2012 Fourth International Conference on Computational Intelligence and Communication Networks

A Mobile Real-time Video System Using RTMP

Pengyu Zhao
Institute of Educational Technology
Beijing University of Posts and Telecommunications
Beijing, China
zpy888134348@hotmail.com

Jianwei Li, Jianxiao Xi, Xuerong Gou
Institute of Educational Technology
Beijing University of Posts and Telecommunications
Beijing, China
jwli321@126.com

***Abstract*—This paper introduces a mobile real-time video system which is different from other existing real-time video system that is mainly based on wired computers. The system proposed in paper is basically designed to enable users under wireless circumstances to publish what they are capturing with their mobile devices and also watch what others are capturing. The system mainly contains mobile devices with a Flex [1] client application on them, a media server which handles live streams using RTMP [2] (Real-Time Messaging Protocol), and a web server which used to deal with HTTP requests.**

Keywords- real-time; mobile; Flex; RTMP

I. INTRODUCTION

Today there are many real-time video system using various technologies and protocols. People can watch, for example, live NBA games not only in front of TV, but also in front of a computer. However, there is a trending that mobile devices are becoming more and more important because their mobility enables people to use them whenever and wherever they need them. Besides, in existing real-time video systems, users are usually receivers who receive what is provided, rather than providers who are able to share precious moments to others in a real time way.

This paper comes up with a real-time video system that enables people to not only watch live video with their mobile devices but also be providers who make their moments live and share with friends and families by using mobile devices. There are mainly three parts in this system:

(1) Media server: Media server is a server that deals with live video streaming including input streaming which comes from providers' mobile devices and output streaming distributed to receivers who request live videos. The choice of media server is fundamentally based on the protocol used in this system to transmit live video streaming. There are many protocols for streaming media, such as RTP/RTCP [3], RTSP [4], RTMP, and so on. Because the protocol chosen here is RTMP, we use Red5 [5] which is primarily designed for RTMP as our media server.

(2) Flex client application: The Flex client application mentioned here is software application installed on mobile devices and used as client for users to communicate with media server and web server. The reason we use Flex which is provided by Adobe to develop the client is that it has convenient interfaces of video encoding and transmission when the video file format is FLV and the transmission protocol is RTMP (FLV and RTMP are also developed by Adobe).

(3) Web server: Web server is a server that deals with HTTP requests which contains commands of users, such as signing in, getting live video list, and so on. Also web server is in charge of managing database according to requests from users.

This article is organized as follows: Section II addresses about the related work. Section III addresses the system architecture and shows each part of the system in details. Section IV addresses the system implementation to present the whole process of how the system works in details. Section V draws conclusions and introduces some future work.

II. RELATED WORK

A. Protocols

There are mainly two popular kinds of basic protocols used for streaming media today, (1) RTP/RTCP which can work on both UDP and TCP but mainly UDP, and (2) RTMP which only works on TCP.

RTP (Real-time Transport Protocol) defines a standardized packet format for delivering audio and video over IP networks [6]. RTP is primarily designed for video teleconference applications. Now it is widely used in television services, web-based push-to-talk and telephony. For RTP, RTCP works as a controller which monitors transmission statistics and quality of service (QoS) and helps to synchronize multiple streams. Because RTP is mainly based on UDP which is simple and fast but lack control over congestion of network, there are many researches dealing with congestion control of RTP using RTCP. And as its unfair competition with TCP streams in bandwidth, many kinds of TCP-friendly protocols emerge to enable media streams based on UDP to be smoother and compete more fairly with TCP streams, such as TFRC [7] (TCP-Friendly Rate Control).

RTMP is a TCP-based protocol for streaming audio, video, and data over the Internet smoothly between a Flash player and a server. As it is based on TCP, the congestion control is much better than RTP based on UDP.

B. Applications

The most common applications using streaming media technology are video teleconference, VOD (Video on Demand), live video, and video call.

978-0-7695-4850-0/12 $26.00 © 2012 IEEE
DOI 10.1109/CICN.2012.18

Video teleconference is designed to serve a conference or multiple locations rather than individuals. Video teleconference is widely used in many fields. For instance, in the field of law, videoconferencing has allowed testimony to be used for an individual in the United States; in the field of education, videoconferencing enables students to learn by participating in two-way communication forums, and remote virtual class can be more like a real one, for example, an enhancing learning system proposed in [8]; in the field of business, videoconferencing provides individuals in distant location an opportunity to participate in meetings on short notice with money and time savings.

VOD are systems that allow users to select and watch video content on demand. An example using VOD is IPTV which brings video on demand to televisions and personal computers.

Live video system refers to a system which is able to publish a live video via Internet with supports of streaming media technologies. A popular instance is SINA NBA live games [9] which provides live NBA games on Internet.

Video call is a call that allows two sides in this call to see each other on telephone or computer. A popular application for video call is Skype [10] that provides service which allows users to communicate with peers by voice, video and instant messaging over the Internet.

