Adaptive Streaming Scheme for Improving Quality of Virtualization Service

Sunghee Lee and Kwangsue Chung
Department of Electronics and Communications Engineering
Kwangwoon University
Seoul, Korea
e-mail: shlee@cclab.kw.ac.kr, kchung@kw.ac.kr

Abstract— **The streaming-based virtualization services require Quality of Service (QoS) support for achieving a seamless display and low latency. Dynamic Adaptive Streaming over HTTP (DASH) has been proposed to support QoS of multimedia transmission. Existing bitrate adaptation schemes based on DASH are unsuitable for virtualization services due to latency problem. This paper proposes a DASH based adaptive streaming scheme to improve the QoS of virtualization service. The proposed scheme provides seamless display by adjusting the quality of the segment based on the segment throughput and the buffer status. It also reduces latency by using the server push mechanism of HTTP 2.0. The simulation results show that the proposed scheme has a better performance.**

Keywords- virtualization service; DASH; QoS.

I. INTRODUCTION

With the introduction of fast and reliable core networks and wide-spread availability of Internet access, a trend towards moving more and more services away from the end devices to remote data centers has established itself. This is widely referred to as streaming-based virtualization service. This results in greatly increased requirements on Quality of Service (QoS) to achieve a seamless display and low latency.

Meanwhile, Dynamic Adaptive Streaming over HTTP (DASH) has been proposed to support QoS of multimedia transmission. DASH can support seamless display by handling the bitrate of contents based on the time varying bandwidth conditions. In DASH, the server response depends on the client's requests when the server is otherwise idle or blocked for that client. To adapt the bitrate of the content according to the network status, the content is divided into short-duration media segments, each of which is encoded at various bitrates and can be decoded independently. During download, the client dynamically picks the segment with the right encoding bitrate that matches or is below the bandwidth, and requests that segment from the server.

Several bitrate adaptation schemes, such as Rate Adaptation for Adaptive HTTP Streaming (RAHS) and Adaptive Streaming of Audiovisual Content (ASAC) have been proposed for improving QoS of DASH. These schemes adjust bitrate of segment based on the ratio of Media Segment Duration (MSD) to Segment Fetch Time (SFT) [1][2]. However, these schemes cannot guarantee the quality of virtualization service because ratio of MSD to SFT cannot estimate the network status precisely due to VBR characteristics of contents. Moreover, DASH clients receive a manifest file, request and download the referred segments over HTTP, and play them back. This procedure introduces additional latency making HTTP streaming unsuitable for virtualization service that requires low latencies [3].

To reduce the latency of DASH, the server push based streaming scheme has been proposed. The server push mechanism pushes a resource directly to the client without the client request. However, server push is not suitable for DASH as the DASH adjusts the bitrate of contents based on the client request. In this paper, we propose a DASH based adaptive streaming scheme to improve the QoS of virtualization service. The proposed scheme provides seamless display by adjusting the quality of the segment based on the segment throughput and the buffer status. It also reduces latency by using the server push mechanism.

The rest of this paper is organized as follows: in Section II, we describe the concepts and algorithms introduced in the proposed scheme. In Section III, we show the simulation results. Conclusions and future works are presented in Section IV.

II. ADAPTIVE STREAMING FOR VIRTUALIZATION SERVICE

A. Overview of adaptive streaming scheme

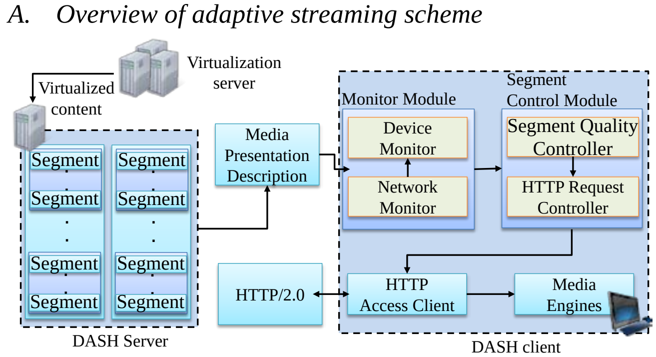

Figure 1. Architecture of adaptive streaming for virtualization service

Figure 1 illustrates the architecture of adaptive streaming for a virtualization service. In this architecture, HTTP 2.0 is used for server push. The proposed system performs segment quality control and HTTP request control algorithm to

