

基于 MQTT 的企业消息推送系统^①

盖荣丽^{1,2,3}, 钱玉磊^{1,2}, 李鸿彬², 贾军营²

¹(中国科学院大学, 北京 100049)

²(中国科学院 沈阳计算技术研究所, 沈阳 110168)

³(大连大学, 大连 116622)

摘要: 消息推送系统在实现企业办公自动化方面有着重要的作用, 通信协议和系统架构的设计是企业消息推送系统性能优化的关键因素. 传统的标准化协议方案在移动互联网领域的应用存在着实现复杂和流量消耗大等缺点. 针对这一问题, 结合企业消息推送系统的特点, 采用物联网领域的轻量级协议 MQTT 来设计企业消息推送系统的通信协议, 通过设计 MQTT 话题区分社交通知、即时消息和状态呈现等功能, 在满足了应用需求的同时保证了协议的简单易实现和流量消耗小等适用于移动互联网的优势. 另一方面, 在传统的对称和非对称设计的服务器架构之上, 提出了信息上非对称和功能上对称的系统架构, 为企业消息推送系统的构建提供了一个详细的解决方案. 该方案在满足系统设计需求的同时提高了系统的可扩展性.

关键词: 消息推送; MQTT; 通信协议; 系统架构; 即时通信

Enterprise Push Notification System Based on MQTT

GAI Rong-Li^{1,2,3}, QIAN Yu-Lei^{1,2}, LI Hong-Bin², JIA Jun-Ying²

¹(University of Chinese Academy of Sciences, Beijing 100049, China)

²(Shenyang Institute of Computing Technology, Chinese Academy of Sciences, Shenyang 110168, China)

³(University of Dalian, Dalian 116622, China)

Abstract: Push notification system plays an important role in the implement of enterprise office automation. And communication protocol and system architecture have become the main factors in the performance optimize of enterprise push notification system. Traditional standardized protocol scheme has encountered problems such as too complex to implement and huge traffic cost in mobile internet. This paper, for the stated problem and the features of enterprise push notification system, designed a communication protocol based on MQTT, a lightweight protocol in Internet of Things, and partitioned social network notification, instant messaging and presence by the design of MQTT topics. The new designed protocol is easy to implement and small traffic cost. On the other hand, this paper, based on the traditional symmetric and asymmetric system architecture, proposed a system architecture which is asymmetric on information and symmetric on function, by which supplied a detailed solution for push notification system for enterprise. This solution can meet the requirements of system design as well as improve the scalability of the system.

Key words: push notification; MQTT; communication protocol; system architecture; instant messaging

1 概述

企业消息推送系统是企业办公自动化定制的消息推送服务, 其核心功能包括企业内部社交平台、企业即时消息和状态呈现、企业内部通讯录和上下游通讯录等. 随着移动互联网的兴起, 企业消息推送系统

也从传统的 PC 端发展成为跨平台的多终端(手机端和 PC 端)系统. 在这个过程中, 移动互联网相对于传统的互联网的特点以及移动终端相对于 PC 端的性能状况给企业消息推送系统的设计提出了新的问题和挑战. 其中, 最显著的问题是消息推送方案的制定.

① 收稿时间:2015-03-26;收到修改稿时间:2015-05-07

当前主要的推送方案包括 GCM(Google Cloud Messaging)服务^[1-3]、XMPP(The Extensible Messaging and Presence Protocol)^[4,5]、MQTT(Message Queuing Telemetry Transport)^[2,6]以及使用 HTTP 轮询方式。HTTP 轮询方式简单易实现,但是实时性和可扩展性都较差。XMPP 是 IETF 标准化的通信协议,其功能强大、可扩展性强等特性使得其在互联网领域有着广泛应用。但是协议复杂和信息冗余使得其不适合移动互联网的应用特点。GCM 是 Google 公司推出的云消息服务,在能源消耗方面较 MQTT 有一定的优势,但是受限于网络环境,不适于构建企业消息推送系统。而 MQTT 协议在硬件限制较低、低带宽、低延时和推送速度等方面具有明显的优越性^[2]。但是, MQTT 作为物联网领域的协议,本身实现过于简单。在应用到具体的场景中时,需要进行基于 MQTT 协议的二次开发。

