

A Method for Audio and Video Call on Smart Phones Based on Android and VoIP

Haiyao Wang¹, Huiyong Wang^{*2}, Jingyang Wang² and Min Huang²

¹Fujian Jiangxia University, Fuzhou, 350000, China

²School of Information Science and Engineering, Hebei University of Science and Technology, Shijiazhuang, 050018, China

* wanghuiyong815@163.com (corresponding author), why-helen@163.com, ever211@163.com

Abstract

Today more and more WIFI (Wireless Fidelity) networks are built, and their cover area becomes larger and larger. With the rapid development of the technologies of VoIP (Voice over Internet Protocol), smart phone and mobile Internet, more and more demand for audio and video call on smart phones are required. Android is an operating system which is widely used in smart phone. A method for audio and video call on smart phones based on Android and VoIP is presented in this paper, it realizes audio and video call on smart phones using AVCS (Audio and Video Call Service) provided by the third party cloud service platform. This paper introduces the system architecture, describes the working process in detail, takes experiments and analyzes its running effect. The method is simple and efficient. It shortens the development cycle and reduces the difficulty and cost of development.

Keywords: audio call; video call; VoIP; cloud service; smart phone; AVCS; Android

1. Introduction

With the rapid development of mobile device technology, people using SP (Smart Phone) is becoming more and more prevalent, and application running in SP is becoming more and more widespread. The common operating system of SP is iOS (iPhone Operating System) or Android. Some advantages of Android are open source, many supporting manufacturers and many built-in Google services, so Android is widely used in SP [1-2].

The rapid development of VoIP lays the foundation of technology for AVC (Audio and Video Call) over Internet. 4G mobile Internet's speed is improved and its cost is reduced, more and more WIFI networks are built and their cover area becomes larger and larger, these lays the foundation of good running environment for AVC. More and more demand for AVC on smart phones over Internet is required [3-4].

Although some IMP (Instant Messaging Platform) based on Android such as QQ and WeChat have the ability of AVC, but the API (Application Programming Interface) of AVC is not provided by these software development company. The AVC ability of QQ or WeChat can't be used in enterprise's Android application directly, so enterprise must develop the function of AVC [5-6]. In order to shorten the development cycle and reduce the difficulty and cost of development, the method of realizing AVC using AVCS provided by the third party cloud service platform is put forward in this paper.

2. System Architecture

The system architecture of the method for AVC on smart phones based on Android and VoIP is shown in Figure 1.

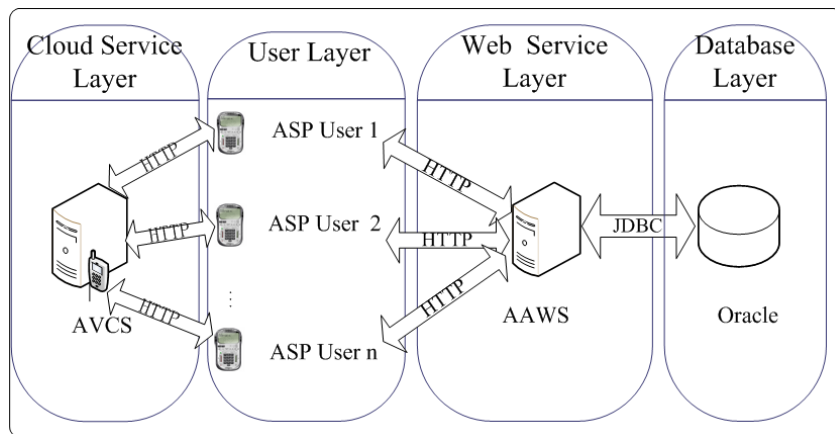


Figure 1. The System Architecture of the Method for AVC on Smart Phones Based on Android and VoIP

The system consists of User Layer, Cloud Service Layer, Web Service Layer and Database Layer. The AVC application which is based on Android and VoIP is installed in ASP (Android Smart Phone). The functions of ASP user register, user login and AVC are provided by AVCS. The functions of user register, user login and other business services are provided by AAWS (Android Application Web Service). The AVC application installed in ASP accesses AVCS through HTTP [6-7].

