An Adaptive Buffering Method for Practical HTTP Live Streaming on Smart OTT STBs

Hyun-Sik Kim[1], Inki Kim[2], Kyungsik Han[2], Donghyun Kim[3], Jong-Soo Seo[1] and Mingoo Kang[3]
[1]School of Electrical and Electronic Engineering, Yonsei University, Seoul, South Korea
[e-mail: hskim@keti.re.kr, jsseo@yonsei.ac.kr]
[2]Quber Co. LTD, Pangyou, South Korea
[e-mail: {ikkim,ihanks}@quber.net]
[3]Divisions of Information & Telecommunication, Hanshin University, South Korea
[e-mail: ditel89@gmail.com, kangmg@hs.ac.kr]
* Corresponding author: Mingoo Kang

Received October 17, 2015; revised January 16, 2016; accepted February 18, 2016; published March 31, 2016

Abstract

In this paper, we address the channel zapping time problem of video streaming services based on HTTP Live Streaming (HLS) on smart Over-The-Top Set-Top Boxes (OTT STBs). Experimental analysis of the channel zapping time, show that smart OTT STBs inevitably suffer from the accumulated zapping time through channel change request, Internet Group Management Protocol (IGMP) leave/join, synchronization delay, video buffer delay, and STB processing delay when providing HLS services. As a practical solution for the zapping time reduction, an adaptive buffering method is proposed. The proposed method exploits two adaptive buffers added to the basic HLS player. These two adaptive buffers are responsible for constantly buffering previous and next channels relative to the current channel. Implementation and test results show that a stable zapping time less than one second can be achieved even under diverse video bitrate changes and varying network conditions by the proposed adaptive buffering method.

Keywords: Smart OTT, HTTP Live Streaming, zapping time, adaptive buffering

http://dx.doi.org/10.3837/tiis.2016.03.028 **ISSN : 1976-7277**

1. Introduction

The HLS standard was adoped by Apple for delivering video to mobile devices over varying bandwidths [1]-[2]. Adaptive streaming was developed to enable delivery of audio and video to multimedia devices from a standard HTTP web server. HTTP Live Streaming (HLS) has rapidly become the de facto standard for multimedia streaming to smartphones, tablets, smart Over-The-Top Set-Top Boxes (OTT STBs), and diverse multimedia enabled devices. There are many smart OTT STBs on the market that support HLS, such as Apple TV, Roku 3, D-Link MovieNite Plus, and Boxee Cloud DVR [1]-[3]. However, the user experience that comes with HLS services on smart OTT STBs has been rather unsatisfying. Viewers expect the same level of user experience that comes with traditional television viewing in smart OTT STBs. However, users have been experiencing long zapping time delays when using HLS based services on smart OTT STBs. The zapping time or zap time is the duration of time from which the viewer changes the channel to the point that the picture of the new switched channel is displayed on the screen[4]. In this paper, a method for zapping time analysis for HLS based services on smart OTT STBs is presented. Based on the analysis, a solution to reduce the channel zapping time is proposed. The remaining of this paper is organized as follows. In Section 2, we address and analyze the channel zapping time problem. Section 3 describes the proposed adaptive burrering method. The implementation and test results of the proposed method as well as its performance analysis are provided in Section 4. Concluding remarks are commented in Section 5.

2. Analysis of Channel Zapping Time on Smart OTT STBs

In analog TV broadcast and cable technology, channel change is almost instantaneous since it only involves the TV receiver tuning to a specific carrier frequency, demodulating the content and displaying it on the TV screen. The zapping delay in these systems is typically less than 200 ms. Therefore, TV viewers consider zapping times to be virtually instant, and have become used to this surfing (through channels) experience. However, with the introduction of digital video broadcasting or streaming which includes content compression, zapping times have increased significantly [5].

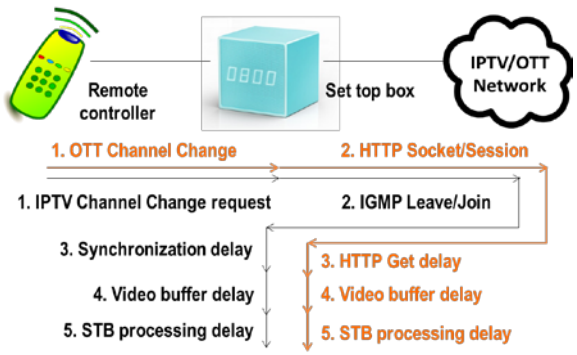

Fig. 1. Channel change procedure and zapping time delay on smart OTT STBs

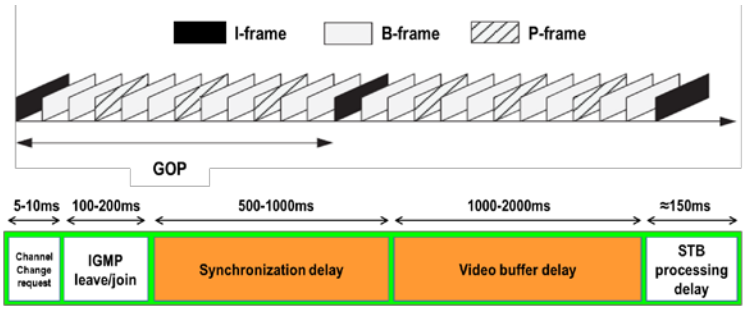

Fig. 2. Experimental measurements and analysis of channel zapping time for HLS services

