

基于 MQTT 协议的即时消息业务设计与实现^①

林 浒¹, 张家铭^{1,2}, 杨海波¹

¹(中国科学院 沈阳计算技术研究所, 沈阳 110168)

²(中国科学院大学, 北京 100049)

摘要: 近年来, 宽带接入技术正以十分惊人的速度发展. 与此同时, 移动互联网技术也日益成熟, 即时消息业务已成为移动互联时代的应用热点. 在互联网中 XMPP 和 SIMPLE 被广泛使用, 但其并不能很好的适用于移动互联网. 采用发布/订阅模型的 MQTT 协议是一种轻量级的消息传输协议, 具有低功耗、节省流量和可扩展性强的优点. 本文首先分析了 XMPP 和 SIMPLE 协议的不足之处, 研究了 MQTT 协议的消息格式以及协议的使用方式, 之后对即时消息业务进行了设计和实现. 并在功能和性能上进行了相关的测试和分析.

关键词: 即时消息; 发布/订阅; MQTT; 移动互联网

Design and Implementation of Instant Messaging Business Based on the MQTT Protocol

LIN Hu¹, ZHANG Jia-Ming^{1,2}, YANG Hai-Bo¹

¹(Shenyang Institute of Computing Technology, Chinese Academy of Sciences, Shenyang 110168, China)

²(University of Chinese Academy of Sciences, Beijing 100049, China)

Abstract: In recent years, broadband access is developing with astonishing speed. At the same time, the mobile Internet technology is also increasingly mature, instant messaging business have become a hot point of the era of mobile Internet applications. XMPP and SIMPLE are widely used in the Internet, but they are not suited to the mobile Internet. The MQTT is a publish/subscribe model based and lightweight messaging protocol, which has the advantages of reducing power consumption, decreasing flow and has strong flexibility. Firstly, this article analyzes the shortcomings of XMPP and SIMPLE protocol. Secondly, it introduces the message format and the usage of the MQTT protocol. After the design and implementation of the instant messaging business, the related testing and analysis are carried on the function and performance.

Key words: instant messaging; Pub/Sub; MQTT; mobile Internet

随着移动互联网和移动智能终端的普及, 移动智能终端已经成为网民接入互联网的重要途径, 即时通信迎来了前所未有的发展契机. 与此同时, 由于用户通常会随身携带手机, 移动消息应用在手机客户端上的重要性也逐渐凸显, 移动消息中的即时消息业务成为一个炙手可热的研究领域. 目前, 即时消息除了通过采用私有协议(如 QQ、微信等)实现外, 还主要通过 SIMPLE 和 XMPP 两大类协议. 然而, 由于移动智能终端对流量有限制, 对网络的带宽以及终端的功耗等有特殊的要求, 因此即时消息业务对即时通信协议有更

高的要求. 然而上述两种协议均不能较好的满足移动消息的应用场景. 本文通过对解决当前即时消息业务的相关协议进行研究和分析, 提出了更适合移动互联网的基于 MQTT 协议的即时消息业务解决方案.

本文主要分为四部分, 第一部分对 SIMPLE 和 XMPP 协议进行了相关的研究, 并分析了在移动互联网下两协议的不足之处. 第二部分提出了基于发布/订阅的轻量级的 MQTT 协议, 对其进行了简单地分析, 并阐述了它的消息格式. 第三部分对即时消息业务的话题、payload 以及消息路由进行了设计和实现. 第四

^① 基金项目: 教育部-中国移动科研基金(2015)(MCM20150103)

收稿时间: 2016-06-30; 收到修改稿时间: 2016-08-18 [doi: 10.15888/j.cnki.csa.005678]

部分,对本文设计的即时消息系统进行了相关的测试和分析。

1 相关协议分析

1.1 SIMPLE 协议

SIMPLE(SIP for Instant Messaging and Presence Leveraging Extensions)协议簇是由 IETF SIMPLE 工作组制定的,主要是扩展 SIP 协议,以使其支持 IM 服务^[3]。SIMPLE 工作组为 SIMPLE 协议的规范制定做了大量的工作,并且发布了一系列 RFC 文档。SIMPLE 提出 session 模式和 page 模式^[3]。session 模式中,即时消息被作为一种类似于音视频的媒体,通过 SIP 信令进行交互,即时消息协商后建立一个会话,通过建立的会话进行消息媒体的交互。page 模式中,通过 SIP 请求包裹即时消息的内容发送消息的方式,通过对 SIP 协议的扩展, SIMPLE 协议有了很大的改进。在完成了 SIP 信令协商之后,一般的多媒体会话需要使用一些其他协议来建立用户代理之间的会话通道以此完成会话数据的交互。基于 SIMPLE 协议的即时消息交互不需要建立会话通道,消息通过 SIMPLE 协议的消息命令直接发送,每个消息都由一个单独的消息命令来传播,彼此独立。

