An Experiment for an Interactive Internet Live Broadcasting System with a High-Quality Snapshot Function

Yoshia Saito[*] and Yuko Murayama[*]

[*] Faculty of Software and Information Science, Iwate Prefectural University, Japan
{y-saito, murayama}@iwate-pu.ac.jp

Abstract – The video quality on the Internet does not come up to TV quality yet. Although audience often ask for high-quality videos, it is difficult to improve the quality because of its huge video traffic. In this paper, we propose an Internet broadcast system with a high-quality snapshot function to improve user experience. While the proposed system delivers low-quality video to audience, it provides a high-quality snapshot function which enables the audience to take a snapshot of a desired and favorite scene anytime. To assess effectiveness of the snapshot function, we designed and implemented a prototype system. This paper reports results of an experiment in a commencement ceremony of our university using the prototype system.

Keywords: Internet Broadcasting, Interactive TV

1 INTRODUCTION

With the spread of broadband Internet connections, Internet users can watch video contents all over the web in recent years. Video players are integrated into most web browsers and the Internet users can watch videos easily on their web browsers without any installation of proprietary client software. These video platforms such as Adobe Flash and Microsoft Silverlight rapidly popularize video viewing on the web. Tens of millions of videos are shared in video sharing services like YouTube [1] and enormous numbers of on-demand videos are watched by Internet users in various countries. Furthermore, live video broadcasting by Internet users also gets popular nowadays. A lot of Internet users broadcast their original live video contents using PCs and web cameras. It is expected that these video sharing and broadcasting services would become widely used much further and network traffic of the videos would grow more in the next couple of years.

The huge video traffic, however, causes a problem of communications expenses. Although most video sharing and broadcasting services run on income from advertisements on their websites, it is difficult to make profits because of its expenses more than its advertising income [2]. Moreover, current online video services distribute videos with a few hundred kilobits per second (kbps). The video quality on the Internet does not come up to TV quality yet. While the video services should provide more high-quality videos, it is not easy to improve the video quality for the above reason.
There are two solutions to increase user experience; one is to improve video quality upgrading video resolution and bit rate, and the other way is to enhance added value of the video contents introducing interactive features and so on. Under present circumstances, the later is a realistic solution. Interactive television (iTV) [3-6] is a research area which provides interactive features to video contents in order to improve user experience. Ustream [7] and Justin.tv [8] which are typical services for live video broadcasting apply the iTV technologies to their system. In these services, live video viewers can communicate with broadcasters and other viewers using chat and social communication tools (e.g. Twitter) watching live video contents. These interactive functionalities are attracting the attention of many Internet users despite low resolution and bit rate of the live videos.
Meanwhile, we have been trying and conducting several experiments with Internet broadcasting in graduation ceremony of our university [9, 10]. In these experiments, we had to deliver live video over the Internet with a few hundred kbps because we only have 100 Mbps connections between our university and the Internet. Therefore, it was difficult to satisfy parents who could not attend the ceremony and would like to watch their children's proud moment because of its low-quality video. We needed to add something extra to our live broadcasting system introducing special functionalities to improve user experience without increasing network traffic.

In this paper, we propose an interactive internet live broadcasting system called *Photographable TV* which provides a high-quality snapshot function so that audience can take high-quality pictures of favorite scenes for their memories at any time watching live video. In case of graduation ceremony, parents of graduates can take ceremonial pictures remotely as if they were attending the ceremony. The pictures can be saved to local disks for their personal memory albums. Since the data size of still pictures is far small than that of video, the proposed system can improve user experience without increasing network traffic. To study the effectiveness of the high-quality snapshot function, we design and implement a prototype system. We also conduct an experiment in our graduation ceremony to evaluate how to use our system by audience and find issues.

The paper is organized as follows. In the next section, we describe related work discussing originality of our proposed system. Section 3 introduces the model of the Photographable TV system. Section 4 presents the design of the prototype system of the Photographable TV, its system architecture and user interface. Section 5 evaluates the prototype system and reports the experiment results. Section 6 gives some conclusions and our future work.

