ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC BÁCH KHOA

----------o0o----------



Tp.HCM, Tháng 12/2017

# 

# MỤC LỤC

MỤC LỤC 1

PHẦN 1 . SƠ ĐỒ MÔ HÌNH TỔNG QUÁT HỆ THỐNG. 2

PHẦN 2 . GIAO TIẾP GỬI DỮ LIỆU TỪ CÁC CLIENT LÊN GATEWAY. 4

2.1 Giao diện 4

2.1.1 Code 4

PHẦN 3 . GIAO TIẾP GỬI DỮ LIỆU TỪ GATEWAY LÊN “”BROKER DÙNG GIAO THỨC MQTT. 6

3.1 Cài đặt “broker” MQTT. 6

3.2 Chương trình chuyển dữ liệu từ gateway lên “broker”. 6

PHẦN 4 . GIAO TIẾP PHÁT DỮ LIỆU TỪ “”BROKER” CHO CÁC “SUBSCRIBER”. 9

4.1 “Subscriber” bằng ngôn ngữ java. 9

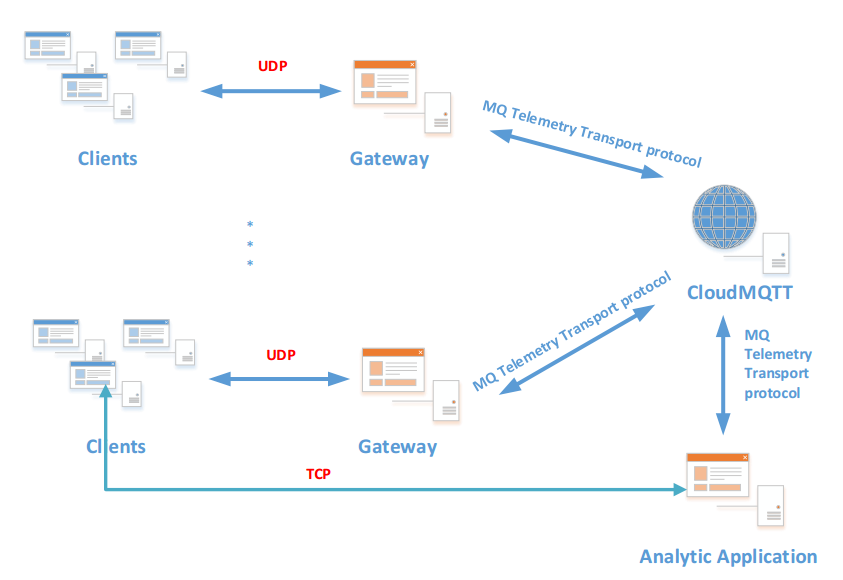
4.2 “Subscriber” bằng phần mềm MQTTFx v1.4.2. 9

PHẦN 5 . DEMO & KẾT QUẢ. 11

5.1 Demo & kết quả đạt được 11

5.2 Kết luận: 11

# . SƠ ĐỒ MÔ HÌNH TỔNG QUÁT HỆ THỐNG.



* **Đề bài:** Hệ thống có thể được chia thành bốn phần chính, bao gồm:
* Client: Ứng dụng thu thập dữ liệu cảm biến trên thiết bị di động Android theo nhu cầu ứng dụng. Dữ liệu cảm biến thu thập được sẽ chuyển về một Gateway thông qua kết nối UDP.
* Gateway: Điểm thu thập dữ liệu. Sẽ có một hoặc nhiều Gateway. Mỗi gateway chịu trách nhiệm quản lý một nhóm gồm một hoặc nhiều client. Dữ liệu từ các client sẽ được gửi trực tiếp tới gateway quản lý nó thông qua giao tiếp UDP.
* CloudMQTT: Trung tâm lưu trữ, mọi dữ liệu cảm biến sau khi được thu thập tại các Gateway sẽ được Gateway chuyển về đây. Module này được tạo bằng cách đăng ký dịch vụ trên cloudmqtt.com. Dữ liệu từ gateway được chuyển về CloudMQTT qua MQ Telemetry Transport Protocol (MQTT).
* Analytic Application: một ứng dụng phân tích dữ liệu có nhiệm vụ đọc dữ liệu cảm biến từ các client gửi lên được lưu trong CloudMQTT. Ứng dụng sẽ phân tích dữ liệu và đưa ra các thông tin cần thiết cho người sử dụng. Phần phân tích dữ liệu sẽ được thực hiện trên một server. Kết quả phân tích sẽ được chuyển xuống client (ứng dụng trên di động) qua protocol do nhóm tự định nghĩa (sử dụng giao thực TCP ở tầng transport) và hiển thị lên ứng dụng trên di động.
* Các “client” gửi dữ liệu gps của “smart phone” lên cho GateWay qua port 5000.
* GateWay được viết bằng ngôn ngữ Java nhận dữ liệu từ các client và lưu thành file.
* GateWay sẽ gửi các dữ liệu này lên cho “broker” qua giao thức MQTT qua port chuẩn 1883.
* “Broker” được cài trên một máy tính. Dùng chương trình “broker” miễn phí EMQTT từ trang web: <http://emqtt.io/downloads>.
* Các “Subscriber” được viết bằng Java và nhận dữ liệu từ “broker” qua giao thức MQTT qua port chuẩn 1883. Hoặc có thể dùng các phần mềm “Subscriber” miễn phí MQTTFx v1.4.2 từ trang web: <http://mqttfx.jensd.de/>.
* Nhóm chưa làm được phần phân tích dữ liệu “Analytic Application” theo hình trên.

