Research and Implementation of Internet of Things Communication System Based on MQTT Protocol

Zueling Niu\textsuperscript{1,*}

\textsuperscript{1}XI’AN TRAFFIC ENGINEERING INSTITUTE, 710300, China

*Corresponding author e-mail: niuzueling@xjgyedu.cn

Abstract. With the advent of the information age, many communication systems and chat systems have appeared on the market. At present, every system has the function of subscribing to news, and MQTT is the main protocol to realize this function. It mainly uses the IBM Bluemix IoT service to build mobile applications and realize device interconnection communication. This paper mainly studies the design and implementation of the Internet of Things communication system, which is based on the MQTT protocol.

Keywords: IoT, MQTT, IBM Bluemix, Gateway, Wireless Terminal

1. Introduction

In the past two decades, information and communication technology (ICT) has changed the way of life. The Internet of Things (IoT), which includes knowledge in many different fields, will be the next milestone in ICT. The communication between traditional devices will improve the quality of life. How to connect these devices effectively is the challenge we are facing [1].

2. Research on MQTT protocol

2.1. MQTT structure

As shown in Figure 1, MQTT is divided into two parts: MQTT client and MQTT message broker. The MQTT client and MQTT message broker are connected through the MQTT protocol.

\begin{figure}
\centering
\includegraphics[width=\textwidth]{mqtt_structureDiagram.png}
\caption{MQTT structure diagram}
\end{figure}

2.2. The message format of the MQTT protocol

The MQTT message body is mainly composed of three parts: a fixed header, a variable header, and a payload. The fixed header is a part that must be included in each command message, and the fixed...
header is fixed at 2 bytes. The fixed header format is shown in Table 1. Among them, Message Type: There are 14 types of messages. QoS level: There are 3 types of service quality levels, QoS 0, QoS 1, QoS 2, and the higher the level, the more system overhead is required, and the greater the impact on communication efficiency. Remaining Length: Represents the length of the message except the fixed header, including variable headers and payload parts. The maximum length can be extended to 4 bytes, and the maximum length can be 256MB [2].

| bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----|---|---|---|---|---|---|---|---|
| byte1 | Message Type | DUP flag | QoS level | RETAIN |
| Byte2 | Remaining Length |

### 2.3. Communication process of MQTT protocol

The MQTT communication protocol that provides network communication based on TCP/IP, like the HTTP protocol, belongs to the application layer communication protocol. The communication protocol uses the publish/subscribe message model. The so-called publish/subscribe message model means that the publisher and subscriber do not directly exchange data through the MQD protocol, but indirectly complete the message interaction through the MQQT protocol and the message broker. The MQTT protocol supports the quality of publishing services based on the Qos identification definition.

The publisher publishes the message process: The client publishes a message to the server, and its publishing and confirmation process is related to Qos [3].

- When Qos=0, it means the unreliable delivery of the message. When the server receives the PUBLISH message from the publisher, it sends the PUBLISH message to all subscribers who subscribe to this topic. In these two processes, the publishing end does its best to deliver the message to the receiving end, and there is no need to return confirmation messages to be sent continuously, which is more efficient.

- When QoS=1, it means reliable delivery. When the server receives the PUBLISH message from the publisher, it needs to return the confirmation message PUBACK to the publisher, and then the PUBLISH message can be sent to all subscribers who subscribe to this topic. The receiving end message is possible. There will be duplication, and the efficiency is low.

- When Qos=2, it means that the server needs to persistently record the PUBLISH message received from the publisher, and temporarily not send it to all interested subscribers. When the server receives the PUBREL message from the publisher, it can send the PUBLISH message to all subscribers who subscribe to this topic, then send PUBCOMP message to the publisher [4].

2) Subscriber's message subscription process: Subscription process: First, the client establishes a TCP connection with the server and sends a CONNECT message to the server; then, after receiving the CONNACK confirmation message authorized by the server, the client sends a SUBSCRIBE message, specifying the topic list of interest (one or Multiple topics), that is, send a subscription request; finally, the server will return a SUB ACK confirmation message to the message that sends the SUBSCRIBE, indicating that the subscription is successful.

### 3. IBM Bluemix IoT service

#### 3.1. Bluemix

Bluemix is built on Cloud Foundry's open source technology and enables developers to more easily develop applications through Platform as a Service (PaaS). In addition, Bluemix also provides pre-built mobile back-end as a service (MBaaS) capabilities. The goal is to achieve large-scale Internet development by providing various services that can be used immediately at any time and hosting functions, thereby simplifying the delivery process of applications. Bluemix also has a cloud deployment that can meet demand. Whether it is a small business planning to expand or a large business that needs more isolation, borderless development can be carried out in the cloud, that is,
dedicated services can be connected to public Bluemix services provided by IBM and third-party vendors. All service instances are managed by IBM, and you only need to pay for the content you choose to use. Through a series of services and runtimes in Bluemix, developers not only gain control and flexibility, but also access various data options from predictive analysis to big data [5,6].