3. Working Process

The main steps of developing AVC function are as follows:

- 1) Building development environment;
- 2) Registering AVC cloud service account;
- 3) Login AVCS;
- 4) Taking Audio and Video call;
- 5) Exiting AVCS;

3.1. Building Development Environment

Building development environment is setting up the Android AVC project environment in Eclipse ADT.

1) Importing AVC SDK

These archives (including `httpmime-4.1.2.jar`, `rongkecloud-sdkbase-android.jar` and `rongkecloud-sdkav-android.jar`) provided by AVCS are imported to the lib folder of the Android AVC project in Eclipse ADT. The library files of correlation network request are included in `httpmime-4.1.2.jar`. The library files of correlation cloud interaction underlying infrastructure are included in `rongkecloud-sdkbase-android.jar`. The library files of correlation AVC are included in `rongkecloud-sdkav-android.jar`.

2) Configuration permission

These permissions are added to `AndroidManifest.xml` of the AVC project.

```
<uses-permission android:name="android.permission.VIBRATE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.CHANGE_NETWORK_STATE" />
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
<uses-permission android:name="android.permission.BROADCAST_STICKY" />
<uses-permission android:name="android.permission.WRITE_SETTINGS" />
<uses-permission android:name="android.permission.DISABLE_KEYGUARD" />
```

3) Configuration service

The service is added to AndroidManifest.xml of AVC project.

```
<service
    android:name="com.rongkecloud.sdkbase.LPSDaemonService"
    android:process="remote"
/>
```

3.2. Registration

Registration realizes that the Android Application gets the account from AVCS. The Android Application functions such as login AVCS, AVC and exit AVCS will be realized through the account. The activity diagram of registration from AVCS is shown in Figure 2 and the main steps are as follows:

- 1) Request the RKCLOUD_KEY of the Android application from AVCS;
- 2) Input user name and password, and then click "Register" button;
- 3) Validate the input info. The length of user name and password must be greater than 6 and less than 50. If verification is successful then turn to step 4, otherwise to step 2;
- 4) Send user name, password and RKCLOUD_KEY to AVCS through HTTP;
- 5) Accept the result from AVCS, analyze the value of the result, if 0 indicates register success then turn to step 6, otherwise to step 8;
- 6) Send user name, password and RKCLOUD_KEY to AAWS through HTTP;
- 7) AAWS saves user name, password and RKCLOUD_KEY to database;
- 8) End.

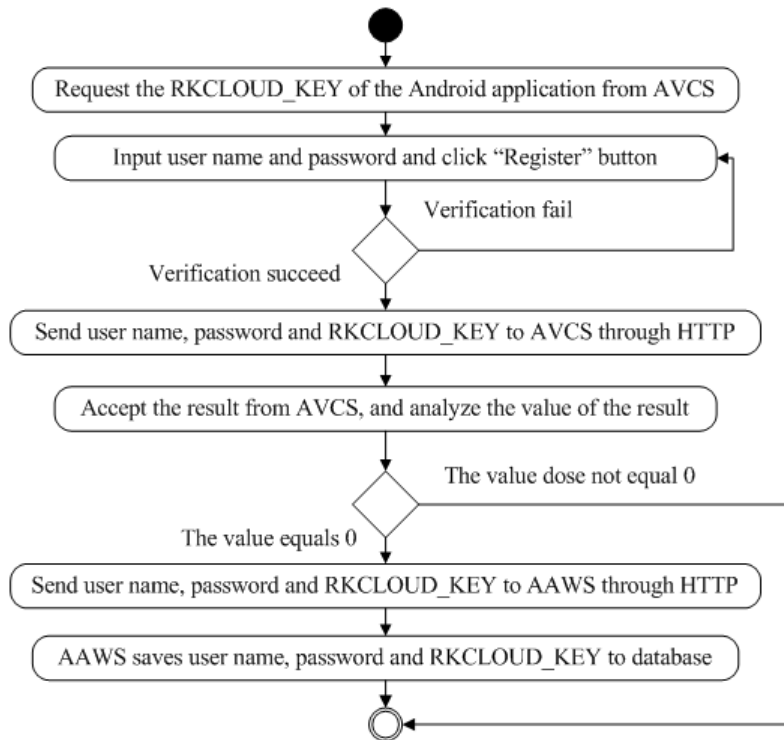


Figure 2. The Activity Diagram of Registration from AVCS

3.3. Login

The precondition of login is that registration has been finished. The activity diagram of login AVCS is shown in Figure 3 and the main steps are as follows:

