Video Quality Adaptation to Improve The Quality of Experience in DASH Environments

Ukheon Jeong and Kwangsue Chung,

Department of Communication Engineering, Kwangwoon University, Korea

Summary
Recently, DASH (Dynamic Adaptive Streaming over HTTP) is gaining attention because it is possible to use an existing web server, and not be restricted by the firewall or NAT (Network Address Translator). However, the existing video quality adaptation methods for DASH do not consider the frequent change of video quality, which degrades the QoE (Quality of Experience) of DASH service. In this paper, we propose an S-DASH (Smooth Dynamic Adaptive Streaming over DASH) scheme to improve the QoE of DASH service by avoiding frequent changes in the video quality. The proposed S-DASH scheme consists of a video quality adaptation algorithm, and a segment duration determination algorithm. The video quality adaptation algorithm controls the video quality based on MSD (Media Segment Duration), SFT (Segment Fetch Time), and the buffer state of the client to reduce the change of video quality and to prevent buffer underflow of the client. The segment duration determination algorithm determines the segment duration based on variation of chunk throughput to reduce the change of video quality and to improve the video quality. As a result, the proposed scheme can improve the QoE of DASH service. Through the simulation results, we prove that the proposed S-DASH scheme improves the QoE of DASH service by utilizing the smooth video quality adaptation algorithm and the segment duration determination algorithm.

Key words:
Multimedia streaming services, DASH, Video quality adaptation, QoE

1. Introduction

Recently, thanks to the development of network technologies and mobile devices, on-line multimedia streaming is becoming more and more popular [1]. Therefore, expectation to the QoE (Quality of Experience) of multimedia streaming is increasing [2, 3]. Traditionally, RTP (Real-time Transport Protocol)/UDP (User Datagram Protocol) is used as transport layer protocol in multimedia streaming. However, unlike RTP/UDP, HTTP (Hypertext Transfer Protocol) is easy to configure and is typically granted traversal of firewalls and NAT (Network Address Translator), which makes it attractive for multimedia streaming application. Well known implementations of the HTTP adaptive streaming are Microsoft Smooth Streaming, Apple HTTP Live Streaming, and Adobe Dynamic Streaming [4-6]. In this trend, a new standard called DASH (Dynamic Adaptive Streaming over HTTP) has been developed by MPEG and 3GPP to enable interoperability in the industry [7]. For seamless streaming, the DASH automatically switches video quality according to the current network condition. However, the quality adaptation algorithm of DASH does not consider the frequent change of video quality, which degrades the QoE of streaming service. Therefore, some researchers proposed the video quality adaptation schemes in DASH to reduce the frequent change of video quality.

Liu et al. [8] proposed a video quality adaptation algorithm based on SFT (Segment Fetch Time), called the RAHS (Rate Adaptation for Adaptive HTTP Streaming) scheme, to prevent client buffer underflow, and reduce the change of video quality. Thang et al. [9] proposed a video quality adaptation algorithm based on throughput estimation, called the ASAC (Adaptive Streaming of Audiovisual Content) scheme, to reduce the unnecessary change of video quality. Mok et al. [10] proposed a video quality adaptation algorithm based on stepwise decrease of video quality, called QDASH (QoE-aware DASH system), to improve the QoE of DASH service. Also, other researchers proposed various segment duration determination algorithms to reduce the frequent change of video quality. Liu et al. [11] proposed a variable segment duration algorithm based on the variance of the TCP congestion window, which is calculated by TCP congestion window expectation. Lievens et al. [12] proposed a variable segment duration algorithm based on different segment duration according to video quality. However, the existing schemes are still insufficient for consideration of the frequent change of video quality.

In this paper, we propose an S-DASH (Smooth Dynamic Adaptive Streaming over HTTP) scheme to reduce the change of video quality in DASH. The S-DASH scheme consists of a video quality adaptation algorithm and a segment duration determination algorithm. The video quality adaptation algorithm adapts the video quality based on MSD (Media Segment Duration), SFT (Segment Fetch Time), and the buffer state of the client, to reduce the change of video quality and to prevent buffer underflow of the client. The segment duration determination algorithm determines the segment duration based on variation of chunk throughput to reduce the change of video quality and to improve the video quality. As a result, the S-DASH scheme is able to improve the QoE of DASH service.

The rest of this paper is organized as follows. Section 2 presents a brief overview of the DASH and related works.

