Measurement and Analysis of Network Data Based on MQTT Protocol

Fu Chen
School of Information
Central University of Finance and Economics
BeiJing, China
chenfu@cufe.edu.cn

Yongfeng Huang
School of Information
Central University of Finance and Economics
BeiJing, China
huangyongfenghyf@163.com

JianMing Zhu
School of Information
Central University of Finance and Economics
BeiJing, China
zjm@cufe.edu.cn

Sheng Gao
School of Information
Central University of Finance and Economics
BeiJing, China
sgao555@163.com

Zhiyuan Sui
School of Information
Central University of Finance and Economics
BeiJing, China
zx20work@126.com

MeiJiao Duan
School of Information
Central University of Finance and Economics
BeiJing, China
duanmeijiao@cufe.edu.cn

***Abstract*—As a lightweight protocol, the Message Queuing Telemetry Transport (MQTT) is widely used in the scopes of Internet of Things on account of its quite low bandwidth requirements. We built a temperature and humidity monitoring system using the Raspberry Pi and NodeMcu to communicate with each other through the MQTT protocol. This work applies network analytics tools such as Wireshark to capture and analyze network data. We looped message to broker with different QoS level and time interval and then observed the RTT (Round Trip Time) to analyze correlation between them.**

Keywords—IOT, MQTT, Analyze, Round Trip Time, NodeMcu

I. Introduction

The concept of the Internet of Things means that everything can be connected to the network. The Internet of Things is a concept that is getting closer to life. The official definition of the Internet of Things is: based on the Internet, the process of communicating between non-communicable objects and objects is called Internet of Things [1]. In layman's terms, the Internet of Things is to connect all objects to the network, and to realize the exchange and sharing of information between people and things, things and things. The development of the Internet and artificial intelligence has also led to the rapid development of the Internet of Things. More and more smart devices, terminals, and sensors are connected to each other, allowing people to live in an "interconnected environment." According to the image description of the Internet of Things structure in the White Paper of the Internet of Things, it can be concluded that the Internet of Things consists of three parts.

The bottom layer is the technology part, including sensor technology, embedded technology and connection technology, as well as sensor devices, embedded devices and communication devices related to these three technologies. The middle layer is the software part, including device driver software, server software and application client software. Above the software is the application part of the Internet of Things, which can be divided into industrial applications and civilian applications. The Internet of Things plays a major role in smart cities, smart transportation, smart homes, smart government, smart industries, smart security and other construction projects [2]. The popularity of the Internet of Things has brought about tremendous changes in people's daily lives and industrial production activities. Cisco IBSG predicts that by 2020, the number of IoT devices will reach 50 billion, and the Internet of Things era has arrived.

A variety of communication protocols are currently used in IoT messaging, such as Hypertext Transfer Protocol (HTTP), Constrained Application Protocol (CoAP), Extensible Messaging and Presence Protocol (XMPP), Advanced Message Queuing. Protocol (AMQP) and MQ Telemetry Protocol (MQTT). Considering resource consumption, performance, resource usage and reliability, MQTT is the best choice for IoT communication protocols. MQTT is a very simple and lightweight messaging protocol with a publish/subscribe architecture that is designed to be easy to deploy and capable of supporting thousands of clients using a single server. In addition, MQTT guarantees the reliability and efficiency of message delivery under the unfavorable conditions of network environment and low bandwidth. MQTT is widely used in the communication between IoT devices because of its own advantages [10]. It has also caused many scholars to study, including the application of MQTT protocol, data transmission security, robustness of transmission mechanism, and measurement of MQTT network performance [3].

In [4] and [5], the authors mainly implemented the real application scenarios of MQTT, including heart rate monitoring, environmental monitoring in agriculture, etc.

