The Design of a NPVR System for an IPTV Platform

Tzung-Yu Wu
Delta Networks, Taipei, Taiwan
Eamil: Gary.wu@delta.com.tw

Ho-Ting Wu and Kai-Wei Ke
National Taipei University of Technology, Taipei, Taiwan
Email: htwu@csie.ntut.edu.tw; kwk@csie.ntut.edu.tw

***Abstract*—IPTV (Internet Protocol Television) is one of the main Internet applications developed in recent years. It provides an unprecedented variety of multimedia services in the form of streaming media, delivers real time contents to users directly, and supports new services such as user interactive TV programs. In this paper, we design and implement a cross-platform web IPTV systems based on the functional definitions of the OIPF (Open IPTV Forum) standard. We implement a few multimedia streaming functions, including video upload, NPVR (Network Personal Video Recorder), VoD (Video on Demand) and live video services.**

***Index Terms*—IPTV, Steaming Services, NPVR, VoD, OIPF**

I. INTRODUCTION

Due to the booming digital technology of the past few years, the convergence of telecommunication, broadcasting and Internet has become an inevitable trend to promote IP based triple-play services. Among these IP services, IPTV has attracted wide attentions [1]-[4]. Therefore, a lot of international standard organizations are formed accordingly to develop a series of related technical standards [5] [6], such as OIPF (Open IPTV Forum) [7], HbbTV (Hybrid Broadcast Broadband TV) [8] and the MHP (Multimedia Home Platform) [9]. IPTV may provide an unprecedented variety of multimedia streaming services and features to end users directly, such as VOD (Video on Demand, VoD), audio and video broadcast (Scheduled Content Service), time shift (Time-Shifting), network personal video recorder (NPVR), web browsing, e-mail and voice phone services.

With the advances of mobile network technology and the widespread of smart handheld devices, more and more people are relying on mobile handheld devices to replace PC or notebook computer for general internet services as well as streaming services. It is noted that the multimedia streaming services require much more bandwidth than the traditional internet data services. Fortunately in nowadays, mobile service providers can provide satisfied network bandwidth to support video streaming services, so people start to use video streaming service on a handheld device. For example, a HDTV movie requires streaming bandwidth of 8 to 15 Mbit / s, while the regular digital TV program encoded by MPEG schemes may require the stream bit rate of 1.5 ~ 6 Mbit / s. On the other hand, the LTE downlink channel can provide sufficient bandwidth up to 300 Mbit / s theoretically.

We built the IPTV system based upon the OIPF standard. There are two types of IPTV services defined in the OIPF standard, including the open Internet services and managed network services. The digital wireless TV broadcast programs are provided by scheduled content services in the OIPF standard using the multicast mechanisms. These programs are supported under the managed network scope specified in the standard. However, since it is difficult for a handheld device to receive the digital wireless TV programs without any extra attached TV tuner module, we thus provide such services via the Open Internet Service mode. Any client that can activate the internet web function will be able to enjoy the services provided by this system using the most common web browser mechanism. We implement a few multi-media streaming functions, including video upload, NPVR (Network Personal Video Recorder), VoD and live video services in this cross platform IPTV system.

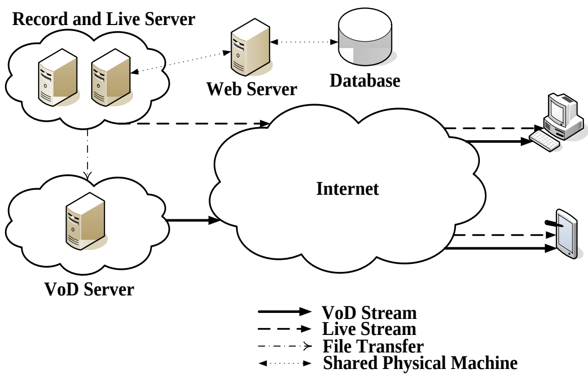

Figure 1. The Built IPTV System Architecture

Manuscript received March 29, 2013; revised May 30, 2013.

doi: 10.12720/ijeee.1.3.140-144

II. System Architecture And Implementation

The server system consists of 3 PC (personal computers), as shown in Fig. 1. There are two Record and Live Servers. One Record and Live server also serves as the Web Server and Database Server. The other Record and Live Server resides at another PC. The third PC contains the VoD Server.

The Web Server provides user interface for users to access this IPTV Service and also performs a few decision-making functions. The Database server is used to store information related to IPTV Service. The Record and Live Server not only performs video recording services but also provides rendering services of the live TV program. That is, it can also support real-time live video streaming service while performing the recording functions, so that the user is able to watch TV programs immediately without delay. The VoD Server stores the third party uploaded video programs as well as recorded TV video programs. Therefore, users can download video streaming data through VoD Service to watch the video.

