional cryptographic.

Wukkadada et al. [8] proposed the comparison of protocol to reduce power of the communication will be less on the IoT. The comparison of system was using two protocols MQTT and HTTP protocol. MQTT protocol uses less power to maintain connection and receive message and to send them. MQTT will be best technique to use on IoT communication. Zabasta et al. [9] proposed enhance the quality and performance of urban service for smart city system on IoT. The service broker implemented for control utilities systems. The systems implemented of event handler to support the handling of event and interacting via publish/subscribe. The event handler as a MQTT enables service broker and API for online service. and then it's using programming node-RED as a distributed data to wire sensor, node and gateways enable time and cost saving.

Amaran et al. [10] performed evaluate the Constrained Application Protocol (CoAP) and MQ Telemetry Transport for Sensor Nodes (MQTT-SN) in robot communication in terms of transmitting time. Wukkadada et al. [11] performed the comparison of HTTP and MQTT protocols for sending the temperature sensor data to server in terms of power used and packet delivery rate.

Different from the previous researches, the aims of this paper is to obtain the performance of the MQTT-SN protocol implemented in WSN-based smart city applications and compare the performance of the MQTT-SN protocol with Hyper Text Transfer Protocol (HTTP) and Constrained Application Protocol (CoAP). Performance testing is carried out with three test parameters, namely delay, throughput, and packet loss.

II. THE SYSTEM DESIGN

The proposed system implements the MQTT-SN protocol on WSN in smart city applications. This implementation functions to analyze the performance of the MQTT-SN protocol when applied to WSN-based smart city applications using smartcities sensors.

Implementation is built on the waspmote hardware using the MQTT-SN protocol for sensor communication with the server. Data transmission uses the XBee module to optimize and still be done in real time. The implementation uses a star topology and analyzes the quality of service (QoS) performance.

Fig. 1 shows the system design which contains smart cities sensor board collect data of urban environment. A sensor node uses the battery as a power resource and XBee module as a communication tool. Board waspmote Smart cities sensor consist of temperature sensor, light sensor, dust sensor, and ultrasound sensor.

Publisher is a collection of sensors consisting of particle sensors (air pollution sensors), dust sensors, temperature sensors, and luminosity sensors (light sensors). This sensor is placed in several different places and serves as a data source. A gateway is a device used to connect devices that use the MQTT-SN protocol and are converted into the MQTT protocol. MQTT broker guarantees subscribers to get the desired information from the publisher and handle publish data subscriber. Subscriber / client are client that subscribes or who is interested in getting certain info, registering as an interested person from certain info. The server computer functions to store the data into the database and is ready to be processed for monitoring applications in real time to be forwarded to the subscriber. The database used in research uses a MySQL database.

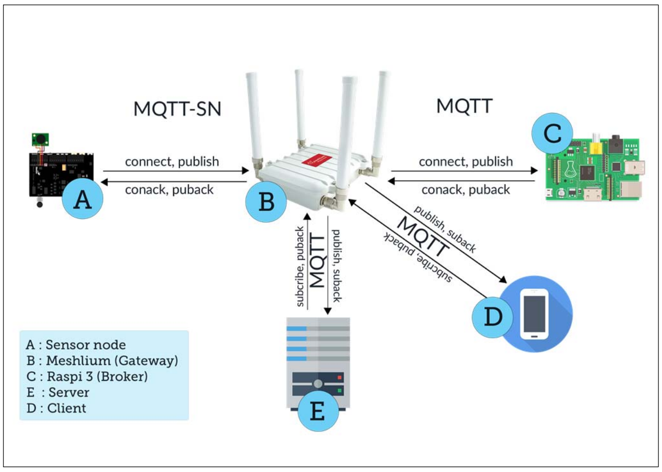

Fig. 1. The system design

The sensor node as publisher will capture the value of urban environment. The publisher placed in some point and collect data from around environment as resource data. It will send the value to the meshlium use the MQTT-SN protocol with Xbee module 802.15.4 Zigbee communication. The Meshlium as gateway will recieve the value and forward to raspberry pi use MQTT protocol with WIFI module. The raspberry pi as MQTT broker receive the value and send data to the all client which subcribe a topic as subcriber. The value of sensor node will be shown on the subscriber side and will be saved in server database MySql.

Communication used between the publisher to the gateway using MQTT-SN protocol, the gateway to the raspberry pi as MQTT broker using MQTT protocol, and The MQTT broker will send data to the subscriber using MQTT protocol. The subcriber can be save data on the storage.