2 RELATED WORK

To increase video quality for improvement of the user experiment, there are many researches. Typical one of them is IP multicast [11, 12] In the IP multicast, a sender transmits a single data stream to the receivers. Since the routers on the

ISSN1883-4566

path to the receivers replicate the data stream so that multiple receivers can receive it if required, it can deliver high-quality videos without increasing network traffic on the sender. However, it is not easy to use over the Internet because all routers on the path to receivers must support IP multicast. On the other hand, unicast is widely used in the Internet although it delivers multiple same copies to each receiver. This is because the unicast does not need special functionalities of the routers and can be used in any different network environments. In this regard, our proposed system uses unicast for the video delivery over the Internet.

Nowadays the P2P technologies are popularly used in the Internet to distribute network traffic over the Internet [13-16]. In these researches, hosts built an overlay network on the Internet by the P2P technologies and forward the received data stream to the other hosts like overlay multicast so that it can avoid concentration of the network traffic on a sender. However, the P2P technologies often require proprietary client software. The installation of the software prevents Internet users from casually watching videos with these P2P technologies. Moreover, the P2P software is often prohibited to use in a particular environment such as office network and university network because it goes through firewalls ignoring its network architecture. For these reason, it is difficult to widely use P2P software and enhance video quality over the Internet without increasing network traffic on a sender.

For similar ideas to our high-quality snapshot function, there are several studies in educational system. Ichimura proposes Chalk Talks [17] which is a remote lecture system with high-quality pictures. The Chalk Talks uses a HDV camera to provide a lecture with high-resolution. Since the high-resolution video consumes network resources, it compresses the video for the Internet broadcasting. The Chalk Talks also provides high-quality pictures at fixed intervals to clients so that the students can watch the white board clearly. While the Chalk Talks provides high-quality pictures at fixed intervals, the Photographable TV provides high-quality pictures when audience requests. In addition, the Photographable TV aims to make a personal memory album so that audience can remember the live broadcasting for improvement of the user experience although the Chalk Talks aims to improve readability of the white board.

One of video sharing services, PANDORA.TV [18] provides a snapshot function to the viewers. In the service, there is an image capturing button on the video player and still pictures of favorite scenes can be captured in JPEG format watching videos. However, it does not provide high-quality pictures because resolution and quality of the pictures are same as that of videos. Our system offers high-quality pictures to the audience more than video quality.

3 PHOTOGRAPHABLE TV

Photographable TV is an interactive broadcasting system for Internet live video streaming for the purpose of improvement of user experience without increasing network traffic by a high-quality snapshot function to enjoy high-quality pictures of favorite scenes for a personal memory album. The high-quality pictures do not increase network traffic so much if audience sends requests in moderate intervals. Even if the video quality is not so high, the high-quality picture would improve user experience and activate social communication on the web.

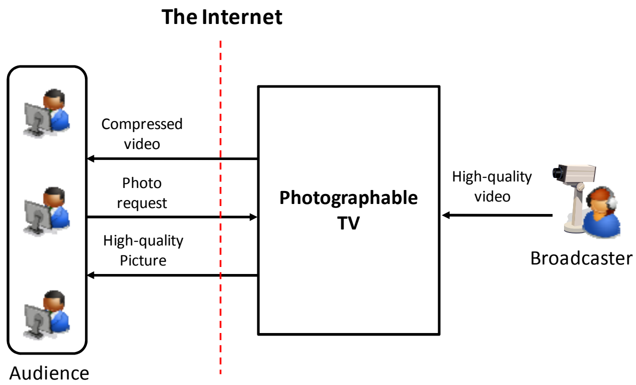


Figure 1: System model

Figure 1 shows the system model of the Photographable TV. This model consists of a broadcaster, its audience and the proposed system. Firstly, the broadcaster sends a high-quality video source to the system. The system receives and encodes the high-quality video storing the original source. The audience receives the compressed video from the system in real time and also can send a picture request to the system anytime watching the video. When the system receives the picture request, a high-quality picture is made from the stored original video source and sent to the audience. The audience can see and save the high-quality picture.

