

A Bandwidth Estimation Scheme to Improve the QoE of HTTP Adaptive Streaming in the Multiple Client Environment

Sangwook Kim and Kwangsue Chung

Department of Electronics and Communications Engineering, Kwangwoon University
447-1, Wolgye-dong, Nowon-gu, Seoul, Korea
[e-mail: swkim@cclab.kw.ac.kr, kchung@kw.ac.kr]

*Corresponding author: Kwangsue Chung

*Received March 14, 2017; revised July 11, 2017; revised August 23, 2017; accepted September 15, 2017;
published January 31, 2018*

Abstract

Abstract-HTTP adaptive streaming (HAS) is a promising technology for delivering video content over the Internet. HAS-based video streaming solutions rely on bandwidth estimation to select the appropriate video bitrate. Video streaming solutions that consider network conditions provide users with seamless video playback. However, when multiple clients compete for a common bottleneck link, conventional bandwidth estimation schemes that consider only one client overestimate the network bandwidth due to the ON-OFF traffic pattern. The bandwidth overestimation can cause Quality of Experience (QoE) degradation, such as unnecessary changes in video quality, and unfairness of video quality. In this paper, we propose a client-side bandwidth estimation scheme to obtain a better QoE of HAS in the multiple-client environment. The proposed scheme differentiates the client buffer status according to the buffer occupancy, and then estimates the available network bandwidth based on the buffer status and segment throughput. We evaluate the performance of HAS implemented in the ns-3 network simulator. Simulation results show that compared with the conventional schemes, the proposed scheme can enhance the QoE.

Keywords: HTTP Adaptive Streaming, Bandwidth Estimation, Quality of Experience (QoE), Multi-client Environment, Fairness

1. Introduction

With the development of network technology and proliferation of various devices such as desktop, tablet PC, and smart phone, the number of users watching video over the Internet has increased explosively. According to Cisco's Visual Networking Index (VNI) forecast [1], video traffic is expected to account for 80 percent of all Internet traffic in 2019. With the explosion of video traffic, many researchers have studied video streaming solutions to improve the Quality of Experience (QoE) [2, 3]. In the past, Real-time Transport Protocol (RTP) based on the User Datagram Protocol (UDP) was used to deliver video over the Internet. However, UDP-based video streaming may cause network congestion because UDP does not support a congestion control mechanism.

Recently, video streaming based on Hypertext Transfer Protocol (HTTP) is the de facto standard for delivering video contents over the Internet. HTTP-based video streaming technologies are typically classified the HTTP progressive download, and HTTP adaptive streaming (HAS). Both technologies transfer video contents via the existing Internet infrastructure. However, in order to adjust the video quality, HAS takes into account the network condition. As a result, even when the network condition is not good, HAS can provide a relatively better user experience than HTTP progressive download. Thanks to this advantage of HAS, many video streaming solutions use it. Typical implementations of HAS are Microsoft Smooth Streaming [4], Apple HTTP Live Streaming [5], and Adobe Dynamic Streaming [6]. Also, Dynamic Adaptive Streaming over HTTP (DASH) is defined in the standard [7, 8].

Fig. 1 shows that in order to avoid waste of bandwidth and buffer overflow, HAS operates by distinguishing two different states of the status of the buffer. B_{cur} denotes the buffer occupancy of the client, B_{max} is the maximum buffer threshold, and N means the N -th segment. When the buffer occupancy of the client is lower than the maximum buffer threshold, the buffer status is in the buffering state; while if the buffer occupancy of the client is equal to or higher than the maximum buffer threshold, the buffer status is in the steady state. In the buffering state, the client receives segments continuously; on the other hand in the steady state, it receives segments periodically.

In the buffering state, a client can estimate the network bandwidth appropriately due to the Transmission Control Protocol (TCP) congestion control algorithm. TCP is based on Jacobson's AIMD principle [9]. AIMD-based TCP performs additive increase or multiplicative decrease of the window size, depending on the packet loss. Due to this TCP behavior, multiple clients can be close to the fair share bandwidth. However, in the steady state, when multiple clients compete for a common network bottleneck, conventional bandwidth estimation schemes of the HAS incorrectly estimate the available bandwidth as the existence of ON-OFF traffic pattern [10, 11]. In the steady state, during the ON period, a client downloads a segment; and during the OFF period, the client remains idle. Fig. 2 (a) shows that when the downloading cycle between two clients completely overlaps, the client properly estimates the available bandwidth. However, if the ON period partially overlaps as in Fig. 2 (b) or the ON period does not perfectly overlap as in Fig. 2 (c), a client overestimates the available bandwidth because the client estimates the bandwidth only during the ON period. In a real network, three different ON-OFF traffic patterns exist. The less the ON period overlaps among the clients, the more the bandwidth is overestimated. Bandwidth overestimation may result in unnecessary changes of video quality and unfairness of playback quality.

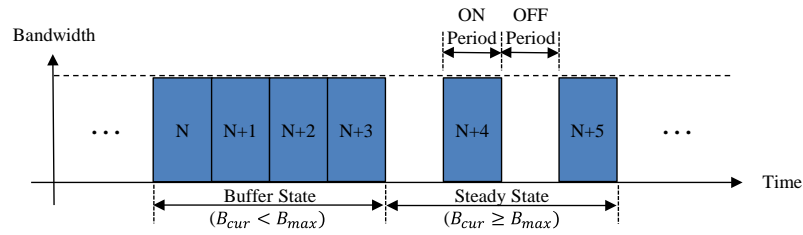
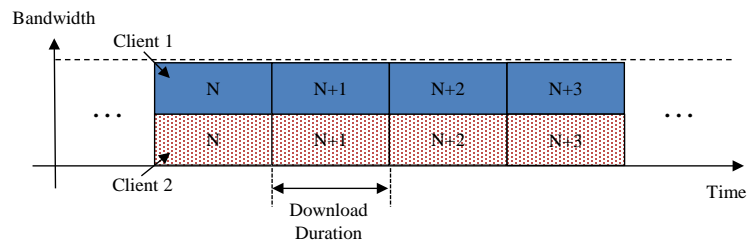
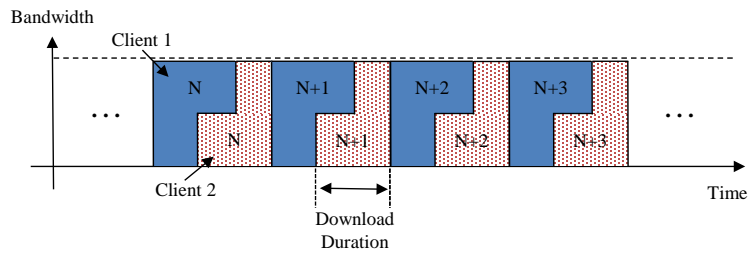


