Implementation of a Prototype Personal Live Broadcasting System

Jaegeol Yim and Gyeyoung Lee

Dept. of Computer Engineering, Dongguk University, Gyeongju, Korea
{yim, lky}@dongguk.ac.kr

Abstract

This paper introduces our implementation of a personal real-time live broadcast system. This system is comprised of three main components: the caster program, data manipulation program, and management program. The caster program is a client program running on the end user's personal computer equipped with a webcam. This program provides a convenient environment (cue sheet, prompts, reference image, buttons to control broadcasting) in which the user can create his video contents easily. The data manipulation program receives user's webcam video stream, combines the stream with the reference image and texts, then outputs HDMI, SID or RTSP as the manager directs. The management program manages broadcasting schedules, cue sheets, and users. User registration, user information modification, and deletion of a user are handled by the management program. A producer creates a broadcasting schedule (cue sheets) with the management program. One of the distinguished features of our personal real-time live broadcast system is its less than 1 second delay time.

Keywords: personal live broadcast, cue sheet, prompts, caster

1. Introduction

As the internet came to be recognized to the public through Web later in 1980, it has become to develop into a communication network connecting the world. Therefore, the internet on the basis of such a communication network is becoming a necessary factor of modern society in 21st century, and the industry and technology based on the internet are developing day by day [1].

Television (TV) is a telecommunication medium for transmitting and receiving moving images with or without accompanying sound. Commercially available since the late 1920s, the television set has become commonplace in homes, businesses and institutions [1] and TV is the most popular resource people rely upon to get relaxation and at the same time useful information such as news [3].

In order to enjoy the advantages of both Internet and TV, many efforts of combining these two have been made by researchers. Internet Protocol television (IPTV) is a system through which television services are delivered using the Internet protocol suite over a packet-switched network such as the Internet, instead of being delivered through traditional terrestrial, satellite signal, and cable television formats [4]. IPTV services may be classified into three main groups: live television, time-shifted television, video on demand (VOD). Among the service types, VOD service is the most popular [5].

Internet television (Internet TV, or Online TV) is the digital distribution of television content via the Internet [6]. Web television (web TV) is original television content produced for broadcast via the World Wide Web [7].
The popularity of internet TV and web TV gave birth to the demand for personal broadcasting. According to Wikipedia, personal broadcasting is a term for participatory journalism that focuses on television webcasting over the mobile Internet [8]. AfreecaTV [9] and USTREAM [10] are examples of popular personal broadcast systems. These existing systems lack real-time management functions. In other words, any subscribed user can take any video including suggestive or violent video and broadcast it on the Internet without any restriction. Our personal broadcast system provides management functions such as monitoring channels and stopping programs to managers. One of the most distinguished features of our system is its short delay time that is less than one second. Another distinguished feature of our system is that it outputs SDI, HDMI or RTSP as the manager directs.

2. Related Works

Live personal broadcasting can be used as a resource of IPTV and internet TV and these are closely related to this paper. There are many different types of IPTV set-top boxes. Consequently, setting up a digital channel usually requires reprogramming. However, a channel service producer usually lacks programming skill. The author of [11] proposed a service generation apparatus with which a channel service producer can easily set up an IPTV interactive digital channel.

Service Level Agreements (SLAs) specify the quality levels customers expect during service provisioning. The authors of [12] proposed a protocol for secure Mobile IPTV service delegation to support SLAs of an IPTV provider through access control extension to different security domains using Single Sign-On technique. Furthermore, they demonstrate the efficiency of collaborative IPTV service when SLA is strictly considered in case of limited resources of a single IPTV provider.

On-demand programs provide freedom of choice to the users. Streamed on-demand TV shows, movies, downloadable contents, pay-per-view contents are examples of such choices. In such an environment, lack of channel capacity becomes a critical problem. The authors of [13] introduced virtual channel systems that support bi-directional interconnection with different types of media. The virtual channel works both on Open Cable Application Platform and on Advanced Common Application Platform for Internet TV.