# . GIAO TIẾP GỬI DỮ LIỆU TỪ CÁC CLIENT LÊN GATEWAY.

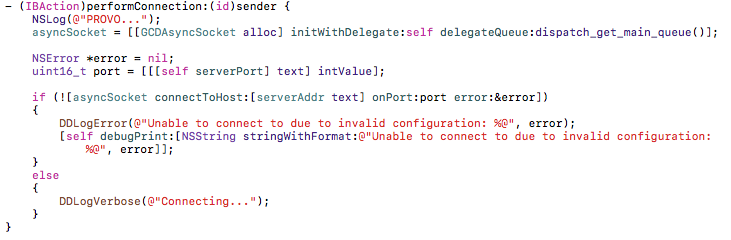
* Client thực hiện kết nối với gateway theo giao thức TCP/IP dựa trên thư viện GCDAsyncSocket được hỗ trợ trong lập trình IOS.
* Chương trình được viết trên ngôn ngữ Objective C, Thư viện được download theo link sau: <https://github.com/robbiehanson/CocoaAsyncSocket>.
* File *“ICViewController.m”* trình bày giao diện chính và thực hiện các chức năng gửi lệnh cũng như nhận kết quả từ gateway.

## Giao diện

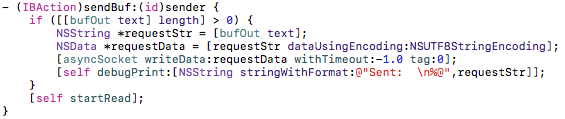


### Code

* **Connect:**



* **Send message:**



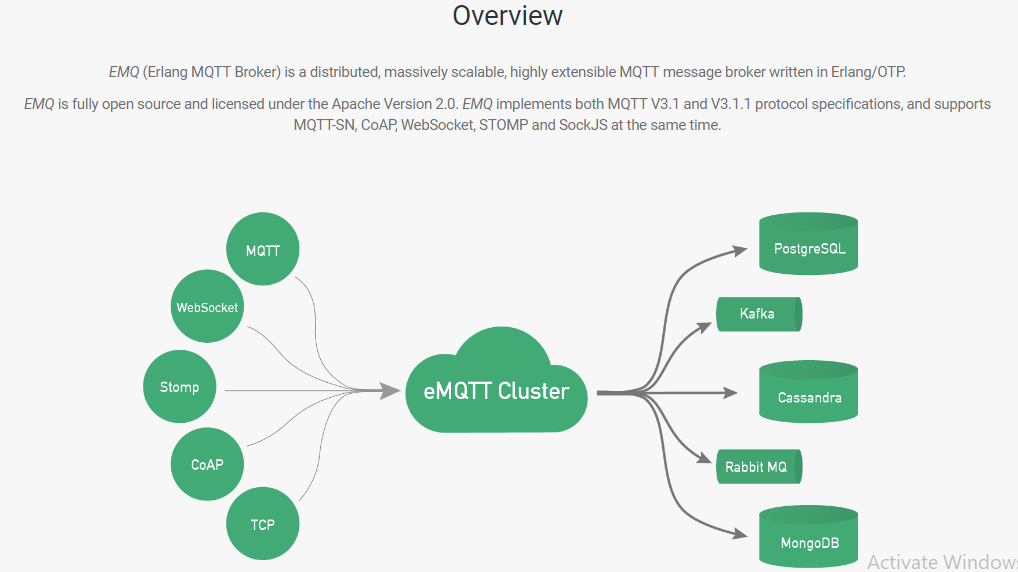
* **Nhận tin từ Gateway:**



# . GIAO TIẾP GỬI DỮ LIỆU TỪ GATEWAY LÊN “BROKER” DÙNG GIAO THỨC MQTT.

## Cài đặt “broker” MQTT.

* “Broker” được cài đặt trên window 10 với phần mềm EMQTT, hiện tại là miễn phí theo link sau: “http://emqtt.io/downloads”.
* “Broker” mở port mặc định chuẩn cho giao tiếp MQTT là 1883.