There are several issues to realize the Photographable TV. The Photographable TV requires encode functions for video and pictures. Since quality of pictures is equivalent of the original video quality, the original video should be uncompressed and high-resolution so that high-quality pictures can be made from it. However, it is difficult to send the original video over the Internet because the data size of the uncompressed and high-resolution video is too large. This is an issue. The encode functions must be near the broadcaster side not to across the Internet. Besides video encoding, the broadcaster's PC has to extract a frame from the video and encode the frame to make a still picture. It is expected to consume CPU resource of the PC and we should take care of its load. Another issue is frequency of the high-quality picture requests from audience. The proposed system is available in accordance with an idea that picture traffic is much less than video traffic. If the audience frequently requests high-quality pictures, the picture traffic would be considerable amount. We have to study how many times the audience requests the high-quality pictures and control the picture traffic not to exceed network capacity.

4 PROTOTYPE SYSTEM

We developed a prototype of the Photographable TV to conduct experiments for its evaluation. In this section, we describe the design and implementation of the prototype system.

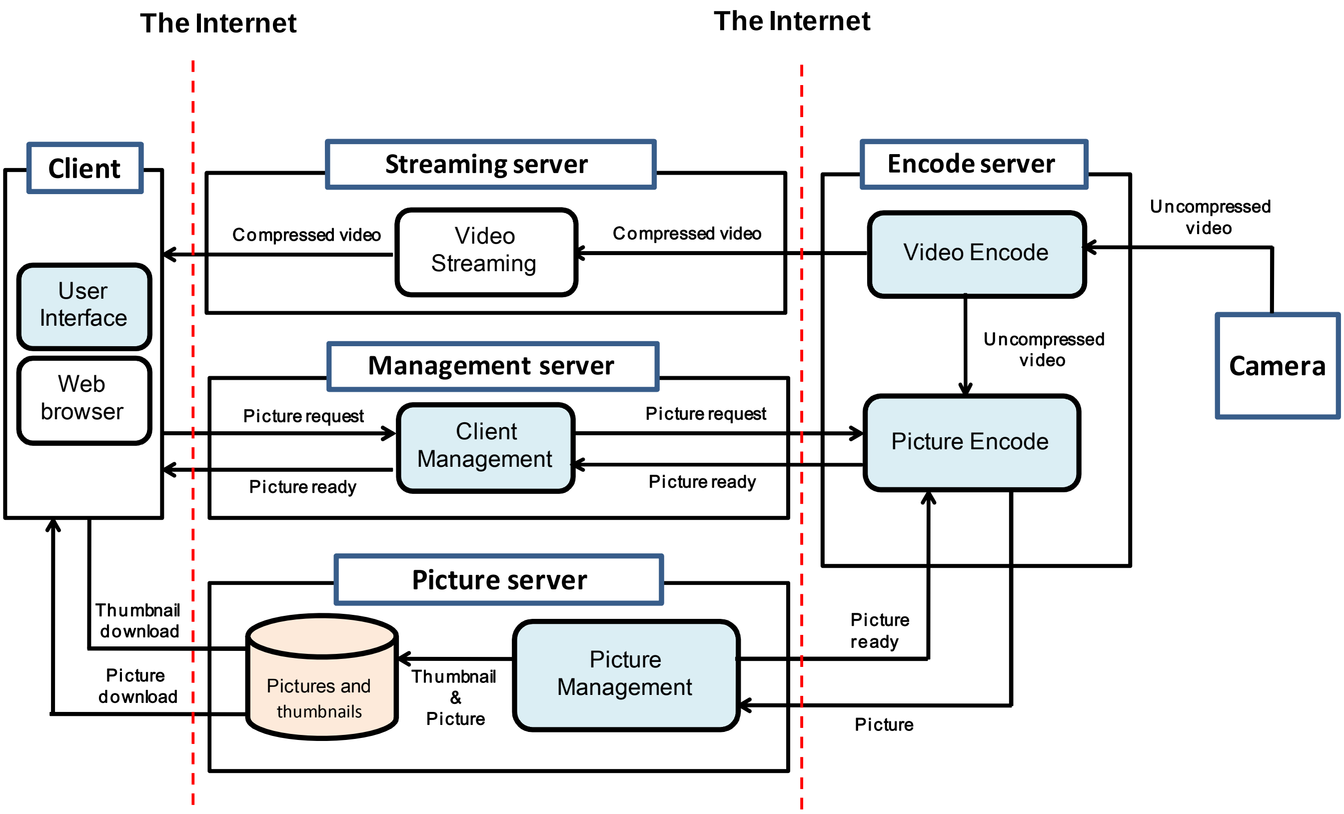


Figure 2: System Architecture

Table 1: Server specifications

Picture server	
OS	Windows Server 2003
CPU	Intel Xeon 2.00 GHz
Memory	1024 MB
Web server	Apache 2.2.14
PHP	PHP 5.3.1

