https://doi.org/10.5392/IJoC.2018.14.4.030

A Buffer-Status Based HAS Video Transmission Scheme in Wireless Environments

In-Hye Kim
Department of Computer Engineering,
Kyungnam University, Changwon, Republic of Korea

Seung-Joon Seok
Kyungnam University, Changwon, Republic of Korea

ABSTRACT

Recently, HTTP Adaptive Streaming(HAS), a video streaming service over the HTTP based web platform has become common. The use of HAS service in mobile communication devices such as mobile phones and tablet PCs is rapidly expanding. This paper addresses ways to improve the quality of HAS service by enhancing the terms of viewer satisfaction. HAS systems have several internal operational processes, which can affect viewer satisfaction. Such processes include, the quality determination for the next video chunk, the TCP connections-setup procedure and the congestion control operation of the TCP. This paper proposes a transmission scheme to improve the HAS quality services over mobile web. The proposed scheme takes into consideration the past implicit communication state of the receiver's playback buffer occupancy. The results of these experiments indicate that the proposed scheme can improve the quality of HAS service from the mobile viewer's point of view.

Key words: *HAS, Video Steaming, BSVT, HAS's TCP Connection, Video Chunk Quality.*

1. INTRODUCTION

In the increasingly-used web-based adaptive video streaming service (HTTP Adaptive Streaming, HAS), the server stores video chunks encoded in various qualities, and the client uses a web browser to request and subsequently play video chunks of a quality appropriate to network conditions. That is, the HAS service transmits video content through HTTP GET request and response messages over a TCP connection between the client and the server [1]-[3]. These requests are made continuously at timed intervals, a process which allows for determining the quality of the next video chunks based on the transmission performance of the previously received video chunks. This reduces playback buffering issues due to changes in bandwidth, since it takes into account the bandwidth between the client and the server as well as the client's CPU state [3], [4]. If the available bandwidth increases, the quality of the next video chunk improves, and if the available bandwith decreased, the quality of the next video chunk degrades. However, this scheme causes frequent quality changes in the video stream, which lowers viewer satisfaction. These frequent changes are likely due to high congestion and a long delay in wireless transmission, resulting from a change in signal strength and interference from radio waves stemming from movement of a mobile communication device. When a video streaming service is accessed on a mobile communication device in particular, it can lead to an increased frequency of quality change and quality variation of the video chunks. In addition, the congestion control operation of TCP is designed for a wired environment, leading to more playback delay or even stopped playback over a wireless network. In this paper, we propose a scheme to increase viewer satisfaction in a wireless environment by modifying HAS and TCP operation.

The rest of the paper is organized as follows: in section 2, we describe the behaviors of HAS, analyzing the schemes of determining HAS transmission video quality, the impact of TCP congestion control on HAS, and the TCP connection-setup procedure. Section 3 describes the BSVT video chunk transmission scheme for improving HAS service operation. Section 4 describes the occupancy and implementation of a simulation environment for performance evaluation. Finally, we conclude in section 5.

* *Corresponding author, Email: sjseok@kyungnam.ac.kr*
Manuscript received Jul. 26, 2018; revised Aug. 13, 2018; accepted Aug. 14, 2018

2. BEHAVIORS OF HTTP ADAPTIVE STREAMING SERVICE

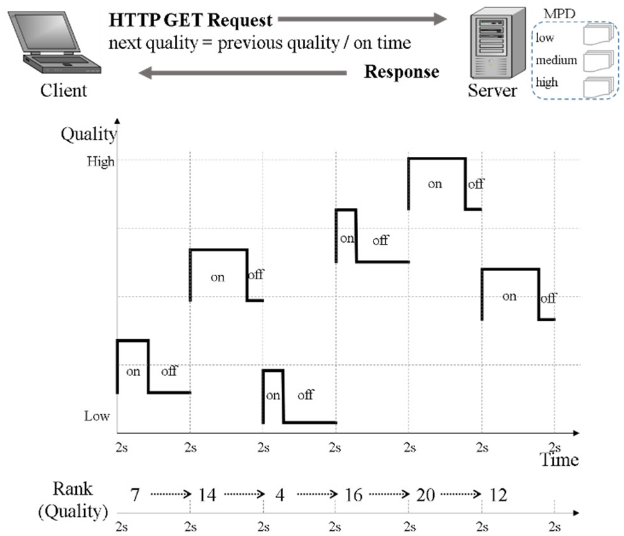


Fig. 1. HAS Transmission Quality Determination

This section describes the internal operational process of the HAS service. It also describes the characteristics of the video quality determination scheme that was previously studied for HAS services.

2.1 Video Quality Determination

Fig. 1 shows that the HAS service scheme receives video information stored on the server through the MPD file and calculates the quality of the request chunk using the default baud rate (bps) value in order to determine the video quality. Prior to an HTTP GET request, the scheme calculates the quality by dividing the size of the downloaded video chunk by the time it took to download the previous video chunk [4], [5]. This video quality determination scheme can provide adaptive quality by requiring bit rate chunks that match the currently available bandwidth. Additionally, if the user stops playing the video, the amount of wasted data consumption can be reduced by not asking for any more video chunks. However, video quality determination of this sort that is appropriate for such a variable network degrades the user's subjective quality of experience (QoE) [6]. The characteristics of this HAS transmission quality determination scheme that have been studied in order to address these problem are as follows:

The Rate Adaptation Algorithm (RAA) compares the segment download time to the media period in order to predict TCP throughput, and determines the quality of the next chunk based on the throughput that smoothes the predicted capacity [7]. Because this method increases quality a single step at a time, it can take a long time to reach maximum quality. Furthermore, when the quality is reduced, it is adapted to the throughput and subsequently shows a sudden change in quality.