Manuscript received August 5, 2014
Manuscript revised August 20, 2014

The proposed S-DASH scheme is presented in Section 3. The simulation results of the proposed S-DASH scheme are provided in Section 4. Finally, Section 5 concludes the paper.

2. Related Work

In this section, we provide a review of the DASH and then describe the quality adaptation algorithms and the segment duration determination algorithms, for seamless and smooth streaming.

2.1 DASH (Dynamic Adaptive Streaming over HTTP)

Fig. 1 shows the system architecture of the DASH. The DASH system consists of a DASH server and a number of DASH clients. The media file in the DASH server is subdivided into segments, and each of the segments is provided at multiple bitrates. The HTTP Access module accesses the DASH server using HTTP/1.1, and requests the MPD (Media Presentation Description) for the information about media contents. The DASH server sends the MPD to the DASH client, in response to the request for the MPD. Then, the HTTP Streaming Control module selects an appropriate segment based on the information about media contents and network bandwidth. Finally, the Media Player plays the received segments.

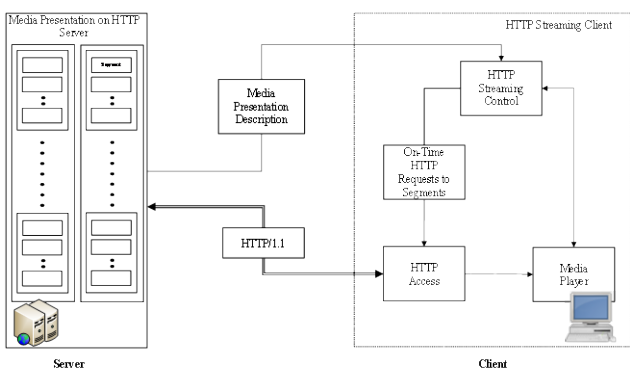


Fig. 1 System architecture of the DASH

2.2 Existing Scheduling Mechanism

A video quality adaptation scheme based only on the network condition in the DASH service leads to a frequent change of video quality. Therefore, many researchers proposed various video quality adaptation schemes in DASH to reduce the frequency of video quality changes.

Liu et al. [8] proposed a RAHS (Rate Adaptation for Adaptive HTTP Streaming) scheme, to prevent the client buffer underflow, and reduce the change of video quality. The RAHS scheme determines the network condition as

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

where SFT denotes a period of time from the time instant of sending a GET request for a media segment to the instant of receiving the last bit of the requested media segment and MSD denotes a media segment duration. If μ is larger than the maximum threshold, it means that the segment throughput is higher than the bitrate of the current quality. Then, the RAHS scheme selects the next higher video quality. On the other hand, if μ is smaller than the minimum threshold, it means that the segment throughput is lower than the bitrate of the current quality. Then, the RAHS scheme selects the video quality to match the current segment throughput. However, it still cannot reduce the frequent change of video quality.

Thang et al. [9] proposed an ASAC (Adaptive Streaming of Audiovisual Content) scheme, to reduce the unnecessary change of video quality. To reduce the change of video quality in short-term rate fluctuation, the ASAC scheme uses estimated throughput, as

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

where $T_e(i)$ and $T_s(i)$ denote the i-th estimated segment throughput and the i-th measured segment throughput, respectively, and δ denotes a smoothing factor, which is changed by the difference between the estimated segment throughput and the measured segment throughput. If δ is large, the measured segment throughput has a large effect on the current estimated segment throughput; and if δ is small, the past estimated segment throughput has a large effect on the current estimated segment throughput. As a result, the ASAC scheme reduces the unnecessary change of video quality in short-term rate fluctuation. However, it cannot reduce the frequent change of video quality in significant rate fluctuation.

Mok et al. [10] proposed a QDASH (QoE-aware DASH system), to improve the QoE of DASH service. The QDASH improves the QoE of DASH service by using stepwise decrease of video quality. However, it requires a measurement proxy to measure the throughput.

2.3 Segment Duration Determination Schemes

Long segment duration reduces the change of video quality. However, it leads to the client slowly responding to the network condition. As a result, the long segment duration may cause client buffer underflow. On the other hand, short segment duration leads to the client quickly responding to the network condition. However, it causes a frequent change of video quality. Therefore, it is important to determine an appropriate segment duration to the network condition and many researchers proposed segment duration determination algorithms to reduce the frequent change of video quality.