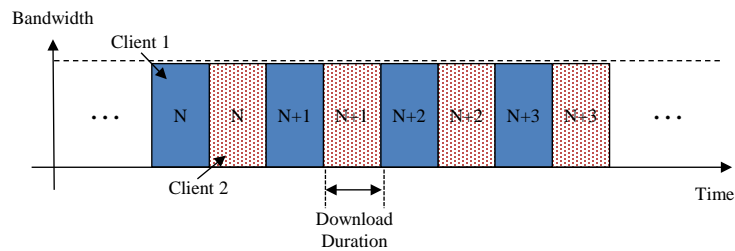
Fig. 1. The buffering state and the steady state



(a) ON period overlaps perfectly



(b) ON period partially overlaps



(c) ON period does not overlap

Fig. 2. Three case of ON-OFF traffic pattern

In this paper, we propose a bandwidth estimation scheme to improve the QoE of HAS in the multiple client environment. The proposed scheme differentiates the buffer status by observing the buffer occupancy of the client. Then, our scheme estimates the available bandwidth, based on the buffer status and segment throughput. The rest of the paper is organized as follows. Section 2 describes the related works in. Section 3 presents the proposed bandwidth estimation scheme for improving the QoE in the multiple client environment. In Section 4 conducts a performance evaluation of the proposed scheme. Section 5 concludes the paper.

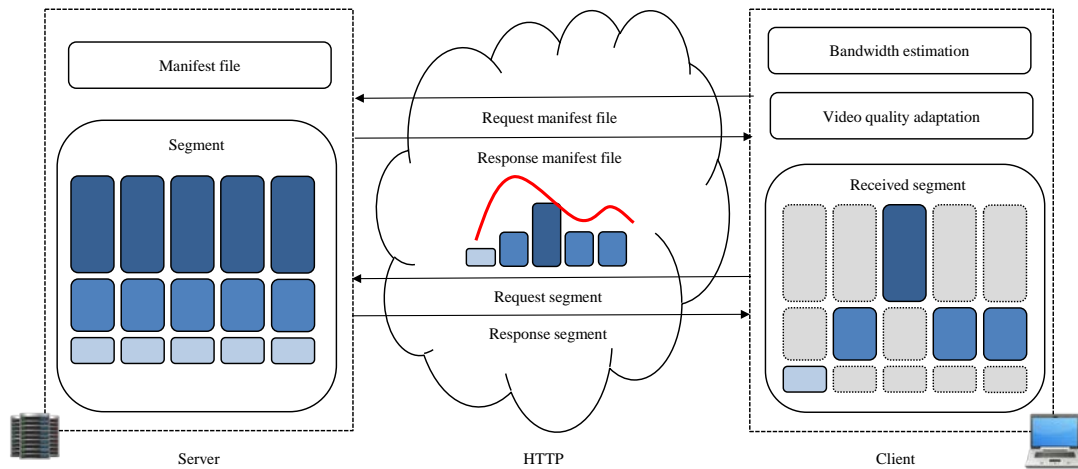


Fig. 3. The behavior of HAS

2. Related Work

HAS-based video streaming solutions are widely used, due to their advantages, such as network adaptability, scalability, and deployment simplicity. HAS-based video streaming solutions deploy a bandwidth estimation scheme to select the appropriate bitrate of a video segment. In other words, bandwidth estimation is a very important factor for improving the HAS performance. Many bandwidth estimation methods have been proposed to improve the QoE. In this section, we explain an overview of HAS, and then review the HAS-based streaming solutions that use bandwidth estimation to adapt video quality.

2.1 Overview of the HAS

The goal of HAS is to provide an enhanced user experience in a dynamic network environment [12, 13]. Although the existing HAS solutions are implemented in a variety of ways, the basic idea remains the same. Fig. 3 shows the behavior of HAS. The video file is divided into small segments, which are encoded at multiple bitrates and resolution. Segments generally have a constant length of between 2 seconds to 30 seconds. The encoded segments are stored on the media server. Before starting video streaming, the HAS client requests the manifest file that contains information about the audio and video stream. The client obtains the Uniform Resource Locator (URL) of video segments through the manifest file. Thereafter, the client estimates the available bandwidth during the download of a segment, and then determines the bitrate of the next video segment. The video quality adaptation algorithm selects the bitrate of the segments. This algorithm switches the video quality by taking into account the available bandwidth, CPU processing capacity, screen size, status of the buffer changes, etc.

2.2 HAS-based streaming solutions using bandwidth estimation

The main goal of the HTTP adaptive streaming is to select a video rate to optimize the viewing experience. Video rates and playback interruption events are important factors in improving the QoE [14][15]. The frequency of video rate changes has been found to annoy the viewer [16]. In addition, the majority of existing works focus on reducing the waiting time [17]. The HTTP clients select the video rates based on the estimated bandwidth; Therefore, bandwidth estimation has a significant effect on the performance of HAS in terms of QoE. In

order to improve the QoE, numerous studies related to bandwidth estimation scheme have been conducted. Reference [18] proposed a Rate Adaptation for Adaptive HTTP Streaming (RAHS) scheme to avoid the short-term throughput variations incurred by TCP congestion control, which results in a saw-tooth shaped instantaneous transmission rate. The RAHS scheme estimates the available bandwidth using the ratio of Media Segment Duration (MSD) and Segment Fetch Time (SFT), as in (1).

$$u = \frac{MSD}{SFT} \quad (1)$$

MSD denotes the period that a one segment can playback, SFT is the time elapsed between sending an HTTP GET request for a media segment and receiving the last bit of the requested media segment, and u is the ratio of the MSD and SFT . If u is higher than the maximum threshold, then the estimated bandwidth is higher than the selected video bitrate. RAHS then switches up to select the next higher video quality. On the other hand, if u is lower than the minimum threshold, then the estimated bandwidth is lower than the selected video bitrate. In this case, RAHS switches down to the next lower video quality. RAHS has the advantage of obtaining a high average bitrate of video quality, because it quickly adapts to network bandwidth variation. However, when multiple clients compete for the same bottleneck link, RAHS results in unnecessary changes of video quality, due to inaccurate bandwidth estimation. Reference [19] proposed the Adaptive Streaming of Audiovisual Content using MPEG DASH (ASAC) scheme to decrease the unnecessary changes of video quality. The ASAC scheme estimates bandwidth using segment throughput as in (2).