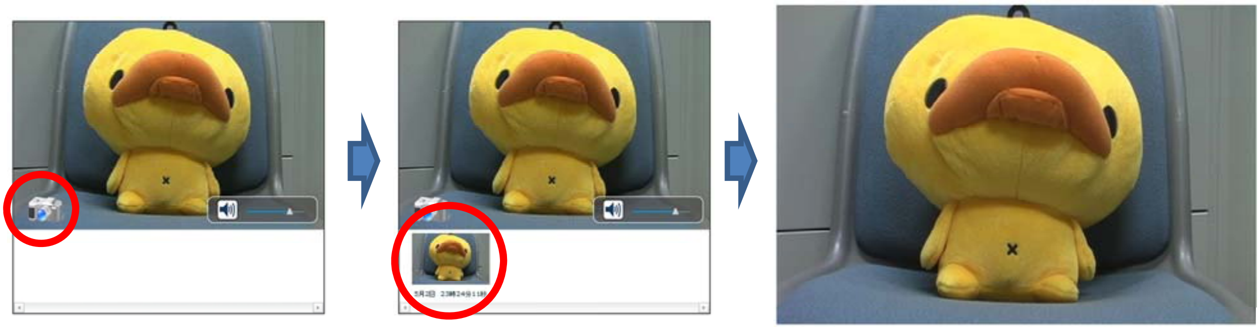
Video streaming server and management server	
OS	Windows Server 2003
CPU	Intel Xeon 2.00 GHz
Memory	1024 MB
Streaming server	Adobe Flash Media Server
JAVA	JDK 5.0

Encode server	
OS	Windows Vista
CPU	Intel Core2 Duo P8700 12.53GHz
Memory	4096 MB
Flash	Flex SDK 3.0
Video camera	Panasonic NV-GS320-S

4.1 System Design

Figure 2 shows the system architecture of the prototype system. The system consists of four servers; an encode server, a streaming server, a management server and a picture server. The encode server has two functions. One is a video encode function and the other is a picture encode function. The video encode function receives an uncompressed video source from a camera and compresses the video for broadcasting. The uncompressed video source is also passed to the picture encode function. The picture encode function stores the uncompressed video so that high-quality picture could be made from the source.

The compressed video is sent to a video streaming function on the streaming server. The streaming server sends the video to each client by unicast when requested. The audience can send a picture request to a client management function on a management server watching the video through a user interface on the browser of the client when they would like to take pictures of specified scenes. The client management function keeps client IDs and forwards the picture requests with their client IDs to the encode server. When the picture encode function receives a picture request from the management server, it encodes high-quality picture from the uncompressed video. The encoded picture is sent to a picture management function on a picture server. A thumbnail is made from the picture and they are stored in a database on the server. After that, a picture ready message is sent to the picture encode function and it is forwarded to the client management function with the client ID and location information of the thumbnail and the picture. The client management function forwards location of the pictures based on the client ID. The client only downloads the thumbnail from the picture server to save network resource in case the audience does not like the shot. After the audience confirmed the thumbnail, the client downloads the picture from the picture server and displays it on the user interface. The audience can save the high-quality picture to the local disk on the client to enjoy the pictures after the broadcasting.



1. Request a photo 2. Select a photo 3. The High-quality photo is shown

Figure 3: User interface for clients

4.2 Implementation

We implemented the prototype system based on the system design. The video/picture encode function on the encode server and the user interface on the client are implemented based on Adobe Flash written in ActionScript 3.0 for ease of video delivery and viewing over the Internet. The client management function is implemented by JAVA because it works on various environments. The picture management function is implemented by PHP script on a web server so that it is easy to upload pictures on the web server and make thumbnails of the pictures. We used Adobe Flash Media Server for the video streaming on the streaming server. The specifications of these servers are shown in Table 1. We used same PC for the video streaming server and the management server.

