

# Optimization of Data Communication on Air Control Device Based on Internet of Things with Application of HTTP and MQTT Protocols

F Luthfi\*, E A Juanda, and I Kustiawan

Department of Electrical Engineering Education, Universitas Pendidikan Indonesia,  
Jl. Dr. Setiabudhi No. 207, Bandung 40154, Indonesia

\*fityaluthfi@student.upi.edu

**Abstract.** In this research, we proposed ALDEBARAN (Air Control Device Based on Internet of Things), that is, a device the HTTP protocol which can be applied as a protocol to communicate with the IoT ANTARES (Internet of Things Application and Technology Platform as your Reliable Solution) platform. Wi-Fi connectivity is needed to build a communication between devices and platforms, also to sends data continuously. Because the HTTP has a high overhead protocol resulting in waste of power consumption, so it needs to be optimized. The focus of this research is to optimize data communications on our device to create data communication and power saving efficiently. The method used is implementing the HTTP and MQTT protocols that are combined with several data transmission algorithms. Parameters such as latency and current consumption are compared and analysed to obtain the most optimal combination of protocols and algorithms. The result of this research is that MQTT protocol can be applied to our device and data communication with our scheme becomes more optimal.

## 1. Introduction

Recently, Internet of Things (IoT) received more attention because of the development of wireless technology [1]. The development of wireless technology allows electronic devices to be interconnected into a network and exchange data. IoT are the objects that are interconnected to one another so as to form a group that can obtain data infrastructure and access data [2].

Every day, IoT is developed by individuals and industries resulting in many web domains being used. And in the coming years will grow rapidly because IoT applications for RFID communication systems, NFC, M2M, and V2V continue to be developed [1]. Industry in Indonesia such as PT. Telekomunikasi Indonesia, Tbk opened a new field of research, that is, IoT platform. The field developed is IoT ANTARES platform (Internet of Things Application and Technology Platform as your Reliable Solution). One of IoT devices which have been developed by the authors and colleagues in internship program was named ALDEBARAN.

In 2013, research of lightweight protocol on smartphones said that MQTT was designed for communication with a very limited resource such as sensors, but the protocol can also be implemented on a smartphone [3]. In 2015, research regarding the evaluation of MQTT protocol in terms of the



stability of the mobile communication network said that MQTT performance remained good even though the network is not stable [4]. In 2016, research on comparisons between HTTP and MQTT performance have been compared and declared that MQTT have a better performance than HTTP [5].

The focus of this research is to analyse the data communications on ALDEBARAN by implementing the HTTP and MQTT protocols combined with several data transmit algorithms. The purpose of this research is to optimize data communication, especially to make communication with servers more efficient and make power-efficient data communications on ALDEBARAN devices.

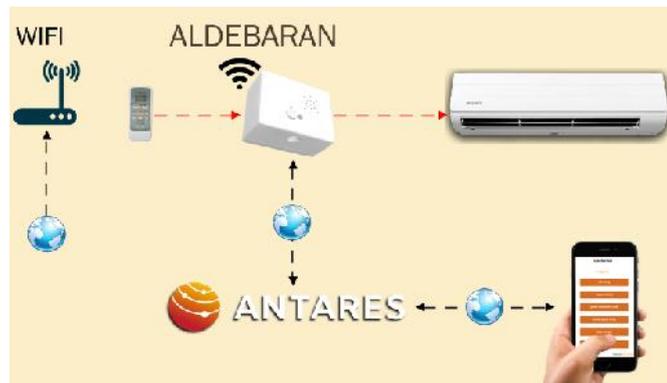
## 2. Research Method

### 2.1 Use Case

ALDEBARAN, a device with a room temperature keeper on air conditioner uses an infrared communication. A device can serve controlling the air conditioner via the Internet, making it easy to monitor room temperature in a building or an office which has a lot of air conditioner. The data logger for the temperature of the room is integrated with the IoT ANTARES platform using a Wi-Fi connection. Our device can be seen in Figure 1.



