Research on Transmission System of Video Monitoring for theater Based on Live555

Hui Ren[1, 2, 3], Liu Gao[1, 2, 3] and Zhen Li[1, 2, 3]

[1]School of Information and Engineering, Communication University of China

[2]Key Laboratory of Acoustic Visual Technology and Intelligent Control System, Ministry of Culture

[3]Beijing Key Laboratory of Modern Entertainment Technology

Beijing, China

E-mail: renhui@cuc.edu.cn

Keywords: RTSP, streaming media server, Live555

Abstract. At present, in the transmission system of theater video monitoring, it usually uses the Winsock interface to set up streaming media server, which leads to the defect that the remote monitoring terminal of this system must rely on the Windows operating system, but not the smart TV based on Android operating system. To overcome the above disadvantage, this paper puts forward a solution about using Live555 open source library to set up the streaming media server.

Introduction

At present, in the transmission system of theater video monitoring, the theater remote monitoring terminal must be PC, but not the Android smart TV. In order to solve the above problem, this paper adopts the Live555 open source library based on RTSP and RTP/RTCP protocol to build the streaming media server, which exploits the Live555 open source library for the second time to achieve the real-time playback and switch function of monitoring video.

Real-time Video Transmission Protocol

RTSP Protocol. RTSP (Real-time Streaming Protocol) is an application layer protocol, which is used to transmit video data. The data source can be real-time video data or media files. RTSP provides the scalable architecture to control the online playback of video data, which can also simultaneously control the multiple data transmission.

There are two kinds of messages about RTSP, including the "request message" and "response message". RTSP request message is initiated from the client to server, including the following categories:

（1） Option: client sends the "Option" request message to inform the server about the corresponding interactive method.

（2） Describe: client sends the "Describe" request message to the streaming media server. Then the server does the permission validation, and responds the request message through sending the corresponding media information to client. The client is registered in the streaming media server.

（3） Setup: the client sends the "Setup" request message to inform the server about protocol name and port number.

（4） Play: the client sends the "Play" request message to inform the server about sending the video data.

（5） Teardown: the client sends the "Teardown" request message to inform the server about closing the real-time media transmission session.

Firstly, the server receives and analyzes the request message from the client. Then the server will send the different corresponding response message to the client according to the different request message [1,2].

RTP/RTCP Protocol. In the network service of video multiple transmission, in order to compensate for the shortcomings of traditional UDP protocol that can't transmit the data reliably, RTP protocol emerges at the right moment. It adds the RTP head to UDP data to ensure the reliable transmission of video packets. At the same time, RTP provides the end to end network communication service, suitable for setting up the streaming media server in real-time [3].

RTP protocol does not provide the resource reservation for the real-time video, and also can't guarantee the QoS, which leads to RTCP protocol emerges. RTCP protocol can provide the solutions for the network delay and congestion problem [4].