QDASH(A QoE-aware DASH), which uses an available bandwidth measurement module and a quality evaluation module, installs a detection module on the server side for accurate bandwidth measurement and selects a quality corresponding to the measured bandwidth. This method minimizes the amount of quality change by choosing the right video quality for the specific bandwidth [8]. This method is costly, however, because the hardware must be installed on both the server and the client side.

ASAC (Adaptive Streaming of Audiovisual content using MPEG DASH) minimizes quality fluctuations by choosing video quality as an Exponential Weighted Moving Average. The weights are also calculated as the deviation between the past throughput and the measured throughput, so the quality changes only if the observed change in transmission rate is large [9]. If the change in bit rate is small, however, a bit rate lower than the available bandwidth is selected, and if the throughput changes drastically, that frequent quality changes occur .

The BAHS (Buffer-base Adaptation for Adaptive HTTP Streaming) scheme adjusts the transmission quality by choosing a path of selectable quality from the grid type based on the remaining capacity of the buffer, and then calculating the increase or decrease in the video quality level [10]. By limiting the number of quality change steps, this scheme cannot cope with sudden network changes.

QAAD (QoE-enhanced adaptation algorithm over DASH) is an algorithm that measures the available bandwidth at a fixed time. After smoothing the measured values, it changes the quality based on the difference between the measured value and the maximum quality that does not exceed the current available bandwidth [11]. In a poor network environment, this limits quality selection depending on the bandwidth.

2.2 Impact of TCP Congestion Control

TCP performs congestion control to reduce packet loss resulting from differences in the data processing speed and network speed between the client and server [12]-[14]. CUBIC TCP applied to a wireless environment starts in slow-start mode and increases the congestion window (CWND). If congestion occurs, CWND increases over time according to a cubic function. In this case, it is quickly restored to ssthreshold(slow start threshold), its size before the last congestion. When CWND is restored, it is slowly increases in order to try to get additional bandwidth. Then, CWND increases gradually and rapidly. If the existing algorithm increases the CWND as an answer to the ACK response, the CUBIC algorithm operates independently of the RTT and grows the window according to the time through the convex function and the concave function, in order to occupy the available bandwidth quickly [15].

As shown in Fig. 2, the HAS scheme is a service that receives video through periodic chunk requests of 2 to 10 seconds. The HAS scheme shows On / Off traffic patterns that differ from those of traditional TCP traffic. The on period is the time when the requested chunk was downloaded, and the Off period is the idle time before requesting the next chunk [16], [17]. In a situation where the bandwidth remains constant, the quality is high when the On period is long, and the quality is low when the On period is short. This adjustment scheme cannot use the actual available bandwidth, since the window size is not reflected during the Off period and the next chunk starts at CWND of the On period of the previous chunk. HAS's request chunk quality determination scheme predicts the

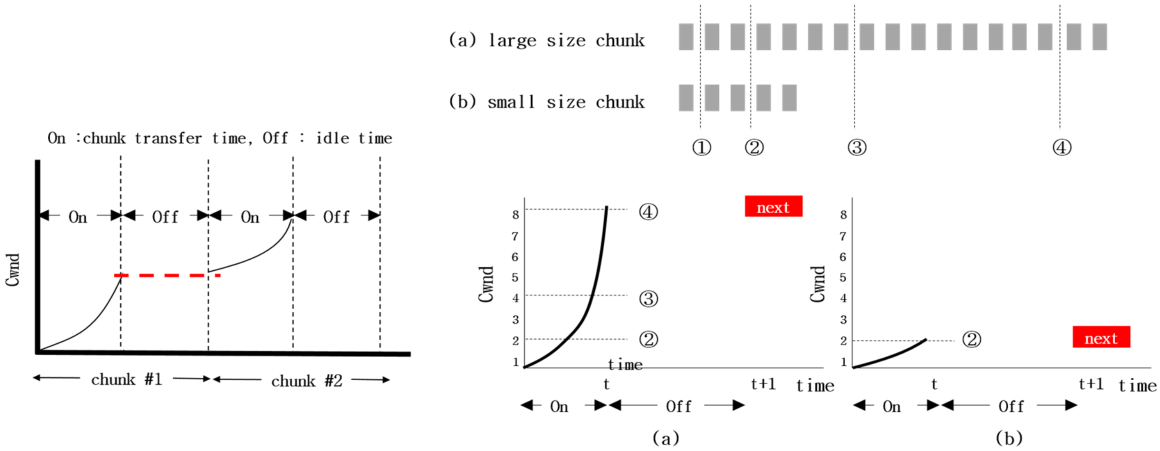


Fig. 2. Cwnd by On/Off traffic patterns(left), And Cwnd by file size(right)

current and future network conditions according to the time of downloading the previous video chunk quality and the size of the chunk. This scheme, however, does not take slow start intervals into account. In general, if the transfer file is large, it size has little effect on the segment transfer rate calculation. However, in the case of HAS, which mainly downloads small segments, most transmissions are terminated within a slow-start interval. If CWND does not increase enough, it can become a problem [15]. This is due to the fact that even if the actual available bandwidth is the same, the available bandwidth measured in relation to the previous video chunk size may appear differently.