Liu et al. [11] proposed a variable segment duration algorithm based on the variance of the TCP congestion window, which is calculated by the TCP congestion window expectation. This algorithm calculates the optimal segment duration, using the variance of the TCP congestion window. It determines the segment duration at every segment request as

$$SD = RTT\,\frac{8+\frac{6p}{b}\sqrt{\frac{2b}{3p}}+6\sqrt{\frac{16}{9}-4bp\varepsilon+\frac{2p}{3b}\left(4\sqrt{\frac{2b}{3p}}+1\right)}}{9bp\varepsilon} \quad (3)$$

where SD and RTT denote the segment duration and round-trip time, respectively, b denotes the number of packets that are acknowledged by a received ACK, p denotes the packets loss rate, and ε denotes the upper limit of variance. However, it has to encode the segment with the optimal segment duration at every segment request. As a result, it has a lot of complexity in encoding.

Lievens et al. [12] proposed a variable segment duration algorithm based on different segment duration according to video quality. It sets long segment duration for high quality, and short segment duration for low quality. Therefore, the client quickly responds to the network condition, when the available bandwidth is low. However, it slowly responds to the network condition, if the current video quality is high.

3. S-DASH (Smooth DASH)

This section presents an advanced video quality adaptation algorithm in DASH. Our algorithm adapts the video quality, using SFT and MSD. Also, it determines the MSD using chunk throughput, to reduce the change of video quality. The streaming system architecture of our scheme is shown in Fig. 2. The S-DASH scheme is operated in client, and consists of segment throughput measurement, chunk throughput measurement, video quality adaptation, segment duration determination, and player.

In Fig. 2, the Segment Throughput Measurement module calculates the segment throughput, which is obtained by dividing a segment by a SFT. The Chunk Throughput Measurement module calculates the chunk throughput, which is obtained by dividing a chunk size by a chunk fetch time. The Video Quality Adaptation module adapts the video quality, based on the buffer state and the segment throughput. The Segment Duration Determination module determines the MSD, based on the variation of chunk throughput. The Player module plays the media content. Segments in the server are provided with multiple bitrates and various MSDs.

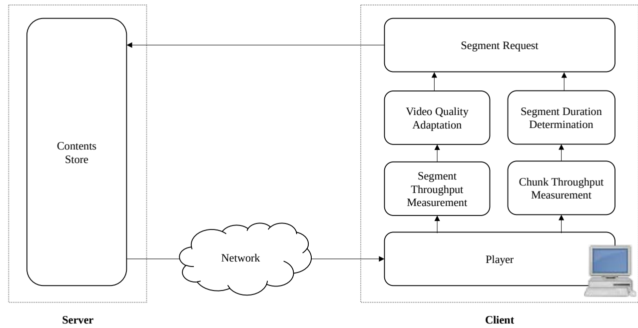


Fig. 2 Streaming system architecture based on the S-DASH scheme

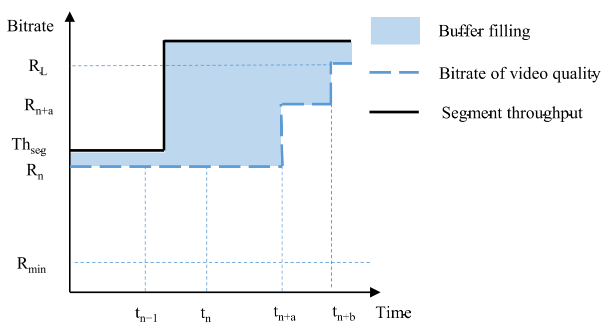
3.1 Overview of the AWS Mechanism

In this section, we propose a video quality adaptation algorithm based on MSD, SFT, and the buffer state of the client. A summary of the terms and a definition of the symbols appearing in Section 3.1 are shown in Table 1.

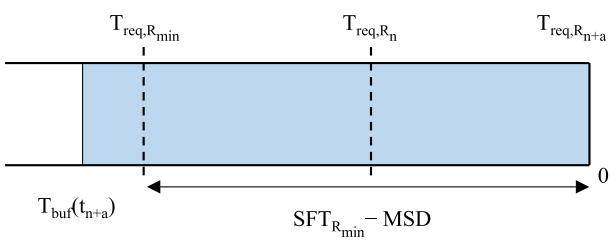
Table 1: Terms and definition of symbols

