

# MQTT 协议在 Android 平台上的研究与应用<sup>①</sup>

关庆余<sup>1,2</sup>, 李鸿彬<sup>2</sup>, 于波<sup>2</sup>

<sup>1</sup>(中国科学院大学 工程管理与信息技术学院, 北京 100049)

<sup>2</sup>(中国科学院沈阳计算技术研究所 网络与通信实验室, 沈阳 110168)

**摘要:** MQTT 是一种基于发布/订阅的轻量级消息传输协议, 它的出现为 Android 平台上实现消息推送提供了新的途径. 针对 Android 平台消息推送协议的需求, 研究了 MQTT 协议的结构及消息格式, 对比分析了 Android 平台上其他消息推送方案, 设计并实现了一款基于 MQTT 协议的即时通信应用, 完成了登录/注册、即时通信、好友管理等功能. 测试结果表明此应用能够满足即时通信应用的基本功能及需求.

**关键词:** MQTT; 发布/订阅; Android; 即时通信

## Research and Application of the MQTT Protocol on the Android Platform

GUAN Qing-Yu<sup>1,2</sup>, LI Hong-Bin<sup>2</sup>, YU Bo<sup>2</sup>

<sup>1</sup>(University of Chinese Academy of Science, College of Engineering and Information Technology, Beijing 100049, China)

<sup>2</sup>(Shenyang Institute of Computer Technology, Chinese Academy of Science, Networks and Communication Laboratory, Shenyang 110168, China)

**Abstract:** MQTT is a lightweight protocol based on publish/subscribe messaging transport. It provides a new way for message push on Android. For the demand of message push protocol on Android, analyzed the structure of MQTT and the format of its message, and compared with other message push solutions on the Android platform. Then, designed and implemented an instant messaging application based on the MQTT protocol. Has finished many functions such as login/register, send instant message, manage friends and so on. Finally, the test result indicated that this application is satisfy the basic functionality and requirements of the instant messaging applications.

**Key word:** MQTT; publish/subscribe; Android; instant messaging

MQTT(Message Queuing Telemetry Transport, 消息队列遥测传输)是一种轻量级基于代理的发布/订阅的消息传输协议, 其设计思想是开放、简单、轻量、易于实现, 适合在带宽、计算和处理能力受限的环境下工作<sup>[1]</sup>. 例如, 网络代价昂贵, 带宽低、不可靠的环境或者处理器和内存资源有限的环境. 这使得 MQTT 成为 M2M、物联网以及移动应用的理想选择. MQTT 技术于 1999 年, 由 IBM 的 Andy Stanford-Clark 博士以及 Arcom 公司的 Arlen Nipper 博士发明, 其支持所有平台, 几乎可以把所有联网设备与外部连接起来, 被用作传感器和驱动器的通信协议. 也被称为 SCADA 协议(Supervisory Control And Data Acquisition, 数据采集

与监视控制), 成为物联网协议的重要部分. 目前, 美国结构化信息标准发展委员会(OASIS)成立专门的 MQTT 技术委员会, 旨在负责制定基于 MQTT 技术的物联网标准协议. OASIS 专门的 MQTT 技术委员会将以 MQTT 协议为基础, 进一步研究与规范更为开放、简单、轻量的, 适合在有限网络和多平台环境中使用的 MQTT 物联网协议.

## 1 MQTT 协议研究

### 1.1 MQTT 结构

如图 1 所示, MQTT 可以分为两部分: MQTT 客户端, MQTT 消息代理<sup>[2-3]</sup>. MQTT 客户端直接使用 MQTT

① 收稿时间:2013-08-24;收到修改稿时间:2013-09-25

协议与 MQTT 消息代理相连接。

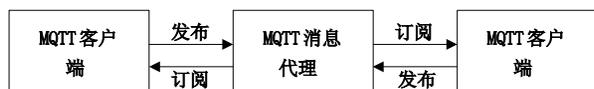


图 1 MQTT 结构图

### 1.2 MQTT 的消息格式

每个 MQTT 命令消息的消息头都包含一个固定的报头。某些消息也需要包含一个可变的报头和一个 payload。并且 MQTT 的报头固定为 2bytes，这不仅节省了开销，而且降低了网络流量的消耗<sup>[4]</sup>。

