International Journal of Engineering & Technology, 7 (3.24) (2018) 125-130

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

An MQTT based Real Time LBS System for Vehicles and Pedestrians

In-Hwan Jung[1*], Jae-Moon Lee[2], Kitae Hwang[3]

[1,2,3] *Department of Computer Engineering, Hansung University, Seoul, Korea*
[*] *Corresponding author E-mail:ihjung@hansung.ac.kr*

Abstract

Background/Objectives: We introduce a real time location based IoT system using MQTT protocol that collects location information of moving devices and provides messaging service between based on administrative area unit.
Methods: We implemented an MQTT based a smartphone application for pedestrian location information service and a Raspberrypi IoT device for vehicle information processing and messaging. IoT clients can send messages to the server according to administrative area units by publishing data to an MQTT topic, which is equal to administrative area names. We also implemented an LBS data server and LBS application server for location based data analysis and messaging service.
Findings: The SLIMS (Seoul Location based IoT Messaging System) implemented in this research is able to analyze the real time traffic statistics of pedestrians and vehicles. It also can deliver messages to clients based on coordinate range and administrative area units. SLIMS is implemented for Seoul city but can be applied to any part of the country if map data that distinguish administrative districts is replaced. In addition, it is designed and implemented so that any IoT devices with GPS sensors can be applied even though it is currently implemented for smart phones and automotive devices.
Applications: The SLIMS can be used as a real-time location-based information service for large-scale IoT devices such as real-time pedestrian population and vehicle traffic analysis and location-based message delivery.

Keywords: *MQTT, IoT, LBS, Administrative Area Name, Topic Based*

1. Introduction

Recently, there is an increasing need for a location based IoT system that collects and processes data of mobility devices together with location information in an environment where IoT is widely used. Especially, with the emergence of mobility devices such as IoT for vehicles, location - based IoT application systems are emerging that process existing sensor information and GPS information of IoT devices together. In this location-based IoT system, IoT devices must be able to track their location based on a specific range, such as an administrative area, and be able to deliver messages on a per-administrative basis. As IoT devices become more and more smart, it is important not only to deliver data to the server, but also to receive and process messages from the server. In this paper, we propose a location based IoT messaging system, which can efficiently transmit moving object or data of IoT devices together with real time location information and deliver messages to pedestrians or vehicle drivers located in specific areas such as administrative districts. For example:

"An application user wants to check the current state of pedestrian traffic and the traffic volume of the whole city in Seoul in real time, and also wants to send a message only to the pedestrian or the vehicle driver staying in a specific administrative area such as Seongbuk-GU or Samseon-DONG".

In order to process the above-mentioned request, the location information of pedestrians and vehicles should be collected and stored in a database. Especially, it is necessary to confirm the position of pedestrians and vehicles based on the GPS location information. In this paper, we propose a real-time location based IoT service which can analyze the real-time traffic volume of pedestrians and vehicles using the MQTT(Message Queue Telemetry Transport) environment[1]. In this respect, we designed and implemented a MQTT based LBS(Location Based Service) system, which is called SLIMS(Seoul Location Based IoT Messaging System). We implemented a Raspberrypi[2]client device with a GPS sensor for vehicles and a smartphone application client for pedestrians, and implemented a location-based application system that can store and visualize location information and deliver messages in terms of administrative area units.

The composition of this paper is as follows. Section 2 describes the research related to this paper, and Section 3 describes the design of a real-time location-based messaging system using MQTT topics. Section 4 introduces the implementation and experiment contents. Section 5 describes the conclusion and future research.

2. Related Works

2.1. Iot Data Processing

Various communication protocols such as CoAP(Constrained Application Protocol)[3], Extensible Messaging and Presence Protocol (XMPP)[4] and MQTT(Message Queue Telemetry Transport) have emerged to efficiently process IoT data. The MQTT protocol is a communication protocol for sending and receiving messages in a specific topic by using the *Subscribe* and

Publish method. It provides an efficient communication environment for collecting data of a large number of IoT devices due to a short length of data and simple communication procedure[5][6][7]. Although MQTT is suitable for collecting data for a large number of clients considered in this study, MQTT does not have any function as a default to handle location information when collecting moving client data such as pedestrians or vehicles.

In this paper, therefore, in addition to utilizing the advantage of MQTT which provides an efficient IoT device data processing function, we have added the function to manage the location of clients based on administrative district name that is used as an MQTT topic name.