2.3 TCP Connection for HAS

HTTP works as a server / client model as it sends a request message through the TCP socket interface and receives a response message from the server. Connection-oriented TCP establishes a connection via three-way handshaking and transmits data through the connected bidirectional session according to the window size. When the connection is terminated, a four-way handshaking process is performed that independently closes both bidirectional connections. These frequent connections and disconnects are costly [18]. In the case of HAS service with the 1.0 version of HTTP, a new TCP connection is created for each chunk of the transport. In version 1.1, however, multiple video chunks are downloaded from a single connection. However, if the user skips a video while the video chunks are being downloaded, or if there are no additional chunk requests even after video chunk downloading has been completed, then a new TCP connection is created. A new TCP connection is also created when the video resolution changes. Additionally, when the playback buffer is full, the TCP connection is closed, and the next video chunks are downloaded over a new TCP connection [19]-[25] If a new TCP connection is established for each transport chunk, unnecessary RTTs resulting from these frequent connections as well as disconnects increase network latency and overhead costs. Due to the slow start mode behavior of TCP, video chunk downloading may also continue to be transmitted at a slower rate, or chunks that were being transmitted over an existing TCP connection may be lost [18].

3. PROPOSED BUFFER STATE BASED HAS VIDEO CHUNK TRANSMISSION SCHEME(BSVT)

This section describes the proposed Buffer Status Based HAS video chunk transmission scheme, BSVT (Buffer Status Based HAS video chunk transmission). In a wireless environment, selecting video quality based on the network transmission status leads to an increased frequency of quality changes, resulting in a lowered video quality experienced by the user. The BSVT is a scheme that aims to reduce the number of transmission quality changes based on the occupancy (the amount of data and the number of chunks) of the buffered data of the playback buffer. The amount of data remaining in the playback buffer can be the same in two cases, if there are fewer high quality chunks or a lot of low quality chunks. If a large amount of data is in the playback buffer but the number of chunks is small, playback may stop in a poor network environment. Fig. 3 compares the quality determinations of the HAS and BSVT schemes. As shown, in the BSVT scheme, the quality of the requested video chunks is determined by the average communication quality based on the data occupancy rate of the playback buffer. It also determines the new TCP connection based on the share of the data in the replay buffer. Prevents new TCP connections as much as possible to reduce the overhead of opening and closing a TCP connection.

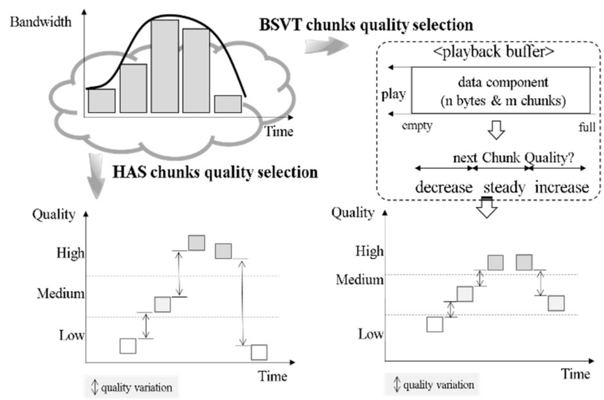


Fig. 3. HAS vs BSVT chunk quality selection

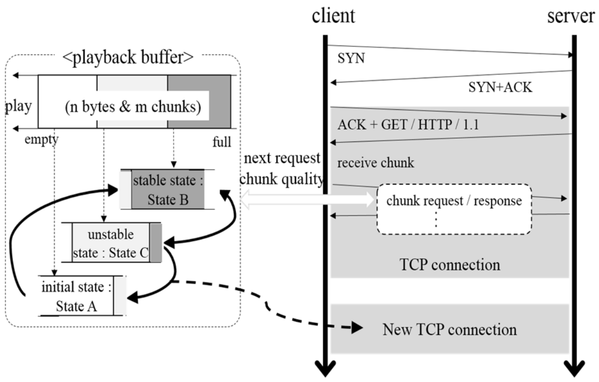


Fig. 4. BSVT overview

3.1 BSVT Overview

The "BSVT" scheme classifies video chunks into one of three states, based on the data occupancy rate of the client's playback buffer, as shown in Fig. 4 It then uses the estimation equation to calculate the quality of the video chunks, then determines the quality of the next video chunks to be requested based on the computed value and the state. The states classified according to the data occupancy (the amount of data and the number of chunks) of the playback buffer are as follows. First, State A is the initial buffering state. In this state, a certain number of video chunks is continuously requested without an off period to fill the playback buffer. Additionally, if state transitions are set to State A for rebuffering, the existing TCP connection is broken and a new TCP connection is established. Second, State B is the state in which more data occupancy is maintained than the amount of initial buffering and the number of reference chunks. Finally, State C is a state for quickly filling the playback buffer because the data occupancy of the playback buffer is less than the number of reference chunks. High-level video chunks do not necessarily indicate good chunk quality levels. Depending on the network environment, the highest quality video chunks that are still within the available bandwidth and minimize quality changes should be chosen.

3.2 Determination of Video Quality

The video quality to be transmitted next in the BSTV is determined by calculating the quality according to the ON period and adjusting the quality according to the occupancy of the playback buffer.

Step 1 : Video Quality Calculation. The HAS scheme determines the quality of the next video chunks based on the download time, which is according to the changing network. This video quality determination scheme often changes the quality of the video stream, lowering viewer satisfaction. The HAS scheme also uses the previously downloaded On time to determine the quality of the next video chunk, and network changes during the Off period are not applied. The proposed BSVT scheme is calculated using the exponentially weighted moving average (EWMA) of Eq. (1) in order to minimize the quality changes. The proposed scheme is also calculated as in Eq. (2) based on the measured RTT value in order to apply the network change during the Off period. As shown in Fig. 5, when the On period is compared with the Off period and the Off period is more than twice the On period, the quality is estimated as follows to apply the network change during the Off period: RTT is measured using ICMP Echo (Internet Control Message) at 1/4 time in the 1/4 interval of the video chunk period, and the estimated quality is determined by Eq. (2) based on the measured RTT value.