**Figure 1.** ALDEBARAN Physical Form



**Figure 2.** Illustration of ALDEBARAN Working Principle

The work principle of our device is to replace the conventional air conditioner remote controller into controller remotely via the internet. ALDEBARAN is equipped with LED (Light Emitting Diode) infrared 920 nm as infrared signal transmitter, TSOP34838 infrared receiver as infrared signal receiver, temperature sensor DHT22 as a reader at room temperature, and the microcontroller ESP12-E are packed together with Wi-Fi transceivers so it can be a Wi-Fi client as well as Wi-Fi access point (AP). The device runs with 2 modes, that is, ie running mode and config mode. The working principle of the device can be illustrated in Figure 2.

ALDEBARAN can access an Internet connection from Wi-Fi access point/ router. Communication link is illustrated by a black line so that our device, ANTARES servers, and smartphones can connect via the internet to exchange data. Smartphone data which is sent to the server contain control commands AC, while the data from ALDEBARAN sent to the server contain information of room temperature. Both figures are bridged by IoT ANTARES platform as cloud storage for exchanging data by accessing the API (Application Programming Interface).

### 2.2 Software Aspect

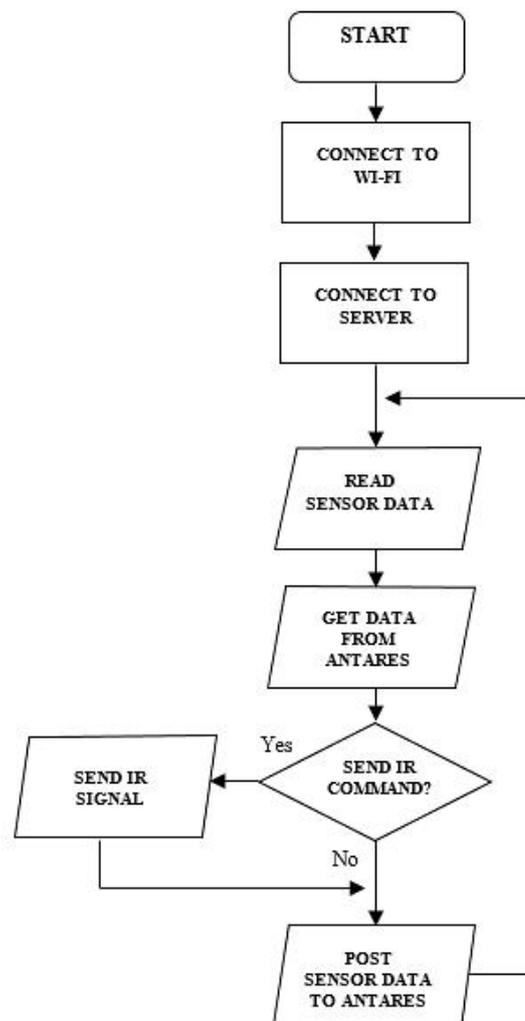
Algorithms to send and retrieve data that have been designed applied to the pre-existing ALDEBARAN algorithm. The certain algorithm only changes the part of running mode. The algorithm is written in C ++ programming language with Arduino IDE (Integrated Development Environment) as a compiler.

The application of this protocol can be done by adding library on Arduino IDE. The HTTP protocol can use the library <WiFiClient.h> while MQTT protocol can use the library <PubSubClient.h>. Library is used to support all the needs of the protocol implementation.