The authors of [14] introduced a novel method for extracting "topics" as interesting events in a video. When the target character's appearance pattern is abnormal we say the situation is interesting. Shot durations in thrilling events are very short whereas those in romantic events are very long, and they are all considered to be abnormal patterns. The proposed method divides the video into events characterized by specific patterns of the target character's appearance and disappearance.

Live streaming media distribution in a large scale P2P network and the impact on the real-time caused by the dynamicity of peers are still not well resolved. In order to address this problem, the authors of [15] presented a scalable live streaming media distribution service in P2P network. The service builds the P2P network combining tree with clusters and reserves some super-peers in all clusters. They adopted the maximum accepted connection number of each super-peer and its average disposing delay to select the optimal father peer. They also designated the limitation of packet discard between super-peers.

Nowadays, the Internet is a main means of transferring information over the world in our lives. As the communication networks to home is upgraded, high quality multimedia services become available. Networking operation cost is high since it requires several data copies from one place to another as data passes from one protocol layer to another. In order to provide efficient multimedia streaming service over the network, we need to minimize the network
operation cost. The authors of [16] proposed an efficient system including hardware and software in terms of the design concept and system architecture for the multimedia streaming service.

P2P based live streaming applications are growing up rapidly. However, when extending them to WLANs, the bandwidth bottleneck and the high rate of packet loss are usually the major obstacles. In order to overcome this problem, the authors of [17] proposed a new wireless multicast agent mechanism based on the IP multicast scheme for buffer management and scheduling.

3. Our Personal Broadcast System

Our personal broadcast system is comprised of the caster program, data manipulation program and broadcasting management program. The caster program is a client program running on the user's personal computer equipped with a webcam. The main purpose of the caster program is to stream out the video content generated by the webcam. In addition to that, it provides a convenient broadcasting environment to the user as shown in Figure 1. It shows the typical user interface of the caster program. It consists of six windows: ① a window in which webcam video is displayed, ② cue sheet is listed, ③ a reference image is displayed, ④ status of the broadcast, ⑤ control buttons (server status, receive cue sheet, rehearsal, standby, start broadcasting, next, stop broadcasting, set-up, automatic), ⑥ prompts (scripts for the caster are displayed and scrolled).

![Figure 1. A Typical User Interface of the Caster Program](image)

The broadcasting management program approves or disapproves user's request for broadcasting and streaming out user's broadcast. This program provides management functions for authentication, basic information, broadcasting programs, and cue sheets. A typical user (manager or operator) interface of broadcasting management program is shown in Figure 2.

Our data manipulation program combines the reference image with the video content from the webcam. Then, using FFMPEG [18], it transcodes the resulting video into SDI, HDMI or RTSP for IP transmission.

The typical operation scenario of our personal broadcast system is shown in Figure 3. Once the PD confirms the broadcasting schedule for a caster using broadcasting management program, the caster writes a cue sheet for his/her broadcasting and uploads it. After receiving the cue sheet, the PD notifies the caster of the broadcasting time twice: at 1 hour before the time and 10 minutes before the time. At the broadcasting time, the PD sends out On Air
signal to the caster. The caster, then, starts his/her broadcasting. The data manipulation program transcodes the broadcasting signals into SDI, HDMI or RTSP as it is indicated and sends it out to the broadcasting management system.

Figure 2. A Typical User Interface of the Broadcasting Management Program

Figure 3. A Typical Operation Scenario of our Personal Broadcast System

4. Design

The caster program consists of the following components:

- CasterClient: This component displays the main form as shown in Figure 1 for the caster.

- CastCommon: Basic functions such as date-to-time conversion, difference between two dates and basic data types for broadcasting program information, caster information, media status information are defined

- MdiaSDK: Functions for media manipulation such as setting media devices and data communication with the computer graphic server are defined.
- **CasterClientControl**: Functions for the control buttons - server status, receive cue sheet, rehearsal, standby, start broadcasting, next, stop broadcasting, set-up, automatic - are defined.

- **CueSheetFileDownloader**: Functions for downloading cue sheets are defined.

- **WpfTeleprompterControl**: Initializes the variables associated with the prompt window (Figure 1).