1.2 XMPP 协议

XMPP 由 IETF 于 2004 年完成了标准化工作,符合 RFC2778 和 RFC2779 规范。XMPP 源于 Jabber 协议,它是以 XML 为基础开放式的即时通信协议^[4]。XMPP 协议继承了 XML 的灵活性、扩展性,它以 TCP/IP 传输为基础,定义了客户端、服务器和网关三种角色。服务器不仅要承担客户端信息记录,还要负责连接管理和信息的路由功能。网关负责各异构即时通讯系统之间的通信。

1.3 XMPP、SIMPLE 协议在移动互联网中的不足

XMPP 和 SIMPLE 协议均是基于字符文本进行传输的通信协议,字符文本协议通信的效率较低,为了确保通信的安全,二者采用了 TLS 等加密传输机制计算量较大,功耗较高。

XMPP 和 SIMPLE 协议是基于相对可靠的网络上的应用层协议,在底层网络不稳定的情况下,容忍机制不够健全,应用在移动互联网即时通信时,网络的不稳定现象经常出现,恢复网络连接需要较多交互数据包,浪费用户流量,降低了用户体验。

XMPP 和 SIMPLE 协议是基于消息体寻址的协议,消息体中包含了消息的发送者、消息的接收者以及消息路由用的会话标示等头域,这使得消息体过大,极大的降低了带宽的利用率。

2 基于发布/订阅的MQTT协议

2.1 发布/订阅模型概述

发布/订阅模型是一种消息传播模式,消息的发布者不直接将消息发送到消息的订阅者,而是根据某种特征将发布的消息进行分类,在这个过程中消息的发布者不需要关注消息的订阅者。同样,消息的订阅者通过订阅某个感兴趣的消息,不需要关注消息的发布者。在这种机制下,消息的若干个发布者与若干个订阅者之间不需要直接进行通信,而是通过建立消息代理作为中介互相通信。通过这种发布者与订阅者之间的解耦合关系,这种模式提供了更好的网络扩展性和更动态的网络拓扑。

发布/订阅模型中,消息的订阅者往往只接受消息发布者所发布消息的某个子集。消息的过滤指的是对接受和处理的消息进行选择的过程。通常有两种过滤形式:分别是基于主题和基于内容的过滤, MQTT 协议采用基于主题的发布/订阅模式。

在基于主题(Topic)的发布/订阅模型中,消息以特定的主题名标识被发布者发布至消息代理服务器上。基于主题的发布/订阅模式中,订阅同一主题的订阅者将会收到相同的消息,消息的订阅者可以订阅多个主题,发布者负责定义订阅者可以订阅的消息类别。消息代理负责维护消息队列,执行消息的存储转发功能。

2.2 MQTT 协议简介

MQTT(Message Queuing Telemetry Transport, 消息队列遥测传输)是由 IBM 公司开发的即时通讯协议,它是一个基于发布/订阅模型、轻量级的消息传输协议^[1]。具有开放、易用、精简等特点。MQTT 协议是为那些计算能力有限,且需要工作在低带宽、不可靠的网络通讯而设计的协议^[2]。

MQTT 主要有以下几个特性:

① 使用基于主题的发布/订阅消息模式,提供一对多的消息发布,屏蔽了应用程序之间的耦合性;

② MQTT 协议有三种级别的消息发布服务质量^[7]。“至多一次”,发生的消息会丢失或重复,这一级别可

用于一些传感器传输数据, 消息丢失一次对系统不会有严重影响, 因为在不久之后系统还会进行消息的第二次发送; “至少一次”, 该情况下能确保传输消息的准确到达, 但有可能产生消息重复的传输的现象; “只有一次”, 确保消息只到达一次. 这一级别可用于一些计费系统, 使用该级别表明进行传输的消息内容特别重要;

③ MQTT 协议采用二进制的形式表达, 固定长度的头部只有 2 字节, 协议交换达到最小化, 极大的降低了网络带宽的开销.

