Segment Duration Based HTTP Adaptive Streaming Scheme for UHD Content

Heekwang Kim and Kwangsue Chung
Department of Electronics and Communications Engineering
Kwangwoon University
Seoul, Korea
Email: hkkim@cclab.kw.ac.kr, kchung@kw.ac.kr

***Abstract*—Due to the explosive growth of the mobile devices and development of the network technologies, HyperText Transfer Protocol (HTTP) adaptive streaming has become a new trend in video delivery to provide efficient multimedia streaming services. Another important development is the provision of high quality contents, such as Ultra High Definition (UHD) videos. UHD content supports longer Group Of Pictures (GOP) size, as well as higher spatial resolution and higher frame rates. Due to these characteristics, the UHD content is characterized by the longer segment duration and higher bandwidth requirements in HTTP adaptive streaming system. In an HTTP adaptive streaming system, the long segment duration, which is the characteristic of UHD content, degrades the responsiveness to network fluctuations. It increases the video stalling period and degrades the user Quality of Experience (QoE). In this paper, we propose the segment duration based HTTP Adaptive streaming scheme for UHD contents. In our scheme, we assign different segment duration based on each quality level and propose the rate adaptation scheme to improve the QoE for UHD content. Using the simulation, we prove that our scheme significantly reduces the playback discontinuity.**

Keywords-HTTP adaptive streaming; UHD (Ultra High Definition) content; QoE (Quality of Experience)

I. Introduction

Due to the development of the network technologies and the explosive growth of the mobile devices such as smart phone and tablet PC, there have been significantly increasing demands for multimedia streaming services. Video traffic is a growing fraction of Internet traffic. Because of video traffic growth, HyperText Transfer Protocol (HTTP) adaptive streaming is receiving attention to provide efficient multimedia streaming services. In the HTTP adaptive streaming, a video content is encoded at multiple bitrates and the encoded video content is segmented into small parts of fixed durations. When the HTTP adaptive streaming player starts video streaming, it requests a segmented video content by sending an HTTP GET message. Each video segment is downloaded over HTTP and TCP using the conventional HTTP web servers. Typical implementations of the HTTP adaptive streaming are Apple HTTP Live Streaming, Microsoft Smooth Streaming, and Adobe dynamic Streaming. Also, Dynamic Adaptive Streaming over HTTP (DASH) is defined in the standard [1].

Another important development is the provision of high quality contents, such as Ultra High Definition (UHD) videos attracting the attention of content providers. The UHD content format is defined in the Recommendation ITU-R BT.2020 and SMPTE ST 2036-1 [2]. Compared to High Definition (HD), UHD content supports longer Group Of Pictures (GOP) size, as well as higher spatial resolution and frame rates to reduce the data size. The UHD content is characterized by the long segment duration and higher bandwidth requirements in the HTTP adaptive streaming system.

In HTTP adaptive streaming system, a client can select the quality level of next segment based on the quality adaptation method to cope with bandwidth fluctuations. The long segment duration, which is the characteristic of UHD content, degrades the responsiveness to network fluctuations [3]. It increases the quality switching delay and video stalling period (also called as buffer freezing). These degrade the user Quality of Experience (QoE).

In this paper, we propose a segment duration based HTTP adaptive streaming scheme for UHD content. The proposed scheme assigns different segment duration according to each quality level and requests the extra segment of the selected quality by the rate adaptation algorithm to improve the responsiveness of UHD content. The proposed rate adaptation algorithm determines the number of extra segments and the quality at which each of these segments should be downloaded. The rest of the paper is organized as follows. The related works are presented in Section II. The proposed scheme is presented in Section III. The experimental results are provided in Section IV, and finally the concluding remarks are given in Section V.

