A Study on Dynamic Adaptive Streaming System over HPPT for Various Streaming Services

Chang-Su Kim[1], Bong-Im Jang[2] and Hoe-Kyung Jung[3]*

[1]*Business Support Division, Daejeon Technopark, Korea*
[2]*Department of Multimedia Engineering, College of Engineering, Hannam University, Korea*
[3]*Department of Computer Engineering, College of Engineering, Paichai University, Korea*

nagirong69@gmail.com, jangg11@nate.com, hkjung@pcu.ac.kr (Corresponding author)

Abstract

The streaming service over conventional HTTP has received only one kind of screen quality image which is provided by server. The contents which were provided by the streaming service provider to the general users have been only high definition image. Also, the contents business units who used to establish existing streaming service can provide various streaming services through web modulation.

In conventional streaming service, simple method of one content is transmitted to client was used, whereas streaming technology without blockage suiting to network interference and communication environment at real time using dynamic adaptive streaming over HTTP was implemented in this study

In this study, it was also focused on the validation of streaming service which conventionally used to provide with one original image by developing and testing the images adaptively played with BITRATE suiting to network status through converting original image into BITRATE of several screen quality to provide streaming service without blockage even under limited network environment.

Keywords**:** *DASH, HTTP Streaming, MPD, QoS*

1. Introduction

With recent increases in multimedia contents such as UCC and VoD etc., and various services like IP-TV, Smart TV, and Open Hybrid TV(OHTV) etc. are being provided in multi-platform environments, QoS issues for the service is one of the main concerns that should be address to be solve. For the resolution of this QoS issues, the system to efficiently utilize network and to improve contents quality become necessary.

The streaming service over conventional HTTP has received only one kind of screen quality image which is provided by server. The contents which were provided by the streaming service provider to the general users have been only high definition image. Also, the contents business units who used to establish existing streaming service can provide various streaming services through web modulation.

Also, streaming mode either uses exclusive streaming protocol like Real-Time Streaming Protocol (RTSP) etc. or receives required image at current time from media serve using exclusive line. However, recently in YouTube, Video etc. progressive download system which does buffering and playing media contents using web standard protocol HTTP is

widely being spread. Unlike existing streaming mode, the transmitting type using web standard protocol has merit of accessing media service using internet whenever and anywhere by utilizing conventional structural system which supports HTTP as such and effective using network by using caches or proxy servers that exist on internet. Besides, there is another merit of not generating issue of NAT/Firewall just like in the case of transmitting protocol such as RTP/RTSP, therefore it is widely being used as video sending mode over web in current days. Further, recently HTTP adaptive streaming(HAS) technology draws attention as a hybrid transmitting mode which focuses merit of HTTP protocol and streaming mode that request required image directly to server and receives service for play [1-5].

Therefore, in this study dynamic adaptive streaming system over HTTP was designed and implemented. Dynamic adaptive streaming over HTTP means when video is played and reproduced, by considering interference such as network or traffic etc. it provides screen quality of BITRATE suiting to current network status. Therefore, MPEG DASH related research was carried out and DASH server and client system based on basic on-demand profile was designed and implemented.

2. Related Researches

2.1 Dynamic Adaptive Streaming over HTTP (DASH)

DASH-Dynamic Adaptive Streaming over HTTP provides media contents from standard HTTP server to DASH compatible clients and makes them activated and designates types to cashing the contents of standard HTTP caches. Also, DASH MPD is to provide streaming service to the users by accessing segments through protocols designated in schema of defined resources. It provides sufficient information to the clients enable to access DASH type MPD [6,7].