In IPTV/OTT video delivery, contrast to cable networks, for instance, typically only the channel the user is watching is delivered to the set-top box (STB) at any one time. This is due to bandwidth limitations in the access network. When a user switches to a new channel, the STB has to issue a new channel request towards the network. Since video distribution is done via multicasting, this is translated into leave/join multicast requests. These operations are handled by a group management protocol, IGMP (Internet Group Management Protocol) [5].

The smart OTT STB sends a leave request from the current multicast session (the TV channel the user is currently watching) and a join request to the new multicast group (the TV channel the user is switching to). This channel change request reaches the first upstream network node that has the channel available and the routing infrastructure sets up the multicast forwarding state to deliver the packets of the multicast session to the STB.

After the STB receives the first set of packets from the recently joined multicast group, there is still a time lag before it can start consuming the audiovisual data because the STB must wait for the next I-frame before it can start decoding the content. The maximum synchronization delay is equal to the duration of the Group of Pictures (GOP) [6]-[7], which occurs when the STB just misses the start of an I-frame and thus has to wait for the next. On average, this delay is half the GOP duration. Therefore, synchronization delay comprises a substantial portion of the channel change time (recall that GOP duration is typically in the range of 1 to 2 seconds) [8].

Many video services also employ content encryption, so the encryption keys must be acquired and provided to the decryption engine for decrypting the content, and this also adds to the synchronization delay [5]-[8].

Therefore, adaptive bitrate design for reducing the zapping time of HLS based services in smart OTTs can be challenging, especially if many different encoding profiles are required [13]-[16]. Even for small, standalone OTT deployments, flexibility is required in the encoding solution. For hybrid IPTV and OTT deployments, which have larger subscriber bases, scalability for adaptive stream buffering is challenging [18]-[20].

In this paper, a smart OTT is designed to meet the live multicast and adaptive bitrate (ABR) encoding challenge by using HLS based on dynamic bitrate design of adaptive stream buffering.

3. Proposed Adaptive Buffering Method

Zapping time is assossiated with channel transition, STB tuner operation, STB stream server, HLS streaming, and video presentation [8]-[12].

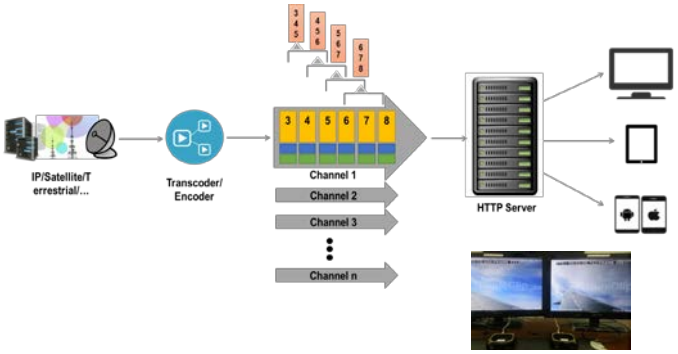

Fig. 3. Design and implementation of HLS based adaptive streaming on smart OTT STBs

In oder to optimize adaptive bitrate on smart OTTs, HLS based adaptive streaming mode is designed as shown in **Fig. 3**. The analysis of zapping time is proposed with a proxy as shown in **Fig. 4**.

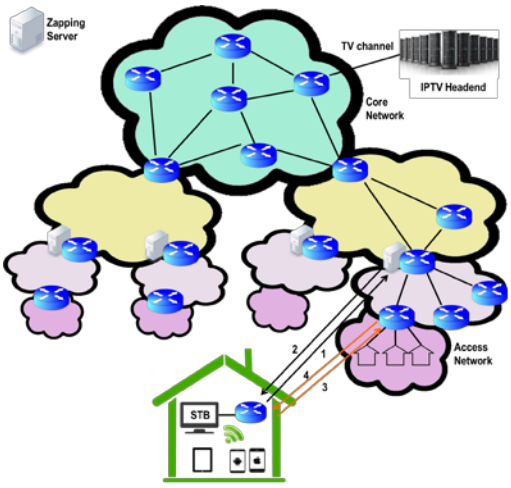

Fig. 4. Zapping time analysis on smart OTT STBs with proxy

3.1 Design of Single Buffering

Single buffering is a buffer structure that provides dynamically seamless natural media conversion among multiple media streams in the same channel. The sequence of operations and the framework structure of single buffering are shown in **Fig. 5**.