### 1.3 QoS(服务质量)

MQTT 定义了 3 个级别的 QoS。QoS 决定了服务器和客户端如何确保消息的接收。消息可以以任意级别的 QoS 发布，客户端也可以以任意级别的 QoS 订阅话题。客户端接收消息时会选择 QoS 等级最高的进行接收<sup>[5]</sup>。

## 2 Android平台上消息推送方式对比

目前在 Android 平台上比较好的推送实现方案有以下几种：C2DM(Cloud to Device Messaging, 云端推送)、MQTT 协议、XMPP 协议。下面就几种方案的优缺点进行对比。

### 2.1 C2DM

C2MD 是 Google 提供的用来帮助开发者从服务器向 Android 应用程序发送数据的服务<sup>[6]</sup>。

C2DM 的优点是 Google 提供服务、原生、简单，无需实现和部署服务器。缺点是 Android 版本限制(必须大于 2.2 版本)，并且在国内不够稳定，需要用户绑定 Google 帐号，受限于 Google。

### 2.2 MQTT 协议

MQTT 协议是由 IBM 推出的，目前技术已经比较成熟，有许多开源项目以及开源的 MQTT 消息代理，如 Mosquito 等。

MQTT 协议实现的优点是协议简洁、小巧、可扩展性强、适合二次开发省流量、省电，目前已经应用到企业领域。缺点是硬件部署成本较高。

### 2.3 XMPP 协议

XMPP 协议是基于 XML 协议的通讯协议，前身是 Jabber，目前已由 IETF 国际标准化组织完成了标准化工作。C2MD 的底层也是由 XMPP 协议实现的。

XMPP 协议实现的优点是协议成熟、强大、可扩

展性强、目前主要应用于许多聊天系统中，且已有开源的 Java 版的开发实例 Androidpn。缺点是协议较复杂、冗余(基于 XML)、费流量、费电，部署硬件成本高。

对各个方案的优缺点的研究和对比，本文使用 MQTT 协议的方案进行实现。

## 3 MQTT协议在Android平台上的应用

消息推送方案主要应用于即时通信方面。通过上节的对比分析，本文采用 MQTT 协议在 Android 平台上实现一款即时通信应用。

### 3.1 服务器端

此应用使用两个服务器，Mosquito 服务器和辅助服务器(主要用来存储用户信息及通信数据)。

Mosquito 服务器主要分三个层次，第一层是 MQTT 消息推送 broker，负责完成协议底层的网络通信机制以及消息的收发机制；第二层负责数据的处理，包括解析/分类；第三层是数据存储层，为第二层提供数据的支持，与辅助服务器进行数据的传递。

### 3.2 客户端

#### 3.2.1 客户端整体架构

整个客户端共分 3 个层次，分别是数据传输层、界面层和逻辑控制层<sup>[7]</sup>。客户端的结构图如图 2 所示。

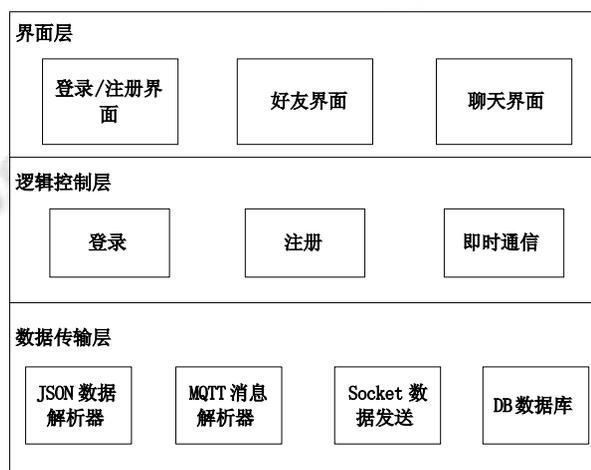


图 2 客户端结构图

#### 3.2.2 客户端设计与实现

客户端功能模块包括登录/注册模块、好友管理模块、即时通信模块、消息编解码模块、数据传输模块、数据存储等模块。

##### 3.2.2.1 登录/注册模块的设计与实现

###### ① 用户登录界面设计

用户登录界面主要包括账号输入框、密码输入框、登录按钮和注册按钮。

② 用户登录处理逻辑

用户输入账号和密码后，点击“登录按钮”进行登录，客户端处理逻辑如下：

首先，与数据存储服务器建立连接，发送账号密码进行验证，连接成功后，进行 MQTT 服务器的连接，初始化 MQTT 客户端，使用 MqttClient.connect()方法进行连接，MQTT 服务器对用户进行验证，通过后，客户端就可以与服务器进行通信了。

