A High Q&S RTSP Server's Architecture and Implementation Based On Android

Yang Zhang, Zhiguo Hong, Ning Lv
School of Computer Science, Communication University of China, Beijing 100024, China
cuczhangyang@gmail.com, hongzhiguo@cuc.edu.cn, lvning@gmail.com

Abstract - FFmpeg is RTSP server's core technology. Since 2010, Android mobiles have the highest speed of development. Android mobiles have enough high-quality devices to help us build server-based applications to provide audio, images and videos for web users. In this paper, by using FFmpeg to encode and decode stream media and choosing ADT tool as development software, a high Q&S RTSP server is constructed to realize video monitoring system.

Index Terms - RTSP Server, Android, ADT, FFmpeg, JAVA

1. Introduction

FFmpeg is an open source, free cross-platform video and audio streams and it uses LGPL or GPL license (based on the components you choose) [1]. FFmpeg has strong engines to encode and decode RTSP stream. FFmpeg also supports RTSP transfer protocol. Android is a widely used mobile system. Most of the Android mobiles have strong devices such as MSM8974 and Exmore RS CMOS &etc. A MI2S mobile will be used to build RTSP Server. Android Developer Tools (ADT) is an Eclipse-based developing tool under JAVA environment. A typical RTSP Server can be built by ADT which is distributed by Google. The main workflow can be summarized as follows. Firstly, Android mobile camera captures the real-time video. Then FFmpeg encodes the video and the encoded video is transmitted by RTSP protocol. When a mobile client user requests the RTSP Server, the full-duplex channel between RTSP Server and stream media player is established. If the bandwidth of the network is not enough to offer high-quality video's transmission, the corresponding code rate in the channel will be adjusted to fit the channel automatically. Recently, Android platform has taken a large proportion of mobile terminals (such as iPad, iPhone etc.) Therefore, we will consider the popularity by choosing related developing tools.

ADT is a widely used tool which helps developers to design Android applications. For the sake of good compatibility and future maintainment, we take ADT 22.3.0 package and SDK API 19(Android 4.4) [2] as developing tools to establish.

A. RTSP Server

Android mobile system is a Linux-core mobile system, which is created by Google. Android system contains Java Virtual Machine (JVM), and almost Android apps are developed by Java. Programmers can implement the RTSP Server with JVM. Since Linux platform is used as Android system's core, it offers the following powerful functions.

(1) Graphic Layout: with Linux core, Android support bigger Image resolution than IOS or Windows Phone. It works for better display effect.

(2) Built-In SQLite: a small database management system to store data. It helps easy to save streaming data.

(3) Various network types (especially such as WiMAX, Wi-Fi, Evdo): Programmers can choose free Wi-Fi to transport encoded streams.

(4) RTP/RTSP stream media (even HTML5 stream media): What's more, Android system can support many devices such as Camera, touch screen, 3D graphics accelerator. Due to the related APIs, programmers can make access hardware and devices easily [3].

2. Architecture RTSP Server

A. Use FFmpeg to Encode And Decode Streaming

In order to make further processing of video, we capture and digitalize the videos setting the configurations as follows. The resolution is 1080P. The fps (frames per second) is 25, the audio sampling rate is 44.1 KHz and sound cannel is stereo.

The uncompressed video's size for a few minutes is considerably large because the size is the product of time, sampling rate, fps etc. In the case of short time, it reaches 30MB. It would cost a lot of time when the video data is transmitted via a bottleneck bandwidth. Consequently, how to choose an effective compressing component by minimize the impact of user's visual experience is an urgent problem. So we use Libavcodec to encode the stream, and the encoded data is only 3MB which is fit for online show when the bandwidth is not enough for uncompressed video. The comparison of the original video and encoded one is shown as Figure 1.

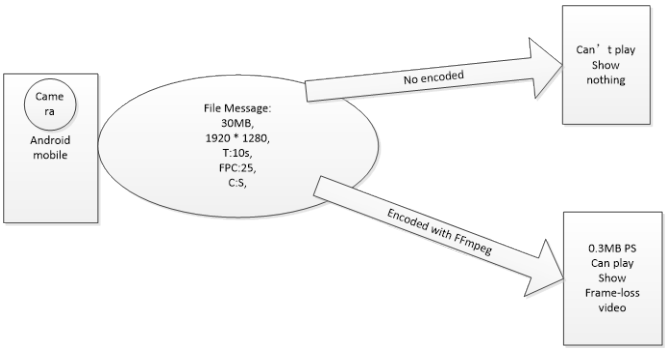

Figure 1 Comparison of Original video and encoded one

Figure 2 records the related statistics data of client when it plays the video from RTSP server. From Figure 2, we can observe the number of lost frames is 50.

