

Design and Implementation of Push Notification System Based on the MQTT Protocol

Konglong Tang^{1,a}, Yong Wang^{1,2,b}, Hao Liu^{1,c}, Yanxiu Sheng^{1,d},
Xi Wang^{3,e} and Zhiqiang Wei^{1,f}

¹ Department of Computer Science, Ocean University of China, Qingdao, China

² State Key Laboratory of Software Engineering, Wuhan University, Wuhan, China

³ Central Research Institute of Haier Group, Qingdao, china

^a851831319@qq.com, ^bmarkwy@126.com, ^cliu.hao@msn.com, ^dshengyanxiu@ouc.edu.cn,

^ewangxi@haier.com, ^fweizhiqiang@ouc.edu.cn

Keywords: MQTT Protocol, instant messaging, Message Broker, Mobile Applications

Abstract. MQTT (Message Queuing Telemetry Transport) is a broker-based publishing/subscribing, instant messaging protocol. It's designed to be open, simple, lightweight and easy to implement. This paper described a method of pushing notification system based on the MQTT protocol. It can be used to solve the problem of instant pushing various messages from the server to the mobile client. This method is suitable for developing the smartphone applications. This paper also carried out research and design on the realization of MQTT protocol. With support of the proxy server message broker, the protocol put forward solutions for the server sending the specified messages to the specified mobile devices and the mobile clients receiving pushed messages from the server.

Introduction

With the popularity of smartphone, people are increasingly dependent on their mobile phones to get real-time messages, so instant messaging is particularly important^[1]. Currently we can bypass the carriers, through a standard TCP/IP network to send messages directly to the mobile phone^[2,3]. PointCast companies in the United States in 1996 first put forward the information push technology. The purpose of this technology is to make full use of the Internet, push the different areas of information to the company's customers, such as science technology, economics, sports, education. In after a few years, the push technology has become the most popular network of technology, many companies are aiming at the technology for research, and launched their own products, and apply this technology to many operating systems.

There was already the perfect push server in Apple's IOS platform, but in Android platform it is relatively hard to implement. There also has several mainstream push programs in Android platform^[4]. It's simple to achieve by polling way, but there are many deficiencies and conflicts among real-time, power and data traffic consumption. Using SMS mode, it will spend a lot of costs if you don't cooperate with operators. Although the persistent connections program has been widely used, it also has some shortcomings. Aiming at the deficiency of these solutions, we can make good designs to allow the solution to work effectively^[5].

Therefore, the persistent connections approach is a more ideal solution, MQTT protocol become our first choice because of its obvious advantages^[6]. It is an open, simple, lightweight, and easy to implement messaging protocol. Initially it was designed to connect large numbers of remote sensors and control devices. The protocol has been applied in a variety of embedded systems. Hospitals use this protocol to communicate with pacemakers and other medical equipment supplier. Oil and gas companies use it to monitor oil pipeline thousands of miles away.

In this paper, the program based on the MQTT protocol is proposed to achieve instant messaging push. It can avoid the cost of electricity and reduce data traffic consumption. The server publishes messages actively and the client receives the message in real time, in this way the message push can be optimized and innovated.

Protocol Overview

MQTT. MQTT is developed by IBM. It is both an instant messaging protocol and a lightweight broker-based publishing/subscribing messaging protocol^[7]. The protocol supports all platforms and a variety of popular programming languages. It is an ideal solution for pushing messages. Many domestic enterprises widely use MQTT protocol as an Android phone and server-side pushing message protocol. Because of its simplicity and scalability, it is widely used in the field of mobile pushing messages.

MQTT offers three quality news services: "at most once" news release is completely dependent on the underlying TCP/IP network; Lost or duplicated messages will occur; "at least once" ensures that the message arrived, but the message repetition may occur; "only once" ensures that messages arrive once.

Message Broker. Really Small Message Broker is a simple MQTT agency, which is provided by IBM. You can use it to easily build a highly efficient instant messaging server. In applications, it is responsible for receiving messages from the server and forwarding it to the designated mobile device. Each message must go through Message Broker, received or sent to a different client.

After Message Broker message format is as follows. Byte1: including the message type and tag field (DUP, QoS, TETAIN). Byte2: transfer data we need to transfer (at least one byte).

Table 1 Message Format

Bit	7	6	5	4	3	2	1	0
Byte 1	Message Type				Dup flag	QoS level		RETAIN
Byte 2	Remaining Length							

Overall Design

Publishing/Subscribing Message Model. The design is based on publishing/subscribing messaging model^[8]. In this model, the publisher and subscriber are both clients, the message destination is called theme. By connecting to the message broker they transfer data across the network. Publishers send a specific topic of messages to message broker, subscribers subscribe specific news topics to the message broker, and the connection between the subscriber and publishers managed by the message broker. When the message broker receives the published messages, it delivers the message to subscriber. Publishing/subscribing messaging model allows multiple providers to publish messages to the same topic. It also allows multiple users to subscribe messages with a subject. Then the message broker will broadcast to different subscribers.