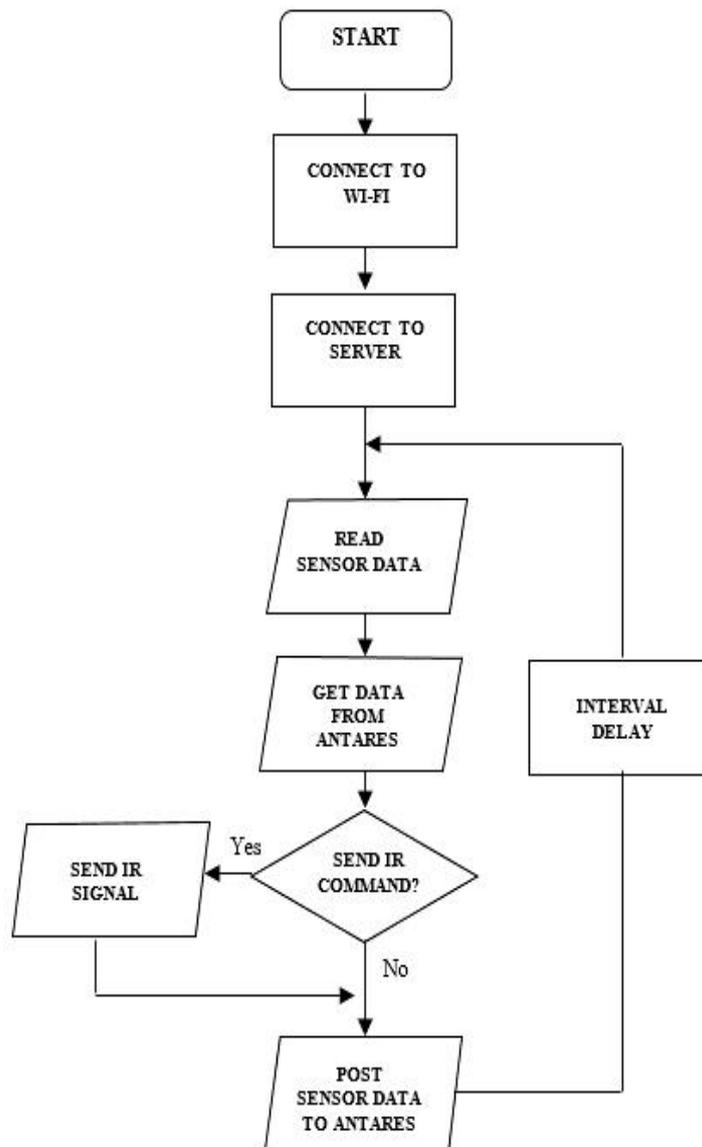
IoT ANTARES platform is used to store sensor data and device control commands, as well as an ideal platform for local developers, due to free account creation and free API access up to 10,000 access/ day.

### 2.3 Algorithm Design

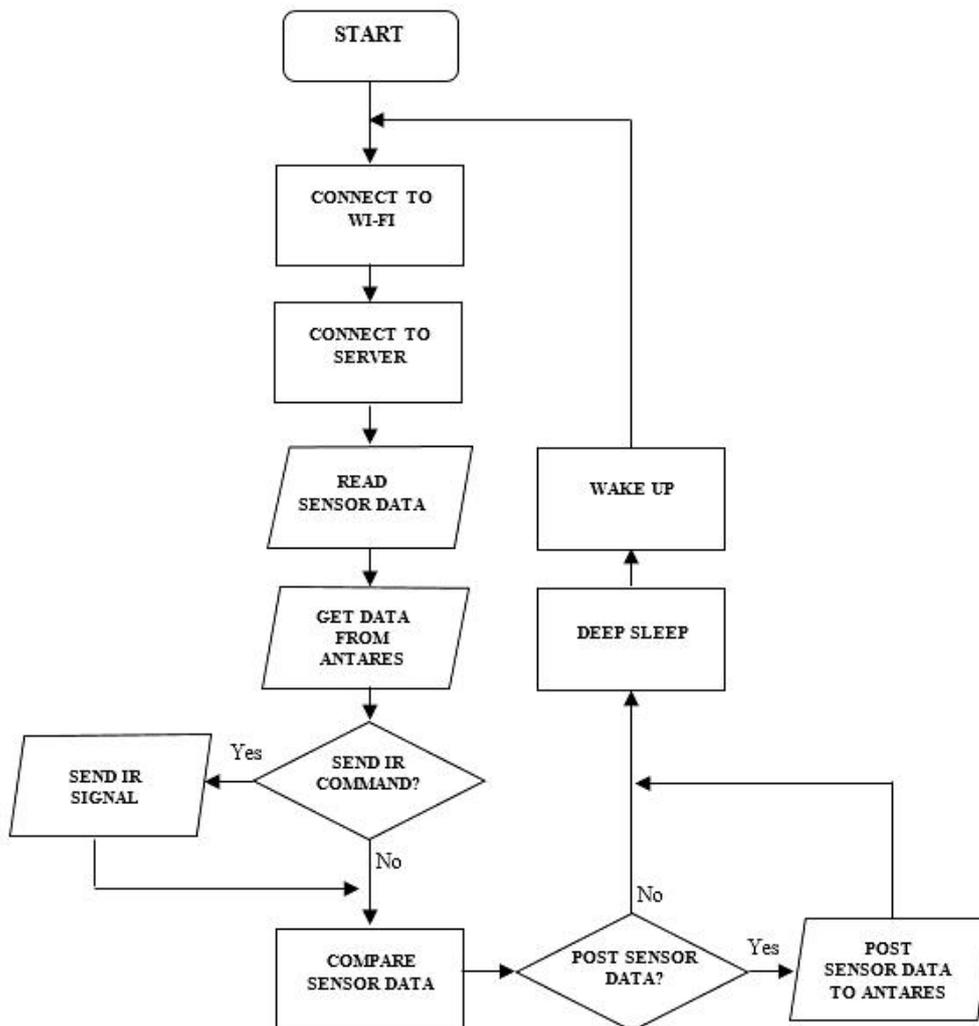
In this research, we apply continuous, periodic and adaptive data transmit algorithms to the running mode of ALDEBARAN. This algorithm is divided into several stages, reading data from sensors, connected to the Wi-Fi, establishing a connection with the server, GET data from the server, decision making IR control systems, and POST data to the server. Some stages are made into several design algorithms and will be applied and compared. The purpose of the design of the algorithm is to get the most efficient in terms of power consumption when the algorithm is applied to our device. The designed algorithms can be applied to devices using Wi-Fi connectivity as ESP8266 and other microcontrollers are equipped with Wi-Fi Shield. The algorithm for storing and retrieving data continuously can be seen in Figure 3(a) and periodically can be seen in Figure 3(b).



