

*Article*

## Portable Fog Gateways for Resilient Sensors Data Aggregation in Internet-less Environment

Ahmad Muzaffar bin Baharudin<sup>1</sup>, Mika Saari<sup>2</sup>, Pekka Sillberg<sup>2</sup>, Petri Rantanen<sup>2</sup>, Jari Soini<sup>2</sup>, Hannu Jaakkola<sup>2</sup>, Wanglin Yan<sup>1</sup>

<sup>1</sup> Graduate School of Media and Governance, Keio University, Kanagawa, Japan

<sup>2</sup> Pori Department, Tampere University of Technology, Pori, Finland

E-mail: muzahmad@sfc.keio.ac.jp, mika.saari@tut.fi

**Abstract.** Fog computing is gaining attention due to the potential of aggregating and processing time-sensitive data at the nearby intelligent gateways. It reduces the latency of sensors data aggregation and response time therefore, improve real-time action which is beneficial in environmental monitoring and early warning systems. However, deploying edge computing in Internet-less environment seems unpractical and the mobility of gateways is less focused in current literature. In this paper, we present a practical design of a portable gateways scheme for sensors data aggregation and processing in Internet-less environment. The proposed gateways can locate their geographical locations which can be automatically converted into location names at the central gateway. The proposed portable Fog Gateways are developed by using open-source hardware and integrated with Cloud database for data storage. Data processing techniques such as data parsing and Reverse Geocoding are conducted for reliable data transmission by using GSM/GPRS technology and geographical location name detection respectively. Finally, a case study has been conducted to evaluate the feasibility of our proposed Fog Gateways scheme in real-time application.

**Keywords:** Internet of Things (IoT), fog computing, fog gateways, cloud/crowd system, delay analysis.

**ENGINEERING JOURNAL** Volume 22 Issue 3

Received 12 January 2018

Accepted 13 February 2018

Published 28 June 2018

Online at <http://www.engj.org/>

DOI:10.4186/ej.2018.22.3.221

This article is based on the presentation at *The International Symposium of the 11<sup>th</sup> SSMS and the 5<sup>th</sup> RCND 2017* in Bangkok, Thailand, 20<sup>th</sup>-21<sup>st</sup> September 2017.

## 1. Introduction

Recently, Internet of Things (IoT) has gained tremendous attentions across fields in technology and developer communities as well as applications in social contexts. The significance of various “things” being connected to the internet is the creation of values that contributes for substantial solutions in countless areas such as health-care [1-3], smart city [4-5], industry [6-8] and others. The information created assists in many ways such as optimization, mitigation and adaptation. Connecting “things” together through Wireless Sensor Networks (WSN) feasible especially when the internet connectivity is available. Recent development of IoT research have been focusing on solutions for internet available area such as indoor and urban area. However, the challenges of connecting “things” from internet-less outdoor areas with remote internet network is not tackled well.

In conservative IoT system, the data collected from WSN is delivered and uploaded to Cloud server or database. Problems arise when the deployment area of WSN are physically located far away from the Cloud platforms. In fact, the deployment of WSN in an area without internet connection is a highly concerned challenged. Problems such as operational cost, scalability, latency and Quality of Service (QoS) are critical especially in real-time systems. In this perspective, new paradigms called Fog Computing and Edge Computing besides the widely developed Cloud Computing have emerged. [9] discusses the comparative analysis between all the Cloud Computing, Edge Computing and Fog Computing paradigms, and finally shows the big picture of these three paradigms. [10], [11] and [12] also describe in details about general concepts of Cloud Computing, Edge Computing and Fog Computing for IoT applications.

Environmental issue is one of the most anticipated applications of IoT. For instance, IoT and WSN systems have been developed in various works to collect environmental data such as in [13-15]. In real-time system for environmental sensing applications, the speed of data transmission and gathering process are critical aspect to be considered, especially for natural disaster and hazard monitoring. Thus, latency is one of the major concern in this context. A fast IoT system will enable disaster preparedness and response including through multi-hazard early warning systems. Very often, such area has unstable internet connectivity or completely without internet connectivity.

In this condition, the question is how to aggregate environmental sensors data in such remote areas? Current thinking in IoT which is connecting the WSN to the Cloud directly is not a viable solution for outdoor remote areas. This is because connecting massive amount of sensors with large volume of data streaming to the Cloud platform cannot meet the prospect demand of large-scale IoT applications in the future. Moreover, in real-time data logging and delay-sensitive applications, uploading large amount of sensors data instantly into the Cloud is inefficient in term of data traffic and system reliability. Due to these challenges, a new sensors data aggregation framework is required to enable data aggregation and wireless transmission in such environments. In addition, to reduce the burden of data processing on cloud computing platform a reliable local processing system is needed.