Implementation of the Transmission System of Theater Video Monitoring

The Overall Framework of Transmission System of Theater Video Monitoring. The transmission system of theater video monitoring proposed by this paper is consisted of three parts, including the IP network camera, streaming media server based on Live555, and the multiple remote monitoring terminals. The architecture chart of transmission system of theater video monitoring refers to Fig. 1.
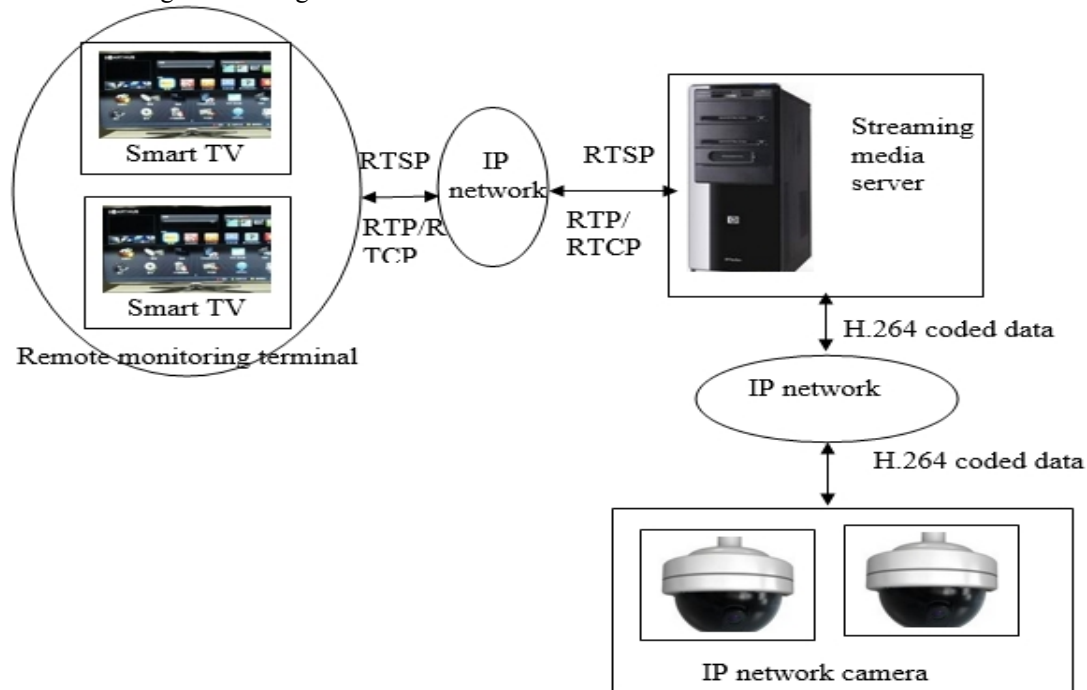

Fig. 1 Architecture chart of the transmission system of theater video monitoring

The video resource of streaming media server comes from the IP network camera. So how to extract video data from the IP network camera is vital important. This paper uses the SDK kits provided by the IP network camera company for the second exploitation. At the same time, it states that the video data from the IP network camera is coded to H.264 data, which does not need the second coding to transmit in network.

The IP network camera has the settlement IP address and port number, while the different network camera have the corresponding IP address. So the client uses the IDeviceProxy class provided by SDK kits to access the video data of IP network camera after the physical connection.

In the process of data acquisition from the IP network camera, firstly, the client receives the video data through calling the VideoInputsfunction of IDeviceProxy Interface. Secondly, there stores the video data in the buffer. Finally when the remote client (Android smart TV installing VLC player) sends the access request message to streaming media server through inputting the link of rtsp: // IP address, the server start to send the real-time video data to client.

Streaming Media Server. This paper designs the streaming media server based on the second exploitation of Live555 open source library. The server can realize the real-time transmission and remote playback of H.264 video data, and provides the selection buttons to switch the video data from different IP network cameras.

Live555 open source library includes four basic libraries, all kinds of test code and Live555 Media Server. The four basic libraries are UsageEnvironment&TaskScheduler/ Groupsock/ LiveMedia/ BasicUsageEnvironment. Among of these, UsageEnvironment library is used for message input& output and user interaction. TaskScheduler library is used for the event scheduling and reading the handle settings of events asynchronously. Groupsock library is used as the network

interface for sending and receiving video data packets, which realize the real-time video transmission in IP multicast so that improving the transmission efficiency. BasicUsageEnvironment library achieve the console application of UsageEnvironment library. LiveMedialibrary declares an abstract class—Medium, from which all other classes are derived [5, 6]. These inheritance classes are described as following:

（1） RTSPClient class: this class is used for sending the corresponding response message according to the RTSP request message, and creating the corresponding RTP session based on the parse results finally.

（2） MediaSession class: this class is used for creating one RTP session named as the MediaSession, which may contain multiple MediaSubSession classes.

（3） RTCPInstantance class: this class is based on RTCP protocol.

（4） Source& Sink classes: Source class is the abstract data source, and Sink class is the abstract data consumption. The direction of data flow is from the Source to Sink.