**Figure 3(a).** Flowchart of Continuous Data Transmit



**Figure 3(b).** Flowchart of Periodic Data Transmit



**Figure 4.** Flowchart of Adaptive Data Transmit

Flow chart on Figure 3 shows several stages including connecting to Wi-Fi, reading sensor data, establishing a connection with the server, retrieving data from the server that contain AC control commands, AC control decision-making functions, and storing sensor data to the server. In step of connection to Wi-Fi, ALDEBARAN tries to enter the Wi-Fi of the selected SSID. Once the device has connected to the Internet, the device will try to connect to the server. In this research, the host used as a server is platform.antares.id with port 8080 for HTTP and 1883 for MQTT. The device reads sensor data in the room such as temperature and humidity values that are accommodated on the variable with float data type. Once connected to the server, then ALDEBARAN retrieves latest data from a server with access to the IoT ANTARES platform API using HTTP requests with the GET method. The data contains an AC control command sent by the smartphone and data received by ALDEBARAN in the form of data containing headers and body with JSON / XML format. The received data is parsed until an AC control command is obtained.

**2.3.1. Continuous Data Transmit Algorithm** This algorithm describes the stages of storing and retrieving data continuously without delay interval and used for data communication in real time. Our device applies this algorithm to support the air conditioner control functions via infrared communication.

At this step of the decision-making control function, the control command is received then is used as basis for a decision to send an infrared signal. The last step of the algorithm is to store data to the server with access the ANTARES API. The sent data contains value of sensor obtained previously. Furthermore, the program will perform the reading of the sensor data again and repeat the process of storing and retrieving data continuously until the device is turned off.

*2.3.2. Periodic Data Transmit Algorithm* This algorithm describes the stages of storing and retrieving data periodically with interval delay and used for asynchronous data communication. ALDEBARAN applying this algorithm to support the air conditioner control functions via infrared communication.

The difference in this algorithm is delay interval that can be set when ALDEBARAN are in config mode. As the time delay was performed, the status of Wi-Fi is idle. Furthermore, the program will repeat to perform the reading of the sensor data periodically until the device is turned off.

*2.3.3. Periodic Data Transmit Algorithm with Deep Sleep Mode* This algorithm describes the stages of storing and retrieving data periodically with interval delay using the RTC in deep sleep mode and is used for asynchronous data communication. Network reliability and overall improvement of energy efficiency can be done using deep sleep [6]. ALDEBARAN applies this algorithm to support the air conditioner control functions via infrared communication.