A. VoD Service

The system architecture of the VoD Service is shown in Fig. 2. The stored VoD program table and related information is stored in the VoD database. When a user wants to watch VoD program, he accesses the VoD webpage of the web server for a list of available programs. The associated PHP engine of the web server is triggered to fetch the updated information of stored VoD programs from the VoD database, and then returns web pages back to users. The web page contains a list of available video programs with associated URL and information. The user can thus select a VoD program from the webpage, which redirects the request to the appropriate VoD server containing the requested VoD program. Upon receiving the request, the VoD server downloads the requested video streaming to users.

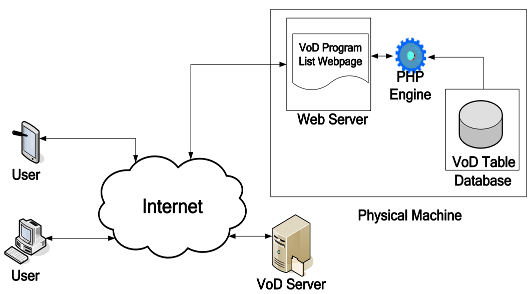

Figure 2. VoD Service Architecture

B. Video Upload

Our system, exhibited in Fig. 3, also provides the video upload function, similar to the video upload service in YouTube. Users may upload the video clips and associated information to the video clip upload control unit in the web server, where the related information is recorded in the upload data table. The scheduling system periodically activates the file conversion unit to convert the uploaded video clip into MP4 format by using the FFmpeg program. The converted MP4 video clip is then stored in the VoD server. The associated VoD table of the database is updated as well.

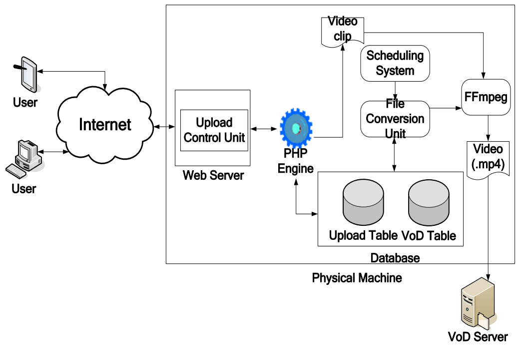

Figure 3. Video Upload Service Architecture

C. EPG

To realize the NPVR system, we must provide program schedule contents, which includes program name, profile, broadcast time and duration, such that users may send requests to record their desired programs on demand according to the provided information. These services are enabled by providing the EPG (Electronic Program Guide) in the system as shown in Fig. 4. The Dvbsnoop program periodically fetches the MPEG2 stream of wireless digital TV programs from the TV tuner attached to the Record Server. The embedded EIT (Event Information Table) packets within the MPEG2 stream is extracted by EIT resolver, and then stored in the EPG table. When a user requests the EPG web, the PHP Engine obtains the program schedule contents from the EPG table dynamically, updates the EPG webpage and sends it back to the user.

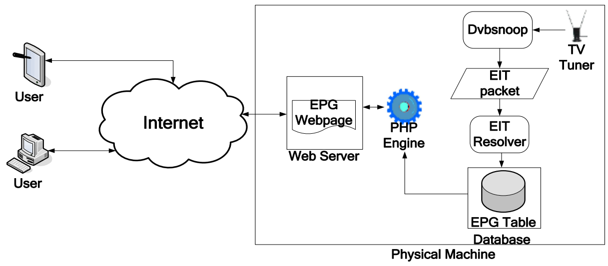

Figure 4. EPG Architecture

D. NPVR

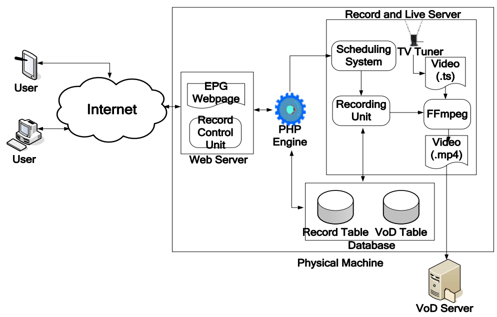

Figure 5. NPVR System Architecture

With the available program schedule contents provided by the EPG function, the NPVR system can be constructed as shown in Fig. 5. There are two Record and Live servers in our NPVR system, where each server is equipped with a few TV tuners. If a user wants to record the TV programs listed in the EPG pages, he may send a recording request message to the Web Server.

The request message contains related information about the desired program. The record control unit of the web server then decides whether to accept the request or not based upon whether the TV tuners are available during the desired recording time periods. If the request is accepted, the requested information is written into the Record table. The scheduling system then activates the recording unit to perform recording function (by checking the information in the Record table) as the program begins. During the time period when the TV tuner captures video steaming data, the FFmpeg program is also activated simultaneously to convert the TV program (.ts format) into MP4 format in real time. After the video program ends, the converted MP4 program is loaded into the VoD Server, while the program information is updated in the VoD tables. This completes the recording progress.