$$T_e[i] = (1 - \delta) \cdot T_e[i - 2] + \delta \cdot T_s[i - 1] \quad (2)$$

$T_e[i]$ denotes the i -th estimated segment throughput, $T_s[i]$ means the i -th measured segment throughput, and δ is a smoothing factor, which is adjusted according to the difference between the estimated segment throughput and the measured segment throughput. If δ is large, ASAC gives more weight to the $(i-1)$ -th measured segment throughput than $(i-2)$ -th estimated segment throughput. On the other hand, if δ is small, ASAC gives more weight to the $(i-2)$ -th estimated segment throughput than the $(i-1)$ -th measured segment throughput. This has the advantage of decreasing unnecessary changes of video quality in long-term network bandwidth variation. Reference [20] proposed the QoE-aware DASH system (QDASH) to improve the QoE of DASH service. The QDASH integrates available bandwidth measurement into the video data probes with a measurement proxy architecture. This scheme improves the QoE of DASH service by using stepwise decrease of video quality. However, it requires a measurement proxy to monitor the network condition. Akhshabi et al. [21] evaluate the performance of Microsoft Smooth Streaming and Netflix player using the running average of the throughput of several segments as the estimated throughput. The method performs well under persistent throughput variations. Ran et al. [22] use the median of the throughput of the last several segments to estimate the throughput of the next segment. Rahman et al. [23] show that the McGinley dynamic indicator offers a stable response to the throughput fluctuations, while maintaining a stable playback buffer. The moving average technique [23] is accurate in slow throughput variation, but reacts late to sudden variations in the throughput. The VLC media player [24] uses the averages of all previous throughputs as the estimated throughput to

download the next segment. The method responds slowly to the actual throughput variations, which increases the risk of buffer underflow.

When multiple clients compete for the bottleneck, authors in [10] showed that the clients incorrectly estimate the bandwidth. The bandwidth estimation methods proposed previously work well for single-client scenarios. In case of a multi-client scenario, the estimation methods do not estimate the bandwidth correctly which leads to issues of instability, unfairness and bandwidth underutilization [10]. We propose an estimation method that correctly estimates the bandwidth for both single- and multiple-client scenarios. The proposed scheme differentiates between the buffer status by observing the buffer occupancy of the client. Then, our scheme estimates the available bandwidth, based on the buffer status and segment throughput.

3. The Proposed Bandwidth Estimation Scheme

In this section, we describe the proposed scheme to overcome the problem of conventional video streaming schemes based on bandwidth estimation. Our scheme estimates the available bandwidth by considering the buffer status of the client in order to mitigate the bandwidth overestimation due to the ON-OFF traffic pattern. In the buffering state, the proposed scheme directly uses the segment throughput to estimate bandwidth. On the other hand, in the steady state, our scheme estimates the bandwidth by considering the number of clients that share a common bottleneck link. We first present the design goals, and then discuss the algorithm of the proposed bandwidth estimation scheme in detail.

3.1 Design goals

In the video streaming service, the Quality of Service (QoS), which is the ability to guarantee a certain level of performance, is expressed by network parameters such as packet loss, delay, and jitter. However, a good QoS does not necessarily guarantee the user a good viewing experience. Therefore, to provide a better viewing experience for users, we consider the QoE metrics, rather than QoS metrics. In HAS, the number of video quality changes and the playback bitrate are important factors that affect the QoE. In particular, when multiple clients are competing for a common bottleneck link, fairness is also a key factor that has an effect on the QoE. Thus, to improve the QoE in the multiple client environment, we focus on decreasing the number of video quality changes, and improving fairness among the multiple clients, while maintaining the playback bitrate at a similar level to the conventional schemes.

3.2 Bandwidth estimation scheme for improving the QoE in multiple client environment

In HAS, video segments are delivered by a sequence of consecutive HTTP request-response transactions. Traditionally, the client measures segment throughput for seamless video streaming whenever a segment arrives at the client. The HTTP client sends HTTP GET request to download the segment. The client requests the segment encoded with a particular data rate B . The length of the segment is denoted by τ . The HTTP server provides this information to the client. The client calculates the throughput by dividing the size of the segment by the time it takes to download the segment. The download time is computed from the instant when the HTTP request is sent to the instant when the last byte of the requested segment is received. The segment throughput is calculated as follows:

$$BW_{ins} = \frac{S_{size}}{D} \quad (3)$$

where S_{size} is the size of a segment which is equal to $B^* \tau$, and D denotes the time that it takes to download a segment. Since BW_{ins} reflects only a short period of network conditions, it can induce frequent changes of video quality. In order to remove the high frequency component of instantaneous bandwidth estimation, we use the running average filter to smooth out BW_{ins} . The smoothed bandwidth can be presented as:

$$BW_{avg}[i] = \begin{cases} \gamma \cdot BW_{avg}[i-1] + (1-\gamma) \cdot BW_{ins}, & i > 1 \\ BW_{ins}, & i = 1 \end{cases} \quad (4)$$

where i means the i -th segments, $BW_{avg}[i]$ denotes the smoothed bandwidth of the i -th segment, and γ is a coefficient that determines the degree of smoothing, which is bounded from 0 to 1. The closer γ is to 1, the more the estimated bandwidth reflects the smoothed bandwidth. On the other hand, the closer γ is to 0, the more the estimated bandwidth reflects the instantaneous bandwidth. We divide the proposed bandwidth estimation algorithm into two buffer statuses, based on the buffer occupancy. The status of the buffer can be determined as:

$$BufferStatus = \begin{cases} BufferingState, & B_{cur} < B_{max} - MSD \\ SteadyState, & B_{cur} \geq B_{max} - MSD \end{cases} \quad (5)$$

where B_{cur} denotes the buffer occupancy of the client, B_{max} means the maximum buffer threshold, and MSD is the media segment duration. If the buffer status is in the buffering state, the client can rely on the TCP congestion control algorithm to obtain an accurate estimated bandwidth. Therefore, we directly use the smoothed bandwidth. When the buffer status switches to the steady state, we initially use the same bandwidth estimated value as obtained in the buffering state, as given in (6).

$$BW_{est} = BW_{bs} = BW_{avg}[i] \quad (6)$$

BW_{est} denotes the estimated bandwidth for video quality adaptation and BW_{bs} is the latest estimated bandwidth in the buffering state. In the steady state, in order to avoid bandwidth overestimation due to ON-OFF traffic pattern, the proposed scheme uses the BW_{bs} to estimate the available bandwidth. In the case of multiple clients, it has been observed that the client overestimates the bandwidth as shown in **Fig. 4 (a)**. However, if the client uses BW_{bs} for bandwidth estimation when the actual bandwidth increases, then the client underutilizes the available bandwidth. Therefore, we should differentiate between the case when the client overestimates the bandwidth, and the case when there is an actual increase in the bandwidth as shown in **Fig. 4 (b)**. BW_{avg} denotes the smoothed bandwidth, K means the number of clients that are in the steady state, and BW_{bs} is the latest estimated bandwidth in the buffering state.