## Chương trình chuyển dữ liệu từ gateway lên “broker”.

* Gateway thực hiện kết nối với “broker” theo giao thức MQTT trên port chuẩn 1883.
* Chương trình viết trên ngôn ngữ java, sử dụng thư viện theo link sau: “https://www.eclipse.org/paho/clients/java/”.
* File “*MqttApp.java*” gọi chương trình chính. Với tham số là “Publisher” thì chương trình sẽ đọc từ file dữ liệu gps đã nhận được từ các “smart phone” và gửi dữ liệu cho “broker” với “topic” là “iot\_gps” . Với tham số là “Subscriber” thì chương trình đóng vai trò là “Subscriber” sẽ đăng ký với “broker” để nhận dữ liệu có “topic” là “iot\_gps”.

*package com.mapr.demo.mqtt;*

*import com.mapr.demo.mqtt.simple.Publisher;*

*import com.mapr.demo.mqtt.simple.Subscriber;*

*import org.eclipse.paho.client.mqttv3.MqttException;*

*/\*\**

*\* Basic launcher for Publisher and Subscriber*

*\*/*

*public class MqttApp {*

*public static void main(String[] args) throws MqttException {*

*if (args.length < 1) {*

*throw new IllegalArgumentException("Must have either 'publisher' or 'subscriber' as argument");*

*}*

*switch (args[0]) {*

*case "publisher":*

*Publisher.b(args);*

*break;*

*case "subscriber":*

*Subscriber.a(args);*

*break;*

*default:*

*throw new IllegalArgumentException("Don't know how to do " + args[0]);*

*}*

*}*

*}*

* File “*Publisher.java*” thực hiện gửi dữ liệu lên “broker” theo giao thức MQTT port 1883 với “topic” là “iot\_gps”.

*package com.mapr.demo.mqtt.simple;*

*import java.io.BufferedReader;*

*import java.io.File;*

*import java.io.FileInputStream;*

*import java.io.FileNotFoundException;*

*import java.io.FileReader;*

*import java.io.IOException;*

*import java.io.InputStreamReader;*

*import java.util.logging.Level;*

*import java.util.logging.Logger;*

*import org.eclipse.paho.client.mqttv3.MqttClient;*

*import org.eclipse.paho.client.mqttv3.MqttException;*

*import org.eclipse.paho.client.mqttv3.MqttMessage;*

*import org.eclipse.paho.client.mqttv3.persist.MemoryPersistence;*

*public class Publisher {*

*public static void b(String[] args) throws MqttException {*

*String messageString = "Hello World from Java!";*

*FileInputStream fin;*

*if (args.length == 2 ) {*

*messageString = args[1];*

*}*

*MemoryPersistence persistence = new MemoryPersistence();*

*System.out.println("== START PUBLISHER TOPIC IOT\_GPS ==");*

*MqttClient client = new MqttClient("tcp://172.28.27.230:1883", MqttClient.generateClientId(), persistence);*

*client.connect();*

*System.out.println("Type string GPS to send to TOPIC IOT\_GPS.");*

*MqttMessage message = new MqttMessage();*

*File file = new File("D:\\Master\_UoT\_Khoa2017\_Dot1\\Subject\_ComputerNetwork\_4TC\\data\\test.txt");*

*FileReader fileReader;*

*try {*

*fileReader = new FileReader(file);*

*BufferedReader bufferedReader = new BufferedReader(fileReader);*

*while ( messageString != null) {*

*try {*

*messageString = bufferedReader.readLine();*

*} catch (IOException ex) {*

*Logger.getLogger(Publisher.class.getName()).log(Level.SEVERE, null, ex);*

*}*

*message.setPayload(messageString.getBytes());*

*client.publish("iot\_gps", message);*

*System.out.println("\tMessage '"+ messageString +"' to 'iot\_gps'");*

*}*

*} catch (FileNotFoundException ex) {*

*Logger.getLogger(Publisher.class.getName()).log(Level.SEVERE, null, ex);*

*}*

*client.disconnect();*

*System.out.println("== END PUBLISHER ==");*

*}*

*}*

# . GIAO TIẾP PHÁT DỮ LIỆU TỪ “”BROKER” CHO CÁC “SUBSCRIBER”.

* Nhóm thử test “subscriber” theo 2 cách thì đều nhận được cùng 1 dữ liệu theo “topic” là “iot\_gps”

