Computer Science and Information Systems 18(4):1139–1157 https://doi.org/10.2298/CSIS200820021P

Buffer-Based Rate Adaptation Scheme for HTTP Video Streaming with Consistent Quality

Jiwoo Park, Minsu Kim, and Kwangsue Chung

Department of Electronics and Communications Engineering,
Kwangwoon University, Seoul, South Korea
{jwpark, mskim}@cclab.kw.ac.kr, kchung@kw.ac.kr

Abstract. Recently, HyperText Transfer Protocol (HTTP) based adaptive streaming (HAS) has been proposed as a solution for efficient use of network resources. HAS performs rate adaptation that adjusts the video quality according to the network conditions. The conventional approaches for rate adaptation involve accurately estimating the available bandwidth or exploiting the playback buffer in HAS clients rather than estimating the network bandwidth. In this paper, we present a playback buffer model for rate adaptation and propose a new buffer-based rate adaptation scheme. First, we model the playback buffer as a queueing system that stores video segments. The proposed scheme selects the next video bitrate that minimizes the difference between the current buffer occupancy and the expected value from the playback buffer model. The evaluation results indicated that the proposed scheme achieves higher video quality than conventional algorithms and can cope with various environments without the tuning of the configuration parameters.

Keywords: adaptive algorithms, queueing analysis, streaming media, transport protocols.

1. Introduction

Global Internet video traffic has been growing rapidly with the emergence of popular video streaming services such as YouTube, Netflix, and Amazon Prime. According to Cisco's Visual Networking Index, worldwide video traffic accounted for 75% of total Internet traffic in 2017 and is expected to reach 82% by 2022 [1]. To handle the increasing video traffic, many video service providers adopt adaptive bitrate streaming technology to provide the best possible streaming experience for users. Recently, HyperText Transfer Protocol (HTTP) based adaptive streaming (HAS) technology has attracted attention owing to the simplicity of its implementation and deployment [2]. In contrast to the existing real-time transport protocol (RTP) based streaming technology, which transmits video packets through User Datagram Protocol (UDP), HAS streams video over HTTP/Transmission Control Protocol (TCP), which is a traditional protocol stack used to deliver web messages. Video streaming technologies such as Microsoft's Smooth Streaming, Apple's HTTP Live Streaming, and Adobe's HTTP Dynamic Streaming rely on HTTP-based adaptive bitrate streaming [3-6]. In the HAS system, the video content is encoded at various bitrates, and the encoded video content is divided

into small video segments of a certain length and stored in the HTTP web server [7]. The HAS client sends an HTTP GET message to download the video segments. The transmitted video segment is stored in the playback buffer of the client, and when enough video segments are stored, the decoder consumes the first video segment and displays it on the screen.

Research on HAS is being actively conducted to improve the service quality and user experience. A general research topic is a methodology for improving the performance of the rate adaptation algorithm implemented in the HAS client and applying it to various environments [8]. In the conventional video streaming service, quality degradation is caused by the interruption of playback or the distortion of the image in a situation where the network bandwidth is insufficient. Because HAS dynamically adjusts the video bitrate, unnecessary changes in video quality can make users feel uncomfortable.

Recent studies have shown that requesting a segment in an ON-OFF pattern to maintain the buffer occupancy causes the HAS client to incorrectly measure the available bandwidth and repeat unnecessary quality changes in a multi-client environment [9]. To solve this problem, techniques for bandwidth measurement and playback buffer-based adaptation methods have been studied [10-12]. The existing approach is expected to improve the performance of HAS by accurately measuring the available bandwidth or setting a threshold for the buffer occupancy. However, most of the conventional approaches have been designed by targeting to a specific scenario. This leads to require the setting of configuration parameters such as weights and thresholds, degrading adaptability to the various scenarios. The conventional approaches are hard to achieve consistent quality for the media-consumption environments that the network bandwidth, videos watching by users, and number of users are changing over time.

In this paper, we propose a buffer-based rate adaptation scheme to achieve consistent quality for HAS. The main contributions of the proposed scheme are as follows.

- We analyze the relationship among the video bitrate, network bandwidth, and playback buffer occupancy of HAS.
- We present a playback buffer model for rate adaptation by considering the analyzed results for the relationship among the affecting factors to the performance of HAS.
- We then propose a novel rate adaptation scheme that controls the video bitrate by using the current buffer occupancy and the average buffer occupancy predicted by the playback buffer model.
- To compare the performance of the proposed scheme with the conventional approaches, we perform simulations in various network environments by using the ns-3 network simulator.

The reminder of the paper is organized as follows. Section 2 reviews related work on HTTP-based adaptive streaming. Section 3 presents the playback buffer model for HAS. Section 4 describes the proposed rate adaptation scheme and its buffer-based adaptation algorithm. Section 5 presents the results of the proposed scheme, and Section 6 concludes the paper.

2. Background and Related Work

In this section, we describe the basic operation of the HAS system and analyze the behavior of the HAS client. We also classify rate adaptation schemes according to adaptation factors such as the bandwidth, buffer occupancy, and video bitrate.

2.1. HAS System

As the demand for video streaming over the Internet increases, various technologies and standards have been proposed and developed. Recently, HTTP-based adaptive streaming has attracted attention owing to its efficient use of limited network resources and fast start-up time. Conventional streaming technology frequently uses the RTP over UDP, which does not perform error recovery for fast media delivery. By using HTTP, network address translation and firewall problems of existing streaming protocols can be easily solved. The HAS system also has the advantage of a low implementation cost because it can use the existing HTTP web servers and cache servers that are already installed globally.

In the HAS system, the HTTP web server stores video contents encoded with different resolutions, frame rates, and bitrates depending on the quality level. Each video is divided into segments of short length. The HAS client requests consecutive video segments while performing rate adaptation to adapt the video bitrates to the changing network environment. In general, rate adaptation algorithms use segment throughput to estimate the available bandwidth.

