

# 基于 Android 平台的多人视频聊天系统<sup>①</sup>

杨 健, 周渊平

(四川大学 电子信息学院, 成都 610065)

**摘 要:** 随着互联网技术和计算机网络技术的快速发展, Android 平台系统普遍应用到智能移动终端上, 基于 Android 平台的多人视频聊天系统能使得人们在何时何地都能实现了远程多人视频聊天, 提高工作效率, 受到很多公司和科研机构的关注. 本系统以 Websocket 为通信信令, 利用 WebRTC 技术传送视频和音频, 设计了多人视频聊天系统, 包括了服务器及 Android 客户端; 在同一个房间的 Android 客户端可以实现远程多人视频聊天. 本文重点介绍了采用 Websocket 信令方式和利用 WebRTC 相关等技术实现了多人视频通信.

**关键词:** Android; 多人视频聊天; Websocket; 服务器; WebRTC

## Multi Video Chat System Based on Android Platform

YANG Jian, ZHOU Yuan-Ping

(School of Electronics and Information Engineering, Sichuan University, Chengdu 610065, China)

**Abstract:** With the rapid development of Internet technology and computer network technology, Android platform system has been widely applied to intelligent mobile terminal to make remote multi video chat anytime and anywhere possible thus improving work efficiency. Such multi video chat system based on Android platform has been concerned by many companies and research institutions. In our system Websocket is used as a communication signal, and WebRTC technology is used to transmitting video and audio in multi video chat system, which includes the server and the Android client. Remote multi video chat is realized by Android client within the same room. This paper focuses on the use of Websocket signaling and WebRTC technology to achieve multi video communication.

**Key words:** Android; multi video chat; Websocket; server; WebRTC

随着科技和互联网技术的发展, 人们对生活的要求更加智能化, 便捷化. 人们希望在千里之外能看到自己的亲人或者家里的情况. 这样视频聊天使得生活更加便利. 自从 Google 将 Android 系统开源后, Android 系统普遍应用到各种环境下.

目前, 多人视频聊天技术大多数都是基于 Web 端开发. Web 端的开发主要用于浏览器去访问, 每次的访问都会给用户带来不必要的繁琐. 本文中所用到 Websocket 协议主要应用也是 Web 端. 本系统将其移植到 Android 平台中, 使得多人视频聊天实现了简单化, 便捷化. 使得智能生活触手可及.

本文设计一种在 Android 平台上的多人视频聊天系统, 不仅可以实现随时随地的多人视频聊天的功能.

相比以往的视频聊天系统, 本系统可以跨多平台, 且可以应用于视频会议、远程监控、智能家居等方面. 同时在任何地方在任何两个 Android 设备上可以实时的多人视频通话. 从而使得互联网时代的生活更加智能化和实用化.

## 1 系统开发中的关键技术

### 1.1 Websocket 协议

Websocket 协议是 html5 的一种新的协议, 是实时通信中的关键技术. 主要实现了服务器与客户端之间的全双工通信. 相对于 Http 协议, Websocket 协议只需要一次握手, 就可以打开一个数据通道, 而且请求数据的 header 比较小, 节省了很多带宽<sup>[1,2]</sup>.

<sup>①</sup> 收稿时间:2015-04-16;收到修改稿时间:2015-05-28

Websocket 目前支持很多浏览器, 同样也支持 Android 平台. 所以使其作为本系统的信令通信方式成为了可能.

### 1.2 WebRTC 技术

WebRTC 是一个支持网页浏览器进行实时语音对话或视频对话的技术, 是谷歌 2010 年以 6820 万美元收购 Global IP Solutions 公司而获得的一项技术. WebRTC 实现了三个 API, 分别是 MediaStream API, RTCPeerConnection API, RTCDataChannel API. 分别实现了摄像头的音视频的采集与编码, 网络传输, 显示等功能. 本系统主要采用 MediaStream API 对 Android 终端对音视频的采集, 采用 VP8 对音视频的编解码. 采用 RTCPeerConnection API 实现了点对点的视频聊天功能.