We differentiate between bandwidth overestimation and actual bandwidth increase by using BW_{bs} and the number of clients that are in the steady state. $K \cdot BW_{bs}$ is the maximum bandwidth that the client can get when K is the total number of clients. For example, the available bandwidth is 2Mbps, and we have 2 clients. If the bandwidth allocation is fair, both clients should get 1Mbps. During the Steady state, the client experiences ON-OFF periods. During

the OFF period, the client does not request the segment and stays idle. If two clients are competing for the bottleneck and one of the clients is in the OFF period, the other client will occupy the whole bandwidth and get the whole 2Mbps. This is higher than the fair share of the client i.e. 1Mbps. In this case, the client overestimates its fair share. Let's assume, the bandwidth has increased from 2Mbps to 4Mbps. The fair share of the clients will be 2Mbps. We use $K \cdot BW_{bs}$ to distinguish between bandwidth overestimation during the OFF period and an actual bandwidth increase. In this paper, we assume that whenever the buffer status of a client changes, all clients inform the server of their buffer status. Also, the server counts the number of clients in the steady state, and notifies it to all the clients. This assumption is reasonable, because we can implement the notification module, which uses a simple HTTP message to inform about the buffer status at the server-side and the client-side. As a result, in the case of bandwidth overestimation, the proposed scheme uses BW_{bs} to estimate the available bandwidth. On the other hand, in the case of actual bandwidth increase, our scheme uses BW_{bs} and α to estimate the available bandwidth. The estimated bandwidth to determine the video quality can be expressed as:

$$BW_{est} = \begin{cases} BW_{bs}, & BW_{avg}[i] \leq K \cdot BW_{bs} \\ BW_{bs} + \alpha, & BW_{avg}[i] > K \cdot BW_{bs} \end{cases} \quad (7)$$

where α is a coefficient to reflect the actual bandwidth increase. In the steady state, when the actual bandwidth increases, the proposed scheme estimates the available bandwidth by considering the smoothed bandwidth, the number of clients that are in the steady state, and BW_{bs} , as in (8).

$$\alpha = \{BW_{avg}[i] - (K \cdot BW_{bs})\} \cdot \frac{1}{K} \quad (8)$$

The more clients that share a common bottleneck in the steady state, the smaller α is. The proposed scheme employs a simple video quality adaptation algorithm that selects the highest bitrate of the video quality within the BW_{est} . The bitrate of the next segment can be expressed as follows:

$$Q = \max\{i : R_i < BW_{est}\} \quad (9)$$

where Q is the bitrate of the next segment, and R_i denotes the i -th video quality bitrate.

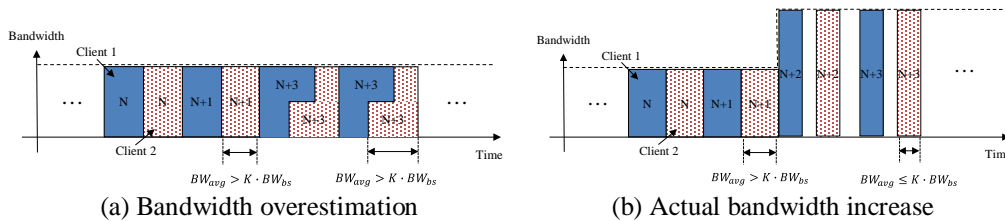


Fig. 4. The two cases of the increased bandwidth estimation

4. Performance Evaluation

In this section, the metrics that we employ to carry out the experimental evaluation are presented. We use simulation to compare the performance of the proposed scheme with conventional video streaming solutions based on bandwidth estimation. In this experiment, conventional HAS-based streaming solutions, RAHS [13] and ASAC [18], are used for the performance evaluation.

4.1 Evaluation metrics

We focus on three QoE metrics that are important to a user's viewing experience. These include average playback quality, the number of video quality changes, and the fairness of playback quality. The average playback quality means the average video bitrate that the client receives during the experimental period, which can be expressed as:

$$R_{avg} = \frac{1}{j} \sum_{i=1}^j R_i \quad (10)$$

where j is the number of total segments that the client receives during the experimental period. The number of video quality changes denotes the number of oscillations in the video quality level during the experimental period. Fairness means that multiple clients competing for a common bottleneck link should be able to obtain an equitable allocation of the bandwidth. Fairness among the multiple clients is calculated by the Jain's Fairness Index (JFI) [25], and it can be expressed as:

$$JFI = \frac{(\sum_{m=1}^M x_m)^2}{M \cdot \sum_{m=1}^M x_m^2} \quad (11)$$

where M denotes the number of clients sharing a common bottleneck link, and x_m means the average bitrate of the client m . The JFI is close to 1 when all clients utilize the bandwidth fairly.

4.2 Experimental setup

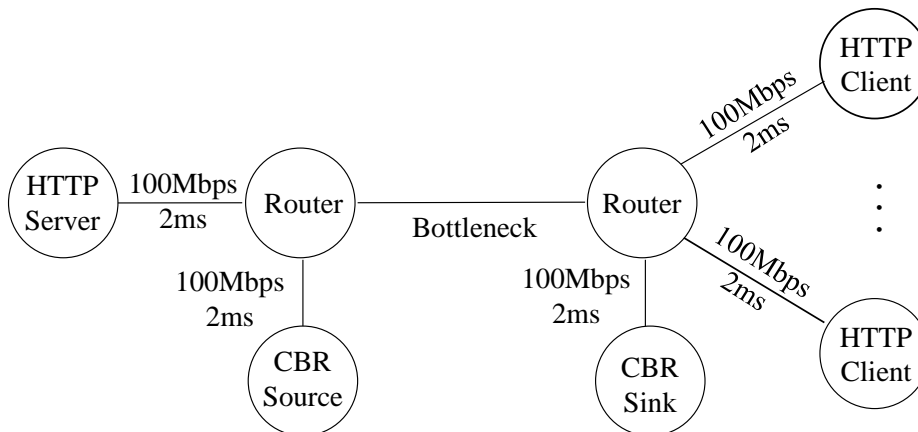
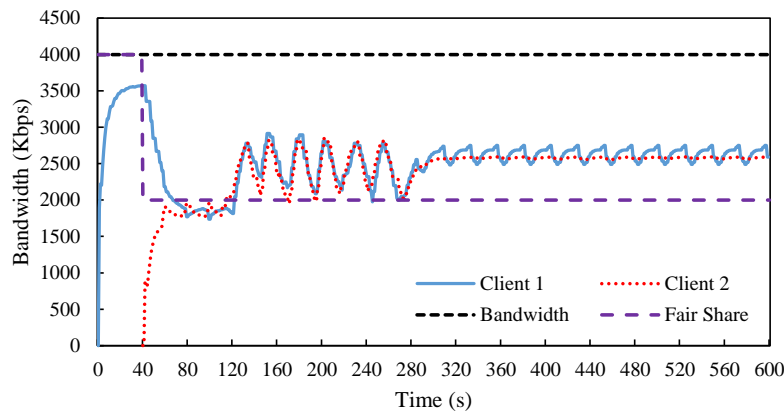