2.2. Behavior of HAS Client

At the beginning of the streaming, the HAS client quickly fills the playback buffer by continuously requesting video segments in the buffering state to prevent playback stalling. When the playback buffer is full, it periodically requests video segments in the steady state. Fig. 1 shows that if the video bitrate is lower than the network bandwidth, the HAS client has an ON-OFF pattern in the steady state. Owing to this ON-OFF pattern, the available bandwidth may be inaccurately measured in an environment where multiple HAS clients compete.

There are two typical problems caused by HAS clients having an ON-OFF pattern. The first problem is that the HAS client underestimates the available bandwidth because the TCP connection is idle during the OFF period [13]. When a TCP sender does not send or receive data for more than one retransmission timeout, the TCP congestion window is reduced to the initial value, and the TCP connection restarts slow-start after an idle period [14]. Unnecessary slow-start reduces the TCP throughput. For example, whenever an HAS client restarts slow-start while competing with greedy TCP flows, the throughput of the HAS client gradually decreases, and the client is unable to obtain the fair share of bandwidth.

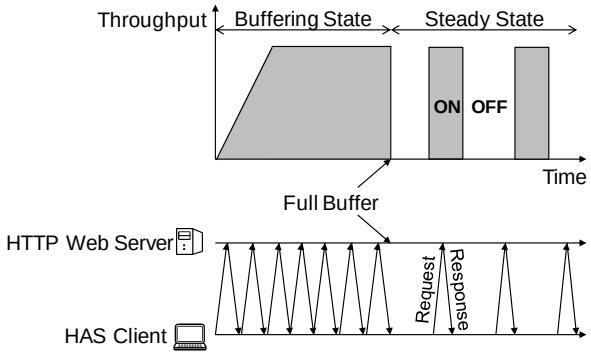

Fig. 1. Request pattern of the HAS client

Another problem is that HAS clients overestimate the available bandwidth when there are multiple HAS clients in the same network and they operate in an ON-OFF pattern [10]. Fig. 2 shows that the download duration varies depending on the overlap of ON periods when two HAS clients request video segments of the same size. Because the available bandwidth is estimated according to past segment throughputs, HAS clients may overestimate the bandwidth if the ON period is not overlapped. For example, if multiple HAS clients overestimate the available bandwidth and unnecessarily improve the video quality simultaneously, the network bandwidth becomes insufficient and network congestion occurs, resulting in poor video quality. In summary, the ON-OFF pattern is known to be a typical factor that degrades the quality of HAS services.

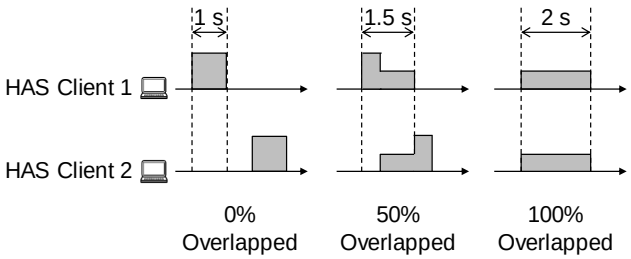

Fig. 2. Download duration of two competing HAS clients in the same network

2.3. Rate Adaptation Schemes

Although HAS is a relatively new application, its popularity has resulted in considerable research. In particular, the rate adaptation scheme is an interesting research topic because it automatically adjusts the video quality to provide video to users at the maximum possible quality. We begin by reviewing the rate adaptation scheme and then describe the key shortcomings of state-of-art solutions.

The most basic method of rate adaptation is to select the highest quality while ensuring a video bitrate lower than the available bandwidth. In general, the rate adaptation scheme is implemented in the client to reduce the load on the server.

- **Estimating**: Estimate the available network bandwidth by measuring the per-segment throughput from the previous segment request.
- **Smoothing**: Remove noise from estimates by applying an exponentially weighted moving average filter or a harmonic mean filter.
- **Quantizing**: Select the video quality using the smoothed version of the estimated bandwidth.
- **Scheduling**: Determine the next request time according to the playback buffer occupancy.

We can classify existing rate adaptation schemes into two main categories: bandwidth-based and buffer-based. Bandwidth-based rate adaptation controls the video bitrate according to the estimated available bandwidth. Early rate adaptation schemes adopted by commercial video providers belong to this category. Because the video quality mainly depends on the accuracy of the bandwidth estimation, measurement techniques considering the network type and traffic characteristics have been proposed. The dash.js video player provided by the DASH Industry Forum estimates the future throughput by using the average throughput of the last three segments to mitigate fluctuations in the bandwidth measurement [15]. PANDA predicts the available bandwidth in a manner similar to TCP congestion control and prevents the problem of ON-OFF patterns when multiple clients share bottleneck links [12]. PANDA updates the segment throughput in an additive increase and multiplicative decrease (AIMD) manner. Under complex network conditions, it is still challenging to accurately predict the network bandwidth.

Buffer-based rate adaptation selects the video bitrate according to the occupancy of the playback buffer implemented in HAS clients. In the buffer-based approach, the video quality is proportional to the buffer occupancy. A few studies have addressed the buffer-based approach to model the playback buffer [16-20]. BBA performs rate adaptation using a function that linearly maps the current buffer occupancy to the video bitrate [21]. It also divides the playback buffer into three sections, and its performance is determined by the length of each section. In [22], the authors modeled the playback buffer as an M/M/1 queue to characterize buffer starvations.

Because rate adaptation schemes are designed to improve the performance of HAS in a specific scenario, they make direct or indirect assumptions regarding the target environment. Most of them also require the setting of configuration parameters, such as weights and thresholds. These are often set arbitrarily through experiments. While fixed parameters may be adequate in certain scenarios, they cannot achieve consistent quality for all scenarios. Therefore, we must identify the factors affecting the video quality and consider these factors for rate adaptation. Clearly, the bandwidth and buffer are the main factors in rate adaptation. However, the bandwidth and buffer are treated separately, and the relationship between them is not well-considered in conventional approaches. In this paper, we present a playback buffer model for HAS clients and analyze the relationship among the bandwidth, buffer occupancy, and video bitrate using queueing theory.

3. Playback Buffer Model for HAS

In this section, we formalize the playback buffer for HAS clients. Before presenting the playback buffer model, we first define the symbols and terms used in the paper, as shown in Table 1.

