

基于 Wifi 的 Android 通信系统^①

单家凌, 叶润发

(广东白云学院 电气与信息工程学院, 广州 510450)

摘 要: 文章在对相关通信协议研究的基础上, 设计出了一种 Android 通信系统, 该系统利用 wifi 网络, 采用点对点体系结构, 客户端之间通过 Socket 传送消息. 除了即时文本信息的传输外, 增加了上下线功能和文件传输功能. 采用多线程技术, 增强了程序对系统资源的利用. 在 Android 平台上测试, 取得了较好的运行效果.

关键词: 安卓; WiFi; 飞鸽传书协议; 即时通讯

Android Communication System Based on Wifi

SHAN Jia-Ling, YE Run-Fa

(Guangdong Baiyun University, School of Electrical and Information Engineering, Guangzhou 510450, China)

Abstract: This paper designs an Android communication system on the basis of studying the related communication research, and the system utilizes the wifi network and point-to-point architecture and messages between client sides are delivered by means of the Socket. In addition to the instant text information transmission, functions of on-line and off-line as well as file transfer are added. The adoption of the multi-threading technology enhances the program's utilization of the system resources. Better operating results are obtained in the test on the Android platform.

Key words: Android; Wifi; ipmsg protocol; instant messaging

1 引言

随着智能手机的普及以及性能的不不断提升, 这为通讯系统从传统的 PC 机到手机的移植提供了很好的条件. 当前的手机操作系统过于封闭, 各大即时通讯软件采用的通讯协议也不统一. 于是利用 Wifi 网络, 设计和开发通信系统成为手机软件开发的热点.

2 即时通讯协议的发展现状

目前主流的 IM(Instant Messaging)协议^[1]主要有即时信息和空间协议(IMPP)、空间和即时信息协议(PRIM), 针对即时通讯和空间平衡扩充的进程开始协议 SIP(SIMPLE)和 XMPP 协议.

IMPP 主要定义必要的协议和数据格式, 用来构建一个具有空间接收和发布能力的即时信息系统. SIMPLE 被定义为建立一个 IM 进程的方法. SIMPLE 使用 SIP(IETF 中为终端制定的协议)来建立一次进程,

再利用 SDP(进程描述协议)来实际传输 IM 数据. XMPP 是一种基于 XML 的协议, 它继承了 XML 环境中灵活的扩展性. 经过扩展以后的 XMPP 允许通过发送扩展的信息来处理用户的需求, 以及在 XMPP 的顶端建立如内容发布系统和基于地址的服务等应用程序, 而且 XMPP 还包含了针对服务器的软件协议, 实现了在不同服务器之间的通话.

3 系统通信机制研究

随着移动通信与 Internet 的飞速发展及其相互融合, Wifi 使无线网络高速接入 Internet 成为现实, 移动用户从而可以享受到 Internet 的服务. 手机客户终端通过 Wifi 无线网络采用 TCP 协议连接到服务器, 通过 ipmsg 即时通讯协议进行即时通讯成为一种潮流^[2]. TCP 提供一种面向连接的、实时的双向传输通道. TCP 将用户数据打包构成报文段, 发送数据后启动一个定

^① 收稿时间:2013-10-03;收到修改稿时间:2013-11-13

时器, 等待对端数据确认; 另一端对收到的数据进行确认, 对失序的数据重新排序, 丢弃重复数据; TCP 提供端到端的流量控制, 并计算和验证一个强制性的端到端检验和. 但是 Wifi 网络对 TCP 链路存在一个限制, 当 TCP 链路在长时间没有数据流量时, 会自动降低此链路的优先级直至强制断开此链路. 所以在应用中, 采用发送心跳的方式来维持此链路. 而且 ipmsg 即时通讯协议最典型的支持协议也是 TCP 协议. 根据这些特点, 文章设计的 Android 通信系统中, 采用 TCP 协议来作为作为构建客户端与服务端之间的网络连接.