The specifications of the hardware in this research as follow. Waspmote Starter Sensor Kit hardware include: Board Waspmote Pro v1.2, Waspmote 802.15.4 PCB Antenna, Module XBee S1, Module Wifi RN-XV, 2300 mAh LiPo Battery, and miniUSB Cable. Waspmote Smart Cities Sensor Kit hardware include: Board Waspmote Smart Cities Sensor Kit V1.4, Temperature sensor, Light sensor, Dust sensor, Ultrasound sensor EZ0, Meshlium 802.15.4-PRO-3G-AP, and Raspberrry pi 3. Table 1 shows a schema of the database for this paper.

TABLE I. LIST OF SCHEMA

No	Field Name	Type	Function
1	id	int	Serves to define unique id that identifies each insert data.
2	sensor	string	Serves to define the identifier sensor receives of publisher.
3	value	string	Serves to define the value of the sensor.
4	timesend	timestamp	Server to define the period in which the sensor send data to collected.
5	timereceive	timestamp	Server to define the period in which the sensor data collected.
6	node_id	timestamp	Serves to define the identifier sensor receives of publisher.
7	numbersend	Int	Serverto define the number that identifier each sending data

Quality of Service (QoS) is a set of parameters that indicate the quality of a network's services and network capabilities in running applications with performance as required. With measuring the QoS can be known condition of a network and can be made adjustments on the network to the application to be used. Some parameters of QoS measured in this study include:

- Throughput: speed (rate) of effective data transfer, measured in bps (bits per second) or the average speed of data received by a node over a period of time as shown equation 1.

$$\boldsymbol{Throughput} = \frac{\boldsymbol{TotalReceivedBits}}{\boldsymbol{End\ time - Start\ Time}} \quad (1)$$

- Delay: the delay time of a packet caused by the transmission process from one node to another node which is the destination.
- Packet Loss Probability: The ratio of between the numbers of loss packets to the number of packets sent (Sent packet) as shown equation 2.

$$\boldsymbol{PDR} = \frac{\boldsymbol{Loss\ Packet}}{\boldsymbol{Sent\ Packet}} \quad (2)$$

The MQTT-SN protocol is a publish/subscribe protocol for WSN. MQTT-SN can be labeled version of MQTT that is adapted to the character of the wireless communication environment. The publish/subscribe system is a pattern messaging where the message sender called publisher and the recipient of the message called subscriber. The publisher is not programmed to send messages directly to a specific subscriber (recipient). Published messages are marked in channels that may not be known to customers. The customer selects one or more channels, and receives only interesting or selected messages only. Decoupling of the publisher and subscriber allows it to be used for greater scalability and more dynamic network topology.

III. EXPERIMENT AND PERFORMANCE EVALUATION

The system designed and developed to analyze performace MQTT-SN protocol in Smart City-based WSN application. The application to obtain data of various environment parameters like temperature, humidity, light and ultrasound sensor. This research will compare between the MQTT-SN protocol and the HTTP protocol. The MQTT protocol has advantages over the HTTP protocol as shown in table 2.

TABLE II. ADVANTAGES MQTT-SN PROTOCOL

DIFFERENT	MQTT-SN	HTTP
Design	Data centric	Document centric
Pattern	Publish/Subcribe	Request/Response
Header	2-4 bytes	HTTP does not define any limit

Fig. 2 shows the waspmote starter kit. The waspmote is a platform for IoT device that can connect to other smart devices. The waspmote provides variuous features including accelerometer, mini SD card slot, solar socket, battery socket, LED and socket for WIFI module or XBEE module and etc.

Fig. 3 shows the waspmote smart cities board. The waspmote smart cities board is a platform of waspmote for smart cities sensor that can connect to other smart devices. The waspmote provides variuous features including temperature, light, humadity, ultrasound sensor.

A. Configuring board sensor

Before develop smart cities board on starter kit, it is necessary to advance the configuration on the waspmote starter kit board. Install the sensor on smart city board and join the board with starter kit board. Before that make sure that the xbee module has been installed on starterkit as shown in fig. 3.

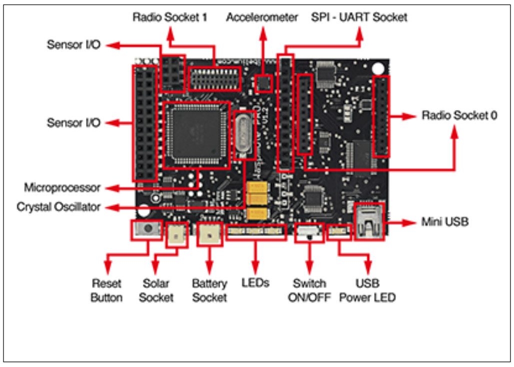

Fig. 2. The waspmote starter kit

The next step is to prepare the program to upload on starter kit board. The program is working to instruct the sensor to be active and its can collecting data from urban environment around.