Table 1. Notation used in this paper

Notation	Definition
r_n	Video bitrate of the n^{th} segment
R_m	Video bitrate of the m^{th} quality level
x	Segment throughput
τ	Segment duration
b	Buffer occupancy
b_{max}	Buffer capacity
k	Number of segments in the buffer
λ	Segment arrival rate
μ	Segment service rate
ρ	Traffic intensity of the buffer
c	Coefficient of variation
N	Average number of segments in the buffer
K	Maximum number of segments in the buffer
W	Average waiting time of buffer

In this paper, the buffer occupancy of HAS clients is expressed in units of time. As shown in Fig. 3, the buffer occupancy is reduced by the time the video is played, and when the segment download is completed, the segment duration is added to the buffer occupancy. We can model the playback buffer as a queue that stores and processes video segments, as shown in Fig. 4.

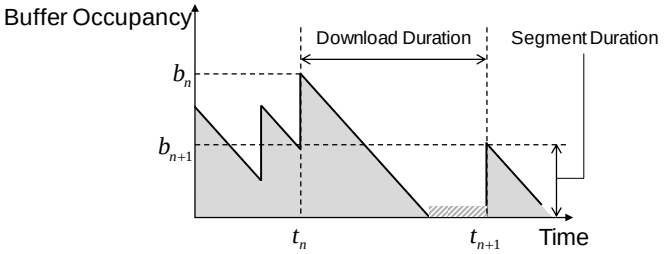

Fig. 3. Buffer occupancy of the HAS client

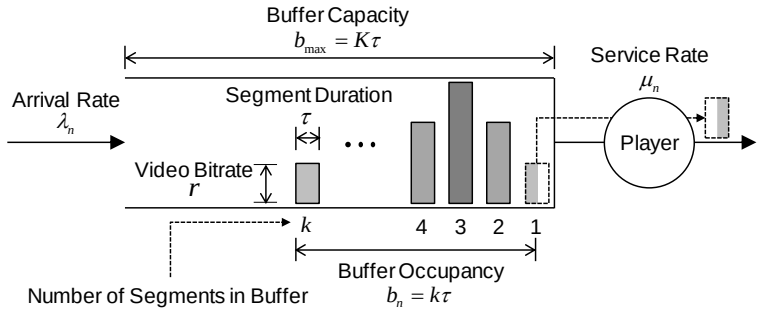

Fig. 4. Queuing model of the playback buffer

In the playback buffer model, the arrival rate is the number of video segments arriving per unit time. The nth arrival rate can be calculated using the estimated network throughput and the size of the nth video segment, as follows.

$$\lambda_n = \frac{x_n}{r_n \cdot \tau} \tag{1}$$

Unless playback is paused, the HAS client consumes one video segment during each segment duration in the steady state. In this case, the service time is equal to the segment duration, and the service rate can be expressed as follows.

$$\mu_n = \frac{1}{\tau} \tag{2}$$

The buffer occupancy is updated according to the following equation.

$$b_n = \max\left(0, b_{n-1} - \left(t_n - t_{n-1}\right)\right) + \tau \tag{3}$$

Here, t_n is the time at which the nth segment download is completed. Assuming that we have a continuous analogue of b_n, the following relationship is satisfied.

$$\frac{db(t)}{dt} = \lambda(t) - \mu(t) \tag{4}$$

Equation (4) shows that the buffer occupancy of HAS clients can be mathematically modeled as a non-linear differential equation.

Because video segments are transmitted over the network and consumed at a constant rate, we suppose that the arrival rate follows a certain probability distribution and that the service rate is fixed. Thus, we model the playback buffer as a G/D/1/K queue, where G represents interarrival times, which have a general distribution; D represents service times, which are deterministic; and K represents the queue size. Because the analytic solution of the G/D/1/K queuing model is unknown and is very difficult to obtain, we use an approximation to predict the average buffer occupancy. In the playback buffer model, the buffer occupancy is equal to the time to wait in the playback buffer until the most recently received segment is decoded.

Kingman's formula is the most widely used approximation for the mean waiting time in a G/G/1 queue [23].

$$E\left[W_{GG1}\right] \approx \left(\frac{c_a^2 + c_s^2}{2}\right)\left(\frac{\rho}{1-\rho}\right)\frac{1}{\mu} \tag{5}$$

Here, $\rho = \lambda/\mu$ is the traffic intensity, which represents how busy a queueing system is. Because the second term in (5) represents the average number of elements in an infinite queue, it must be modified for a finite queue. If the interarrival time and the service time follow the exponential distribution, their coefficients of variation (CVs) are equal to 1, and (5) becomes an equation for M/M/1 queues. An M/M/1/K queue is the finite version of the M/M/1 queue, and its mean waiting time is calculated via summation instead of an infinite series. The mean waiting time for an M/M/1 queue is twice that for an M/D/1 queue. The mean waiting time of M/M/1, M/M/1/K, and M/D/1 queues can be expressed as following equation when $0 < \rho < 1$.

$$\begin{aligned}\lim_{K\to\infty} W_{MM1K} &= \lim_{K\to\infty}\left(\frac{\rho}{1-\rho}\right)\left(\frac{1-(K+1)\rho^K + K\rho^{K+1}}{1-\rho^{K+1}}\right) \\ &= \frac{\rho}{1-\rho} = W_{MM1} = 2\cdot W_{MD1}\end{aligned} \tag{6}$$

According to the relationship among M/M/1, M/M/1/K, and M/D/1 queues, we can predict the mean waiting time of a G/D/1/K queue as follows.

$$E\left[W_{GD1K}\right] \approx \frac{c_a^2}{2}\left(\frac{\rho}{1-\rho}\right)\left(\frac{1-(K+1)\rho^K + K\rho^{K+1}}{1-\rho^{K+1}}\right)\frac{1}{\mu} \tag{7}$$

The CV of the service time c_s is removed because the standard deviation of the service time is equal to 0 in our model. When the playback buffer is in the steady state, where ρ converges to 1, (7) converges to the following equation.

$$\lim_{\rho\to 1} E\left[W_{GD1K}\right] = \frac{c_a^2}{2}\cdot\frac{K}{2}\cdot\frac{1}{\mu} = \frac{c_a^2 \tau K}{4} \tag{8}$$

4. Proposed Buffer-Based Rate Adaptation

This section introduces the proposed rate adaptation scheme, which measures the buffer information rather than the network bandwidth for rate adaptation. Fig. 5 shows a block diagram of the proposed scheme in the HAS system.