II. Related Works

Figure 1 illustrates the architecture of HTTP adaptive streaming. An HTTP server consists of three parts, which include the *media segments*, *manifest file*, and *HTTP module*. A multimedia content is encoded with multiple qualities. Each encoded content is divided into small segments of the fixed duration which start with an I-frame and do not reference frames from the surrounding GOPs. The segments are individually addressable by unique Uniform Resource Identifier (URIs). The *manifest file* specifies the content characteristics (e.g., bitrate, codec information, framerate, segment duration, resolution and URI). The *HTTP module* at the server sends the media segments according to request from

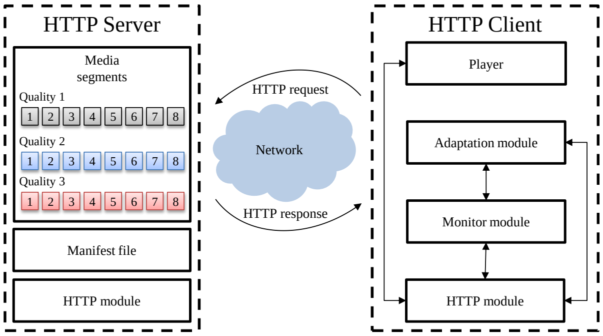

Figure 1. Architecture of HTTP adaptive streaming system.

the HTTP client. An HTTP client consists of four parts, which include the *player*, *adaptation module*, *monitor module*, and *HTTP module*. The *monitor module* estimates the throughput during the download of the segments. The *adaptation module* selects suitable bitrate for adapting the dynamic network conditions depending on the received manifest file and the measured condition, which is the estimated throughput or playback buffer occupancy. The *HTTP module* at the client sends the requests to the HTTP server according to the selected quality. When the HTTP adaptive streaming client starts video streaming, it requests the manifest file using HTTP GET message to the HTTP server. According to this request, the HTTP server sends the manifest file to the client. After receiving the manifest file, the client requests an appropriate segment using throughput measured in monitor module. Then, the HTTP server sends the requested segment. Finally, the *player* plays the received segments.

In order to improve the QoE and Quality of Service (QoS), many adaptation schemes are proposed for HTTP adaptive streaming. The adaptation schemes can be broadly divided into two categories. In the first category, the HTTP adaptive streaming switches the video quality according to the current state of the network, such as throughput [4][5]. Segment throughput is calculated as the ratio of the segment size to the time that it takes to download the segment [6]. The moving average of the throughput of previous segments is used to estimate the throughput [7]. In the second category, buffer occupancy is used in order to provide the seamless playback. The buffer is divided into predefined ranges and different decisions are taken to select the video rates when the buffer level stays in different ranges [8][9]. The method in [9] is more stable as compared to the method in [8] but it is late to react to the changes in the throughput as it waits for the playback buffer to reach a threshold before selecting a higher video rate.

III. Segment Duration Based HTTP Adaptive Streaming Scheme

In this section, we design the architecture of the proposed HTTP adaptive streaming system for UHD streaming services. Then, we also present a rate adaptation scheme for seamless playback of the UHD content.

A. Architecture of the proposed streaming system

Figure 2 shows the architecture of the proposed streaming system. In the proposed system, the server consists of the *media preparation* module, *content annotation* module and *segment scheduler* module.

- The *media preparation* module provides the tools for encoding and encapsulation so that the content can be presented and delivered efficiently to the client in the pre-defined format. The module encodes the video stream into multiple representations and divides the representation into small segments of different fixed duration according to quality level.
- The *manifest file* module provides metadata file or manifest file which contains the information about the characteristics of the stored multimedia content. The content information contains the bitrate, URI, and segment duration of each representation.
- The *segment scheduler* module is responsible for sending segments using multiple connections according to the request message.
- The *HTTP module* at the server sends the media segment based on requested message by client.

In the proposed system, the client consists of the *media player*, *buffer controller*, *manifest file parser*, *monitoring*, *request scheduler* and *quality adaptation* modules.

- The *player* module provides the tools which play and control the multimedia to the client.
- The *manifest file parser* module analyzes the manifest file received from the server.
- The *buffer controller* module stores the segments in the each buffer according to their connections. Further, this module finds the beginning time of the segments to be played next.
- The *monitor module* measures the throughput while downloading each segment and monitors the buffer occupancy while the client plays the received segments.
- The *adaptation module* selects suitable bitrate to request next based on the received manifest file and the measured context information.

