

基于 MQTT 协议 IM 的研究和实现^①

贾军营, 王月鹏, 王少华

(中国科学院大学, 北京 100049)

(中国科学院沈阳计算技术研究所, 沈阳 110168)

摘要: 随着移动互联网和智能终端的发展与普及, IM(Instant messaging 即时通讯)再次成为一个炙手可热的领域. 由于移动终端设备在低功耗、低带宽等方面的限制, 对即时通讯协议提出了更高的要求. MQTT(Message Queuing Telemetry Transport, 消息队列遥测传输)作为一种轻量级基于发布/订阅模式的消息传输协议, 是移动终端应用的理想选择. 介绍了 MQTT 的基本内容和特点, 并与其他即时通讯协议 SIMPLE 协议和 XMPP 协议进行了对比, 从而提出一种基于 MQTT 协议的移动终端即时通讯解决方案. 在 Mosquitto 开源项目的基础上, 设计并实现了一款基于 MQTT 协议的移动即时通讯客户端, 实现了即时通信、状态呈现和群组等功能.

关键词: MQTT; 移动终端; 发布/订阅; 即时通信

Research and Implementation of IM Based on MQTT Protocol

JIA Jun-Ying, WANG Yue-Peng, WANG Shao-Hua

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

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

Abstract: With the development and popularity of the mobile Internet and intelligent terminal, Instant Messaging once again become a hot field. However. Due to limitations of the mobile terminal device in a low power, low bandwidth and other aspects, it calls for higher requirements on Instant Messaging protocol. MQTT(Message Queuing Telemetry Transport) was designed as an extremely lightweight publish/subscribe messaging transport, and it is ideal for mobile applications. This paper introduced the basic content and features of MQTT, and gave a comparison with other instant messaging protocols such as SIMPLE and XMPP, then proposed a instant messaging solution of mobile terminal based on MQTT. Based on open source MQTT Broker Mosquitto, this paper also designed and implemented a instant messaging client for mobile terminal, and this client achieved instant message,presence,group functions and so on.

Key words: MQTT; mobile terminal; publish/subscribe; instant messaging

随着移动互联网的高速发展和智能终端的大量普及, 各种移动社交工具如雨后春笋充斥着市场, 大大满足了人们对于随时随地社交的需求. 而微信的流行, 证明了即时通讯在移动终端领域广阔的市场需求. 目前市场上, 即时通讯的实现除了少数采用私有协议外(如 QQ、微信)外, 主要有 SIMPLE 协议和 XMPP 协议两大类. 但是由于移动智能终端对网络带宽、流量、功耗等有特殊的限制, 移动 IM 对即时通讯协议提出了更高的要求. 而上述两种协议并不是针对移动 IM 所提出的解决方案, 没有考虑到移动智能终端的特殊

性, 使得其在移动 IM 中表现并不突出. 本文通过对当前移动 IM 解决方案进行对比和研究, 提出了一种基于 MQTT 协议的解决方案, 并对其进行了实现.

本文基于 MQTT 协议设计并实现了一个消息订阅客户端, 第一部分介绍 MQTT 相关的内容, 第二部分对多种方案的进行对比分析, 第三部分介绍了客户端系统架构, 第四部分详细描述了系统中主要模块的设计与实现, 第五部分对客户端主要模块进行了功能和性能上的测试, 最后对全文做出总结, 并对下一步的工作进行了展望.

^① 收稿时间:2014-10-25;收到修改稿时间:2014-12-05

1 MQTT协议

1.1 MQTT 协议简介

MQTT(Message Queuing Telemetry Transport,消息队列遥测传输)协议由 IBM 及 Arcom(现在的 Eurotech)公司开发,是一种轻量级基于发布/订阅模式的消息传输协议^[1]. MQTT 为计算能力有限,且工作在低带宽、不可靠的网络的设备所设计的协议.它采用小型传输,耗电量小,可大大降低网络流量,最小数据包并有效分配与传输,适合移动系统上的应用,是物联网的重要通信协议之一

1.2 MQTT 协议的消息格式

MQTT 消息体主要由三部分组成:固定报头,可变电头和有效载荷,固定报头是每个命令消息都必须包含的部分,固定头固定为 2bytes.其固定头格式如表 1 所示.