Establishment and Response of Streaming Media Server Based on Live555. In this paper, the streaming media server is set up through the second exploitation of Live555 open source library. The flow chart of streaming media server refers to Fig. 2.
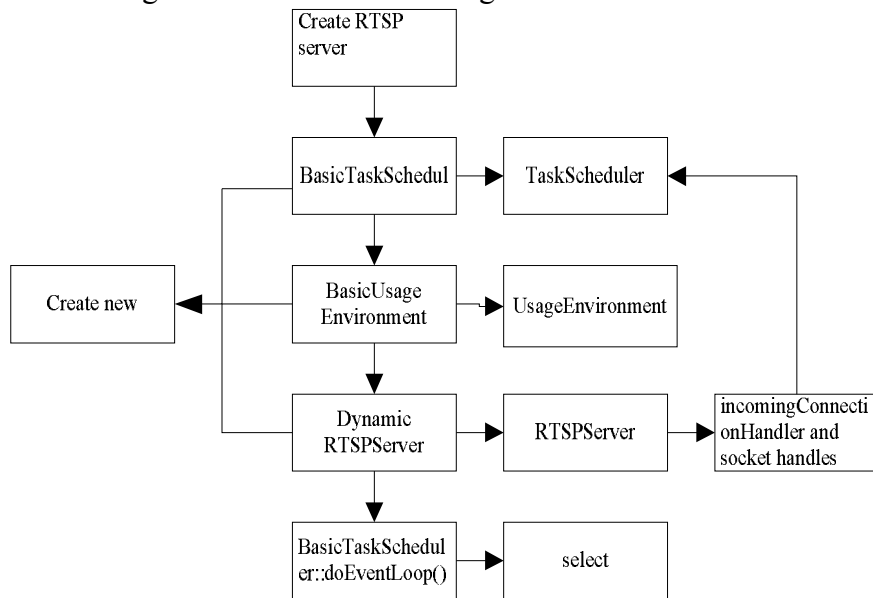

Fig. 2 The flow chart of streaming media server

The BasicUsageEnvironment class is used to set up the task scheduler; the UsageEnvironment class is used to establish the operation environment of program; the RTSPServer class is used to set up RTSP server. The doEventLoop() function is used to do event cycle, which is mainly responsible for three things: listen and respond to the request message from the client, event process, and schedule for the delay task. The process of server building is as following:

（1） Firstly, RTSP server is set up by calling the Dynamic-RTSP Server class. In this process, there creates Socket listening on port 554 based on TCP, then sends the handles of incomingConnectionHandler and Socket to the task scheduler.

（2） Secondly, the task scheduler will associate the handle of incomingConnectionHandler and the handle of Socket.

（3） Finally, the main program will come into the doEventLoop() function to monitor and wait for request command of multiple clients through calling the select() function.

If the streaming media server finds the client request message, it will immediately look for the corresponding incomingConnectionHandler, then set up the RTSP ClientSession class to create the client session. The process of server response is as following:

（1） After receiving the "Describe" request message from the client, the server will look for the corresponding streaming media information according to the handle function. If there isn't the information, the server will setup a new ServerMediaSession class. Then ServerMediaSession class will assemble the SDP information and send to the client.

（2） After receiving the "Setup" request message from the client, the server will parse the head information of corresponding request message according to the handle function, and access to streaming media

transmission parameters. These parameters are packaged into response message and sent to the client. During the process, the Source and RTPSink classes are set up.

（3） After the server receives the "Play" request message from the client, Source start to send video data to the Sink.

