Middleware for Network Interoperability in IoT

Eko Sakti Pramukantoro, Fariz Andri Bakhtiar, Binariyanto Aji, Rasidy Pratama
Faculty of Computer Science
Brawijaya University
Malang, Indonesia
ekosakti@ub.ac.id, fariz@ub.ac.id

***Abstract*—One solution for interoperability issue in IoT is a middleware which is competent on resolving the problems of syntactical, semantic, and network interoperability. In previous study, a middleware capable of addressing semantic and syntactical interoperability challenges has been developed, yet has not responded to network interoperability matter. In this paper we continue our previous research by adding BLE and 6LoWPAN features to the middleware's communication media, so it may communicate with various devices. Interoperability test results show that the middleware is capable of responding to network interoperability challenges and able to receive data from multiple nodes simultaneously.**

Keywords—Middleware, IoT, Interoperability

I. Introduction

Internet of Things (IoT) has rapidly grown and given a quite big impact in daily lives. It enables users to access and manage electronic devices wirelessly through the internet. In the implementation, IoT is facing issues pertaining to device interoperability. The issue arises because IoT is trapped in a "silo" (infrastructure, middleware, and application).

Desai classified this interoperability matter into three: Network Layer Interoperability, Syntactical Interoperability, and Semantic Interoperability. Network Layer Interoperability refers to network protocols used by "things" to connect to other devices; comprising low power networking protocols (Bluetooth Low Energy/BLE, 6LoWPAN) and traditional networking protocols. Syntactical Interoperability refers to the data model or the messaging protocol, e.g. CoAP, MQTT, HTTP, XMPP. Semantic Interoperability refers to the content and data context [1]. To resolve the issue, a middleware supporting interoperability is required. [1] [2].

Previous research has developed a middleware with an event-driven approach that is able to solve semantic and syntactic interoperability issues by providing a gateway to communicate with IoT sensor devices using MQTT and CoAP protocols, and able to communicate with other applications (subscriber) using WebSocket protocol [3]. In its implementation, the communication between the middleware and the sensor still used wireless transmission media, so it has not been able to answer the network interoperability problem.

In an IoT environment, other than Wi-Fi for transmission media, there are BLE and 6LoWPAN which offer low power communication [4]. Communication between the sensor node and gateway using BLE has been implemented in an IoT system prototype by Boualouache. The experimental results showed that the prototype is capable of achieving feasibility, delivery distance up to 6 meters, and efficient power usage[5]. Joshua developed a 6LoWPAN-based sensor node [6], while at other research 6LoWPAN was utilized to arrange communication between a bunch of sensors and a gateway [7]. By evaluating those studies, it then can be concluded that BLE and 6LoWPAN protocols are reasonable choices in providing communication between sensor nodes and the gateway.

In this paper, BLE and 6LoWPAN communication media will be added to the previous middleware, so that it would be a middleware that can answer the challenge of interoperability in general. The discussion at this paper is organized as follows: I. Introduction, II. Existing IoT Middleware, III. Proposed Middleware, IV. Experiment, and V. Conclusion.

II. Existing IoT Middleware

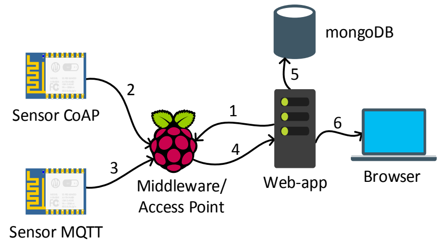

Fig. 1. IoT Environment

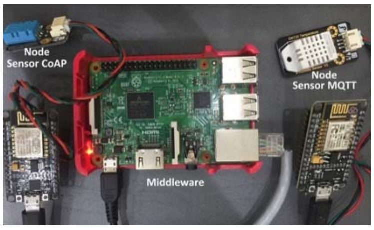

Fig. 2. Midlleware Prototype

Figure 1 represents an IoT-based network architecture that has been developed in previous research, while Figure 2 is an example prototype of a middleware device built from Raspberry Pi. For sensor nodes, NodeMCU ESP8266 equipped with DHT sensors are used. The middleware software consists of: (1) sensor gateway, which is responsible for handling incoming messages from sensor nodes using both the CoAP and MQTT protocols; (2) service unit, which provides an API for storing published data from sensors to Redis as the broker in the system; and (3) application gateway, which provides a WebSocket protocol-based API for exposing topics to subscriber [3]. From the test results, it was found that the CPU and memory usage are under 13% and the message delivery ratio from the sensor node to middleware was under 1 second [8].