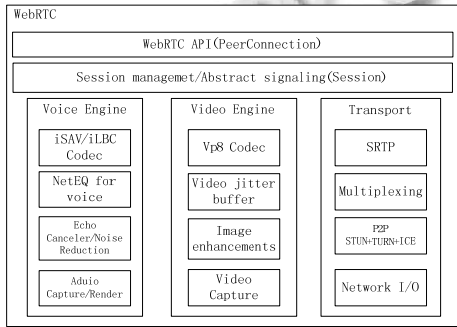


图 1 WebRTC 结构图

## 2 服务器端设计

本系统的系统包括两个部分. 一个是信令服务器, 另一个是转发服务器.

### 2.1 信令服务器

信令服务器(Signaling Server)主要负责 Android 的客户端的登陆、登出与注册, 和其他客户端的信息交换. 本系统信令服务器用 JavaScript 语言写. 本系统信令服务器有两个线程, 主线程主要主要监听端口 3000, 当有客户端登陆时主线程会再交给 Websocket 处理线程.

在 Websocket 线程中, 服务器端通过监听函数 addListener() 监听连接信息, 并解析三种信息: connection, message, disconnection. 若是 connection 信息, 则服务器端会收到客户端的 Socket, 通过此 Socket 接收客户端的信令信息, 服务器进行解析, 获得相应的客户端姓名及 ID 信息, 服务器端做出记录, 并将客户端添加到相应的房间中. 若是 message, 此时信令信息主要是转发到其他客户端的信令信息, 此时

信令服务器只要将信令信息的地址更改为目的客户端的地址并将数据转发到此地址. 若是 disconnection 信息, 服务器就会把客户端的信息从 peers 的列表中删除, 并将相应房间中的客户端删除.

### 2.2 转发服务器

转发服务器(TURN Server)主要的任务就是转发音视频流, 本系统并不是所有的情况都要转发视频, 而是在当有一端客户端在对称型网络结构中时服务器负责转发音视频数据<sup>[3,4]</sup>.

本系统中使用的转发服务器是开源 TURN Server, 在此基础上本系统把原来的最大带宽 1024Mbps 改到 2048Mbps, 为用户提供更快更好更流畅的支持, 也减小了服务器的负荷承载. 还有把登陆此服务器的客户端的 Lifetime 改到最大值 1 hour. 使得用户单次通话时间大大提高.

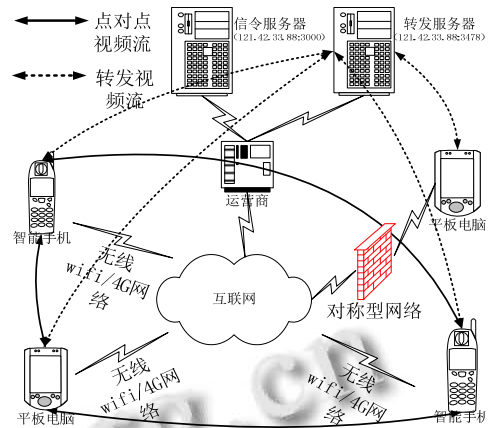


图 2 系统框架图

## 3 Android 客户端设计

### 3.1 客户端设计方案及功能

本系统是多人视频聊天, 我们采用的采用的方案是网型结构, 在此结构中, 每个客户端之间是点对点视频通信, 当多人进入视频聊天中时, 每个人的 Android 终端都会增加一个视频显示. 这样的结构减少了对服务器的依赖.

Android 客户端实现了主要以下几个功能:

(1) 登陆 App 界面, 对其自身进行配置, 登陆相应的房间, 客户通过相应的网络(无线 wifl 或者 4G 网络)登陆服务器, 对 Android 客户端的用户名登记与注册.

(2) Android 客户端通过麦克风和摄像头对音视频采集并进行编码, 然后通过传输层利用 RTP 和 RTCP 协议将音视频信号打包传送到其他 Android 客户端.