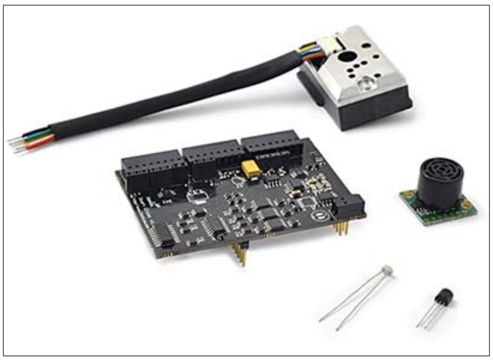

Fig. 3. The waspmote smart cities board

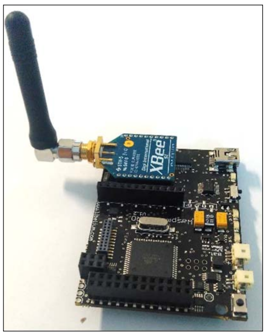

Fig. 4. Xbee module install in starter kit tool

B. Sending data from waspmote to server

The next step is to prepare the program to upload on starter kit board. The program is working to instruct the sensor to be active and its can collecting data from urban environment around. This step use application from waspmote PRO IDE as shown in fig. 5.

C. Configuration Gateway and Broker

After develop the sensor node in waspmote board in this step must configuration meshlium set be gateway for MQTT-SN protocol forward to MQTT protocol as shown in fig. 6. And we must setting raspberry pi be broker MQTT. On raspberry pi install mosquito broker for MQTT.

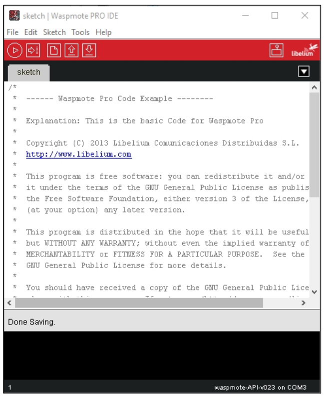

Fig. 5. Waspmote PRO IDE

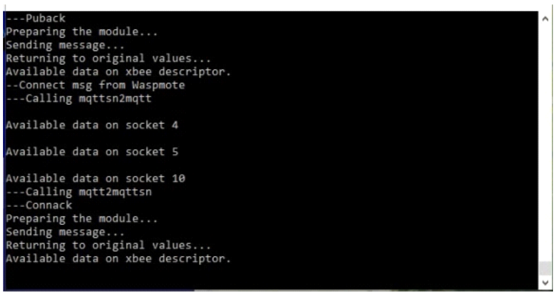

Fig. 6. The meshlium set be gateway for MQTT-SN protocol

D. Retrieval of Data from Sensors

Data retrieval trials are carried out in environmental conditions with sensors and sending data by WIFI and will be received by meshlium servers. This trial uses a serial monitor on the Waspmote IDE. The data obtained from the sensor from the smartcities sensor board before sending, first the node connects to the meshlium if the sensor board successfully connects to the meshlium, the data sensing is sent to the meshlium server. Then, if the sending process to the meshlium server is successful, the server will reply with a reply header 200 "ok", but if it fails the server will reply to an error on the node. Fig. 7 and fig. 8 show the data received by the server and the graph of the sensor data received by the server.