III. System Architecture

The real-time video system proposed in this paper is a client-server system. The client side is mainly referred to mobile devices such as cellphone, pad, which can be both source and destination of live video streaming while users can also watch videos on a computer by visiting the website running on the web server in this system. The server side contains two parts, media server and web server, to handle RTMP and HTTP requests respectively. The overall system architecture is shown in Figure 1.

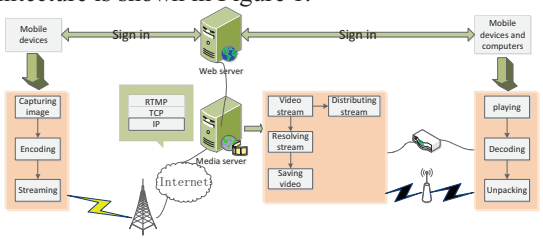

Figure 1. System architecture

A. Client side

We have developed a software application for mobile devices using Flex 4.5 SDK. The application runs on Adobe Air [11] (Adobe Integrated Runtime) which is a cross-platform runtime environment that supports some mobile operating systems such as Android, BlackBerry Tablet OS and iOS. With this application, users are able to put live what they want to share with others or watch others' live videos.

As presented above, the application contains two functions: (1) publish a video (2) watch a video. Publish a video is a function that enables users to capture video with camera on mobile devices, then encode video and finally stream it to media server (packing model is shown in Figure 2). In order to enable others to see the video afterward, the video can be saved on media server while it is live. Watch a video is a function that enables users to play a live video captured by others on mobile devices. When playing a live video, mobile devices receive streaming from media server and handle streaming data with flash player, then decode it so that users can watch it.

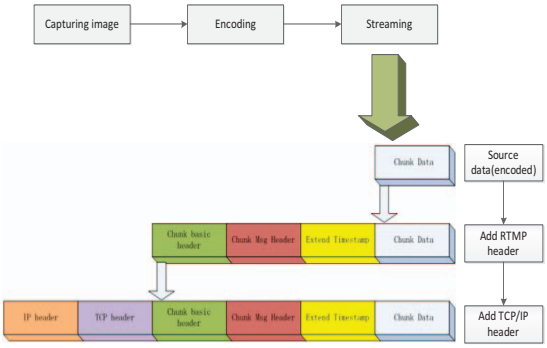

Figure 2. Video streaming

Therefore the Flex client application serves as an encoder, a streamer when users are capturing live videos, and a stream dealer, a decoder when users are playing live videos.

Besides the Flex client application, there is a website serves as another access to live videos. This website built on web server enables computer users to watch or review videos captured by others or themselves.

B. Server side

There are two parts on server side, one is media server, and the other is web server. Media server is specialized in tackling live video streaming because the protocol used in video streaming is specially designed for media streaming. The media server we choose here is Red5 which is a well-designed open-source media server for RTMP. In media server, every video stream has its unique id. When stream is received, it begins to resolve stream according to protocol and save the video data in formats that users choose. When other users require playing live videos, the media server distributes stream to users based on stream id.

Web server is mainly in charge of handling HTTP requests. The Flex client application installed on mobile devices and the web pages mainly designed for computer users share the same back-end program on the web server. When operation of users on client application and web pages send HTTP requests to back-end program, the program reads or changes data in database and then makes response to client application and web pages. The main functions of the web server contain user management and video management. User management is basically in charge of managing user's personal information and user's friends while video management is responsible for administrating video information, such as starting time, duration, author, comment, and so on.

IV. SYSTEM IMPLEMENTATION

To introduce system implementation, we mainly introduce two parts: (1) streaming a live video (2) play a live video. Also in this section we give a photo taken when this system is running.

A. Streaming a live video

The process of streaming a live video involves both RTMP communication between mobile devices and media server, HTTP communication between mobile devices and web server. The steps of this process are shown as follow:

1) Build connection between mobile devices and media server.

2) If success of connection is returned by media server, we can continue to establish stream on the connection we just built in 1) step; if fail to build connection between mobile devices and media server, we are not allowed to begin establishing stream.

3) Users can name the video whatever they want and when the video is named, client application send the name to web server which is responsible for creating a data record in database in order to attain a unique id as the unique id for the stream of this video.

4) Encode data of image captured by camera on mobile devices, and then pack it into packets according to RTMP. Add data of packets into stream established in 2) step and send it to media server.

5) We can stop streaming video by cut off the stream established in 2) step.

The calling sequence in program is shown in Figure 3.

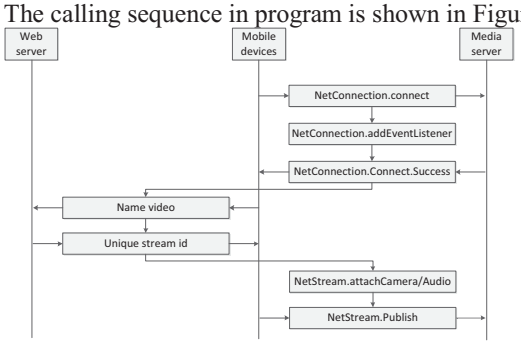

Figure 3. Streaming a live video

B. Play a live video

The process of playing a live video also involves both RTMP communication between mobile devices and media server, HTTP communication between mobile devices and web server. The steps of this process are shown as follow:

1) Users can get a live video list as well as some video information from web server and choose the live video they want to play. Once a video is chosen, client application obtains its unique stream id from video information.