2.3 MQTT 协议的消息格式

MQTT 消息体由固定报头、可变报头以及有效载荷三部分组成^[6], 固定报头是每个消息体都必须包含的部分, 固定报头部分的长度为 2 字节. 消息格式如表 1 所示.

表 1 MQTT 协议的消息格式

固定报头	可变头域	有效负载
Fixed Header	Variable Header	Payload
必有	有些消息可以没有	

消息体的固定报头有 2 个字节长, 具体的格式如表 2 所示.

表 2 固定头域

bit	7	6	5	4	3	2	1	0
Byte 1	MQTT Control Packet Type				DUP Flag	QOS Flag		Retain Flag
Byte 2	Remaining Length							

固定报头的高四位是消息的类型, 最多可以支持 16 种的消息类型. 目前, MQTT 协议已经定义了 14 种消息类型^[5].

Dup Flag 用于标识是否第一次发送该消息. 当 Dup Flag 为 0 时表示第一次发送该消息. 虽然 Dup Flag 为 1 表示该消息是重传消息, 但是不能保证用户之前收到过该消息.

QOS 表明发布消息的交付质量等级, 可选值为 0、1、2.

Retain 标识仅在发布消息中使用. 对于消息的发布者, 当客户端发送消息时, 如果该标识设置为 1, 则消息发送到订阅该主题的所有订阅者之后, 代理服务器仍然保存该消息. 对于消息的订阅者, 当一个主题有了新的订阅用户, 最后被保留的消息主题将被设置保留字段, 然后发送到新的消息订阅者. 当主题不存

在相关的保留信息时, 则不发送.

Remaining Length 是剩余长度, 包含 Variable header、payload 的长度, 默认 Remaining Length 为 1 字节, 最多支持扩展 4 字节, 因此一条 MQTT 可携带的消息最高是 256MB.

3 即时消息业务的设计与实现

即时通信在实际的应用场景中, 客户端与服务器的具体交互流程如图 1 所示.

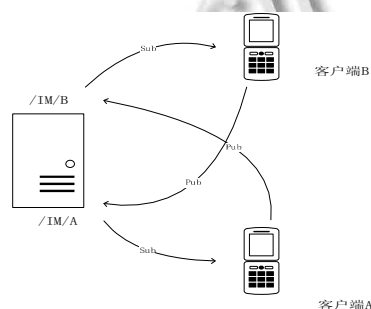


图 1 IM 交互流程简图

客户端 A、B 会分别订阅/IM/A 和/IM/B 主题. 客户端 A 和 B 会向对方的主题上发布消息, 之后会通过代理服务器进行转发^[8]. MQTT 协议中消息不携带路由信息和消息的内容类型. 因此, 我们需要对 payload 中携带的消息格式根据业务的需求进行明确的规范, 使客户端能够从 payload 中获取消息的类别、发送者的信息以及载荷域的内容格式等相关信息. 由此可见, 使用 MQTT 协议实现即时通信的应用关键点在于话题的设计, 消息的发布、订阅流程以及消息路由. 下面将对即时消息的上述问题进行设计和实现.

3.1 话题格式设计

话题格式为: /namespace/im/u/uid.

在 MQTT 协议中话题至少需要一个字符的长度, 话题对大小写敏感的. 通过“/”用来区分相应的话题层级. 规定, 以“/”开头的话题为普通话题, 以“\$”开头的话题为系统话题^[9]. namespace 为用户所在的组织名称, 可以为企业或学校的标识名. 中间部分用来表示话题类别, im 表示即时消息话题. 最后的 uid 代表用户的唯一标示符.

3.2 Payload 设计

对于一条即时消息, 消息的接收方在接收到消息之后, 需要知道该消息的发送时间、消息的类型、消

息内容的类型、消息内容以及该消息的发送者等信息。因此该消息的 payload 的设计如表 3 所示。

表 3 即时消息的 payload 设计

Name	Sender	TS	Type	Con-len	Content
Byte	Sen-len	4	1	4	Con-len

Sender: 发送者消息, 变长, 可为空。

TS: 发送时的时间戳, 4 个字节, 不可为空。

在 Type 中包括发送消息的类型(高 4 位)以及消息内容的类型(低 4 位)。消息的类型, 现阶段只用三种表示, 具体是 0 表示是点对点的即时消息, 1 表示 PC 端到手机端的即时消息, 2 表示的是手机到 PC 端的即时消息, 其余 13 种保留用于今后扩展; 消息内容的类型, 可表示 16 种媒体类型, 具体的示例定义如下: 0 表示普通文本信息, 1 表示图片消息, 2 表示音频文件, 3 表示视频文件, 4 表示 PDF 文档, 5 表示 Word 文档等。这里需要注意的是, 在 Payload 中一次只携带一种媒体文件。采用离线文件传输的方式, 不在 payload 中进行文件传输, 文件上传结束后将对应的下载链接地址在 payload 进行传输。Type 的详细设计如表 4。