As a first step, a media is created and prepared for the media channel. The listener for playing the media is also registered. The audio and video media types are set and media starts playing. An m3u8 type of media streaming HLS address is needed for media configuration. An HLS Streaming Server is needed for streaming. In addition, a media address needs to be assigned for each bitrate to enable dynamic buffering. These addresses must be defined in the m3u8.

Next, before the end of the current media segment, a media player is created for the next media segment, which involves listener registration for media play, audio/video media type setting, and proceeding all the way to "Prepare" state. If onPrepared is called when the media becomes ready for one of the registered listeners, the setNextMediaPlayer of the first media is called, and the next media player is registered. The point of next media player generation can be set freely depending on system capabilities.

Assuming that the second media's "Prepare" is completed during the first media play and the setNextMediaPlayer of the first media is normally set, if the first media is finished, then the registered second media starts automatically, enabling fast and natural media transition. If the first media is finished, but a media loop is created so that the first media starts replaying, the next media maintains "Prepare" without playing, and the first media starts again.

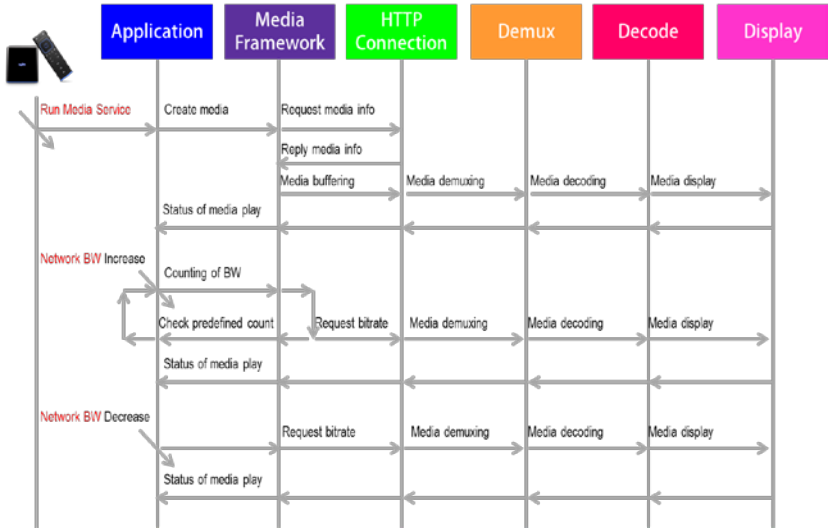

Fig. 5. Design of single buffering

3.2 Design of Adaptive Buffering

Adaptive buffering is a buffer structure that allows dynamic, natural, and fast media conversion in multiple media channels. Multiple adaptive buffering provides the design and optimization of a buffering structure that minimizes the buffering time to enable fast channel transition when different media channels are dynamically played. The operation involves adaptive buffering of the previous and next media channels relative to the current media channel to enable fast transition when media transition occurs. For services that involve operation of multiple media channels, the proposed method can improved the quality of service by providing fast channel transition and used as a model for efficient bandwidth management. The sequence of operations and the framework structure of multiple buffering are shown in **Fig. 6**.

A media is created and prepared for the media channel. The listener for playing the media is registered, the audio/video media types are set, and media play is started. An m3u8 form of media streaming HLS address is needed for media configuration, and an HLS Streaming Server is needed for streaming. In order to enable dynamic buffering, each bitrate has to have an address which needs to be defined in m3u8.

At the point of current media generation, the media players for previous channel and next channel media play are generated. The listeners for playing the media are registered, the

audio/video media types are set, and the process is continued until “Prepare” state. If onPrepared is called when the media becomes ready for one of the registered listeners, the setNextMediaPlayer and setPrevMediaPlayer of the first media are called, and the next/previous media players are registered.

If the user requests channel transition to the previous channel, the media registered to setPrevMediaPlayer is automatically played. If the user requests channel transition to the next channel, the media registered to setNextMediaPlayer is automatically played.

If channel transition to previous/next channel is completed, the next and previous channels based on the current channel are registered using setNextMediaPlayer and setPrevMediaPlayer, and the previous existing media are deleted.