2.2. MQTT Topic and Administrative District

One of the important features of MQTT is that topic can be organized into a hierarchical structure. In this paper, we apply the hierarchical structure of MQTT Topic to administrative districts. Figure 1 shows the change of Client-A's topic when Client-A moves from "Seoul City / Seongbuk-GU / Jungnung-DONG" to "Seoul City / Seongbuk-GU / Gilum-DONG". If Client-A's initial location is "Seoul City / Seongbuk-GU / Jungnung-DONG", Client-A subscribes topics of "Seoul City", "Seongbuk-GU" and "Jungnung-DONG" respectively. If Client-A moves to "Gilum-DONG", Client-A unsubscribes from "Jungnung-DONG" and subscribes to "Gilum-DONG".

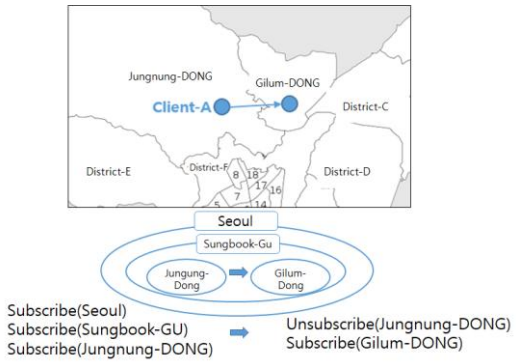


Figure 1: MQTT Topic and District Name

Because the clients are all subscribed to a specific topic which is same as administrative district name, the application will be able to determine which clients are staying in which administrative areas, and to specify a specific administrative area to deliver messages to the clients there. In this paper, we use data that can divide administrative districts in Seoul city. In Seoul, there are total of 25 GU's and 424 DONG's.

3. System Design

3.1. System Architecture

The implemented SLIMS is depicted in Figure 2. The MQTT Broker, the core of the MQTT protocol, a smartphone app for smartphone clients and automotive Raspberrypi clients for vehicles, LBS Data Server for storing location information. In addition, LBS Application Server visualizes information obtained through LBS Data Server and transmitting messaged to clients based on location information. The MQTT broker is designed to use the Moquitto[8][9] open source development environment and acts as a communication relay between clients and servers based on topics.

The smartphone client and the car client periodically transmit the GPS location information to the server through the MQTT broker using the MQTT publish function. The LBS Data Server converts location information and GPS information sent by the client into administrative area and stores it. The LBS Application Server is a location-based application that visualizes the client location information stored in the database in various ways and has the message transfer function in administrative district units.

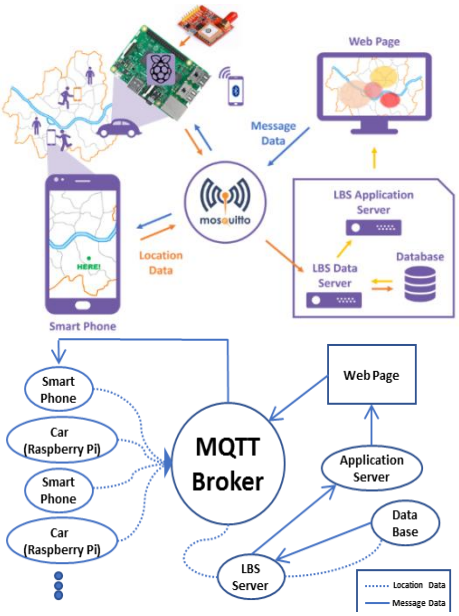


Figure 2: System Architecture and Data Flow of SLIMS

3.2. GPS Information to Administrative District Conversion and Topic Change

As shown in Figure1, in order to join MQTT topic name which is the name of administrative district where mobile clients are located, it is necessary to convert the current GPS information into administrative area name. For devices such as smartphones, itis possible to use the o API, such as Google API, to convert its current location to the administrative area name. For devices that cannot use open APIs, such as automotive IoT devices, LBS servers and application servers are designed to identify administrative areas and deliver them to clients. In case of smartphone client, it is possible to check the administrative area by itself. On the other hand, in this paper, the location information is designed to be managed by the application server. This is because the location information of the clients in location based service application system may not be managed only by the administrative district. That is, the location information set by the application system may be used. For example, the server can use special codes such as "R1 / R11 / R112" instead of "Seoul City / Seongbuk-GU / Jungnung-DONG." In this case, all clients will subscribe using the topic name specified by the LBS server which manages topic names according to business logic.