The research was supported in part by the National Science Foundation of China: A Study on Foundational Theory of Flow Optimizing and Scheduling in Software-defined Networking (No. 61672104), and other National Science Foundation of China U1509214, 61702570, **61602537.**

They sent the data collected by the sensors to the Raspberry Pi. These are the applications of the Internet of Things sensor and the MQTT protocol, and there is no measurement of the performance of the MQTT protocol communication. In the work of[6], the author uses the search engine to statistically analyze the security usage of MQTT in the Internet of Things system, and proposes several attack scenarios to improve the importance of using MQTT security. In the work of the article [7], the author observes the change of the packet loss rate and the end-to-end delay according to the different payload of the MQTT, and compares the data transmission difference between the wireless network and the wired network, considering that the two networks have different transmission performance. The advantage of the author's MQTT measurement method is that it utilizes a real network environment, but not a real IoT environment.

In this paper, based on the transmission performance of MQTT protocol, nodemcu and Raspberry Pi are used to build MQTT protocol measurement environment. In a real IoT environment, the MQTT protocol is used to communicate, measure network performance, and draw conclusions about the correlation between network latency, QOS service quality in MQTT, and message publishing speed. The subsequent chapters of the article are as follows. The second section introduces the MQTT message type, transmission mechanism and quality of service. The third section introduces the system design in the article and the experimental design of network measurement. The fourth section shows the measurement results and the analogy and analysis of the experimental results. At the end of the last section, the conclusions and subsequent work are given.

II. BACKGROUND

Traditional communication mechanism between devices generally applied point-to-point transmission. The communication parties match each other and then transmit the data , as show below Fig.1. In this case, if there are multiple devices communicating at the same time, it will occupy more channels.

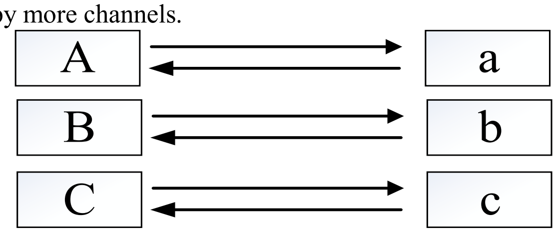

Figure 1. Point-to point communication mechanism

MQTT is a communication protocol widely used between IoT devices. Its communication mechanism mainly includes three part:broker,subscriber and publisher. Different from the traditional point -to point communication mechanism, MQTT is a proxy-based messaging model(this proxy refers to the broker). The subscriber subscribes to the message through the broker, as shown in Figure 2. When a publisher posts the message to the broker, the subscriber can receive the message. This proxy-based subscription/publishing messaging model provides a one-to-many message publishing model that uncouples the application.

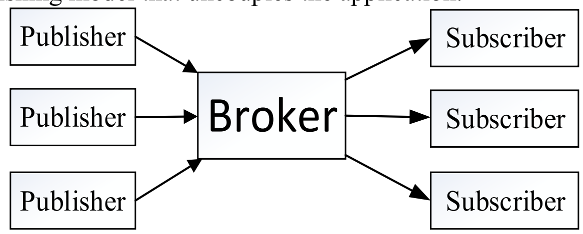

Figure 2. MQTT communication mechanism

The structure of an MQTT control packet consists of a fixed header, variable header and payload[8]. The fixed header consists of two bytes, and the 4 to 7 bits of the byte 1 from 16 possibilities, indicating 14 message types that exist in MQTT ,as shown in Table Ⅰ.

TABLE I. CONTROL PACKET TYPES

Name	Value	Description
Reserved	0	Reserved
CONNECT	1	Client request to connect
CONNACK	2	Connect Acknowledgment
PUBLISH	3	Publish message
PUBACK	4	Publish Acknowledgment
PUBREC	5	Publish Received
PUBREL	6	Publish Release
PUBCOMP	7	Publish Complete
SUBSCRIBE	8	Client Subscribe request
SUBACK	9	Subscribe Acknowledgment
UNSUBSCRIBE	10	Client Unsubscribe request
UNSUBACK	11	Unsubscribe Acknowledgment
PINGREQ	12	PING Request
PINGRESP	13	PING Response
DISCONNECT	14	Client is Disconnecting
Reserved	15	Reserved

There are three types of delivery service quality for MQTT messages, which are represented by 2-1 bits of the first byte of the variable header. The three qualities of service are expressed as:

QoS=0: At most once delivery, the message will only be delivered once, regardless of whether the receiver-broker or the subscriber receives the message. In this case, packet loss may occur.

