Ardulink 기반 MQTT를 이용한 센서 데이터 저장을위한 IoT 게이트웨이 설계

라이오넬* · 장종욱*

*동의대학교

Design of IoT Gateway for Storing Sensor Data using Ardulink based MQTT

Lionel Nkenyereye* · Jong-Wook Jang*

*Dong-Eui University

E-mail : lionelnk82@gmail.com, jwjang@deu.ac.kr

ABSTRACT

The Internet of things (IoT) needs to be an event-driven approach for efficient real time response and processing. An IoT gateway is sometimes employed to provide the connection and translation between devices and the cloud. Storing data in the local database, and then forwarding it on the cloud is a task to be relegated to a gateway device In this paper, we propose the design of the IoT gateway with Fog computing for storing data from sensors into a local database. In the procedure of designing storing tasks, we propose to use the interfacing software known as Ardulink MQTT bridge. MQTT is a protocol for sensors to publish data to the clients. When it comes to needing historical data, MQTT connector can push MQTT data into SQL database. We write an MQTT client and based on the message topic insert the values into a SQL Database The design of IoT gateway with Fog computing adds value because it provides processing of the data across multiple devices before it sends to the cloud.

Key Word

IoT; Ardulink; MQTT broker; smart gateway; fog computing, local database

Ⅰ. Introduction

In IoTs, literally anything can be part of it, whether it is sensor nodes or dumb objects, so very diverse types of services can be produced. In this regard, resource management, service creation, service management, service discovery, data storage, and power management would require much better infrastructure and sophisticated mechanism. The amount of data IoTs are going to generate would not be possible for standalone power-constrained IoTs to handle. Because unnecessary communication not only burdens the core network, but also the data center in the cloud. For this purpose, data can be preprocessed and trimmed before sending to the cloud. This can be done through a traditional Gateway, accompanied with a Fog Computing. In this paper, we present the need of processing of IoT data, on the basis of traditional Gateway and Fog computing. We have tested this concept on the basis of storing sensor data measured by Esp8266-13 shield boards that are placed around our laboratory. About every 10 minutes, a new measurement is made and this data is published to a Mosquitto MQTT broker that runs on a Raspberry Pi considered. Ardulink framework to connect Java MQTT client to the MQTT server and we will process simple commands to control sensors and publishes data to the MQTT broker considered

as the traditional gateway, then storing in a local database at the fog computing. The gateway based fog computing should be having extra functionality to do a little processing before sending it to the cloud.

II. Gateway with Fog computing based IoT processing

The architecture of IoT is usually considered to be 3-layer, having Perception layer, Network layer, and Application layer, but some [1], [2] add two more layers: Middleware layer and Business layer. Middleware layer receives data from Network layer. Its purpose is service management and storage of data. It also performs information processing and takes decisions automatically based on results. It then passes the output to the next layer, the Application layer [2]. Application layer performs the final presentation of data. In single-hop connectivity, sensor nodes and 'things' are directly connected to the gateway, which then assembles the data and sends to the Fog and then to the cloud. Gateway can further send the data to the Fog and then

to the cloud. Fog Computing refers to bringing networking resources near the underlying networks. It is a network between the underlying network(s) and the cloud(s). Fog Computing extends the traditional Cloud Computing paradigm to the edge of the network, enabling creation of refined and better applications or services.

Since number of connected devices is rapidly increasing, so there is going to be a lot of data as well [3]. Storing that data locally and temporarily will not be possible any more. There is going to be a need of rental storage space. . For this purpose, the gateway device connecting IoT to the cloud should be having extra functionality to do a little processing before sending it to the Internet and eventually to the cloud as shown on the Figure 1. Similarly, temporary storage, preprocessing tasks can be done easily and more efficiently in the presence of a Fog network, co-located with traditional Gateway. Fog computing allows real-time delivery of data, specially for delay sensitive. It can perform preprocessing smart tasks and notify the cloud, before cloud could further adapt that data into enhanced services

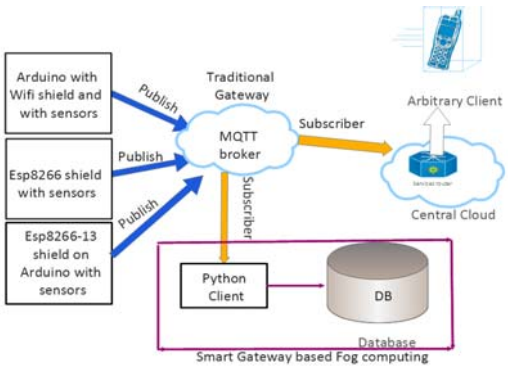

Fig.1. Traditional Gateway with Fog computing for processing of data across multiple devices

III. Design of storing data on the Gateway with Fog Computing

MQTT (MQ Telemetry Protocol) [4]is a lightweight machine-to-machine communication protocol that works on top of TCP/IP and it is very much suited for usage in a n Internet-of-things. The cloud services are nice and easy to use, but if you do not want you data uploaded to a third party storage, you can host an MQTT environment can work as a tradional gateway where data goes to the Internet. In this paperm we have a self-hosted an MQTT environment with the help of a mosquitto broker running on a Raspberry Pi considered as the traditional gateway and let two different Esp8266-13 shield with arduino microcontroller , holding DHT22 sensor publish sensor data through the broker. We will also use Raspberry Pi as a subscriber for the messages and these data will be persisted in a MySQL database. This is considered to act as gateway with fog computing as shown on the Figure 2.