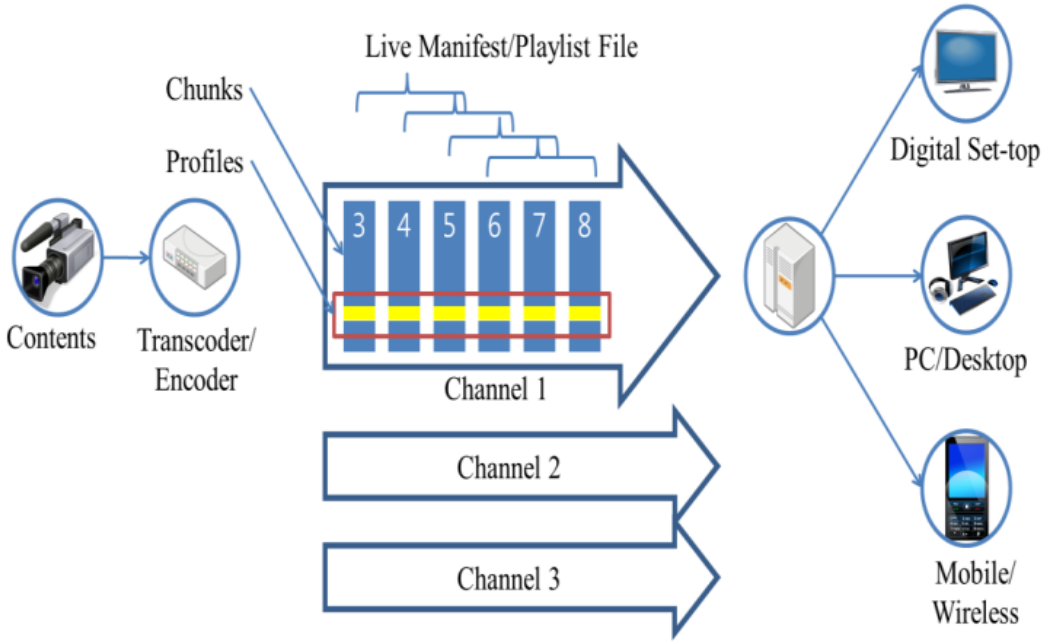

Figure 1. Streaming Structure Through HTTP

The streaming structure through HTTP is shown in Figure 1. It explains the structure that original multimedia contents create files of metadata and chunk unit through Transcoder and

Encoder, these files are registered in HTTP server, and then DASH compatible clients are attached and reproduced.

In DASH, segment files are stored in server and information which are defined in MPD file are analyzed and receives segments which have been stored in server using HTTP in clients. In DASH client, play is executed with MPD and segment, in that Access Client is the part which brings URI among segments properties that have been defined in MPD and takes media data using HTTP, whereas Media engine is the part which analyzes these contents so as finally to make the users watch images.

2.2 Media Presentation Description (MPD)

MPD is an actual medium in DASH for the transmission and reception in server as well as in client. Overall structure of MPD documents are classified into 4 parts that is, MPD, Period, Representation, and Segments [4, 5].

MPD is a data structure which access in DASH clients. It provides downloaded information validated in MPD to provide streaming service to DASH clients' requests and users. All the updated information should be included in MPD as much as possible.