QoS=1: At least once delivery, the message will be delivered at least once. In the case of guaranteeing the delivery of the message, repeated transmission of the message may occur in the working mode of QOS=1.

QoS=2: Exactly once delivery, the message will be guaranteed to be delivered, and will only be delivered once. Like this one-time messaging mode, it can often be used on systems such as network payment.

Combined with the MQTT message structure and message delivery mechanism, a system experiment will be designed to measure the MQTT network environment, and then the conclusion is drawn.

III. SYSTEM DESIGN

We built a temperature and humidity monitoring system using the Raspberry Pi and NodeMcu to communicate with each other through the MQTT protocol. A large part of our experiments on MQTT and the Internet of Things are done on this real IoT system, not on the data of the interaction between the virtual client and the system. The system consists of three main parts, including the Raspberry Pi handheld device, nodemcu, and a personal computer. In the temperature and humidity monitoring system, the handheld device mainly uses the Raspberry Pi - a microcomputer with a Linux system as an intermediate agent and nodemcu as a sensor node. We configured the wireless network information in the placement environment on the Raspberry Pi, connected it to Wi-Fi, and ran mosquitto on its Linux system. This sets up the intermediate agent. The main part of the Raspberry Pi handheld device is shown in the figure 3.

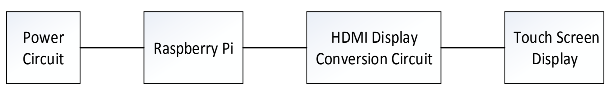

Figure 3. Handheld device composition

The node sensor mainly includes battery, esp8266 module, Wi-Fi antenna and temperature and humidity sensor. We use the Lua language to write code, write wireless network configuration information, such as ssid, password to the esp8266 module, and write the client code to realize the reading and sending of temperature and humidity. The node sensor sends the temperature and humidity information to the Raspberry Pi handheld device through the MQTT protocol, and the Raspberry Pi will display the node data subscribed to on the display screen of the handheld device, and publish the corresponding topic data to nodemcu to implement a complete IoT system. Based on the system we built, we performed network data measurement of the MQTT protocol.

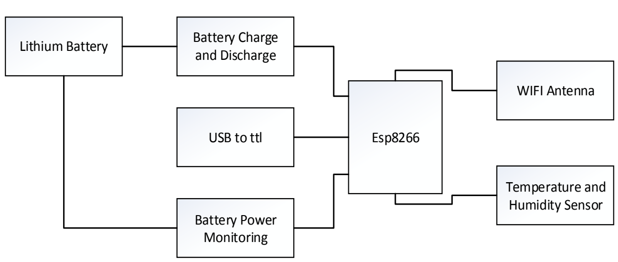

Figure 4. Nodemcu composition

Network measurement is the most effective means to obtain network performance. Through the measurement of the network, the purpose of network monitoring is achieved. Through the analysis of network performance, we can optimize the network, speed up network transmission, improve service quality, and solve problems such as delay and congestion in network transmission. According to the relevant draft and the network performance indicator file published by the IETF's IP Network Performance Measurement Working Group (IPPM), we divide the MQTT network measurement into network delay measurement, MQTT network throughput measurement and MQTT network packet loss rate measurement. There are many reasons for affecting network performance, such as hardware support, network environment, and the data structure of the MQTT protocol we are concerned with, the format of the message, and so on[9]. In this paper, we measure the network conditions of cyclically releasing the same message at different quality of service and different time intervals. We learned about the correlation between them by observing round trip delay (RTT), jitter, and throughput. We set up the code function as shown below.

```
tmr.alarm(2, 50, 1, function()
    if mqttConnectedFlage == 1 then
        client:publish(PublishTopic, playloads, 2,0)
    end
end)
```

Figure 5. Cyclic transmission

Using the timing function, messages are sent at intervals of 1s, 700ms, 500ms, 400ms, 300ms, 200ms, 100ms, 50ms, and Nodemcu will cyclically push messages to our mosquitto server at set intervals. The network measurement structure in this paper is as follows. One nodemcu posts the message to our computer agent, and then another nodemcu subscribes to the message. We use the packet capture software on the computer to obtain the traffic packet on the network card and perform statistics and analysis. The entire experiment included network delay measurements at three QOS service levels. Figure 6 is the experimental schematic diagram.