Fig. 5. The network topology

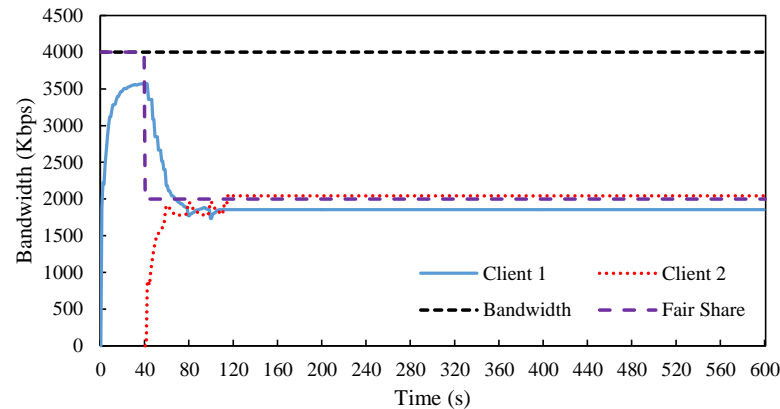
In order to evaluate the proposed scheme, we implemented it in Network Simulator-3 (NS-3) [26]. Commercial video streaming service such as Microsoft Smooth Streaming offers

segment duration of 2 seconds [27]; therefore, we used segment duration of 2 seconds. The server has 5 different pre-encoded video qualities (300, 700, 1000, 1500, 2500 Kbps). Fig. 5 shows the configuration of the network topology. Multiple clients compete for a common bottleneck link, and CBR traffic is injected to control the available bandwidth. The experiment duration is 600 seconds, and we consider two different scenarios. The first scenario fixes the bandwidth at 4Mbps. The other scenario varies the bandwidth between 2Mbps and 4Mbps. In this paper, we set γ to 0.8 throughout the experiment to smooth out the fluctuations in the network bandwidth.

4.3 Simulation results



(a) ASAC



(b) The proposed scheme

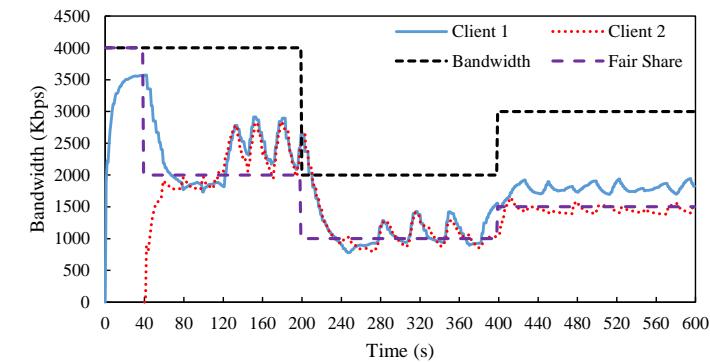
Fig. 6. Comparison of bandwidth estimation for the fixed bandwidth scenario

Fig. 6 compares the estimated bandwidth in the scenario that fixes the bandwidth at 4Mbps. In this scenario, client 1 and client 2 start video streaming at 0 second and 40 seconds respectively. Fig. 6 (a) shows that in the case of the bandwidth estimation of ASAC, both client 1 and client 2 can properly estimate the available bandwidth up to around 100 seconds. The buffer status changes from the buffering state to the steady state in client 1 at 109 seconds, and in client 2 at 119 seconds. After about 120 seconds, both clients overestimate bandwidth, due to the ON-OFF traffic pattern. On the other hand, in the proposed scheme, even though the buffer statuses of both clients change around 110 seconds, our scheme can accurately estimate

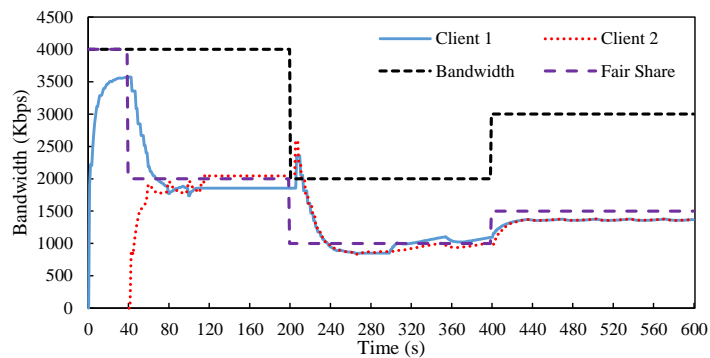
the available bandwidth compared to the ASAC, because it uses the same bandwidth as estimated during the buffering state, as given in Eq. (6).

Fig. 7 compares the estimated bandwidth in the scenario where the bandwidth is dynamic between 2Mbps and 4Mbps. **Fig. 7 (a)** shows that the buffer status of ASAC changes in response to the changes in available bandwidth. In the buffering state, the bandwidth estimation scheme of ASAC is fair, due to the TCP congestion control algorithm. However, the lesser the ON periods overlap between the clients, the more the ASAC overestimates the bandwidth in the steady state. On the other hand, compared to the ASAC, the proposed scheme can appropriately estimate the bandwidth, because our scheme distinguishes the buffering state and the steady state to estimate the available bandwidth.

Fig. 8 compares the video quality levels for the fixed bandwidth scenario. This scenario fixes the bandwidth at 4Mbps, but due to bandwidth overestimation, RAHS and ASAC lead to unnecessary changes in the video quality. In particular, RAHS results in severe unfairness of playback quality between the two clients, because RAHS estimates the available bandwidth by considering only one client. On the other hand, by avoiding the bandwidth estimation in the steady state, the proposed scheme can decrease the number of video quality changes. Also, our scheme can provide better fairness of playback quality than the conventional schemes, since it can estimate the network bandwidth close to the fair share bandwidth.