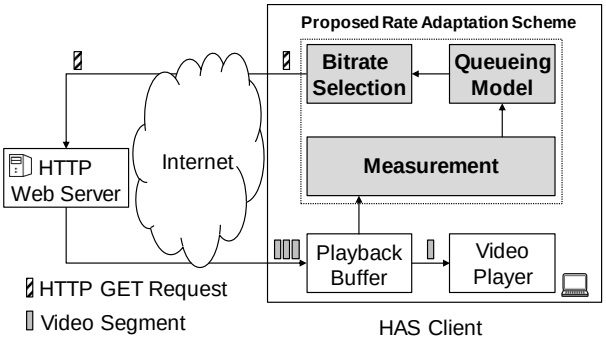

Fig. 5. Block diagram of the proposed scheme

4.1. Measurement

In this step, the proposed scheme measures the available bandwidth and the interarrival time of segments for the playback buffer model. Measurement of the available bandwidth in the HAS client is not accurate, because it is performed in the application layer and involves measurement error. The approximate available bandwidth is predicted via the smoothing of bandwidth samples.

There are many ways to take an average, such as the arithmetic mean, harmonic mean, and moving average. The proposed scheme uses all samples and updates the previous average using the current sample, as follows.

$$A_n = \frac{1}{n}\sum_{i=1}^{n} x_i = A_{n-1} + \frac{1}{n}\left(x_n - A_{n-1}\right) \tag{9}$$

The arithmetic mean of the samples is calculated using (9). The network bandwidth is expressed in terms of the bitrate, i.e. the number of bits transferred per unit of time. When calculating the average of rates, such as speed, bitrate, and bandwidth, the harmonic mean is a more appropriate method than the arithmetic mean. The proposed scheme estimates the network bandwidth using the following equation.

$$H_n = \left(\frac{1}{n}\sum_{i=1}^{n}\frac{1}{x_i}\right)^{-1} = \left(\frac{1}{H_n} + \frac{1}{n}\left(\frac{1}{x_n} - \frac{1}{H_{n-1}}\right)\right)^{-1} \tag{10}$$

We also calculate the CV of the interarrival time, which is a variable in the playback buffer model. The proposed scheme records the arrival time of each segment and calculates the interarrival time. The average of the interarrival times is calculated using (9). Because only a squared CV is needed, we calculate the variance of the interarrival time, as follows.

$$\sigma_n^2 = \left(1 - \frac{1}{n}\right)\left(\sigma_{n-1}^2 + \frac{1}{n}\left(x_n - A_{n-1}\right)^2\right) \tag{11}$$

In the proposed scheme, the CV of the interarrival time represents the network variability, which indicates how often the network changes. However, because there are

insufficient data at the beginning of the streaming, the CV does not contain meaningful information. It may be a smaller than predicted using the proposed playback buffer model. To prevent this, we calculate the CV using (9) and (11) and set the minimum value as follows.

$$c_n^2 = \max\left(1, \left(\frac{\sigma}{A_n}\right)^2\right) \tag{12}$$

4.2. Updating Queuing Model

The proposed scheme estimates the average buffer occupancy through the queuing model to perform buffer-based rate adaptation. We define the expected average buffer occupancy B as a function of the estimated bandwidth and the CV of the interarrival time.

$$B_n = \frac{c_n^2 \tau}{2}\left(\frac{H_n}{r - H_n}\right)\left(\frac{r^{K+1} - (K+1) r H_n^K + K H_n^{K+1}}{r^{K+1} - H_n^{K+1}}\right) \tag{13}$$

Equation (13) gives the relationship among the available bandwidth, buffer occupancy, and video bitrate. We observe that the buffer capacity, segment duration, and CV of the interarrival time affect the buffer occupancy and represent the variability of the device, video, and network, respectively. In this model, the buffer capacity and segment duration are fixed before performing rate adaptation. The proposed scheme updates the queuing model using the data measured in the previous step and thus is able to take into account the variability of the surrounding environment.

4.3. Bitrate Selection

When choosing the video bitrate according to the buffer occupancy, it is necessary to prevent buffer underflow and overflow, which adversely affect the performance, and simultaneously improve the average video quality. Buffer underflow and overflow can be resolved by keeping the buffer occupancy constant. By selecting the video bitrate in proportion to the buffer occupancy, the video quality can be improved as the playback buffer is filled.

If we derive a function such as $f(H_n, B_n) = r$ from (13), we can easily select the appropriate video bitrate. However, it is impossible to obtain the inverse of a multivariate nonlinear function in an analytical manner. Therefore, the proposed scheme follows a heuristic method to determine the video bitrate according to the buffer occupancy. If the video bitrate can be selected to set the buffer occupancy to the target occupancy, the playback buffer can remain stable. The proposed scheme selects the next video bitrate that minimizes the difference between the buffer occupancy and the expected value from (13), as follows.

$$r_{n+1} = \arg\min_R \left| (B_{\max} - B_n) - b_n \right| \tag{14}$$

Here, $B_{\max}$ is the maximum predictable value of the average buffer occupancy and satisfies the following equation.

$$B_{\max} = \lim_{x\to\infty} B_n = 2 \cdot \lim_{x\to r} B_n \tag{15}$$

Fig. 6 illustrates the proposed bitrate selection in a two-dimensional space. Suppose that a video is encoded at five bitrates $\{R_1, R_2, R_3, R_4, R_5\}$. Then, we can draw five points on the $B_{\max} - B$ curve. A larger index of R represents a higher video bitrate. The position of each point is determined by the ratio of the bandwidth to the encoded bitrate. For example, as the bandwidth increases, the points move to the right along the curve. In accordance with (13), the buffer capacity determines the shape of the curve, and the slope increases with the capacity. The CV of the interarrival time and the segment duration scale the curve vertically. The proposed scheme finds the closest point to the buffer occupancy line. Thus, video streaming starts with the lowest bitrate, but a higher bitrate is selected as the buffer occupancy increases from 0. If the selected bitrate satisfies $0 < x/r < 1$, the buffer occupancy decreases because the video bitrate exceeds the network throughput. Conversely, the buffer occupancy increases when the network speed is higher than the selected bitrate. If the rate adaptation is performed for a sufficient time in the proposed method, the buffer occupancy converges to $B_{\max}/2$ and remains stable unless the network bandwidth changes significantly.