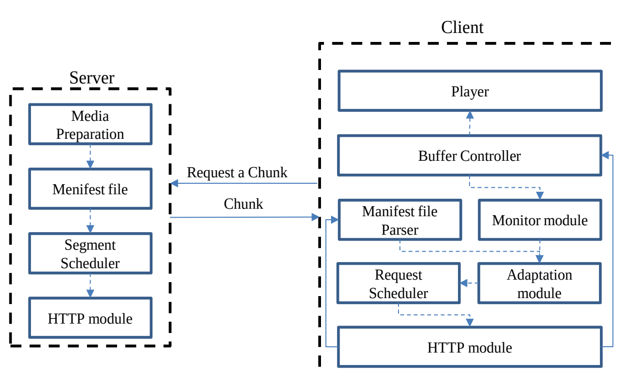

Figure 2. Architecture of the proposed streaming system.

- The *request scheduler* module determines the time to request segment based on the buffer occupancy of the client.
- The *HTTP module* at the client receives the media segment from the server.

B. Segment duration based rate adaptation scheme

We assign the segment duration differently according to each quality level and present the rate adaptation scheme that provides uninterrupted playback even when network bandwidth is decreased.

Figure 3 shows the assigned segment duration for seamless playback of the UHD content. The segment durations are organized hierarchically according to each quality level. The segment duration of the UHD content is 8 seconds due to its longer GOP size. To improve the responsiveness of the network conditions and minimize the wasted bandwidth, the segment duration of other quality level with the exception of UHD content is reduced by half compared to higher quality. For example, the segment durations of Full HD (FHD), HD and Standard Definition (SD) are 4, 2 and 1 seconds respectively. These assignments can quickly increase the quality from SD to UHD by improving the responsiveness of the network conditions.

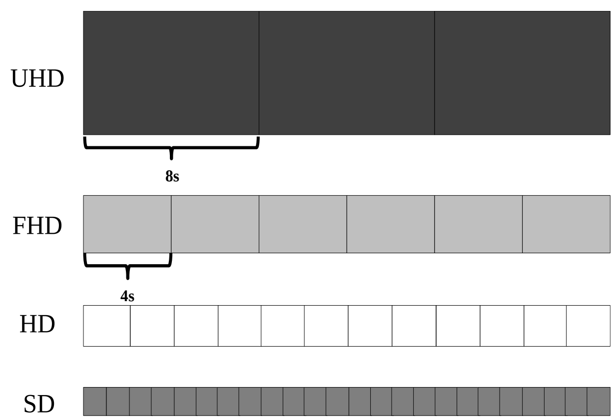

Figure 3. Segment duration assignment.

Figure 4 shows the proposed rate adaptation scheme for seamless playback of the UHD content. The top of the Figure 4 shows the proposed scheme when the bandwidth decreases while downloading the segment. The bottom of the Figure 4 shows the playback scenario of the client. If the requested segment is not downloaded within the segment duration, the client simultaneously requests an extra segment of a lower quality than previously requested using multiple connections while the buffered segments are being played. The extra segment is an additional segment with the same playback time as the segment currently being downloaded. If the lower quality segment gets downloaded before the previously requested segment, the client plays the extra segment while downloading the remaining data of previously requested segment. By playing the extra segment, the client can stream the video smoothly without buffer freezing. After the remaining data of the higher quality level is downloaded, the proposed system is looking for the time to switches to the higher quality segment.

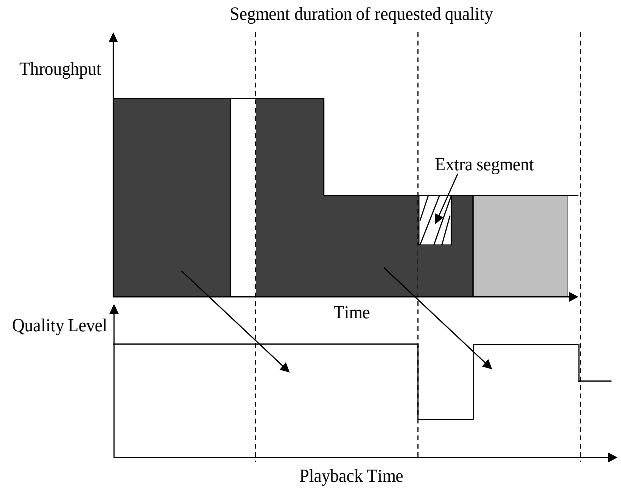

Figure 4. Segment duration based rate adaptation scheme.