## “Subscriber” bằng ngôn ngữ java.

* File “Subscriber.java” để mở kết nối “broker”.

*package com.mapr.demo.mqtt.simple;*

*import org.eclipse.paho.client.mqttv3.MqttClient;*

*import org.eclipse.paho.client.mqttv3.MqttException;*

*import org.eclipse.paho.client.mqttv3.persist.MemoryPersistence;*

*public class Subscriber {*

*public static void a(String[] args) throws MqttException {*

*System.out.println("== START SUBSCRIBER TOPIC IOT\_GPS ==");*

*MemoryPersistence persistence = new MemoryPersistence();*

*MqttClient client=new MqttClient("tcp://172.28.27.230:1883", MqttClient.generateClientId(), persistence);*

*client.setCallback( new SimpleMqttCallBack() );*

*client.connect();*

*client.subscribe("iot\_gps");*

*}*

*}*

* File “SimpleMqttCallBack.java” để callback nhận dữ liệu.

*package com.mapr.demo.mqtt.simple;*

*import org.eclipse.paho.client.mqttv3.IMqttDeliveryToken;*

*import org.eclipse.paho.client.mqttv3.MqttCallback;*

*import org.eclipse.paho.client.mqttv3.MqttMessage;*

*public class SimpleMqttCallBack implements MqttCallback {*

*public void connectionLost(Throwable throwable) {*

*System.out.println("Connection to MQTT broker lost!");*

*}*

*public void messageArrived(String s, MqttMessage mqttMessage) throws Exception {*

*System.out.println("Message received:\t"+ new String(mqttMessage.getPayload()) );*

*}*

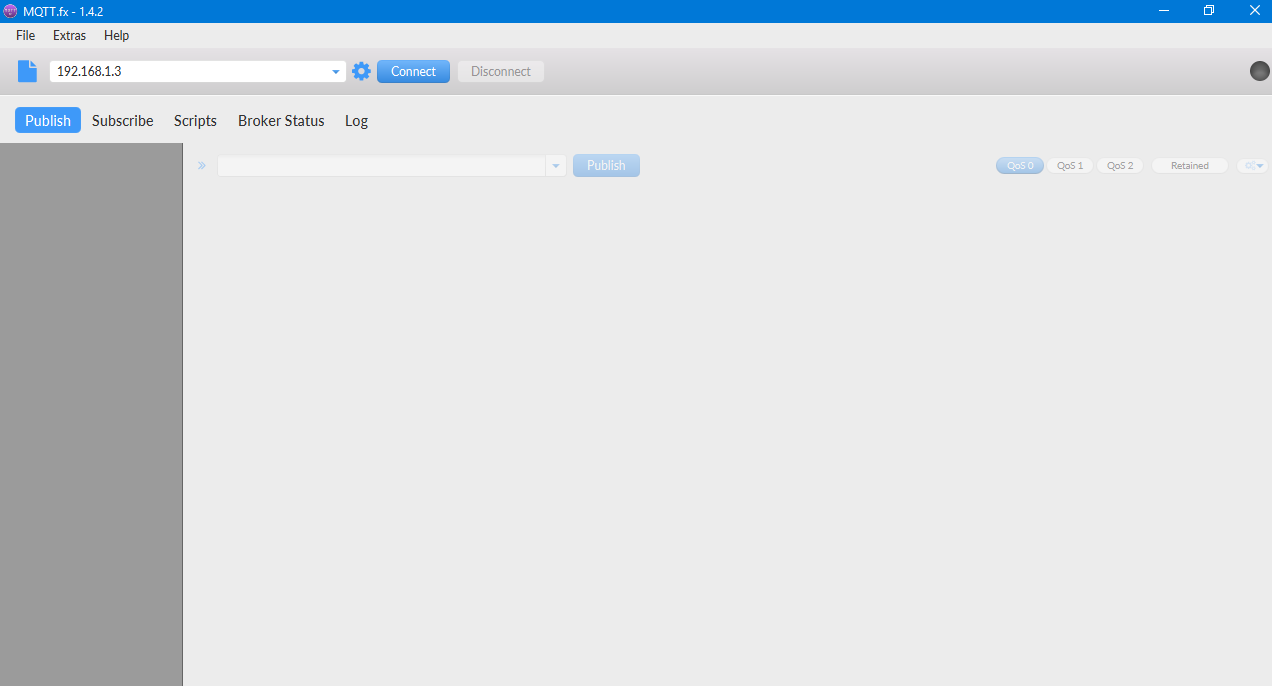
*public void deliveryComplete(IMqttDeliveryToken iMqttDeliveryToken) {*