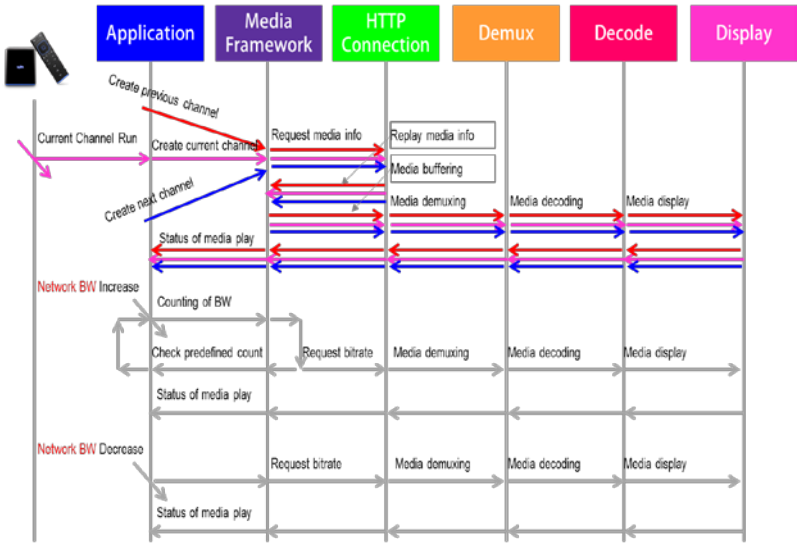

Fig. 6. Architecture design of adaptive buffering

Buffering reduces the smart OTT system sensitivity to short term fluctuations in the data arrival rate by absorbing variations in end-to-end delay and allow margins for retransmissions when packets are lost. There are three other reasons to buffer incoming packets before forwarding them to the decoder as shown in **Fig. 7** [8].

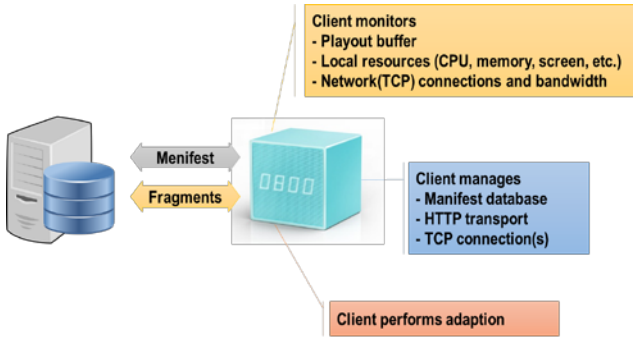

Fig. 7. Role of the client in HTTP ABR streaming on smart OTT

4. Implementation and Test Results

In order to apply adaptive buffering, two buffers are added to the basic HLS player, each responsible for constantly buffering the previous and next channels relative to the current channel [10]. This reduces the buffering delay after an HLS request, i.e., channel zapping time, when a channel switch event is triggered. The setup for the experiment is as shown in **Table 1**. In the experiment, three videos are each encoded at bitrates of 3M, 6M, 10M, 15M and 20M, and the zapping delay is measured during a channel switch [17].

Table 1. Experimental Environment Setup

Devices	**Specifications**	
STB Smart OTT	Hisilicon hi3719M Dual Core CPU based STB RAM 1Gbyte Ethernet 10/100-T	
WiFi_AP	KAON AR3010 802.11 a,b,g,n,ac support Network Bandwidth control (QOS) support	
HLS Server	WowzaMediaServer 3.6.2 HLS, RTP, VOD Streaming Support.	
Channel Information	Channel #1	Animation Running Time : 2m20s Bitrate 3M, 6M, 10M, 15M, 20Mbps
	Channel #2	Animation Running Time : 3m27s Bitrate 3M, 6M, 10M, 15M, 20Mbps
	Channel #3	Flight Video Running Time : 4m18s Bitrate 3M, 6M, 10M, 15M, 20Mbps

4.1 Zapping-time Analysis of Varying Network-bandwidths on Smart OTT STB

The bitrate for each video is set to 6M. The zapping time is measured by changing the network bandwidths from the router. Measurements show that the zapping time for the STB with adaptive buffering is less than 1 sec as shown in **Figs. 8** to **10**. On the other hand, the zapping time for the STB using single buffering was as high as 40 sec. The time decreased to below 2 sec as bandwidth increased, but this is still nearly two times higher than the zapping time of the STB with adaptive buffering.

The channel changing time is calculated using (1) as follows:

$$T = T_{buff} \text{ x } R / B \qquad (1)$$

where T is time interval of channel changing, T_{buff} is the video buffer time in STB Media Player (fixed at 5 sec), R is the video bitrate, and B is the bandwidth. The bandwidth can be changed in the WiFi_AP. The estimation results from (1) are used to compare with the experimental results.

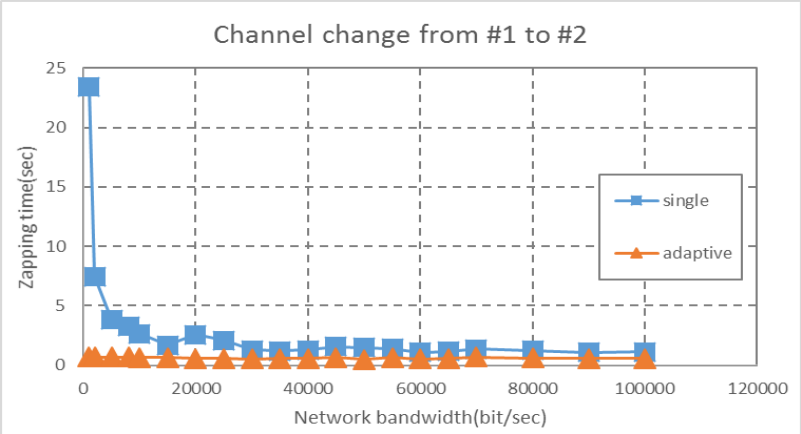

Fig. 8. Zapping time for switching from channel #1 to #2 by changing the network bandwidths