3.2. IoT Service
Bluemix provides predefined services for mobile applications. Bluemix facilitates the implementation, hosting, and expansion of these mobile services for mobile applications. IoT service is a cloud extension hosted by Bluemix. Bluemix hosts and manages middleware services for mobile applications. Application developers can specify the middleware services they need. Then, Bluemix will automatically provision a new instance of the specified middleware service and bind the service instance to the application.

4. Design of IoT communication system based on MQTT protocol
The system hardware is mainly equipped with mobile terminals of Android system, Freescale TWR-LS1021A development board, and KL25Z series single-chip microcomputer with three-axis acceleration sensor [7].

4.1. Cloud deployment
Bluemix IoT distributes messages to connected clients (devices and applications). The equipment includes various machines, respectively publishing the information they detect to the MQTT agent, and the application subscribes to the MQTT agent to obtain the push information. Register the application with the Bluemix IoT service and get the API Key. Register the device and get the device information.

4.2. Local gateway design (gateway)
This system uses mosquitto as the MQTT broker (MQTT Broker), mosquitto is an open source MQTT broker server, supports C, C++, Python language development. Port mosquitto to TWR-LS1021A through cross-compilation, and then register the gateway device with the Bluemix IoT service.

The client (K125Z) on the TWR-LS1021A implements a periodic data release. By simulating the mosquitto source code, the classes and functions that provide the MQTT client are transplanted to the TWR-LS1021A, including connection, release, and subscription. First, initialize the Ethernet, TCP/IP, and MQTT protocols in order to establish a connection, and then periodically publish messages to the proxy server [8]. This communication system supports a Qos level of 0, and can set the maximum message length and heartbeat interval time.

4.3. Device side
K125Z sends sensor data to TWR-LS1021A (gateway) through WiFi, and then releases it to MQTT agent (Bluemix) through the gateway. The gateway establishes a TCP connection to the MQTT server for the device, and uses the MQTT protocol to publish messages to the three-axis acceleration topic of the MQTT agent.

4.4. Mobile client
An Android-based mobile application created based on the Bluemix IoT service, enter the authentication information of the registered device, subscribe to the three-axis acceleration theme, and get the push message of Bluemix. The specific situation is shown in Figure 2 [9].
4.5. Test results
System test environment: 1 Android smart phone, mobile network and WiFi network environment. The smartphone installs an application created based on the Bluemix IoT service. Open the application, enter the authentication information of the registered device in the login interface, and get the push information of Bluemix. The dashboard displays the data of the three-axis acceleration theme published by KL25Z to Bluemix. The test results are shown in Figure 3 [10].

Figure 2. MQTT communication diagram based on Bluemix IoT service

Figure 3. Data display diagram
5. Conclusion
The MQTT protocol makes the IoT system more flexible and reliable. The server and database provide a strong technical support and data storage foundation for the Internet of Things system. I believe that with the development of technology, more concise subscription agreements will be developed in the future, this will bring greater convenience to the IoT system.

References
[1] Heng Ren. Message push server based on MQTT protocol. Computer System Applications, 2014, 23(3): 77-82.
[2] Yao Dan, Xie Xuesong, Yang Jianjun, Han Fei, Wang Shuai. Research and Implementation of Internet of Things Communication System Based on MQTT Protocol [J]. Information Communication, 2016: 39-41.
[3] XU Kan, DING Qiang. A Communication Gateway for Internet of Things Based on MQTT Protocol [J]. Instrument Technology, 2019: 5-8+47.
[4] LIANG Jie. Design of AMI Communication Support Platform Based on MQTT Protocol Family [J]. Computer Times, 2012: 29-32.
[5] FANG Xia. Agricultural Internet of Things Message Pushing System Based on MQTT Protocol [J]. Computer Technology and Development, 2018: 168-171.
[6] Wang Rui, Wu Yujie, Li Yanping. Design of Internet of Things Practical Training Cloud Platform Based on MQTT Protocol [J]. Industrial Control Computer, 2018: 104-106.
[7] Nanjing Gendu Information Technology Co. Ltd. A message push system based on MQTT protocol [P].
[8] Lin Hu, Zhang Jiaming, Yang Haibo. Design and Implementation of Instant Messaging Service Based on MQTT Protocol [J]. Computer System Applications, 2017.
[9] Yao Dan, Xie Xuesong, Yang Jianjun, Han Fei, Wang Shuai. Research and Implementation of Internet of Things Communication System Based on MQTT Protocol [J]. Information Communication, 2016: 39-41.
[10] Design and Implementation of Internet of Things Platform Based on MQTT Protocol [D]. Xi’an University of Posts and Telecommunications, 2018.