表 4 Type 详细设计

Type				Media Count			
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

Con-len: 4 个字节, 表示后面携带的媒体内容的长度。

Content: 是将消息内容按照 json 格式组织的字符串。该字符串默认情况下按照 json 数组的形式来组织的, 这样就允许在消息中支持多种媒体并存, 可以构建较为复杂的应用时满足对应的需求。对于每种媒体, 按照媒体类型不同, 提供不同的描述字段。具体实现如下:

```

"con":[
  {"ty": 0, // 0:文本,1:图片, 2:音频, 3:视频, 4:文档文件, 5:文件, 6:地图, 14 :富媒体
  "co": "url 或者文本", //媒体的内容
  **Optional**
  "fn": {
    "name": "text.txt",
    "size": "500"
  }
}
] // 可以并列多个消息内容
]

```

在上述定义中,

ty: 是媒体的类型, 目前需要考虑支持的媒体类型包括: 文本, 图片, 音频, 视频, 文档, 文件, 地图位置, html 代码, 以及其他富媒体, 如通知、公告、轻应用数据等。

co: 是媒体的内容。

fn: 如果媒体涉及到文件或者文档名字, 这里给出的是文件。

3.3 预订阅设计和实现

IM 是通过话题来组织的, 而话题的订阅一般由客户端完成。当所需订阅的话题过多时, 由于客户端订阅数据量庞大, 必定给网络带来巨大压力。其次, 对于移动设备而言, 因数据流量有限, 势必将造成不必要的浪费。因此, 为了减轻移动客户端压力, 提高用户体验, 本文采用预订阅设计方案, 即在服务器端构造一个预订阅的客户端。流程如图 2 所示。

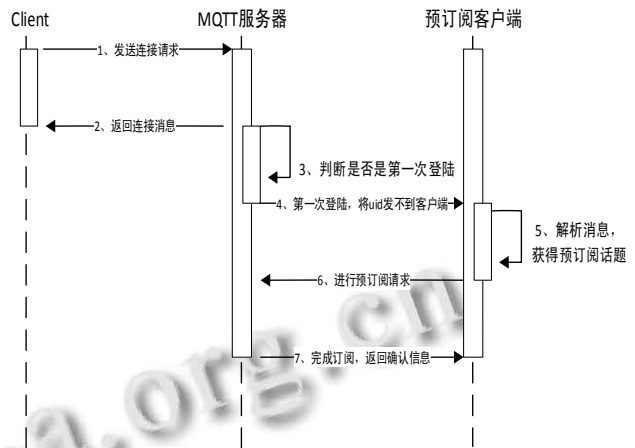


图 2 预订阅序列图

预订阅模块帮助用户完成订阅话题工作的具体流程:

- 1) 当服务器检测到客户端是首次连接时, 就会将 uid 及其他相关信息推送到服务器中的某一话题。
- 2) 预订阅模块接收到服务器发送的消息后, 解析此消息, 获得新连接用户的 uid 及该用户的预订阅话题列表。根据订阅列表, 预订阅模块向代理服务器进行订阅请求。
- 3) 通过代理服务器的接口实现预订阅模块中话题的预订阅和取消预订阅。完成订阅后, 代理服务器向预订阅模块返回确认消息。

3.4 消息路由设计实现

本文要求实现多终端 IM 消息同步. 对于接收方而言, 由于同时订阅了自己的 IM topic, 所以收到消息后可以实现各终端间的同步. 但对于发送方而言, 由于在消息发送时各终端没有联系, 导致发送方的各终端间无法同步^[10]. 为此, 本文提出了一套通过在代理服务器上增加消息路由机制, 实现的基于消息路由规则的多终端消息同步方法.

发送到一话题上的消息需要根据消息路由规则, 判断是否同时复制到其他话题上. 该规则采用话题的模糊匹配, 对于匹配成功的消息, 将其转发至特定话题上. 其中一条消息路由规则的基本组成部分为三部分, 源地址(src Topic)、消息路由动作(action)、目的地址(dest Topic). 消息路由的详细过程如图 3 所示.