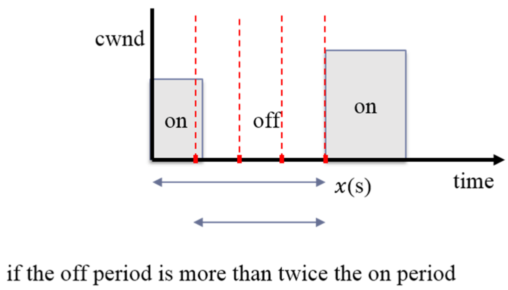
$$\mathrm{chunk.rate}_{t+1} = \\ chunk.rate_{t-1} \times \delta + chunk.rate_t \times (1-\delta) \quad (1)$$

$$chunk.rate_{t+1} = \frac{chunk.size_t}{RTT/2} \quad (2)$$

$$rate.var_t = |chunk.rate_{t-1} - chunk.rate_t| \\ \mathrm{rate.var.avg} = \frac{\sum_{i=0}^{n-1} rate.var_{t-i}}{n} \quad (3)$$

Step 2 : Video Quality Adjusting. In the proposed BSVT, the playback buffer is classified as either State A, State B, and State C.

State A is divided into the initial-buffering phase and the re-buffering phase. The initial-buffering phase refers to when the client's playback buffer is filled with the user's play-start, and re-buffering refers to the phase of filling the playback buffer again when the number of chunks remaining in the playback buffer reaches zero. The initial-buffering phase starts at a consistently low quality without an Off period, and then fills the playback buffer by changing the quality of the video chunks according to the aforementioned quality calculation. In the re-buffering phase, video chunks are continuously requested without an Off period while video is playing, until the number of chunks remaining in the playback buffer is equal to the number of reference chunks.



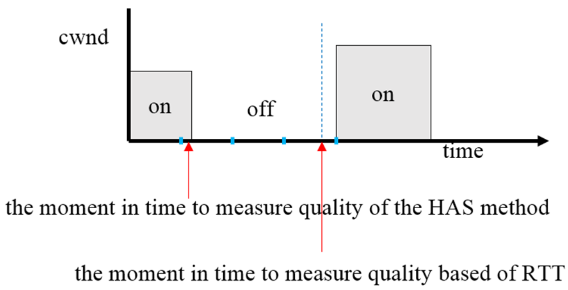


Fig. 5. Quality measurement decision by RTT(left), And RTT-based quality measurement timing(right)

State B is a stable state in which the number of video chunks remaining in the playback buffer is kept constant. The playback buffer occupancy (the amount of data and the number of chunks) is compared with the previous playback buffer for the purpose of classification as follows: If both the number of video chunks and the amount of data increase, it is classified as Case B1, if both numbers remain the same it is classified as Case B3, and if both numbers decrease it is classified as Case B5. Furthermore, if the number of chunks decreases while the amount of data increases, it is classified as Case B2, and conversely, if the number of chunks increases while the amount of data decreases, it is classified as Case B4. The quality of video chunks is determined by this Case classification. The estimated quality for the quality adjustment is calculated by either Eq. (1) or Eq. (2), according to the On period in which the chunks are downloaded. Then, the quality of a video chunk one level higher than the calculated video chunk quality is compared with the quality obtained by adding the average variation calculated by Eq. (3) from the previous video chunk quality. Additionally, in order to adjust the quality according to the Case classification, a smaller video chunk quality should be chosen from these values. Case B1 continues to increase quality at the estimated quality calculated as described above, and Case B2 decides at a quality one level lower than the estimated quality. Case B3 maintains the quality of the estimated quality, Case B4 determines the quality one level higher than the estimated quality, and Case B5 continues to the quality lower than the estimated quality.

The unstable state C, where the number of video chunks remaining in the playback buffer is fewer than the number of reference chunks is classified as either Case C1, Case C2, and Case C3, by comparing the playback buffer occupancy with the previous playback buffer. If the number of chunks increases or the amount of data decreases, it is classified as Case C1, if they remain the same, it is classified as Case C2, and if both decrease, it is classified as Case C3. As for the video chunk quality for adjustment, compare a quality one level lower than the estimated quality calculated by the on period with the quality obtained by subtracting the average change amount calculated by Eq. (3) from the previous quality. The higher of these values is the value determined to adjust the quality. Case C1 is determined to be one level higher than the estimated quality, Case C2 is determined to be one level lower than the estimated quality, and Case C3 is determined to be one level lower than the estimated quality.

3.3 TCP Connection for BSVT

TCP connections are released when a connection close occurs on the server and client side, or when the connection times out. The HAS scheme does not specify a standard for establishing and breaking connections. In the proposed BSVT chunk transmission scheme, the TCP connection state is maintained as in the HTTP 1.1 persistent mode, and the chunk is transmitted through single TCP connection. However, if there is an unstable network situation, such as the presence of channel disconnection and delay, it will consume the number of chunks remaining in the playback buffer, and the state will change back to the buffering phase, leading the network path to be transmitted over the new TCP connection. In other words, if the network conditions are not good and re-buffering occurs, the video chunks will be downloaded from the new network path. This approach reduces the overhead cost of establishing a new TCP connection for each request on a non-persistent connection in HTTP 1.0 and additionally reduces the latency caused by the new connection establishment process.