As an alternative to current system, Fog Computing is a new concept of an extended computing on the network edge. It assists networking, computing and storage between the end devices and Cloud Computing data centre, which can be carried out by smart gateways [16]. Instead of uploading data directly into the Cloud, data can be pre-processed through an intermediate layer. Fog Computing is a highly virtualized platform that provides computation, storage, and networking services between end devices and Cloud Computing data centre [17]. It has several distinctions to Cloud Computing in term of delay, communication interface, control, number of users/devices and virtual infrastructure [18]. There are three capabilities of Fog Computing over Cloud Computing which are: 1) selective data aggregation and retrieval from the Cloud; 2) adaptable and scalable upon geographical expansions; 3) real-time decisions and data analytics [19].

In this research, we adopted the concept of Fog Computing as an intermediate medium to the internet since it is useful for sensors data aggregation in remote environments. For this purpose, our system requires geographic location awareness and real-time operation with low latency in data processing and storage.

Figure 1 depicts the position of Fog Computing in our proposed Internet of Things (IoT) system.

The Local Gateways aggregates data from distributed sensor nodes in the Wireless Sensor Networks (WSN). Then, the aggregated data are passed to the Central Gateways and sent to the Cloud. In this context, the computing layer can be classified into three layers, Wireless Sensor Networks (WSN) layer, Fog layer and Cloud layer. The generic edge computing framework is applicable to any applications that requires large scale data aggregation system.

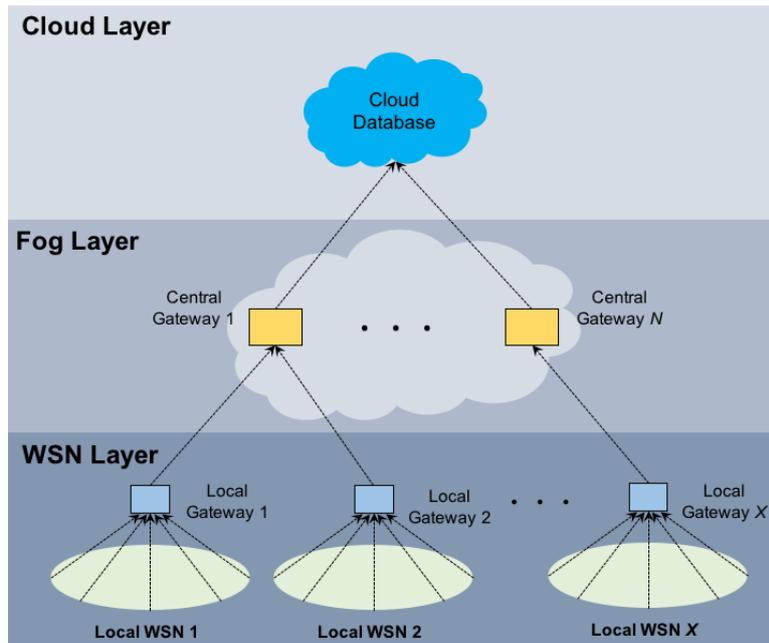


Fig. 1. Overview of Fog Computing in our proposed system.

The rest of this paper is organized as follows. Section II discusses the works related to Fog Computing especially gateways development in previous research and the research gap filled by this research. Section III describes the overview of the proposed Cloud/Crowd system back-end and its components. Section IV presents our analysis on data validation and delay analysis of the proposed framework. Finally, Section V concludes the proposed system with recommendations for future works.

## 2. Related Works

Several studies have demonstrated the design and implementation of gateways for IoT system [20]–[24]. [20] proposed a gateway module with an Ethernet port and connects with the WiMAX gateway thus, a number of local systems can connect to the remote central database and transmit their sensor data wirelessly through WiMAX. Using WiMAX for a large scale monitoring suffers high costs of base station and tower installation since it is not backward compatible with any of the wireless cellular technologies. Besides, WiMAX technology consumes high energy for its deployment. In [21], the design of a general and configurable WSN gateway software system for data collection is proposed. They used MICAz platform with TinyOS as the tested for the gateway system. Rather than designing a gateway with flexible wireless communication capability such as long-range data transmission, the proposed gateway design is intended to be easily configurable for nearly any WSN data collection applications, with diverse data managements and WSN protocols. Besides, [22] proposed a wireless sensor gateway (WSG) within a WSN for health monitoring. The gateway receives data from wireless sensors through ZigBee interface and forward the data to a personal computer through Bluetooth interface. The critical limitation of using these communication interface for a gateway is its short communication range.

Meanwhile in the latest research related to gateways design inspired by Fog Computing, [23] through its preliminary research proposed a Fog Computing based IoT gateway called FOGG. The goal is to bridge the Internet (running on IP, ICN or other future protocols) with the IoT domains running ICN protocols. The role of the proposed gateway is providing Fog based services such as name/protocol translation, security, controller functionality and others. The research in [24] introduced Fog-driven IoT interface (FIT) for processing clinical speech data by using Intel Edison, a Linux-embedded computer. The focus of this work is to demonstrate Intel Edison as a gateway that has computing ability to extract clinically-relevant features that are finally directed to the cloud.