表 1 MQTT 固定头格式表

bit	7	6	5	4	3	2	1	0
byte1	Message Type			DUP flag	QoS level		RETAIN	
byte2	Remaining Length							

其中 Message Type: 消息类型,有 14 种; QoS level: 服务质量等级有 3 种, QoS 0、QoS 1、QoS 2, 等级越高,需要的系统开销越多,通信效率受影响越大; Remaining Length: 表示除固定头之外的消息长度,包含可变头部和有效载荷部分,最大可扩展到 4 字节,最大表示长度可为 256MB.

2 移动终端消息推送方案对比

目前在移动终端上已有的消息传输方案主要有以下几种: SIMPLE 协议, XMPP 协议和 MQTT 协议, 以下是对几种方案的对比分析.

2.1 SIMPLE 协议

SIMPLE(SIP for Instant Messaging and Presence Leverage Extension)^[2,3]协议是 IETF 在 SIP 协议的基础上的扩展,是移动 IM 主流协议之一. SIMPLE 增加了 SUBSCRIBE 和 NOTIFY 方法来支持交换状态信息,增加 MESSAGE 方法用于消息的发送.通过这些方法,可以满足移动 IM 的即时消息和状态服务的需求. SIMPLE 协议有着较成熟的应用基础,较成熟的音视频标准,并支持各种即时消息通信.但是它的缺点也较多,如所需流量较大、数据包较大、存在拥塞问题、

效率不高、扩展新功能比较复杂等.

2.2 XMPP 协议

XMPP(Extensible Messaging and Presence)^[4,5]协议由 IETF 制定,是符合 RFC2778 和 RFC2779 规范,一种以 XML 为基础的开放式即时通信协议.它源于 Jabber 协议,主要针对准实时、扩展即时通信、出席信息、维护联系人等网络实时通信. XMPP 协议继承了 XML 灵活的发展特性,该协议成熟、强大、可靠、安全,可扩展性强;但存在重复转发,网络流量较大的问题,并且在音频、视频方面没有非常稳定的标准,数据包也比较大,协议较复杂、冗余(基于 XML),费流量、费电,部署硬件成本较高^[6].

2.3 MQTT 协议

MQTT^[7,8]协议简单,小型传输,开销很小,协议交换最小化,降低网络流量,固定报头仅为 2 字节.其使用发布/订阅消息模式,提供一对多的消息发布,解除了应用程序耦合,对负载内容屏蔽的消息传输. MQTT 是基于 TCP 上的应用层协议,它支持基于 TLS/SSL 的加密通道传输. MQTT 在进行 connect 的时候,支持认证.另外 MQTT 还提供三种不同消息发布的服务质量^[9],分别是:

“至多一次”: 消息发布完全依赖底层 TCP/IP 网络,会发生消息丢失或重复,

“至少一次”: 确保消息到达,但消息重复可能会发生.

“只有一次”: 确保消息到达一次,可用于计费系统中,消息重复或丢失导致不正确的结果.

对比上述各个方案,我们可以发现, SIMPLE 协议和 XMPP 协议,在流量、功耗、传输方面没有考虑到移动互联网的特点,而这些特点正是移动终端应用所必需考虑的地方.反观 MQTT 协议,不仅能够全面的支持即时通讯的基本需求,也解决了在流量、功耗、传输上面的问题.首先 MQTT 流量消耗非常的小,它的固定长度仅为 2 字节.其次由于协议自身非常简单,再加上简小的头部,所以解析代价低,功耗自然就比较小.此外 MQTT 极易扩展,方便用户的二次开发.综上所述, MQTT 协议更适于移动终端,所以本文采用了基于 MQTT 协议的解决方案.

3 客户端系统架构

客户端系统架构采用分层设计,确保各层业务逻

辑的相对独立, 便于快速响应多样化的业务需求. 其整个客户端部分主要分四个层次, 第一层 OS 适配层, 负责适配各种不同的操作系统; 第二层是支撑层, 提供协议栈、业务组件和基础能力的统一通信的接口, 内部组件的交互采用标准协议完成, 保证协议栈、业务组件的独立性和可扩展性, 降低系统的耦合程度; 第三层为业务逻辑层, 实现了客户端全套业务能力, 并允许未来利用现有业务逻辑进行全新的界面效果、业务能力扩展或业务重组, 包括了即时通信, 群组, 通讯录等; 第四层 UI 表示层, 提供了基本的核心界面及窗口管理器的封装, 通过与用户进行交互来完成各种业务功能. 整个系统的架构如图 1 所示.