4. PERFORMANCE EVALUATION

In this section, performance is analyzed by comparing the existing scheme with the proposed scheme.

4.1 Experiment Environment

To evaluate the performance of the proposed BSVT scheme, we compared the video chunk quality determination scheme of the HAS scheme, the EWMA scheme, and the BSVT scheme. The comparison tool is implemented in the Java language and compares the latency of the transmission interval that can occur in a wireless environment according to congestion. In the simulation, congestion is modeled such that the mean transmission time varies, and on / off period are randomly determined by applied Exponential Probability Density Function. The number of reference chunks of the playback buffer for each state is determined as 5, and δ used for Eq. (1) is set at 0.825. The video content is the DASH Dataset 'BigBuckBunny Video', which is an open license, on demand streaming scheme, and is made up of chunks split into 2 seconds. The various bit rates of the video chunks are shown in Table 1[26].

Table 1. Bit rate and chunk quality rating according to video source resolution

resolution	data quantity (byte)	basic transmission rate for quality determination(bps)	rank (quality)
320x240	11,406	45,625	1
320x240	22,321	89,283	2
320x240	32,772	131,087	3
480x360	44,588	178,351	4
480x360	55,400	221,600	5
480x360	65,634	262,537	6
480x360	83,587	334,349	7
480x360	99,032	396,126	8
854x480	130,572	522,286	9
854x480	148,873	595,491	10
1280x720	197,796	791,182	11
1280x720	258,171	1,032,682	12
1280x720	311,195	1,244,778	13
1280x720	386,726	1,546,902	14
1920x1080	533,423	2,133,691	15
1920x1080	621,034	2,484,135	16
1920x1080	769,647	3,078,587	17
1920x1080	881,731	3,526,922	18
1920x1080	960,090	3,840,360	19
1920x1080	1,054,974	4,219,897	20

4.2 Experiment Result

Three experiments were performed to evaluate the performance of the proposed BSVT scheme.

Table 2 is the result of the analysis of the behavior of the quality determination schemes comparing the quality of video chunks, which was determined after differentially modeling the chunk transfer time for slowly changing networks and rapidly changing networks.

Table 2. Analysis of quality determination schemes

Trans-mission time	slowly change			fast change		
	HAS	EWMA	BSVT	HAS	EWMA	BSVT
100~1000ms	19.94	19.69	19.6	19.95	19.69	19.6
500~1500ms	19.92	19.68	19.68	19.93	19.67	19.67
1000~2000ms	19.86	19.33	19.35	19.91	18.04	19.06
1500~2500ms	4.68	5.41	5.44	2.46	3.77	6.43

Table 3 shows the experimental results of the existing HAS, EWMA, and BSVT schemes, according to the congestion state with the ON interval determined by the exponential distribution probability distribution function. The result of this experiment compares the quality of the determined video chunks with the number of re-buffering.

Table 3. Performance evaluation by congestion state

average rate	quality rating of request chunk			re-buffering count		
	HAS	EWMA	BSVT	HAS	EWMA	BSVT
100ms	19.99	19.94	19.93	0	0	0
300ms	19.98	19.94	19.92	0	0	0
600ms	19.84	19.89	19.65	15	10	0
900ms	19.45	19.69	19.03	62	43	0
1200ms	19.1	19.1	18.5	106	117	9
1500ms	17.99	18.36	18.18	205	238	33
1800ms	16.57	17.86	17.48	305	333	89

In addition, in order to evaluate performance in a wireless environment, the transmission time of the video chunks is determined by the exponential probability density function according to the congestion level 1-5 (level 1 : 100~500ms, level 2 : 100~1000ms, level 3 : 100~1500ms, level 4 : 100~2000ms, level 5 : 100~2500ms), with the congestion occurring randomly. Fig. 6 compares the quality of the video chunks determined in each method, and Fig. 7 shows the result of comparing and analyzing the number of chunks remaining in the buffer and the amount of data. Fig. 8 compares the number of re-buffering times, new TCP connections based on resolution changes, and the new TCP connections due to re-buffering. The BSVT scheme maintains quality similar to that of the existing scheme, but the least amount of re-buffering occurs as the congestion level increases. This is because the amount of data in the buffer and the number of chunks are kept constant.

Table 4. Comparison according to basic transmission rate change (average quality and rebuffering)

basic trans-mission rate	degree of congestion (ms)	average quality rank			rebuffering(count)		
		HAS	EWMA	BSVT	HAS	EWMA	BSVT
1s	100~500	19.80	19.72	19.63	0	0	0
	100~1000	18.90	18.91	19.28	7	7	0
	100~1500	16.82	18.33	18.77	29	40	1
	100~2000	13.82	16.48	15.70	100	103	16
	100~2500	12.46	14.59	13.41	207	199	63
1.5s	100~500	19.89	19.66	19.64	0	0	0
	100~1000	19.67	19.61	19.32	4	4	0
	100~1500	18.07	19.13	18.33	53	33	2
	100~2000	17.05	18.63	18.19	112	102	30
	100~2500	15.00	17.58	16.76	224	216	90
2s	100~500	19.93	19.71	19.64	0	0	0
	100~1000	19.79	19.70	19.72	6	8	0
	100~1500	19.25	19.41	19.06	37	48	3
	100~2000	18.92	19.31	18.63	88	82	16
	100~2500	17.80	18.40	18.20	210	267	67