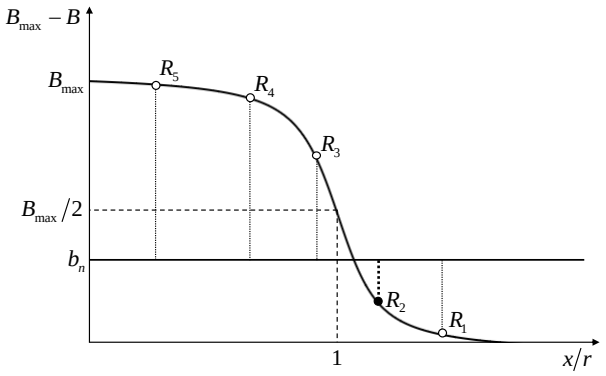

Fig. 6. Proposed buffer-based bitrate selection

5. Simulation Results

We performed a set of simulations to evaluate the proposed scheme in comparison with other conventional algorithms using the ns-3 network simulator [24]. To objectively evaluate the performance of the rate adaptation scheme, we implemented the HAS system in addition to the ABR, PANDA, and BBA algorithms. ABR is a basic rate adaptation scheme, and PANDA and BBA are the most representative algorithms for rate adaptation. A brief description of each algorithm is presented as follows.

• **ABR** estimates the available bandwidth through an arithmetic mean of the three most recent segment throughputs and selects the highest video bitrate that is lower than the measured available bandwidth [15].

• **PANDA** performs AIMD-like bandwidth estimation with an additive increment w and a multiplicative factor κ [12]. We used 0.3 and 0.28 as the defaults for w and κ, respectively.

• **BBA** uses a lower threshold of 90 s and an upper threshold of 24 s, for a buffer capacity of 240 s [21]. We set the thresholds at ratios of 3/8 and 9/10 for the variable buffer capacity.

5.1. Experimental Setup

As shown in Fig. 7, in all the simulation, a simple dumbbell network topology including TCP and UDP applications was used for generating competing traffic according to network profiles.

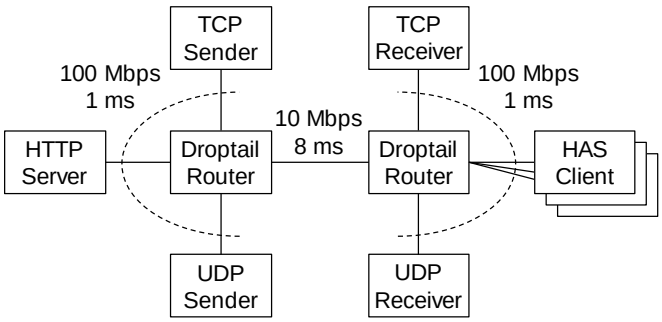

Fig. 7. Network topology used in the simulations

Table 2. Network profiles used in the simulations

Network Profile	Period (s)	Min (kbps)	Max (kbps)	Pattern
1	30	1500	5000	High-low-high
2	30	1500	5000	Low-high-low
3	-	2529	4110	FTP, Exponential ON-OFF, Pareto ON-OFF

We used two basic patterns for the network profile, i.e. high–low–high and low–high–low, according to the guideline of the DASH Industry Forum [25]. To simulate a general network environment, we constructed a network profile that generated highly variable traffic by combining three patterns of traffic models: FTP, Exponential ON-OFF, and Pareto ON-OFF. FTP is a file transfer protocol that sends packets using multiple TCP connections and thus transmits data at the maximum possible speed.

Exponential ON-OFF is a traditional traffic model of circuit-switched networks, whereas Pareto ON-OFF traffic represents a bursty characteristic of packet-switched networks. Detailed information regarding the network profiles is presented in Tables 2 and 3. Fig. 8 shows the bitrate changes of each profile in the simulation.

Table 3. Detailed settings of network profile 3

Pattern	Characteristic	Configuration
FTP	Greedy	TCP NewReno Always ON
Exponential ON-OFF	Poisson/Memoryless	BurstTime = 0.8 s IdleTime = 0.2 s Rate = 3 Mbps
Pareto ON-OFF	Long-tail/Bursty	BurstTime = 0.5 s IdleTime = 0.5 s Rate = 3 Mbps

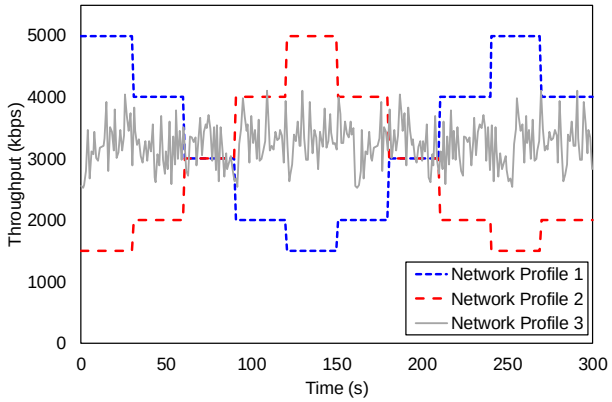

Fig. 8. Bitrate change in the network profiles

The video sample used in the experiment was encoded with six levels of quality, according to YouTube's recommended encoding settings [26]. As the quality level increased, the video bitrate increased exponentially. Table 4 presents the resolution and bitrate for each quality level. For further experiments, we divided the video sample into video segments 2–10 s in length.