Notation	Definition
t_n	time when requesting an n-th segment
R_n	bitrate of video quality that is requested at t_n
R_{n+a}	bitrate of video quality that is requested at t_{n+a}
R_L	bitrate of the highest video quality that is lower than the TCP throughput
R_{min}	bitrate of the lowest quality among all video qualities
Th_{seg}	segment throughput
$T_{buf}(t_{n+a})$	time that video is able to be played using accumulated data in the buffer until t_{n+a}
$T_{req,R_{min}}$	required buffering time to prevent buffer underflow during an MSD, when segment throughput is reduced to R_{min}
T_{req,R_n}	required buffering time to prevent buffer underflow during an MSD, when segment throughput is reduced to R_n
$SFT_{R_{min}}$	time required to receive a segment when the segment throughput is reduced to R_{min}
$SFT_{T_{seg}}$	time required to receive a segment, according to segment throughput
S_{seg}	size of a segment
α	constant that determines the degree of degradation of video quality
H_{min}	minimum threshold to prevent buffer underflow

The video quality adaptation algorithm in the S-DASH scheme to reduce the frequent changes of video quality consists of switching-up, switching-down, or keeping a video quality unchanged. Switching-up to reduce the frequent change of video quality is shown in Fig. 3. Fig. 3(a) shows the increase of video quality, according to the increase of segment throughput. The solid line and dashed line indicate the segment throughput and the bitrate of video quality. Fig. 3(b) shows the buffered media time to prevent buffer underflow, when the bitrate of video quality is selected as R_{n+a}.



(a) Increase of video quality according to the increase of segment throughput



(b) Required buffering time to prevent buffer underflow when the video quality is R_{n+a}

Fig. 3 Switching-up in S-DASH

In Fig. 3(a), if the video quality increases only according to the increase of segment throughput, the bitrate of video quality is selected as R_L when the time is t_n. However, after the increase of video quality, if the segment throughput rapidly decreases, then the video quality is decreased below the segment throughput. Also, this can cause buffer underflow, because the client cannot change the video quality, while receiving the segment. To solve this problem, in the S-DASH scheme, the first condition for switching-up is that the segment throughput is higher than the bitrate of video quality; and the second condition for switching-up is that the client can prevent buffer underflow during an MSD, after the video quality increases. Therefore, switching-up occurs at t_{n+a} in Fig. 3(a).
The first condition for switching-up can be expressed as

$$Th_{seg} \geq R_{n+a} \tag{4}$$

The second condition for switching-up can be expressed as

$$T_{buf}(t_{n+a}) \geq SFT_{R_{min}} - MSD \tag{5}$$

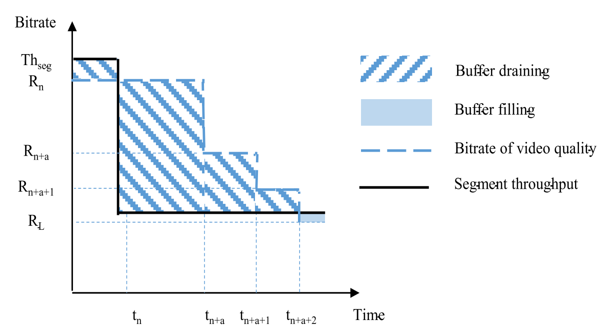
Since $SFT_{R_{min}}$ is equal to SFT when the bitrate of video quality is R_{n+a} and the segment throughput is R_{min}, Equation (5) can be expressed as

$$T_{buf}(t_{n+a}) \geq \frac{R_{n+a}}{R_{min}} \cdot MSD - MSD \tag{6}$$

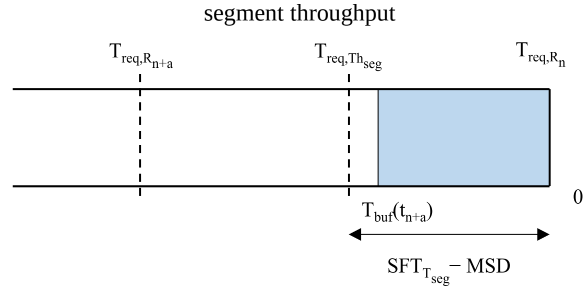
Finally, the bitrate of video quality to prevent buffer underflow can be expressed as Equation (7) using Equation (6).

$$R_{n+a} = \max\left\{R_j \leq \frac{R_{min}(T_{buf}(t_{n+a}) + MSD)}{MSD}\right\}, \quad \forall j = [1,2,\ldots,N] \tag{7}$$

The client in the S-DASH scheme repeatedly increases the bitrate of video quality, until the bitrate of video quality equals R_L.
Switching-down to reduce the frequent change of video quality and prevent buffer underflow is shown in Fig. 4. Fig. 4(a) shows the decrease of video quality, according to the decrease of segment throughput. Fig. 4(b) shows the required buffering time to prevent buffer underflow, when the bitrate of current video quality is R_n, and the segment throughput is rapidly decreased.