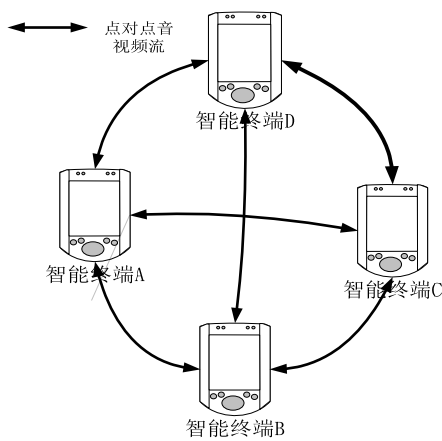


图 3 网型结构实例图

### 3.2 视频通信模块

音视频通信模块是本系统的主要部分，也是多人音视频聊天的基础单元。下面我们首先讲解两个人的视频聊天过程。

#### 3.2.1 信令部分

本系统在 WebRTC 开源项目的基础上对其信令部分做了很多改动。抛弃了项目自带的 GAE 信令方式，改用非常实用的 Websocket 的信令方式。该信令方式对传递音视频信息具有更好的实时性。

本系统通过使用 Java 语言重新编写实现了 open()、onmessage()、onclose() 等函数，用来处理客户端登陆与退出，及转发信息等功能。客户端可通过此构造 ws= new WebSocketClient("121.42.33.88:3000") 实现了与服务器端通信并交换信令信息。在此我们将此实现 Websocket 的工程打包成导出打包成 lib\_android\_Websockets.jar 的包，并在客户端相应的 Android 工程中使用。

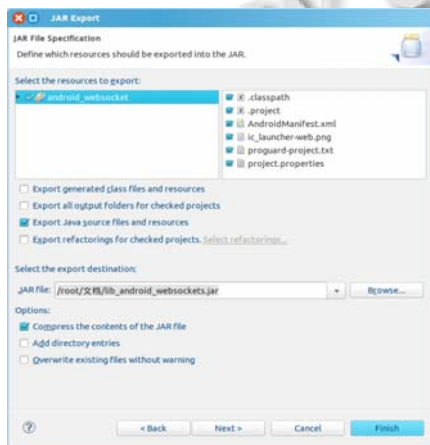


图 4 打包 jar 实例图

#### 3.2.2 视频通信流程

本系统视频通信部分是在 WebRTC 开源项目做了修改，重新编译得到其 libjingle-peerconnection.jar。并导入 Android 工程中。本系统客户端的不仅是呼叫端，也是被呼叫端。本系统实现了自动建立连接。下面我们重点讲解视频通信流程<sup>[5,6]</sup>。

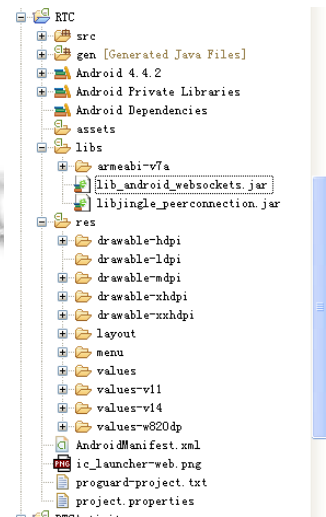


图 5 Android 工程实例图

##### 1) 呼叫流程

当客户端登陆相应的房间之后，信令服务器端进行登陆注册。当有其他用户进入房间中时，服务器发生信息到本地客户端，本地客户端接收到服务器的通知信息之后，由此客户端发起呼叫，服务器先初始化一个连接池，然后与客户端的用户连接工厂 PeerConnectionFactory 实例初始化一个用户连接 PeerConnection，并把音视频流添加到本地客户端，并将请求(offer)信令信息转发给其他客户端。客户端接收应答(answer)信令信息之后，建立视频通信连接。当客户端接收到请求(offer)信令信息时，则向远程客户端发出应答(answer)信令信息。

##### 2) 被呼叫流程

远程客户端接收到应答(answer)信令信息之后，服务器与远程客户端初始化连接，然后服务器向本地客户端发送请求(offer)信令信息。并添加媒体流信息到本地客户端。建立视频通信。在视频连接建立之后，再发送一次请求(offer)信令信息。

