The Dual-Protocol of the Internet of Things (IoT) Platform for Environment Monitoring System

A Mulyana1*, JA Ualubun2**

1,2Computer Engineering Department, Universitas Komputer Indonesia, Indonesia

Email : *agus.mulyana@email.unikom.ac.id, **jonasualubun@gmail.com

Abstract. The purpose of this study is to create an air pollution monitoring system that has an adaptive capability in the place where the device is used so that it can send data to the server. The method used is to detect GSM signal levels that exist at the place where the device is used, and based on the rules specified then the selector unit will make the selection of the appropriate communication protocol according to the situation, whether using HTTP or MQTT. Based on testing for data transmission, MQTT protocol is more reliable in areas outside cities with low GSM signal levels. Under these conditions, the system is able to pass the selection of protocols correctly so that the system is able to work optimally in sending data. The usefulness of this research is to support reliability in the delivery of devices that work with the principle of internet inf things.

1. Introduction

IoT messaging protocols are also known as Instant Messaging Protocols (IM) and mainly used for chat communication on the Internet. HTTP, MQTT, CoAP, XMPP, and AMQP are protocols which are mainly designed for IoT applications. The properties of these protocols are message management, lightweight message overhead and small messaging [1].

In research [2] describes the monitoring system using a smartphone that has a complete system for cellular participatory sensing, ranging from sensing hardware and client software with support for calibration of web-based data visualization tools. In a journal [3] which proposes an air quality monitoring system based on wireless sensor network technology (WSN). It also integrates with the global system for cellular communication (GSM). This system consists of sensor nodes, gateways. In a journal [4] which proposes a dashboard system to monitor air pollution based on PM2.5. In journal [5] proposes a system to connect various sensors to measure sensor analog data and display it in LabVIEW on a monitor using a graphical user interface (GUI).

The main objective of this research is to create a hardware system for monitoring air pollution levels which are equipped with an automatic selector unit HTTP communication protocol and MQTT to produce a system that is reliable in changing environments.

2. Methods

The method used in this research is to measure the GSM signal level and then compare it with a reference so that it becomes a reference unit for the voter to determine whether to use the HTTP or MQTT protocol. This research was conducted in the electronics lab, with the research partner being the environmental department of the city of Bandung. This system has the main function to take
environmental parameters by using several sensors, then send the parameter data to the server. The testing system is done indoors and outdoors.

2.1 Air pollution
Pollution or contamination is also a change in the composition of airborne substances so that the quality of these substances may be reduced or it can no longer reserve according to its function. In 2010, approximately 3.3 million people worldwide died as a result of prolonged exposure to the dust and pollutants in the air. Meanwhile, air pollution in Indonesia ranks eighth most deadly in the world with an average mortality rate of 50 thousand lives each year [2,9].

2.2 The Internet of Thing’s
The Internet of Thing’s (IOT) a wireless network mechanism that allows any object can communicate [3]. With the presence of IOT control and monitor things/objects from a distance is no longer impossible. IOT system with tools for analysis, monitoring, make a decision and have artificial intelligence. Popular IOT technology is currently used in the world to help solve human problems. IOT can be categorized into three, namely (1) people to people, (2) people-to-machine, and (3) machine to machine. IOT main purpose is to allow everything that can be accessed anywhere, anytime using any network service. IOT architecture consists of several layers of different technologies, but communicate each layer to support IOT system performance. IOT layer following:
1. Smart Device / Sensor layer: Function to retrieve the data, changing the scale physical into scale digital/ electrical [3],
2. Gateway and Networks Works to support the communication path data generated by the sensors, communications support machine to machine (M2M) and application [3],
3. Management Service layer: Function to process the data/information, analysis, security control, process modeling, and management tools [3],
4. Application Layer: IOT application in the form of smart environments in domains: transportation, urban, retail, agriculture, etc (See Figure 1).

![Figure 1. Architecture IOT](image-url)
2.3 Message Queue Telemetry Transport (MQTT)

Message Queue Telemetry Transport Protocol (MQTT) is a protocol used to send the message is very simple and lightweight. MQTT protocol architecture using publish/subscribe openly designed and easy to implement, which is capable of handling thousands of clients remotely with just one server [4,6,7]. MQTT minimize network bandwidth and device resource requirements while trying to ensure the reliability and delivery. This approach makes MQTT protocol is suitable for connecting the machine to machine (M2M) and is an important aspect of the concept of the Internet of things [11]. MQTT architecture as shown in Figure 2.

![MQTT Architecture](image)

Figure 2. Architecture MQTT [11]

3. Results and Discussion

In general, Carepol System is divided into two main systems, namely software and hardware. The following illustrates the system architecture of the whole system (See Figure 3).

![System Architecture](image)

Figure 3. System Architecture

Architecture which is shown in Figure 3, thing explains the content of which is on the hardware, the network describes the rules used to communicate between hardware and software. While on the user interface, an information layer that will be accessible to the public [8] (See Figure 4).
Here's a description of each part table at the block diagram level 2, can be seen in Table 1.