本文在 MQTT 协议的基础上设计企业信息推送系统的通信协议,通过对 MQTT 话题的设计实现企业社交通知、即时消息和状态呈现等功能。最后,通过对服务器架构的研究和设计,采用信息上对称和功能上非对称的服务器架构,为企业信息推送系统的设计与实现提供了一个详细的解决方案。同时,企业信息推送系统作为信息推送系统中对性能和安全性要求较高的系统,研究其实现方法也为消息推送系统的实现技术提供了一个有价值的参考。

2 相关工作

消息推送(information push)是一种有别于消息拉取(information pull)的通信方式。相较于后者的用户驱动(user-driven)模式,消息推送的事件驱动(event-driven)模式在消息的实时和高效方面体现出了更大的优势。按照消息的定制和发布方式,消息推送模型可以分为三种^[7]:

(1) Blanket Push: 客户端需要向服务器频繁发送请求,服务器根据客户端的请求进行推送。模型的优点是可以从不同的信息源获取消息。缺点是更多的带宽消耗、低聚度的消息、较少的定制信息以及通信不安全因素。

(2) Filter Push: 客户端选择消息通道(channel),服务器根据客户端的选择将通道内的消息全部推送到客户端。模型的优点是传输的是定制化的消息且低带宽消耗。缺点是需要服务器端更多的管理操作。

(3) Publish/Subscribe Push: 客户端向服务器注册感兴趣的事件(event),发布者发布事件,服务器根据事件的匹配结果向客户端推送相应的消息。模型的优点是定制化的消息和低带宽消耗。缺点是未经验证的可扩展性、服务器端的管理需求和低安全性。

本文研究的消息推送模型以 Publish/Subscribe Push 模型为主,即以 MQTT 协议为主要的通信构件,同时通过服务器上架构上的针对性设计来提高推送模型的可扩展性和安全性。

MQTT 协议由 IBM 在 1999 年设计和提出^[8],其设计目的主要是为大量计算能力有限,且工作在低带宽、不可靠网络的远程传感器和控制设备提供一种开放、精简、轻量级和容易实现的通信方法。其主要特性包括对负载内容屏蔽的消息传输、三种消息发布服务质量和低流量传输等^[9]。自提出以来, MQTT 已经广泛的应用于手机和无线传感器等计算和存储资源受限的环境中,并被医疗机构、石油和天然气公司、大学和互联网公司分别用于科研和工程产品中^[6,10-13]。MQTT 在确保和加速消息推送、节约客户端带宽和电量资源等方面表现出了良好的性能^[12,13]。

在即时通信协议方面,当前主流的方案分为企业私有协议和标准草案两类。前者包括 YMSG (Yahoo! Messenger)、AIM (AOL Instant Messenger)、MSN (Microsoft Messenger)和 QQ 等。后者主要是 IETF 标准化的 XMPP 和 SIMPLE。文献[10]指出 MQTT 是以数据为中心的。其对内容的不可知及二进制数据传输这一特性使得 MQTT 非常适用于那种需要频繁的发送少量更新数据的应用。考虑到手机终端的计算资源有限,本文采用 MQTT 协议来搭建企业信息推送系统。

3 通信协议

通信协议主要分为基于文本的和基于二进制数据的。基于文本的数据便于阅读和解析,基于二进制的二进制数据节省传输流量。对于手机等计算资源有限的终端而言,基于二进制的协议是一种更为理想的选择。MQTT 协议是基于 TCP 的二进制传输协议,本身有一定的传输格式。为了满足企业推送系统的需求,需要在 MQTT 协议的基础上设计更私有化的推送协议。

3.1 发布/订阅模型

发布/订阅(Pub/Sub)模型是一种随着物联网的兴起已经得到广泛应用的通信范式。与传统的通信范式

比较,发布/订阅模型的一个显著的共同点是实现了通信过程和通信实体的三个方面的解耦合^[14]:

(1) 空间解耦(space decoupling): 通信双方不需要知道彼此的存在,也不需要知道双方的 IP 地址、端口号等地址信息.