Figure 3 shows the process in which the client sends the GPS information to the server, receives the new administrative area name, and subscribes the administrative area name as a MQTT topic.

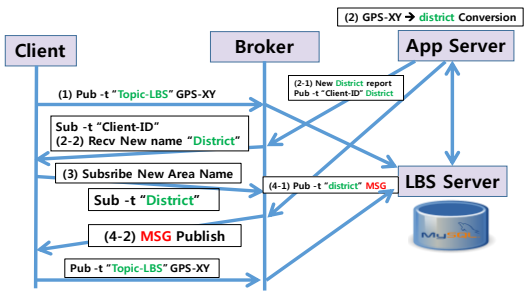


Figure 3: Converting GPS Coordinate to District Name

Step 1: The client periodically publishes the current position (longitude, latitude) obtained through the GPS sensor as a Topic-LBS, and the LBS Server is subscribed to the Topic-LBS and stores the current position of the received clients in the data face.

Step 2: Application Server converts GPS location information into district name, "Seoul City / Seongbuk-GU / Jungnung-DONG", as an example.

Step 2-1: The Application Server publishes the name of the new administrative area ("Seoul City / Seongbuk-GU / Jungnung-DONG") as Topic in Client-ID and informs the client of the new administrative zone name.

Step 2-2: The client subscribes to the Client-ID as topic and receives the new administrative area name ("Seoul City / Seongbuk-GU / Jungnung-DONG").

Step 3: The client subscribes to the new administrative area name ("Seoul City / Seongbuk-GU / Jungnung-DONG") as topic. At this time, the client subscribes to the hierarchical structure of the administrative area as shown in Figure 2. Subscribe(Seoul), Subscribe(Seongbuk-GU) and Subscribe(Jungnung-DONG). If the administrative area has been changed, the previous location is unsubscribed and subscribed to the new location.

Step 4-1: Application Server sends a message to clients in a specific administrative area. Suppose, for example, that it is a "Jungnung-DONG".

Step 4-2: All clients subscribing to the topic "Jungnung-DONG" will be able to receive the message.

3.3. Client Design

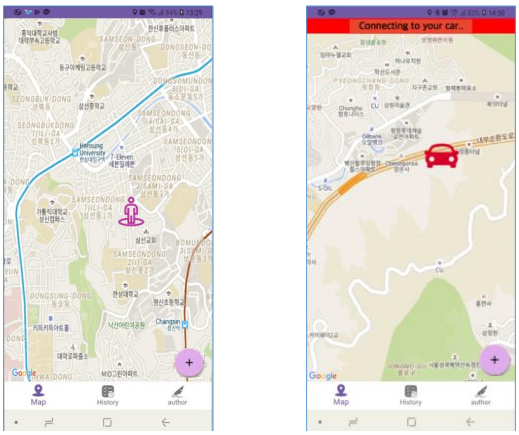
If the client moves and the administrative area changes, a new administrative area name is received from the server. In that case, it unsubscribes from the existing administrative section topic and subscribes to the new administrative name as a topic. Since it is subscribed to the topic, which is same as administrative district name, it can receive messages based on the location of the administrative area sent by the Application Server. Figure 4 shows the processing algorithm when the client location is changed in terms of administrative area.

Both smartphone and vehicle clients use the algorithm in Figure 4. In case of smartphone, it is possible to display map and message on its own but in case of vehicle on-board device, it is designed to display contents on the driver's smartphone using Bluetooth.

```
// Check GPS location
CurGPS = Get_Current_GPS();
// Send GPS information to the server and subscribe to the
new administrative area sent by the server as Topic
Send_Current_GPS_to_Server(CurGPS);
NewMsg = Receive_Msg_from_Server();
NewTopic = NewMsg.NewTopic;
if (NewTopic!=CurTopic) { // area is changed
        Unsubscribe(CurTopic);
        Subscribe(NewTopic);
        CurTopic = NewTopic;
}
```

Figure 4: Client Topic Change Algorithm

Figure 5(a) shows the smartphone client screen and Figure 5(b) shows the smartphone designed screen of the vehicle driver connected to the Raspberrypi for the vehicle. For smartphone clients, the current location is displayed with a human-shaped icon, while the current location is displayed in the form of a car in the app for the driver's application.



