Bridging IoT infrastructure and cloud application using cellular-based internet gateway device

Eko Sakti Pramukantoro*, Maxi Luckies, Fariz Andri Bakhtiar
Faculty of Computer Science, Brawijaya University, 8 Veteran Road, Malang, Indonesia,
tel: (0341) 577-911
*Corresponding author, e-mail: ekosakti@ub.ac.id

Abstract

An Internet of Things (IoT) middleware can solve interoperability problem among "things" in IoT infrastructure by collecting data. However, the sensor nodes' data that is collected by the middleware cannot be directly delivered to cloud applications since the sensor nodes and the middleware are located in intranet. A solution to this problem is an Internet Gateway Device (IGD) that retrieves data from the middleware in intranet then forwards them to cloud applications in the internet. In this study, an IGD based on cellular network is proposed to provide wide-coverage internet connectivity. Two test scenarios were conducted to measure delay and throughput between the IGD and the cloud application; using data from DHT22 sensor and image sensor respectively. The results of the first test scenario using DHT22 sensor show that the average delay is under 5 seconds and the maximum throughput is 120 bps, while the second one using image sensor concludes that the average delay is 595 seconds and the maximum throughput is 909 bps.

Keywords*: GPRS, internet gateway device, internet of things, middleware, publish-subscribe*

1. Introduction

Nowadays the Internet of Things (IoT) is applied in various fields; Sridevi explains the use of IoT in agriculture [1], Amine describes the use of IoT for Measuring Human Activities [2]. IoT can be defined as a system for connecting various kinds of physical objects (things) through a unique addressing scheme and the internet to achieve a single purpose [3]. Meng Ma described that IoT can be described as a 4-layer architecture, which are sensing layer, network layer, a middleware layer, and application layer. Sensing layer consists of devices having the capability to sense and collect raw information. It mostly involves diverse kinds of device, such as a Radio Frequency Identification (RFID) reader, an infrared sensor, Global Positioning System (GPS), camera, etc. The network layer is used to carry out data from the sensing layer to the upper layer by means of IPv4, internet-based IPv6, and cellular network. Middleware layer is a layer for integrating various functions, e.g. device communication management, data processing, semantic reasoning, information security, and service interfaces. The application layer is a layer providing IoT services to the end user in a domain-oriented application [4].

In our previous work, an event-driven middleware has been proposed to overcome the interoperability problem among sensor nodes [5, 6]. Moreover, this middleware has the capability to store data from many kinds of sensor node (e.g. humidity, temperature, images, video) by providing Redis-based message broker. For the further analytic process, heterogeneous data storage based on MongoDB and GridFS has been developed in the Cloud environment. This Cloud system provides API based on RESTful web service to communicate with things or other IoT applications [7]. However, the sensor node's data that is stored in middleware cannot be directly delivered to the Cloud system, because the middleware is on the intranet while the IoT data storage is located on the internet. Integrating IoT infrastructure and Cloud systems is a challenge in a new paradigm called CloudIoT [8]. This problem can be solved by Kamgeue by adding a device that can bridge middleware on the intranet with IoT data storage on the internet [9], known as internet gateway [10]. Furthermore, it must be able to handle sensor data and have the capability to pass information from sensor to the internet and to take care of data [11]. In this paper, we continue our work by proposing an Internet Gateway Device (IGD) based on the cellular network connection. This IGD has two functions: (1) as a subscriber, to collect all data from middleware; and (2) as a forwarder, to send subscribed data

Received October 16, 2019; Revised January 9, 2019; Accepted January 31, 2019

to Cloud system through provided RESTful API web service. Moreover, to measure the performance of the IGD, parameters such as delay, throughput, scalability and reliability testing were used.