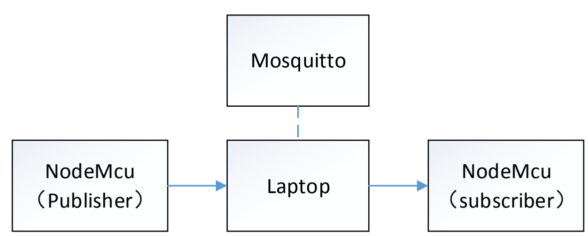

Figure 6. Experimental schematic diagram

IV. EXPERIMENT

According to different QoS, the experiment is divided into three parts. Observe and measure the network data transmitted between nodemcu and mosquitto on laptop at different time intervals when QoS=1, QoS=2, and QoS=3. The sampling method is used to count the average delay and jitter of the published message and the subscribed message.

A. Round Trip Delay for Publishing Messages

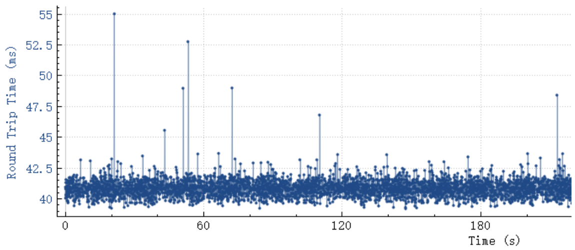

Figure 7. Experiment Figure 1

Figure 7 shows the round-trip delay between the issue of message from nodemcu to laptop when QoS = 0. The cycle time interval is 50 milliseconds, the abscissa indicates the message release time, and the ordinate is the round trip delay. As can be seen from the figure, the overall release delay is about 40 milliseconds, except for the high latency of individual packets. Changing the time interval, we found that the round-trip delay of the published message is basically unchanged at different time intervals.

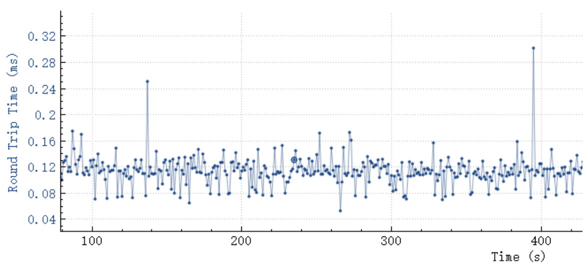

Figure 8. Experiment Figure 2

When QoS=1, the obtained round-trip delay of the published message is as shown in Fig. 8. After observation, it was found that the round-trip delay of the published message was around 0.12 milliseconds. When we change the time interval, the experimental results tend to be consistent.

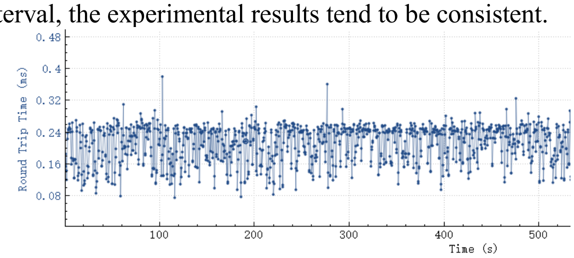

Figure 9. Experiment Figure 3

Figure 9 shows the delay in issuing a message when QoS = 2. The data indicates that the change in delay is small at different time intervals.