The difference, delay interval here is the time to perform deep sleep mode. At the time of deep sleep was performed, the Wi-Fi status and CPU are switched off. Once the countdown of the time interval delay has been completed, then the device will wake-up from sleep mode in the state of Wi-Fi is turn off. Devices need to repeat the steps to connect to Wi-Fi and to connect to the internet. Then the program will repeat to perform the reading of the sensor data periodically and repeat the process until the device is turned off.

*2.3.4. Adaptive Data Transmit Algorithm* This algorithm describes the stages of storing and retrieving data periodically with interval delay using the RTC in deep sleep mode and adaptive phases of data transmission that is used for asynchronous data communication. ALDEBARAN applying this algorithm to support the air conditioner control functions via infrared communication. Storing and retrieving data with adaptive data transmit algorithm can be seen in Figure 4.

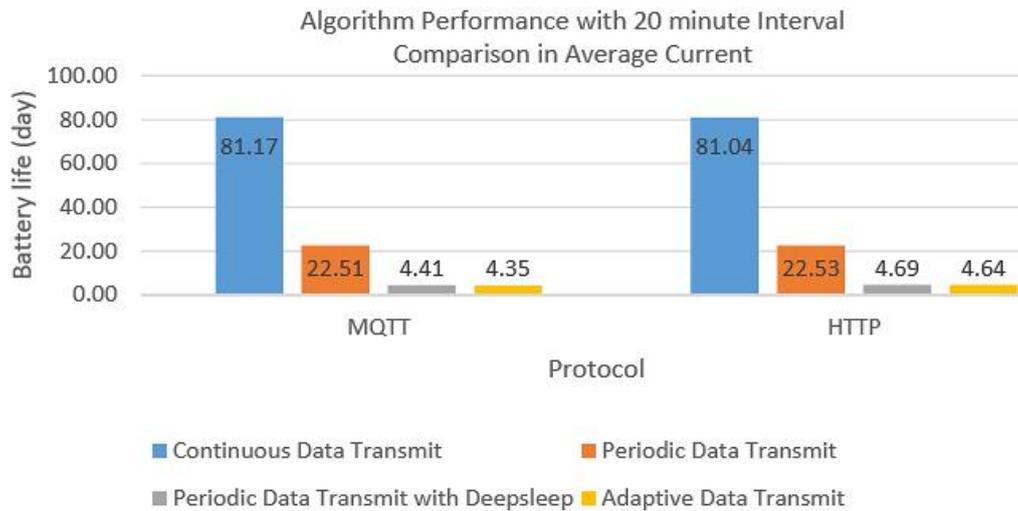
Up to the stage decision-making control function, the algorithm is still the same. The next step is to compare the present sensor data with previous data on the last cycle of deep sleep. Sensor data is sent to the server in the previous cycle and used as a threshold decision. Threshold consists of upper and lower threshold determined by summing  $\pm$  threshold deviation on the values read previously. The threshold deviation can be set in the system config menu. In this research the adaptive algorithm uses a threshold deviation of 1 degree Celsius which means the threshold range of  $\pm 0.5$ . The threshold deviation is determined from the standard temperature changes in air conditioner that is every 1 degree Celsius. If the new data is still within the range of the threshold, then the data may not be sent to the server and the device will be directly into deep sleep mode. If the new data from outside the range of the threshold, the threshold is updated and the data is sent to the server and the device may enter the deep sleep mode. As the time of deep sleep was performed, the Wi-Fi status and CPU are switched off. Once the countdown of the time delay interval has been completed, then the device will wake-up from sleep mode in the state of Wi-Fi is turn off. Devices need to repeat the steps to connect to Wi-Fi for accessing the internet. Then the program will repeat to perform the reading of the sensor data periodically and repeat the process until the device is turned off.

The time to perform one iteration cycles depends on the time connected to Wi-Fi, establishing a connection with the server, the reading of sensor data, data retrieval from the server, sending data to the server, and sleep mode. The total amount of time is the system cycle time of this algorithm.