Table 5. Comparison according to basic transmission rate change (new TCP connection)

basic trans-mission rate	degree of congestion (ms)	count of connections due to resolution change			count of connection due to rebuffering
		HAS	EWMA	BSVT	
1s	100~500	3	2	2	0
	100~1000	52	2	2	5
	100~1500	137	12	12	12
	100~2000	258	68	94	17
	100~2500	344	106	154	24
1.5s	100~500	2	3	3	0
	100~1000	13	2	4	3
	100~1500	105	2	18	11
	100~2000	187	8	28	16
	100~2500	190	38	50	23
2s	100~500	1	2	2	0
	100~1000	5	2	2	2
	100~1500	34	2	4	8
	100~2000	66	2	12	19
	100~2500	177	12	26	23

The bit rate and chunk quality rating according to the video source resolution used in the experiment are based on the default transmission rate of 2 seconds. So, this experiment investigated what quality is determined when a 2 second chunk (transmitted for 2 seconds) is transmitted in 1.5 second and in 1 second. As a result, the average quality and the number of re-buffering are shown in Table 4, and the number of new TCP connections is shown in Table 5. At a 2-second transmission rate, the re-buffering frequency is smaller and the quality is better. Also, as the transmission rate decreases to 1.5 and 1 second, the quality is determined passively and rigorously. That is, the smaller the reference transmission rate, the lower the average quality and the higher the quality variation. As a result of comparing the existing scheme with the proposed BSVT scheme, the proposed method shows better performance when the reference rate is 2 seconds.

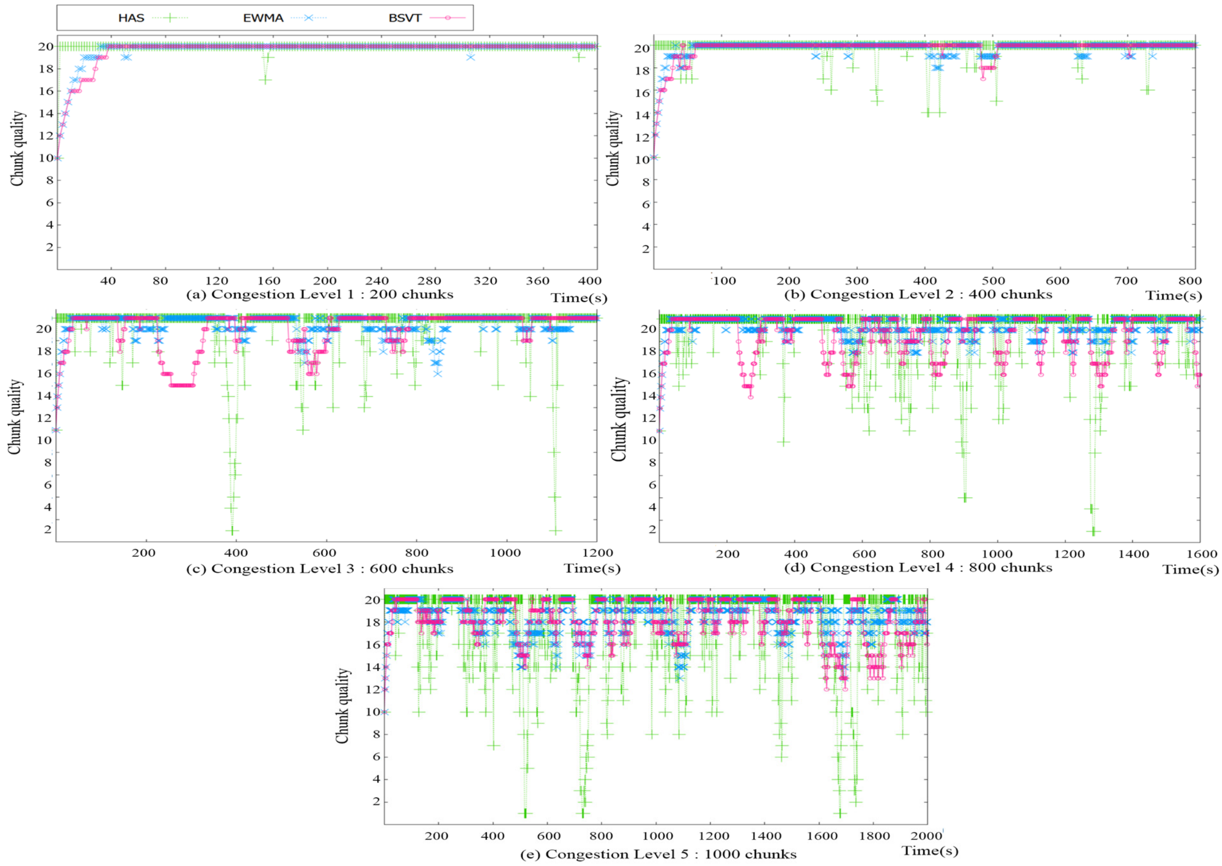


Fig. 6. Comparison of request chunk quality in wireless environment : Congestion level 1~5