III. Proposed Middleware for Network Interoperability

In this study, two communication media, namely BLE and 6LoWPAN, will be added to existing middleware. There are challenges in this research, where both communication media do not work on IPv4. Hence subsystems need to be added at the sensor gateway. Figure 3 shows a 6LoWPAN subsystem

978-1-5386-8402-3/18/$31.00 ©2018 IEEE

added to the sensor gateway so that the middleware be able to communicate using IPv6, then GATT and BLE gateway are added so that the middleware may communicate using BLE.

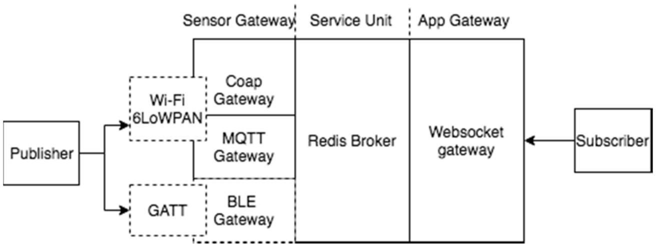

Fig. 3. Adding two communication media to the existing middleware

More detailed discussion will be divided into three parts:

A. BLE Interface

The sensor gateway provides an interface for the middleware so that it can read data transmitted by the BLE client and also serves as a bridge to translate the transmitted data using the BLE network into MQTT protocol, so the initial non-IP-based protocol turns IP-based. The design of the BLE gateway serves to provide an interface for the BLE transmission from the sensor so it can be accepted by the middleware and translated into a MQTT transmission. GATT is required because BLE network cannot directly connect to other devices. GATT defines the services and characteristics of the sensor device to be connected. Connections using GATT are exclusive connections where only one BLE communication between sensors and middleware can occur at any one time. The gateway to be embedded on this Middleware is EspruinoHub[9]. First, the sensor will send data to the middleware. Then the data will be forwarded to the EspruinoHub to be translated into MQTT, as BLE network is non-IP (so it cannot directly transmit data using MQTT protocol which is IP-based).

B. 6LoWPAN Interface

In order for the middleware to communicate with LoWPAN technology, there are several things need to be done: (1) modifying the CoAP and MQTT sensor gateway to listen to IPv6, (2) adding 6LoWPAN communication module. In this research, the MRF24J40MA/RM module is used for 6LoWPAN. The module works to deliver packets to the middleware over the WPAN network.

Noteworthy matters on the 6LoWPAN network configuration are the use of channels, pan id, and same IP network. Table 1 describes the 6LoWPAN configuration used in our IoT environment.

TABLE I. CONFIGURATION 6LOWPAN

Parameter	Sensor node	Middleware
IP Address	fe80::c030:955d:d2b7 :aae5	fe80::c030:955d:d2b7: aae9
Prefix	/64	/64
Channel	11	11
Pan_Id	0x24	0x24

Once the low-level interface is installed, next is to set the middleware software to use the interface. This is done by adding some code so that CoAP and MQTT can listen on IPv6.

C. Sensor Nodes

There are three sensor nodes used in this research: (1) NodeMCU ESP8266 as the Wi-Fi network transmitter, (2) NodeMCU ESP32 as the BLE network transmitter, and (3) Raspberry Pi as the 6LoWPAN network transmitter as shown in figure 5. Each sensor node is directly connected to DHT22 and will deliver payload which consists of humidity and temperature data. The semantics of the payload is shown in Figure 4.

```
Var payload = {
        protocol: protocolName
        timestamp: timeSend
        topic: topicPublish
        sensor: {
            tipe: sensorType
            index: sensorIndex
            ip: ipSource
            module: SensorModule
        }
        humidity: {
            value: valueHum,
            unit: unitHum
        }
        temperature: {
            value: valueTemp,
            unit: unitTemp
        }
}
```