Table 4. Configuration of the video bitrates

Quality level	Resolution	Bitrate (kbps)
1	160p	221
2	240p	614
3	360p	1384
4	480p	2462
5	720p	5535
6	1080p	12453

5.2. Performance Metric

We evaluated the rate adaptation scheme with regard to efficiency and stability. In the HAS system, efficiency corresponds to the overall video quality, and stability corresponds to the lack of changes in video quality. To measure the overall video quality, we calculated the average video bitrate for all segments using (9). To evaluate the change in the video quality and the magnitude of the change simultaneously, we defined a metric for the relative difference in the video bitrate, as follows.

$$\frac{1}{N-1}\sum_{n=1}^{N-1}\frac{\left|r_{n+1}-r_n\right|}{\min\left(r_{n+1},r_n\right)} \tag{16}$$

To take into account the variability of the device, video, and network, we performed 15 simulations for each rate adaptation scheme, while the changing buffer capacity and segment duration. For all the network profiles, the buffer capacity was changed to 30, 60, and 120 s, and the segment duration was changed to 2, 4, 6, 8, and 10 s. The simulation results were averaged, the standard deviations were calculated.

5.3. Performance Evaluation

Before comparing the proposed rate adaptation scheme with ABR, PANDA, and BBA, we describe its behavior. The proposed scheme selects the next bitrate according to the buffer occupancy and maintains the buffer occupancy through the playback buffer model. Fig. 9 shows that the proposed scheme maintained a stable buffer occupancy even in a highly variable environment. The proposed scheme tended to select lower bitrates to fill the playback buffer quickly when the buffer capacity was large. Because network profile 3 had increased network variability, the playback buffer model computed a higher value for the average buffer occupancy, owing to the increased CV. Therefore, the proposed scheme behaves conservatively when the network is unstable.

To compare the performance of the rate adaptation scheme, we calculated the average video bitrate and the relative difference in the video bitrate from all the simulation results, as shown in Fig. 10. Bandwidth-based rate adaptation schemes exhibit lower video bitrates and fewer bitrate changes than buffer-based schemes. PANDA exhibited worse performance than ABR when the network bandwidth was insufficient at the beginning of the streaming. This inefficiency indicates that the bandwidth estimation must be swift to catch network changes. Tuning the configuration parameters may solve this problem but should be done on a per-network basis. BBA exhibited a higher average video bitrate than ABR and PANDA, but there were unnecessary bitrate oscillations. Because BBA set thresholds of the buffer occupancy that divided the playback buffer into several areas, it changed the video quality too frequently when the buffer capacity was small. The proposed scheme exhibited a high average video bitrate, similar to BBA, but reduced the number of changes in the video bitrate by using the difference in the buffer occupancy.

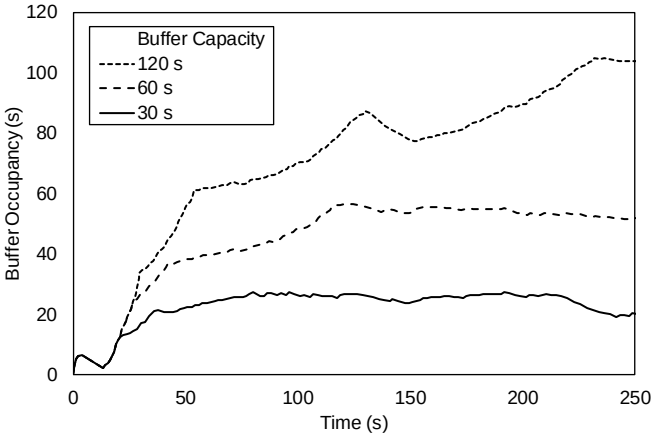

Fig. 9. Buffer occupancy of the proposed scheme in network profile 3

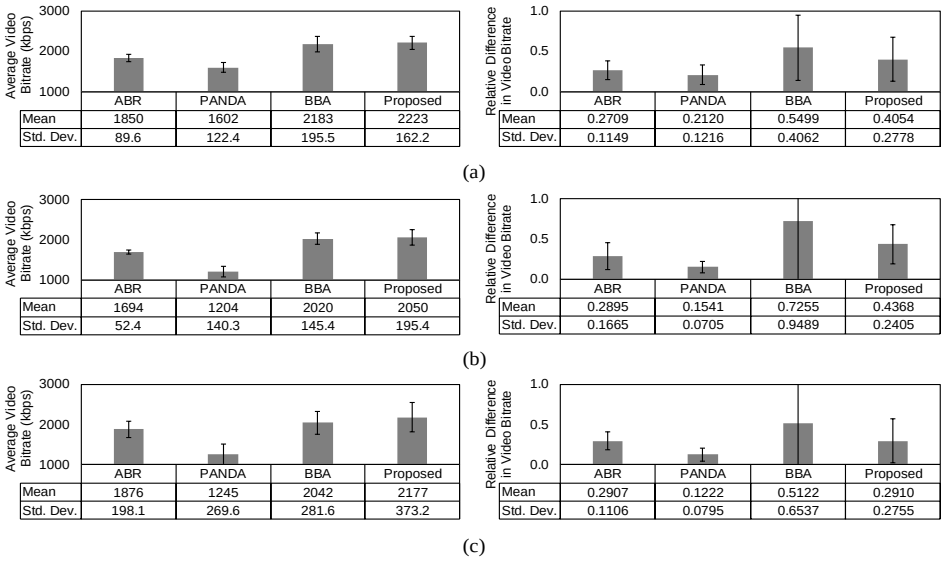

Fig. 10. Average video bitrate and relative difference in the video bitrate for all schemes in (a) network profile 1, (b) network profile 2, (c) network profile 3

Fig. 11 shows the results for all the rate adaptation schemes in network profile 1, where network bandwidth decreased and then increased. All the schemes performed rate adaptation at approximately 100 s owing to the reduction in the available bandwidth or buffer occupancy. ABR and PANDA select the next video bitrate according to the estimated available bandwidth; thus, they responded sensitively to network changes. PANDA performed rate adaptation more conservatively than ABR because of its AIMD-like bandwidth estimation. PANDA exhibited a slow quality improvement at the

beginning of the streaming. BBA selects the video bitrate according to the current buffer occupancy and changes video quality when the buffer occupancy exceeds the threshold. BBA frequently changed video quality when the buffer occupancy remained near the threshold. The proposed scheme also selected the video bitrate based on the buffer occupancy but did not change the video quality directly. The playback buffer model computed an expectation of the average buffer occupancy based on indirect information regarding the network, device, and video. Because the proposed bitrate selection method employed this value for rate adaptation, the proposed scheme was able to achieve consistent quality despite the variability.