- **CasterWebService**: Using web services, it obtains cue sheet data and parses the data in order to identify broadcasting date, duration of the broadcast, the name of the program, the start time, the termination time, and so on.

The caster program needs to know information of casters, broadcasting programs, schedules, and cue sheets. In order to record this information, we have a database described in Figure 4. In Tb_BCastProgram table, programs are recorded. In Tb_User table, user information is recorded. For each broadcast program, the caster and the start time of it is recorded in TB_BCastSchedule table. Information of cue sheets is recorded in Tb_BCastCueSheet table. Information of broadcast items is recorded in Tb_BCastItem.

**Figure 4. Database Structure for the Caster Program**

Management of subscribers, user's authorities, cue sheets and other resources requires a database. The database structure of our management system is shown in Figure 5. In TbUtl_Group table, the authority group names are listed. In TbUtl_Role table, names of role are listed. For each authority group, allowed roles to the group are listed in TbUtl_GroupRole_Xref table. In TbUtl_Resource table, all the resources are listed. For each pair of (role, resource), operations (create, read, update, and/or delete) allowed to the role on the resource are specified in TbUtl_RoleResource_Xref table.
The structure of our data manipulation program is shown in Figure 6. External Capture Interface receives video and audio data from external hardware devices, File Input Interface receives data from files, whereas RTSP/RTMP Input/Output Interface receives data from the network card. Input data is divided into audio and video by the Re-Sampler so that they are treated separately before recombined. Audio Mixer mixes the audio data with other audio resources. The video data is also edited before recombined with the output of Audio Mixer. The outputs of audio and video can be recombined through the Encoder to go out to the network or through Output Streamer in order to be transcoded into SDI or HDMI.

In order to produce HD video, we have to resample at 1080p. This takes 18-34 ms. Using 4 resampling threads, we speeded up the process of producing HD video. We use DeckLink card for HD broadcasting. This card takes 1.2 seconds when it broadcasts both video and audio together. In order to reduce this delay, we made it broadcasts video only for the first 5 seconds before it starts broadcasting both video and audio. With this strategy, we reduced the delay time to 350 ms.

The process time of Image/Subtitle Overlay is 32 ms and that of PIP is 27 ms. Therefore, we implemented 8 layers in Time-Shift manner. As the result, the process times of both
Image/Subtitle Overlay and PIP reduced to less than 16 ms. Finally, we can calculate the delay time of our data manipulation system as follows:

\[ 350 \text{ ms} + 8 \times 16 = 492 \text{ ms.} \]

5. Implementation

For our database implementation, we used Microsoft SQL Server 2008 R2. In our application server, we implemented our business layer consisting of Business API, Exception Manager and Interception Mechanism, ASMX pages (Facade Web Service), and ASPX pages (Web UI). Our caster program interacts with the Application Server through ASMX pages or Interception Mechanism whereas our management system interacts with our application server through ASPX as shown in Figure 7.

![Figure 7. The Structure of our Development System](image)

We have implemented the following web methods of manipulating cue sheets:

- BCastCueSheetData GetBCastCueSheet(int iBCastCueSheetNo): returns the cue sheet whose id is iBCastCueSheetNo.
- GetBCastItem(int iBCastCueSheetNo): returns the broadcast cue sheet item.
- ConfirmBCastCueSheet(int iBCastCueSheetNo, string sConfirmUserId): confirms the cue sheet designated by the iBCastCueSheetNo for the caster designated by sConfirmUserId.
- RemoveBCastCueSheet(int iBCastCueSheetNo): removes the cue sheet identified by iBCastCueSheetNo.
- SaveBCastCueSheet(BCastCueSheetData dsBCastCueSheetData, BCastItemData dsBCastItemData)
- public SComboData GetSCombo(string sComboListID, string sClassID)
- public int ChangeBCastCueSheetStatus(int iBCastCueSheetNo, string sBCastStatusKindCode)
- public BCastScheduleData GetMyBCastScheduleList(DateTime datSearchStartDate, DateTime datSearchEndDate, string sUserId)
We have implemented BCScheduleAdd.aspx for adding a new schedule, BCScheduleList.aspx for listing schedules, BCScheduleListAjax.aspx, Caster_Pop.aspx, and so on. In BCScheduleAdd.aspx, for example, the following functions are defined: chAddValidate(), chkModifyValidate(), ClosePopTab(), SetProgram(), SetCaster(), chkDel(). Similarly, functions validate(), ScheduleAdd(), ScheduleAddModify(), ConfirmCueSheet(), ReBinding(), and GetSelectedValue() are defined in BCScheduleList.aspx. In PasswordChange_Pop.aspx, BindUserData() and ChangePassword() are defined.