③ 用户注册的设计与实现

用户注册主要包括用户昵称、用户账号、用户密码以及性别的输入，这些基本信息输入后，点击注册按钮进行注册，注册成功后自动登录系统。之后的过程与登录完全一致，这里不再重复。

3.2.2.2 MQTT 话题设计

在使用 MQTT 协议进行推送的时候，要根据相应的话题(topic)进行组织。本应用采用 MQTT 协议完成好友的添加与删除以及即时通信，设计了以下话题：

- ① user/昵称/friend/new: 添加好友；
- ② user/昵称/friend/delete: 删除好友；
- ③ /昵称/chat: 即时通信的话题。

依据不同的话题，对应的 payload 存放的内容不同。Payload 定义如下：

0	64	68	70
Sender	Ts	Type	Content

Sender: 发送者信息, 64 个字节, 可为空。

Ts: 发送时戳, 4 个字节, 不可为空。

Type: payload 类型, 1 个字节。

Content: 推送消息正文。在本文的应用中，即时通信类话题的 Content 与其他不同，第一个字节表示内容中包含的媒体类型，后面的包含每种媒体的长度(占 2 个字节)与内容。

3.2.2.3 好友管理模块的设计与实现

① 用户好友管理界面设计

好友管理界面主要包括好友搜索栏、好友列表、添加/删除好友按钮，全局用户搜索按钮。

② 用户好友管理的处理逻辑

用户登录后，进入好友管理页面，在好友列表中，可以看到已经添加的全部好友的昵称和头像，在每一个好友昵称后面，有一个删除好友的按钮。

用户点击删除按钮，客户端通过 MqttClient.Subscribe()方法向服务器订阅删除好友的话题，服务器接收订阅处理并返回处理消息，MQTT 客户端解析消息，确认是否删除成功，若成功刷新好友列表。

如果用户还未添加好友，可以点击全局用户搜索按钮，输入昵称发送给服务器，服务器将相似的数据返回给客户端，客户端解析 JSON 数据，将结果显示在搜索列表上，这时在用户昵称后面会出现添加好友的按钮，点击添加按钮，客户端向服务器订阅添加好友的话题，成功则刷新当前页及好友列表。

③ MQTT 客户端订阅话题的流程

MQTT 客户端订阅话题的流程图如图 3 所示。

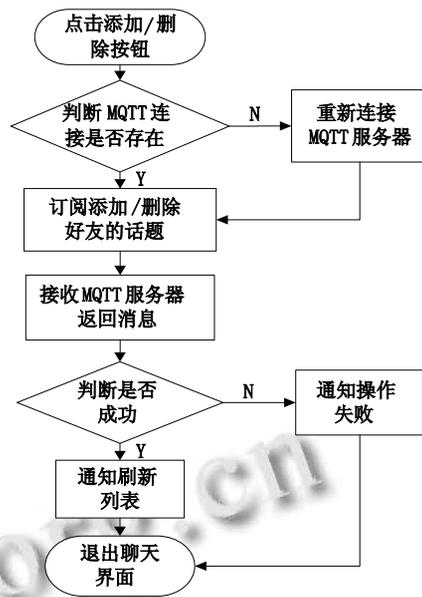


图 3 订阅话题流程图

3.2.2.4 即时通信模块的设计与实现

① 即时通信界面设计

即时通信界面主要包括消息显示列表、表情/图片选择按钮、文本/语音选择按钮以及发送按钮。

② 即时通信的处理逻辑

消息发送: 用户在 EditText 中输入想要发送的文本，然后点击发送按钮，或者选择图片、语音按钮发送语音或图片消息(语音和图片是通过 Socket 上传到数据服务器上的，数据服务器返回相应的下载链接)，客户端先将消息内容传送到消息编解码模块进行编码。编码后，通过 MqttClient.publish()方法将消息发布到即时通信的话题上，然后由 MQTT 服务器进行处理并返回消息，