2. Related Works

There were studies proposing IGD in IoT environment, such as Lee, which proposed an IoT gateway framework to bridge the semantic gap between raw data in sensor node and information context in high-level applications [12]. The gateway is equipped with internet access to connect sensor node in a local network to Cloud systems. Punitha suggested that IoT gateway should have the capability to pass information from sensor to internet and to take care of data [11]. Byungseok explained that existing IoT gateway solved the communication problem between field sensor and Cloud system. Further, he proposed a new feature in the gateway that supported self-configuration for newly added IoT devices in smart home systems [13]. Qian Zhu proposed an IGD that bridged smart home’s wireless sensor network (WSN) to the internet using 2G/G mobile communication [14]. Meanwhile, for bridging Zigbee network and GPRS network, Hong-jiang proposed IGD that actualized the connection between both network; data conversion between both was also designed along with a solution to data loss [15]. A smart IoT gateway proposed by Shang was equipped with 3G/LTE connection along with Ethernet interface in order to provide different bandwidth requirement using many interfaces [16]. A new Capillary Network Platform having ability to connect WSN to cellular network was proposed by Oscar [17]. Previously, a gateway to bring machine-to-machine (M2M) communication between Capillary Network and 3GPP network was proposed by Shubhranshu [18].

In IoT-based smart home system designed by Kang, 3G connection was used along with Ethernet to produce centralized control to support IoT-based new smart home system which integrates information, telecommunication, entertainment, and living systems [19]. On the other hand, a gateway designed by G. Aloi was deployed in a smartphone in order to meet interoperability among different standards and communications [20]. An IoT gateway proposed by Wenlong utilized GPRS, 3G/4G via hardware implementation along with low-power short-distance wireless interfaces to bring a hardware solution [21]. Several authors has implemented the IoT gateway using low-cost devices like Arduino, such as Grygoruk [22] and Kim [23]; while Raspberry Pi was used by Morabito [24] and Pavithra [25].

Figure 1 shows an IoT infrastructure and Cloud data storage in our previous work, where the IoT infrastructure consists of a middleware and two kinds of sensor nodes. The first is image sensor node built from Raspberry Pi Zero and camera module. It has a function to capture and publish images using MQTT protocol to IoT middleware. The second is the humidity and temperature sensor node built from the DHT22 sensor and NodeMCU ESP8266. It has the capability to publish data using CoAP and MQTT protocol. The middleware runs on Raspberry Pi 3. Those three are connected using wireless LAN provided by middleware. On the other hand, an IoT data storage is also developed to store data from the sensor node for the further analytical purpose. This system is deployed in a virtual private server located in the Cloud environment.

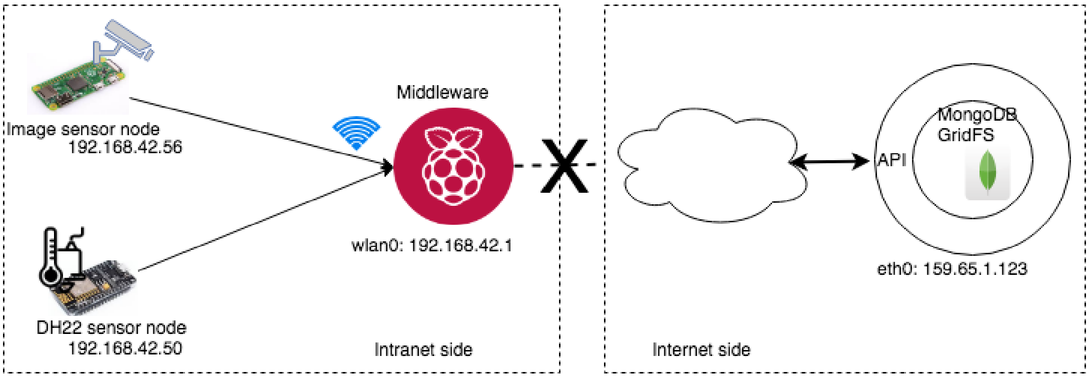


Figure 1. IoT infrastructure and cloud data storage

3. Proposed Internet Gateway Device

As depicted in Figure 2, IGD acts as a subscriber to retrieve all published data stored in the middleware. This is done by subscribing all topics registered in Redis. This will take advantage if the new topic exists, there is no need to add a new topic to the subscriber. After getting the data, IGD will then send it through HTTP POST method to the RESTful API web service provided by IoT data storage. After receiving verification for data arrived at IoT data storage, IGD will flush data related to the previously subscribed topic in the middleware. This action is necessary to ensure that the next data received from middleware is not the previous data that have been stored in the IoT data storage. The pseudocode for subscribing to middleware and sending data to IoT data storage is presented in Figure 3.