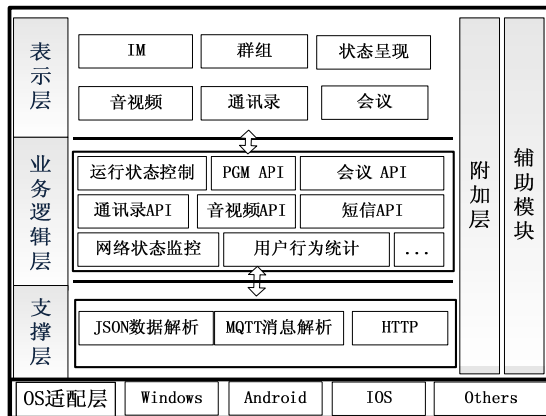


图 1 客户端系统架构图

4 主要功能模块的设计与实现

4.1 MQTT 话题订阅

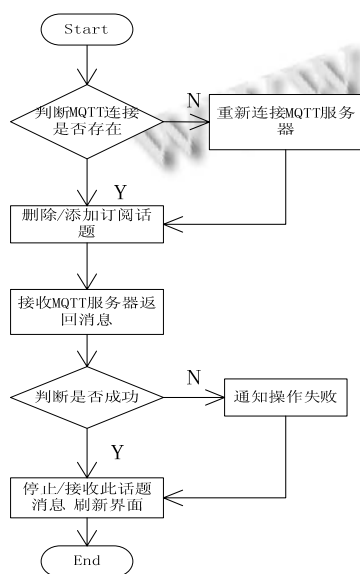


图 2 MQTT 话题订阅

移动 IM 用户会订阅各种各样的话题, 例如系统的广播通知, 好友的上线提醒, 好友发送的即时消息, 同样用户也会取消某些话题的订阅. 客户端通过 MqttClient.Subscribe() 方法向服务器订阅话题, 服务器进行相应的处理, 并将与话题相关的消息采用 JSON 的方式推送给客户端. 客户端收到后解析 JSON 数据, 刷新显示界面. MQTT 客户端订阅话题的流程图如图 2 所示.

4.2 IM 模块

IM 的功能主要包括: 与通讯录中的好友进行 IM, 多终端 IM 消息的同步, 离线终端 IM 消息的同步. 另外, IM 消息支持发送文字、图片、文件、音频等多种媒体.

4.2.1 IM 设计

每个用户拥有一个 IM 的 MQTT 中的话题, 每个用户的每个终端均订阅该 IM 话题. 在进行 IM 的时候, 发送方向自己的 MQTT 话题发送对方的用户信息, MQTT 客户端收到该订阅请求后, 为双方所有已经使用过得客户端均增加订阅话题. 当用户有新终端接入时, 通过 MQTT 客户端进行会话信息的自动预订阅. 为了降低 IM 消息交互的开销, 对于所有 IM 消息的发送采用 QoS=1, IM 对应话题的订阅采用 QoS = 1. 对于 IM 类的消息, 无须 retain.

4.2.2 Payload 设计

在 IM 类的消息中, 对于 payload 要考虑的类型比较多. 由于 IM 类消息中传输的媒体类型比较多, 例如文本、图片、音频、超链接等. 对于文件的传输采用离线文件的方式, 不在 payload 中进行传输, 上传结束后将对应的下载链接(HTTPS/SFTP)在 payload 进行传输. Payload 设计如表 2.

表 2 IM 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.

Type: 媒体种类和消息类型, bit7-bit4 表示 IM 消息的类型, bit3-bit0 表示同时携带媒体的种类, 文本为默认类型, 一次一种媒体传输, 不做复合媒体传输, 文件传输采用离线文件的方式, 不在 payload 中进行传输, 上传结束后将对应的下载链接在 payload 进行传输. 具体设计如表 3.

表 3 Type 详细设计

Type				Media Count			
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Content-length:4 个字节,表示后面携带的媒体内容的长度

Content: 消息的内容,变长,最大为 500k,结构是 JSON 字符串.

IM 发送和接收流程如图 3.