(2) 时间解耦(time decoupling): 通信双方的生命周期可以不一样长,因此双方可以不建立完整的和直达的通信信道.

(3) 同步解耦(synchronization decoupling): 发布和接受的双方在处理消息时并不阻塞各自的进程.

三个方面的解耦合使得发布/订阅模型的通信协议非常适合构建推送系统,尤其适应计算资源受限的手机等智能终端.

发布/订阅通信范式所提供的完全解耦合交互模型完美地适用于如今大规模、动态、点对点应用的互操作性.在系统架构方面,绝大多数发布/订阅范式的实现都比较容易地通过一定的路由策略扩展成大规模的、高可靠性和简单易用的集群架构.这一特性使得最近几年,发布/订阅系统已经成为实现分布式系统群组通信的热门方案^[15,16].

3.2 话题设计

在构建包含即时消息应用的企业信息推送系统时,通信协议的主要功能模块分为通知(Notify)、状态呈现(Presence)和即时通信(Instant Messaging).其中,通知是构成企业社交系统的核心部件,承担了企业内部通知和企业公共账号发布等功能.状态呈现和即时通信是即时通信协议的两个主要部分,标志着即时通信在用户信息管理和即时消息交换两方面的主体功能^[17].为了减轻客户端协议解析的负担,以上三个功能模块分别由 MQTT 三类话题来控制 and 区别,其主要结构如图 1 所示.

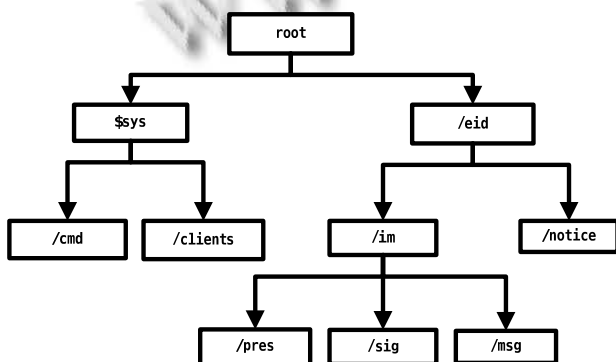


图 1 话题树

图 1 中, \$sys 节点下的话题属于系统通知话题,这一功能由 mosquito 提供. Eid 表示企业编号, /eid 节点对应的子树是相应企业推送系统的话题. 其中 /im 节点对应的子树实现即时通信和状态呈现的功能. /pres 和 /sig 节点分别对应状态呈现中的状态和签名. /msg 节点对应即时通信过程中的话题. /notice 节点对应的子树实现其他社交平台推送功能.

把状态跟消息区别开来的好处在于状态是一种无后效性的消息,即当前状态之前的状态对好友端的状态呈现没有使用价值. 这一特性可以根据 mosquito 的 RETAIN 标志位来实现.

话题树的实现方案突出了按照消息类型进行聚合的优点. 比如, /eid 为企业节点, 将 /eid 作为顶层节点时便于针对企业数量的不同设计 broker 的互联方式. 将 /pres、/sig、/msg 聚合在 /im 节点下, 也有助于在未来的设计中实现 IM 功能的分离, 为 IM 专属服务器的消息路由提供了基础.

3.3 消息格式

即时通信消息体的设计如图 2 所示, 各个字段所表示的含义见表 1.