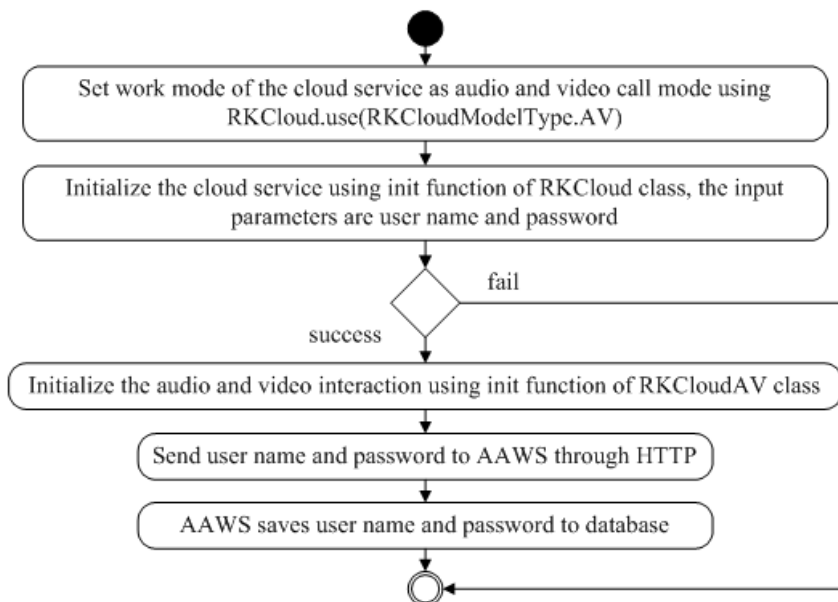


Figure 3. The Activity Diagram of Login AVCS

- 1) Set the work mode of the cloud service as audio and video call mode using `RKCloud.use(RKCloudModelType.AV)`;

- 2) Initialize the cloud service using init function of RKCloud class, the input parameters are user name and password. Get the result of initialization through the InitCallBack class's callback functions of onSuccess() and onFail();
- 3) If InitCallBack class's onFail function is invoked, it means that the cloud service initialization fails, turn to step 8;
- 4) If InitCallBack class's onSuccess function is invoked, it means that the cloud service initialization is successful, turn to step 5;
- 5) Initialize the audio and video call interaction using init function of RKCloudAV class;
- 6) Send user name and password to AAWS through HTTP;
- 7) AAWS saves user name and password to database;
- 8) End.