2) Build connection between mobile devices and media server.

3) If success of connection is returned by media server, we can continue to establish stream on the connection we just built in 1) step; if fail to build connection between mobile devices and media server, we are not allowed to begin establishing stream.

4) Create a video object and the stream established in 3) step is attached to this video object so that the stream data, if any, can be translated into video that users can watch by unpacking packets and decoding.

5) Receive stream data from media server by noticing media server the unique stream id of the video ordered.

The calling sequence in program is shown in Figure 4.

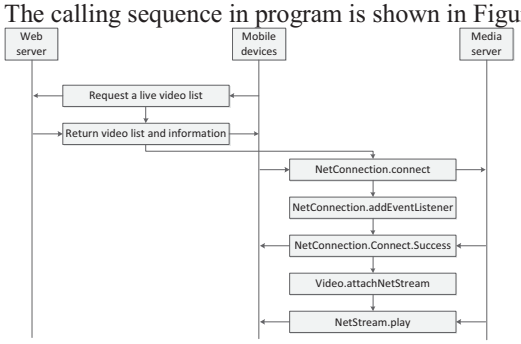

Figure 4. Playing a live video

C. Result

As the photograph shown in Figure 5, the left cellphone is publishing a live video using Flex client application; the right cellphone is receiving this live video using Flex client application; the computer is receiving this live video on website.

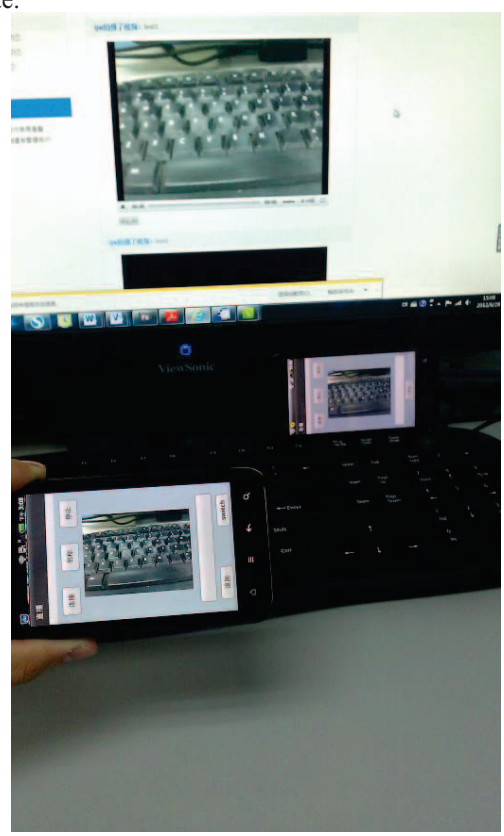

Figure 5. Photograph taken when system is running

V. Conclusion and Future Work

This paper presents a mobile real-time video system using RTMP. This system mainly intends to enable users to publish live videos and watch live videos of others with mobile devices. Also users can watch live videos with computers on website we built. The propose of designing this kind of system is different from both video calls which happens between only two persons and watching live broadcast in front of a computer which lacks of mobility and individuality. Rather, this system basically realizes that people can share things immediately whenever and wherever they want in form of live video if they have access to network.

In future work, firstly, as this system is a client-server system, the bottleneck of system mainly underlies in the capability of media server, that is, when the number of user grows to a level that surpasses the capability of media server, media server may run down. Secondly, as bandwidth is limited, especially the bandwidth for streams from media server out to users, we need to do further research on video encoding to improve compression ratio of video data before being transmitted. Thirdly, feedback from users may contain some advisable suggestions and some bugs, according to which we will improve our system.

Acknowledgment

Wireless Ubiquitous Business Environment, Architecture, Key Technology Research and Demonstration Based on Web (No. 2012ZX03005008); the Fundamental Research Funds for the Central Universities

References

[1] Flex introduction. http://www.adobe.com/pr oducts/flex.html

[2] RTMP Specification 1.0. http://wwwimages.adobe.com/www.adobe.com/content/dam/Adobe/en/devnet/rtmp/pdf/rtmp_specification_1.0.pdf.

[3] RFC 3550, Standard 64, RTP: A Transport Protocol for Real-Time Applications.

[4] RFC 2326, Real Time Streaming Protocol (RTSP), IETF, 1998.

[5] Red5 introduction. http://www.red5.org/.

[6] http://en.wikipedia.org/wiki/Real-time_Transport_Protocol.

[7] RFC 5348, TCP Friendly Rate Control (TFRC): Protocol Specification.

[8] Wichian Premchaiswadi, Anucha Tungkasthan, Nipat Jongsawat. Enhancing Learning Systems by using Virtual Interactive Classrooms and Web-based Collaborative Work. IEEE EDUCON Education Engineering 2010¬—The Future of Global Learning Engineering Education.

[9] http://live.video.sina.com.cn/room/nba/.

[10] www.skype.com/.

[11] Adobe Air introduction. http://www.adobe.com/products/air.html.