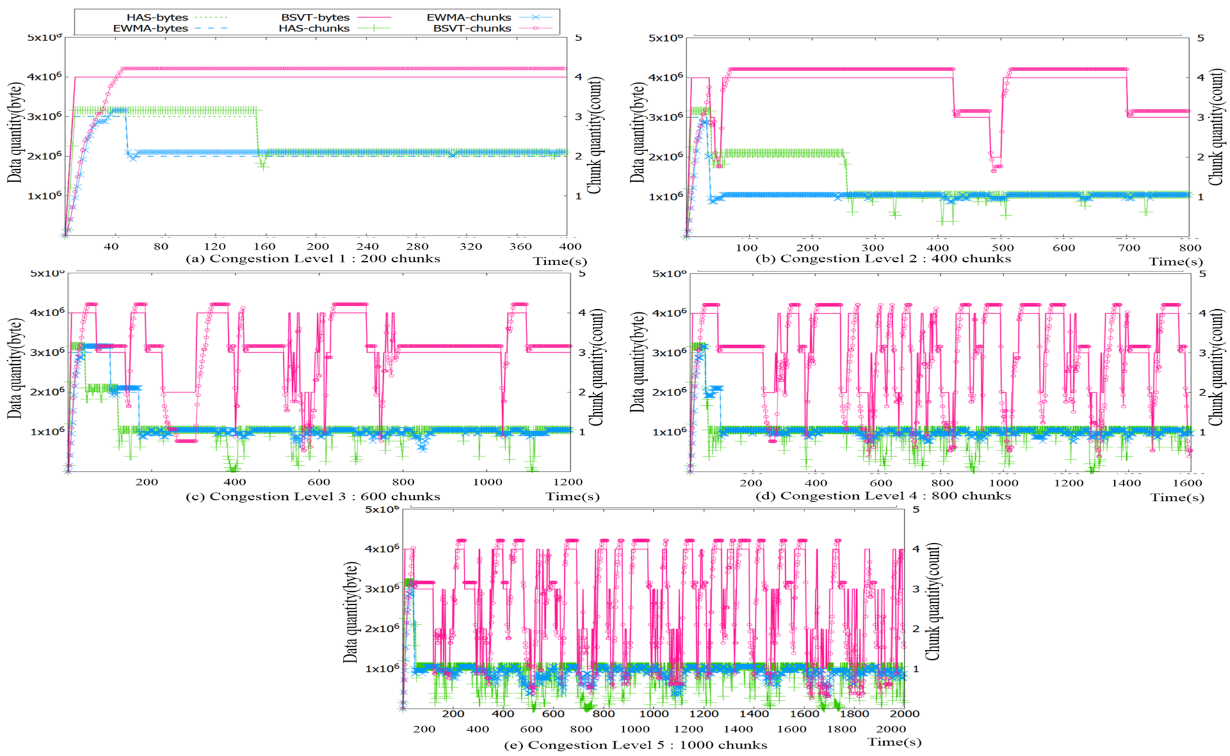


Fig. 7. Comparison of the amount of data remaining in the buffer and the number of chunks in wireless environment : Congestion level 1~5

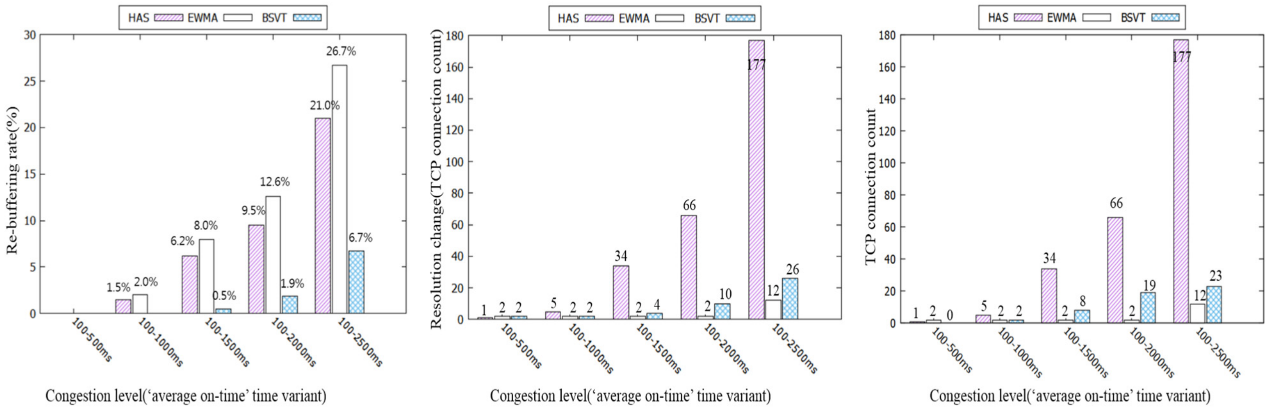


Fig. 8. Comparing the number of re-buffering in wireless environment, comparison of new TCP connection according to resolution, and comparison of TCP connection according to re-buffering

5. CONCLUSION

The traditional HAS video quality determination scheme often changes the quality in a wireless environment, lowering viewer satisfaction. Additionally, if mobile users want to receive video streaming services in a wireless environment using a TCP transmission protocol designed for wired environments, network performance may be further compromised due to TCP traffic congestion control and TCP connectivity issues. To address these issues, the proposed BSVT scheme reduces the variation in video quality and guarantees an average quality level, and the transmission status is categorized by the receiver's playback buffer occupancy (the amount of data and chunks remaining in the client-side playback buffer). The status of these classifications determines the quality of the next request chunk and whether a new TCP connection will be established. In the BSVT scheme, the quality of the request chunk is determined by calculating the exponentially weighted moving average and the average variation according to the time of downloading (On period) the chunks, in order to reduce the quality change of the video chunks. Additionally, in this scheme, new TCP connections are avoided as much as possible in order to reduce the overhead of opening and closing TCP connections.

The proposed BSVT scheme can improve viewer satisfaction in a wireless environment by adjusting the chunk quality according to the state of the playback buffer and determining a new TCP connection. In future studies, we will test our proposed scheme under heterogeneous terminals with various content requests.

REFERENCES

[1] Qi Lin, Yitong Liu, Yun Shen, Hui Shen, Lin Sang, and Dacheng Yang, "Bandwidth Estimation of Rate Adaption Algorithm in DASH," IEEE International Workshop on Management of Emerging Networks and Services, 2014.