In term of wireless communication interface, the proposed gateways in current literature have either direct connectivity to Ethernet or wireless protocol such as the IEEE 802.16, IEEE 802.15.1 and IEEE 802.15.4. A wide area deployment is not considered in the previous works, in which the proposed gateways

used wireless interface with limited communication range which consequently do not support data aggregation from distributed wireless sensor nodes deployed in outdoor remote and wide area.

The key difference of our system with other state-of-the arts is the exploitation of the legacy GSM/GPRS communication interface for wireless data communication in the Fog Gateway. In this paper, we described in detail the architecture of the proposed system and demonstrated the technique on how the aggregated sensors data could be transmitted successfully through the GSM/GPRS medium and reconstructed again as structured data. Furthermore, we also integrate the data aggregation system by the Fog Gateway with Cloud Database and show the evaluation of the proposed system in term of system delay.

### 3. Cloud/Crowd System Back-End

#### 3.1. System Description

In this work, we proposed a novel system called “Cloud/Crowd System” which integrates large scale data aggregation and retrieval in Fog layer. The target application of this system is for environmental monitoring especially for early warning system for disaster preparedness. Figure 2 illustrates the architecture of the proposed Cloud/Crowd System. The proposed system realizes sensors data aggregation by Fog Gateways in which data are collected into a group or crowd Local Gateways, before being sent and stored in remote Cloud repositories.

The distributed gateways architecture can reduce the intensiveness of sending the data to the Cloud where massive amount of sensors data transmission to the Cloud might lead to transmission failure and significant delay. A hierarchical architecture of Fog Computing is chosen to expand the system scalability. Generally, the more the number of Fog Gateways deployed, the more the volume of sensors data can be aggregated from the distributed Local Gateways which are normally deployed in remote areas. Besides, the data in the Cloud repositories can be fetched and sent back to the crowd (or users) as a feedback for warning or actuation purposes.

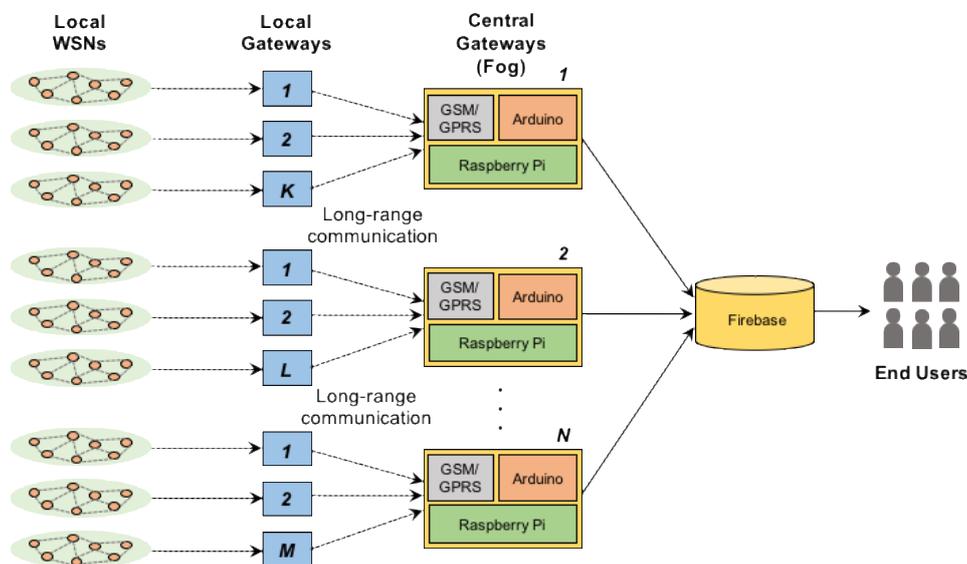


Fig. 2. Overview of the Cloud/Crowd system.

In the proposed system, we assume the deployment of a large scale Wireless Sensor Networks (WSN) in wide and remote areas. The gateways are categorized into two types; 1) Local Gateways and 2) Central Gateways. Local Gateways are assumed mobile and have constraints in internet connectivity, meanwhile the Central Gateways (Fog Gateways) are fixed and have the ability of accessing the internet.

The roles of Local Gateways are to collect local individual data from sensor nodes and process them into bulk data. In addition, Central Gateways aggregate the bulk data from Local Gateways and process them before being stored in the Cloud. In order to improve the system adaptability to internet-free environments,

long-range wireless communication protocol based on GSM/GPRS network is exploited to guarantee a long-range data transmission between Local Gateways and Central Gateways.

In this work, we concentrated on the design of Cloud/Crowd system back-end for monitoring and reporting the number of livestock animals in remote areas. The system back-end consists of the communication between Local Gateways and Central Gateways and data management in Cloud database. Particularly, the roles of data aggregation, processing and retrieval from the Cloud in the system back-end are performed by the Central Gateways. The Local Gateways collect the total number of each types of animals from remote areas and send the aggregated data to the Central Gateways.