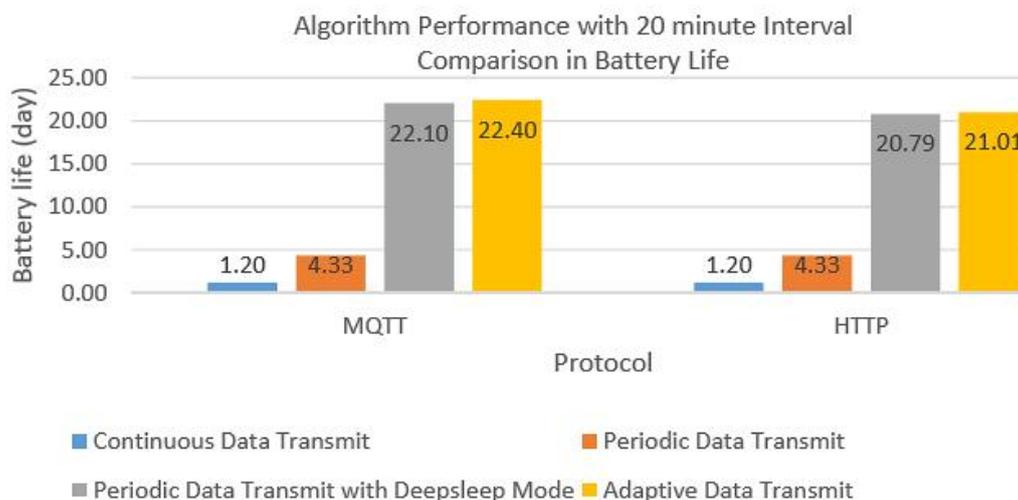
### 3. Results and Discussion

#### 3.1. Algorithm Performance Comparison

The experiments are done through transmitting data every 20 minutes and the results are compared on any combination of algorithms with protocols. The result of algorithm implementation is system cycle time, then obtained the average current and calculated the battery usage in each combination of algorithm with HTTP and MQTT protocol. Algorithm performance comparison results at intervals of 20 minutes can be seen in Figure 5 and 6.



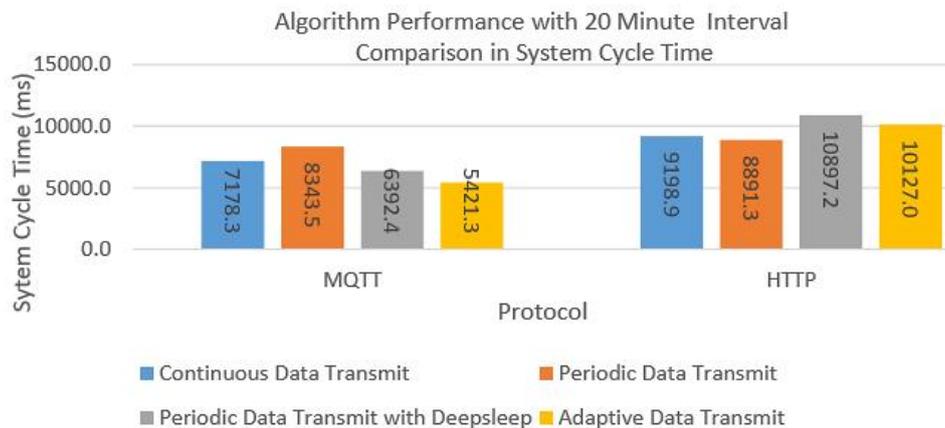
**Figure 5.** Average Current Comparison



**Figure 6.** Battery Life Comparison

Based on Figure 5 and 6, it can be seen that the algorithm with the lowest average current is adaptive data transmit algorithm using MQTT protocol with average current of 4.35 mA. Current consumption can be reduced from other data transmit algorithms due to the data on adaptive algorithm should be larger or smaller than the threshold range to send the data. Data sent to the server when the value of the read sensor is changed significantly. Based on the calculation, application of adaptive algorithm with MQTT protocol at 20 minute intervals is able to make ALDEBARAN in active conditions for more than 22 days.

When compared with the adaptive data transmit algorithm using the HTTP protocol, the average amount of current that is equal to 4.64 mA. The average current of 4.64 mA is resulted in longer battery usage for 21 days. This means that the application of MQTT protocol on adaptive data transmit algorithm at intervals of 20 minutes can make the battery more durable for almost 1 days. The reason is, the implementation adaptive data transmit algorithm is algorithm with the lowest system cycle time in periodic transmission category. System cycle time of each algorithm can be seen in Figure 7.