 ISBN: 978-1-61208-418-3

reduce playback discontinuity and latency. In order to achieve this, information related to network and device status is monitored at the client. *Device Monitor* measures the remaining frames in a playback buffer, and *Network Monitor* calculates the segment throughput. The network and device information are forwarded to *Segment Quality Controller*. *Segment Quality Controller* decides the bitrate of next segment and *HTTP Request Controller* decides when to send HTTP request to server for preventing playback discontinuity.

Figure 2 depicts the behavior of the proposed adaptive streaming scheme. At the beginning of the service, the client requests virtualized content which has the minimum bitrate because there is no information about the network status and the playback buffer is empty. Then, the server pushes the segments until the client requests another bitrate of virtualized content. Based on this approach, the proposed scheme is better able to reduce latency and prevent playback discontinuity.

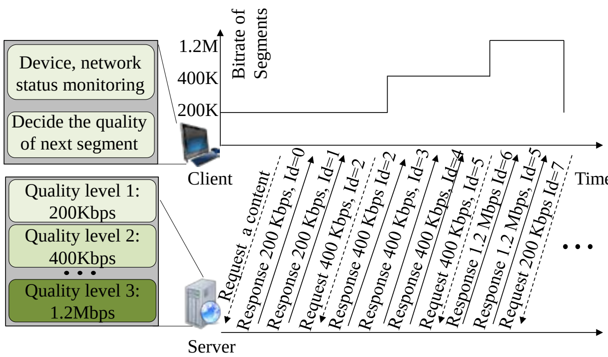

Figure 2. Behavior of the proposed adaptive streaming sceme

B. Quality adaptation algorithm

To provide seamless service, the proposed scheme adjusts timing to send a HTTP request and bitrate segments according to the network and device status. After receiving a segment, the client estimates segment throughput as follows:

$$Th_{seg} = \frac{S_{seg}}{SFT} \quad (1)$$

where Th_{seg} is the segment throughput, S_{seg} is the size of a segment, and SFT is the segment fetch time. Therefore, the video bitrate of the next segment is decided on the basis of the SFT of the most recently downloaded segment. To reduce the latency caused by the HTTP request from a client, the proposed scheme sends HTTP request only when the quality change is required. The quality level increases when the Th_{seg} is higher than the bitrate of currently downloaded segment to improve the quality of the video. On the other hand, the quality level decreases when Th_{seg} is lower than the bitrate of currently downloaded segment and there is not enough data in the playback buffer.

The condition of quality increment is as follows:

$$Th_{seg} > R_{cur+1} \quad (2)$$

where R_{cur+1} is the bitrate of the next higher quality level. If segment throughput is larger than the bitrate of next higher quality level, the proposed scheme increases a quality level as follows:

$$R_{cur} = R_{cur+1} \quad (3)$$

where R_{cur} is the bitrate of current quality level. We use stepwise increment of quality level to prevent buffer underflow when the available bandwidth is drastically decreased after increasing the quality level. The proposed scheme involves switch down the quality level when the amount of playback buffer is not enough to play the video until receiving next segment in the current quality level.

The condition of quality decrement is as follows:

$$T_{buf} < SFT \quad (4)$$

where T_{buf} is the amount of playback buffer in time. If the SFT is larger than the amount of playback buffer, the buffer underflow will occur before receiving next segment. Therefore, the proposed scheme selects the segment that has the maximum video bitrate among video qualities which can be downloaded during the remaining playback buffer time as follows:

$$R_{cur} = \max\{R_n \mid R_n \le \frac{T_{buf} \times Th_{seg}}{MSD}\} \quad (5)$$

where R_{next} is the bitrate of next segment, R_n is the bitrate of n-th quality level.