(a) Smartphone Client (b) App for Drivers

Figure 5: Client for Vehicles

3.4. Server Design

In SLIMS implemented in this paper, the server consists of LBS server and application server. The LBS server has the following two important functions.

(1) Location information storage: LBS server subscribes Topic_LBS to the MQTT broker and receives and stores location information sent by clients. The format for storing location information of each client is (Client-ID, Time, GPS-XY).

(2) Location information to Administrative area conversion: The LBS server converts the GPS-XY information sent by the client into administrative area name. To convert the GPS location into the administrative area, the Ray Casting Algorithm[10] is used for the administrative district map of the Seoul area.

```
client_GPS = mqtt_subscribe_client(Topic_LBS);
location_Name = getLocationName(client_GPS);
mqtt_publish_client(ClinetID, location_Name);
```

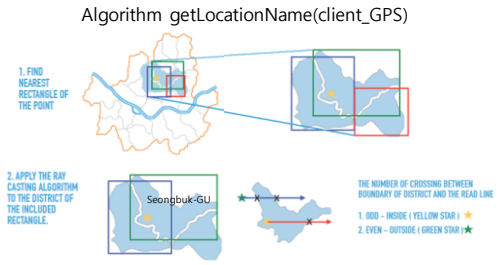


Figure 6: Location Processing Algorithm of LBS Server

As shown in Figure 6, when the LBS server receives the GPS information published by the client, it finds the name of the administrative area using the algorithm *getLocationName* and sends new administrative area name to the client. An example is where the client is in "Seongbuk-GU". The server uses the coordinates of each administrative zone boundary to map the administrative districts and finds the administrative districts using the Even-Odd Algorithm of the Ray Casting algorithm. At first, it looks for administrative regions with the closest orthogonal coordinates that can contain the coordinate of GPS location. At second, it finds the location is inside the "Seongbuk-GU" because the odd number of is obtained by Even-Odd algorithm for adjacent administrative districts. In this way, the server determines and publish the name of the administrative area to the client using the MQTT publish function.

The SLIMS application server uses the client location information stored in the LBS server to visualize information such as real-time floating population and it is designed to have the following

functions so that messages can be transmitted to clients in various ways.

(1) Real-time floating population display: Visualize real-time location information of smartphone users and car clients on the map. It is displayed using icon and thermal map.

(2) Location-based message transmission: Messages are sent to clients based on administrative area, coordinates, etc. Selectable location criteria are as follows.

- Sending Old Sent Messages: When you select a phrase on the map screen and enter a message, the MQTT Topic publishes it with the corresponding phrase name.

(Example: publish -t "Seongbuk-GU" -m "Message to Seongbuk-GU citizens..")

- Sending same unit message: Select the desired message on the map screen and input the message, and publish it in the MQTT Topic with the corresponding name.

(Example: publish -t "Samseon-DONG" -m "Announcement to Samseon-DONG citizens..")

- Administrator-specified square coordinate-based message transmission: If a user specifies a rectangle with the mouse on the map, a message is delivered to all the clients in it.
- Administrator-specified circular coordinate-based message transmission: If a user specifies a circle with the mouse on the map, a message will be sent to all the clients in it.
- KNN (K Nearest Neighbor) transmission: K number of clients is selected at a close distance based on specific coordinates.

(3) Real-time population change display: It periodically shows the current floating population status of the selected administrative area.

(4) Client tracking: It is possible to trace the movement path of clients by displaying the coordinate change of each client stored in the LBS server on the map.

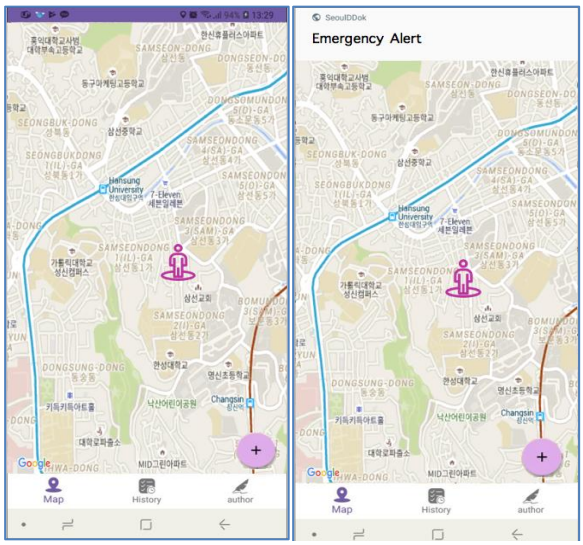
4. Implementation and experiment

In this paper, we implemented a system that can provide location based messaging services for administrative districts in Seoul. We implemented the LBS Data Server and the LBS Application Server to implement the Android app client for pedestrians and the Raspberrypi device for the vehicles.