3.4. Audio Call

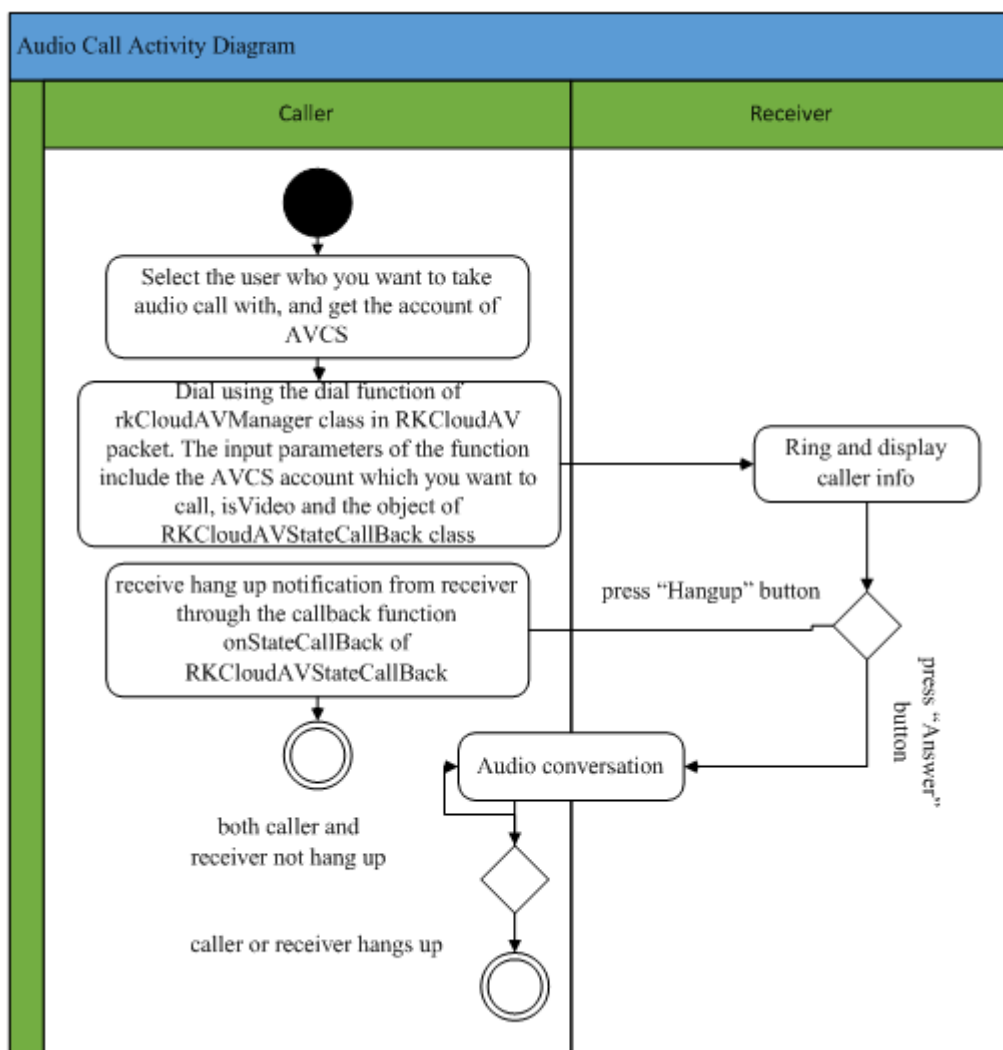


Figure 4. The Activity Diagram of Audio Call

The precondition of audio call is that ASP user has already logged in AVCS. The activity diagram of audio call is shown in Figure 4 and the main steps are as follows:

- 1) Caller selects the user who you want to take audio call with, and gets its account of AVCS;

2) Caller dials using the dial function of `rkCloudAVManager` class in `RKCloudAV` packet. The input parameters of the function include the AVCS account which you want to call, `isVideo` and the object of `RKCloudAVStateCallBack` class. `isVideo` is a Boolean variable, true indicates video call, and false indicates audio call. The object of `RKCloudAVStateCallBack` is used to accept the change of call state from receiver;

3) Ring and display caller info on receiver side;

4) If receiver presses “Hangup” button then the Android application invokes `RKCloudAV.rkCloudAVManager.hangup()` function to hang up the call and turns to step 5. If receiver presses “Answer” button then the application invokes `RKCloudAV.rkCloudAVManager.answer()` function to answer the call and turns to step 6.

5) Caller receives hang up notification from receiver through the callback function `onStateCallBack` of `RKCloudAVStateCallBack`, and then audio call is ended;

6) Audio conversation;

7) If caller or receiver hangs up using `RKCloudAV.rkCloudAVManager.hangup()` function then audio call is ended. If both caller and receiver not hang up then turns to step 6.

3.5. Video Call

The precondition of video call is that ASP user has already logged in AVCS. The activity diagram of video call is shown in Figure 5 and the main steps are as follows:

1) Caller selects the user who you want to take video call with, and gets its account of VCS;

2) Caller dials using the dial function of `rkCloudAVManager` class in `RKCloudAV` packet. The input parameters of the function include the AVCS account which you want to call, `isVideo` and the object of `RKCloudAVStateCallBack` class. `isVideo` is a Boolean variable, true indicates video call, and false indicates audio call. The object of `RKCloudAVStateCallBack` is used to accept the change of call state from receiver;

3) Ring and display caller info on receiver side;

4) If receiver presses “Hangup” button then the Android application invokes `RKCloudAV.rkCloudAVManager.hangup()` function to hang up the call and turns to step 5. If receiver presses “Answer” button then the application invokes `RKCloudAV.rkCloudAVManager.answer()` function to answer the call and turns to step 6.