MQTT 客户端接收消息, 由编解码模块进行解码, 如果发布成功, 表明 MQTT 服务器已将消息转发到目标用户, 并且客户端会刷新消息列表; 否则重新发送消息。

消息接收: 如果用户接收到其他用户发送来的消息, 首先, MQTT 客户端将接收到的消息传递给消息编解码模块, 然后, 判断消息类型, 如果是文本消息, 直接保存到本地数据库, 并将消息显示在消息列表上; 如果是语音或图片消息, 先通过下载地址下载, 并保存在 sdcard 中, 然后再显示到消息列表上。

### ③ 即时通信消息发布和接收的流程

即时通信消息发布和接收的流程图如图 4 所示。

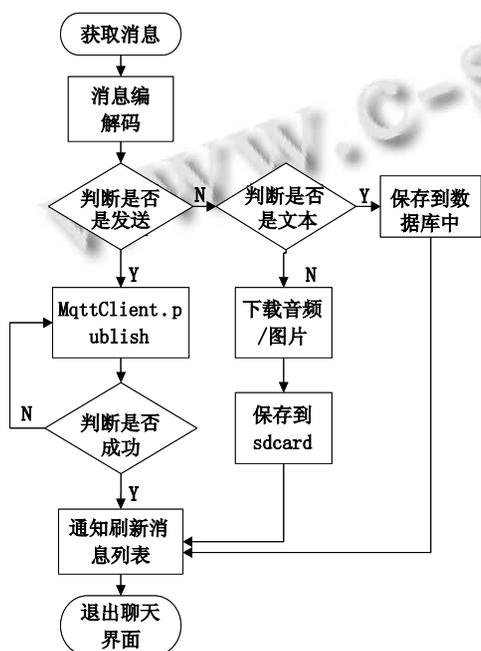


图 4 即时通信消息发布/接收流程图

#### 3.2.2.5 消息编解码模块的设计与实现

消息编解码模块的处理逻辑:

① 编码: 首先, 此模块接收即时通信模块传递过来的文本、图片地址或音频地址; 然后判断所要发布的消息的类型, 将消息按照下节定义的消息体格式进行编码, 最后将编码后的消息发布到即时通信话题上。

② 解码: 首先, 此模块接收由 MQTT 客户端从 MQTT 服务器接收到的消息, 然后判断消息类型, 如果是即时通信话题的消息, 按照定义的 payload 格式进行解码, 判断消息内容的类型, 文本消息保存到数据库中, 图片和音频消息保存到 sdcard 中。

#### 3.2.2.6 其他模块的设计与实现

除了以上介绍的模块, 本应用还包含数据传输模

块和数据存储模块。

数据传输模块: 此模块主要使用 Socket 连接数据服务器, 使用多线程技术将音频和图片数据以分组的形式上传到数据服务器。

数据存储模块: 此模块主要针对即时通信模块, 使用 Android 本身自带的 SQLite 数据库存储用户之间发送的消息, 其中语音和图片下载后保存在手机的 sdcard 中。

## 4 测试

系统测试环境: 2 台 Android 智能手机, 数据服务器, MQTT 服务器, 3G 网络和 wifi 网络环境<sup>[8]</sup>。2 台 Android 智能手机安装本程序。测试步骤如下:

(1)3G 网络下用户 A, 用户 B 开启应用, 登录系统, 互相搜索好友, 并添加。

(2)用户 A 向用户 B 发送文本消息, 用户 B 收到消息后向用户 A 发送确认消息, 然后向用户 A 发送图片消息, 用户 A 接收图片消息并查看, 确认后发图片消息给用户 B。

(3)用户 A 向用户 B 发送语音消息, 用户 B 接收后, 回复一条语音消息。

(4)用户 A 退出应用, 用户 B 向用户 A 发送消息。此时用户 A 登录, 查看是否有未读消息。

(5)更换 wifi 网络, 重复(1)至(4)步。

测试效果图如图 5 所示:



图 5 测试效果图

(下转第 196 页)

由 B-Tree 结构分析, 每次检索最多需要 $(h-1)$ 次 I/O 操作.  $h$  为树的深度, 而且一般很小. 而是否在数据库中创建索引, 决定于多方面因素. 特别是大规模的数据库中, 更需要建立索引来缩小查询时间. 本文在 Android 手机上对存放不同规模数据的数据库进行查询测试, 测试平均时间如表 2 所示. 测试结果表明: 随着数据库规模的扩大, 利用索引查询可以节省大量的时间, 加快数据库访问速度, 提高了整个应用程序的性能.

