Streaming player support protocol adaptation and independent operating system

Sangok Kim, Kanghee Lee, Zhefan Jiang, Hyunchul Bae, Sangwook Kim
Department of Computer Science, Kyungpook National University
1370 Sankyuk-dong, Buk-gu, Daegu, 702-701, Korea
{sokim, khlee, zfjiang, hcbae, swkim}@cs.knu.ac.kr

Abstract

This paper examines technique of media access that flexible media service, provided user interaction, and guaranteeing QoS. Currently, there are various streaming services, but the service environment is limited from specific operating system, each protocol. The media player on the real-time protocol have some problems that media access techniques depend on each network protocol, and the enable commands to control media are different from protocol environments. We propose media player that is able to streaming service using by adaptation on the various protocol and independent operating system. Therefore, we analysis streaming protocol for real-time media service and implement control mechanism of player for protocol.

1. Introduction

In this paper, we introduce media service techniques for real-time streaming service of web browser, and describe contents of implementation. The real-time streaming techniques and development of media codec lead to streaming service of web browser that media contents present high definition and high quality sound, the simple end users make use of streaming service of various contents in the web browser. The MPlayer[1] Plug-in[2] technology of Linux-based Mozilla web browser and the Windows Media Player[3] ActiveX control[4] technology of Microsoft Windows-based Internet Explorer are able to streaming service the media.

However, these techniques depend on their web browser of operating system because service environment has limited Linux and Windows. If we want to use web browser of Linux environment on the Windows operating system, the media service doesn't support. Because the Microsoft Corporation developed and closed project, these cases have problems. So, we propose streaming media player that is able to service media in the heterogeneous operating system. The various media players that supported streaming protocol are able to service real-time play.

Nevertheless, the most of players don't support function of player on the browser of various protocols. In this paper, we describe media service techniques that the streaming media player for Windows and Linux environments, propose a new seamless method that avoids these dependency. The remainder of this paper is organized as follows.

In section 2, a streaming technique for embedding [5] web browser each operating system and analyze protocol to support streaming service. In section 3, we demonstrate some examples of media player with our proposed control method. Finally we conclude our paper and present an outlook to future work.

2. Streaming techniques

The streaming techniques recently compose that protocol supporting real-time delivery and media player.

2.1. Analysis streaming protocol

Currently streaming protocols are: HTTP (Hypertext Transfer Protocol), MMS (Microsoft Media Server), RTSP (Real Time Streaming Protocol). We analyze and examine MMS for media server that to support high-level security, stable streaming service. The MMS protocol consists of command packet for presentation about contents and media packet included real media information.

2.1.1. Command packet. The command packets play an important role in protocol configuration, stop, seek and accept request. The standard fields consist of 40 bytes and append needed information according to

Proceedings of the Fourth Annual ACIS International Conference on Computer and Information Science (ICIS'05)
0-7695-2296-3/05 $20.00 © 2005 IEEE

what kind of commands. Table 1 shows the contents of in front of 40 bytes. The end of second field presents role of command in total data.

2.1.2. Media packet. The media packets include information about media contents that consist of header and real data. The MMS protocol consists of several packets for streaming service. These packets delivery through MMS protocol contained header information in Figure 1.

Table 1. Command packet (40bytes)

4bytes = 01 00 00 [00]	Common format is "00 00 00 00", value of the 4th has irregular value
4bytes = CE FA 0B B0	Current packet is "Command" and fixed "CE FA 0B 0B"
4bytes	This field represents length of the left command packet
4bytes = 4D 4D 53 20	Protocol type is "MMS"
4bytes	If the end of packet length is 8bytes, the field is "1"
4bytes	This field represents "Sequence number". The value starts "0" and increase count "1".
8bytes	Timestamp values for network
4bytes	If the end of packet length is 8bytes, the field is "1"
2bytes = Command	Command value
2bytes = Direction	The client's message is "3", server's message is "4"

4bytes Sequence number	1byte PacketIDtype	1byte UDP sequence TCP flag	2bytes Packet length

Figure 1. Header information of media packet

The Sequence number means the order of packets. If the PacketIDtype is “2”, these packets imply header information of media and case of “4” means data of contents. The UDP sequence means the order of packets for UDP, The TCP flag informs location of length. If this value is “0”, the middle data, case of “4” means the first date, and case of “C” exists only one data in the packet. Finally, The Packet length means total length of transmitted data.