id	sensor	value	timesend	timereceive	node_id	numbersend
1	BAT	3.8	2017-08-15 05:46:06	2017-08-15 05:46:11	smartcity_http	0
2	TEMP	35.4	2017-08-15 05:46:06	2017-08-15 05:46:11	smartcity_http	0
3	LUM	18.3	2017-08-15 05:46:06	2017-08-15 05:46:11	smartcity_http	0
4	DUST	0.0	2017-08-15 05:46:06	2017-08-15 05:46:11	smartcity_http	0
5	BAT	3.8	2017-08-15 05:46:34	2017-08-15 05:46:39	smartcity_http	1
6	TEMP	27.0	2017-08-15 05:46:34	2017-08-15 05:46:39	smartcity_http	1
7	LUM	17.8	2017-08-15 05:46:34	2017-08-15 05:46:39	smartcity_http	1
8	DUST	0.0	2017-08-15 05:46:34	2017-08-15 05:46:39	smartcity_http	1
9	BAT	3.7	2017-08-15 05:46:54	2017-08-15 05:46:58	smartcity_http	2
10	TEMP	29.0	2017-08-15 05:46:54	2017-08-15 05:46:58	smartcity_http	2
11	LUM	17.8	2017-08-15 05:46:54	2017-08-15 05:46:58	smartcity_http	2
12	DUST	0.0	2017-08-15 05:46:54	2017-08-15 05:46:58	smartcity_http	2
13	BAT	3.7	2017-08-15 05:47:13	2017-08-15 05:47:17	smartcity_http	3
14	TEMP	27.0	2017-08-15 05:47:13	2017-08-15 05:47:17	smartcity_http	3
15	LUM	17.7	2017-08-15 05:47:13	2017-08-15 05:47:17	smartcity_http	3
16	DUST	0.0	2017-08-15 05:47:13	2017-08-15 05:47:17	smartcity_http	3
17	BAT	3.7	2017-08-15 05:47:32	2017-08-15 05:47:36	smartcity_http	4
18	TEMP	28.0	2017-08-15 05:47:32	2017-08-15 05:47:36	smartcity_http	4
19	LUM	17.8	2017-08-15 05:47:32	2017-08-15 05:47:36	smartcity_http	4
20	DUST	0.0	2017-08-15 05:47:32	2017-08-15 05:47:36	smartcity_http	4
21	BAT	3.7	2017-08-15 05:47:57	2017-08-15 05:48:02	smartcity_http	5

Fig. 7. Data received by meshlium servers

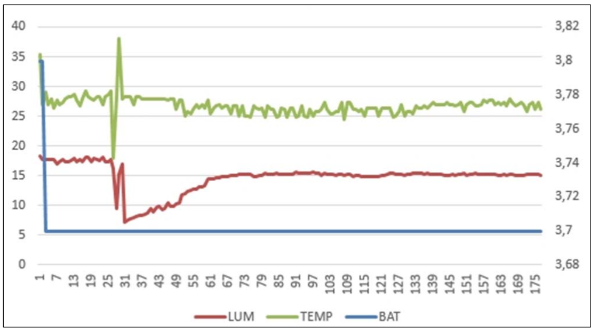

Fig. 8. Graph of sensors received by meshlium servers

E. Analysis and implementation result

In this research use MQTT-SN protocol to be implemented in smart city application based on WSN. The tests were performed using the MQTT-SN protocol, the HTTP protocol, and the CoAP protocol as a comparison and use waspmote board v1.2. The parameters used are delay, throughput, and packet delivery ratio of the protocols, because these parameters have a dominant influence in both protocols. The test is done by sending data from the sensor through the waspmote smart cities board to the meshlium server. Delivery is carried out periodically for 3 hours.

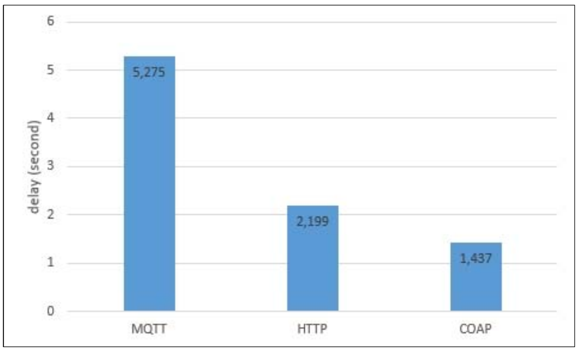

Fig. 9. The result delay measure with range 5 meter

Time delay of a packet caused by the process of transmission from one node to another destination node. Delay testing on the HTTP, MQTT and CoAP protocols shown in fig. 9 begins with an observation time of approximately 3 hours indicating that the delay during HTTP transmission during data transmission varies and has an average delay of 2.199 seconds of the total sent. While the testing delay on the MQTT protocol has a delay of 5.275 seconds and the CoAP protocol has a delay of 1.437 seconds. The biggest MQTT protocol delay is because the process of sending data on the HTTP protocol is directly accepted by the server, while the MQTT protocol sends data from sensors through the gateway and broker.