we will use the following topic name to

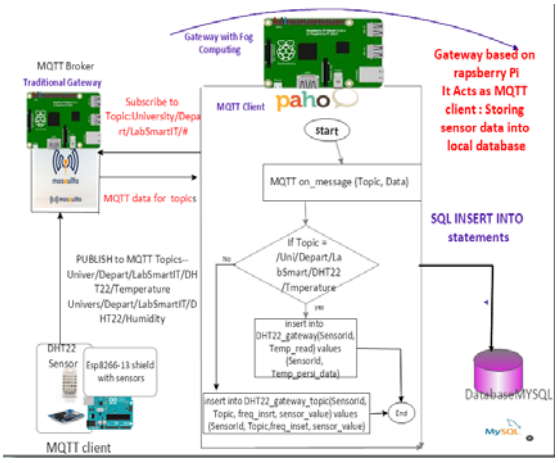

Fig.2. Implementation of Publisher-Subscriber Topic to the MQTT broker for storing sensor data

make the boards publish data retrieved data from the sensors : University/Department/office/sensorname where office must be replaced with a unique name assigned to a office(room), and sensorname must be replaced with a unique name assigned to a sensor. For example, if we assign LabSmartIT as the name for an office(room) and we assign DHT22 as the name of sensor as the name for a humidity sensor data wired to this board, the client that wants to receive data retrieved from this sensor has to subscribe to University/Department/LabSmartIT/DHT22/+ topic because the sensors will provide data in different subtopics.

IV. Performance Evaluation

As explained in the section 3, topics will allow to subscribe to MQTT broker. When we the traditional gateway based on the MQTT server data from any of Subscribed topics, it look into to MQTT topic field, and based on that, it call an appropriate function to save the information into Database Table as shown on the Figure 3. We have implemented this logic to save MQTT Data tables using Ardulink framework based on Java. By leveraging the concept of MQTT wildcard topics to write a simple connecto that can push MQTT data into SQL Database.

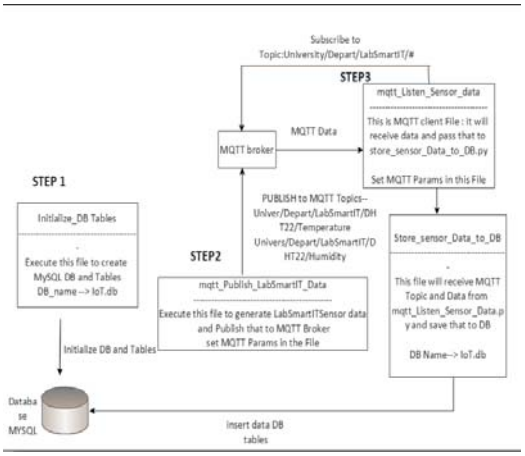

Fig.3. appropriate function to save the information into Database

.

the Figure 4 shows a dashboard of the humidity and temperature data and Figure 5 shows the log trace on the MQTT broker when a new sensor data is published, then stored on the local database

Fig. 4. dashboard of the humidity and temperature data

Fig. 5. log trace on the MQTT broker when a new sensor data is published

V. Conclusions and Future Works

This paper discusses about the expanding IoTs and their integration with a dedicated gateway, for enhanced processing of data across multiples devices before it send to the cloud. We have presented Smart Gateway based communication, along with Fog computing, for the purpose of relegated to the gateway storing data tasks. We have showed how a local MQTT environment where sensor data from ESP23 shield and Arduino boards are stored in a local database (MySQL) on a Raspberry Pi considered as a gateway based fog computing. The extended work could be on the impact of heterogeneous storage and overall performance on the basis of diverse applications.

ACKNOWLEDGMENT

이 논문은 2016년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 지역신산업선도인력양성사업 성과임(No. 한국연구재단에서 부여한 과제번호 : 연구사업 통합시스템에서 확인)(NRF-2016H1D5A1910985).

REFERENCES

[1] Miao Wu et. al., “Research on the architecture of Internet of things”, inthe proceedings of 3rd International Conference on Advanced ComputerTheory and Engineering, 20-22 August, 2012,Beijing, China

[2] Rafiullah Khan, Sarmad Ullah Khan, Rifaqat Zaheer, and Shahid Khan, “Future Internet: The Internet of Things Architecture,PossibleApplications and Key Challenges”, in the proceedings of 10thInternational Conference on Frontiers of Information Technology,Islamabad, Pakistan, 17-19 December, 2012

[3]Dave Evans, “The Internet of Things How the Next Evolution of theInternet Is Changing Everything”, Whitepaper, Cisco Internet BusinessSolutions Group (IBSG), April 2011

[4]Yiming Xu, V. Mahendran, and Sridhar Radhakrishnan, ‘’ Towards SDN-based Fog Computing: MQTT Broker Virtualization for Effective and Reliable Delivery “,COMSNETS 2016 - Workshop on Wild and Crazy Ideas on the interplay between IoT and Big Data, 5-10 Jan. 2016