4 系统客户端解决方案

系统采用点对点架构的体系结构, 客户端基于 Android 平台进行开发^[3]. 通过 Wifi 无线网络与局域网建立连接, 实现与 Android 客户端之间的即时通讯. 客户端负责初始化通信用途, 进行即时通讯时, 由发起方客户端负责向接收方客户端发起创建连接请求. 系统的客户端定位于 Android 手机平台. 采用飞鸽传书协议作为即时通讯协议, IPMsg 飞鸽传书网络协议是基于 TCP/IP 协议. 企业员工可在企业内部或外部通过飞鸽传书进行通讯, 支持消息发送, 文件传输, 语音视频等. 为企业提供安全, 稳定的即时通讯解决方案.

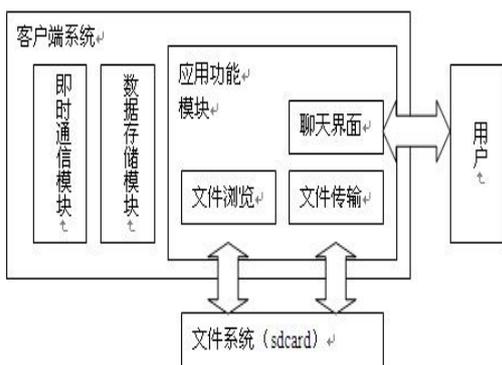


图 1 客户端功能模块图

5 系统客户端的设计与实现

在系统客户端设计中, 采用 MVC (M 是模型层, V 是视图层, C 是控制层) 的设计模式^[4]. 将视图层与逻辑代码以及数据模型层进行分离, 实现系统的松散耦合, 便于系统的维护和代码的重用. 同时 Android 也为采用 MVC 的设计模式来设计和开发应用程序提供了很

好的基础. 在 Android 中, 视图层可以通过定义 XML 文件的方式来设计用户界面, 在数据模型层提供了 ContentProvider 这个类来屏蔽底层具体的存储设备细节, 使开发人员可以专注业务逻辑的开发而无需关心所使用的数据库和数据访问细节, 从而使应用系统具有良好的数据迁移性.

5.1 数据存储模块

在客户端中, 文章采用创建用户类和信息类来存储用户信息以及即时通讯聊天信息, 使用文件存储设备 sdcard 来存储接受的文件.

用户类: 对应局域网中每个在线用户的信息.

```
private String userName; // 用户名
```

```
private String ip; // ip 地址
```

```
private String hostName; // 主机名
```

消息类: 对应局域网中每个在线用户发送的信息.

```
private String senderIp; // 消息发送者的 ip
```

```
private String senderName; // 消息发送者的名字
```

```
private String receiverIp; // 消息接收者的 ip
```

```
private String msg; // 信息内容
```

```
private Date time; // 发送时间: 格式
```

5.2 即时通讯模块^[5]

系统客户端通过 Wifi 无线网络采用 TCP 协议连接到另一个客户端, 实现客户端与客户端之间的即时通讯. 在 Android 应用程序中, 与用户进行交互的应用活动都是运行在主线程中, 在与对方的通信过程中采取队列的方式, 创建两个队列, 一个队列负责消息的发送, 一个队列负责消息的接收, 使用 Android API 中的 android.OS.Helper 这个类, 用于消息的分发和处理. 用户线程从消息队列中取出消息进行一些处理, 然后实例化一个 Helper 类, 发送消息时, 将要发送的消息封装成 ipmsg 语句的形式, 并将其放入消息队列中, 当队列中有消息的时候, 就自动通知发送线程发送消息. 部分代码如下:

```
sendUdpData(ipmsgSend.getProtocolString(), udpResPacket.getAddress(), udpResPacket.getPort()); // 发送数据
```

```
receiverUdpData(ipmsgReceiver.getProtocolString(), udpResPacket.getAddress(), udpResPacket.getPort()); // 接收信息
```

在即时通讯中, 当聊天界面活动不处于运行状态时, 如用户正在文件浏览, 则消息以通知的形式发送

到当前处于运行状态活动的最上方, 通知用户接收消息.