4.1. Client Implementation

4.1.1. Smartphone Client

Figure 7 shows the screen Android app. As shown in Figure 7(a), pedestrians can see the current location on the map. Figure 7(b) shows an example of a message received from the server to a client in "Seongbuk-GU"



(a) Current Location (b) Message Receiving

Figure 7: Android Client for Pedestrians

4.1.2. Client for Vehicles

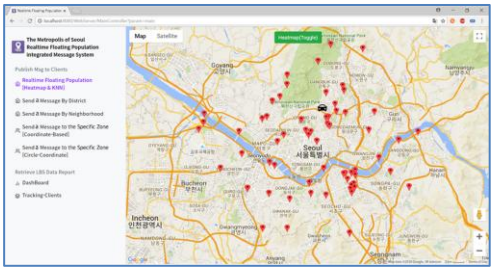
The on-board device based on Raspberrypi has a GPS module for location verification and includes a Bluetooth communication module to communicate with the driver's smartphone application. Communication between the MQTT broker and the Raspberrypi device was implemented using Python [11].

4.2. Lbs Data Server

The LBS data server is implemented using MySql and Java. The LBS server subscribes to the Topic_LBS with MQTT broker and stores the location information received from the client in the form of (*Client-ID*, *Time*, *GPS-XY*, *Administrative Districts*). The Administrative District again takes the form of [*Administrative District 1*] [*Administrative District 2*] ... [*Administrative Region n*]. For example, if the client is located in "Seoul City / Seongbuk-GU / Samseon-DONG", it is like [Seoul City] [Seongbuk-GU] [Samseon-DONG]. By classifying the clients into administrative divisions, the application server can identify the current population of administrative districts and use them to deliver messages to the administrative districts[12].

4.3. Lbs Application Server

SLIMS's LBS Application Server is implemented in Java and JSP, and the main screen is shown in Figure 8. In Figure 8, the main menu is displayed on the left side of the screen, and the location of clients in the Seoul map is displayed on the right side. Figure 8 also shows the real-time flow population in the form of icons and thermal maps.



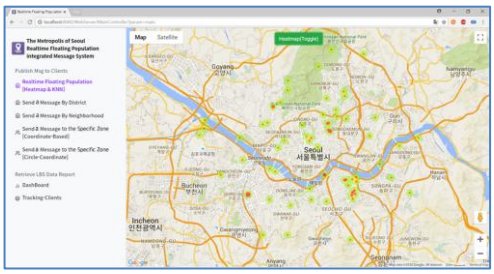


Figure 8: Application Server Screen and Population Thermal Map

4.3.1. Location-Based Message Transmission

Figure 9 shows the GU area based message transmission. When the administrator selects send the GU on the map, the selected area is displayed in red and the message input window is displayed. When a message is entered, the message is sent only to the clients located in that district, GU. Messages sent from the server are displayed at the top of the smartphone screen.

Figure 10 shows an example where the administrator sends a message specifying a rectangular or circular area. There are two possibilities according to the size of the area selected by the administrator.

(1) When the selected area is in the administrative zone: In this case, the clients should not send messages to the administrative zone. This is because messages are sent to clients outside the selected area. In this case, the server publishes the message to each client in a unicast manner, using Client-ID as a topic.

(2) When the selected area is larger than the administrative area: In this case, the clients are distributed in several administrative areas, so the process of finding administrative areas in rectangular or circular coordinates should be preceded. If all the administrative districts are included in Cartesian coordinates, publish the name of the administrative name as a topic. If only part of the administrative district is included in the selected area, publish it to clients in that area in a Unicast manner. If the selection area is larger than the administrative area, the number of message transmission can be reduced by publishing the administrative area as topic as much as possible.