(a) Decrease of video quality, according to the decrease of segment throughput



(b) Required buffering time to prevent buffer underflow when video quality is R_n

Fig. 4 Switching-down in S-DASH

In Fig. 4(a), if the video quality decreases only according to the decrease of segment throughput, the bitrate of video quality is selected as R_L, when the time is t_n. However, after the decrease of video quality, if the segment throughput increases, then the video quality has to be increased. Therefore, it can cause a frequent change of video quality. To solve the above problem, in the S-DASH scheme, the first condition for switching-down is that the segment throughput is lower than the bitrate of current video quality; and the second condition for switching-down is that the client cannot prevent buffer underflow, during an MSD. Also, we perform the stepwise decrease of video quality, to improve the QoE of DASH service, because the rapidly decrease of video quality has a negative effect on the QoE service.
The first condition for switching-down can be expressed as

$$Th_{seg} < R_n \tag{8}$$

The second condition for switching-down can be expressed as

$$T_{buf}(t_{n+a}) \geq SFT_{Th_{seg}} - MSD \tag{9}$$

Since $SFT_{Th_{seg}}$ is equal to SFT, when the bitrate of video quality is R_n and the segment throughput is Th_{seg}, Equation (9) can be expressed as

$$T_{buf}(t_{n+a}) < \frac{R_n}{Th_{seg}} \cdot MSD - MSD \tag{10}$$

Also, we perform the stepwise decrease of the video quality to improve the QoE of DASH service as

$$R_{n+a} = \frac{R_n+R_L}{\alpha}, \qquad \frac{R_n+R_L}{R_n} < \alpha < \frac{R_n+R_L}{R_L} \tag{11}$$

In Equation (11), we set α higher than $\frac{R_n+R_L}{R_n}$ and lower than $\frac{R_n+R_L}{R_L}$, to set the bitrate of the next requested video quality to be higher than R_L, and lower than R_n. Therefore, we must calculate the buffer consumption in the client by considering the stepwise decrease of video quality. The buffer consumption in the client can be calculated by Equation (10), Equation (11) and an infinite series.

$$T_{buf}(t_{n+a}) < \left(1 + \frac{1}{\alpha} + \frac{1}{\alpha^2} \cdots\right) \cdot \left(\frac{R_n}{Th_{seg}} \cdot MSD - MSD\right) \tag{12}$$

$$T_{buf}(t_{n+a}) < \frac{\alpha}{\alpha-1} \cdot \left(\frac{R_n}{Th_{seg}} \cdot MSD - MSD\right) \tag{13}$$

The client in the S-DASH scheme repeatedly decreases the bitrate of video quality, until the bitrate of video quality equals R_L. If the buffered media time is lower than the minimum threshold, the bitrate of video quality is set lower than the segment throughput, to prevent buffer underflow.

3.2 Segment Duration Determination Algorithm of S-DASH

In this section, we describe the segment duration determination algorithm of the S-DASH scheme, to reduce the change of video quality. The S-DASH scheme determines the segment duration, based on variance of chunk throughput. In order to calculate the variance, we divide a segment by n, called chunk, and measure the chunk throughput. The measurement of chunk throughput is shown in Fig. 5.

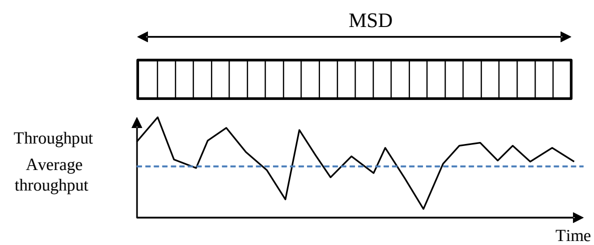


Fig. 5 Measurement of chunk throughput

To reduce the change of video quality, we calculate the variance of chunk throughput, based on n-chunk throughputs. We increase the segment duration, if the variance of chunk throughput is lower than the predefined threshold.