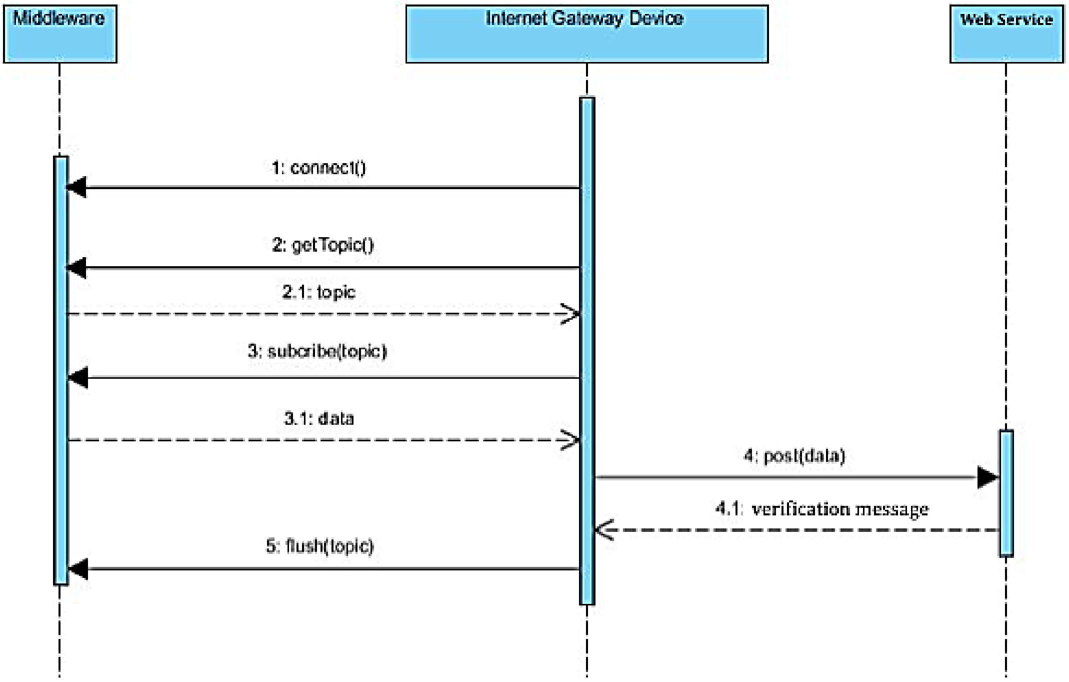


Figure 2. Sequence diagram: middleware, IGD, and IoT data storage

```
DEFINE function getTopic()
   DO send get topics command to Middleware
   RETURN topics

DEFINE function flushTopic()
   DO send flush topics command to Middleware

DEFINE function on_receive(data)
   SET headers = JSON content-type HTTP headers
   DO function sendHTTP(headers,data)

DEFINE sendHTTP(headers,data)
    DO connect to webservice
    DO send data and headers with HTTP POST
    DO function flushTopic()
    RETURN HTTP response

DO connect to Middleware
SET topics = DO function getTopic()
FOR each topic in topics
    SET on_receive callback = function on_receive(data)
    DO subcribe topic
```

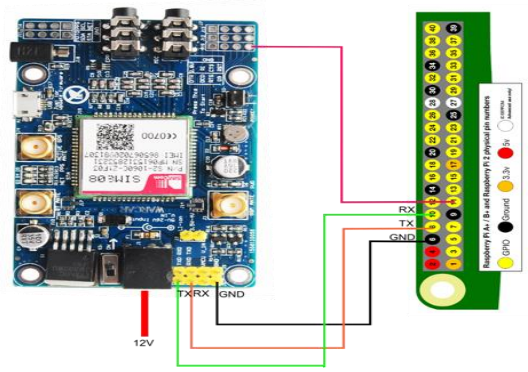
Figure 3. Pseudocode: subscribing and sending data