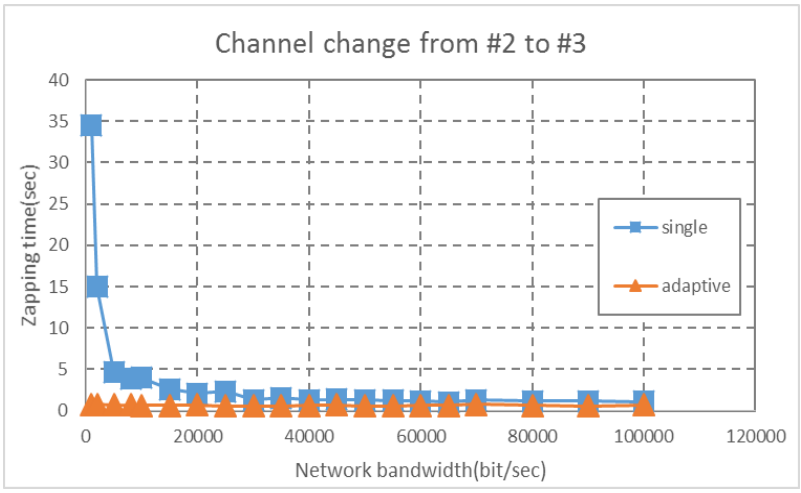

Fig. 9. Zapping time for switching from channel #2 to #3 by changing the network bandwidths

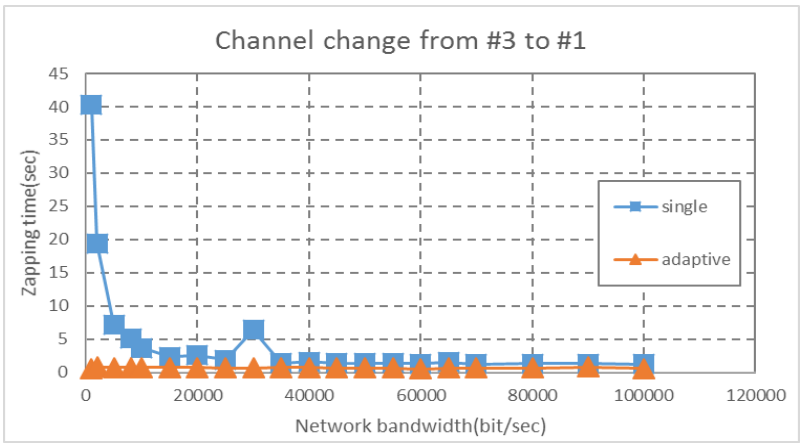

Fig. 10. Zapping time for switching from channel #3 to #1 by changing the network bandwidths

No major differences were observed in zapping time for the different types of video, as seen in **Figs. 11** and **12**. Since the HLS server and the test environment are connected through Internet, unusually long zapping times were measured in some test cases, which is in accordance to the network environment.

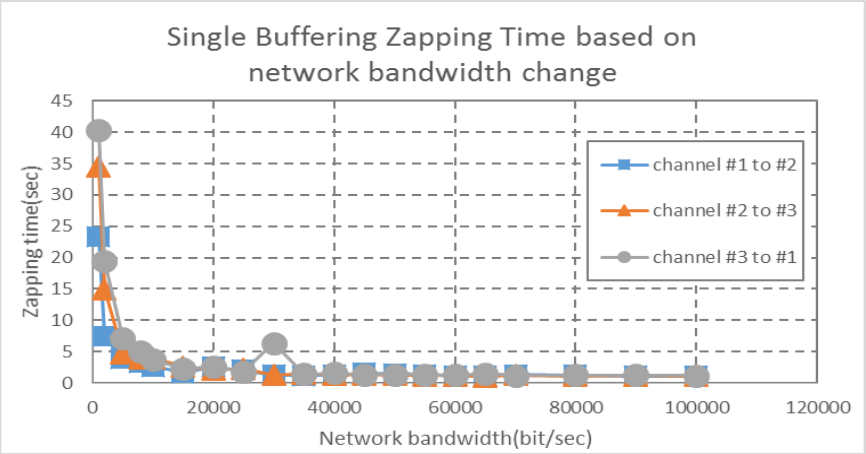

Fig. 11. Single buffering zapping time of channels by changing the network bandwidths