Video Encode

The video encode function receives raw video data from the camera with resolution 720x480. The raw video is resized and compressed by Flash framework to broadcast it in real time. Then, the compressed video data is sent to the streaming server in few hundred kbps. A still picture is captured in BMP format and passed to the picture encode function.

Picture Encode

Since the BMP file is not compressed, the file size becomes about 1 MB for a 720x480 still picture. It is too large to upload the file to the picture management server over the Internet. Therefore, we compressed the BMP data in JPEG format. After the compression, the files size will be from 30 KB to 50 KB and the network traffic between the encode server and the picture management server can be reduced.

When the BMP data is compressed in JPEG format, it increases CPU load of the encoder server. If the clients frequently send picture requests to the encoder server, the picture generation would be aborted. To make matters worse, the frequent requests would cause huge network traffic between the encode server and the picture server even if the data size of the JPEG files is small. Therefore, we introduced periodic picture buffering scheme into the picture encode function. The picture encode function stores BMP data on the memory at fix intervals. In this implementation, we set the interval to 500 msec taking into account the server load. When a picture request is arrived, the encode function searches latest picture on the memory. To reduce the delay from the moment when the shutter is clicked at the client to the moment when a snapshot actually is taken at the encode server, video buffering time on the client is sent and the picture encode function minus the video buffering time from the arrival time of the picture request to search the latest picture. We assume the transmission delay between the client and the picture encode function is small and the delay time is ignored. The BMP data is encoded in case it was not previously encoded. If the BMP data has been already encoded, it does not process the picture encode and returns only the picture URL to the client. The buffering scheme can reduce the server load and network traffic between the encode server and the picture server.

Picture Management

The compressed JPEG picture is sent to the picture server. The picture management function receives the picture. At the same time, a thumbnail of the picture with resolution 120x90 is made from the picture. The data size of the thumbnail is a few Kbytes. The high-quality picture and its thumbnail are saved in a public directory on the local web server. The client receives URL addresses of the picture and thumbnail. Although there is no user authentication to see the pictures in the prototype system, access control technologies should be introduced so that the pictures can be accessed by audience who owns them.

Client Management

We implemented four functions for the client management; 1) client ID management, 2) connection management, 3) picture request forwarding and 4) logging. The function of client ID management generates and keeps client IDs for each client which accesses to the management server. The client ID is a unique 22 characters and sent to the client when connected to the management server for first time. The client keeps the unique ID as a cookie on the web

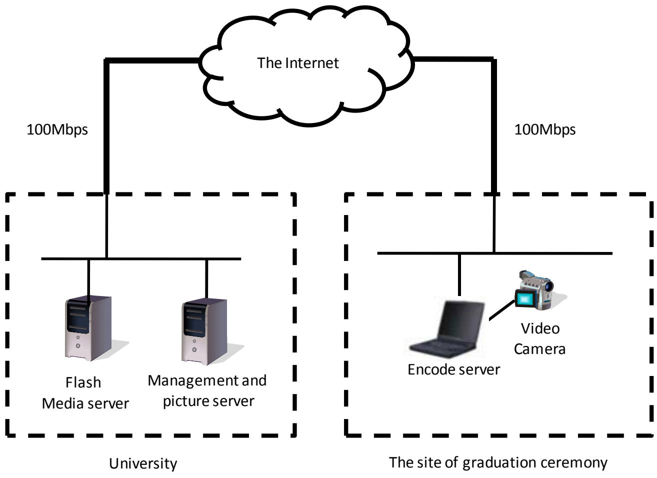


Figure 4: Network configuration

browser and can use the same ID thereafter. The ID is used for the picture request forwarding and system logging to identify the clients. The connection management function associates the client ID with its connection. The function of picture request forwarding notifies the encode server of a picture request with a client ID and replies the result to the client based on the ID. The logs maintain connected and disconnected time and the picture requests with their IDs.