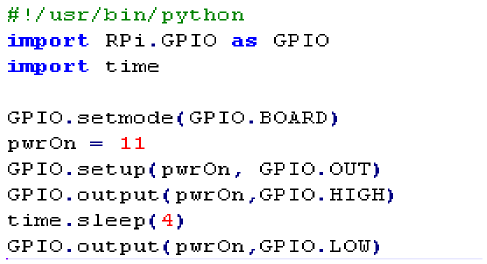
4. Experiment Setup

This section describes how IGD is built and integrated into the IoT environment. The topics used in this paper are office/roomA13, office/roomA14, and office/roomA15. They contain data from sensor nodes in campus rooms.

4.1. SIM808

SIM808 module for Raspberry Pi with GPRS network connection is used by this IGD to be able to connect to the internet. To install the SIM808 module to Raspberry Pi, a scheme shown in Figure 4 is used. In order to work properly, TX pin from SIM808 is connected to GPIO pin 10 functioning as RX pin for Raspberry Pi. Then RX pin from SIM808 module is connected to GPIO pin 8 functioning as TX pin for Raspberry Pi, and the Ground pin from SIM808 module is connected to GPIO pin 6 functioning as a ground pin in Raspberry Pi. For powering the SIM808 module, we use a separated power supply from Raspberry Pi providing 12 volts and 1 ampere. A configuration is needed so that SIM808 module can be turned on automatically after a reboot. We connect D9 pin with one of GPIO digital pin, therefore the command to turn the SIM808 module on can be done by sending a digital signal through GPIO with a Python-based program as shown in Figure 5.





```
#!/usr/bin/python
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BOARD)
pwrOn = 11
GPIO.setup(pwrOn, GPIO.OUT)
GPIO.output(pwrOn,GPIO.HIGH)
time.sleep(4)
GPIO.output(pwrOn,GPIO.LOW)
```

Figure 5. Program to turn on SIM808 module

Figure 4. GPIO connection scheme

4.2. Network Topology

After the IGD is configured properly, the next step is integrating it to IoT environment from the previous work. Its position and the network addressing for this research is depicted in Figure 6. The IGD uses two interfaces: WLAN (192.168.42.50) for communicating with IoT device on the intranet and ppp0 (114.125.124.111) for internet connection.

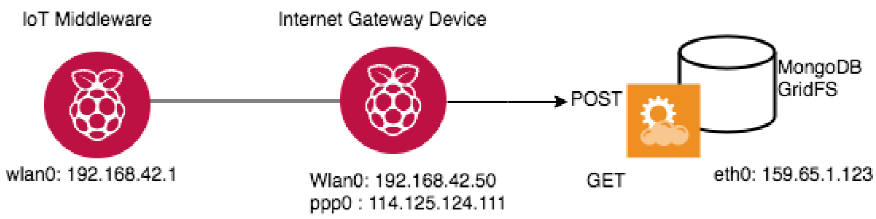


Figure 6. Network addressing

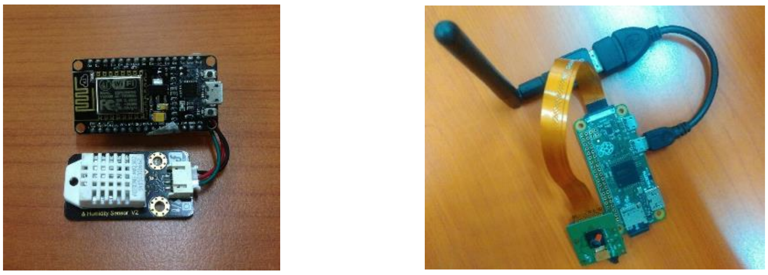
4.3. Sensor Node

This experiment uses three sensor nodes for sensing data. Two of them are NodeMCU ESP8266 equipped with DHT22 for sensing temperature and humidity. They publish data in JSON, as shown in Figure 7.

```
{"protocol": "coap", "temperature": {"unit": "celcius", "value": "24.4.4"},
"timestamp": 0, "humidity": {"unit": "%", "value": "59.2.2"}, "topic":
"office/roomA14", "sensor": {"index": 16446677, "tipe": "esp8266", "module":
"dht22", "ip": "192.168.42.50"}}
```