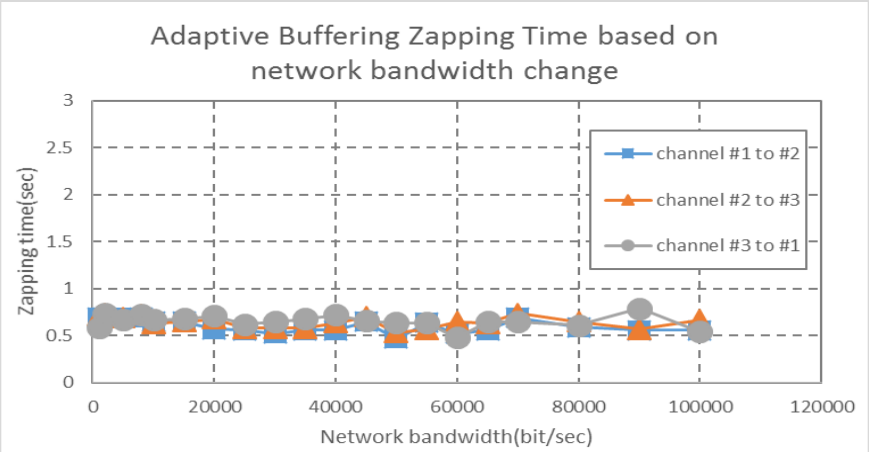

Fig. 12. Adaptive buffering zapping time of channels by changing the network bandwidths

4.2 Zapping-time Analysis of Varying Video Bitrate on Smart OTT STB

From the network bandwidths variation tests, it was observed that when the bandwidth is greater than 20 Mbps, there are no significant differences in zapping time. Therefore, the AP's bandwidth was set to a constant 20 Mbps, and the zapping time was measured by changing the video bitrate. To obtain accurate results, the zapping time was measured 3 times for each stage. **Figs. 13** to **15** show the average of the measured zapping times.

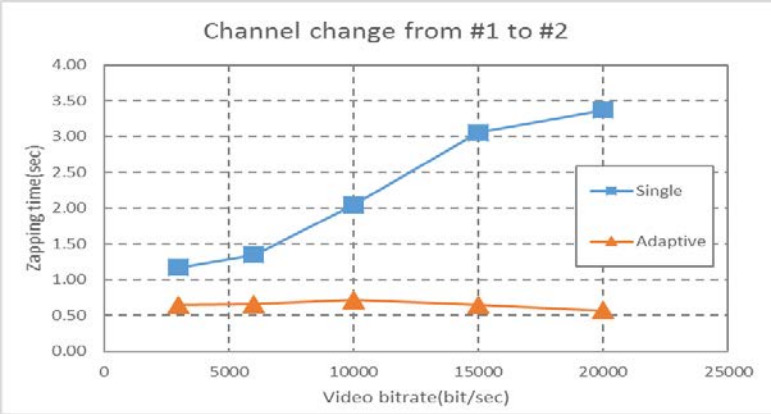

Fig. 13. Zapping-time analysis of switching from channel #1 to #2 by changing the video bitrate

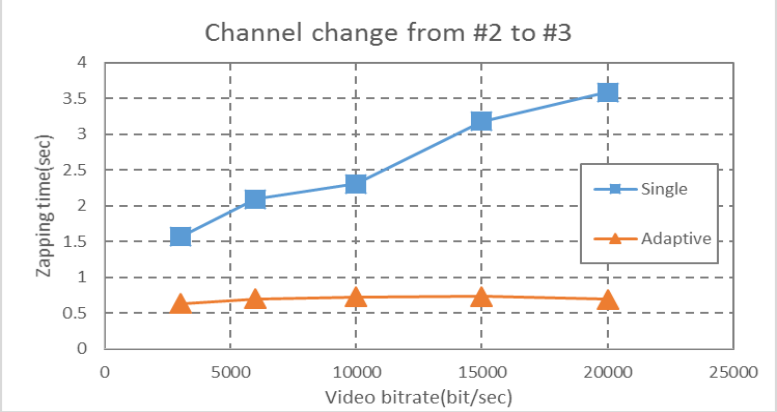

Fig. 14. Zapping-time analysis of switching from channel #2 to #3 by changing the video bitrate

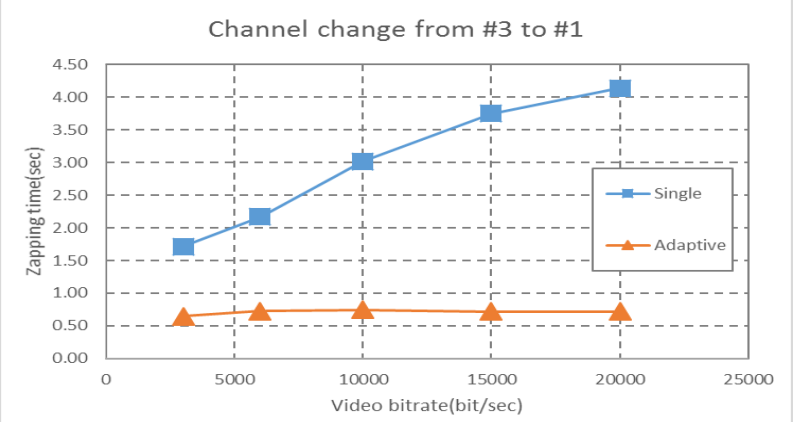

Fig. 15. Zapping-time analysis of switching from channel #3 to #1 by changing the video bitrate

It can be observed that the zapping time for the single buffer increases linearly when video bitrate is changed. However, the zapping time for STB with adaptive buffering is nearly unaffected by changes in video bitrate. The zapping time for the different video bitrates are shown in **Figs. 16** and **17**.