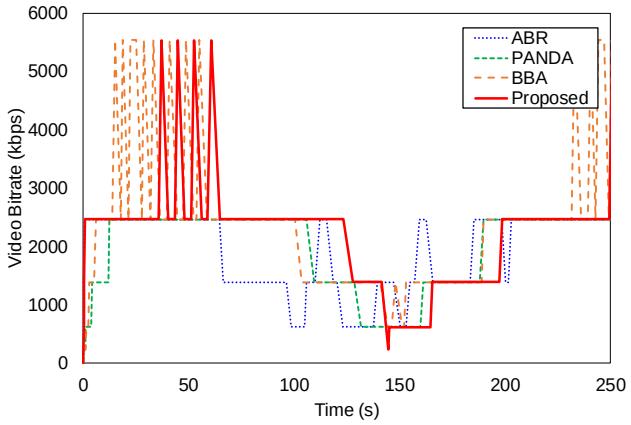

Fig. 11. Rate adaptation in network profile 1

By performing many experiments with various configurations, we observed that there was a tradeoff between the efficiency and stability in HAS. We plotted 12 points representing each rate adaptation scheme in each network profile, as shown in Fig. 12. More points close to the top left of the graph are interpreted as a better rate adaptation scheme. In the experiment, the performance of ABR and PANDA was determined by the accuracy of the bandwidth estimation. BBA could achieve better video quality regardless of the network conditions but made unnecessary quality changes that adversely influenced the user experience. The proposed scheme struck a balance between efficiency and stability and achieved better performance in some cases. The proposed scheme also employed a buffer-based bitrate selection algorithm, but its rate adaptation was performed using the playback buffer model comprising bandwidth, buffer, and video segment information. Therefore, the proposed scheme can provide consistent quality for HAS despite the variability of the network, device, and video.

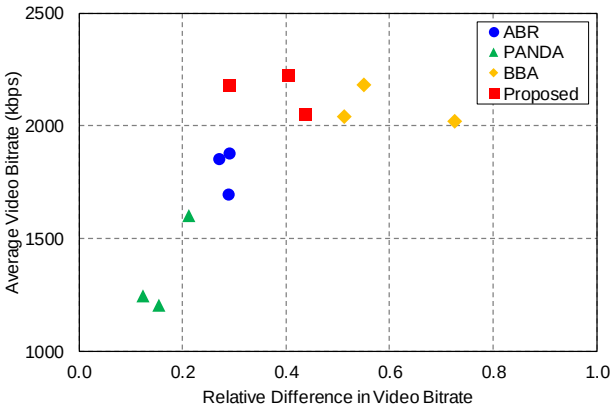

Fig. 12. Comparison of performance in network profiles 1, 2, and 3

6. Conclusion

To investigate the relationship between the bandwidth and the buffer in HAS, we developed a playback buffer model. The playback buffer was modeled based on the queueing theory, which is a proper way to analyze waiting entities. We predicted the average buffer occupancy by exploiting the playback buffer model. Because the average buffer occupancy is determined by the available bandwidth, segment duration, and buffer capacity, we propose a novel bitrate selection algorithm based on the playback buffer model. The proposed scheme sets the average buffer occupancy as a target and thus performs buffer-based rate adaptation for achieving consistent quality despite the variability of the network, device, and video. To evaluate the performance of the rate adaptation scheme, we implemented the HAS system in the ns-3 network simulator and conducted simulations with various configurations. We compared the proposed scheme with well-known rate adaptation algorithms with regard to the average video quality and the change in video quality. The simulation results indicated that the proposed scheme achieves very high video quality on average, even under unstable network conditions. Because the proposed scheme updates the expectation of the average buffer occupancy whenever it receives video segments, it responds to network changes without adjusting any parameters. However, the playback buffer modeling based on queuing theory has a limit in the real-world environments where the network bandwidth, videos watching by users, and number of users are changing more severely than simulation environments. To address this issue, we plan to extend the proposed scheme with a more practical buffer model for the real-world commercial HAS clients as a future work.

Acknowledgements. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIT) (No. NRF-2020R1F1A1048627). It has also been conducted by the excellent researcher support project of Kwangwoon University in 2021.