Figure 7. Semantic data example

The third sensor node is Raspberry Pi Zero with a camera for sensing images in the room and sending them to the middleware. Figure 8(a) shows the sensor node for capturing temperature and humidity data, whilst Figure 8(b) shows the sensor node used to capture image data. Each sensor publishes data on different topics and protocols. In our experiment, we used "office/roomA13" topic for data from DHT sensor using MQTT, "office/roomA14" topic for data from DHT sensor using COAP, and "office/roomA15" topic for the captured image published using MQTT.

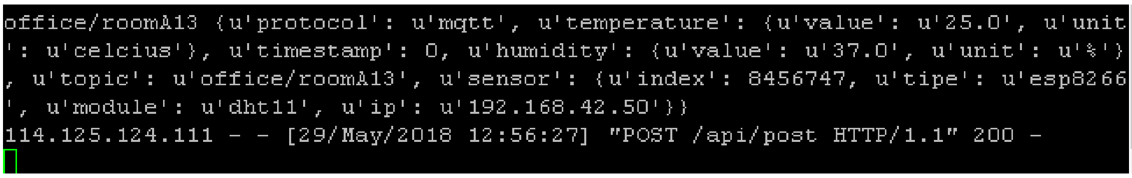


(a) NodeMCU+DHT22 (b) Raspberry Pi Zero+camera

Figure 8. Sensor nodes

4.4. Sending the Sensor Data

After the IGD receives data from the middleware, it will send the data to the IoT data storage through provided RESTful API Web Service. An example result from web service after IGD sends the data is shown in Figure 9.



```
office/roomA13 {u'protocol': u'mqtt', u'temperature': {u'value': u'25.0', u'unit
': u'celcius'}, u'timestamp': 0, u'humidity': {u'value': u'37.0', u'unit': u'%'}
, u'topic': u'office/roomA13', u'sensor': {u'index': 8456747, u'tipe': u'esp8266
', u'module': u'dht11', u'ip': u'192.168.42.50'}}
114.125.124.111 - - [29/May/2018 12:56:27] "POST /api/post HTTP/1.1" 200 -
```

Figure 9. IGD sends data to IoT data storage

5. Result and Analysis

Experimental results about performance are presented in this section. This section will be divided into two parts. Firstly, experiment with scalability scenario. Secondly, experiment with reliability scenario. The experiment was done by subscribing three topics at the same time: office/roomA13, office/roomA14, and office/roomA15 then sending it to data storage.

5.1. Scalability Testing

For scalability in delivering data using GPRS, this experiment is done by publishing topics with several data sizes i.e. 1MB, 2MB, 5MB, and 10MB, which results in delay values as shown in Figure 10. There is 1144 seconds delay for sending 1MB data, 2287 seconds for 2MB data, 5696 seconds for 5MB data, and 11398 seconds (3 hours 9 minutes and 58 seconds) for 10MB data. We conducted some tests to receive and send data for 1 hour as seen in Table 1.

The results show that packet loss occurs at data sending for office/roomA15 topic, with the value of 80%. Another result shown is that the average throughput of office/roomA15 topic is higher than ones of office/roomA13 and office/roomA14 topics.

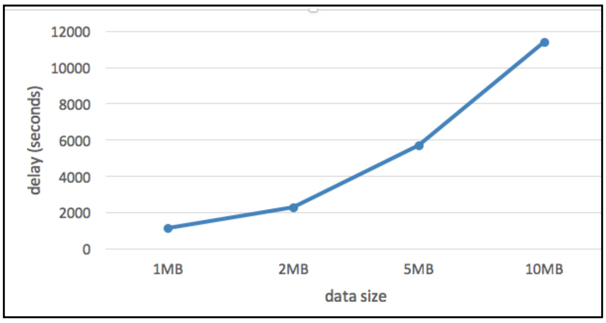


Figure 10. Delay in delivering various kinds of data size

Table 1. Delivery Ratio

Topic	Expected	Sent	Packet Loss	Average Delay	Average Throughput
office/roomA13	20	20	0%	2.919225	120.5075484
office/roomA14	20	20	0%	5.818605	84.34177844
office/roomA15	20	4	80%	595.9642	909.6762563