(a) ASAC



(b) The proposed scheme

Fig. 7. Comparison of bandwidth estimation for the dynamic bandwidth scenario

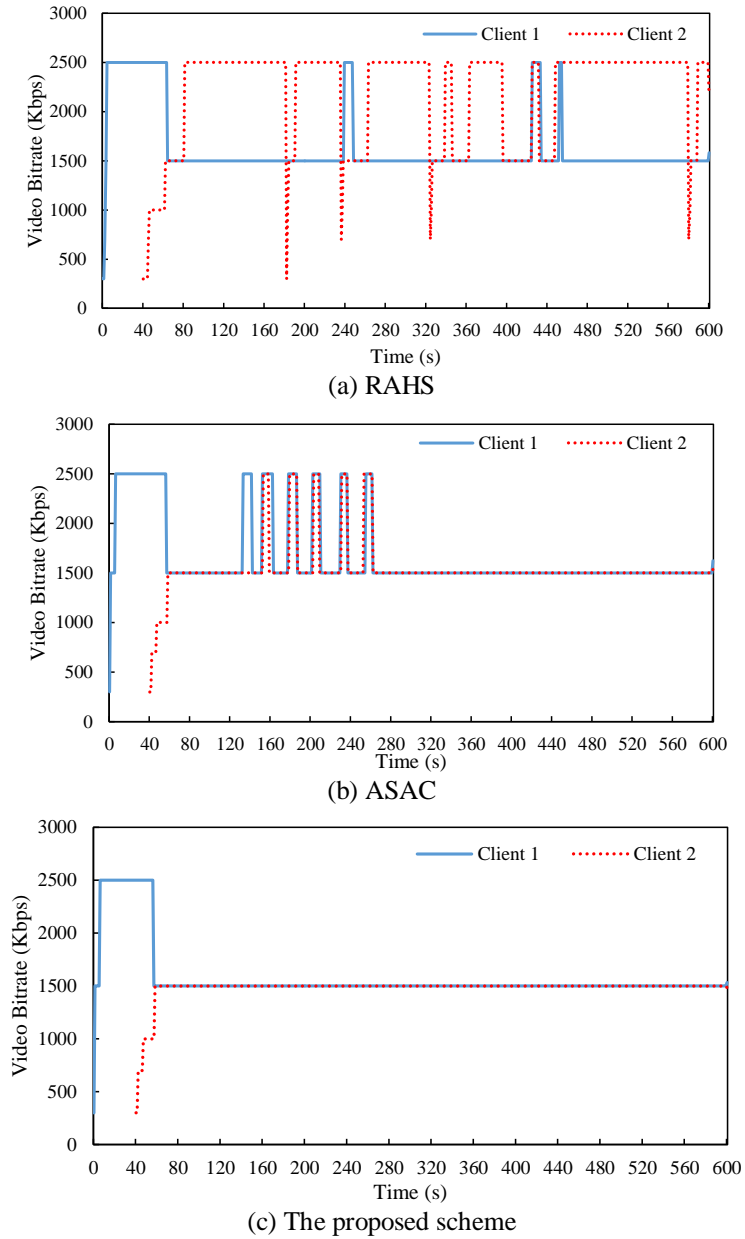
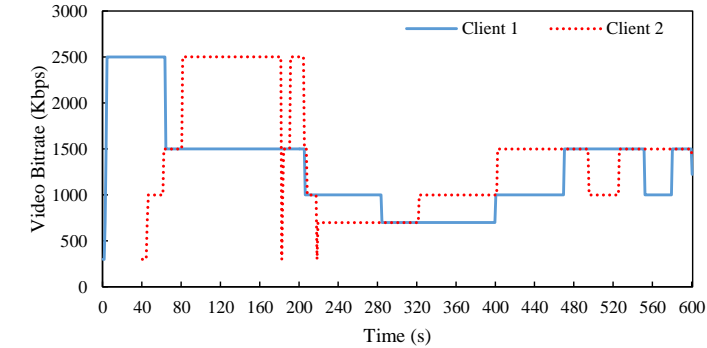
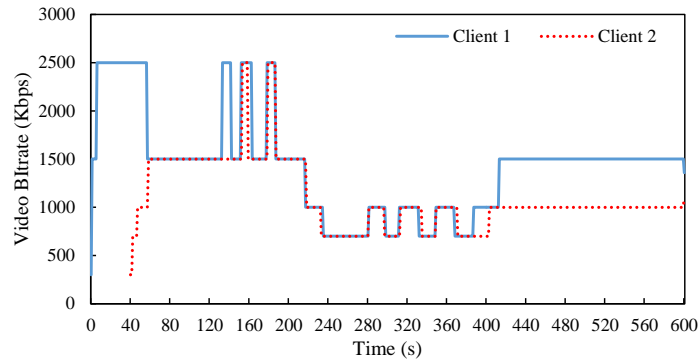


Fig. 8. Comparison of video quality levels for the fixed bandwidth scenario

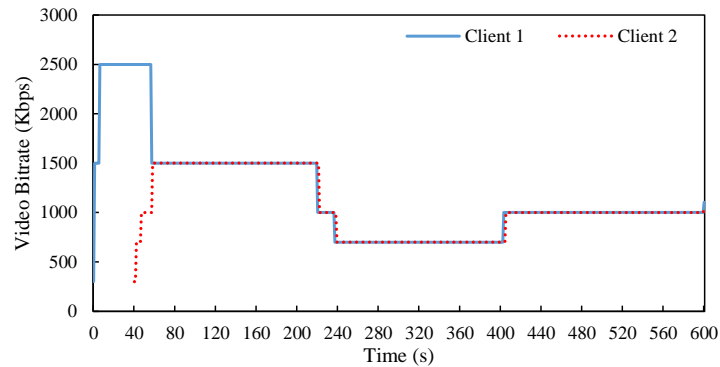
Fig. 9 compares the video quality level for the dynamic bandwidth scenario. The buffer occupancy of the client changes according to the bandwidth variation. When the network bandwidth is higher than the bitrate of the video, the client buffer is filled; whereas, when the network bandwidth is lower than the bitrate of the video, the client buffer is consumed. In this scenario, due to the ON-OFF traffic pattern in the steady state, RAHS and ASAC cause unnecessary changes. Furthermore, even though the buffer occupancy is less than the maximum buffer threshold, RAHS results in video quality changes. On the other hand, the proposed scheme can significantly reduce the number of video quality changes by distinguishing the status of the buffer in order to accurately estimate the network bandwidth. Our scheme also ensures fairness between client 1 and client 2.