Fig. 4. Sensor's payload

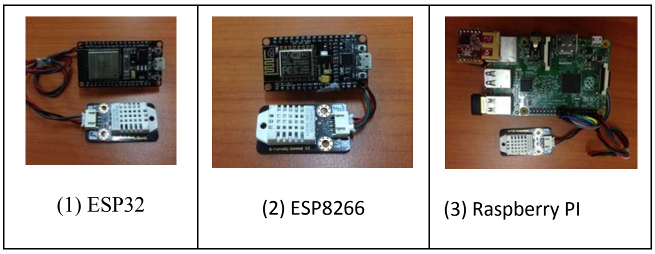

Fig. 5. Sensor Nodes

IV. EXPERIMENT AND DISCUSSION

As discussed in previous research, the middleware was developed on a Raspberry Pi version 3. The version was particularly selected as it already has Wi-Fi and BLE transmission media, so the only additional modules i.e. GATT and EspruinoHub are needed. The experiment is conducted on campus network involving several sensor nodes and one middleware. The sensor nodes will send/publish messages from temperature sensors to middleware. The discussion will be divided into three parts: BLE performance, 6LoWPAN performance, and Network Interoperability Testing.

A. BLE Performace

As described in the previous section, in order for the middleware to be able to communicate with BLE devices, additional modules i.e. GATT and espruinihub are needed. The GATT being used is BlueZ GATT which can run on Raspberry Pi having Raspbian OS.

As depicted in Figure 6, the sensor node having MAC Address 24:0A:C4:10:FC:8E will have its data transmitted over the BLE network to be captured by GATT middleware, then forwarded to the BLE gateway that will translate the BLE transmission into an MQTT transmission using IP Address 10.34.8.5. Figure 7 is print out from the pm2 log that indicates published data from node sensor successfully added to Redis.

```
pi@TheMiddleware:~ $ sudo hcitool lescan
LE Scan ...
24:0A:C4:10:FC:8E ESP32 SimpleBLE
```

Fig. 6. GATT Middleware has detected BLE sensor

```
0|qoap    | 5/30/2018 2:42:52 PM MQTT - Client mqttjs_27f10f1a
publish a message to /ble/advertise/24:0a:c4:10:fc:8e/office/room20

0|qoap    | 5/30/2018 2:42:52 PM MQTT - Client mqttjs_27f10f1a
publish a message to /ble/advertise/24:0a:c4:10:fc:8e/rssi
```

Fig. 7. Data from BLE Sensor to Middleware

Next, the distance change test is done to find out the performance of BLE. Delay in the delivery process is measured and used as the parameter in this test.

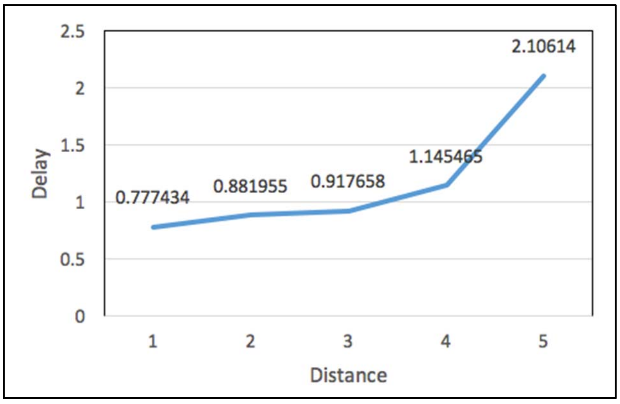

Fig. 8. Influence distance to delay

Figure 8 depicts the test results from the scenario of delay against the distance changes. The test results show that performance on sending data using BLE transmission still has a low delay, even though the distance between Middleware and NodeMCU is different. However, there is a very significant delay change when the distance between Middleware and NodeMCU reaches 5 meters, which is 1 second long. The results of this test indicate that the distance parameter still affects the performance of BLE.