III. SIMULATION RESULTS

To evaluate the performance, we have implemented the proposed scheme in a DASH reference player developed by DASH Industry Forum (IF). We compared the amount of playback buffer among the proposed scheme, RAHS, ASAC, and default adaptation algorithm of DASH IF player. Figure 3 shows that the proposed scheme stably maintains the buffer level without buffer underflow because the quality is adjusted on the basis of the segment throughput and buffer status.

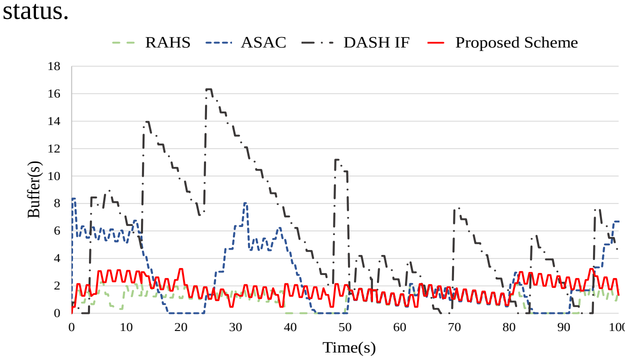

Figure 3. Comparison of the amount of playback buffer

We compared the latency between the proposed scheme with server push and one without server push mechanism. Figure 4 shows that the latency is evidently decreased when the server push mechanism is deployed.

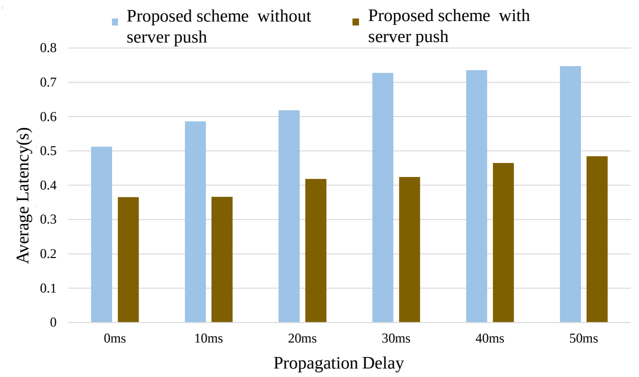

Figure 4. Comparison of latency

IV. CONCLUSION AND FUTURE WORK

DASH can improve the QoS of streaming-based virtualization service by adjusting the quality of content according to the network condition. However, HTTP request/response procedure of DASH introduces additional latency which degrades the quality of virtualization service. In this paper, we propose an adaptive streaming scheme for DASH to improve the QoS of streaming-based virtualization service. The simulation results show that the proposed scheme provides seamless playback and low latency by adjusting the quality of content based on the server push mechanism of HTTP 2.0. In the future work, we will enhance the proposed adaptive streaming scheme to reduce the frequent quality change due to the varying network condition.

ACKNOWLEDGMENT

This work was supported by the ICT R&D program of MSIP/IITP. [R0112-14-1059, Development of Reducing Power Consumption Technology for Pay Broadcast Set-top Box]

REFERENCES

[1] T. Thang, Q. Ho, J. Kang, and A. Pham, “Adaptive streaming of audiovisual content using MPEG DASH,” IEEE Trans. Cons. Elec. Fukushima, vol. 58, no. 1, pp. 78-85, February 2012.

[2] S. Garacia, J. Cabera, and N. Garcia, “Quality-optimization algorithm based on stocastic dynamic programming for MPEG DASH video streaming,” IEEE Int. Conf. Cons. Elec. Madrid, pp. 574-575, January 2014.

[3] S. Wei and V. Swaminathan, “Low latency live video streaming over HTTP 2.0,” ACM Work. Net. Oper. Sys. Sup. Digi. Aud. Vid. San Jose, pp. 37-42, March 2014.