### 3.2. Data Communication and Processing in System Back-End

GSM/GPRS is one of the earliest generation of wireless communication protocols. Short Message Service (SMS) has been used for social messaging tools. We leverage SMS as a medium to send aggregated data from Wireless Sensor Networks (WSN). One of the challenge of using GSM/GPRS network to send sensors data is the protocol of communication and data structure where this network is normally used for mobile communication such as Short Message Service (SMS). To our best knowledge, this is the first work that demonstrate the usage of SMS as a medium to transmit sensors data in long range.

Since our aim is to transmit sensors data through SMS medium, data processing is required before sending the data by the Local Gateways and after receiving the data by the Central Gateways. However, our main focus in this work is the data received by Central Gateways and the continuous process of uploading the data into Cloud Database. Here, when the GSM/GPRS receiver in the Central Gateways receive SMS, it reconstructs the sensors data by parsing them. This results in the reconstruction of the tuple. Then, the gateways conduct reverse geocoding to convert the GPS location into location name and includes it in the tuple. Finally, the data is uploaded into the Cloud Database for further processing such as data retrieval and data analytics.

Figure 3 describes the operational flow of data processing conducted by a Central Gateway in the system back-end.

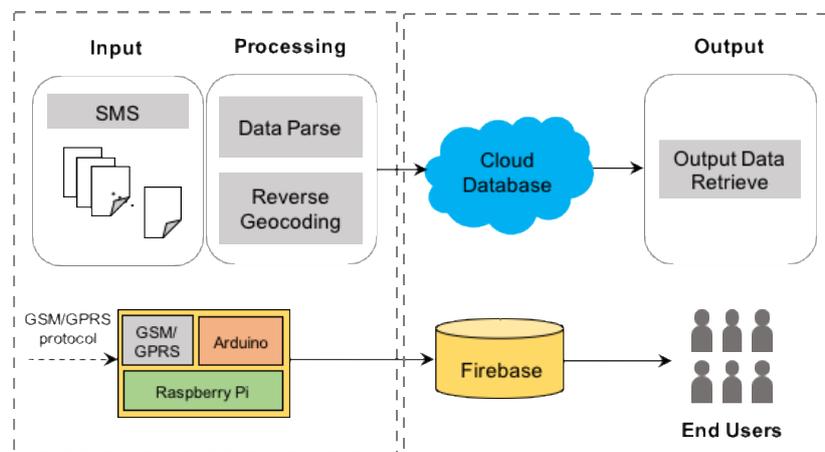


Fig. 3. Data processing in Cloud/Crowd system back-end.

The details of the technical step of the process are explain as below. These process have been carried out through embedded programming on the open source hardware platform, Raspberry Pi [25] and Arduino platform.

#### 3.2.1. Receiving and Parsing Data

The process required as the Central Gateways receive the data is to transform the input data which is in form of SMS into processable and storable datasets by parsing them into a set of tuple. Each SMS contains data string such as sender phone's number (Local Gateway's ID), spatial data in latitude and longitude format and the number of each livestock animal (goat, sheep, cattle, horse and camel) separated by comma signs.

Temporal data is added to the tuple by requesting current date and time when Central Gateway is receiving the data.

### 3.2.2. Reverse Geocoding

The locations of Local Gateways are indicated in latitude and longitude format, which is less significant for the end users. Meanwhile, tracing location's name manually is burdensome. Therefore, we automatically convert the latitude and longitude data into location names through Reverse Geocoding process by using service provided by Google and the location names are added into the tuples afterwards.

### 3.2.3. Uploading Data

The transformed versions of tuples are uploaded into Cloud database through REST API. Each data from the same Local Gateway is stored under the same ID inside the database. This means, the ID is the identification value of the Local Gateways that can be used to specify the source where the sensors data are coming from supported by the geolocation data retrieved from the Reverse Geocoding process.

## 3.3. Hardware Platforms for Central Gateways

Appropriate hardware platforms are selected to carry out specific communication and processing operations mentioned in the previous section. The setup of Local and Central Gateways is shown in Figure 4 (a) and (b) respectively.

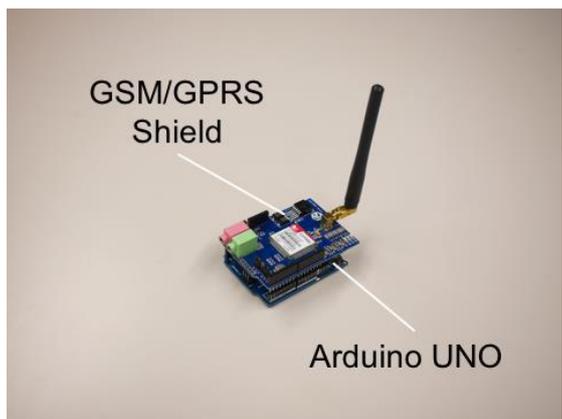


Fig. 4 (a). Hardware set-up of a Local Gateway.