References

1. Cisco Public.: Cisco Visual Networking Index: Forecast and Trends, 2017-2022, (2019). [Online]. Available: https://davidellis.ca/wp-content/uploads/2019/05/cisco-vni-feb2019.pdf, last accessed date 2019/5/24
2. T. Stockhammer.: Dynamic Adaptive Streaming over HTTP - Standards and Design Principles. In Proceedings of the Second Annual ACM Conference on Multimedia Systems (MMSys*)*. San Jose CA, USA, 133–144. (2011)
3. Microsoft Azure.: Playback with Azure Media Player, (2019). [Online]. Available: http://www.iis.net/downloads/microsoft/smooth-streaming, last accessed date 2019/7/17
4. Apple Developer.: HTTP Live Streaming (HLS), (2019). [Online]. Available: https://developer.apple.com/streaming, last accessed date 2019/8/12
5. Adobe Live Video Streaming Online.: What is HTTP Dynamic Streaming? (2019). [Online]. Available: http://www.adobe.com/products/hds-dynamic-streaming.html, last accessed date 2019/6/29
6. Multimedia Communication.: HTTP Streaming of MPEG Media, (2012). [Online]. Available: https://multimediacommunication.blogspot.com/2010/05/http-streaming-of-mpeg-media.html, last accessed date 2012/4/26
7. O. Oyman, S. Singh.: Quality of Experience for HTTP Adaptive Streaming Services. IEEE Communications Magazine, Vol. 50, No. 4, 20–27. (2012)
8. J. Kua, G. Armitage, P. Branch.: A Survey of Rate Adaptation Techniques for Dynamic Adaptive Streaming over HTTP. IEEE Communications Surveys & Tutorials, Vol. 19, No. 3, 1842–1866. (2017)
9. S. Akhshabi, L. Anantakrishnan, A. C. Begen, C. Dovrolis.: What Happens When HTTP Adaptive Streaming Players Compete for Bandwidth?. In Proceedings of the 22nd International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV). New York, USA, 9-14. (2012)
10. X. Zhu, Z. Li, R. Pan, J. Gahm, H. Hu.: Fixing Multi-client Oscillations in HTTP-based Adaptive Streaming: A Control Theoretic Approach. In Proceedings of IEEE 15th International Workshop on Multimedia Signal Processing (MMSP). Santa Margherita di Pula, Sardinia, Italy, 230–235. (2013)
11. L. D. Cicco, V. Caldaralo, V. Palmisano, S. Mascolo.: ELASTIC: A Client-side Controller for Dynamic Adaptive Streaming over HTTP (DASH). In Proceedings of the 20th International Packet Video Workshop. San Jose, CA, USA, 1–8. (2013)
12. Z. Li, X. Zhu, J. Gahm, R. Pan, H. Hu, A. C. Began, D. Oran.: Probe and Adapt: Adaptation for HTTP Video Streaming at Scale. IEEE Journal on Selected Areas in Communications, Vol. 32, No. 4, 719–733. (2014)
13. T. Huang, N. Handigol, B. Heller, N. McKeown, R. Johari.: Confused, Timid, and Unstable: Picking a Video Streaming Rate is Hard. In Proceedings of the 2012 Internet Measurement Conference (IMC). Boston Massachusetts, USA, 225–238. (2012)
14. M. Allman, V. Paxson, E. Blanton.: TCP Congestion Control, (2009). [Online]. Available: https://tools.ietf.org/html/rfc5681, last accessed date 2020/1/21
15. DASH Industry Forum.: dash.js, (2019). [Online]. Available: https://github.com/Dash-Industry-Forum/dash.js, last accessed date 2020/7/26
16. C. Mueller, S. Lederer, R. Grandl, C. Timmerer.: Oscillation Compensating Dynamic Adaptive Streaming over HTTP. In Proceedings of the 2015 IEEE International Conference on Multimedia and Expo (ICME). Turin, Italy, 1–6. (2015)
17. P. Juluri, V. Tamarapalli, D. Medhi.: SARA: Segment Aware Rate Adaptation Algorithm for Dynamic Adaptive Streaming over HTTP. In Proceedings of the 2015 IEEE International Conference on Communication Workshop (ICCW). London, UK, 1765–1770. (2015)

18. K. Spiteri, R. Urgaonkar, R. K. Sitaraman.: BOLA: Near-optimal Bitrate Adaptation for Online Videos. In Proceedings of the 35th IEEE International Conference on Computer Communications (INFOCOM). San Francisco, CA, USA, 1–9. (2016)
19. R. Huysegems, B. D. Vleeschauwer, T. Wu, W. V. Leekwijck.: SVC-based HTTP Adaptive Streaming. Bell Labs Technical Journal, Vol. 16, No. 4, 25–41. (2012)
20. C. Sieber, T. Hoßfeld, T. Zinner, P. Tran-Gia, C. Timmerer.: Implementation and User-centric Comparison of a Novel Adaptation Logic for DASH with SVC. In Proceedings of the 2013 IFIP/IEEE International Symposium on Integrated Network Management (IM). Ghent, Belgium, 1318–1323. (2013)
21. T.-Y. Huang, R. Johari, N. McKeown, M. Trunnell, M. Watson.: A Buffer-based Approach to Rate Adaptation: Evidence from a Large Video Streaming Service. In Proceedings of the 2014 ACM Conference on SIGCOMM (SIGCOMM). Chicago Illinois, USA, 187–198. (2014)
22. Y. Xu, E. Altman, R. El-Azouzi, M. Haddad, S. Elayoubi, T. Jimenez.: Analysis of Buffer Starvation with Application to Objective QoE Optimization of Streaming Services. IEEE Transactions on Multimedia, Vol. 16, No. 3, 813–827. (2014)
23. J. F. C. Kingman.: The Single Server Queue in Heavy Traffic. Mathematical Proceedings of the Cambridge Philosophical Society, Vol. 57, No. 4, 902–904. (1961)
24. NSNAM.: ns-3 Network Simulator, (2019). [Online]. Available: https://www.nsnam.org, last accessed date 2020/10/7
25. DASH Industry Forum.: Guidelines for Implementation: DASH-AVC/264 Test Cases and Vectors, (2014). [Online]. Available: https://dashif.org/docs/DASH-AVC-264-Test-Vectors-v1.0.pdf, last accessed date 2014/3/24
26. Brandee.: Recommended Upload Encoding Settings, (2019). [Online]. Available: https://brandee.edu.vn/glossary/1722171-youtube-en/, last accessed date 2019/11/30

Jiwoo Park received his B.S. and Ph.D. degree from the Electronics and Communications Engineering Department, Kwangwoon University, Seoul, South Korea, in 2009 and 2019, respectively. His research interests include network protocols, multimedia systems, and video communications—in particular, QoS support in adaptive bitrate streaming.

Minsu Kim received his B.S. degree from the Electronics and Communications Engineering Department, Kwangwoon University, Seoul, South Korea, in 2017, where he is currently working toward a Ph.D. degree. His research interests include QoS/QoE support, multimedia systems, and streaming protocols.

Kwangsue Chung received his B.S. degree from Hanyang University, Seoul, South Korea, his M.S. degree from KAIST (Korea Advanced Institute of Science and Technology), Seoul, South Korea, Ph.D. degree from University of Florida, Gainesville, Florida, USA, all from the Electrical Engineering Department. Before joining the Kwangwoon University in 1993, he spent 10 years with the Electronics and Telecommunications Research Institute (ETRI) as a member of the research staff. He was also an adjunct professor at 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 mechanisms, and video streaming.

Received: August 20, 2020; Accepted: February 27, 2021.