Fig. 10 shows delay testing on the HTTP, MQTT and CoAP protocols. The delay results when using the HTTP protocol vary and have an average delay of 2.516 seconds of the total sent when the distance is 10 meters. While the testing delay on the MQTT protocol has a delay of 6.765 seconds. While the CoAP protocol has a delay of 1,453 seconds. This shows that the biggest delay is generated in the MQTT protocol, followed by HTTP and CoAP. The biggest delay of MQTT protocol because the process of sending data to the HTTP protocol is directly received by the server, while the MQTT protocol sends data from the sensor through the gateway and broker.

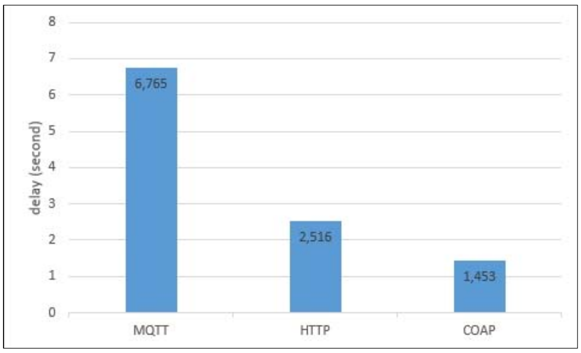

Fig. 10. The result delay measure with range 10 meter

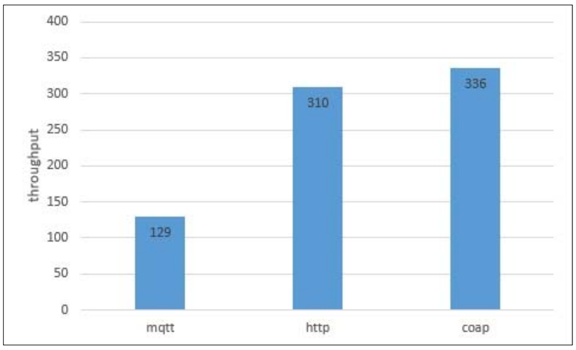

Fig. 11. The result throughput measure range 5 meter

Fig. 11 shows the results of measurement of throughput when using the HTTP protocol, the MQTT protocol, and the CoAP protocol, each measurement was carried out at a distance of 5 meters. Throughput measurements were carried out by observing when sending packages from the Waspmote Smart Cities board to the Meshlium server. Acquisition of throughput is obtained by comparing the initial time of observation to the end of the observation time of all packages sent. When using the HTTP protocol, throughput is 310 bits/s while when using the MQTT protocol the resulting throughput is 129 bit/s and 336 bit/s when using CoAP.

Fig. 12 shows the result of measurement of throughput when using the HTTP protocol, the MQTT protocol of the CoAP protocol is carried out at a distance of 10 meters. When using the HTTP protocol, it produced 307 bit/s throughput while using the MQTT protocol the resulting throughput was 117 bit/s and 333 bit/s when using the CoAP protocol. The above trial shows that MQTT throughput is smaller than the HTTP and CoAP protocol. The throughput on the MQTT protocol is smaller because the activity is only done once and then will CONNACK again when communication is not established. Different from the HTTP protocol where SYN ACK activities are carried out continuously.

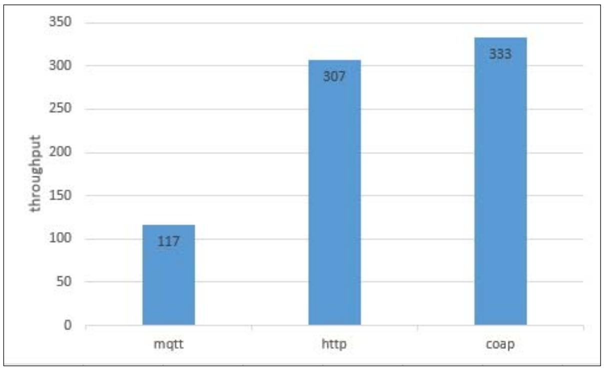

Fig. 12. The result throughput measure range 10 meter

Fig. 13 shows the packet loss measurements in the 5 and 10 meter distance test with 3 hours. The observation time can be seen that the packet loss is 0, this indicates that all packages sent from waspmote smart cities sensor board to meshlium servers are sent perfectly so communication between the two nodes without any missing data.