To improve the user QoE, we present the rate adaptation algorithm for the extra segment. We first determine whether to request an extra segment every segment duration of each quality level. This decision is made by comparing the available buffer level and the required download time. The required download time is calculated as in (1).

$$t_{required} = \frac{S_{remain}}{th_{est}} \quad (1)$$

$t_{required}$ denotes the required download time, S_{remain} denotes the size of remaining data to download, and th_{est} denotes the estimated throughput. The size of remaining data to download is calculated as in (2).

$$S_{remain} = R_{origin} \times t_{origin} - S_{complete} \quad (2)$$

R_{origin} denotes the video rate of original segment, which is the higher quality level, t_{origin} denotes the segment duration of original segment, and $S_{complete}$ denotes the size of downloaded data. The proposed scheme has a condition for decision as in (3).

$$t_{required} > t_{buf} \quad (3)$$

t_{buf} denotes the buffering time as the available download time. If the time required to download the remaining data is larger than the buffering time, the client will experience playback interruptions. Therefore, the client needs to request the extra segment. The proposed scheme also determines the quality level of the extra segment. The quality of extra segment is calculated as in (4).

$$R_{extra} = \max R_k$$
$$\text{, where } \frac{S_{remain} + n(R_k \times t_k)}{th_{est}} < t_{buf} + n \times t_k, n < \frac{t_{origin}}{t_k} \quad (4)$$

R_{extra} denotes the video rate of extra segment, R_k denotes the video rate of k-th quality level, t_k denotes the segment duration of k-th quality level, and n denotes the number of extra segments. This equation means that the quality adaptation module selects the extra segment that is encoded with the maximum bitrate and avoids buffer underflow.

IV. PERFORMANCE EVALUATION

This section presents the simulation results for the proposed scheme. To evaluate the performance of the proposed scheme, we implement it in the Network Simulator-3 (ns-3). The simulation topology is shown in Figure 5. In this simulation, the bottleneck link is set at 20Mbps. The segment duration of the conventional HTTP adaptive streaming scheme is 8 seconds for all quality level. The server has 4 different pre-encoded video qualities (430Kbps, 1500Kbps, 2700Kbps, and 10000Kbps). We compare the conventional HTTP adaptive streaming scheme and the proposed scheme. The simulation is run for 400 seconds. In order to generate the cross traffic, we inject 10Mbps of Constant Bitrate (CBR) traffic between the server and client at 200 second.

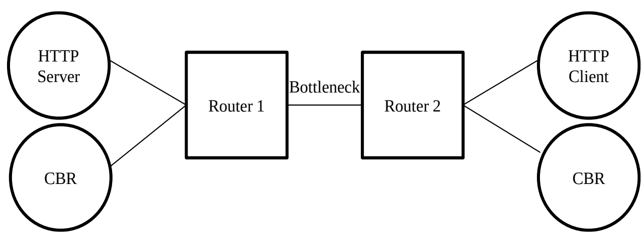

Figure 5. Simulation topology.

Figure 6 shows the playback video level and buffer occupancy of the conventional HTTP adaptive streaming scheme. The conventional HTTP adaptive streaming scheme experiences buffer underflow after generating the cross traffic because the client responds slowly to network changes due to long segment duration. On the other hand, Figure 7 shows the playback video level and buffer occupancy of the proposed scheme. In the proposed scheme, buffer underflow does not occur after generating the cross traffic because the proposed scheme requests an extra segment by predicting buffer underflow. The proposed scheme downloads an extra segment at 207 second and plays the extra segment while downloading the original segment. After receiving the original segment, the player switches to the quality of original segment. Simulation results show that the proposed scheme does not experience buffer underflow and provides a better QoE than conventional HTTP adaptive streaming scheme.