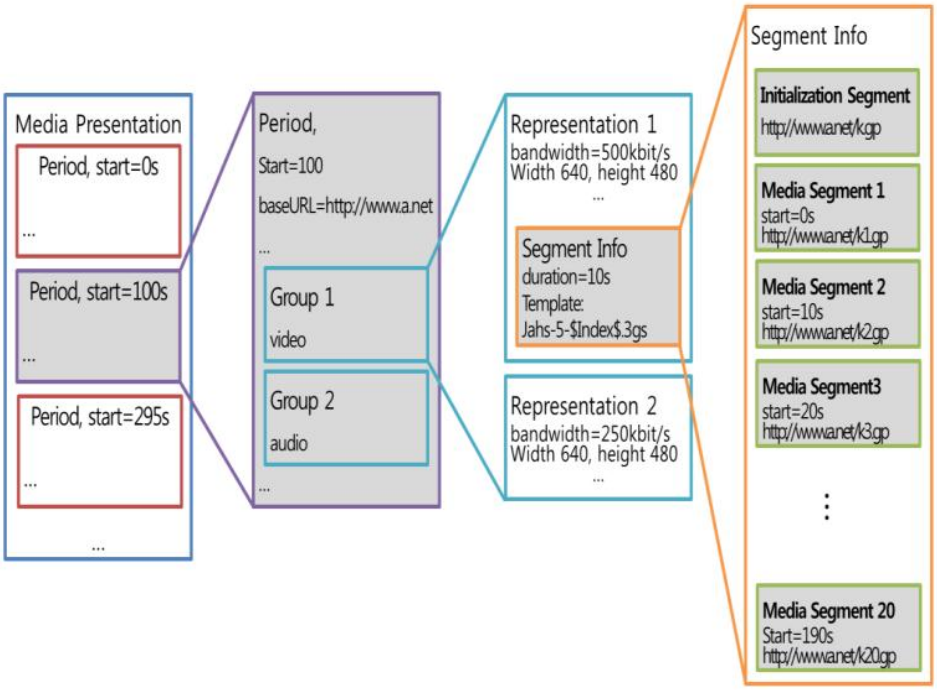

Figure 2. MPD data Structure

MPD data structure should satisfy below details as in Figure 2.

- Periods more than one (1) are sequentially existed.
- Each period includes more than one (1) representation for the same media content.
- Each representation is constituted of more than one (1) segment and can be grouped also.
- Segment includes media data or metadata for decoding.

3. Designing the DASH System

In this Chapter, We illustrate and describe the DASH system designing by classifying it into client and server. Though server uses conventional HTTP, the work that converts multimedia contents which are input into this server into DASH type is called as DASH server. In DASH client, MPD documents are downloaded, analyzed, and validated based on URL which has been stored in web server. On the basis of contents which are described in MPD documents, media segment is requested to server and after receiving segment suiting to network status, this segment is reproduced.

3.1 Designing the DASH Server System

In DASH server, the process is largely constructed with 3 stages. Figure 3 shows the process stages which are Bitstream creation stage, segment creation stage, and finally MPD document creation stage. In Bitstream creation stage, original multimedia file is converted into audio video codec which has been designated in DASH, bit stream is created by receiving codec information, BITRATE, definition, and frame rate etc. which are required at representation level.