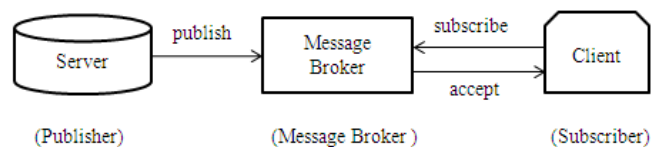


Fig. 1 Overall Architecture Diagram

As a message broker, it completes message routing function, receiving the message sent by the server, and then forwarded. Subscribers subscribe to the interesting topics, to submit information to the Message Broker, and to maintain a persistent connection. If publisher check into a new message,

this message will be released by topic category. Depending on the topic, the Message Agent receives the message as well as the client's subscription, and releases this news to the corresponding phone.

Push Process.

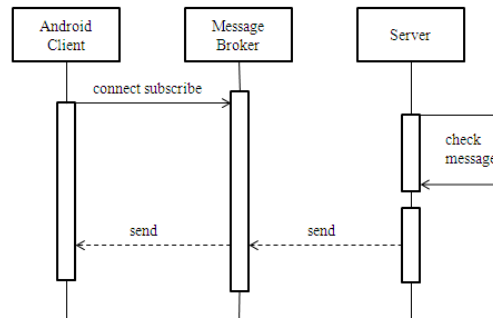


Fig. 2 Push Process

Process is as follows.

First, connecting *MQTT*. The client specifies the message broker host 's IP address and port number assigned by MQTT, establishes a connection and creates MQTT client object. MQTT object subscribes to a topic, requiring both subscription array parameter and service quality array parameter.

Second, releasing Information to the Agency. Interface server directly using the API functions provided by MQTT, requires theme, message, service quality and other parameters. Send push messages to the Message Broker.

Third, receiving a Message from the Agent. To make sure that the subscriber can receive messages, you must create a callback function and register in MQTT client. Callback Function informs system that topic message has arrived at the client.

System Development

In the push operation, the server providing information and mobile client receiving messages are seen as "Client" of Message Broker. In other words, whether the server sends a message based on the subject or the client receives messages by subject, they all go through a proxy server Message Broker. In this way, you can achieve own servers and clients to integrate well with Message Broker, also take advantage of Message Broker stability Message Broker to treat message queue. It can improve system stability.

At the Android client application beginning, whether you are running or not, we make clients to receive notices and thus open a separate process of background services. Service registered in file named Androidmanifest.xml. In the service, open threads. Use MQTT Simple Callback interface to create a simple objects, connected to the Message Broker, When push message arrives, message Arrived method is called, then call notification mechanism unfolded. When the connection with the broker terminates unexpectedly, it calls connection Lost method. This method attempts to reconnect to the broker. If cancel message push, you need to stop the service and then end up thread.

Server only uses functions provided MQTT publish to publish messages, and did not use the Android API, so we are able to execute Java programs. You can also update the data in the database, perform the above procedure to send push messages to the Message Broker and enters issued queue. If you do not ask for real-time push message, you can put the polling of client on server without the need to worry about data traffic and power consumption.

To analyze consumption of this system, we choose 12 hours and 24 hours to evaluate the power

and network data consumption by my own Smartphone. After several tests, the average results are shown in table.

Table 2 Effect of the power and Network data consumption

Time	Power consumption	Network data consumption	Loss rate
12 hours	10%	20KB	0.3%
24 hours	20%	40KB	1%

According to the Table 2, we can come to the conclusion that network data consumption is about 1MB per month. It consumes very little power, and has a very high accuracy. So it basically meets the needs of application.

Conclusion

In the information age, people are immersed in the huge amounts of data. The effective and active information push service improves the efficiency of access to information and enhances mobile applications stickiness.

In this paper, we specified the design of a pushing notification system and discussed details of key techniques that make the system effective and easy to maintain. This message push system is based on MQTT protocol and makes it possible to push messages to clients in real-time, can be applied to many mobile phone applications.

References

- [1] Zhang Wenmao, Zhang Miao, Bi Jun. Instant Messaging: the Present and the Future [J]. Journal of Chinese Computer Systems, 2007, 28(7): 1162-1168.
- [2] Liu Peiji, Wu Yanai. The Application of Push Technology in Mobile Internet [J]. Communications World, 2001(31):31-32.
- [3] Suo Chuanjun. Review of Development and Application Research of Push Technique [J]. New Technology of Library and Information Service, 2003(3): 48-50.
- [4] Wang Yifeng. Realization of Push Mechanism in Android Platform [J]. Programming Skills & Maintenance, 2011(22): 29-31.
- [5] Zhang Changxue, Zhang Wei, Dong Zhiming. Aspects of Mobile Push Technology [J]. Mobile Communications, 2011, 5: 21-27.
- [6] He Shaoyue, Xu Xiaodong, Ma Zuyuan. The Application of Active Push Technology in Mobile Collaboration Education [J]. Modern Education Technology, 2012, 04: 100-103.
- [7] Whittaker S, Swanson J, Kucan J, et al. TeleNotes: Managing lightweight interactions in the desktop [J]. ACM Transactions on Computer-Human Interaction, 1997, 4(2): 137-168.
- [8] Gero Mühl. Large scale content based publish subscribe systems: [Ph D Thesis]. Darmstadt University of Technology, 2002.

Acknowledgement

This work was supported by Technology development and Cooperation (No.20120465); Doctoral Fund of Ministry of Education of China (No.2010013211009); National Natural Science Foundation of China (No.61170312).