Figure 2 Statistics data of client when playing the video from RTSP server

We conduct the experiment under the following environment. The maximum bandwith is 1000Mbps Ethernet and a PC's CPU ratio is i3 3.4GHz and the RAM size is 4GB.

B. Decode The Encoded With FFmpeg By Libvlc Core

The encoded stream data is transported by Ethernet network and can be played by a RTSP player such as VLC Player. The core of VLC is libvlc, a library to decode stream data and support basic control such as change aspect ratio and volume. The core of libvlc is FFmpeg; all of the operation by VLC can be replaced by FFmpeg but for a suitable network channel for playing stream media [4].

Figure 3 and Figure 4 reports the related statistics data of client when it plays the decoded video from RTSP server with VLCPlayer and KMPlayer respectively.

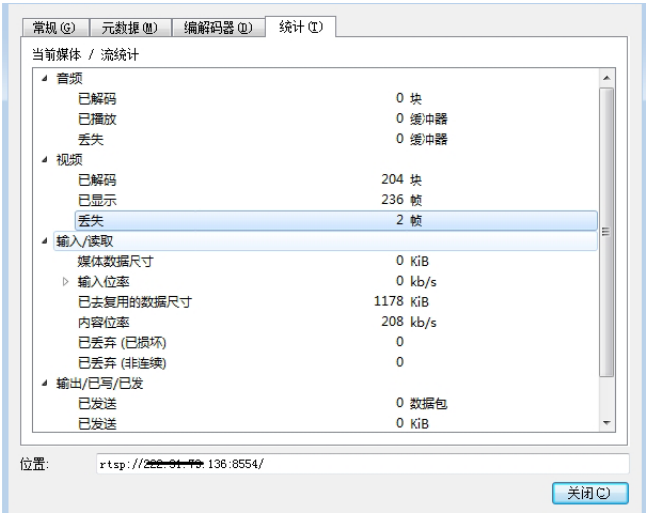

Figure 3 Statistics data of client when playing the video with VLCPlayer from RTSP server

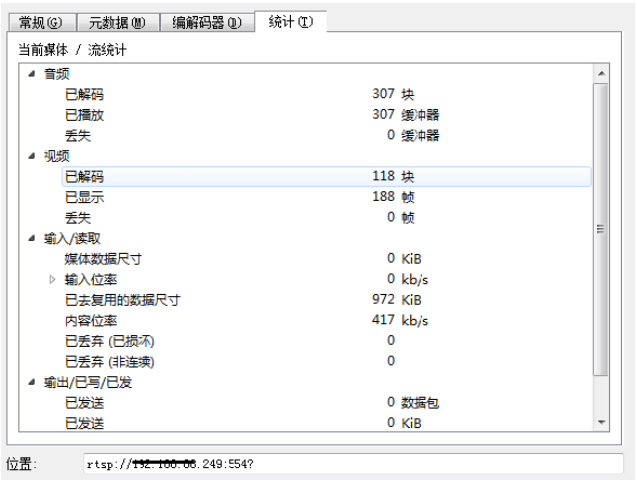

Figure 4 Statistics data of client when playing the video with KMPlayer from RTSP server

3. Build the ADT Development Environment

Use JDK (better upon level 5) to design a JVM. The next step is to install Eclipse, build the EV (Environment Variable). For previous windows platform which is easily than windows 7(such as Windows XP), we will manually set the environment variable's value [5].

A. Built JDK.

Download the JDK (the latest are better) from http://www.oracle.com/technetwork/cn/java/javase/downloads /index.html?ssSourceSiteId=null , choose the windows version. If you have installed JDK, just wait to do the next. The system path is required to configure under windows 7.

B. Built ADT

- Download the latest ADT version form http://developer.android.com/tools/sdk/eclipse-adt.html and install it. Then open it and set the workspace path.
- The ADT contain the latest Android API. If you want to download APIs of other versions, just use ADT's Android SDK Manager to download and install APIs you want. Figure 5 shows the Android SDK Manager's GUI.