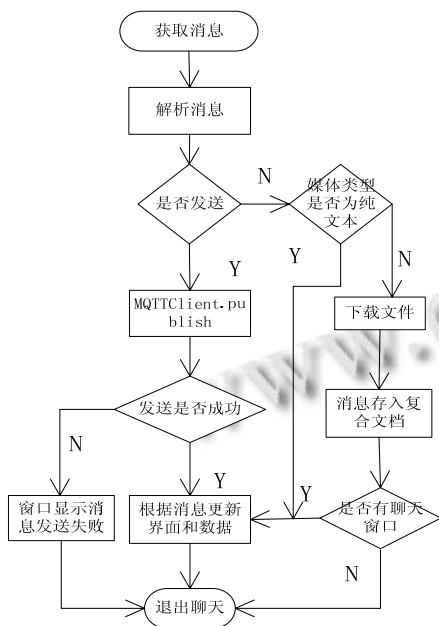


图 3 IM 消息发送与接受流程图

4.3 Presence 模块

主要是对用户的各个终端的状态进行呈现. 状态是和终端类型相关联的. 移动终端只有基本状态在线和离线. 用户通过订阅好友的 Presence 话题. 每当被订阅者的状态发生变化时, 客户端会向服务器相关话题发布消息, 服务器将消息推送给订阅者.

4.3.1 Presence 设计

每个用户均有一个关于基本状态的 MQTT 中的话题, 每个用户终端类型均需要订阅自己用户名下的状态话题, 同时也需要订阅自己通信录好友的状态话题. 只有用户在线时, 才可接收到 Presence 状态消息, 用户离线时, 不会接收到 Presence 状态消息. 如果离线订阅者在连接成功后, 可以将标记为 retain 的消息推送给订阅者. 考虑到每次上线都由客户端来对这些话题进行订阅, 不但会影响客户端的性能, 还会占用移动终端较多的网络资源, 尤其是在用户关系复杂,

需要发送大量消息订阅大量话题的时候. 因此, 客户端不需要主动订阅所有 presence 相关的话题, 其订阅过程可以通过服务器侧客户端的预订阅完成, 减少网络上的数据交互.

4.3.2 Payload 设计

Payload 由三部分组成, 占一个字节, 设计如表 4.

表 4 presence payload 设计

Type		Flag		Status			
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Type: 表示基本状态或者详细状态. 其移动客户端只有基本状态: 在线和离线.

4.4 群组模块

群组包括工作组(固定群组)和讨论组. 工作组是包括预定义群组和自定义群组. 预定义群组是由企业的管理员进行创建的, 自定义群组是由每个用户创建的.

工作组主要功能有工作组成员管理、工作基本信息维护、工作组文件共享、工作组解散与工作组转让、工作组加入.

讨论组是临时的群组, 不具有工作组的上述功能. 讨论组的功能主要有创建、退出、添加成员、移除成员等. 讨论组是工作组操作的一个子集.

4.4.1 工作组设计与实现

群组中的操作通知, 由系统通知部分采用 MQTT 话题的方式来实现, 群组成员自动订阅此话题, 有通知消息时, 自动发送给群组成员. IM 采用 MQTT 的话题的形式, 群组成员自动订阅此话题. 群组文件采用离线方式上传, 群组临时文件有时间限制, 最长为 7 天, 永久文件有大小限制, 最大为 500M. 对于文件获取, 如果文件没有其他特殊限制, 群组成员都可获取.

4.4.1 讨论组设计与实现

讨论组直接通过 MQTT 来完成, 不对其进行统一管理. 创建者创建讨论组的话题, 并发布对应的讨论组成员话题. 讨论组成员自动订阅此话题. 创建者以及所有参与者可进行多人的 IM.

5 测试

5.1 IM 流量测试对比

为了验证 MQTT 协议能更好的适应移动互联网, 将其 MQTT 协议的实现与 SIMPLE 和 XMPP 协议的实现分别进行测试, 使用 WireShark 抓取数据包, 分析数

据包, 三种协议所消耗流量如图 4 所示. 测试环境如下:

SIMPLE 测试环境:opensips、opencap、link;

XMPP 测试环境: openfire、spark;

MQTT 测试环境: 基于开源项目 mosquitto 的服务器和 android 客户端.

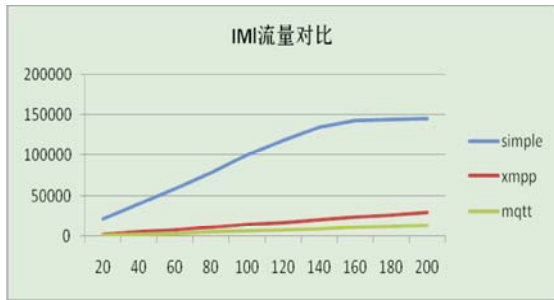


图 4 三种协议所耗流量对比图

横轴表示发送消息的条数, 每发送二十条文本消息记录数据包大小, 纵轴表示发送消息所需的流量, 单位是字节. 从图中可看出: SIMPLE 协议消耗流量最多, XMPP 协议其次, MQTT 协议消耗流量最少.