Figure 7, Figure 8, and Figure 9 respectively report the delays of sending messages from the nodemcu to the laptop under three quality of service. The time interval for sending messages is all from 1 second to 50 milliseconds. The experimental results show that if the time interval is changed, the round-trip delay of posting messages under the same quality of service is basically unchanged, and the difference may be due to our hardware environment and network fluctuations. However, the delays of the published messages of different quality of service are different. In the case of QoS=1 and QoS=2, the delay is small, and the MQTT transmission performance is better. In the case of QoS=0, the delay of issuing messages is much larger than the former two. . Comprehensive observation of these three figures shows that when QoS=0, the jitter is larger, and when QoS=1 and QoS=2, the jitter is smaller.

B. Round Trip Delay for Subscription Messages

This part of the experiment is mainly to measure the delay of the nodemcu subscription message.

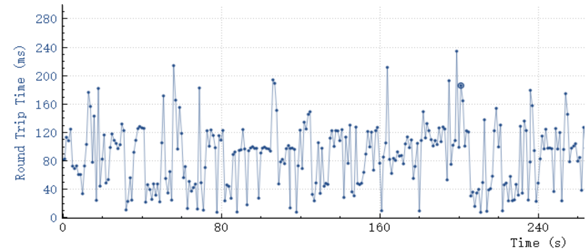

Figure 10. Experiment Figure 4

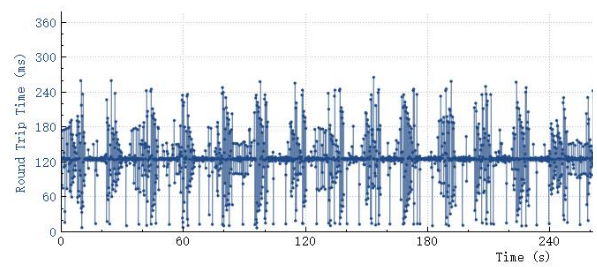

Figure 11. Experiment Figure 5

When QoS=0, Fig. 10 and Fig. 11 respectively show experimental results of setting the timer time interval to 1 second and 50 milliseconds. We can use wireshark to fetch network data, and the round-trip delay of the subscribed message is as shown. Obviously, the round-trip delay of the subscription messages with an interval of 1 second and 50 milliseconds is quite different. We change the time interval to 1 second, 700ms, 500ms, 400ms, 300ms, 200ms, 100ms, 50ms, and use the sampling method to obtain the average value and draw the trend chart as shown in Figure 12.

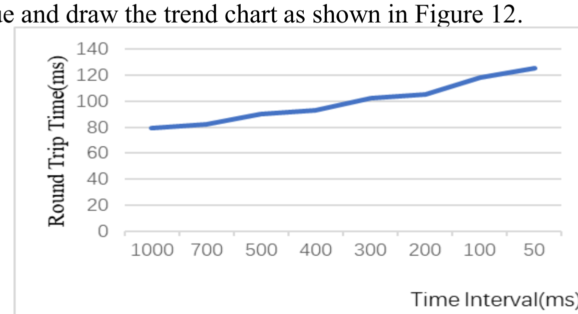

Figure 12. Data analysis chart

The experimental results show that as the time interval decreases, the subscription delay under QoS=1 gradually becomes larger.

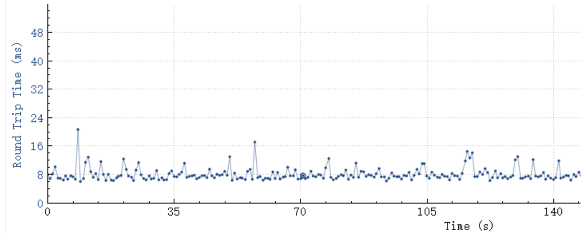

Figure 13. Experiment Figure 5

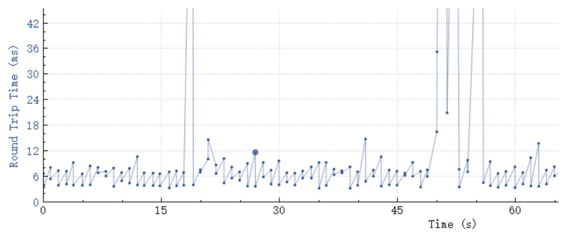

Figure 14. Experiment Figure 6

Figures 13 and 14 show the delay of the subscription message when QoS = 1 and QoS = 2, respectively. Observing the statistical data, it is known that, in the case of the same quality of service, if the time interval is changed, the delay of subscribing to the message remains almost unchanged.