User Interface for Clients

Figure 3 shows the user interface for clients. The compressed video is shown on the upper portion of the interface. The camera icon is a snapshot button to send a picture request to the management server. When the camera icon is clicked, it is not available before its response arrival. After completion of a picture request, a small snapshot picture is added to the thumbnail list by downloading from the notified URL address. A high-quality picture is displayed with resolution 720x480 on the other browser window when the thumbnail is clicked. The high-quality picture can be saved to the local disk of the client by an image saving function of the web browser.

5 EXPERIMENT

We conducted an experiment in our graduation ceremony with the prototype system in order to evaluate how to use our system by audience and find issues.

5.1 Methodology

Figure 4 shows the network configuration in the experiment. A flash media server and a management/picture server were placed in the University. These servers connected to the Internet at 100 Mbps. The graduation ceremony was held in the other place and we prepared 100 Mbps connections for the venue. An encode server and a digital video camera were employed. The bit-rate of the video streaming was 200 kbps. Any Internet users could watch the video streaming in real-time on our website for the broadcasting. We recorded number of the viewers, CPU load of the encode server and a log of photo requests on the picture/encode server.

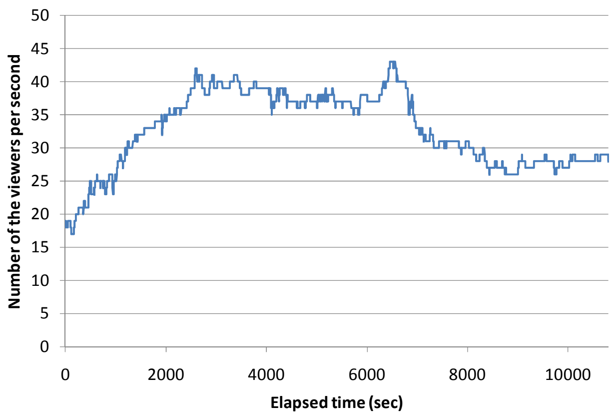


Figure 5: Number of the viewers per second

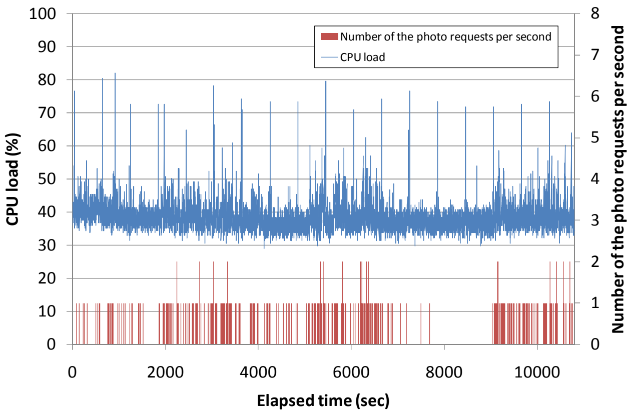


Figure 6: CPU load of the encode server and number of the photo requests per second

5.2 Results

We analyzed the results to know how many people used and its scalability. Note that we excluded broadcasters and researchers related to the experiment from these results.

At first, we counted the number of the viewers through the broadcastings to study how many/long people used our system. Figure 5 shows the number of the viewers per second from 13:00 to 16:00. The total number of unique viewers is 148. The maximum and average numbers of the viewers are 43 and 33 respectively. As a whole, the prototype system kept the number of viewers throughout the broadcast. We presume the photographable TV could improve user experience of the audience and attract them not to stop viewing.

We also analyzed CPU load of the encode server and number of the photo requests per second to study its scalability. Figure 6 shows the results from 13:00 to 16:00. From the graph, the CPU load was around 40%. Although the load momentarily marked around 80%, the encode PC remained power throughout the experiment. While the number of the photo requests per second constantly occurred, the prototype system could provide the snapshot function. Since the total number of the photo requests was 423, the viewers used the snapshot function frequently.

From the experiment, we found the prototype system could be used by several tens of viewers at least and load of the encode server was suppressed by the periodic picture buffering scheme. The prototype system could provide the snapshot function for small-scale live broadcasting.

6 CONCLUSION

In this paper, we proposed an Internet broadcast system with a high-quality snapshot function toward improvement of user experience. The proposed system delivers low-quality video to audiences. Meanwhile, it provides a high-quality snapshot function which enables the audiences to take a picture of a desired and favorite scene anytime. We designed and implemented a prototype system and evaluated the system in our graduation ceremony. From the result, prototype system worked stably throughout the experiment even if more than 40 users watched the broadcasting simultaneously and the snapshot function was used 423 times. We confirmed the prototype system could provide the snapshot function for small-scale live broadcasting.