[2] ISO/IEC 23009-1, "Dynamic adaptive streaming over HTTP (DASH)," Part 1, 2014.

[3] T. Stockhammer, "Dynamic Adaptive Streaming over HTTP-Standards and Design Principles," Proc. of the 2nd ACM Multimedia Systems Conference, 2011, pp. 133-144.

[4] O. Oyman and S. Singh, Quality of experience for HTTP adaptive streaming services," IEEE Commun. Mag., vol. 50, no. 4, 2012, pp. 20-27.

[5] Yong Han Kim and Min Kyu Park, "MPEG-DASH Services for 3D Contents Based on DMB AF," JEB, vol. 18, no. 1, 2013, pp. 115-121.

[6] 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 Commun. Surv. Tuts, vol. 17, no. 1, 2015, pp. 469-492.

[7] C. Liu, I. Bouazizi, and M. Gabbouj, "Rate Adaptation for Adaptive HTTP Streaming," Proc. for the ACM MMSys'11, 2011, pp. 169-174.

[8] R. Mok, X. Luo, E. Chan, and R. Chang, "QDASH: A QoE-aware DASH system," in Proc. ACM MMSys 2012, 2012.

[9] T. Thang, Q. Ho, J. Kang, and A. Tham, "Adaptive Streaming of Audiovisual Content Using MPEG DASH," IEEE Transactions on Consumer Electronics, vol. 58, no. 1, 2012, pp. 78-85.

[10] Hung T. Le, Duc V. Nguyen, Nam Pham Ngoc, Anh T. Pham, and Trung Cong Thang, "Buffer-based Bitrate Adaptation for Adaptive HTTP Streaming," Proc.of the international Conference on Advanced Technologies for Communications, 2013, pp. 33-38.

[11] D. Suh, I. Jang, and S. Pack, "A Video Bitrate Algorithm for DASH-Based Multimedia Streaming Services to Enhance User QoE," The Journal of Korea Information and Communications Society, vol. 39, no. 6, 2014, pp. 341-349.

[12] Y. J. Lee, "Mathematical Model for Mean Transfer Delay of Web Object in Initial Slow Start Phase," The Journal of Korean Institute of Industrial Education, vol. 33, no. 2, 2008, pp. 248-258.

[13] Laura Chappell, "Inside the TCP Handshake," NetWare Connectiion, 2 sheets, 2000.

[14] J. Postel, "Transmission control protocol request for comments 793," 1981.

[15] S. Agarwal, A. Jain, and A. Chadha, "CUBIC High-Speed Algorithms Implemented In Linux-2.6.31," Int. J.Innov. Res. Develop, vol. 2, no. 3, 2013, pp. 428-439.
[16] http://www.mediaentertainmentinfo.com/2015/04/6-concept-series-what-is-thedifference-between-progressive-download-rtmp-streaming-and-adaptive-strea ming.html/
[17] T. Kupka, P. Halvorsen, and C. Griwodz, "Performance of On-Off Traffic Stemming From Live Adaptive Segmented HTTP Video Streaming," In Proc. of IEEE Conference on Local Computer Networks, 2012, pp. 405-413.
[18] S. Bae, D. Jang, and K. Park, "Why is http adaptive streaming so hard," Proceedings of the 6th Asia-Pacific Workshop on Systems, 2015.
[19] P. J. Leach, T. Berners-Lee, J. C. Mogul, L. Masinter, R. T. Fielding, and J. Gettys, "Hypertext Transfer Protocol - HTTP/1.1," IETF RFC 2616, 1999.
[20] C. Mueller, S. Lederer, C. Timmerer, and H. Hellwagner, "Dynamic adaptive streaming over HTTP/2.0," in Proceedings of the IEEE International Conference on Multimedia and Expo, 2013, pp. 1-6.
[21] Tomas Kupka*, On the HTTP segment streaming potentials and performance improvements*, Oslo University, Ph.D. thesis, 2013.
[22] H. Nam, K. H. Kim, B. H. Kim, D. Calin, and H. Schulzrinne, "Towards A Dynamic QoS-aware Over-The-Top Video Streaming in LTE," IEEE WoWMoM 2014, 2014, pp. 1-9.
[23] Wen Hu, Zhi Wang, and Lifeng Sun, "A Measurement Study of TCP Performance for Chunk Delivery in DASH," IEEE Communications Surevys&Turorials, 2016.
[24] Saamer Akhshabi, Ali C. Begen, and Constantine Dovrolis, "An experimental evaluation of rate-adaptation algorithms in adaptive streaming over HTTP," Proceedings of the second annual ACM conference on Multimedia systems, 2011.
[25] William David Diego Maza, "A Framework for Generating HTTP Adaptive Streaming Traffic in ns-3," International Conference on Simulation Tools and Techniques-2016, 2016.
[26] http://www-itec.uni-klu.ac.at/ftp/datasets/DASHDataset 2014/BigBuckBunny/2se

In-Hye Kim
She received the Ph.D degrees in computer science from Gyeongnam National University, Korea in 2017. Currently she is working at College of General Education, Kyungnam University. Some of her research interests include the future internet, wireless Internet protocol, DTN, and HAS, ect.

Seung-Joon Seok
He received his Ph.D degrees in Electronics Engineering from Korea University, Seoul, Korea, in 2003. He has worked for computer engineering department, Kyungnam University since 2004. Some of his research interests include SDN, future Internet, quality of Internet service models, USN protocols, wireless networkstensor networks, etc.

.