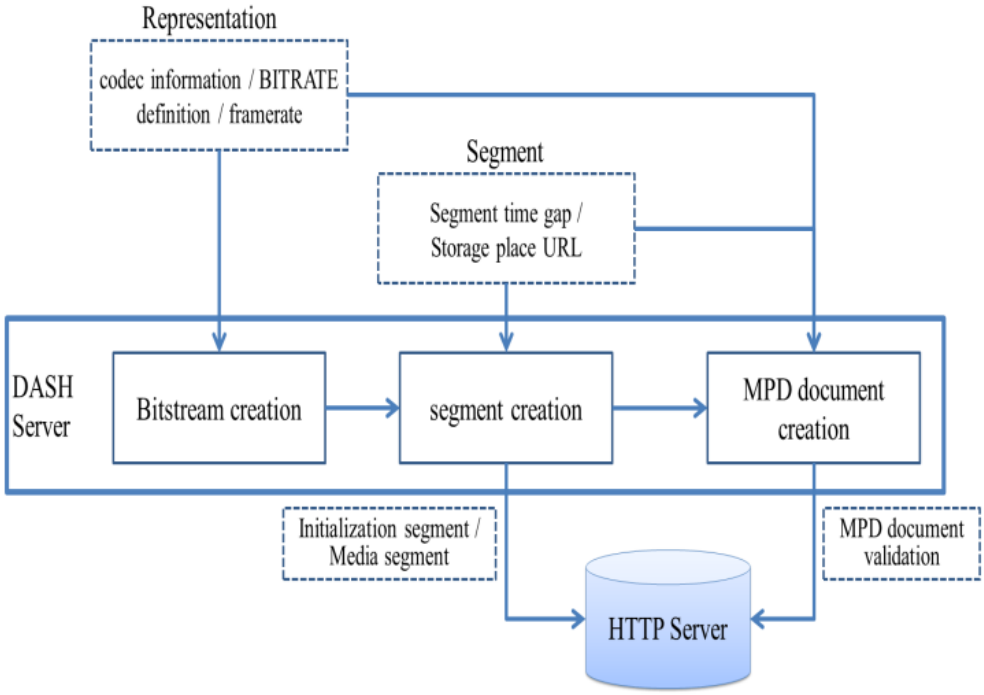

Figure 3. DASH Server Structure

During segment creation stage, time gap, storage place URL which are being used at segment level are received, media data are divided as specified time gap and then initialization segment and media segment which are required for reproduction are created. During this time, it is decided whether to go MPEG2-TS or ISO Base Media File Format (ISOBMFF). Segment is largely classified into two (2) i.e., initialization segment and media segment. Initialization segment can be omitted but at this time, initialized information should be included in the media segment.

In MPD document creation stage, documents are created matching with DASH schema and these documents are validated and stored in HTTP server.

3.2 Designing the DASH Client System

DASH client is largely constructed with 3 stages as shows in Figure 4. First stage comes as metadata analysis. During this stage, MPD file is requested and received from server and then this MPD file is analyzed and checked if this MPD document is correct one and judges if this MPD is that one which can be supported in client. At this time the information about structure, location which is existed in DASH server and server type etc. are collected. In this study, only On-Demand type is supported whereas if it is Live, it marks the contents which are not supported. Also Multiple-Views, SVC views, and Content Protected by Multiple Schemes are not supported but only On-Demand form is supported.

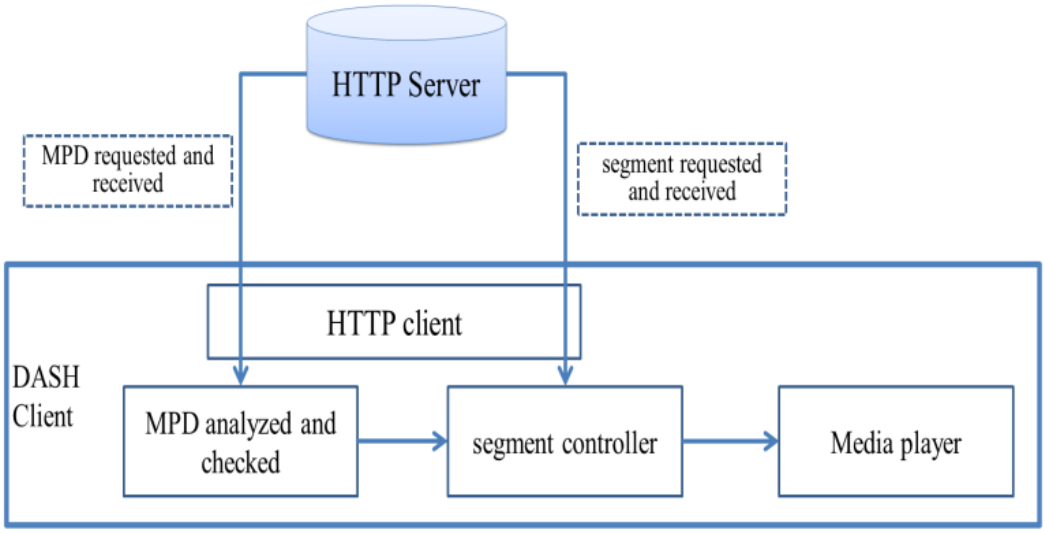

Figure 4. DASH Client Structure

Second, Segment Controller, the segment which is the most suiting to network through analysis of network status on the basis of measured network transmitting speed while MPD document is downloaded is requested over HTTP-URL.

Lastly, in the media player, audio and video signal are categorized based on the stream which has been received from segment controller, these are stored at each buffer. Decoding is executed using media codec prepared in player on the basis of contents which have been stored at each reception buffer, and by checking and adjusting segment sequence of time information for received segments to be played without blockage, these are reproduced in the media player.