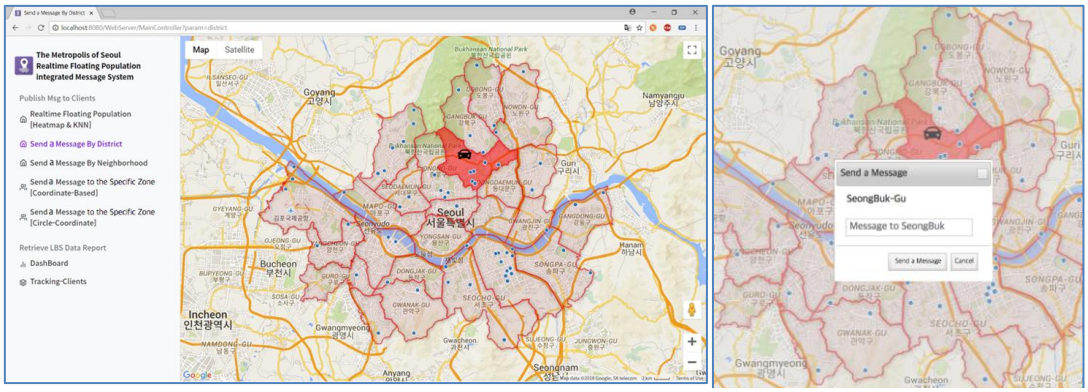


Figure 9: GU District Region Message Sending

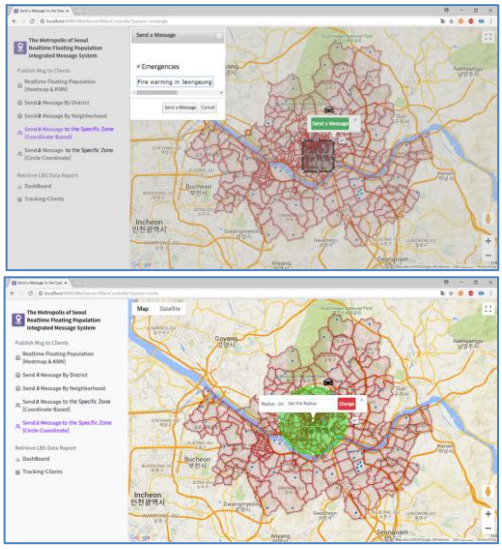


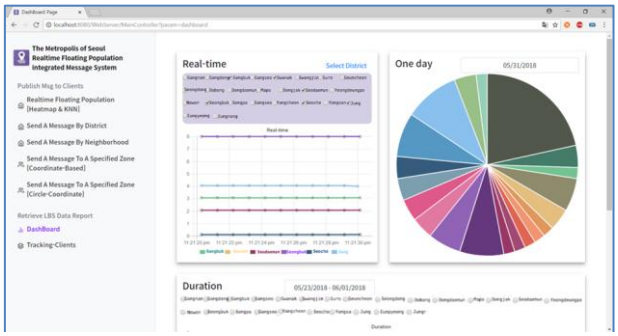
Figure 10: Rectangular and Circular Region Message Sending

4.3.1. Real-Time Flow Population Change And Tracking

In SLIMS, based on the location information of clients, real time monitoring of the floating population is implemented. In addition, a tracking function is implemented so that it can identify the movement path of each client using the stored location information.

Figure 11 shows a real-time graph showing the changes in the floating population of each district. If a user selects a city to monitor, it shows real-time changes in current population including the vehicles in that region. Figure 11 also shows the change in the floating population over a given period. If a user set the area and the period, the graph shows the daily floating population change.

Figure 12 shows a screen that tracks the route of a client. When a user selects a client and a time period, the map shows how clients moved during that time period.