$$\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2 < \frac{R_c - R_{c-1}}{2} \tag{14}$$

where N denotes the number of chunks in a segment, x_i denotes the i-th chunk throughput, $\bar{x}$ denotes the average of chunk throughput, and R_c, R_{c-1} denote the bitrate of current video quality, and the bitrate of video quality is one step lower than the current video quality. In other words, if the variance of chunk throughput is lower than the bitrate interval of video quality, then the segment duration is increased by one step. On the other hand, if the variance of chunk throughput is higher than the bitrate interval of video quality, then the segment duration is decreased by one step, to respond quickly to the network condition.
If the segment duration is long, the change of video quality is reduced. However, the response to the network condition is slow. To solve this problem, we predict whether the buffer underflow will occur or not in a segment, every chunk. If the buffer underflow is predicted, we stop receiving the segment, and set the minimum segment duration and the bitrate of video quality to match the current segment throughput. Also, if the segment duration is long, the required buffered media time for

switching-up is large. As a result, the video quality is slowly increased. To solve this problem, we set the shortest segment duration when the video quality is changed. Fig. 6 shows the pseudocode of the S-DASH scheme.

Calculate x, $\bar{x}$
Calculate Th_{seg}
if($Th_{seg} \geq R_n$){
    if($T_{buf}(t_{n+a}) \geq \frac{R_{n+a}}{R_{min}} \cdot MSD - MSD$){
        $R_{n+a} = max\{R_j \leq \frac{R_{min}(T_{buf}(t_{n+a})+MSD[0])}{MSD[0]}\}$
        MSD=MSD[0]
    }
    Else{
        $R_{n+a} = R_n$
        If($\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2 < \frac{R_c - R_{c-1}}{2}$)
            Increase MSD by one-step
        Else
            Decrease MSD by one-step
    }
}
Else{
    if($T_{buf}(t_{n+a}) < \frac{\alpha}{\alpha-1}(\frac{R_n}{Th_{seg}} \cdot MSD - MSD)$){
        $R_{n+a} = max\{R_j \leq \frac{R_n + R_L}{\alpha}\}$
        if($T_{buf}(t_{n+a}) < H_{min}$){
            MSD = MSD[0]
            $R_{n+a} = R_L$
        }
    }
    Else
        $R_{n+a} = R_n$
}

Fig. 6 Pseudocode of the S-DASH scheme

4. Simulation

In order to evaluate the proposed scheme, we implemented the S-DASH scheme in NS-2(Network Simulator) [13]. All segments have durations of 2s, 4s, or 8s, and the length of media presentation is 100s. A server provides 25 different video rates, ranging from 100kbps to 2500kbps, with rate gap between two adjacent versions of 100kbps, using JSVM [14]. Fig. 7 shows the network topology used in the simulations. To generate the cross traffic, we inject background UDP traffic between the server and the client. We compared the S-DASH scheme with the RAHS scheme and the ASAS scheme.

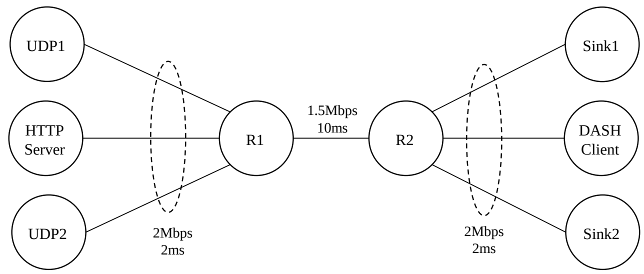


Fig. 7 Network topology

Two different scenarios have been considered in order to investigate the dynamic behavior of the S-DASH scheme: 1) in long-term rate fluctuation of the UDP traffic; 2) in short-term rate fluctuation of the UDP traffic.

4.1 Performance in Long-Term Rate Fluctuation

In this section, we evaluate the performance of the S-DASH scheme in long-term rate fluctuation. In order to make the simulation environments, we generate the UDP flow 1, with 1000kbps from 0s to 15s and from 40s to 100s. Also, we generate the UDP flow 2, with 300kbps from 70s to 100s. Fig. 8 shows the change of video quality according to the long-term rate fluctuation. In Fig. 8, we can see that the change of video quality in the S-DASH scheme is reduced compared to the other schemes.