Through Fig. 10, Fig. 11, Fig. 13 and Fig. 14, it can be found that when QoS=0, the delay of the subscription message gradually increases with the decrease of the time interval, and the jitter is large. When QoS=1 and QoS=2, the delay of the subscription message is independent of the time interval and the jitter is very small.

V. CONCLUSION

In the real IoT environment built by nodemcu and Raspberry Pi, the article measures the round trip delay of MQTT messages with different quality of service and different time intervals. The experimental results show that the round-trip delay of QoS=1 and QoS=2 is much smaller than the round-trip delay of QoS=0, and the jitter is small, whether it is a published message or a subscribed message. Moreover, in the subscription message, when QoS=0, the round-trip delay increases as the time interval decreases, but when QoS=1 and QoS=2, the round-trip delay remains substantially unchanged with the decrease of the time interval. Therefore, in the future we will pay more attention to the message passing performance of QoS=0. In the subsequent work, it is hoped to improve the transmission efficiency by optimizing the code structure of the MQTT protocol.

ACKNOWLEDGMENT

The research was supported in part by the National Science Foundation of China under No.61672104, 61170209, 61702570, 61602537, U1509214. National Science Foundation of China: A Study on Foundational Theory of Flow Optimizing and Scheduling in Software-definedNetworking (No. 61672104).

REFERENCES

[1] Deepsubhra Guha Roy, Bipasha Mahato, Debashis De, Rajkumar Buyya,” Application-awareend-to-enddelayandmessagelossestimationin InternetofThings(IoT)—MQTT-SNprotocols”,Contents lists available at ScienceDirect Future Generation Computer Systems,Future Generation ComputerS ystems89(2018)300–316.

[2] PietroColombo,ElenaFerrari,” AccessControlEnforcementwithinMQTT-basedInternetof ThingsEcosystems”, SACMAT’18, June 13-15, 2018, Indianapolis, IN, USA.

[3] Muhammad Ikrar Yamin, Son Kuswadi, Sukaridhoto,” Real Performance Evaluation On MQTT and COAP Protocol in Ubiquitous Network Robot Platform (UNR-PF) for Disaster Multi-robot Communication”, EMITTER International Journal of Engineering Technology Vol. 6, No. 2, December 2018 ISSN:2443-1168.

[4] Komkrit Chooruang,Pongpat Mangkalakeeree,”Wireless Heart Rate Monitoring System using MQTT”Procedia Computer Science 86(2016)160-163[2016 International Electrical Engineering Congress,iEECON2016,2-4 March 2016,Chiang Mai,Thailand].

[5] Nagesh U.B,Uday D.V,Shamitha Gurunath Talekar,pooja S,”Application of MQTT Protocol for Real Time Weather Monitoring and Precision Farming”,2017 International Conference on Electronics,Communication,Computer and Optimization Techniques(ICEEECCOT).

[6] Syaiful Andy,Budi Rahardjo,Bangus Hanindhito,”Attack Scenarios and Security Analysis of MQTT Communication Protocol in IoT System”,Proc.EECSI2017,Yogyakarta,Indonesia,19-21 September 2017.

[7] Shinho Lee, Hyeonwoo Kim, Dong-kweon Hong,Hongtaek Ju, “Correlation Analysis of MQTT Loss and Delay According to QoS Level,” ICOIN2013.

[8] The MQTT protocol, http://www.mqtt.org, cited August, 2012.

[9] Cai Z P, Yin J P, Liu F, et al. Efficiently monitoring link bandwidth in IP networks. In: Proceedings of IEEE Global Telecommunications Conference, St. Louis, 2005.

[10] Konglong Tang, Yong Wang, ,Hao Liu, ,Yanxiu Sheng,et al,” Design and Implementation of Push Notification System Based on the MQTT Protocol”, International Conference on Information Science and Computer Applications (ISCA 2013).