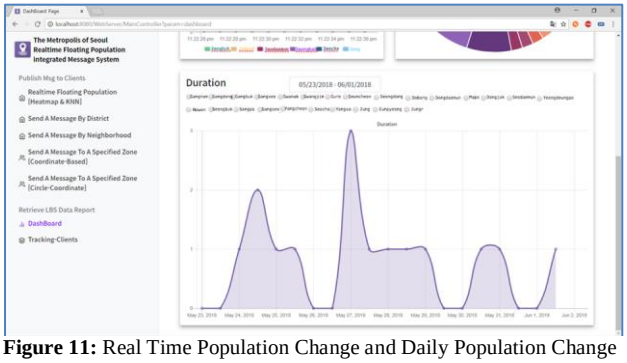


Figure 11: Real Time Population Change and Daily Population Change

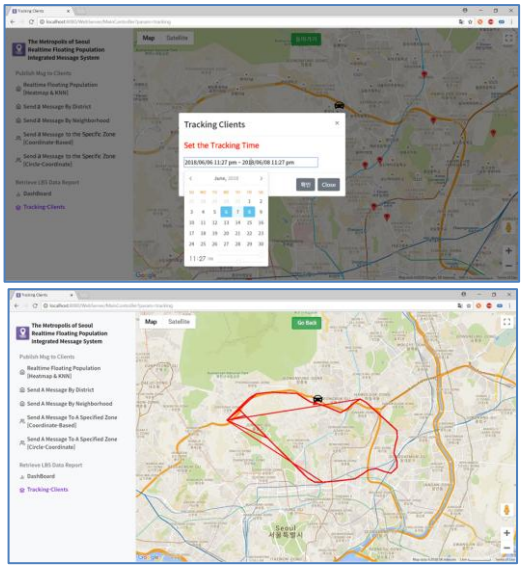


Figure 12: Tracking Screen

5. Conclusion

In this paper, we design and implemented a real time IoT messaging system that can collect location information of moving vehicles and pedestrians in real time using MQTT protocol and provide location based information service in terms of administrative area unit. In order to apply the topic subscribe and publish structure, which is the merit of MQTT protocol, to the location information service, the conversion function is implemented in the LBS server so that the GPS information can be converted into the administrative area name and used as the MQTT topic. Clients can access the administrative area designated by the server as a topic and send or receive messages to the administrative area unit. In this paper, in order to verify administrative area unit based messaging service, we designed and implemented SLIMS (Seoul Location based IoT Messaging System). In SLIMS, it is possible to analyze real time traffic volume of pedestrians and vehicles by tracking the location of clients based on the city and county, which is the administrative district unit of Seoul Metropolitan Government.

SLIMS is implemented for Seoul city but can be applied to any part of the country if map data that distinguish administrative districts is replaced. In this paper, we have designed and implemented clients so that any type of IoT devices can be integrated if it has GPS sensors and basic communication functions. That is, even though the IoT devices are only able to execute the GPS sensor and MQTT basic protocol, it can be integrated because the area codes are managed by application system.

As a conclusion, the location based IoT messaging system implemented in this paper can be used as real time location based information providing service for real time large scale IoT devices.

In future work, we will implement and test a location-based messaging system for various types of IoT devices that can communicate with GPS sensor and MQTT. In the case of MQTT broker, there are many number of topics required when applying administrative area as a topic name proposed in this paper. Therefore, in order for improving the performance, it is necessary to analyze the overhead that occurs when managing a large number of topics. In case of LBS server, it is expected to study high-performance storage structure such as main memory database for real-time location information tracking to improve the structure and performance, and add the ability to search clients in various coordinates besides Cartesian coordinates.

Acknowledgment

This research was financially supported by Hansung University.

References

[1] https://en.wikipedia.org/wiki/MQTT
[2] https://www.raspberrypi.org/
[3] https://en.wikipedia.org/wiki/CoAP
[4] https://en.wikipedia.org/wiki/XMPP
[5] Lampkin V, et al. Building smarter planet solutions with MQTT and IBM WebSphere MQ telemetry IBM. ITSO
[6] Hermes Aslava, Luis Alejandro Rojas and Ramon Pereira. Implementation of Machine-to-Machine Solutions Using MQTT Protocol in Internet of Things (IoT) Environment to Improve Automation Process for Electrical Distribution Substations in Colombia. Journal of Power and Energy Engineering, pp. 92-96, 2015. DOI:https://doi.org/10.4236/jpee.2015.34014
[7] Kitae Hwang, Heyjin Park, Jisu Kim, Taeyun Lee, Inhwan Jung. An Implementation of Smart Gardening using Raspberry_pi and MQTT. The Journal of the Institute of Internet, Broadcasting and Communication, Vol. 18, No. 1, pp.151-159, Feb. 2018. DOI:https://doi.org/10.7236/JIIBC.2018.18.1.151
[8] https://mosquitto.org/
[9] R. A. Light. Mosquitto: server and client implementation of the MQTT protocol. The Journal of Open Source Software. vol. 2, no. 13, May 2017
[10] Roth, Scott D. (February 1982). Ray Casting for Modeling Solids. Computer Graphics and Image Processing. 18 (2): 109-144. https://doi.org/10.1016/0146-664X(82)90169-1
[11] https://en.wikipedia.org/wiki/Python
[12] Seoul Mobile Platform. Seoul Metropolitan Government Public Data. https://data.seoul.go.kr/, 2018-07-17