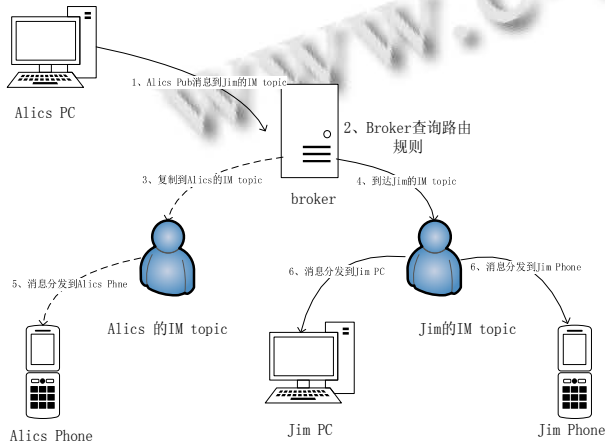


图 3 消息路由示例

具体过程说明如下:

- ① Alice 的 PC 端向 Jim 的 IM topic 上发送一条消息
- ② Broker 收到消息后, 经过路由规则查询, 判定该消息是否有匹配的路由规则, 如果存在则继续进行, 否则忽略③和⑤
- ③ 根据规则中的消息路由动作进行消息路由, 找到路由转发的目的 topic, 图中为 Alice 的 IM topic.
- ④ 把 IM 消息 Pub 给 Jim
- ⑤ Alice PC 端发送的消息在其他终端得到接收, 并呈现
- ⑥ 消息分发到 Jim 的所有终端

4 即时消息业务的测试

4.1 功能测试

表 5 测试环境参数

CPU/主频/核数	Dell R320 工作站/2.5GHz/8 核
内存	1T
系统	Cent OS 6.4(X64)
网卡	100Mbps

本文考虑到平台间兼容性问题, 在该功能测试部分采用了以下环境: 1 台 Android 手机, 1 台 iPhone5s 手机, 1 台 MQTT 服务器(图 4 所示), 20Mbps 网络带宽.

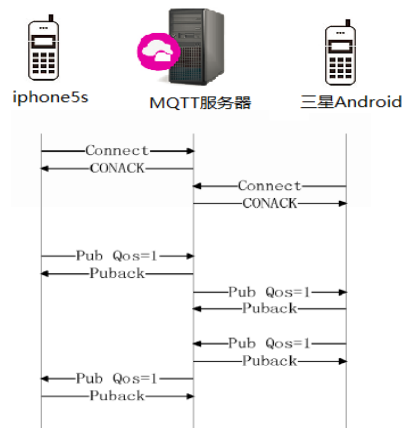


图 4 消息时序图

IOS 用户和 Android 用户之间通过 MQTT 服务器, 分别发送文字、语音、图片、视频等不同类型的媒体消息, 消息发送流程如图 4 所示, 呈现效果如图 5 所示.

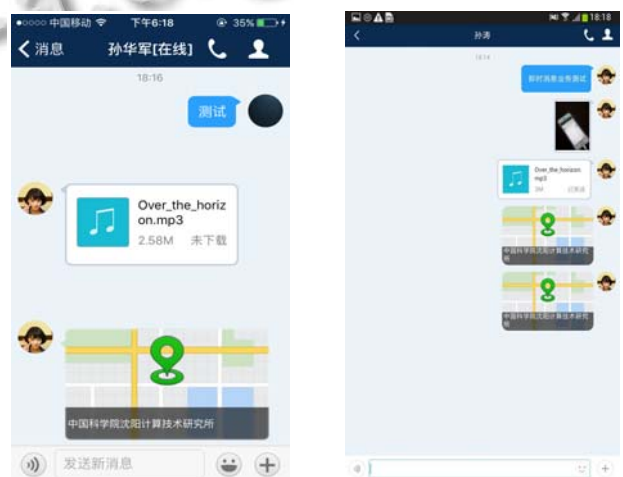


图 5 IM 效果图

测试结果表明, 各种类型媒体消息发送均能即时

显示.

4.2 IM 流量测试

为了测试 MQTT 在移动互联网中的性能优势,笔者在测试过程中分别对 SIMPLE、XMPP、MQTT 三种协议进行了封装,并使用 Wireshark 进行了抓包,对同网络环境下三种协议所消耗的网络流量进行了对比,结果如图 6 所示.