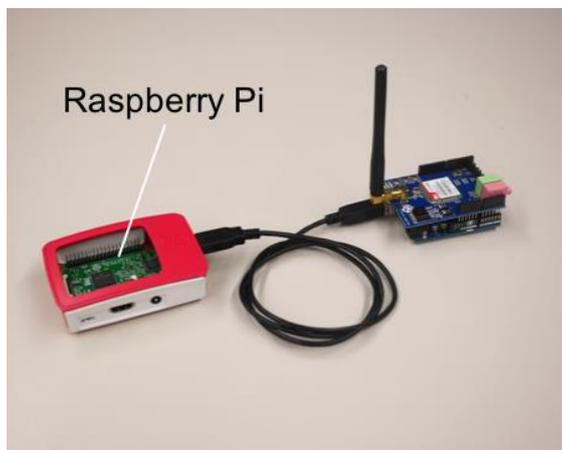


Fig. 4 (b). Hardware set-up of a Central Gateway.

We used Arduino UNO as Local Gateways platform. For Central Gateways, we used the combination of Arduino UNO and Raspberry Pi since they require more powerful processing ability which can be

supported by the Raspberry Pi platform. These open source platforms are adopted because of their high flexibility of embedded programming with high reliability as prototyping platforms as well as for deployment in real applications. Moreover, the small form factor enables the deployment of portable platforms.

### 3.3.1. GSM/GPRS Module

To establish a long range communication link, both Local Gateways and Central Gateways are equipped with GSM/GPRS wireless modules. The GSM/GPRS is advantageous compared to the Ethernet and wireless protocol such as the IEEE 802.16, IEEE 802.15.1 and IEEE 802.15.4 in term of communication range and deployment since it enables a very long range wireless communication across regions in the country. Furthermore, it significantly reduces the additional cost for expensive new facilities construction in remote areas. The Local Gateways act as both receiver and transmitter meanwhile the Central Gateways act as the receiver. We used the SIM900 Quad-band GSM/GPRS Shield, an Arduino compatible shield that enables data retrieval from a remote location via SMS, GPRS or audio service. The module has quad-band which are 850 MHz, 900 MHz, 1800 MHz and 1900 MHz that works on GSM around the world, based on the band available in the specific country.

### 3.3.2. Arduino UNO

Arduino UNO is a compact-sized open source microcontroller broadly used for fast prototyping, appropriate to conduct real-time and repetitive computing tasks. It is a small-sized board with ATmega328 microcontroller based on 32 KB Flash Memory, 2 KB SRAM and 1 KB EEPROM. In our proposed system, Arduino UNO is used as a microcontroller in both Local Gateways and Central Gateways. The role of Arduino UNO is commanding the communication instructions and conducting lightweight data processing. The specifications of Arduino UNO is sufficient as a Local Gateway to perform lightweight computation for sensors data aggregation.

### 3.3.3. Raspberry Pi

Raspberry Pi is a Linux embedded computer board which is programmable in various programming language and can be connected directly to the internet. It consists of the Broadcom BCM2836 (CPU, GPU, DSP, SDRAM) SoC, 900 MHz quad-core ARM Cortex A7 CPU and Broadcom VideoCore IV @ 250 MHz GPU. Besides, it has 1 GB memory (shared with GPU), 10/100Mbps Ethernet port, and 7 GPIO plus specific functions. In our case, we use Python language for data parsing, Reverse Geocoding and uploading in the Raspberry Pi. Using Raspberry Pi board with Arduino UNO can reduce the heavy computing burden on Arduino UNO.

## 3.4. Cloud Database

We chose Firebase as our Cloud database. Firebase is a Cloud services provider and back-end as a service (BaaS) providing services such as real-time database, authentication, analytics, storage and Cloud messaging. In the real-time database, REST API is provided to store and sync data across multiple clients. To interact with the REST API, library is used through Python in embedded platform. HTTP method such as POST, GET, PUT, PATCH, and DELETE which correspond to Create, Read, Update, and Delete (CRUD) operations are used. Firebase is selected for our prototype because it supports real-time database, REST API and JSON format for data storage, import and export. User authentication service is also an advantage of Firebase for future data access by multiple users.

## 4. Evaluation

### 4.1. Data Validation

In this evaluation, we validated the results of data communication and processing in the system back-end particularly on receiving, parsing, Reverse Geocoding and uploading data in the Central Gateways side. We validated that the output data are reliably uploaded into Firebase's real-time database and those uploaded data are consistent with our input data in few repeated trials.