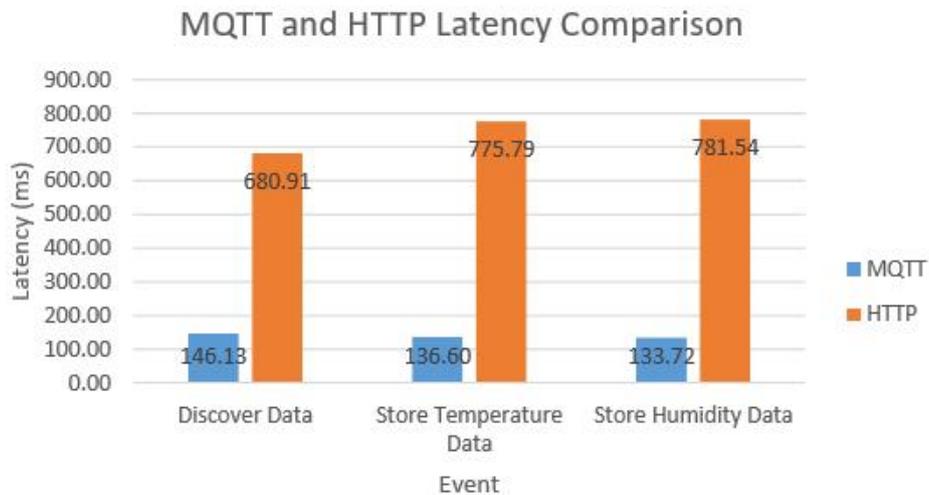


**Figure 7.** Graph of System Cycle Time Comparison on Each Algorithm

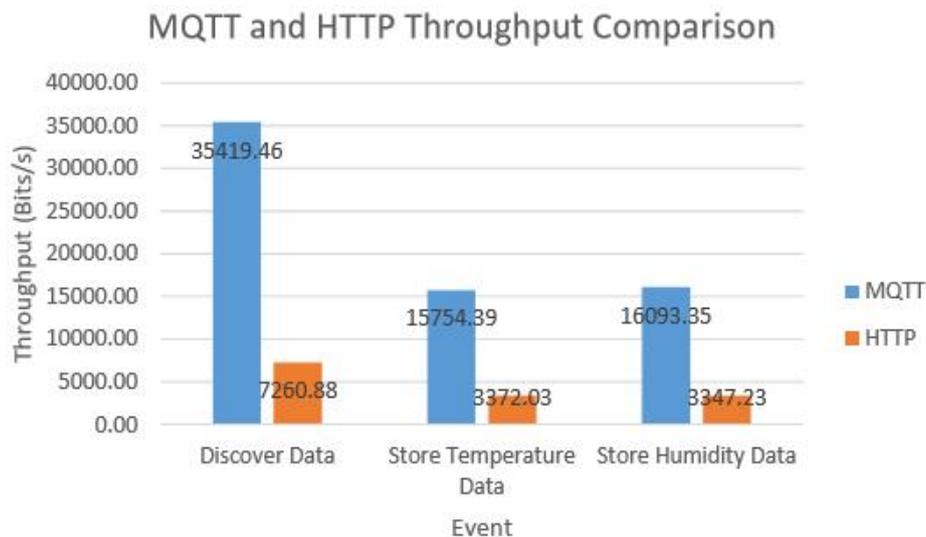
From all the comparison with several parameters, there is a continuous algorithm using the HTTP and MQTT protocol. We can see the comparison of continuous algorithm parameter obtained with a total system cycle time of 7178.3 ms and 9198.9 ms for the MQTT and the HTTP protocols, respectively. Implementation of MQTT protocol can reduce the system cycle time so that when applying to the continuous data transmit algorithm, it will serve better real-time communication.