5.2. Reliability Testing

The reliability test was done by sending 10 data, 20 data, 30 data, 40 data, and 50 data for every topic. Its results show that office/roomA13 topic's average delay is 5.53 seconds for 10 data, 4.68 seconds for 20 data, 4.28 seconds for 30 data, 5.06 seconds for 40 data, and 4.23 seconds for 50 data. Figure 11 depicts those values. As the throughput results, we get 63.5 bytes per second for 10 data, 58.8 bytes per second for 20 data, 63.5 bytes per second for 30 data, 62.8 bytes per second for 40 data, and 55 bytes per second for 50 data. The values of throughput for office/roomA13 topic are depicted in Figure 12.

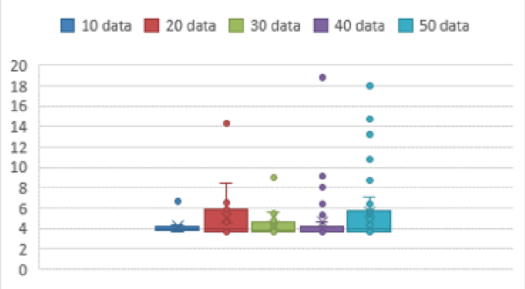


Figure 11. Delay for office/roomA13 topic

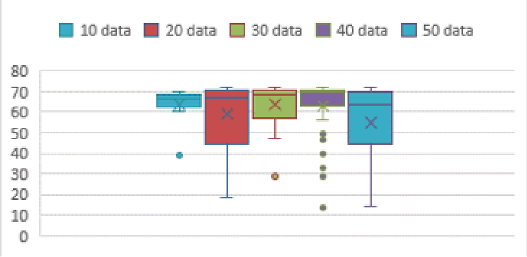


Figure 12. Throughput for office/roomA13 topic

As shown in Figure 13, the delay results of office/roomA14 topic are 3.6 seconds for 10 data, 4.4 seconds for 20 data, 3.9 seconds for 30 data, 5.11 seconds for 40 data, and 4.4 seconds for 50 data. The throughput results are 70.3 bytes per second for 10 data, 59.1 bytes per second for 20 data, 72.8 bytes per second for 30 data, 81.6 bytes per second for 40 data, and 61.62 bytes per second for 50 data, as depicted in Figure 14. The delay results for office/roomA15 topic show 984.47 seconds for 10 data, 1048.2 seconds for 20 data, 1005 seconds for 30 data, 837 seconds for 40 data, and 706 seconds for 50 data, as shown in Figure 15. Figure 16 depicts the throughput results for the same topic, which are 891.55 bytes per second for 10 data, 914 bytes per second for 20 data, 911 bytes per second for 30 data, 913 bytes per second for 40 data, and 912 bytes per second for 50 data.