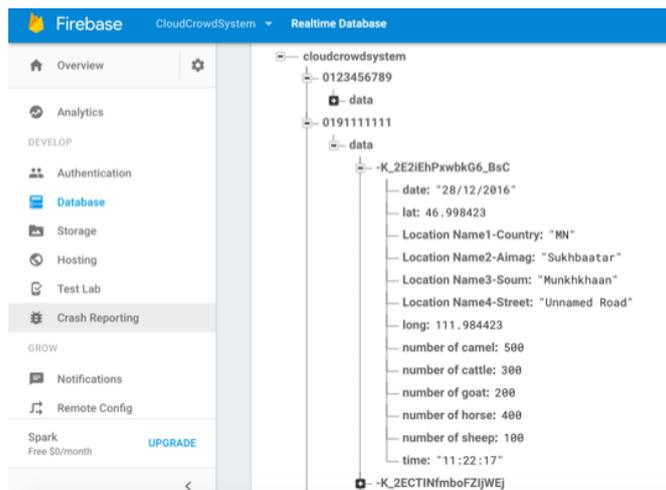


Fig. 5. Data stored inside Firebase’s real-time database.

### 4.2. Processing Time Analysis

We investigated the average time taken for processing operations conducted by the Central Gateway to analyse the time distribution in various processing tasks. To characterize in detail, we classify the operations into three processes as; Process 1: Receiving and Parsing, Process 2: Reverse Geocoding and Process 3: Uploading to Firebase. A total of 50 iterations for similar sensor datasets were executed with 20 seconds time interval between incoming data. The content and value of data are kept constant. The result plotted in figure 6 shows a steady trend of time taken for each processing task.

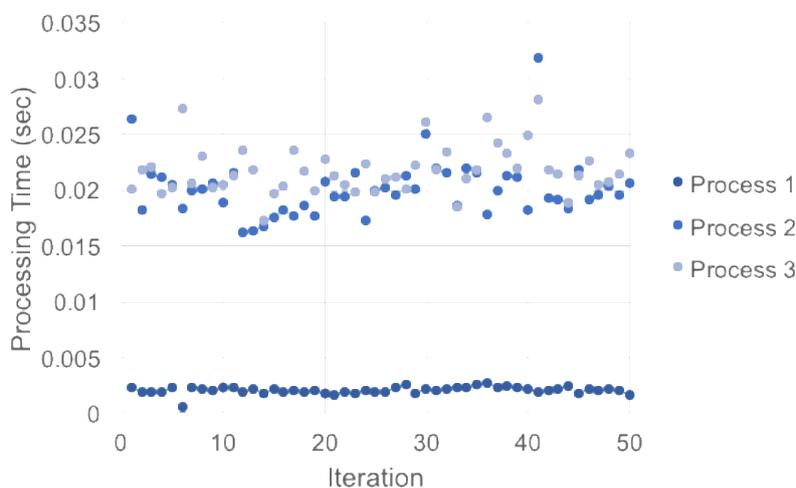


Fig. 6. Processing time for repeated iterations.

The average processing time for each process is also calculated, as shown in table 1. Operations involving Reverse Geocoding and Firebase are approximately 10 times longer than receiving and parsing data in Arduino platform.

Table 1. Average processing time taken for each process.

Average Processing Time 1 (ms)	Average Processing Time 2 (ms)	Average Processing Time 3 (ms)
2.03	20.03	21.72

### 4.3. Cumulative Delay Analysis

We also assessed cumulative delay to observe the total delay caused by processing tasks from receiving data to uploading into Cloud database. From the result shown in Figure 6, the time taken for internal processing generally does not affect the total delay when the input data are received at a constant interval. However, input data received by the Central Gateways are coming from multiple Local Gateways, with different time interval. This might yield delays due to different geographical distance and unsynchronized timing.

To observe the effect of time interval between incoming input data on the total delay of all processes in the Central Gateway, we conducted 50 iterations for every time interval of input data insertion ranging from 20 seconds to 200 seconds. The start time is measured before the connection to serial communication is established. From Figure 7, larger time intervals between input data insertion causes larger cumulative delays. The cumulative delays are up to 1.5 second for time intervals between 20 to 120 seconds. Meanwhile for time interval between 140 to 200 seconds, the cumulative delays are approximately double. Longer time intervals might cause larger delays which can lead to failure in data retrieval.

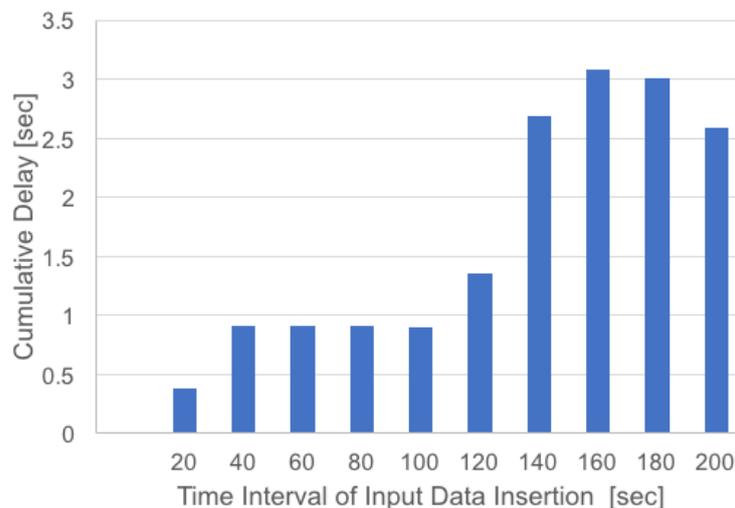


Fig. 7. The effect of different time intervals of input data insertion on cumulative delay.