Version (1 bytes)	SeqNumber (2 bytes)	Type (1 bytes)	TimeStamp (4 bytes)
Source (4 bytes)		DataLength (4 bytes)	
Data (variable length)			

图 2 即时消息格式

DirType (4 bit)	ContentType (4 bit)
--------------------	------------------------

图 3 消息类型字段格式

DataLength 是数据域的数据长度(字节数). Data 是数据域, 长度可变, 可以传输无格式的二进制数据, 也可以传输序列化的对象.

一般的 IM 消息格式中需要同时携带消息接收者和发布者的信息. 考虑到消息接收者事实上表明的是消息的路由方式, 所以本文在设计消息格式时不设置消息接收者字段, 而将消息接收者的信息放在话题中体现. 这种设计方式充分利用了 MQTT 的特性, 同时节省了消息传输的流量.

表1 即时消息各字段含义

定义域	注释
Version	版本号, 包括本协议的版本号和 MQTT 的版本号
SeqNumber	消息的序列号
Type	消息类型
TimeStamp	时间戳
Source	发送方 ID

Type 字段区分消息的方向类型(DirType)和消息的内容类型(ContentType), 格式如图3所示. 方向类型占4个比特, 表示消息的传递介质, 主要包括: 点对点的消息、PC到手机的消息、手机到PC的消息和群组消息. 内容类型是占4个比特, 分别标记消息内容为: 纯文本、音频、视频、超链接、图片、文件以及用户自定义数据类型等. 由于MQTT不允许消息分片, 所以在传输大文件(比如超过3M的文件)时, 为了节省客户端流量及简化传输过程, 首先使用HTTP协议将完整的文件上传至一个文件传输服务器, 然后将该文件在服务器中的资源定位符通过IM消息(消息的内容类型标记为“超链接”)发送到接收端. 接收端接收到该类型的消息时, 做流量预警提示, 然后由用户决定是否下载该文件.

4 系统架构

4.1 对称体系和非对称体系

IM服务提供者和协议设计者面临的一个基本的问题是系统如何处理大量用户的情况. 在系统架构方面, 两种主要的解决方案分别是对称的(symmetrical)服务器集群架构和非对称的(asymmetrical)服务器集群架构. AIM和MSN使用非对称的服务器集群架构, 其特点是不同的服务器实现不同的系统功能. YMSG使用对称的服务器集群架构, 其特点是每一台服务器都执行相同的功能, 用户在登陆和使用时不区分不同的服务器. YMSG在对称的服务器架构之外, 使用通过中央服务器的重定向功能将语音聊天请求重定向到特殊的语音聊天服务器^[18]. 可见, 即使是对称的服务器架构依然有对称/非对称功能设计的需求.

4.2 静态信息和动态信息

即时通信协议中的用户信息分为静态信息(static information)和动态信息(dynamic information). 其中, 静态信息主要定义为在用户建立连接后不会发生变化的信息. 静态信息主要是在登陆过程为用户所请求或发送, 其内容包括用户账号密码、好友关系、用户头

像等详细信息和用户所属企业组织架构等信息. 动态信息定义为在用户建立连接后实时传输的信息. 这种信息主要包括即时聊天信息、用户的登陆状态和签名等信息^[17]. 静态信息和动态信息之间的关系并不是绝对的. 用户在登陆之后的操作既可能涉及到动态信息也可能涉及到静态信息.

对通信过程中的静态信息和动态信息的划分有助于厘清系统设计过程中的消息流向. 静态信息作为不会发生变化或很少发生变化的信息, 往往是相对较大的信息, 比如用户的通讯录列表或者用户头像. 用户的通讯录列表一般作为一种结构化的有组织的数据, 存在着大量的冗余字段, 比如XML格式中的标签, 或者Json格式中的“key”值. 而用户头像一般是图片等比较消耗带宽资源的数据. 此类数据的加载分为两种情况: 一种是本地存储, 服务器通知更新; 一种是初次登陆, 数据全部加载. 无论哪种情况, 都体现出了静态信息的低传输频率和较大数据量的单次传输特性. 因此, 在设计的过程中, 可以将静态信息进行聚合, 采用可以实现分片和多次批量传输的协议. 而对服务器的设计 requirements 是 I/O 密集型应用.