5) Caller receives hang up notification from receiver through the callback function `onStateCallBack` of `RKCloudAVStateCallBack`, and video call is ended;

6) Caller and receiver set video display mode as vertical using `RKCloudAV.rkCloudAVManager.setOrientation(true)`;

7) Caller and receiver initialize video info using `RKCloudAV.rkCloudAVManager.initVideoInfo()`;

8) Caller and receiver set usage camera mode using `RKCloudAV.rkCloudAVManager.setCamera(1)`, 1 indicates using front camera, and 0 indicates using back camera;

9) Caller and receiver create local and remote video views;

10) Caller and receiver start to video call using `RKCloudAV.rkCloudAVManager.startVideo()`;

11) Video conversation;

12) If caller or receiver hangs up using `RKCloudAV.rkCloudAVManager.hangup()` function then video conversation is ended. If both caller and receiver not hang up then turns to step 11.

4. Effect

We take video call using the method provided in this paper, QQ and WeChat in the same experimental conditions.

The caller and receiver interface of video call using the method is shown in Figure 6.

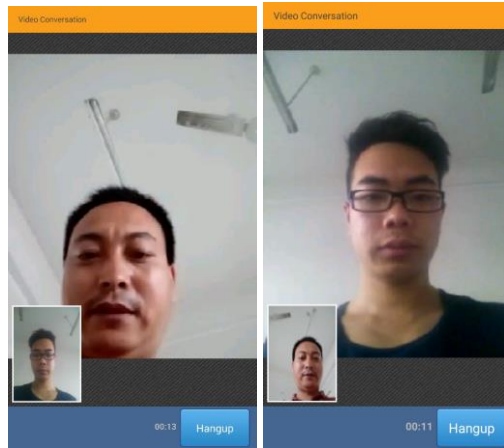


Figure 6. The Caller and Receiver Interface of Video Call Using the Method



Figure 7. The Caller and Receiver Interface of QQ Video Call

The quality of video call using the method is good. Its voice is clear, smooth, no delay and no echo. Its picture shows clearly, displays no skips, no stutters and no mosaic.

The QQ video call interface of caller and receiver is shown in Figure 7.

The WeChat video call interface of caller and receiver is shown in Figure 8.



Figure 8. The Caller and Receiver Interface of Wechat Video Call

The experiments show that there are no significant differences among the video conversation effect of video call using the method provided in this paper, QQ video call and WeChat video conversation.

5. Conclusion

This paper proposes a method for audio and video call on smart phones based on android and VoIP. The method realizes the AVC function of Android application using AVCS provided by the third party cloud service platform. Developer can ignore the implementation detail of AVC and can't master the AVC correlation technologies (such as video data capture, transmission and encoding, *etc.*), so the method is simple. It shortens the development cycle and reduces the difficulty and cost of development. After the experiments, its effect is good. The method for audio and video call can be widely used in applications which are based on Android such as Internet hospital, online trading platform, enterprise office automation, and so on. However, the method has also the following deficiencies:

- 1) Both the caller and the receiver must be on line. The caller can't take AVC if the receiver does not log in AVCS.
- 2) Caller does not know the network type (such as WIFI network or 3G/4G mobile Internet) used by the receiver, so some receivers will hang up the audio and video call if WIFI network is not used by their smart phone. It reduces the phone answer rate of audio and video call.

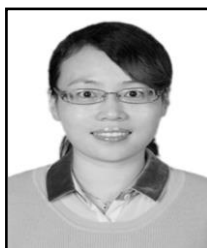
Acknowledgment

The authors would like to thank the editor and referees for their careful review and valuable critical comments. This work is supported by Natural Science Foundation of Hebei Province (No. F2013208107). This work is also supported by the fund of Hebei University of Science and Technology No. 2014PT21 and the Initial fund for doctors of Hebei University of Science and Technology No. QD201223.

