Signal Processing: Image Communication 47 (2016) 242–251

Contents lists available at ScienceDirect

Signal Processing: *Image Communication*

journal homepage: www.elsevier.com/locate/image

Eliminating bandwidth estimation from adaptive video streaming in wireless networks

Jaehyun Hwang [a,1], Junghwan Lee [b,*], Chuck Yoo [c]

[a] *Next Generation R&D Group, Samsung Electronics, Suwon, Gyeonggi-do, Republic of Korea*
[b] *Digital Technology & Biometry Division, National Forensic Service, Wonju, Gangwon-do, South Korea*
[c] *Department of Computer Science and Engineering, Korea University, Seoul, Republic of Korea*

ARTICLE INFO

Article history:
Received 21 June 2015
Received in revised form
26 June 2016
Accepted 27 June 2016
Available online 28 June 2016

Keywords:
Dynamic adaptive streaming over HTTP
Scalable video codec
Wireless quality adaptation algorithm
Video freeze

ABSTRACT

Dynamic Adaptive Streaming over HTTP (DASH) is the state-of-the-art technology for video streaming and has been widely deployed in both wired and wireless environments. However, mobile DASH users often suffer from video quality oscillation and even video freeze in wireless environments, which results in poor user experience. This is mainly because most quality adaptation algorithms in DASH rely highly on bandwidth estimation to adjust the video quality while wireless network bandwidth is unstable in nature and changes frequently according to wireless channel contention and condition. To provide stable performance, even during severe bandwidth fluctuation, this paper proposes the Wireless Quality Adaptation (WQUAD) algorithm, which eliminates bandwidth estimation from quality adaptation. Thanks to the Scalable Video Codec (SVC), the proposed scheme always prioritizes to lower layers over higher ones as long as the play-out buffer is not completely filled by the lower layers. As a result, the client always fills the buffer with the base layers first and then the upper enhancement layers sequentially. This horizontal adaptation is straightforward and does not require any bandwidth estimation. Through NS-2 simulations, we show that WQUAD achieves (i) stable performance, keeping the video quality level with respect to the long-term network bandwidth, (ii) effective video freeze prevention, and (iii) high video quality on average.

1. Introduction

Internet video has been an extremely popular application for mobile end-users and the video traffic generated by mobile devices has exponentially increased in recent years [1]. Meanwhile, HTTP Adaptive Streaming has become the new state-of-the-art video streaming technology based on the success of HyperText Transfer Protocol (HTTP) [2]. HTTP Adaptive Streaming can adjust video quality to the most appropriate level on a moment-by-moment basis according to the current network condition, for example, the available network bandwidth. Specifically, in this framework, a video is encoded at multiple bitrates and each encoding is divided into several video chunks. Since end-users can request each video chunk at a specific bitrate, it is possible to change the video quality smoothly during streaming. At this point, the quality adaptation algorithm is responsible for determining the video quality of the next chunk based on the bandwidth estimation and buffer occupancy level.

This adaptive video streaming mechanism has been developed for commercial products, for example, Microsoft's HTTP Adaptive Streaming [3], Apple's HTTP Live Streaming [4], and Adobe's HTTP Dynamic Streaming [5]. They employ different logic and formats [6], but their basic behavior is essentially the same. Eventually, Dynamic Adaptive Streaming over HTTP (DASH) [7] was successfully standardized by the Moving Picture Experts Group (MPEG) [8], and now is a company independent, open, and international standard. Generally, the main strength of DASH is that it is easy to traverse various Internet middle boxes such as Network Address Translation (NAT) and firewalls as standard HTTP is used. However, it has been observed that mobile users sometimes suffer from video quality oscillation and even video freeze in wireless environments. The main reasons are as follows:

- Most existing quality adaptation algorithms were designed to provide the best quality video even with network dynamics. Thus, it is reasonable for the algorithms to rely highly on the available bandwidth estimation for rate adaptation. However,

* Corresponding author.
E-mail address: ljh815@korea.kr (J. Lee).
[1] This work was done when the author was with Bell Labs, Alcatel- Lucent, Seoul, South Korea.

http://dx.doi.org/10.1016/j.image.2016.06.013
0923-5965/

this estimation-based algorithm may cause the video quality to oscillate if the network bandwidth continuously fluctuates, as it does in wireless networks.

- At the peak point of the bandwidth estimation, the adaptation algorithm may select a high quality level, which implies a large sized video chunk. At this point, if the mobile user suddenly experiences unavailability of the network bandwidth because of wireless channel contention, the client will struggle to download the *large* chunk, taking a long time without filling the buffer. Consequently, the video playback may be *frozen* when the buffer becomes empty.