动态信息作为经常发生变化的信息, 往往是即时通信过程中的即时消息等信息. 这类信息显著的特点是消息的数据量比较小, 但是传输频率比较高, 对及时性的要求也比较高. 同时这类信息往往携带了消息转发所需要路由信息. 因此, 针对动态信息的设计应当是满足CPU密集型的应用.

4.3 系统架构设计与实现

系统架构的设计主要是为了提高消息推送系统的消息吞吐量和并发性. 本文在设计服务器架构时, 首先根据静态信息和动态信息的区分将服务器信息功能的子模块主要分为静态模块和动态模块. 其中, 静态模块负责处理用户的静态信息, 包括账号系统、用户关系系统及用户操作日志等. 而动态模块是用户之间建立通信关系的主要模块, 负责用户即时消息的存储和转发. 为了提高系统的适应性, 本文将不同的信息功能模块部署在不同的服务器上, 构成如图4所示的信息上非对称的服务器架构. 同时, 在各种信息内部, 使用功能上对称的服务器架构. 这样架构的一个显著的优点在于使得系统的信息流简单明确, 同时服务器可以在各种信息类型内部做功能扩展, 比如通过一定的中央路由机制可以在不改变静态服务器的同时将动

态服务器扩展成集群架构. 反之亦然.

静态服务器主要由通信过程中静态模块的功能部件构成, 动态服务器由动态模块的功能部件构成. 静态服务器和动态服务器之间通过控制单元和消息总线来实现一些内部的操作和通信. 操作的内容主要包括:

(1) 话题控制. 根据用户通讯录(静态服务器)控制用户话题(动态服务器), 包括话题的创建和删除以及话题的预订阅等.

(2) 系统通知的生成. 根据用户静态控制操作(静态服务器)生成系统通知消息(动态服务器).

(3) 日志维护. 日志维护主要为系统管理和数据分析提供便捷, 将日志维护的压力从静态和动态服务器转移到控制单元也有助于提高静态和动态服务器的吞吐量.

客户端在登陆的过程中, 首先要登陆静态服务器, 如图 4 流程①所示. 此时, 使用的账号是用户注册使用的账号. 登陆成功后, 从静态服务器获取通讯录及其加密密钥等个人信息, 然后获取动态服务器的动态登陆口令. 口令获取成功后, 客户端登陆动态服务器, 验证成功之后即可以正常使用即时通信功能, 如图 4 流程②所示. 客户端与静态服务器的通信过程主要采用 webservice 接口和 json 格式的数据, 与动态服务器的通信过程使用 MQTTv3.1 自带的 TLS/SSL 加密功能, 数据格式为自定义的二进制协议.

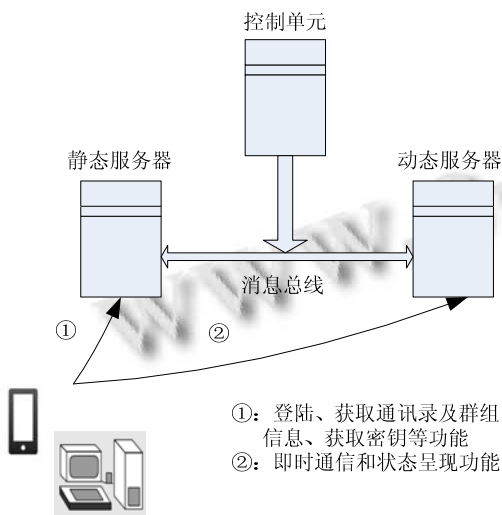


图 4 系统架构图

此外, 为了减缓静态服务器在客户端登录时承受的瞬时数据同步压力, 可以将用户通讯录等信息加密存储在客户端, 客户端在登陆静态服务器时, 如果可以

正常加载本地存储的通讯录, 则只向服务器发送通讯录更新请求. 否则, 向服务器请求全部的通讯录信息.