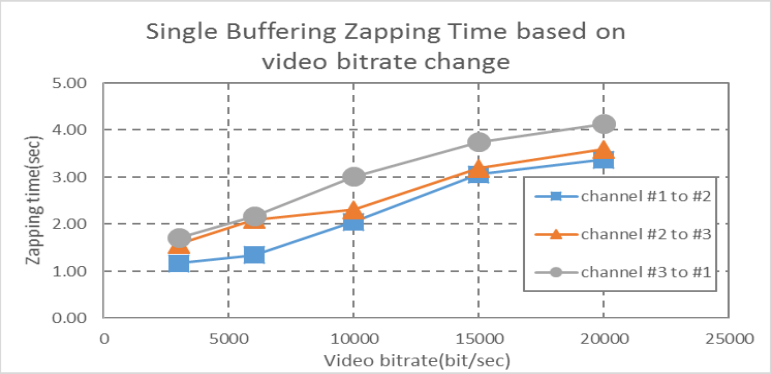

Fig. 16. Zapping-time analysis of channels by changing the video bitrate on single buffering

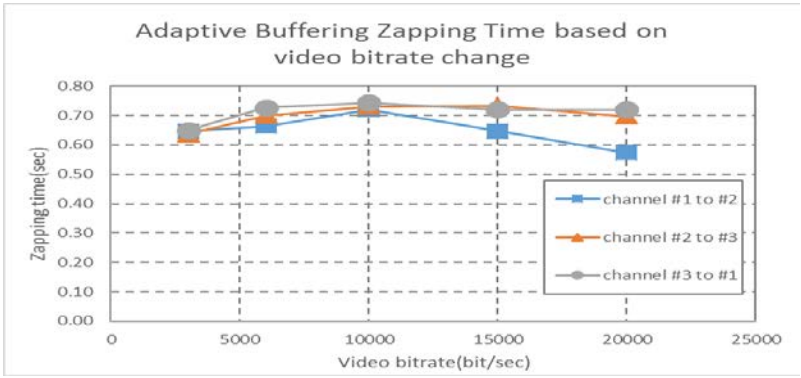

Fig. 17. Zapping-time analysis of channels by changing the video bitrate on adaptive buffering

The results from **Figs. 16** and **17** show that there are no significant differences in zapping time for the different video types. Through the above series of experiments, it was proven and verified that a stable zapping time less than 1 second can be provided by using adaptive buffering, even under diverse video bitrate changes and varying network conditions.

5. Conclusion

In this paper, an adaptive buffering method was proposed to reduce the zapping time of HLS based streaming services in smart OTT STBs. The inclusion of buffers resulted in zap times of less than 1 sec, which is tolerable by the viewer. Adaptive stream buffering is proposed for the dynamic bitrate of HLS from the different frames encoded at different data rates. Since the network flow is typically constant bitrate (or capped variable), the mismatch between the input and the output of the encoder is solved with the inclusion of a smoothing buffer: in an IP network traffic is asynchronous, so packets have to wait in buffers. While the amount of protection offered by a buffer grows with its size, so does the latency it introduces.

Acknowledgement

This work was supported by a research grant(Advanced Technology Center`#10045816]) of MOTIE/KEIT, the Korean government , and Hanshin University.

References

[1] I.K.Kim et al., "Design of Dynamic Bitrate Optimization based on the Adaptive Stream Buffering of HLS," *KSII The 10th APIC-IST 2015*, Jul. 2015. Article (CrossRef Link)

[2] C. Yang, Y. Li and J. Chen, "A new mobile streaming system base-on http live streaming protocol," in *Proc. of Int. Conf. Wireless Commun., Netw. Mobile Comput.*, 2011. Article (CrossRef Link)

[3] Adobe, 2015. Article (CrossRef Link)

[4] B. Dekeris, and L. Narbutaite, "IPTV Channel Zap Time Analysis," in *Proc. of International Conference on Ubiquitous and Future Networks*, China, 2009. Article (CrossRef Link)

[5] Benjamin Schwarz, "A Harmonic, Viaccess-Orca and Broadpeak OTT White Paper," May 2012. Article (CrossRef Link)

[6] MPEG Group of Pictures. Article (CrossRef Link)

[7] H. Schwarz , D. Marpe and T. Wiegand , "Analysis of hierarchical B-pictures and MCTF," in *Proc. of ICME*, pp.1929 -1932 , 2006. Article (CrossRef Link)

[8] A. C. Begen, N. Glazebrook, and W. V. Steeg, "A unified approach for repairing packet loss and accelerating channel changes in multicast IPTV," *CCNC*, Las Vegas, 2009. Article (CrossRef Link)