B. 6LoWPAN Performance

In this section, we will discuss transmission data from sensor nodes to IoT middleware using 6LoWPAN, involving MQTT and CoAP protocols. In this experiment, the first sensor node transmits data using CoAP while the second transmit using MQTT. In addition, tests were performed to determine the effect of transmission distance to the delays. The tests were conducted in a public area where some people occasionally passed by until the 6LoWPAN signal was disconnected. The packets were sent for 10 minutes, once every 10 seconds.

```
0|qoap        | 2018-5-30 11:09:13 COAP - Incoming POST request from
fe80::c030:955d:d2b7:aae5 for office/roomA16
0|qoap        | 2018-5-30 11:09:23 COAP - Incoming POST request from
fe80::c030:955d:d2b7:aae5 for office/roomA16
```

Fig. 9. Middleware received data using CoAP

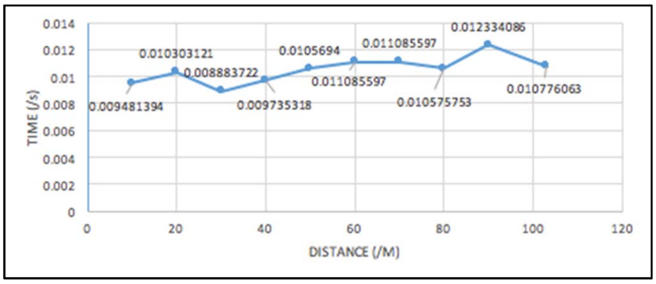

Fig. 10. The effect of distance towards delay in the delivery process using CoAP

The test results showed that 6LoWPAN can reach a distance of 103m. There was an anomaly at a distance of 90m, though. It was due to slight obstruction by an object. 6LoWPAN signals were found to be very weak against objects interference.

In Figure 9, the first sensor node (IP address e80::c030:955d:d2b7:aae5) published data in topic office/room16 to the middleware using CoAP.

```
0|qoap    | 2018-5-30 11:09:17 MQTT - Client
fe80::c030:955d:d2b7:aae6 has connected

0|qoap    | 2018-5-30 11:09:18 MQTT - Client
fe80::c030:955d:d2b7:aae6 publish a message to office/roomA17
```

Fig. 11. Middleware received data using MQTT

Figure 11 shows middleware received data with topic "office/room17" from node sensor 2.

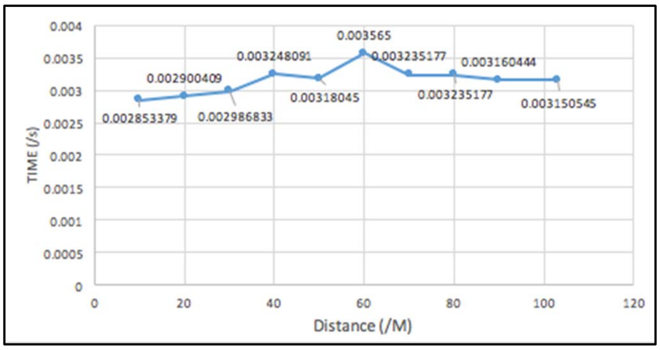

Fig. 12. The effect of distance towards delay in the delivery process using MQTT

In MQTT tests, QoS level 2 was used to focus on average delay, so it needed to take maximum travel time from the amount of data that should be obtained. At a distance of 60m, an anomaly result occurred. It was because the 6LoWPAN module was exposed to direct sunlight during the tests, which decreased 6LoWPAN's performance.