**Table 1. Table Description Block Diagram Level 2**

| No. | Sub-Section                  | Information                                                                                           |
|-----|------------------------------|--------------------------------------------------------------------------------------------------------|
| 1.  | Microcontroller              | Microcontroller Atmega 328p is used to process all of the sensor data and algorithms to perform a predetermined action. The microcontroller is connected to a voltage of 5V. |
| 2.  | Sensor module                | Sensors are used, is a module that already exists, of readings, are then processed in the microcontroller. Each sensor operating at a voltage of 5V. |
| 3.  | Indicator                    | The indicators used in the form of an LED. Where the leg length of the LEDs (VCC) is connected to a voltage source via a resistor. To turn on the indicator, using active-low. |
| 4.  | Communication Module (Wireless) | GSM (Sim 800L) and WIFI (Esp8266), a communication module that is used to transmit data to a web server. |
| 5.  | Power Supplies               | Source voltage is used, divided into two, namely, Solar panel as a battery charger, while the AC voltage source (PLN), is used when available at a place to be in pairs Carepol device. |

Communication protocol testing was conducted to test the suitability of the data sent from the sender and received by the recipient, as well as examine the damage caused to the shipping process. Data is sent through two protocols, HTTP and MQTT (See Table 2).

**Table 2. Data Protocol Testing Results HTTP**

| No | id parameter; "key_alat"; "co"; "co2"; "temperature"; "humidity"; "voltage"; |
|----|--------------------------------------------------------------------------------|
| 1  | "8sbt8201912050402568sbt8"; "19:00"; "25.00"; "32.00"; "59.00"; "100" |
| 10 | "8sbt8201912050402568sbt8"; "20.00"; "27.00"; "32.00"; "59.00"; "100.00" |
| 20 | "8sbt8201912050402568sbt8"; "20.00"; "26.00"; "32.00"; "59.00"; "100.00" |
| 30 | "8sbt8201912050402568sbt8"; "20.00"; "27.00"; "32.00"; "59.00"; "100.00" |
| 40 | "8sbt8201912050402568sbt8"; "20.00"; "27.00"; "32.00"; "59.00"; "100.00" |
| 50 | "8sbt8201912050402568sbt8"; "20.00"; "27.00"; "32.00"; "59.00"; "100.00" |
| 60 | "8sbt8201912050402568sbt8"; "19:00"; "26.00"; "32.00"; "59.00"; "100.00" |
Data showed Table V is data taken from the recipient (server). The following is an analysis error data communication using the HTTP protocol.

transmit total = 105 data of devices
receipt total = 80 data in webserver
error data = 0 data in web

\[
\text{receipt (web)} = \frac{\text{receipt total} - \text{error data}}{\text{transmit total}} \times 100\% = \frac{80 - 0}{105} \times 100\% = 76\%
\]

From the analysis of the HTTP protocol, the data that were successfully received by the webserver is 76% (See Table 3).

| No. | id_parameter; "key_alat"; "co"; "co2"; "temperature"; "humidity"; "voltage" |
|-----|------------------------------------------------------------------------|
| 70  | 8sbt8201912050402568sbt8"; "20.00"; "28.00"; "33.00"; "60.00"; "100.00" |
| 80  | 8sbt8201912050402568sbt8"; "20.00"; "27.00"; "33.00"; "59.00"; "100.00" |

Data shown in Table 4, it is the data taken from the recipient (server). The following is an analysis of data communications using an error protocol MQTT (See Table 4).

transmit total = 105 data of devices
receipt total = 15 data in webserver
error data = 0 data in web

| No. | Supplies (MQTT) | Filling (MQTT) | Supplies (HTTP) | Filling (HTTP) |
|-----|----------------|---------------|----------------|---------------|
| 1   | 4.0            | 3.8           | 4.0            | 3.7           |
| 2   | 4.0            | 3.8           | 4.0            | 3.7           |
| 3   | 4.0            | 3.9           | 4.0            | 3.7           |
| 4   | 4.0            | 3.9           | 4.0            | 3.8           |
| 5   | 4.0            | 3.9           | 4.0            | 3.8           |
| 6   | 4.0            | 4.0           | 4.0            | 3.8           |
| 7   | 4.0            | 4.0           | 3.9            | 3.9           |
| 8   | 4.0            | 4.1           | 3.9            | 3.9           |
| 9   | 3.9            | 4.1           | 3.9            | 4.0           |
| 10  | 3.9            | 4.2           | 3.9            | 4.0           |
| 11  | 3.9            | 4.2           | 3.9            | 4.1           |
| 12  | 3.9            | full          | 3.9            | 4.1           |
\[
\text{receipt (web)} = \frac{\text{receipt total} - \text{error data}}{\text{transmit total}} \times 100\% = \frac{80 - 0}{105} \times 100\% = 100\%
\]

From the analysis on MQTT protocol, the data were successfully received by the webserver is 100%.
From the data in Table 4 and Table 5, made a comparison chart like the following protocol (See Figure 5).