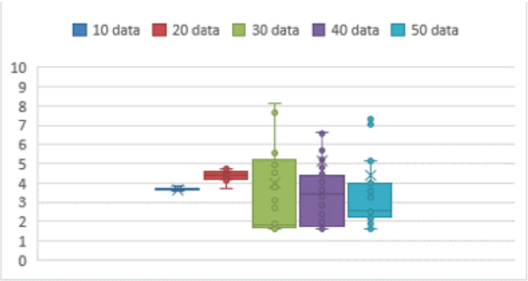


Figure 13. Delay for office/roomA14 topic

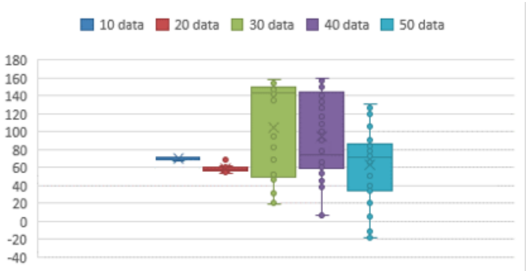


Figure 14. Throughput for office/roomA14 topic

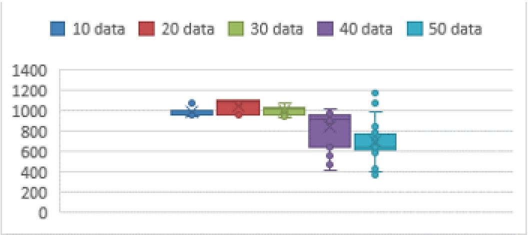


Figure 15. Delay for office/roomA15 topic

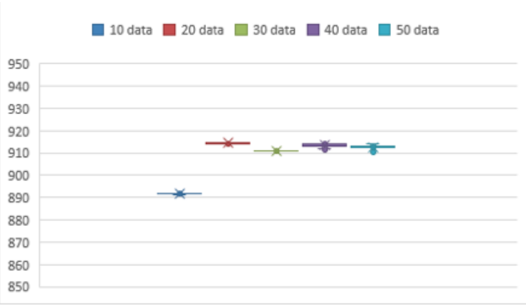


Figure 16. Throughput for office/roomA15 topic

6. Conclusion

We have built an internet gateway device which receives data from the middleware using publish-subscribe approach and uses GPRS network to make internet connection so that data from the intranet (which is the location of the middleware) can be sent to the data storage (which is located on the internet). From the experiments, we get the internet gateway device is also able to send up to 10MB of data, though needs 11398 seconds (3 hours 9 minutes and 58 seconds) delay. The reliability tests are done by sending 10, 20, 30, 40, and 50 data for each topic. From the sending of office/roomA13 topic data, we get 4.79 seconds of average delay and 60.72 bps of average throughput. From the sending of office/roomA14 topic data, we get 4.28 seconds of average delay and 69 bps of average throughput. And from the sending of office/roomA15 topic data, we get 916 seconds of average delay and 908.3 bps of average throughput. We have 80% packet loss for sending office/roomA15 topic because it sends bigger data than office/roomA13 and office/roomA14 topics. In future work for answer higher packet lost in delivering images data, a GPRS module should replace with a 3G/4G module. Otherwise, must be provided separate IGD for images data using 3G/4G and for other small sensor data using GPRS or try with narrowband IoT.

References

[1] Navulur S, Sastry AS, Prasad MG, Prasad G. Agricultural management through wireless sensors and internet of things. *International Journal of Electrical and Computer Engineering (IJECE)*. 2017; 7(6): 3492-9.

[2] Rghioui A, Oumnad A. Internet of Things: Surveys for Measuring Human Activities from Everywhere. *International Journal of Electrical and Computer Engineering (IJECE)*. 2017; 7(5): 2474-82.

[3] Tan L. *Future internet: The Internet of Things*. 2010 3rd Int. Conf. Adv. Comput. Theory Eng. 2010; 5: V5-376–V5-380.

[4] Ma M, Wang P, Chu CH. *Data Management for Internet of Things: Challenges, Approaches and Opportunities*. IEEE Int. Conf. IEEE Cyber, Phys. Soc. Comput. 2013: 1144–1151.

[5] Pramukantoro ES, Anwari H. An Event-Based Middleware For Syntactical Interoperability In Internet Of Things. *International Journal of Electrical and Computer Engineering.* 2018; 8(5): 3784-3792.

[6] Pramukantoro ES, Yahya W, Bakhtiar FA. *Perform evaluation of IoT middleware for syntactical Interoperability*. International Conference on Advanced Computer Science and Information Systems (ICACSIS). 2017: 29-34.

[7] Pramukantoro ES, Yahya W, Arganata G, Bhawiyuga A, Basuki A. *Topic based IoT data storage framework for heterogeneous sensor data*. 11th International Conference on Telecommunication Systems Services and Applications (TSSA). 2017: 1-4.

[8] Alessio Botta, Walterde Donato, Valerio Persico, and Antonio Pescapé. Integration of Cloud computing and Internet of Things: A survey. *Future Generation Computer Systems*. 2016; 56: 684-700

[9] K Kamgueu PO, Nataf E, Djotio T. Architecture for an efficient integration of wireless sensor networks to the Internet through Internet of Things gateways. *International Journal of Distributed Sensor Networks*. 2017; 13(11).