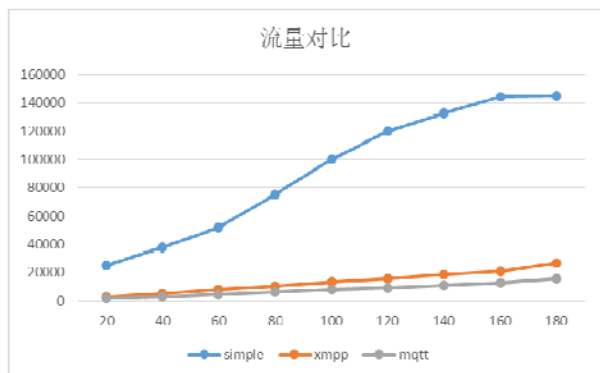


图 6 三种协议消耗流量对比图

从上图可以看出,随着发送消息文本数据包(横坐标)的增加,三种协议消耗的流量(纵坐标)均在不停地增加.其中, MQTT 协议的增幅最小, XMPP 协议次之,但增幅也不大, SIMPLE 协议增幅最大.由以上试验数据可以看出,在流量消耗这方面 MQTT 协议的性能最好.

4.3 服务器性能测试

最后,对服务器负载能力进行测试.在客户端模拟大规模用户对服务器发起连接请求,分别模拟 10000 到 100000 个连接请求,所有请求的连接建立完成后,分别进行不同数目的消息推送,测试所用的时间以及服务器 CPU 的占用率,测试结果如表 6 所示.

表 6 服务器负载性能测试

连接数	CPU 占用率(%)	1 到 100 推送	1 到 1000 推送	全部推送
10000	14.5	0.091	0.114	13.213
20000	25.0	0.105	0.225	27.167
30000	33.2	0.287	0.434	42.254
40000	40.1	0.397	0.667	68.887
50000	45.1	0.487	0.976	110.997
75000	55.3	0.639	1.213	207.876
100000	62.4	0.855	1.433	313.876

由表中数据可以看出,本文设计的即时消息业务基本满足了在移动互联领域上的需求,且服务器的负载也在可以接受的范围内.

5 结语

即时消息业务已经发展成为人们日常沟通的重要方式之一,现有的即时通信协议不能很好地满足移动互联网环境下的网络不稳定、流量花费高以及移动设备的低功耗等特点.采用基于发布/订阅模型的轻量级消息传输的 MQTT 协议,具有低功耗和移动互联网带宽利用率高的特点.本文在分析了相关私有协议的基础上阐释了 MQTT 协议在移动互联网即时消息业务中的优势,研究分析了 MQTT 协议的消息格式,并对即时消息的话题格式、预订阅以及消息路由进行了设计和实现.最后对实现的即时消息业务进行了相关的测试和分析.

参考文献

- 1 Banks A, Gupta R. OASIS Standard MQTT Version 3.1.1. <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html>. [2014-10-29].
- 2 Tang K, Wang Y, Liu H, et al. Design and implementation of push notification system based on the MQTT protocol. Proc. International Conference on Informationence & Computer Applications. 2013. 92. 116-119.
- 3 Day M, Rosenberg J, Sugano H. A model for presence and instant messaging. IETF, 2000, 2: RFC2778.
- 4 Extensible Messaging and Presence Protocol <http://xmpp.org/>.
- 5 IBM. MQTT Telemetry Transport. <http://msq.org>. [2013-06-05].
- 6 IBM, Eurotech. MQTT3.1 Protocol Specification. <http://public.dhe.ibm.com/software/dw/webservices/ws-mqtt/mqtt-v3r1.html>. [2010-08-24].
- 7 Lee S, Kim H, Hong D, Ju H. Correlation analysis of MQTT loss and delay according to Qos level. Information Networking (ICOIN). Bangkok. 2013.
- 8 马跃,孙翱,贾军营,孙建伟,于碧辉,杨雪华. MQTT 协议在移动互联网即时通信中的应用. 计算机系统应用, 2016, 25(3): 170-176.
- 9 杨海波,王默涵,贾正锋,卜立平. 面向移动互联网的 Presence/IM 机制研究. 小型微型计算机系统, 2015, 36(11): 2549-2553.
- 10 任亨. 基于 MQTT 协议的消息推送服务器. 计算机系统应用, 2014, 23(3): 77-82.