#### 3.2.3 音视频处理及传输流程

在音视频采集模块中，音频数据与视频数据采集之后分别将数据存放在 AudioTrack 和 VideoTrack 中<sup>[7,8]</sup>。

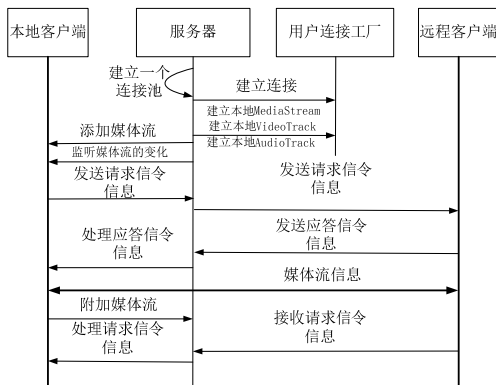


图6 呼叫流程图

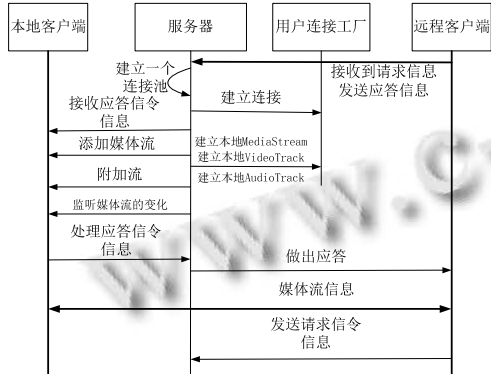


图7 被呼叫流程图

在音视频流处理模块中,采用的是的 WebRTC 中音视频引擎,其中使用 iSAV 技术与 VP8 技术分别对语音信号和视频图像进行编码.如图 8 所示,将编码后的音视频数据添加到 MediaStream 中,当客户端之间建立连接后,将 MediaStream 添加到用户连接 PeerConnection 中,通过服务器的传输,使用 RTP 与 RTCP 协议将音视频数据进行打包发送到其他目的客户端.

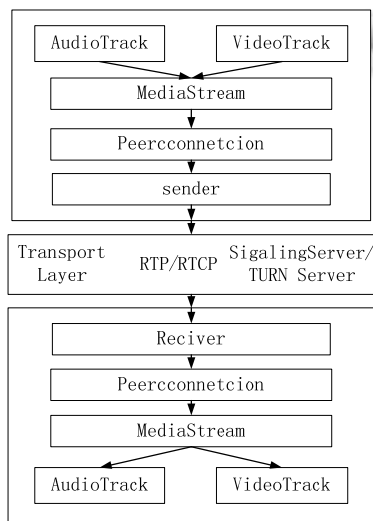


图8 音视频流处理流程图

在接收显示模块中,把用户连接 PeerConnection 中 MediaStream 数据提取出来,再将解码后的音视频数据分别放进 AudioTrack 与 VideoTrack 中,然后进行重新播放和显示.

### 4 系统测试过程及结果

本次测试中,服务器部署在阿里云上,IP 地址为 121.42.33.88,服务器系统为 Ubuntu 12.04.信令服务器所用端口为 3000,转发服务器所用端口为 3478.使用命令 node app.js 启动服务器.

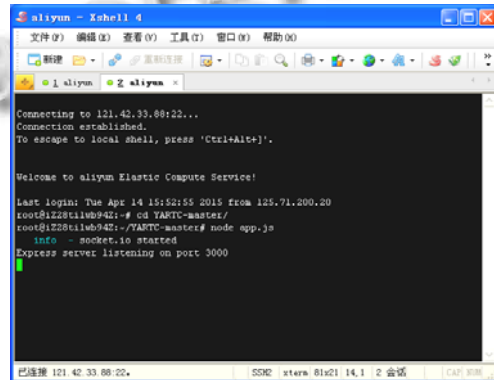
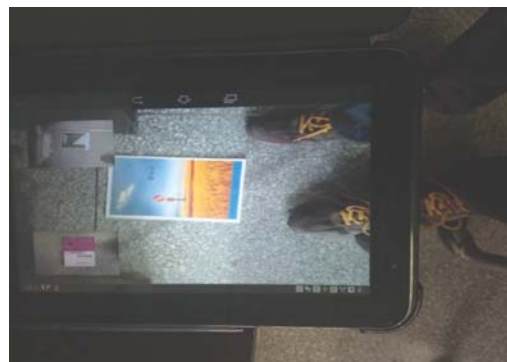


图9 服务器启动成功

客户端测试所用的智能终端分别是三星平板电脑,小米手机,魅族手机.系统均为 Android 4.0 以上系统.