In other words, the two phenomena (i.e., video quality oscillation and video freeze) are caused by frequent bandwidth changes and the previous algorithms were not designed to cope with the situation. Since it is commonly accepted that both problems can result in poor user experience, it is very important to provide stable performance in wireless networks while maximizing the overall video quality. To achieve this goal, we propose a new method, the Wireless Quality Adaptation (WQUAD) algorithm, which eliminates bandwidth estimation from quality adaptation. Basically, we claim that accurate bandwidth estimation is not that meaningful if network bandwidth is unstable; hence, we present an example where the existing bandwidth estimation-based algorithm suffers from video freeze even when the long-term network bandwidth is much higher than the lowest encoding bitrate. Therefore, our basic strategy in WQUAD is not to use the network bandwidth information. Instead, using the Scalable Video Codec (SVC) based DASH approach [9], we divide each video chunk into several layers so that the video quality level increases when the layers are stacked one by one. We then simply give priority to the lower layers over the higher ones as long as the buffer is not full of lower layers. In other words, the client fills its buffer with the lowest layers first and the next layers sequentially if more bandwidth is available. In this manner, the overall video quality in the buffer is maintained relative to the long-term network bandwidth without the need to estimate the bandwidth. This approach also helps avoid video freeze because it always tries to fill up the buffer with the lower layers. Indeed, this is the best approach to prevent video freeze. Currently, four major SVC encoders exist: *Joint Scalable Video Model* (JSVM) [10], *MainConcept* [11], *Vanguard Software Solutions* (VSS) [12], and *bSoft* [13]. We believe that our new algorithm can also be deployed in practice when the SVC-based DASH becomes popular based on the availability of SVC solutions.

We evaluated our WQUAD using NS-2 simulations and show that the proposed scheme achieves (i) stable performance, keeping the video quality level with respect to the long-term network bandwidth, (ii) effective video freeze prevention, and (iii) a high video quality level on average. The rest of this paper is organized as follows. Section 2 gives a brief review of SVC-based DASH and related work. Section 3 describes the motivation of the paper and explains the proposed algorithm in detail. In Section 4, we evaluate our algorithm with NS-2 simulations. Finally, we conclude the paper in Section 5.

2. Background and related work

This section presents the background on adaptive video streaming as well as some previous studies that focus on wireless environments.

2.1. AVC-DASH vs. SVC-DASH

In the DASH framework, H.264/AVC (Advanced Video Coding) is used for the encoding process. Each video is encoded at multiple bitrates (i.e., multiple quality levels), typically between eight and 20 quality levels ranging from, for example, 100 kbps to 6 Mbps [14,15]. Each encoding is then divided into several video chunks whose length is typically between one and 15 s [6,15]. Each video chunk starts with a key frame and consists of one or more closed groups of pictures (GOPs) so that it can be decoded as an independent video clip at the client. Next, the media server maintains a Media Presentation Description (MPD) file for each video, which includes available video streams, their encodings, and chunk durations and provides the file to the clients at the beginning of each video streaming, so that the clients can request the appropriate quality of video chunk one by one through HTTP requests. Whenever a requested chunk is downloaded, the client estimates the network bandwidth by computing the achieved throughput and runs a quality adaptation algorithm to determine the quality level of the next video chunk, mostly based on the network bandwidth and buffer occupancy level. Note that the media server does not need to keep any per-stream information because the main agent that runs the adaptation algorithm is the client, not the media server. This client-based adaptation mechanism, therefore, is scalable in terms of the number of concurrent video clients.

Recently, SVC-based DASH solutions [16,9,17] have been proposed because the H.264/AVC usually introduces a significant amount of content redundancy across different quality levels, requiring a large amount of storage. In the SVC encoding scheme, each video chunk is encoded into one base layer (BL) and several enhancement layers (ELs). The BL is mandatory for playback of the video chunk, whereas the ELs are optional. From a quality perspective, the lowest quality level in the AVC-DASH scheme is equivalent to the BL. The quality of each video chunk increases as the ELs are sequentially requested and stacked on top of the BL. In this manner, the storage size for video content is significantly reduced. According to [9], the storage overhead of AVC-DASH can be 200–300% that of SVC-DASH. On the other hand, the SVC encoding scheme introduces a coding penalty, the so-called SVC encoding overhead, which is estimated at 10% per enhancement layer [18,19] compared to AVC-DASH. To overcome this problem, we may consider introducing intermediate cache servers when SVC-DASH is used. Because the BLs should always be requested regardless of the chunk quality level, the probability that the BLs persist in the caches for a long period is generally quite high, which results in a high cache hit rate [20]. This effect would help reduce the overall traffic overhead.

Fig. 1 depicts the main difference between AVC-DASH and SVC-DASH; both schemes start by downloading an MPD file from the media server. AVC-DASH then requests an independent chunk file with the selected quality, whereas SVC-DASH requests the BL and several ELs sequentially according to the quality level. For example, in Fig. 1, when the client selects quality level 3 (Q3) for chunk 35, AVC-DASH requests only one file, "segment #35 at quality Q3." On the other hand, SVC-DASH repeats requests and downloads for the BL and two enhancement layers (EL1 and EL2), combined together at the client to present segment #35 at quality Q3. This could be an overhead for the SVC-DASH scheme because it may need more than one request for each chunk. However, one advantage is that SVC-DASH has a better flexibility; it can choose whether to move on to the next chunk or improve the quality of the current chunk at each layer [21,22].

2.2. DASH for wireless environments

Traditionally, many commercial Content Delivery Network (CDN) companies have used adaptive video streaming solutions mainly for wireline services (e.g., over-the-top video delivery [23]). However, today, wireless environments have attracted more