### 4.4. Specific Delay Analysis

To assess delays in each process, we examined how different time intervals of input data insertion affect the delay of data processing and data uploading by the gateways. Figure 8 shows the impact of different time intervals of input data. Reverse Geocoding and data uploading shows fluctuating delays, however in an increasing trend as the interval becomes larger. Delays in receiving data also show an increasing trend even though they are not much significant compared to the delays in Reverse Geocoding and data upload into Firebase.

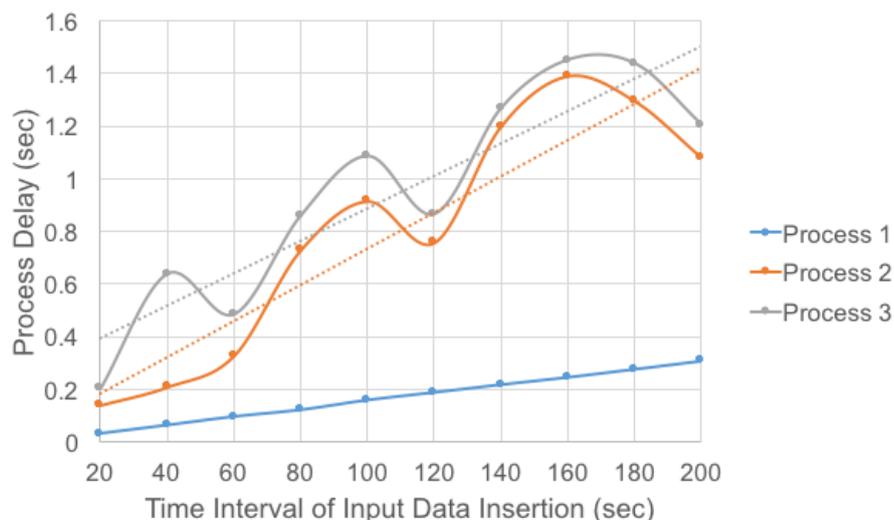


Fig. 8. Time intervals of input data insertion versus delay in each process.

## 5. Conclusion

In this paper, we introduced a new scheme of portable Fog Gateways as a medium for Fog Computing in the Cloud/Crowd system back-end. The deployment of the proposed Fog Gateways supports larger integration of Wireless Sensor Networks (WSN) with Cloud services. The proposed system has advantages for environmental monitoring system in term of cost, portability and the flexibility of deployment over other systems. In the Fog Gateway prototype, we addressed the time delays of processing tasks which can affect the performance of gateways in real-time applications. It is significant to consider the cumulative delays caused by different time intervals of the input data since different applications may comply different time delays. By examining the impact of input time interval on the output delays, we estimated the time buffer required between the period of receiving and retrieving data as a solution to avoid failure in data retrieval. As future works, the performance comparison of Cloud/Crowd system with and without multiple Fog Gateways deployment can demonstrate how Fog Computing could improve the whole system performance. Moreover, other constraints in Fog Computing such as the limit in numbers of gateway is also another topic of interests for future research.

## Acknowledgement

This work was funded by Keio University Young Researchers Development Monozukuri Special Business Research Grant, Keio University Doctoral Students Research Support Program, Global Environmental Leaders Program (GESL) and supported in part by MEXT Grant-in-Aid for the Program for Leading Graduate Schools in Japan.

## References

- [1] T. Wu, F. Wu, J. M. Redouté, and M. R. Yuce, "An autonomous wireless body area network implementation towards IoT connected healthcare applications," *IEEE Access*, vol. 5, pp. 11413–11422, 2017.
- [2] K. H. Yeh, "A secure IoT-based healthcare system with body sensor networks," *IEEE Access*, vol. 4, pp. 10288–10299, 2016.
- [3] S. Amendola, R. Lodato, S. Manzari, C. Occhiuzzi, and G. Marrocco, "RFID technology for IoT-based personal healthcare in smart spaces," *IEEE Internet of Things Journal*, vol. 1, no. 2, pp. 144–152, 2014.
- [4] D. Shin, V. Sharma, J. Kim, S. Kwon, and I. You, "Secure and efficient protocol for route optimization in PMIPv6-based smart home IoT networks," *IEEE Access*, vol. 5, pp. 11100–11117, 2017.