E. Live Video Service

The Live Video service function of this system, shown in Fig. 6, is also provided by the NPVR system. A live TV program list is located in the Web server. As a user sends request to watch live TV programs, the associated NPVR functions are activated. The live TV stream obtained from the TV tuner is sent to the FFserver via FFmpeg program, while the NPVR system is performing the recording process. The FFserver converts the video contents into FLV format, allowing the user using a web browser to download the live stream and activate the player to watch the live TV programs.

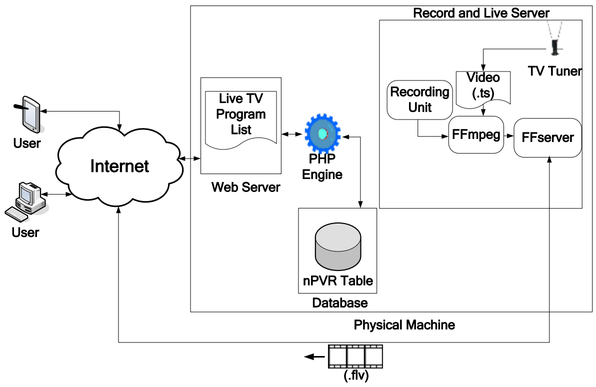

Figure 6. Live Video Service Architecture

III. System Demo and Comparison

A. System Demo

We have implemented a NPVR system including the following functions: Video Upload, VoD Service, NPVR and Live Video Service. The UI of the VoD functional system is shown in Fig. 7. The video program list webpage lists all the available recorded video programs according to the sorting index of program name, program duration, or popularity. When the user clicks on the specific program icon or its name, he may start to playback the video program, as shown in Fig. 8.

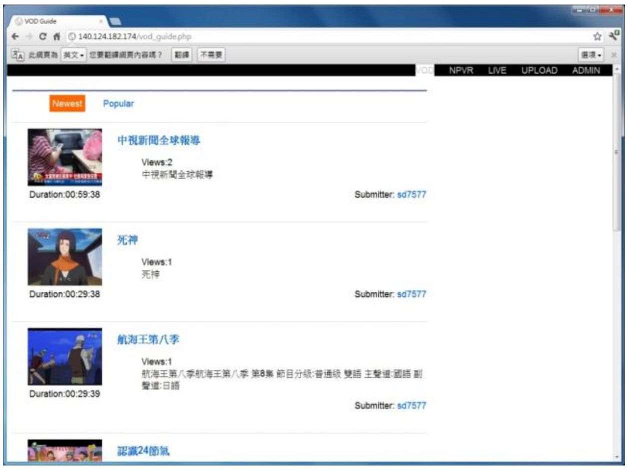

Figure 7. The UI of the VoD function in the NPVR system

Figure 8. The playback of the program in the NPVR system

B. The Comparison with OIPF Interface

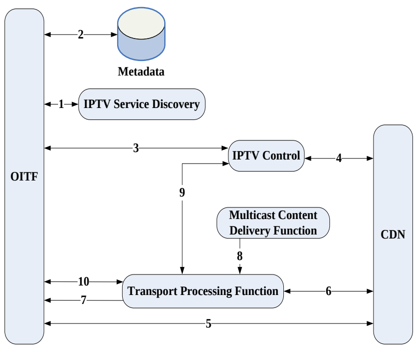

Figure 9. The interface diagram of OIPF system

Fig. 9 and TABLE I present the generic OIPF system interface diagram as well as its description. It is noted that other alternative protocol which can realize the same function may be implemented. Fig. 10 and TABLE II show the system interface diagram with description of our NPVR system. The IPTV Service Discovery and IPTV Control defined separately in the OIPF standard is integrated into the Web Server in our NPVR system. The similarities and differences of both system interfaces can be observed and compared from these two tables.

TABLE I. OIPF SYSTEM INTERFACE DESCRIPTION

Number	Description	Protocol
1	Deliver IPTV Service Menu	HTTP
2	Service or Movie Metadata Delivery	HTTP
3	Create and Maintain Service Session	SIP
4	Create and Maintain Service Session	SIP
5	Create and Maintain Content Delivery Session	RTSP
6	Deliver Movie and Broadcast Streaming Data	RTP or HTTP
7	Deliver Movie and Broadcast Streaming Data	RTP or HTTP
8	Deliver Broadcast Streaming Data	RTP or HTTP
9	Create and Maintain Service Session	SIP
10	Join and Maintain Multicast Group	IGMP