*}*

*}*

## “Subscriber” bằng phần mềm MQTTFx v1.4.2.

* Tải phần mềm miễn phí từ địa chỉ: <http://mqttfx.jensd.de/>.
* Giao diện chính của chương trình:

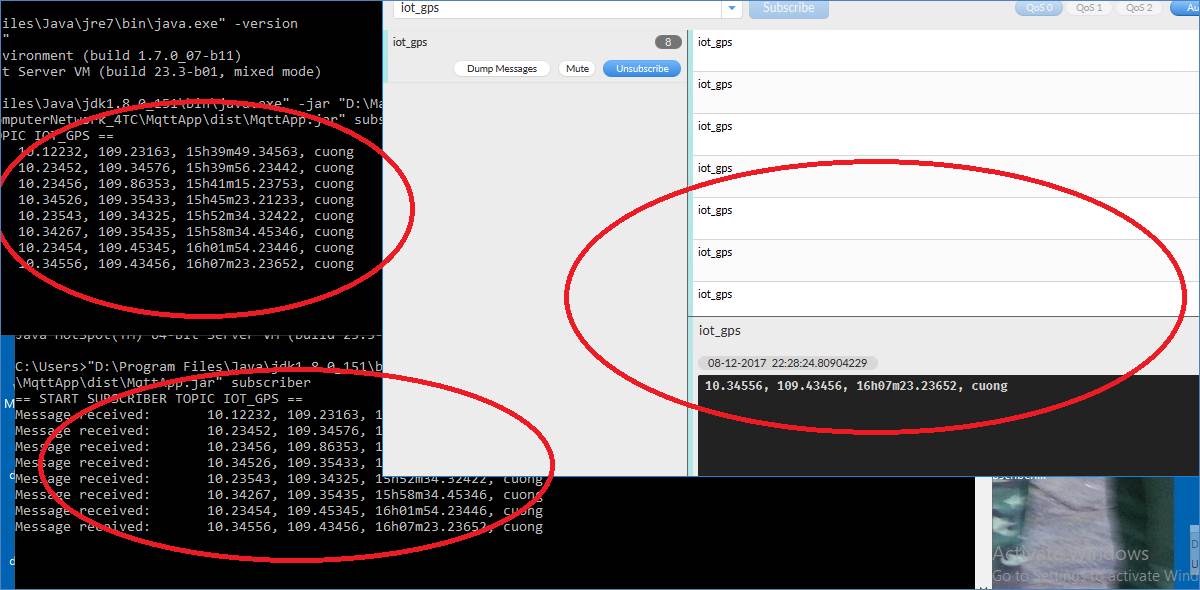


* Soạn thảo server “broker” và nhấn connect để kết nối đến “broker”. (Chương trình demo nên nhóm không đặt password cho “Subscriber”)
* Bấm vào tab “Subscriber” và điền vào “topic” là “iot\_gps”.
* Dữ liệu nhận được dữ liệu khi đã được “publisher” ngay cửa sổ bên dưới.

# . DEMO & KẾT QUẢ.

## Demo & kết quả đạt được

* Nhóm đã thực hiện test trên mạng LAN thì đã thu thập dữ liệu tốt.
* Khi “publisher” gửi lên “broker” các dữ liệu thu thập được từ các “client” với “topic” là “iot\_gps” thì các “subscriber” mà đã đăng ký “topic” là “iot\_topic” đều nhận được dữ liệu.



## Kết luận:

* Đạt được: Hiểu và chạy được giao thức MQTT giữa Gateway và Server Broker.
* Chưa đạt được: Lập trình trên các thiết bị di động là rất rộng nên nhóm chỉ thử nghiệm giao tiếp đơn giản giữa 1 thiết bị Ios và Gateway. Thiếu phần phân tích dữ liệu.