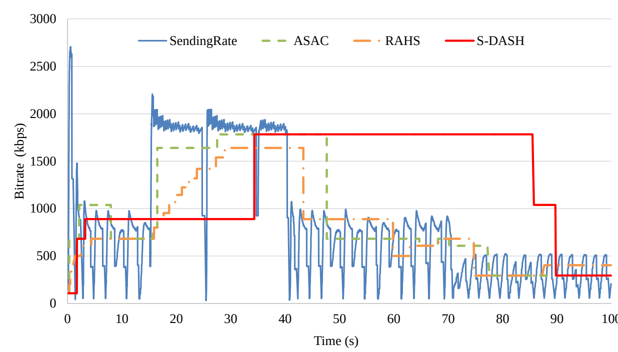


Fig. 8 Changes of video quality according to the long-term rate fluctuation

To objectively assess Fig. 8, we used the WAQT (Weighted Average Quality Transition) [15]. The WAQT indicates the number of changes of video quality. Therefore, if the WAQT is large, the number of changes of video quality is large. In Table 2, we can see that the change of video quality in S-DASH is reduced, compared with the other schemes.

Table 2: Comparison of WAQT according to long-term rate fluctuation

	ASAC	RAHS	S-DASH
WAQT	11	22	5

Also, we measured the average PSNR for video quality evaluation. In Table 3, we can see that the average PSNR of the S-DASH scheme is higher than that of the RAHS scheme, and is not significant lower than that of the ASAC scheme. Therefore, the S-DASH scheme improves the QoE of DASH, because WAQT is small while the average PSNR of the S-DASH scheme is not significantly decreased.

Table 3: Comparison of average PSNR according to long-term rate fluctuation

	ASAC	RAHS	S-DASH
Average PSNR	49.76	45.30	48.72

Finally, we compared the S-DASH scheme with segment duration determination algorithm to the S-DASH scheme without segment duration determination algorithm. The effect of the segment duration determination algorithm in long-term rate fluctuation is shown in Table 4. When we used the segment duration determination algorithm, the S-DASH scheme improved the video quality by increasing the average PSNR and decreasing the WAQT. Also, it reduced data overhead by decreasing the number of segment requests.

Table 4: Effect of the segment duration determination algorithm in long-term rate fluctuation

	Without segment duration determination algorithm	With segment duration determination algorithm
Average PSNR	48.23	48.72
WAQT	8	5
The number of segment requests	50	24

4.2 Performance in Short-Term Rate Fluctuation

In this section, we evaluate the performance of the S-DASH scheme in short-term rate fluctuation. In order to make the simulation environments, we generate the UDP flow 1, and terminate the UDP flow 1 every 20s. Fig. 9 shows the change of video quality according to the short-term rate fluctuation. In Fig. 9, we can see that the change of video quality in the S-DASH scheme is reduced compared to the other schemes.

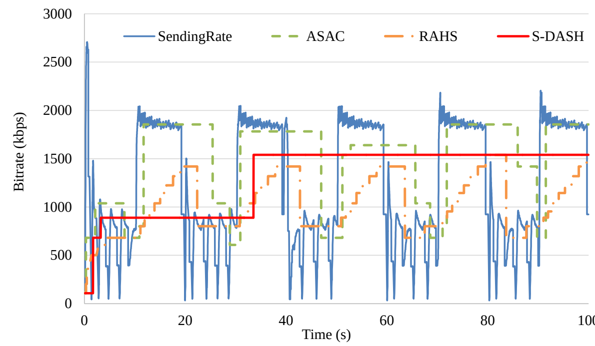


Fig. 9 Changes of video quality according to the short-term rate fluctuation

In Table 5, we can see that the change of video quality in the S-DASH scheme is reduced, compared to the other schemes.

Table 5: Comparison of WAQT, according to short-term rate fluctuation

	ASAC	RAHS	S-DASH
WAQT	18	51	3

Also, we measured the average PSNR for video quality evaluation. In Table 6, we can see that the average PSNR of the S-DASH scheme is higher than that of the RAHS scheme, and is not significantly lower than that of the ASAC scheme. Therefore, the S-DASH improved the QoE of DASH service, because the WAQT is small while the average PSNR of the S-DASH scheme is not significantly decreased.

Table 6: Comparison of average PSNR, according to short-term rate fluctuation

	ASAC	RAHS	S-DASH
Average PSNR	49.76	45.30	48.72

Finally, the effect of the segment duration determination algorithm in long-term rate fluctuation is shown in Table 7. When we used the segment duration determination algorithm, the S-DASH scheme improved the video quality by increasing the average PSNR, and by decreasing the WAQT. Also, it reduced data overhead by decreasing the number of segment requests.