2.2. Streaming techniques

The streaming service for web browser demanded that VCR control of player and steaming server, embedding technique for web browser. We explain adaptive method of detail module for media player in this section. We suggested that embedded media player for streaming service consists of various modules according to operating system (as shown in Figure 2). The Delivery Manager plays roles that configure to connecting media server, receive media data, send command data, and support interaction between user and player. The dataset received from server restore in the Stream Buffer, and the De-Multiplexer extract video and audio data from set of streams. The Decoding Buffer is temporary buffer, and structured on equality with Stream Buffer. The System Decoder decodes compressed video and audio data, and then restores to the original data. The Presenter display video frames on the web browser. This module plays data of buffer with presentation time stamp on the screen. The VCR controller is user interface for VCR processing. The GUI buttons for VCR control are similar to other players.

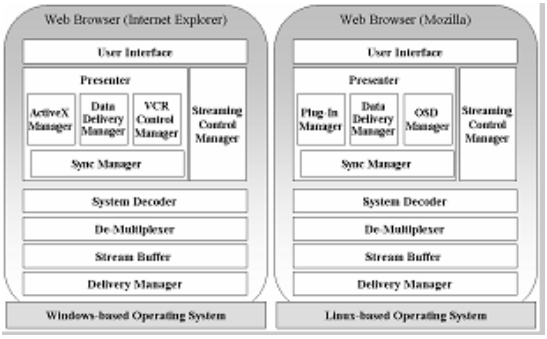

Figure 2. Structure of player

The Sync Manager controls time information of each data for media player. A timer of sync module started as reference system time for process. And this module controls time of media data by relative values of time. If the player arrived at value of time that configured to media, created system call for each module. Generally, a sync method is that tune a unit of frame for video to a unit of frame for audio, because data of audio composed continuous and data of video composed separative. The Streaming Control Manager processes user’s input: play, pause, restart, seek, and stop. If a user interfaces receive a command, this manager identifies what kind of command with ID of GUI and calls Delivery Manager with argument of value. The Delivery Manager transmits and not only reports adaptive messages of command by parameters of Streaming Control Manager but also manages system resources with command. The resources manage play roles allocate memory and delete on the

system. And these managements come into use thread. If a user plays contents, internal process creates a thread. And if "seek" event comes from user then a buffering thread is dead. The existed threads initialize resources that used before create a new thread. For this process, the stream, decoding and presentation buffer empty. These initializations of resources always take place in the control of "seek". The Plug-in Manager is an operating module between media player and web browser. Therefore it depends on an operating system. The Plug-in is able to install easily, and play a sound through web browser, download programs, configure, and define the function. Then, the web browser automatically recognizes the Plug-in Manager, and the execution codes of embedded player are inserted in HTML file. This manager plays important roles to setup distributive program and restore file of configuration. It controls messages for Internet Explorer GUI with Data Delivery Manager in the Windows. It creates thread that locates Mozilla web browser and fixes size, transmits commands to each processor. Therefore, it can directly control the media player. The process model of Plug-in manger is shown in Figure 3.These processes download cabinet file including ActiveX control and install media player to defined directory in the Windows.

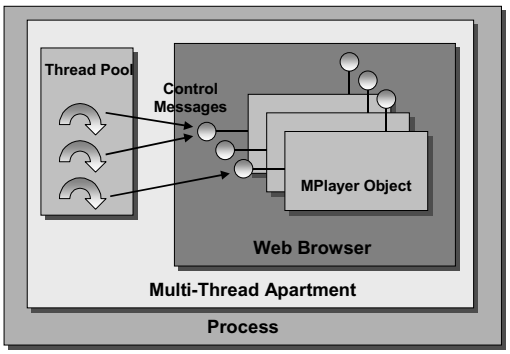

Figure. 3 Process model of Plug-in Manger

The Data Delivery Manager takes charge of transmission between Plug-in and media player. The embedded media player indicates important points of media content. Because, it is capable of seek that point and reuse. The script languages in the HTML codes easily delivery to control data of the player.