- [5] T. Anagnostopoulos, A. Zaslavsky, K. Kolomvatsos, A. Medvedev, P. Amirian, J. Morley, and S. Hadjieftymiades, “Challenges and opportunities of waste management in IoT-enabled smart cities: A survey,” *IEEE Transactions on Sustainable Computing*, vol. 2, no. 3, pp. 275–289.
- [6] N. Erratt and Y. Liang, “The design and implementation of a general WSN gateway for data collection,” in *Proc. IEEE Wireless Communications and Networking Conference, WCNC*, 2013, pp. 4392–4397.
- [7] S. Guoqiang, C. Yanming, Z. Chao, and Z. Yanxu, “Design and implementation of a smart IoT gateway,” in *Proc. IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, GreenCom-iThings-CPSCom 2013*, pp. 720–723.
- [8] K. Becher, C. P. Figueiredo, C. Mühle, R. Ruff, P. M. Mendes, and K. P. Hoffmann, “Design and realization of a wireless sensor gateway for health monitoring,” in *Proc. 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC’10*, 2010, pp. 374–377.
- [9] S. Sankar, H. El-Sayed, M. Prasad, D. Puthal, A. Gupta, M. Mohanty, and C.-T. Lin. “Edge of things: The big picture on the integration of Edge, IoT and the cloud in a distributed computing environment,” *IEEE Access*, vol. 6, pp. 1706-1717, 2018.
- [10] P. Bellavista, L. Foschini, and D. Scotece, “Converging mobile edge computing, fog computing, and IoT quality requirements,” in *Proc. IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud)*, 2017, pp. 313 – 320.
- [11] W. Yu, F. Liang, X. He, W.G. Hatcher, C. Lu, J. Lin, and X.J. Yang, “A Survey on the Edge Computing for the Internet of Things,” *IEEE Access*, vol. 99, pp. 1–1, 2017.
- [12] G. Peralta, M. Iglesias-Urkiá, M. Barcelo, R. Gomez, A. Moran, and J. Bilbao, “Fog computing based efficient IoT scheme for the Industry 4.0,” *IEEE International Workshop of Electronics, Control, Measurement, Signals and their Application to Mechatronics (ECMSM)*, 2017, pp. 1–6.
- [13] M. T. Lazarescu, “Design of a WSN platform for long-term environmental monitoring for IoT applications,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 3, no. 1, pp. 45–54, 2013.
- [14] F. Ingelrest, G. Barrenetxea, G. Schaefer, M. Vetterli, O. Couach, and M. Parlange, “SensorScope: Application-specific sensor network for environmental monitoring,” *ACM Transactions on Sensor Networks*, vol. 6, no. 2, pp. 1–32, 2010.
- [15] G. Werner-Allen, K. Lorincz, M. Welsh, O. Marcillo, J. Johnson, M. Ruiz, and J. Lees, “Deploying a wireless sensor network on an active volcano,” *IEEE Internet Computing*, vol. 10, no. 2, pp. 18–25, 2006.
- [16] M. Aazam and E. N. Huh, “Fog computing and smart gateway based communication for cloud of things,” in *Proc. International Conference on Future Internet of Things and Cloud, FiCloud*, 2014, pp. 464–470.
- [17] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, “Fog computing and its role in the Internet of Things,” in *Proc. First Ed. MCC Work. Mob. Cloud Comput.*, 2012, pp. 13–16.
- [18] D. S. Chan, M. S. Prabhu, P. Natarajan, and F. Bonomi, “Improving web sites performance using edge servers in fog computing architecture,” in *Proc. IEEE Seventh Int. Symp. Serv. Syst. Eng.*, 2013, pp. 320–323.
- [19] M. Yannuzzi, R. Milito, R. Serral-Gracia, D. Montero, and M. Nemirovsky, “Key ingredients in an IoT recipe: Fog computing, cloud computing, and more fog computing,” in *Proc. IEEE 19th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks, CAMAD*, 2014, pp. 325–329.
- [20] J. De Huang and H. C. Hsieh, “Design of gateway for monitoring system in IoT networks,” in *Proc. IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, GreenCom-iThings-CPSCom*, 2013, pp. 1876–1880.
- [21] N. Erratt and Y. Liang, “The design and implementation of a general WSN gateway for data collection,” in *Proc. IEEE Wireless Communications and Networking Conference, WCNC*, 2013, pp. 4392–4397.
- [22] K. Becher, C. P. Figueiredo, C. Mühle, R. Ruff, P. M. Mendes, and K. P. Hoffmann, “Design and realization of a wireless sensor gateway for health monitoring,” in *Proc. Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC’10*, 2010, pp. 374–377.
- [23] S. S. Adhatarao, M. Arumaithurai, and X. Fu, “FOGG: A fog computing based gateway to integrate sensor networks to Internet,” in *Proc. 29th International Teletraffic Congress (ITC 29)*, 2017, pp. 42–47.
- [24] A. Monteiro, H. Dubey, L. Mahler, Q. Yang, and K. Mankodiya, “Fit: A fog computing device for speech tele-treatments,” presented at *IEEE International Conference on Smart Computing, SMARTCOMP*, 2016.

- [25] M. Saari, A. M. bin Baharudin, and S. Hyrynsalmi, "Survey of prototyping solutions utilizing Raspberry Pi," in *Proc. 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2017, pp. 991–994.