5. Conclusion

In this paper, we present a novel video quality adaptation scheme, called S-DASH (Smooth DASH), to improve the QoE of DASH service. Our scheme consists of a video quality adaptation algorithm, and a segment duration determination algorithm. By controlling video quality based

Table 7: Effect of the segment duration determination algorithm in short-term rate fluctuation

	Without segment duration determination algorithm	With segment duration determination algorithm
Average PSNR	49.65	49.99
WAQT	4	3
The number of segment requests	59	21

on the segment throughput and the client buffer state, the video quality adaptation algorithm in the S-DASH scheme reduces the frequent change of video quality, and the significant decrease of video quality. Also, by controlling segment duration based on the variance of chunk throughput, the segment duration determination algorithm in the S-DASH scheme reduces the frequent change of video quality and the unnecessary overhead. As a result, we can see that the S-DASH scheme improves the QoE of the DASH service. Through the simulation results, we prove that the proposed S-DASH scheme improves the QoE of DASH service by using the smooth video quality adaptation and the segment duration determination algorithm.

Acknowledgments

This work was supported by the ICT R&D program of MSIP/IITP. [14-000-11-002, Development of Object-based Knowledge Convergence Service Platform using Image Recognition in Broadcasting Contents]

References

[1] 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, March 2011.

[2] K. Kilkki, “Quality of Experience in Communications Ecosystem,” Journal of Universal Computer Science, vol. 14, no. 5, pp. 615-624, March 2008.

[3] N. Bouten, J. Famaey, S. Latre, R. Huysegems, B. Vleeschauwer, W. Leekwijck, and F. Turck, “QoE Optimization Through In-network Quality Adaptation for HTTP Adaptive Streaming,” in Proc. of International Conference and workshop on Systems Virtualization Management, pp. 336-342, October 2012.

[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/httpdynamic-streaming/

[7] T. Stockhammer, “Dynamic Adaptive Streaming over HTTP: Standards and Design Principles,” in Proc. of ACM MMSys'11, pp. 133-144, February 2011.

[8] C. Liu, I. Bouazizi, and M. Gabbouj, “Rate Adaptation for Adaptive HTTP Streaming,” in Proc. of ACM MMSys'11, pp. 169-174, February 2011.

[9] T. Thang, Q. Ho, J. Kang, and A. Pham, “Adaptive Streaming of Audiovisual Content using MPEG DASH,” IEEE Trans. on Consumer Electronics, vol. 58, no. 1, pp. 78-85, February 2012.

[10] R. P. Mok, X. Luo, E. Chan, and R. Chang, “QDASH: A QoE-aware DASH System,” in Proc. of ACM MMSys'12, pp. 11-22, February 2012.

[11] C. Liu, I. Bouazizi, and M. Gabbouj, “Segment Duration for Rate Adaptation of Adaptive HTTP Streaming,” in Proc. of IEEE International Conference on Multimedia and Expo, pp. 1-4, July 2011.

[12] J. Lievens, S. Satti, N. Deligiannis, P. Schelkens, and A. Munteanu, “Optimized Segmentation of H.264/AVC Video for HTTP Adaptive Streaming,” in Proc. of IFIP/IEEE International Symposium on Integrated Network Management, pp. 1312-1317, May 2013.

[13] The Network Simulator NS-2, [Online]. Available: http://www. isi.edu/nsnam/ns/

[14] Joint Video Team(JVT) of ISO/IEC MPEG and ITU-T VCEG, “Joint Scalable Video Model JSVM-9,” JVT-V202, January 2007.

[15] T. Kim and M. H. Ammar, “Optimal Quality Adaptation for Scalable Encoded Video,” IEEE Journal on Selected Areas in Communications, vol. 23, no. 2, pp. 344-356, February 2005.

Ukheon Jeong received the B.S. and M.S. degrees in Communications Engineering from Kwangwoon University in 2012 and 2014, respectively. His research interests include communication networks, protocols, and QoS mechanism. He is now with LG Electronics Co., Seoul, Korea.

Kwangsue Chung received his B.S. degree from Hanyang University, Seoul, Korea, M.S. degree from KAIST (Korea Advanced Institute of Science and Technology), Seoul, Korea, and Ph.D. degree from the University of Florida, Gainesville, Florida, USA, all from the Electrical Engineering Departments. Before joining 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.