4. Implementation of DASH System

4.1 Implementation Environment

As implementation contents, among two kinds of video files of MPEG2-TS file and MP4 file, video format can be used as desired by user. Video representation was constructed into 3 representations that 128, 256, and 512 were used for the first part, 512, 1024, and 2048 were used for the 2nd representation, and 1024, 2048, and 4096 were used for the 3rd

representation. The unit of representations in three parts was "kbps". The time unit of segments which divides contents was set as 10 sec.

The construction of server in DASH implementation is one for one (1:1) and multiple gateways. Client can receive segment data either through one server or from several dispersed servers.

4.2 Implementation of DASH Server

The DASH server implemented image is constructed as Dash Converter part to convert desired BITRATE which calls original contents, Dash Segment part to divide original contents into specific size, finally the part from where final output MPD documents can be created and validated.

In Dash Converter, for one (1) original image three(3) Bitstream contents are created, thus each BITRATE can be set up as desired for designated set up (128,256,512)kbps, (512,1025,2048)kbps, and (1024,2048,4096)kbps. Other than BITRATE which has been already set up, it is possible to set up BITRATE as desired by administrator. Figure 5 shows validation result of MPD documents which are created by applying BITRATE (128,512, 2048) kbps on the original contents.

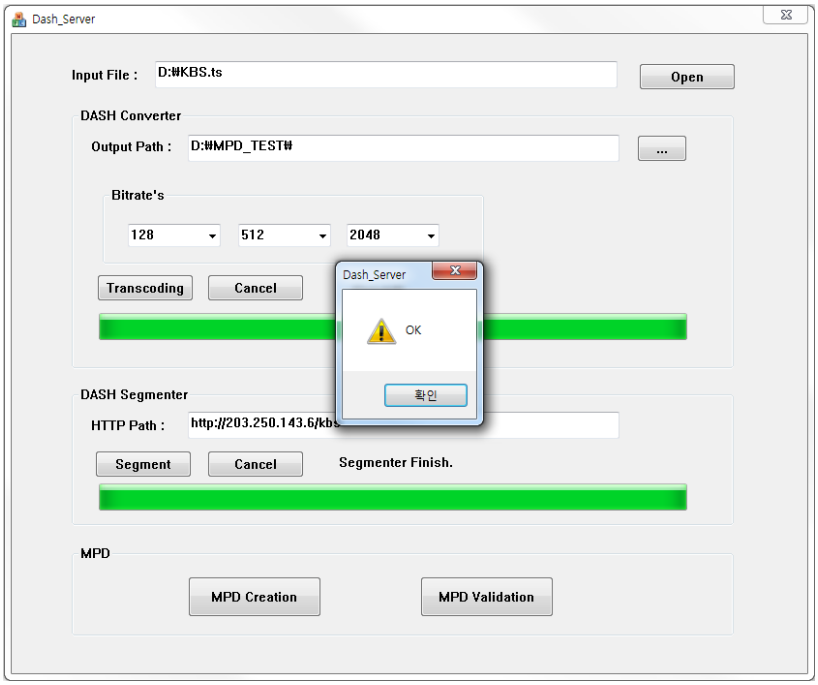

Figure 5. Validation Result of MPD Documents

4.3 Implementation of DASH Client

Below Figure 6 shows the first reproduced image when execution screen of DASH client and MPD document are input. In this part, URL of server and MPD document are only entered in clients and once play button is pressed client downloads MPD documents and does passing to receive segment files which have been defined inside MPD into server and reproduces image per BITRATE compatible to current network status.