5.2 功能测试

系统测试环境: 2 台 Andriod 智能手机, 数据服务器, MQTT 服务器, 3G 网络和 wifi 网络环境, 2 台 Andriod 智能手机安装本程序, 作为功能测试的终端, 测试发送消息时序如图 5 所示.

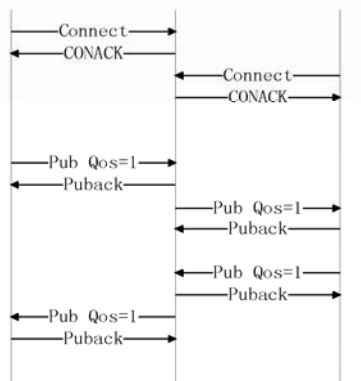


图 5 消息时序图

用户“测试”在连接 3G 网络的终端登录, 用户“无

敌”在连接 Wifi 的终端登录. 两者互为好友关系, 用户“无敌”发送不同媒体类型的即时消息给用户“测试”, 用户“测试”也以同样的方式发送消息给用户“无敌”, 这样就实现了用户之间的即时通信, 终端显示效果如图 6.

同时使用 WireShark 数据包侦听和获取发出的数据包, 部分结果如图 7 所示.



图 6 测试效果图

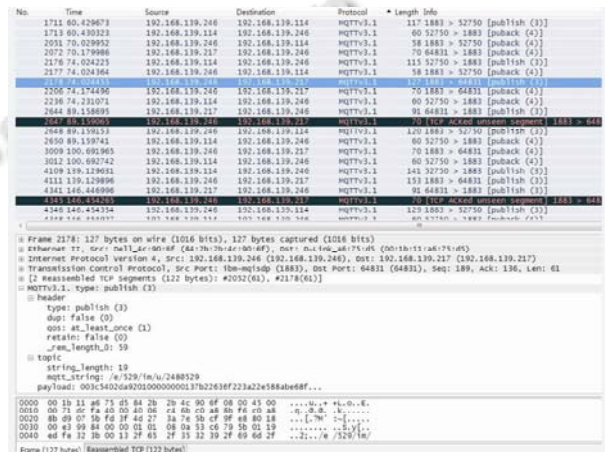


图 7 功能测试部分数据包

其中上半部分为用户测试和用户无敌之间互相发送消息的数据包, 中间部分为 MQTT 报文头头部填充内容和 topic 相关信息以及 payload, 最后部分为发送消息的内容.

6 总结

针对目前标准体系即时消息实现方案不能很好的适应移动网络低功耗、低带宽的特殊需求,本文采用基于 MQTT 协议的解决方案实现了即时通讯.文中研究了 MQTT 协议的结构、消息格式,重点介绍了各个功能实现中相关 payload 的设计.实验结果表明,不同移动终端的用户可以实时、准确的进行消息通信,并且 MQTT 协议消耗流量较小,在功耗和带宽方面能较好的适应移动互联网.本文中的即时通信应用并不完善,尤其在安全性方面考虑不足,在未来工作中将会侧重这方面的研究.

参考文献

- 1 IBM.MQTelemetryTransport[2013-06-05].http://msq.org.
- 2 Campbell B, Rosenberg J, Schulzrinne H, Huitema C, Gurle D. Session initiation protocol (SIP) extension for instant messaging. RFC, 2002.
- 3 Niemi A. Session initiation protocol (SIP) extension for event state publication. Networking & Communication Engineering, 2004, (2).
- 4 Saint-Andre P. Extensible messaging and presence protocol (XMPP): Core. University of Helsinki Department of Computer Science, 2004.
- 5 Saint-Andre P. Extensible messaging and presence protocol (XMPP): Instant messaging and presence, 2011.
- 6 潘晓丰.基于 XMPP 的企业 IM 的研究及实现[硕士学位论文].北京:中国科学院计算技术研究所,2006.
- 7 IBM, Eurotech.MQTT3. 1ProtocolSpecification. http://public.dhe.ibm.com/software/dw/webservices/ws-mqtt/mqtt-v3r1.html.[2010-08-24].
- 8 Lee S, Kim H, Hong D, Ju H. Correlation analysis of MQTT loss and delay according to Qos level. Information Networking (ICOIN). Bangkok. 2013.
- 9 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.

www.c-s-a.org.cn