| No. | Supplies (MQTT) | Filling (MQTT) | Supplies (HTTP) | Filling (HTTP) |
|-----|----------------|----------------|-----------------|----------------|
| 13  | 3.9            | 3.9            | 3.9             | 4.2            |
| 14  | 3.9            | 3.9            | 3.9             | 4.2            |
| 15  | 3.9            | 3.9            | 3.9             | full           |
| 16  | 3.9            | 3.9            | 3.9             | 3.8            |
| 17  | 3.9            | 3.9            | 3.9             | 3.8            |
| 18  | 3.9            | 3.9            | 3.8             | 3.8            |
| 19  | 3.9            | 3.9            | 3.8             | 3.8            |
| 20  | 3.9            | 3.8            | 3.8             | 3.8            |
| 21  | 3.8            | 3.8            | 3.8             | 3.8            |
| 22  | 3.8            | 3.8            | 3.8             | 3.7            |
| 23  | 3.8            | 3.8            | 3.8             | 3.7            |
| 24  | 3.8            | 3.8            | 3.7             | 3.7            |
| 25  | 3.8            | 3.8            | 3.7             | 3.7            |
| 26  | 3.8            | 3.8            | 3.7             | 3.7            |
| 27  | 3.8            | 3.8            | full            | 3.7            |
| 28  | 3.8            | 3.8            | 3.7             | 3.7            |
| 29  | 3.8            | 3.8            | 3.7             | 3.7            |

Battery testing aims to determine the length of time the battery decreased when the supply voltage electronic circuits, and the longer it takes to recharge the battery. Here is the test data. The data in Table 5, show data from the testing and use of the battery charging. Charging and use of tested using two protocols with the goal, to know the difference battery consumption, in order to more clearly then graphed as shown in Figure 6.

Figure 5. Comparison of Protocols HTTP and MQTT
Figure 6. Graph changes in the battery voltage

From the graph in Figure 6, displays four curves, the curves were linear rise, indicating charging. While the linear curve down, showing the use of battery power.

4. Conclusion

Based on the results of tests and observations made in this study, it was concluded that the MQTT protocol can be used in this study and works well. When viewed from the total data transmission, the MQTT protocol is safer in sending a lot of data to the server. In addition, savings in power consumption when shipping by using MQTT. The reliability of the battery can last for 26 hours without recharging. In addition, charging takes 2 hours to fully charge the battery. The test results can be seen in Table 6 which shows that using the MQTT protocol, battery usage is more efficient.

References

[1] D Rani, N S Gill, 2019. Review of Various IoT Standard and Communication Protocols, International journal of research and technology, 12(5).
[2] Hasenfritz, D., Saukh, O., Sturzenegger, S., & Thiele, L. 2012. Participatory air pollution monitoring using smartphones. Mobile Sensing, 1, pp.1-5.
[3] Liu, J. H., Chen, Y. F., Lin, T. S., Chen, C. P., Chen, P. T., Wen, T. H., ... & Jiang, J. A. 2012. An Air Quality Monitoring System for Urban Areas Based on The Technology of Wireless Sensor Networks. International Journal on Smart Sensing & Intelligent Systems, 5(1).
[4] Hananto, V. R. 2018. A Dashboard System for Monitoring Air Pollution in Surabaya based on PM2.5.
[5] Telagam, N., Kandasamy, N., & Nanjundan, M. 2017. Smart Sensor Network Based High Quality Air Pollution Monitoring System Using Labview. International Journal on Online and Biomedical Engineering (iJOE), 13(08), pp.79-87.
[6] A. Budiyono, 2015. Air Pollution, The Impact of Air Pollution In Lingkukangan, 2(1).
[7] KK Patel and SM Patel, 2016. Internet of Things-IOT: Definition, Characteristics, Architecture, Enabling Technologies, Applications and Future Challenges, International Journal of Engineering Science and Computing (IJESC), VI(5), pp. 6122-6125
[8] T. Budioko, 2017. “Sriti,” temperature monitoring system using remote protocol based iot mqtt, V1, pp. 353-356
[9] Sawita, WS Gusti Agung Putra Adnyana1,2017. temperature monitoring tool on android applications using lm35 sensor and module sim800l microcontroller atmega16,” Physics Bulletin, 18(2), pp. 58-62.
[10] A. Mulyana, W. and R. Sutriadi Wiradinata, 2018. Materials Science and Engineering, the Internet of Things (IOT) for Urban Detailed Spatial Plan with Zoning Map, p. 8..
[11] Mohammed AL Muslihudin, 2014. Application System Design Admissions in STIMIK Pringsewu Using PHP and MYSQL, Technology Acceptance Model, 3, pp. 33-34
[12] S. Ch, 2010. Performance comparison between cell panel solar cell panel suryaberpenjejakdengan diam. Electrical Technology, 9, pp. 6-7.
[13] Ilham IHZMA Firdiyanto, PLADN Ruth and. NAR, 2017. Check the pollution, Air Quality Interactive Systems, 07(14), p. 14.
[14] A. m. SS, 2015. Parameter Test Equipment Land and Environment Based Android Smartphone, Scientific Journal of Informatics, II, pp. 2407-7658.
[15] A. Mulyana and MN Arifin, 2018. Design for Smart Home Smart Socket-based Message Queuing Telemetry Transport (MQTT), Komputika, 7.