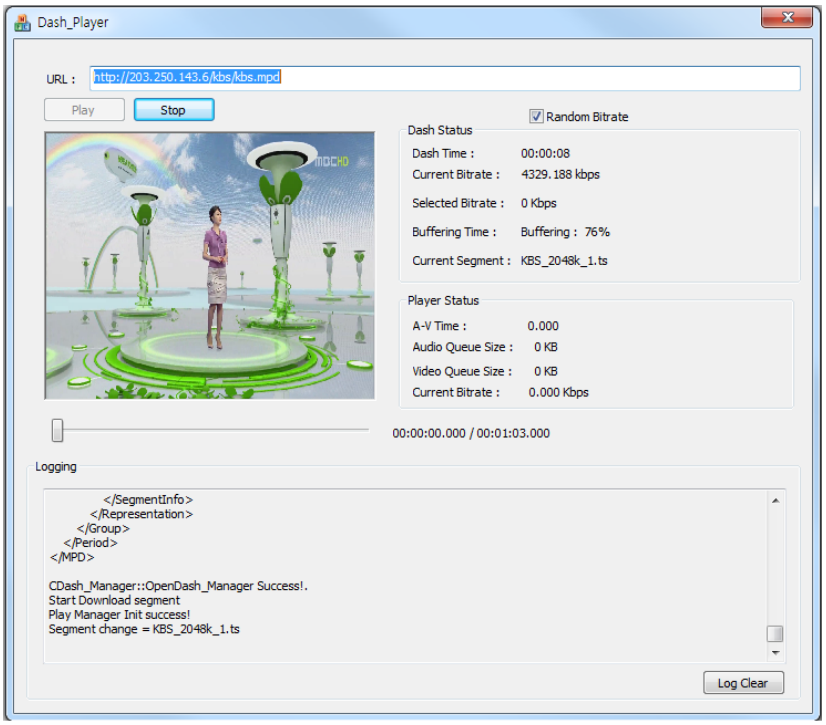

Figure 6. Execution Screen of DASH Client

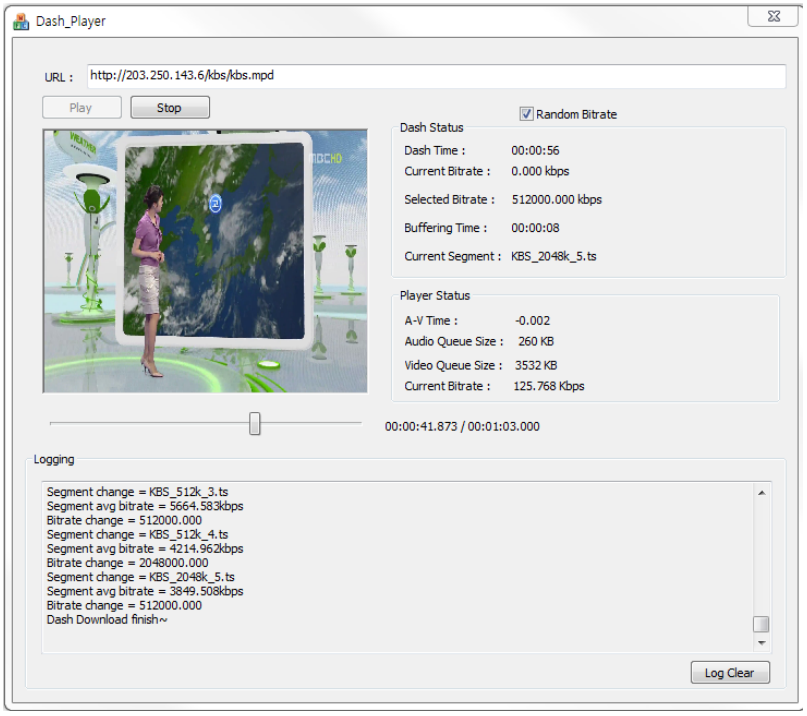

Figure 7. DASH Client Reproduction

Figure 7 shows DASH client reproduction image. In this screen, inside Edit box at right side of URL, once the route in which server address and MPD documents are included are entered and play key is pressed, information about MPD below logging and the segments are downloaded suiting to current network status. Same as these operations, at right side of reproduced image, Dash Status and Play Status can be confirmed that current reproduction time, current BITRATE, selected BITRATE, buffering time, and current segment file can be checked from Dash Status whereas time gap of Audio and Video, reproduction quality of audio and video, and BITRATE currently being reproduced can be checked from Play Status.