三星平板和小米手机是使用无线 wifi,魅族手机使用 4G 网络.

如图 10 所示,分别是三星平板,小米手机,魅族手机视频聊天的画面.本系统理论上最多可以支持 5 人视频聊天.当测试人数增多时实时性就受到影响.本次测试选用 3 人聊天视频,3 组测试时间分别是 30'23", 53'49", 47'55";可以得出时间均在 30 分钟以上,最高视频通话时长可以达到 1 小时左右.画面和语言测试很流畅,实时性非常好.





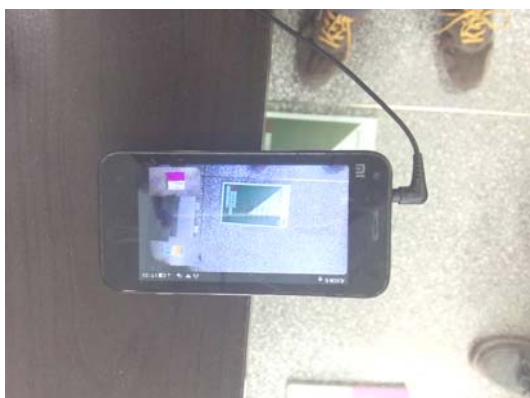


图 10 各视频聊天成员的画面

## 5 结语

开源 WebRTC 项目中的 GAE 信令方式对多平台的兼容性不高,而且实时性差,本文为了实现更好的实时性与兼容多平台的视频聊天系统,故采用了 Websocket 信令方式,并参考了 Web 端的 Websocket API,用 Java 语言重写 Android 端的 Websocket 相关函数,从而实现对 Android 端的移植。音频和视频的处理部分采用了 WebRTC 项目中相关技术处理,从而实现了 Android 平台的多人视频聊天系统<sup>[9,10]</sup>。本系统为

了适用于不同网络环境下视频聊天,故搭建了两服务器的结构,从而实现了网络穿透,适用于各种网络环境。该系统为研究在 Android 平台上也对 Android 视频聊天、视频监控和视频会议提供了一种参考方案。在下一步的工作中,我们还有优化此系统,使视频和音频中的去噪优化,而且也要提高其安全性,丰富其功能。

## 参考文献

- 1 陆晨,冯向阳,苏厚勤.HTML5 WebSocket 握手协议的研究与实现.计算机应用与软件,2015,32(1):128-131,178.
- 2 龙奇.下一代 Web 通信技术 HTML5 Web Socket 的研究.科技信息,2011,(36):273-273.
- 3 郭朝鑫.SIP 电话的 NAT 穿越技术研究.通信技术,2011,44(12):125-127,130.
- 4 朱光,张云华,卢娟.基于 ICE 的 VOIP 穿越 NAT 方案的研究.计算机应用与软件,2011,28(10):222-224.
- 5 屈振华,李慧云,张海涛,龙显军.WebRTC 技术初探.电信科学,2012,28(10):106-110.
- 6 董振江,李从兵,王蔚,吕达.移动互联网 WebRTC 及相关技术.中兴通讯技术,2013,19(6):28-32.
- 7 蔡晓霞,崔岩松,邓中亮,常志峰.下一代视频编码标准关键技术.电视技术,2012,36(2):80-84.
- 8 杨光平.基于 Android 的移动流媒体实时传输系统设计与实现[硕士学位论文].西安:西安电子科技大学,2012.
- 9 李琳.基于 Web 浏览器的实时通信技术 RTCWeb.数字通信,2012,39(5):55-58.
- 10 胡敏,刘六程.基于 WebRTC 的视频会议系统的设计与实现.电视技术,2013,37(1):141-143.