We employed 4 layer (event, web service, business, data access) software architecture in our implementation. As an example, the layered architecture of our implementation of the cue sheet management system is shown in Figure 8.

Table: Layer Diagram

<table>
<thead>
<tr>
<th>Event</th>
<th>Data Service</th>
<th>Business</th>
<th>Database</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search cue sheet</td>
<td>chAddValidate()</td>
<td>chkModifyValidate()</td>
<td>ClosePopTab()</td>
</tr>
<tr>
<td>Add cue sheet</td>
<td>SetProgram()</td>
<td>SetCaster()</td>
<td>chkDel()</td>
</tr>
<tr>
<td>Edit cue sheet</td>
<td>Validate()</td>
<td>ScheduleAdd()</td>
<td>ScheduleAddModify()</td>
</tr>
<tr>
<td>Confirm Cue Sheet</td>
<td>ConfirmCueSheet()</td>
<td>ReBinding()</td>
<td>GetSelectedValue()</td>
</tr>
</tbody>
</table>

Figure 8. Program Structure Diagram

6. Experiments

We have tested our system in various circumstances. Figure 9 shows a screenshot of retrieving broadcasting schedules that are permitted to the caster for the specified period of time. The caster can click the "select" button to create, edit, or just check the cue sheet associated with the schedule. If the caster clicks the "Execution" button, then the window for broadcasting pops out.

Figure 9. A Screenshot of Retrieving Schedules

Figure 10 shows a screenshot of our caster system in the broadcasting mode. The window at the top-left shows the live webcam video, the down-left window shows the reference image and the down-right window shows the prompts that the caster reads. Figure 11 shows a screenshot of the IPTV with which an end user watches our personal broadcast. We can see the live webcam video is embedded into the reference image.

Figure 10. A Screenshot of Broadcasting Mode

Figure 11. A Screenshot of IPTV

302
7. Conclusions

We have introduced our personal real-time live broadcast system. It is natural that people want to show their own contents to others. The success of Google's YouTube proves this axiom. They create their own videos first and then upload the videos later. Using our system, people can broadcast their own contents while they are creating them. Delay time of the system is less than 1 second and it can broadcast HD video. For the further study, we are enhancing the data manipulation system with various editing functions.

Acknowledgements

This work (Grants No. C0033172 and 2012-0002611) was supported by Business for Cooperative R&D between Industry, Academy, and Research Institute funded Korea Small and Medium Business Administration in 2012 and by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology. Lee’s research was supported by the ‘Development of Global Culture and Tourism IPTV Broadcasting Station’ Project through the Industrial Infrastructure Program for Fundamental Technologies funded by the Ministry of Knowledge Economy (10037393).

References


Authors

Jaegeol Yim received the M.S. and Ph.D. degrees in Computer Science from the University of Illinois at Chicago, in 1987 and 1990, respectively. He is a Professor in the Department of Computer Science at Dongguk University at Gyeongju Korea. His current research interests include Petri net theory and its applications, Location Based Service, AI systems, and multimedia systems. He has published more than 50 journal papers, 100 conference papers (mostly written in Korean Language), and several undergraduate textbooks.

Gyeyoung Lee received the M.S. degree in Computer Science from the Dongguk University in 1982 and the Ph.D. degree in Computer Engineering from the Dankook University in Korea in 1992, respectively. He is a Professor in the Department of Computer Science at Dongguk University at Gyeongju, Korea. His current research interests include Petri net theory, AI systems and Speech processing systems.