As future work, we will study server load when audience increases and its scalability. We will also conduct experiments at various events to evaluate the Photographable TV further in practical situations.

REFERENCES

[1] YouTube, http://www.youtube.com/.

[2] C. Huang, J. Li, and W. K. Ross, “Can Internet Video-on-Demand be Profitable,” SIGCOMM’07, pp. 133-144, 2007.

[3] J. Jensen, “Interactive Television - A Brief Media History”, In Proc. of EuroITV2008, pp. 1-10, 2008.

[4] P. Cesar and K. Chorianopoulos, “Interactivity and user participation in the television lifecycle: creating, sharing, and controlling content,” In Proc. of UXTV2008, pp. 125-128, 2008.

[5] M. Ursu, et al., “Interactive TV narratives: Opportunities, progress, and challenges,” ACM TOMCCAP, Vol. 4, Issue 4, 2008.

[6] C. Metcalf, et al., “Examining presence and lightweight messaging in a social television experience,” ACM TOMCCAP, Vol. 4, Issue 4, 2008.

[7] Ustream, http://www.ustream.tv/.

[8] Justin.tv, http://www.justin.tv/.

[9] N. Takamizawa and Y. Murayama, “Use of Telepresence in Informal Broadcasting over the Internet, In Proc. of the ACM SIGMM 2004 Workshop on Effective Telepresence,” pp. 12-15, 2004.

[10] Y. Saito and Y. Murayama, “A Proposal of an Interactive Broadcasting System for Audience-driven Live TV on the Internet,” Journal of Information Processing, Vol. 18, pp. 26-37, 2010.

[11] S. Deering, “Host Extensions for IP Multicasting,” RFC 112, 1989.

[12] H. Eriksson, “MBONE: The Multicast Backbone,” Communications of the ACM, Vol. 37, No. 8, pp. 54-60, 1994.

[13] D. Tran, K. Hua, and T. Do, “Zigzag: an efficient peer-to-peer scheme for media streaming,” IEEE INFOCOM 2003, pp. 1283-1292, 2003.

[14] M. Guo and M. H. Ammar, “Scalable live video streaming to cooperative clients using time shifting and video patching,” IEEE INFOCOM 2004, pp. 15001-1511, 2004.

[15] Y. Cui, B. Li, and K. Nahrstedt, “oStream: Asynchronous streaming multicast in application layer overlay networks,” IEEE JSAC, Special Issue on Recent Advances in Service Overlays, Vol. 22 (1), pp. 91-106, 2004.

[16] C. Qu, B. Li, and S. Zhao, “Exploring large-scale peer-to-peer live streaming topologies, Transactions on Multimedia Computing,” Communications, and Applications (TOMCCAP), Vol. 4, Issue 3, 2008.

[17] S. Ichimura, “Delivering Chalk Talks on the Internet,” 40th Hawaii International Conference on System Sciences (HICSS), Collaboration Systems Track, IEEE CS Press, pp. 4-11, 2007.

[18] PANDORA.TV, http://www.pandora.tv/.

(Received July 28, 2010)
(Revised August 15, 2011)

Yoshia Saito received his Ph.D. degree from Shizuoka University, Japan, in 2006. He had been an expert researcher of National Institute of Information and Communications Technology (NICT) from 2004 to 2007, Yokosuka, Japan. He is currently a lecturer at Iwate Prefectural University since October 2007. His research interests include internet broadcasting and interactive TV. He is a member of IEICE, IPSJ, IEEE, and ACM.

Yuko Murayama is a professor at Iwate Prefectural University. She had M.Sc. and Ph.D. both from University of London in 1984 and 1992 respectively. She had been a visiting lecturer from 1992 to 1994 at Keio University, a lecturer at Hiroshima City University from 1994 to 1998. She has been with Iwate Prefectural University since April 1998. Her interests include internetworking, network security and trust. She is a member of IEEE, ACM, IPSJ, IEICE, and ITE.