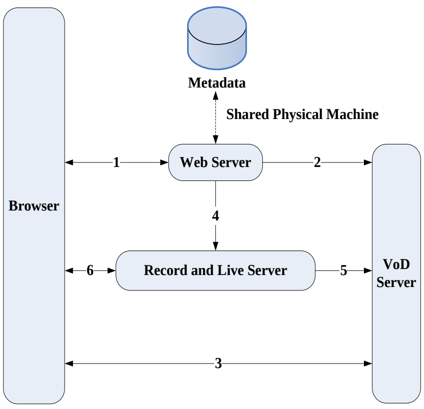

Figure 10. The interface diagram of our NPVR system

TABLE II. THE SYSTEM INTERFACE OF OUR NPVR SYSTEM

Number	Description	Protocol
1	Deliver IPTV Service Menu and Movie Metadata	HTTP
2	Deliver user uploaded Movie	SCP
3	Deliver Movie Streaming Data	HTTP
4	Deliver Scheduling message for Recording	SSH
5	Deliver Recorded Movie Data	SCP
6	Deliver Broadcast Streaming Data	HTTP

IV. CONCLUSION

We have designed and implemented the web based NPVR function for a cross-Platform IPTV system based on OIPF standard. The Apache and H.264 Streaming Module are used to build the VoD server for VoD Service. For Video upload service, we use Apache, PHP and MySQL to build the Web Server. The FFmpeg program converts the video streams into MP4 format and then uploads them to the VoD Server for storage. For the EPG function, the Dvbsnoop is chosen to capture DVB-T MPEG-2 stream, and parse the program information embedded in the EIT packet. The PHP engine updates the EPG webpage dynamically as a user queries the EPG webpage. The FFserver serves as the live streaming server to deliver live video service. The JW Player is embedded in the client's browser as the video player to playback the fetched video program. The JW player is able to switch between HTML5 or Flash mode automatically to adapt to different platforms for achieving cross-platform characteristics.

ACKNOWLEDGEMENT

This work was supported by National Science Council of Taiwan, R.O.C. under Grant NSC-101-2219-E-027-003.

REFERENCES

[1] M. Sadiku and S. Nelatury, "IPTV: An alternative to traditional cable and satellite television," *IEEE Potentials*, vol. 30, no. 4, pp. 44-46, July 2011.

[2] S. Zeadally, H. Moustafa, and F. Siddiqui, "Internet protocol television (IPTV): architecture, trends, and challenges," *IEEE Systems Journal*, vol. 5, no. 4, pp. 518-527, Dec. 2011.

[3] K. Ahmad and A. Begen, "IPTV and video networks in the 2015 timeframe: the evolution to medianet," *IEEE Comm. Magazine*, vol. 47, no. 12, pp. 68-74, Dec. 2009.

[4] T. Wiegand, L. Noblet, and F. Rovati, "Scalable video coding for IPTV services," *IEEE Trans. on Broadcasting*, vol. 55, no. 2, pp. 527-538, June 2009.

[5] J. Malisonneuve, M. Deschanel, J. Heiles, W. Li, H. Liu, R. Sharpe, and Y. Wu, "An overview of IPTV standards development," *IEEE Trans. on Broadcasting*, vol. 55, no. 2, pp. 315- 328 , June, 2009.

[6] J. Wey, J. Luken, and J. Heiles, "Standardization activities for IPTV set-top box remote management," *IEEE Internet Computing*, vol. 13, no. 3, pp. 32-39, May 2009.

[7] Open IPTV Forum. [Online]. Available: http://www.oipf.tv/.

[8] HbbTV Consortium. [Online]. Available: http://www.hbbtv.org/.

[9] MHP [Online]. Available: http://www.mhp.org/.

Tzung-Yu Wu received The B.S. degree in Computer Science & Information Engineering from I-Shou University, Taiwan in 2008, and received The M.S. degree in Computer Science & Information Engineering from National Taipei University of Technology, Taiwan in 2012. Since 2012, he works as a Software Engineer at Delta Networks, Inc., Taiwan and researches on wireless networks.

Ho-Ting Wu received his B.S. degree from National Taiwan University in 1986, and both the M.S. and Ph.D. degrees from University of California, Los Angeles in 1989 and 1994, all in electrical engineering. Since 1996, he has been on the faculty at National Taipei University of Technology, Taiwan, where he is currently a professor in the department of computer science and information engineering. His current research interests include multimedia communications, optical and wireless networks.

Kai-Wei Ke received the Ph.D. degree in Electrical Engineering from Georgia Institute of Technology, USA, in 1996. Currently, he is a professor of Computer Science and Information Engineering, National Taipei University of Technology, Taiwan, ROC. His research interests include traffic modeling, Internet protocols, and call admission, resource allocation, routing algorithms, and performance evaluation for wireless multimedia communication networks.