### 3.2. Comparison of Latency and Throughput on the HTTP and MQTT Protocol

Next, we discuss on latency for each implementation of the algorithm and protocol. Its value are averaged and compared. The high latency data transmission is resulted in a long delay and low latency value indicates that the data transmitted more quickly to the receiver. Latency on MQTT protocol can be lower because MQTT have a low overhead protocol. MQTT is an application layer protocol on OSI based TCP/ IP model that has a fixed header size of 2 bytes for each data packet. Therefore, MQTT is very suitable for low bandwidth networks [8]. Latency to retrieve the 2 bytes data by using MQTT protocol shows 4.66 times lower than the HTTP. Latency to transmit 5 bytes of temperature data using MQTT protocol shows 5.68 times lower than the HTTP. Latency to send the 5 bytes of humidity using MQTT protocol shows 5.84 times lower than the HTTP. This means that by using MQTT protocol, the efficiency and performance of data transmission time can be improved. Latency comparison results can be seen in Figure 8.



**Figure 8.** Comparison of Average Latency on the HTTP and MQTT Protocols



**Figure 9.** Comparison of Throughput on the HTTP and MQTT Protocols

Large throughput is influenced by the amount of data sent and latency. The lower latency is resulted in the higher throughput, while the higher latency is resulted in lower throughput. Based on previous calculations, MQTT protocol has a low latency as a consequence of higher throughput than the HTTP protocol. The MQTT throughput of retrieving data from the server is 4.88 times higher than HTTP, while sending temperature data to the server is 4.67 times higher than HTTP, and when sending humidity data to the server, the result is 4.81 times higher than HTTP. Comparison of throughput on the HTTP and MQTT protocol can be seen in Figure 9.

#### 4. Conclusion

HTTP and MQTT protocols can be applied to ALDEBARAN devices. Based on our experiments, MQTT protocol characteristics have low latency, low overhead, high throughput and low transmission power compared to HTTP. Each application MQTT algorithm becomes longer battery usage. In comparison, the combination of adaptive data transmit algorithm with MQTT protocol is chosen as the most optimal combination used for non-real time communication on ALDEBARAN devices. This is because of lower power usage, bandwidth efficiency, and increased throughput.

Another consideration, for real time communication, we can use a combination of Continuous Data Transmit Algorithm with MQTT protocol as the fastest combination for AC control function because using this method can reduce latency and system cycle time, however it may be wasteful in the use of power when using the battery as a voltage source.

### References

- [1] S. H. Shah and I. Yaqoob, "A survey: Internet of Things (IOT) technologies, applications and challenges," In 2016 IEEE Smart Energy Grid Engineering (SEGE), pp. 381–385.
- [2] B. Dorsemayne, J. P. Gaulier, J. P. Wary, N. Kheir, and P. Urien, "Internet of Things: A Definition and Taxonomy," Proc. - NGMAST 2015 9th Int. Conf. Next Gener. Mob. Appl. Serv. Technol., pp. 72–77, 2016.
- [3] N. De Caro, W. Colitti, and K. Steenhaut, "Comparison of two lightweight protocols for smartphone-based sensing," In: Proceedings of the 2013 IEEE 20th symposium on communications and vehicular technology in the Benelux (SCVT), Namur, 21 November 2013, pp.1–6. New York.
- [4] J. E. Luzuriaga, M. Perez, P. Boronat, J. C. Cano, C. Calafate, and P. Manzoni, "A comparative evaluation of AMQP and MQTT protocols over unstable and mobile networks," 2015 12th Annu. IEEE Consum. Commun. Netw. Conf. CCNC 2015, pp. 931–936, 2015.
- [5] T. Yokotani and Y. Sasaki, "Comparison with HTTP and MQTT on required network resources for IoT," 2016 Int. Conf. Control. Electron. Renew. Energy Commun., pp. 1–6, 2016.
- [6] H. Lin, "Deep Sleep IEEE 802.11 Enhancement for Energy-Harvesting Machine-to-Machine Communications.pdf," pp. 5231–5236, 2012.
- [7] Y. Upadhyay, A. Borole, and D. Dilepan, "MQTT based secured home automation system," 2016 Symp. Colossal Data Anal. Networking, CDAN 2016, 2016.
- [8] Chyi-Ren Dow, Syuan Cheng, Shio-Fen Hwang, "A MQTT-based guide and notification service system," 2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON) , pp. 1-4 , 2016-10. University of British Columbia.