The OSD (On Screen Display) supports user interaction. The functions of OSD Manager present control data on the web browser and handle user's input. The OSD functions consist of display menus, easily control presentation, and show what kind of executive function. The web browser is display into place, the signal processor for user's input senses events by the button, and the indicator of data represents items from playing file on the list.

The OSD controller shows information for event list through the indicator of data. These menus consist of several functions that play, pause, seek menu, open directory, help, option, console mode, and quit. The Figure.4 shows streaming service of embedded media player with plug-in. If the player executes, it initializes for the system and the web browser, therefore creates threads of plug-in for delivery streaming service and command.

The threads included messages connect to server during open the streams. If the configurations of connection success between server and client, the client is able to receive previous data during process of buffering. Because the De-Multiplexer divides each stream, completes buffering and initializes codec. Since then, the player repeats decoding for unit of data, play decoded data, synchronize about video, audio, and handle information of events until playing is over, or input user's command of stream control.

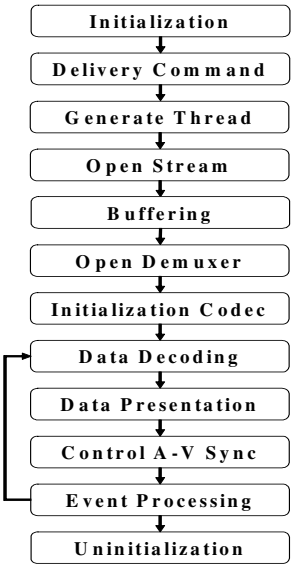

Figure 4. Process of the streaming service

3. Implementation

In this paper, we proposed the embedded streaming media player for the web browser that execute Windows 2000 Server (Build 5.0.2195) and Linux Fedora Core 1(Kernel 2.4.22-1.2115) environment, in the Figure.5 shown embedded media player for Mozilla web browser on the Linux, and in the Figure.6

shown demonstrated display for Internet Explorer on the Windows.

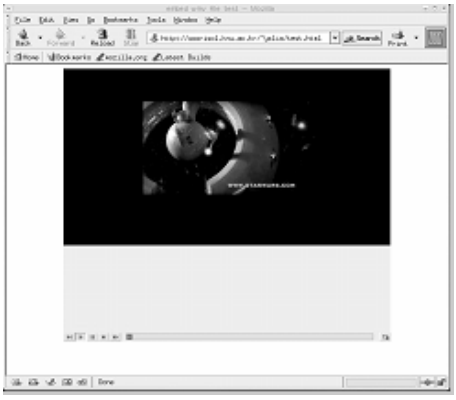

Figure 5. Mozilla

Figure 6. Internet Explorer

4. Conclusion and future work

We proposed the streaming media player that is capable of embedded on the web browser. The techniques of embedded media player construct ActiveX and Plug-in based open project, therefore implemented streaming media player is able to control VCR on the web browser. That player supports streaming service with various web browsers in the heterogeneous operating system, and user interacts by controller between server and client.

The future works are protection of contents, implementation of related modules, and services that is capable of transmission for contents adaptation [6]. Therefore, the embedded media player is optimized that supports various environments: set-top box or embedded OS (PDA, Cellular phone, Hand-held devices).

Acknowledgements.

This research was supported by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Assessment

5. References

[1] http://www.mplayerhq.hu/,. 2004.

[2] http://mplayerplug-in.sourceforge.net/., 2004.

[3] http://www.microsoft.com/windows/windowsmedia/default.aspx., 2004.

[4] Adam, D., "ActiveX Controls Inside Out 2nd Edition" Microsoft Press 1999.

[5] Phillip, S., Andrew, M., Jens, H., "Network-Centric Migration of Embedded Control Software A Case Study" Proceedings of the 2003 conference of the Centre for Advanced Studies conference on Collaborative research, pp. 54-65, 2003.

[6] Surya N., Uma S., "A system for quality driven adaptive video delivery" Proceedings of the 5th ACM SIGMM International Workshop on Multimedia Information Retrieval, pp. 223-230, 2003.