(a) RAHS



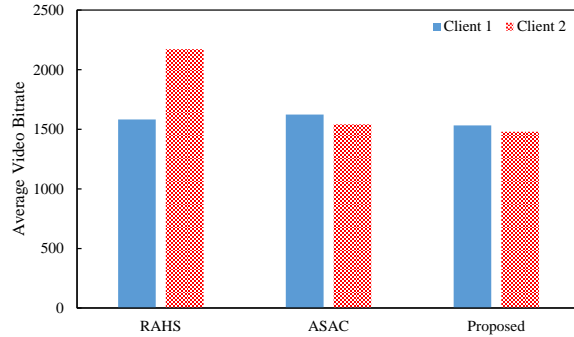
(b) ASAC



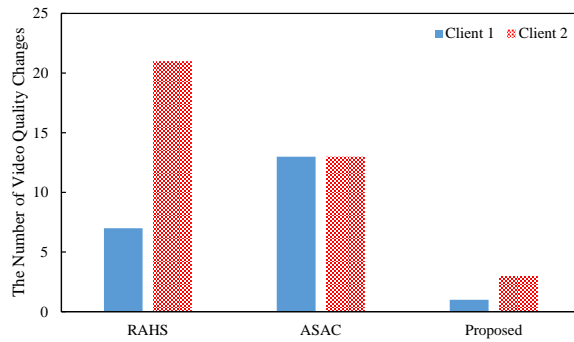
(c) The proposed scheme

Fig. 9. Comparison of video quality levels for the dynamic bandwidth scenario

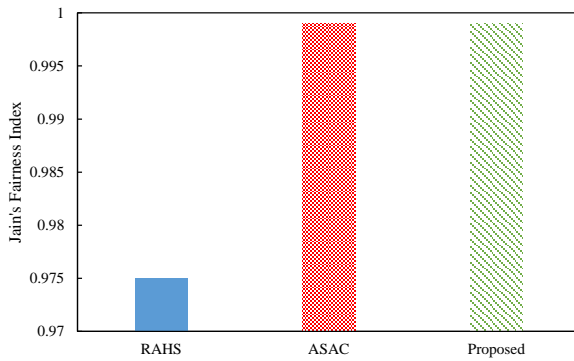
To compare the performance of the bandwidth estimation schemes, we measure the three QoE metrics, which are the average playback bitrate, the number of video quality changes, and JFI. The average playback bitrate is calculated using Eq. (11), and JFI is calculated using Eq. (12). **Fig. 10** compares the QoE for the fixed bandwidth scenario, while **Fig. 11** compares the QoE for the dynamic bandwidth scenario. In both scenarios, as compared to the conventional schemes, the proposed scheme achieves slightly less video bitrate. However, our scheme significantly reduces the number of video quality changes. In addition, the proposed scheme provides better fair sharing of the bottleneck with other competing clients than the conventional schemes. As a result, our scheme significantly improves QoE in terms of the number of video quality changes and fairness while maintains the average playback bitrate at a similar level to the conventional schemes in both scenarios.



(a) Average playback bitrate



(b) The number of video quality changes



(c) Fairness

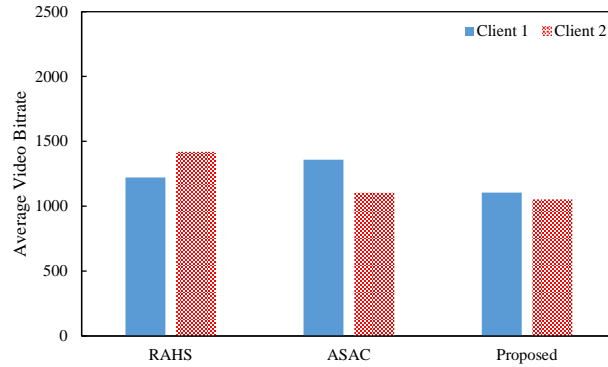
Fig. 10. Comparison of the QoE for the fixed bandwidth scenario

5. Conclusion

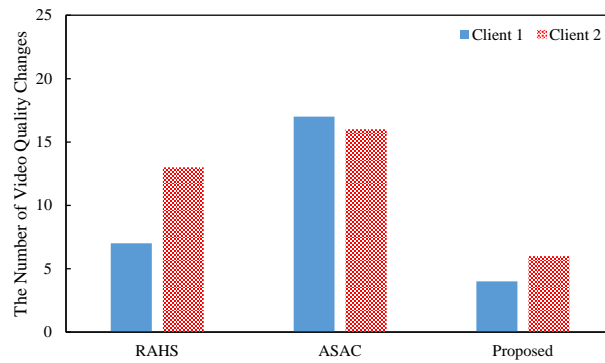
HAS has recently attracted attention recently. In HAS, the bandwidth estimation has a strong impact on the user-perceived QoE. The conventional HAS-based video streaming solutions that employ bandwidth estimation can provide seamless video playback to users. However, when multiple clients share a common bottleneck link, conventional video streaming solutions based on bandwidth estimation overestimate the network bandwidth. This overestimation of the bandwidth leads to unnecessary video quality changes, and unfairness of playback quality. To solve this problem, we propose a bandwidth estimation scheme to improve the QoE of

HAS in the multiple client environment. Our scheme aims to avoid bandwidth overestimation due to the ON-OFF traffic pattern, and to accurately estimate network bandwidth. We show that the proposed scheme provides a better QoE than the conventional schemes.

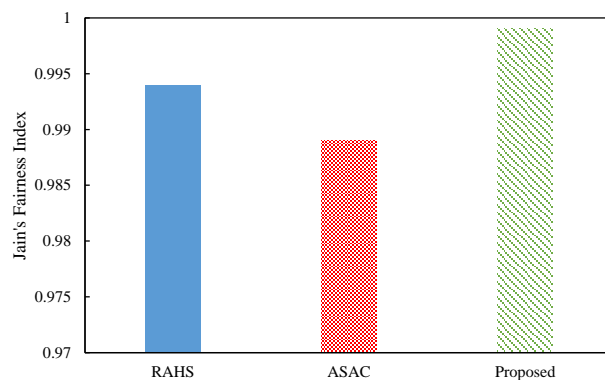
Future work will focus on extending our scheme to enhance the video quality adaptation algorithm. In addition, the parameter related to the bandwidth estimation scheme needs to be optimized, in order to further enhance the overall system performance.