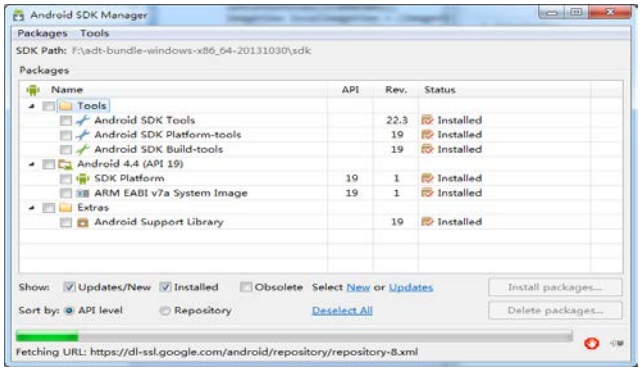

Figure 5 Android SDK Manager's GUI

- The following procedure is to open a device to run the "RTSP Server" application. In ADT, you can built a

virtual device or even use your Android devices. Since the PC and virtual machine share the same hardware. An important principle of setting virtual machine's performance is applying appropriate portion of the whole resources to virtual machine. Excessively high configuration of virtual machine lead to the decrease of system's performance or even crack because computational intensive application would cost a lot of CPU and RAM resources. When using physical devices, you must ensure that the devices' drivers are installed correctly. Windows. Figure 6 show ADT's Virtual Device Manager.

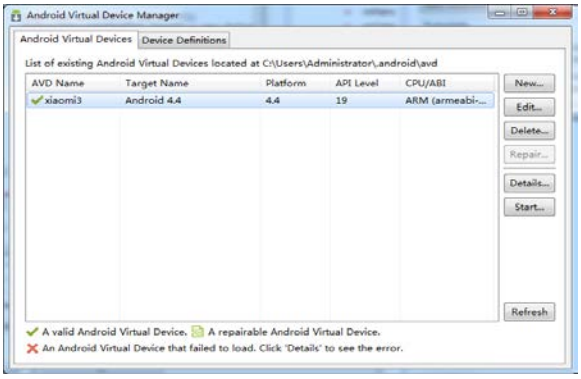

Figure 6 ADT's Virtual Device Manager's GUI

A. Other Development Environment

Browse http://www.ffmpeg.org/download.html, and download encode and decode core on Windows with Java. Then download RTSP project and install the SDK to Windows. Finally, install VLC open-source project's SDK.

Attentions that before you install or build the environment, look through the documentation form every project. If you ignore the document, you will probably fail.

4. Procedure of Developing

After the environment built, we can begin to develop from the Part I to Part III.

A. Conrtol Android devices

In Windows, most hardware can be used with Microsoft's API from MSDN. In Android, we can also use Android API to control the device.

To control the device such as camera, network card, storage, you should first get the permission from the device [6]. It is very important to understand how Android ADT gets the permission of device. Android use "AndroidManifest.xml" to manage the permission when the application accesses the related devices.

B. Generating the Network Stream

Programmers can make an API to use RTSP to generate stream from network. Import RTSP's code or project, open the key code and copy to the project. We can create a virtual network to transfer the coded stream [7].

Before transfer the stream, we store the stream data, in case of network fault and realize double buffering to make the data transfer better [8]. In Windows, the double buffer is widely used. If you develop with stream, you have to use double buffer to make a high Q&S app. Figure 7 is the double buffer's algorithm flow map [9]. The RTSP client also use double buffer to play the buffer stream when getting the stream data.