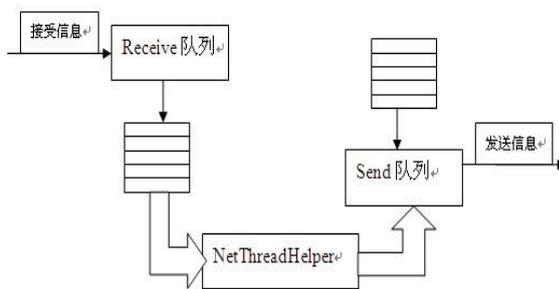


图 2 系统客户端通讯模块图

如图 2, 由接收消息队列 Receiver 负责接收从对方发送过来的消息, 将接收到的消息放入消息队列中。对象 Reader 从消息队列中取出消息, 以 ipmsg 语句的形式进行发送的, 调用对象函数 ChatMessage 负责对语句的解析, 调用对象函数负责对 udp 语句的解析, 并调用解密模块对解析的消息内容进行解密操作, 然后根据解析元素的不同类型封装成不同的数据对象。通过对象 NetThreadHelper 的 sendUdpData 方法封装成特定的消息格式, 并在 sendUdpData 方法中调用应用程序模块进行处理^[6]。应用程序模块发送消息时, 通过对象 Send 将用户输入的消息内容以及相关附加内容封装成 udp 语句的形式, 并放到消息队列中, 通过发送消息队列发送消息, 在消息发送之前, 对消息的内容进行加密操作。

5.3 应用模块

5.3.1 文件浏览

本功能模块使用 java.io.File, java.util.ArrayList, java.util.HashMap, java.util.List, java.util.Map 类, 这些类包提供了底层文件浏览的工具, 用于让用户选择文件, 返回所要发送文件的绝对路径。核心算法如下:

```
//放上根目录
Map<String, Object> root = new
HashMap<String, Object>();
root.put("name", "/");
root.put("img", R.drawable.file_root);
root.put("path", "回根目录");
adapterList.add(root);
//放上父目录
Map<String, Object> pMap = new HashMap
```

```
<String, Object>;
pMap.put("name", "..");
pMap.put("img", R.drawable.file_parent);
pMap.put("path", "上一级");
adapterList.add(pMap);
//得到父文件
File pFile = file.getParentFile();
if(pFile == null){
    Toast.makeText(this, "当前路径已经是根目录,
    不存在上一级",
    Toast.LENGTH_SHORT).show();
    refreshListItems(path);
}else{
    path = pFile.getAbsolutePath();
    refreshListItems(path);
}
```

5.3.2 文件传输

通过系统, 用户登录到客户端, 向对方客户端发送文件传输的请求, 同时根据对方客户端转发过来语句判断对方是否确定接收文件的状态。对于文件的传输, 文章创建两个线程来负责文件的发送和接受。在 AndroidAPI 中采用了 android.OS.Handle 这个类用于文件的分发和处理。用户线程从文件 Handle 中取出数据进行一些处理, 然后实例化一个 Handle 类, 发送文件时, 通过 socket 建立连接, 将要发送的文件封装成 tcp 语句的形式, 并将其放入文件 Handle 中, 当 Handle 中有消息的时候, 就自动通知发送线程中的文件。部分算法如下:

```
//发送文件
socket = server.accept();
Log.i(TAG, "与 IP 为" + socket.getInetAddress().
getHostAddress() + "的用户建立 TCP 连接");
BufferedOutputStream bos = new BufferedOutput
Stream(socket.getOutputStream());
Log.d(TAG, "本次发送的文件具体路径为" +
filePathArray[sendFileNo]);
File sendFile = new File(filePathArray[sendFileNo]);
//要发送的文件
BufferedInputStream fbis = new Buffered
InputStream(new FileInputStream(sendFile));
MyFeiGeBaseActivity.sendMessage(UsedCo
nst.FILESENDSUCCESS); //文件发送成功
```

6 系统部署及测试

系统采用点对点的体系结构. 采用飞鸽传输协议作为系统的即时通信协议, 客户端基于 Android 平台进行开发. 客户端采用创建表来存储用户信息和聊天信息, 文件则存储在存储设备 sdcard 中^[7]. Android 客户端通过连接 Wifi 可实现与 Android 客户端或 PC 机客户端之间的即时通信.