[9] B. Cain, S. Deering, I. Kouvelas, B. Fenner, and A. Thyagarajan, "Internet group Management Protocol, version 3," *RFC 3376*, 2002 Article (CrossRef Link)

[10] Chandrashekhara Anantharamu, "SYSTEM AND METHOD FOR ADAPTIVE STREAMING OF PREDICTIVE CODED VIDEO DATA," *United States Patent Application*, US 2002/0136298 A1 Article (CrossRef Link)

[11] Apple, "HTTP Live Streaming Overview," Feb. 2014. Article (CrossRef Link)

[12] dndoding.com "HTTP Live Streaming" Article (CrossRef Link)

[13] X. Tian, Y. Cheng and X. Shen, "Fast channel zapping with destination-oriented multicast for IP video delivery," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24 , no. 2 , pp.327 -341 , 2013. Article (CrossRef Link)

[14] Gil Jin Yang, Byoung Wook Choi, and Jong Hun Kim, "Implementation of HTTP Live Streaming for an IP Camera using an Open Source Multimedia Converter," *International Journal of Software Engineering and Its Applications*, Vol.8, No. 6, pp. 39-50, 2014. Article (CrossRef Link)

[15] Thomas Stockhammer, "Dynamic adaptive streaming over HTTP --: standards and design principles," *The second annual ACM conference on Multimedia systems*, February 23-25, San Jose, CA, USA, 2011. Article (CrossRef Link)

[16] Fernando M. V. Ramos, "GREEN IPTV: a resource and energy efficient network for IPTV," University of Cambridge, 2012. Article (CrossRef Link)

[17] RFC 3550, "RTP: A transport protocol for real-time applications." Article (CrossRef Link)

[18] D. E. Smith, "IP TV bandwidth demand: Multicast and channel surfing," in *Proc. of IEEE Int. Conf. Computer Communications (INFOCOM)*, 2007 Article (CrossRef Link)

[19] A. Takahashi, D. Hands, and V. Barriac, "Standardization activities in the ITU for a QoE assessment of IPTV," *IEEE Comm.*, Mag., vol. 46, no. 2, pp. 78-84, Feb. 2008. Article (CrossRef Link)

[20] AV Acoustics, "HTTP Live Streaming Architecture." Article (CrossRef Link)

Hyun-Sik Kim received the BS and MS degrees in information and communication engineering from the Inah University, South Korea, in 2002 and 2004, respectively. Currently, he is a Ph.D. candidate in the electrical & electronic engineering from the Yonsei University, South Korea. He is also working as a senior researcher in the Contents Convergence Research Center at the Korea Electronics Technology Institute (KETI), South Korea from 2004. His research interests are in contents sharing, wearable computing, human-computer interaction, and Machine-to-Machine/Internet of Things.

Inki Kim is a candidate for Ph.D course from Hanshin University, Osan South Korea in Telecommunication Engineering. He was a research engineer at Samsung Electronics and CTO in KaonMedia and CEO in InnoDigital. His research interests broadcasting including OTT, smart eco-system, mobile device and service application and UI/UX for STB & mobile devices.

Kyungsik Han is a Senior Research Engineer at QUBER, Pangyo South Korea from 2010. He has received the Ph.D. degrees from Hanshin University, Osan South Korea in Information Science & Telecommunications in 2013. He has received the B.S. degrees from Soongsil University, Seoul, Korea in Electronic-Science Engineering in 2004. His research interests include OTT, wired & wireless networking for STB and network streaming system architecture.

Donghyun Kim is a candidate for master course from Hanshin University, Osan South Korea in Telecommunication Engineering. His research interests include embedded platform, mobile devices, and Android Application

MinGoo Kang is a professor in the Division of Information & Telecommunication at Hanshin University, Osan South Korea from 2000. He has received the B.S., M.S., and Ph.D. degrees from Yonsei University, Seoul, Korea all in Electronic Engineering in 1986, 1989 and 1994, respectively. He was a research engineer at Samsung Electronics from 1985 to 1997. His research interests include wireless communication algorithm, mobile devices, and Smart UX for Mobile TV & DTV. He also served the ICONI 2014 hosted by the Korean Society of Internet Information, as the conference chair.

Jong-Soo Seo received the B.S. degree in electronics engineering from Yonsei University, Seoul, Korea, in 1975, and the M.S. and Ph.D. degrees from the University of Ottawa, Ottawa, ON, Canada, in 1983 and 1988, respectively. He was with IDC and CAL, Canada, engaged in research on digital satellite communications and data broadcasting systems for six years. Since 1995, he has been with the Department of Electrical and Electronic Engineering, Yonsei University, where he is currently a Professor. Prof. Seo is a Fellow of IEEE and associate editor of the IEEE TRANSACTIONS ON BROADCASTING. His current research interests include next generation broadcasting and 5G radio systems.