5 实验

表 2 服务器端配置

分类	配置
硬件	Intel(R) Pentium(R) 4 3GHz, 2G RAM, 100Mbps
软件	CentOS release 5.5(final), mosquito-1.3.1

5.1 功能测试

功能测试主要测试 IM 的会话效果和状态呈现的效果. PC 端与手机端(Android)IM 会话的效果图和状态呈现(Presence)效果图分别如图 5 图 6 所示. 其中, 一些关键的分组信息和用户名称做了马赛克处理.

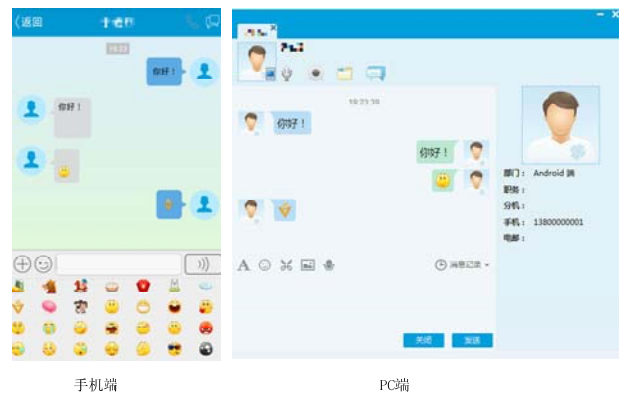


图 5 IM 测试效果图

从图 5 可以看出, 所设计的话题和协议能够满足正常的 IM 通信需求. 所设计的协议在满足企业消息推送系统的基本需求同时, 具有简单易扩展低流量消耗等特点, 适用于移动互联网领域.



图 6 Presence 测试效果图

手机端显示“在线”两个字, PC端做头像亮化,同时显示用户的签名信息.从图6可以看出,所设计的话题可以实现正常的状态呈现的需求.

5.2 性能测试

性能测试主要测试所设计的服务器系统架构的并发性.结果如表3所示.

表3 性能测试结果

用户数(个)	丢包率(%)	延时(s)
100	0	0
1000	0	0
5000	0.30	0.02
10000	0	0.01
15000	1.10	0.2
20000	1.03	0.17
25000	3.40	1.25
30000	3.00	2.3
35000	5.83	4.4

单点架构的静态服务器和动态服务器,最佳的在线用户数是10000左右,最大可以支持20000个用户同时在线.当同时在线的用户数超过20000时,通信过程的丢包率会显著提高,同时,通信延时明显加大. Mosquitto本身的并发数要远远大于20000.分析发现,系统性能的瓶颈主要出现在静态服务器端.静态服务器承担用户登陆信息验证和通讯录加载等服务,在高并发的场景中,单点的架构很难突破带宽和计算资源的限制.所以需要在进一步的研究中将静态服务器拓展为集群或分布式架构.这一结果也进一步表明了将服务器设计为逻辑上分离的静态服务器与动态服务器的必要性.因为通过将静态服务器和动态服务器进行逻辑上的分离,有助于针对具体应用需求来分别扩展其架构.这种消息上非对称和功能上对称的系统架构有效地提高了系统的可扩展性.