V. CONCLUSION

The UHD content is characterized by the long segment duration and higher bandwidth requirements in the HTTP adaptive streaming system because it supports not only higher frame rates but also longer GOP size. These characteristics degrade the user QoE due to lower responsiveness of the network changes. In this paper, we propose the segment duration based HTTP adaptive streaming scheme for UHD

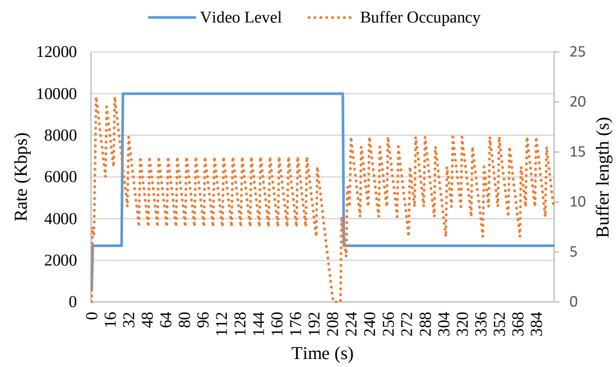

Figure 6. Playback video level and buffer occupancy of the conventional HTTP adaptive streaming scheme.

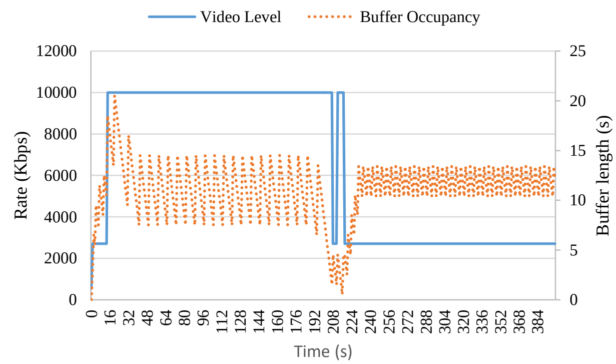

Figure 7. Playback video level and buffer occupancy of the proposed scheme.

content. The proposed scheme aims to provide the seamless playback for improving the QoE. To achieve this goal, we assign the different segment duration according to each quality level and request the extra segment from the server based on proposed rate adaptation scheme. Through the simulation results, the proposed scheme is proven to the seamless playback without buffer underflow. In the future work, we will analyze the proposed scheme in various network environments and implement the proposed system in real network.

ACKNOWLEDGMENT

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. 2017-0-00224, Development of generation, distribution and consumption technologies of dynamic media based on UHD broadcasting contents)

REFERENCES

[1] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hobfeld and P. Tran-Gia, "A Survey on Quality of Experience of HTTP

Adaptive Streaming," *IEEE Communications Survey and Tutorials*, vol. 17, no. 1, pp. 469–492, March 2015.

[2] ITU-R Recommendation BT.2020 "Parameter Values for UHD System for Production and International Programme Exchange," August 2012.

[3] D. M. Nguyen, L. B. Tran, H. T. Le, N. P. Ngc, and T. C. Thang, "An Evaluation of Segment Duration Effects in HTTP Adaptive Streamin over Mobile Networks," *in Proceeding of the 2nd National Foundation for Science and Technology Development Conference on Information and Computer Science (NICS '15)*, pp. 248-253, September 2015.

[4] C. Liu, I. Bouazizi, and M. Gabbouj, "Rate Adaptation for Adaptive HTTP Streaming," *in Proceeding of the ACM Multimedia Systems*, pp. 169-174, February 2011.

[5] S. Akhshabi, A. C. Begen, and C. Dovrolis, "An Experimental Evaluation of Rate-adaptation Algorithms in Adaptive Streaming over HTTP," *in Proceeding of ACM Conference on Multimedia Systems*, pp. 78-85, February 2011.

[6] T. C. Thang, Q. D. Ho, J. W. Kang, and A. T. Pham, "Adaptive Streaming of Audiovisual Content using MPEG DASH," *IEEE Transactions on Consumer Electronics*, vol. 58, no. 1, pp. 78.85, February 2012.

[7] T. Y. Huang, N. Handigol, B. Heller, N. McKeown, and R. Johari, "Confused, Timid, and Unstable: Picking a Video Streaming Rate is Hard," *in Proceeding of ACM Coference on Internet Measurement*, pp. 225-238, November 2012.

[8] K. Miller, E. Quacchio, G. Gennari, and A. Wolisz, "Adaptation Algorithm for Adaptive Streaming over HTTP," *in Proceeding of the IEEE Packet Video Workshop*, pp. 173-178, May 2012.

[9] D. Suh, I. Jang, and S. Pack, "QoE-Enhanced Adaptation Algorithm over DASH for Multimedia Streaming," in Proceeding of IEEE Conference on Information Networking, pp. 497-501, February 2014.