表 2 索引查询时间测试

规模(条)	自动提交	手动提交
10	1.1ms	0.91ms
100	1.4ms	0.93ms
1000	1.9ms	1.1ms
10000	11.2s	1.2s
100000	103.2s	1.4s

#### 4 结语

SQLite 数据库是一款轻量级嵌入式数据库, 广泛应用于 Android 等移动设备. 数据库性能调优旨在适应数据库规模的扩大化和复杂化, 缩短查询、插入等数据库操作的时间, 提高应用程序性能. 而 SQLite 数据库实际是磁盘上的文件, 频繁或大量的对文件操作, 降低了程序运行速度. 本文充分分析了 SQLite 事务机

制和索引结构, 通过手动添加事务减少了重复打开、提交事务, 减少耗时的 I/O 读写次数, 并通过创建 B-tree 结构的索引降低缩短查询时间. 在 Android 手机上测试, 测试表明二者分别在大量插入数据和从大数据库中查询上, 时间的缩短是非常可观的. Android SQLite 数据库性能的优化, 扩大了应用范围和建立更加良好的用户体验.

#### 参考文献

- 1 Lv JY, Xu SG, Li YJ. Application research of embedded database SQLite. 2009 International Forum on Information Technology and Applications. IEEE, Computer Society. 2009. 539-543.
- 2 尧有平, 薛小波. 基于 ARM-Linux 的 SQLite 嵌入式数据库的研究. 微计算机信息, 2008, 24(2): 64-66.
- 3 Wang Y, Liu ZJ, Xiang W, Liu XH. Research on concurrency and transaction optimization of relational database. 计算机技术与应用发展, 2008: 321-325.
- 4 Allen G, Owens M. The definitive guide to SQLite. American, Apress, 2010: 193-144.
- 5 Database Speed Comparison. <http://www.sqlite.org/speed.html>.
- 6 禹亮. 基于内容的图像索引和浏览算法研究[学位论文]. 长沙: 湖南大学, 2007.

(上接第 200 页)

测试结果表明, 此应用满足即时通信应用的基本功能及需求.

#### 5 结语

本文研究了 MQTT 协议的结构、消息格式, 并且使用此协议在 Android 平台上实现了一款基于 MQTT 协议的即时通信应用, 并验证了应用的使用效果. MQTT 作为一款轻量级的消息发布/订阅协议, 其在手机上的发展前景十分可观, 在 Android 推送方面的应用还有很大的发展空间, 尤其是此协议低带宽的特点, 使其在带宽受限的地方倍受青睐. 本文中的即时通信应用并不完善, 尤其是安全性方面未做考虑, 在未来的工作中将会侧重这方面的研究.

#### 参考文献

- 1 MQ Telemetry Transport. <http://mqtt.org>.

- 2 姜梦兰. 基于消息中间件服务可靠性保证方案的研究与实现[硕士学位论文]. 成都: 电子科技大学, 2010.
- 3 Hunkeler U, Truong HL, Stanford-Clark A. MQTT-S - A publish/subscribe protocol for wireless sensor networks. Communication Systems Software and Middleware and Workshops. Bangalore. 2008. 791-798.
- 4 MQTT V3.1 Protocol Specification. <http://wenku.baidu.com>. 2012
- 5 Lee S, Kim H, Hong D, Ju H. Correlation analysis of MQTT loss and delay according to QoS level. Information Networking (ICOIN). Bangkok. 2013. 714-717.
- 6 安卓推送技术探讨. <http://wenku.baidu.com>. 2012.
- 7 袁远. 基于 Android 平台端到端即时通信系统的分析与设计[硕士学位论文]. 北京: 北京邮电大学, 2012.
- 8 王楠, 宋飞, 周华春. 一种基于 Android 平台的即时通信方案. 计算机应用与软件, 2013, (4): 107-109, 148.