C. Network Interooeprability Testing

The network interoperability tests were done by simultaneously sending data from five sensor nodes to the middleware. Fig. 13 shows that the first node with client id mqttjs_805d5fda published data from sensor nodes using BLE. The second with client id fe80::c030:955d:d2b7:aae5 published data using MQTT protocol on 6LoWPAN, while the third node with client id fe80::c030:955d:d2b7:aae6 published data using CoAP on 6LoWPAN. The fourth node with client id 192.168.42.14 published data using CoAP on Wi-Fi, while the last node with id client 8456747 published data using MQTT on Wi-Fi

```
0|qoap    | 5/30/2018 3:20:39 PM MQTT - Client mqttjs_805d5fda publish a message to /ble/advertise/00.15.83:00:33:e5/office/room20
0|qoap    | 5/30/2018 3:20:39 PM MQTT - Client mqttjs_805d5fda publish a message to /ble/advertise/00.15.83:00:33:e5/rssi
0|qoap    | 5/30/2018 3:20:39 PM MQTT - Client mqttjs_805d5fda has closed connection


0|qoap    | 5/30/2018 3:20:46 PM MQTT - Client fe80::c030:955d:d2b7:aae5 has connected
0|qoap    | 5/30/2018 3:20:50 PM MQTT - Client fe80::c030:955d:d2b7:aae5 publish a message to office/roomA17
0|qoap    | 5/30/2018 3:20:51 PM MQTT - Client fe80::c030:955d:d2b7:aae5 has closed connection

0|qoap    | 5/30/2018 3:20:53 PM COAP - Incoming POST request from fe80::c030:955d:d2b7:aae6 for office/roomA17
0|qoap    | 5/30/2018 3:20:59 PM COAP - Incoming POST request from 192.168.42.14 for office/roomA14

0|qoap    | 5/30/2018 3:23:26 PM MQTT - Client 8456747 has connected
0|qoap    | 5/30/2018 3:23:26 PM MQTT - Client 8456747 publish a message to office/roomA13
0|qoap    | 5/30/2018 3:23:26 PM MQTT - Client 8456747 has closed connection
```

Fig. 13. Print out pm2 logs

The results of interoperability tests show that the middleware can simultaneously receive data from various sensor nodes using heterogeneous transmission media.

V. CONCLUSSION

By observing the experiments' results, it can be concluded that the middleware is able to answer the issue of network interoperability. It means that it is capable to answer overall challenges of interoperability. Some tests were also run to see its performance in messaging. On the parameter of transmission distance, usage of BLE gives a good result at 4 meters with 1 second delay, while 6LoWPAN can reach distance of 103 meters with under 1 second delay. Compared to CoAP, MQTT provides better guarantee in the quality of delivery. This study still limits the communication protocol tests only on the effect of delay, whereas more parameters are notable for testing. In the next study, there will be a more complete comparison between CoAP and MQTT performance to know the quality of data delivery using 6LoWPAN in depth.

REFERENCES

[1] P. Desai, A. Sheth, and P. Anantharam, 'Semantic Gateway as a Service Architecture for IoT Interoperability', in *Proceedings - 2015 IEEE 3rd International Conference on Mobile Services, MS 2015*, 2015, pp. 313–319.

[2] M. A. Razzaque, M. Milojevic-Jevric, A. Palade, and S. Cla, 'Middleware for internet of things: A survey', *IEEE Internet Things J.*, vol. 3, no. 1, pp. 70–95, 2016.

[3] E. S. Pramukantoro and H. Anwari, 'An Event-Based Middleware For Syntactical Interoperability And Enabling Web-Oriented In Internet Of Things', *Int. J. Electr. Comput. Eng.*, vol. 8, no. 5, 2018.

[4] R. Tabish, A. Ben Mnaouer, F. Touati, and A. M. Ghaleb, 'A comparative analysis of BLE and 6LoWPAN for U-HealthCare applications', in *2013 7th IEEE GCC Conference and Exhibition, GCC 2013*, 2013, pp. 286–291.

[5] A. E. Boualouache, O. Nouali, S. Moussaoui, and A. Derder, 'A BLE-based data collection system for IoT', *NTIC 2015 - 2015 1st Int. Conf. New Technol. Inf. Commun. Proceeding*, 2015.

[6] J. Arndt, F. Krause, R. Wunderlich, and S. Heinen, 'Development of a 6LoWPAN sensor node for IoT based home automation networks', *Proc. - 2017 Int. Conf. Res. Educ. Mechatronics, REM 2017*, 2017.

[7] R. Xua, S. H. Yang, P. Li, and J. Cao, 'IoT architecture design for 6LoWPAN enabled Federated Sensor Network', *Proc. World Congr. Intell. Control Autom.*, vol. 2015–March, no. March, pp. 2997–3002, 2015.

[8] E. S. Pramukantoro, W. Yahya, and F. A. Bakhtiar, 'Perform evaluation of IoT middleware for syntactical Interoperability', in *International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, 2017.

[9] espruino, 'espruino/EspruinoHub'. [Online]. Available: https://github.com/espruino/EspruinoHub.