Implementation of Real-time Video Data Transmission. If RTSPSink sends the request information, the real-time video will be sent from Source to Sink. The streaming media server based on Live555 open source library can't send originalH.264 real-time video data, therefore this paper add the new class--H264myFrameSourceinheriting from the FramedSource class, which can access to the H.264 real-time video data from IP network camera using the way of circular buffer. The process of sending data from streaming media server to remote monitoring terminal refers to Fig. 3.
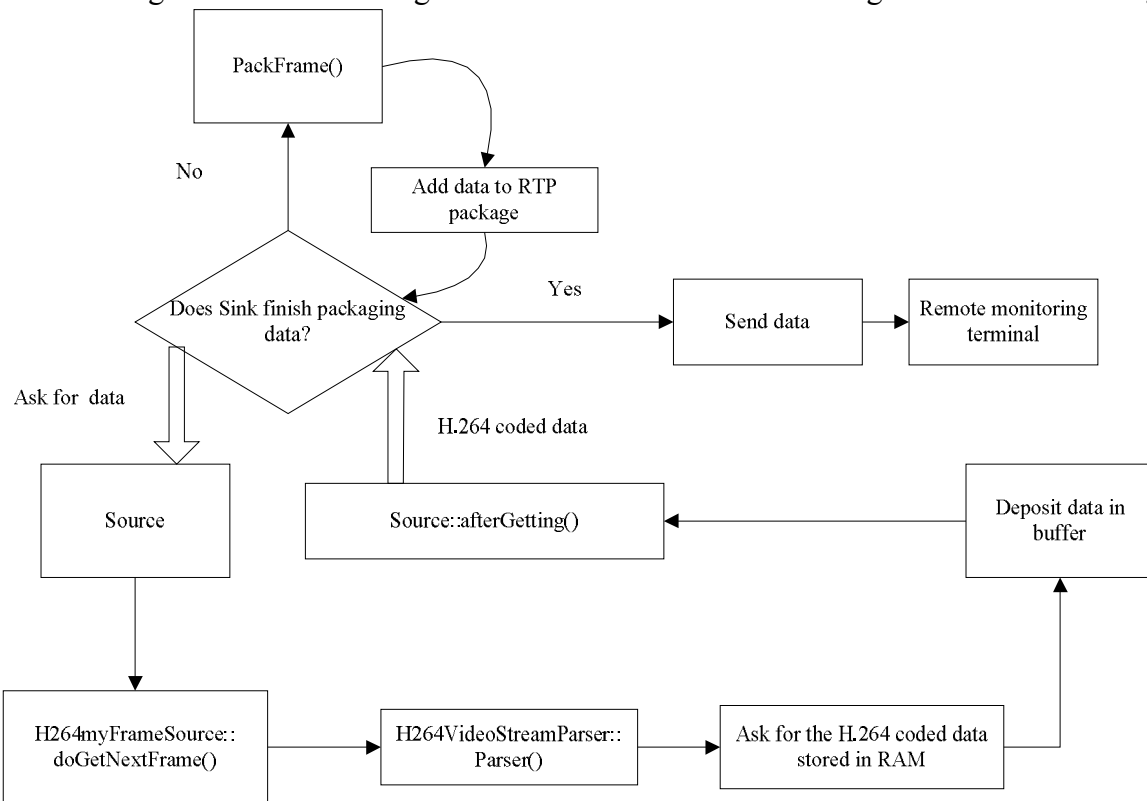
Fig. 3 The flow chart of requiring and sending data

Firstly, the client sends the "Play" request message. In response, RTPSink asks for video data. Secondly, If there isn't data in the input buffer, the Source class calls the H264myFrameSource::doGetNextFrame() function, then calls the H264VideoStreamParser::Parser() funcation to access to the H.264 real-time video data in the buffer. Finally RTPSink packages the video data and sends to remote monitoring terminals.

Summary

This paper introduces the method of how to acquire the video data from IP network camera, and setup the streaming media server based on Live555 open source library. Based on the above method, this system implements the real-time video transmission and switch. In the system, there is Android smart TV as the remote monitoring terminal. Users can use VLC app installed in the Android smart TV to watch monitoring video. With the improvement of Live555 open source library, the streaming media server based on Live555 open source library will have broad prospect in the transmission system of theater video monitoring.

Acknowledgment

References

[1] Yong Cheng Y, Yuan Zhi Y (2003). IP multicast and its application based on Winsock [J]. Computer Application 23(6):247-250.

[2] Guo Hao P. (2007). Research and implementation of video transmission system based on IP multicast [J]. Computer and Modernization 142(6):81-84.

[3] Dong Xing J. Windows Sockets Network programming [M]. Beijing: Tsinghua University Press, 1999.

[4] Xiao Lian C. (2003) Research of the digital video transmitting system based on IP multicast [D].

Nanjing: 01-68.

[5] Jian C, Wei C. Web senior programming based on Visual C++ [M]. Beijing: Post& Telecom Press, 2001.

[6] Information on http://www.ArtisticLicense