6 结语

现代企业办公自动化系统以企业社交平台、企业通讯录、企业即时通信和状态呈现等消息推送服务为核心.为了适应互联网应用的环境,其消息推送系统的构建成为系统性能体验的核心因素.本文针对传统的标准化协议方案在移动互联网领域的应用存在着实现复杂和流量消耗大等缺点,在现有的消息推送方案的基础上,选择使用MQTT协议作为构建企业消息推

送系统的基本协议,同时针对发布/订阅通信范式的特点和企业消息推送系统的需求,在MQTT协议的基础上设计了新的应用层通信协议.另一方面通过MQTT话题的设计来区分社交通知、即时消息和状态呈现等功能,在满足了应用需求的同时保证了协议的简单易实现和流量消耗小等适用于移动互联网的优势.在系统架构方面,本文通过区分IM通信中的静态消息和动态消息提出了消息上非对称架构和功能上对称架构的服务器架构,在满足企业用户需求的基础上提高了系统的可扩展性.实验表明,本文提供的通信协议和服务器架构为企业消息推送系统提供了一个详细而可靠的解决方案.

参考文献

- 1 Google Cloud Messaging. Android.com. <http://developer.android.com/google/gcm/gcm.html>, 2015 Jan.
- 2 Warren I, Meads A, Srirama S, et al. Push notification mechanisms for pervasive smartphone applications. Published in Pervasive Computing, IEEE, 2014, 13(2): 61-71.
- 3 邹海,李强,邱慧丽.基于Android C2DM服务的云端推送研究与实现.计算机技术与发展,2012,22(7):30-32.
- 4 Hornsby A, Belimpasakis P, Defee I. XMPP-based wireless sensor network and its integration into the extended home environment. IEEE 13th International Symposium on Consumer Electronics (ISCE2009), 2009. 794-797.
- 5 Ji ZL, Ganchev I, O'Droma, et al. A push-notification service for use in the UCWW. 2014 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery. 2014. 318-322.
- 6 Tang K, Wang Y, Liu H, et al. Design and implementation of push notification system based on the MQTT protocol. International Conference on Information Science and Computer Applications (ISCA 2013), Sep 2013.
- 7 Li N, Du YH, Chen GX. Survey of cloud messaging push notification service. 2013 International Conference on Information Science and Cloud Computing Companion (ISCC-C). 2013. 273-279.
- 8 MQ-Telemetry Transport. "MQTT.org". <http://mqtt.org/>. 2015 Jan.
- 9 Lampkin V, et al. Building Smarter Planet Solutions with MQTT and IBM WebSphere MQ Telemetry. ibm.com/

- redbooks, 2015 Jan.
- 10 De Caro N, Colitti W, Steenhaut K, et al. Comparison of two lightweight protocols for smartphone-based sensing. *Communications and Vehicular Technology in the Benelux (SCVT)*. 2013: 1–6.
- 11 Wilson S, Frey J. The SmartLab: experimental and environmental control and monitoring of the chemistry laboratory. *Collaborative Technologies and Systems*, 2009: 85–90.
- 12 Why Facebook is using MQTT on mobile. https://www.ibm.com/developerworks/community/blogs/mobileblog/entry/why_facebook_is_using_mqtt_on_mobile?lang=en, 2013 Sep.
- 13 MQTT: born for IoT. <http://www.leiphone.com/0828-danice-mqtt.html>, 2012 Aug.
- 14 Eugster P, Felber P. The many faces of publish/subscribe. *ACM Computing Surveys*, 2003, 35(2): 114–131.
- 15 Baldoni R, Beraldi R, Quema V, et al. TERA topic-based event routing for peer-to-peer architectures. *Proc. of the 2007 Inaugural International Conference on Distributed Event-based Systems*. 2007. 2–13.
- 16 Mayer T, Brunie L, Coquil D, et al. On reliability in publish/subscribe systems: a survey. *Int. J. of Parallel, Emergent and Distributed Systems*, 2012, 27(5): 369–386.
- 17 路林. 即时通信协议的特征与通联关系分析[学位论文], 解放军信息工程大学, 2012.
- 18 Jennings RB, Nahum EM, Olhefski DP, et al. A study of Internet instant messaging and chat protocols. *IEEE Network*, 2006, 20(4): 16–21.