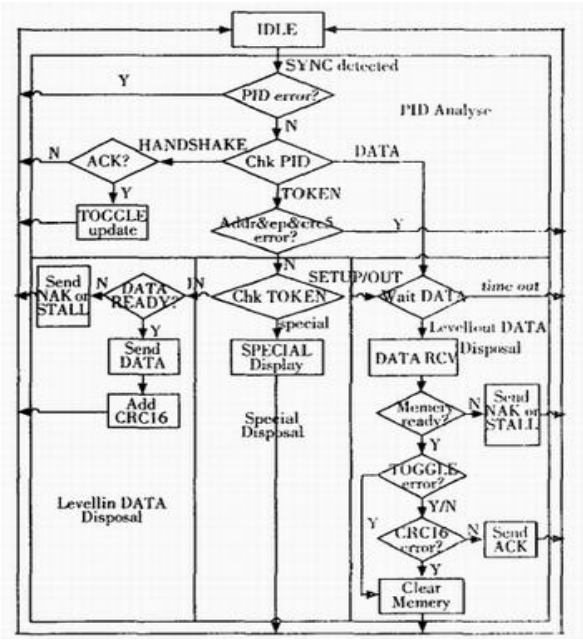

Figure 7 Double Buffer's Algorithm Flow Map

C. Transport and Decode

1. Transfer the stream data with RTSP protocol is the best way on this app.

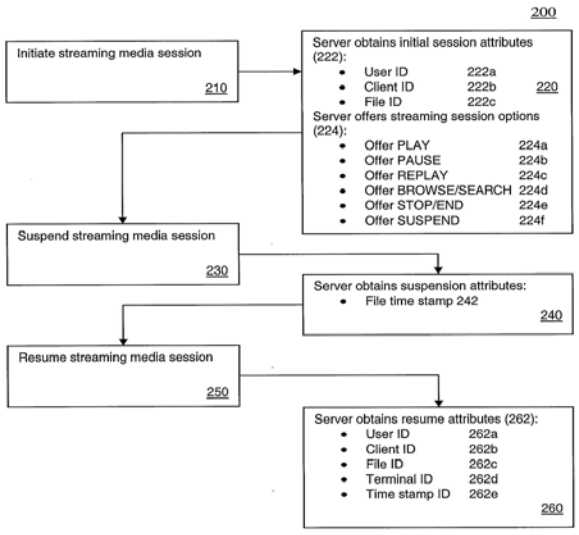

Figure 8 Main procedure of RTSP Protocol.

From the Figure 8[10], we show the virtures of RTSP protocol which help us capture the stream data efficiently and offer us some APIs to check and control the transfer channel. Additionally, it costs a small amount of network workload.

2. Decode with FFmpeg

Decode scheme is on RTSP client, but it is important to show the RTSP result. There are many RTSP players to play the RTSP streams such VLC, MX Player. All the players use FFmpeg's decode core, so we can use VLC play the RTSP stream from our RTSP Server[11].

5. Realize High Q&S

In our experiment, we use the Double Buffering, transfer the coded data, and use the buffer pool & connection pool to realize High Q&S of RTSP Server. Figure 9 shows the statistics data of client when playing the video from High Q&S RTSP Server.

Future improvements on our constructed High Q&S of RTSP Server can be conducted through the following aspects.

(1)Improve the network bandwidth.

(2)Utilize some parallel algorithms to get the startup of server's performance.

(3) Use CDN to speed up [12].

Figure 9 Statistics data of client when playing the video from High Q&S RTSP Server.

Acknowledgements

The paper is supported by the Plan Project of Beijing College Students' Scientific Research "Research on improving the robustness of streaming media server" (B2012017);, National Science and Technology Supporting Program (2012BAH37F02); Engineering Planning Project for Communication University of China (3132013XNG1325).

References

[1] Voznak, M., Rezac, F., Zdralek, J.: Danger Alert Communication System. In: IWSSIP 2010 - 17th International Conference on Systems, Signals and Image Processing, Rio de Janeiro, Brazil, June 17-19 2010, ISBN 978-85-228-0565-5.

[2] http://developer.android.com/tools/sdk/eclipse-adt.html

[3] http://zh.wikipedia.org/wiki/Android

[4] VLC Project. VLC Wiki – libVLC [Online]. Available. http://wiki.videolan.org/libvlc [Accedssed: Jan.9,2014].

[5] J. Steele, N. To, "The Android Developer`s Cookbook: Building Applications with de Android SDK", Addison-Wesley Professional, First edition, October 27, 2010.

[6] J. Y. Khan and P. Das, "MPEG4 Video over Packet Switched Connection of the WCDMA Air Interface" Proc. of the 13th IEEE International Symposium on Personal Indoor and Mobile Radio Communications, vol. 5, pp. 2189 – 2193, Sept. 2002

[7] C. Li, "Blur the boundary between the virtual and the real," IEEE J. Compul. Small Coil., vol. 24, pp. 39-45, Jan. 2009.

[8] W. K. Fong, S. W. Ng, B. H. Leung, and C. Surya, "Characterizationof low-frequency noise in molecular beam epitaxy-grown GaN epilayersdeposited on double buffer layers,"J. Appl. Phys., vol. 94, no. 1, pp.387–391,Jul.2003.

[9] http://forum.eaw.com.cn/event/images/2044.jpg

[10] http://patentimages.storage.googleapis.com/US20090037596A1/US20090037596A1-20090205-D00002.png

[11] WU Jin, HE Hui, HONG Hui. Research on Real-time Transport Technology of Multimedia Data Flow. Communications Technolog.

[12] CDNs and P2P, http://www.pandonetworks.com/node/185, 2007.