[10] Zhong CL, Zhu Z, Huang RG. S*tudy on the IOT Architecture and Gateway Technology*. 2015 14th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES). 2015: 196–199.

[11] Venkatesan VP, Devi CP, Sivaranjani M. *Design of a smart gateway solution based on the exploration of specific challenges in IoT*. Proc. Int. Conf. IoT Soc. Mobile, Anal. Cloud, I-SMAC 2017. 2017: 22–31.

[12] Lee YH, Nair S. *A Smart Gateway Framework for IOT Services*. Proc. - 2016 IEEE Int. Conf. Internet Things; IEEE Green Comput. Commun. IEEE Cyber, Phys. Soc. Comput. IEEE Smart Data, iThings-GreenCom-CPSCom-Smart Data 2016. 2017: 107–114.

[13] Kang B, Kim D, Choo H. Internet of everything: A large-scale autonomic iot gateway. *IEEE Transactions on Multi-Scale Computing Systems*. 2017; 3(3): 206-214.

[14] Zhu Q, Wang R, Chen Q, Liu Y, Qin W. *IOT gateway: Bridging wireless sensor networks into Internet of Things*. Proc. - IEEE/IFIP Int. Conf. Embed. Ubiquitous Comput. EUC 2010. 2010: 347–352.

[15] He HJ, Yue ZQ, Wang XJ. *Design and realization of wireless sensor network gateway based on zigbee and GPRS*. 2009 2nd Int. Conf. Inf. Comput. Sci. ICIC 2009. 2009; 2: 196–199.

[16] Guoqiang S, Yanming C, Chao Z, Yanxu Z. *Design and implementation of a smart IoT gateway*. Proc. - 2013 IEEE Int. Conf. Green Comput. Commun. IEEE Internet Things IEEE Cyber, Phys. Soc. Comput. GreenCom-iThings-CPSCom 2013. 2013: 720–723.

[17] Novo O, Beijar N, Ocak M, Kjallman J, Komu M, Kauppinen T. *Capillary networks - Bridging the cellular and IoT worlds.* IEEE World Forum Internet Things, WF-IoT 2015 - Proc. 2015: 571–578.

[18] Singh S, Huang KL. *A robust M2M Gateway for effective integration of capillary and 3GPP networks*. Int. Symp. Adv. Networks Telecommun. Syst. ANTS. 2011.

[19] Bing K, Fu L, Zhuo Y, Yanlei L. *Design of an Internet of things-based smart home system*. Proc. 2nd Int. Conf. Intell. Control Inf. Process. ICICIP 2011. 2011; 2: 921–924.

[20] Aloi G, Caliciuri G, Fortino G, Gravina R, Pace P, Russo W, Savaglio C. Enabling IoT interoperability through opportunistic smartphone-based mobile gateways. *Journal of Network and Computer Applications*. 2017; 81: 74-84.

[21] Yue W, *et al. Gatewaying the Wireless Sensor Networks*. 2013 IEEE 9th Int. Conf. Mob. Ad-hoc Sens. Networks. 2013: 61–66.

[22] Grygoruk A, Legierski J. *IoT gateway-implementation proposal based on Arduino board*. Proc. 2016 Fed. Conf. Comput. Sci. Inf. Syst. FedCSIS 2016. 2016: 1011–1014.

[23] Kim SM, Choi HS, Rhee WS. *IoT home gateway for auto-configuration and management of MQTT devices*. 2015 IEEE Conf. Wirel. Sensors, ICWiSE 2015. 2016: 12–17.

[24] Morabito R, Petrolo R, Loscri V, Mitton N. LEGIoT: a Lightweight Edge Gateway for the Internet of Things. *Future Generation Computer Systems*. 2018; 81: 1-15.

[25] Pavithra D, Balakrishnan R. *IoT based Monitoring and Control System for Home Automation*. Proc. 2015 Glob. Conf. Commun. Technol. 2015; 35(1): 24.