6.1 系统部署

由于设备的准确性, 消息的发送和接收通过 Wifi 网络进行传输. 与手提电脑上进行即时通讯的软件采用飞鸽传书即时通讯软件, 根据系统的特点, 文章搭建测试环境如下:

- (1) 已安装飞鸽传书客户端软件的手提电脑一台.
- (2) 安装了该系统的 Android 手机两台.

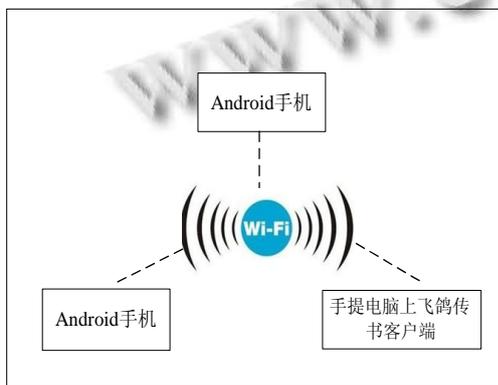


图3 测试环境搭建图

6.2 系统测试

启动 Wifi 网络及安装客户端软件的 Android 手机, 并在手机上运行客户端软件, 电脑上运行飞鸽传书客户端软件. 文章主要从系统消息的发送和接收、图片/文件的发送和接收以及文件的浏览几个方面进行测试.

(1) 消息的发送和接收

测试要求:

Android 手机用户 A 和 Anroid 手机用户 B 之间和 Android 手机用户 A 和手提电脑用户 C 之间进行即时通讯, 互相发送文本消息, 通讯双方能够准确的收到对方的消息.

(2) 图片文件的发送和接收

测试要求:

Android 手机用户 A 分别向 Anroid 手机用户 B 以及手提电脑用户 C 之间发送文件名为“01.jpg”大小

为 27.7KB 和文件名为“AP3.0.jpg”大小为 5.32KB 的图片文件, 双方能够准确的发送和接收.

测试结果如图 4 所示.



图4 用户之间发送图片请求和成功

7 结语

Android 是基于 Linux+Java 的手机操作系统平台, 文章根据 ipmsg 传输协议的规则, 利用该平台实现多线程、handler 通信、Socket 编程和对消息进行解析, 设计并开发出了基于 Wifi 的 Android 通信系统, 实现了手机与手机、电脑之间的相互通信. 经测试表明, 该系统与传统通信系统相比具有文件传输速度快, 系统资源占用率低的优点. 然而在获取群组列表时, 系统只能获取已登录用户的列表, 解决这一问题, 还有待对 XMPP 协议和 OPenfire 服务器进行进一步研究.

参考文献

- 1 公磊,周聪.基于 Android 的移动终端应用程序开发与研究.计算机与现代化,2008,(8):85-89.
- 2 徐虎,彭正涛,等.基于 WiFi 的 Android 移动设备语音通信系统的设计开发.计算机应用与软件,2012(11):225-228.
- 3 耿东久,索岳,陈渝,等.基于 Android 手机的远程访问和控制系统.计算机应用,2011,(2):559-561.
- 4 宋冉昕,谢维波.基 Android 平台的实时自适应音频传输技术及其实现.电脑知识与技术,2011,(2):892.
- 5 张焯.基于 Android 平台 WIFI 的实现及应用[硕士学位论文].北京:北京邮电大学,2012:56-63.
- 6 张利国,代闻,龚海平.Android 移动开发案例详解.北京:人民邮电出版社,2010,2.
- 7 贾琴勇.即时通讯系统的研究与实现[硕士学位论文].武汉:武汉理工大学,2008:28-36.