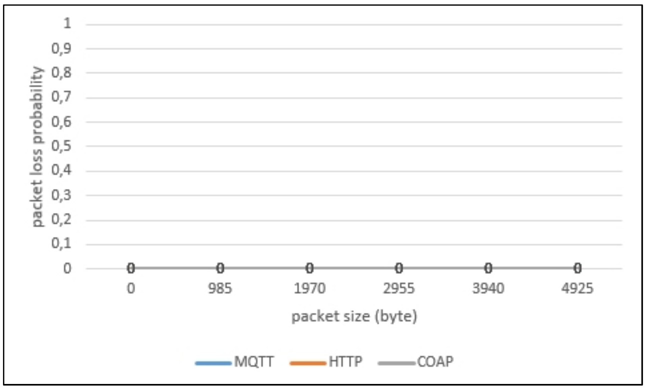

Fig. 13. Packet loss probability

IV. Conclusion

This paper explains the implementation MQTT-SN protocol on smart cities application use waspmote board. The experiments and analysis result show that performance of MQTT-SN protocol, comparison with HTTP and CoAP protocols. In MQTT-SN protocol required broker to handle publish subscribe service. Delay generated by MQTT protocol is bigger than HTTP and CoAP protocol because data transfer process on HTTP and CoAP protocol is received directly by server, while MQTT protocol send data from sensor through gateway and broker. And the throughput generated by the MQTT protocol is smaller than that of the HTTP protocol because the SYN ACK activity on the HTTP protocol is performed continuously, whereas in the MQTT protocol it is only done once and then will CONNACK again when the communication is not established. For the future research, the system will be implemented not only on one node of an end device but will be implemented by using a lot of end devices so that the results of the analysis are more accurate.

Acknowledgment

This research was supported in part by Ministry of Research, Technology and Higher Education of Indonesia, under scheme 'Penelitian Terapan Unggulan Perguruan Tinggi (PTUPT)', Grant No. T/140/E3/RA.00/2019.

References

[1] U. Umar, M.U.H.A. Rasyid, S. Sukaridhoto, "Distributed database semantic integration of wireless sensor network to access the environmental monitoring system," International Journal of Engineering and Technology Innovation (IJETI) 2018.

[2] M.U.H.A. Rasyid, I. Syarif, I.A.H. Putra, "Linked data for air pollution monitoring," International Electronics Symposium on Knowledge Creation and Intelligent Computing (IES-KCIC), 26-27 Sept. 2017, Surabaya, Indonesia.

[3] S.R. Akbar, K. Amron, H. Mulya, S. Hanifah, "Message queue telemetry transport protocols implementation for wireless sensor networks communication — A performance review," International Conference on Sustainable Information Engineering and Technology (SIET), 24-25 Nov. 2017, Malang, Indonesia.

[4] A. Antonić, M. Marjanović, P. Skočir, I.V. Žarko, "Comparison of the CUPUS middleware and MQTT protocol for smart city services," International Conference on Telecommunications (ConTEL), 13-15 July 2015, Graz, Austria.

[5] R.K. Kodali, K.S. Mahesh, "A low cost implementation of MQTT using ESP8266," International Conference on Contemporary Computing and Informatics (IC3I), 14-17 Dec. 2016, Noida, India.

[6] A. Sahadevan, D. Mathew, J. Mookathana, B.A. Jose, "An Offline Online Strategy for IoT using MQTT," International Conference on Cyber Security and Cloud Computing (CSCloud), 26-28 June 2017, New York, USA.

[7] A. Rahman, S. Roy, M.S. Kaiser, Md.S. Islam, "A Lightweight Multi-tier S-MQTT Framework to Secure Communication between low-end IoT Nodes," International Conference on Networking, Systems and Security (NSysS), 18-20 Dec. 2018, Dhaka, Bangladesh.

[8] B. Wukkadada, K. Wankhede, R. Nambiar, A. Nair, "Comparison with HTTP and MQTT In Internet of Things (IoT)," International Conference on Inventive Research in Computing Applications (ICIRCA), 11-12 July 2018, Coimbatore, India.

[9] A. Zabasta, N. Kunicina, K. Kondratjevs, A. Patlins, L. Ribickis, J. Delsing, "MQTT Service Broker for Enabling the Interoperability of Smart City Systems," Energy and Sustainability for Small Developing Economies (ES2DE), 9-12 July 2018, Funchal, Portugal.

[10] M.H. Amaran, N.A.M. Noh, M.S.Rohmad, H. Hashim, "A Comparison of Lightweight Communication Protocols in Robotic Applications," Procedia Computer Science Elsevier, IEEE International Symposium on Robotics and Intelligent Sensors (IRIS 2015).

[11] B. Wukkadada, K. Wankhede, R. Nambiar, A. Nair, "Comparison with HTTP and MQTT In Internet of Things (IoT)," International Conference on Inventive Research in Computing Applications (ICIRCA), 11-12 July 2018, Coimbatore, India.