References

- [1] Ni Jian, Xu Sheng, "Audio and Video Real-time Communication Design and Implementation Based on Android", *Journal of Civil Aviation Flight University of China*, Vol. 24, No.3, (2013), pp. 48-51.
- [2] Wang Chen, Zhou Jie, Ji Ju, "Implementation of Real-Time Video Technology Based on Android Platform", *Computer Applications and Software*, Vol. 31, No.6, (2014), pp. 120-122.
- [3] Yang Jiasheng, "The Design and Implementation of Android-Based Video Calling System", *Dalian University of Technology*, (2011).
- [4] Wu Dan, "Design and Implementation of Smartphone Video Call Software Based on Android", *Beijing University of Posts and Telecommunications*, (2012).
- [5] YANG Jian and ZHOU Yuan-Ping, "Multi Video Chat System Based on Android Platform", *Computer Systems & Applications*, Vol. 25, No.1, (2016), pp. 75-79.
- [6] Ryota Nishimura, Koji Sugiokay, Daisuke Yamamoto, "A VoIP-based Voice Interaction System for a Virtual Telephone Operator Using Video Calls", *2014 IEEE 3rd Global Conference on Consumer Electronics*, (2014), pp.529-532.
- [7] Jian Huang, Jiangtao Luo, Xiaolei Chai and Bin Zheng, "Research and implementation on the IMS smart terminal for mobile video call application", *2010 2nd IEEE International Conference on Network Infrastructure and Digital Content*, (2010), pp.536-540.
- [8] Dong Xu, "The research and implementation of the SIP video call management based on android OS", *Suzhou University*, (2013).
- [9] TONG Fang-yuan, Yu Qiang, "Real-time video stream transport system based on Android", *Computer Engineering and Design*, Vol.33, No.12, (2012), pp.4639-4642.
- [10] YUAN Hua, Yang Zean, LI Zhenning, et al. "Design and implementation of android-based intelligent remotecontroller for the video conferencing terminal", *Journal of Guangxi University*, Vol.36, No.1, (2011), pp.234-237.
- [11] TIAN Feng, "Design and Implementation of the SIP Soft Phone Based on Android", *Science Technology and Engineering*, Vol.13, No.17, (2013), pp.4946-4950.
- [12] Liu Zhen, Fang Weiwei, Shi Konggui, "iTDTs: a SIP-based telephone system for train dispatching", *JDCTA*, (2011), pp.88-100.

Authors



Haiyao Wang, Lecturer, born in 1976. She received the B.Eng. degree in Mechanical Manufacture from HeFei University of Technology, China, in 1998. She received the M.Sc. degree in Industrial Engineering from HeFei University of Technology, China, in 2009. Now she is working in College of Industrial Engineering, Fujian Jiangxia University, Fuzhou Fujian, China. Her main research areas include network and communication, multimedia technology, and industrial engineering.



Huiyong Wang, Lecturer, born in 1980. He received the B.Eng. degree in Mechanical and Electronic Engineering from Hebei University of Technology, China, in 2002. He received the M.Sc. degree in Mechanical and Electronic Engineering from Hebei University of Technology, China, in 2005. Now he is working in School of Information Science and Engineering, Hebei University of Science and Technology, Shijiazhuang Hebei, China. His main research areas include data mining, swarm intelligence, and distributed computing.



Jingyang Wang, Professor, born in 1971. He received the B.Eng. degree in computer software from Lanzhou University, China, in 1995. He received the M.Sc. degree in software engineering from Beijing University of Technology, China, in 2007. Now he is working in School of Information Science and Engineering, Hebei University of Science and Technology, Shijiazhuang Hebei, China. His main research areas include transmission control, image processing and distributed computing.



Min Huang, Associate Professor, born in 1979. He received the B.Eng. degree in automatic control from Hebei University of Science and Technology, China, in 2000. He received the M.Sc. degree in computer science from Beijing Institute of Technology, China, in 2003. His main research interests include network and communication, system modeling and identification, image processing and distributed computing.