5. Conclusion

In conventional streaming service, simple method of one content is transmitted to client was used, whereas streaming technology without blockage suiting to network interference and communication environment at real time using dynamic adaptive streaming over HTTP was implemented in this study. Unlike existing mode of providing contents suiting to only one set up BITRATE, in dynamic adaptive streaming service administrator converts one content to content with BITRATE suiting to network environment and the converted content is divided into each segment and stored in the server and then divided information is made into MPD document, therefore server provides only MPD document to client. Client then does passing MPD document and get provided segment contents which have been stored in server suiting to current situation while checking current network status to make blockage-less streaming service possible even under limited network environment.

Through DASH system designing and implementation in this study, it was confirmed that streaming service compatible to network status can be provided.

In this study, it was also focused on the validation of streaming service which conventionally used to provide with one original image by developing and testing the images adaptively played with BITRATE suiting to network status through converting original image into BITRATE of several screen quality to provide streaming service without blockage even under limited network environment. This dynamic adaptive streaming service over HTTP is judged as improving satisfaction by providing good quality service to the clients, and it can sufficiently supplement overload of concentration at server, further it could be a platform of infinite possibility in IP-TV and Smart TV business field.

As a future research task, it is required to carry out quality of digital contents by applying it on cloud computing or set-top boxes not on the current PC environment, and researches about "Live" type service through optimization of performance of video and audio and contents protection are also required.

References

[1] R. S. Cruz, M. S. Nunes and J. P. Goncalves, “A Personalized HTTP Adaptive Streaming WebTV”, in Proceedings of the First International Conference on User Centric Media Workshop, **(2009)**.
[2] H. Schulzrinne, “RTP:A Transport Protocol for Real-Time Application”, IETRF RFC 3550, **(2003)**.
[3] X. Cheng, C. Dale and J. Liu, “Understanding the characteristics of internet short video sharing: YouTube as a case study”, Technical report, Cornell Univ., **(2007)**.
[4] A. Zambelli, “IIS Smooth Streaming Technical Overview”, Microsoft Corporation, **(2009)**.
[5] ISO/IEC, “Information technology-MPEG systems technologies-Part 6: Dynamic adaptive streaming over HTTP (DASH)”, ISO/IEC CD 23001-6: 2010. 10, **(2010)**.
[6] ISO/IEC, “Information technology-Coding of audio-visual objects-part3: Audio”, ISO/IEC 14496-3(MPEG-4), **(2005)**.
[7] ISO/IEC, “Information technology-Coding of audio-visual objects-part11: Scene description and application engine”, ISO/IEC 14496-11(MPEG-4), **(2005)**.

Authors

Chang-Su Kim

He received a B.S., M.S. and Ph.D. degree from the Department of Computer Engineering from Paichai University, Korea, in 1996, 1998 and 2002, respectively. From 2005 to 2010, he worked for Department of Internet Chungwoon University as a professor. Since 2011, he has worked in the Business Support Division at the Daejeon Technopark, where he now works as a section chief. His current research interests include multimedia document architecture modeling, web 2.0, and the semantic web.

Bong-Im Jang

She received the M.S. and Ph.D. degree from the Department of multimedia Engineering from Hannam University, Korea, in 2003 and 2012, respectively. Since 2010, she has worked in the Business Support Division at the Daejeon Technopark. Her current research interests include USN, Sensor Web, Multimedia and Web Service.

Hoe-Kyung Jung

He received the B.S degree in 1987 and Ph. D. degree in 1993 from the Department of Computer Engineering of Kwangwoon University, Korea. From 1994 to 2005, he worked for ETRI as a researcher. Since 1997, he has worked in the department of Computer Engineering at Paichai University, where he now works as a professor. His current research interests include multimedia document architecture modeling, information processing, information retrieval, and databases.