(a) Average playback bitrate



(b) The number of video quality changes



(c) Fairness

Fig. 11. Comparison of the QoE for the dynamic bandwidth scenario

References

- [1] Cisco, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2014-2019," 2015. [Article \(CrossRef Link\)](#).
- [2] 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, Mar. 2015. [Article \(CrossRef Link\)](#).
- [3] T. Mangla, N. Theera-Ampornpant, M. Ammar, E. Zegura, and S. Bagchi, "Video Through a Crystal Ball: Effect of Bandwidth Prediction Quality on Adaptive Streaming in Mobile Environments," in *Proc. of the ACM 8th International Workshop on Mobile Video*, May 2016. [Article \(CrossRef Link\)](#).
- [4] Microsoft, Smooth Streaming, [Online]. Available: <http://www.iis.net/downloads/smooth-streaming/>
- [5] Apple, HTTP Live Streaming, [Online]. Available: <http://developer.apple.com/resources/http-streaming/>
- [6] Adobe, HTTP Dynamic Streaming, [Online]. Available: <http://www.adobe.com/products/httpdynamicstreaming/>
- [7] X. Yin, V. Sekar, and B. Sinopoli, "Toward a Principled Framework to Design Dynamic Adaptive Streaming Algorithm over HTTP," in *Proc. of the ACM workshop on Hot Topics in Networks*, pp. 1-9, Oct. 2014. [Article \(CrossRef Link\)](#).
- [8] T. Stockhammer, "Dynamic Adaptive Streaming over HTTP: Standards and Design Principles," in *Proc. of the ACM Multimedia Systems*, pp. 133-144, Feb. 2011. [Article \(CrossRef Link\)](#).
- [9] V. Jacobson, "Congestion avoidance and control," in *Proc. of the ACM Conference on Symposium proceedings on Communications architectures and protocols*, pp. 314-329, 1988. [Article \(CrossRef Link\)](#).
- [10] S. Akhshabi, L. Anantkrishnan, A. C. Begen, and C. Dovrolis, "What Happens When HTTP Adaptive Streaming Players Compete for Bandwidth?," in *Proc. of the ACM Workshop on Network and Operating Systems Support for Digital Audio and Video*, pp.9-14, Jun. 2012. [Article \(CrossRef Link\)](#).
- [11] Z. Li, X. Zhu, J. Gahm, R. Pan, H. Hu, A. C. Began, and D. Oran, "Probe and Adapt: Rate Adaptation for HTTP Video Streaming At Scale," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 4, pp. 719-733, Apr. 2014. [Article \(CrossRef Link\)](#).
- [12] A. Begen, T. Akgul, and M. Baugher, "Watching Video over the Web, Part 1: Streaming Protocols," *IEEE Internet Computing*, vol. 15, no. 2, pp. 54-63, Mar. 2011. [Article \(CrossRef Link\)](#).
- [13] T. Huang, N. Handigol, B. Heller, N. McKeown, and R. Johari, "Confused, Timid, and Unstable: Picking a Video Streaming Rate is Hard," in *Proc. of the ACM Conference on Internet Measurement Conference*, pp. 225-238, Nov. 2012. [Article \(CrossRef Link\)](#).
- [14] F. Dobrian, V. Sekar, A. Awan, I. Stoica, D. Joseph, A. Ganjam, J. Zhan, and H. Zhang, "Understanding the impact of video quality on user engagement," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 4, pp. 362-373, Aug. 2011. [Article \(CrossRef Link\)](#).
- [15] Y. Liu, S. Dey, D. Gillies, F. Ulupinar, and M. Luby, "User Experience Modeling for DASH Video," in *Proc. of the IEEE Packet Video Workshop*, pp. 1-8, Dec. 2013. [Article \(CrossRef Link\)](#).
- [16] P. Ni, R. Eg, A. Eichhorn, C. Griwodz, and P. Halvorsen, "Flicker effects in adaptive video streaming to handheld devices," in *Proc. of ACM International Conference on Multimedia*, pp. 463-472, Nov. 2011. [Article \(CrossRef Link\)](#).
- [17] L. Zhou, Z. Yang, Y. Wen, H. Wang, and M. Guizani, "Resource Allocation with Incomplete Information for QoE-driven Multimedia Communications," *IEEE Transactions on Wireless Communications*, vol. 12, no. 8, pp. 3733-3745, Aug. 2013. [Article \(CrossRef Link\)](#).
- [18] C. Liu, I. Bouazizi, and M. Gabbouj, "Rate Adaptation for Adaptive HTTP Streaming," in *Proc. of the ACM Multimedia Systems*, pp. 169-174, Feb. 2011. [Article \(CrossRef Link\)](#).

- [19] T. Thang, Q. Ho, J. Kang, and A. Pham, "Adaptive Streaming of Audiovisual Content Using MPEG DASH," *IEEE Transactions on Consumer Electronics*, vol. 58, no. 1, pp. 78-83, Feb. 2012. [Article \(CrossRef Link\)](#).
- [20] E. Chan, R. Mok, X. Luo, and R. Chang, "Qdash: A Qoe-aware DASH System," in *Proc. of the 3rd ACM Multimedia Systems Conference*, pp. 11-22, Feb. 2012. [Article \(CrossRef Link\)](#).
- [21] S. Akhshabi, A. C. Begen, and C. Dovrolis, "An experimental evaluation of rate-adaptation algorithms in adaptive streaming over HTTP," in *Proc. of ACM Conference on Multimedia System*, pp. 157-168, Feb. 2011. [Article \(CrossRef Link\)](#).
- [22] R. Dubin, O. Hadar, and A. Dvir, "The effect of client buffer and MBR consideration on DASH adaptation logic," in *Proc. of IEEE Wireless Communications and Networking Conference*, pp. 2178-2183, Apr. 2013. [Article \(CrossRef Link\)](#).
- [23] W. Rahman and K. Chung, "Buffer-based adaptive bitrate algorithm for streaming over HTTP," *KSII Transactions on Internet and Information Systems*, vol. 9, no. 11, pp. 4585-4622, Nov. 2015. [Article \(CrossRef Link\)](#).
- [24] VideoLAN. 2013. Vlc source code. [Online]. Available: <http://www.videolan.org/vlc/download-sources.html>.
- [25] R. Jain, D. Chiu, and W. Hawe, "A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems," Digital Equip. Corp., Littleton, MA, DEC Rep., DEC-TR-301, Sep. 1984. [Article \(CrossRef Link\)](#).
- [26] The Network Simulator 3 [Online]. Available: [Article \(CrossRef Link\)](#)
- [27] A. Zambelli. Microsoft Corporation. "IIS smooth streaming technical overview". [Online]. Available: [Article \(CrossRef Link\)](#)



Sangwook Kim received B. S. and M. S. degrees from Kwangwoon University, Seoul, Korea, from the Communications Engineering Department, in 2015 and 2017, respectively. His research interests include multimedia communication protocols, video streaming, and Internet QoS.



Kwangsue Chung received B.S. degree from Hanyang University, Seoul, Korea, M.S. degree from KAIST (Korea Advanced Institute of Science and Technology), Seoul, Korea, Ph.D. degree from University of Florida, Gainesville, Florida, USA, all from Electrical Engineering Department. Before joining the Kwangwoon University in 1993, he spent 10 years with ETRI (Electronics and Telecommunications Research Institute) as a research staff. He was also an adjunct professor of KAIST from 1991 to 1992 and a visiting scholar at the University of California, Irvine from 2003 to 2004. His research interests include communication protocols and networks, QoS mechanism, and video streaming.