BANQUET: Balancing Quality of Experience and Traffic Volume in Adaptive Video Streaming

Takuto Kimura*, Tatsuaki Kimura†, Arifumi Matsumoto*, Jun Okamoto*
*NTT Network Technology Laboratories, NTT Corporation,
3–9–11 Midori–cho, Musashino–shi, Tokyo 180–8585 Japan
†Department of Information and Communications Technology, Graduate School of Engineering, Osaka University
2–1 Yamadaoka, Suita–shi, Osaka 565–0871 Japan
Email: {takuto.kimura.mx, arifumi.matsumoto.hf, jun.okamoto.nw}@hco.ntt.co.jp, kimura@comm.eng.osaka-u.ac.jp

***Abstract*—Bitrate-selection algorithms are key to improving the quality of experience (QoE) of adaptive video streaming. Although current bitrate selection algorithms maximize the QoE, video consumers are concerned with QoE and traffic-volume usage due to the pay-per-use or data-capped plans. To balance between the QoE and traffic volume, some commercial video-streaming services enable users to set the upper limit of the selectable bitrate. However, it is difficult for users to set an appropriate limit to obtain sufficient QoE. We propose BANQUET, a novel bitrate-selection algorithm that enables users to control intuitively the balance between the QoE and traffic volume. Assuming a user-set *target QoE* as a balancing parameter, BANQUET selects the bitrate that minimizes the traffic volume while maintaining the estimated mean opinion score (MOS) above the target QoE. BANQUET calculates the appropriate bitrate based on estimations of the throughput and buffer transition. A trace-based simulation shows that BANQUET reduces the traffic volume by up to 47.0% compared to a baseline while maintaining the same average estimated MOS.**

***Index Terms*—Adaptive Bitrate, Bitrate Selection Algorithm, QoE, Traffic Volume Reduction**

I. Introduction

Video traffic has recently become the dominant type of Internet traffic. The video-traffic volume is expected to grow four fold from 2017 to 2022 and account for 82% of all traffic in 2022 [1]. Unlike web services, video streaming services demand high and stable throughput, and the video quality of experience (QoE) may deteriorate if the communication quality degrades. Studies [2], [3] have shown that users abandon watching videos if the QoE significantly degrades. Therefore, users demand better QoE to enjoy watching videos, and video-content providers are struggling to improve the QoE.

To improve the QoE, adaptive bitrate streaming (ABR) is widely used in current video-streaming services. Typically, an ABR system consists of a video-streaming server that stores the video data in chunks and a client. These chunks that comprise several seconds of video are pre-encoded into multiple bitrates before streaming. The client requests a chunk with a specified bitrate that can be selected using bitrate selection algorithm and sequentially plays the received chunks. If the algorithm always selects the highest bitrate, this may cause rebuffering and degrade the QoE. Conversely, if the lowest bitrate is always selected, the QoE degrades due to poor image quality even though there is no rebuffering. Therefore, the bitrate-selection algorithm is key to improving the QoE.

Video consumers are also concerned with traffic-volume usage because many users regardless of being a mobile or fixed-line user use a pay-per-use plan *e.g.* $10 per 1 GB, or a data capped plan *e.g.* 7 GB per month. In fact, a recent report on mobile video watching in North America showed that data usage is the most frustrating aspect [4]. Selecting the lowest bitrate is a simple solution to reducing the traffic volume, but the QoE may also degrade to below a sufficient QoE level. To balance the QoE and traffic volume, some commercial video streaming services such as YouTube [5] implement a functionality enabling users to set the upper limit for the selectable bitrate. However, it is difficult for users to set the appropriate limit to obtain a sufficient QoE. Even if users know the sufficient QoE level to set, selecting the appropriate bitrate is also difficult because there are several bitrate series that achieve the same QoE while differing greatly in terms of traffic volume. This difference is caused by the fact that the QoE is affected by the selected bitrate and rebuffering. For example, if there is a bitrate series with a high average bitrate and rebuffering, and a bitrate series with a low average bitrate and no rebuffering, the QoE can be the same while the traffic volume can be less for the latter. Therefore, we need to construct a bitrate selection algorithm that can (1) control the balance between the QoE and traffic volume using an intuitive balancing parameter and (2) achieve a better QoE with a lower traffic volume.

Due to their importance, many bitrate-selection algorithms have been proposed, in which the bitrate is selected based on the buffer length [6], model prediction control [7], proportional-integral-differential control [8], automatic ABR parameter tuning [9], and deep reinforcement learning [10]. The objective for these algorithms is the same *i.e.* selecting the highest bitrate while reducing rebuffering and bitrate switching. Therefore, current algorithms do not consider the traffic volume aspect.

We propose a **BA**la**N**cing **QU**ality of **E**xperience and **T**raffic volume algorithm, called BANQUET. BANQUET is a novel bitrate-selection algorithm that enables users to control intuitively the balance of the QoE and traffic

978-3-903176-24-9

volume. Assuming a user-set *target QoE* as a balancing parameter, BANQUET selects the bitrate that minimizes the traffic volume while maintaining the QoE above the target. As a QoE metric, many bitrate-selection algorithms [6]–[10] use *utility functions*, which are certain weighted linear sums of several performance metrics such as the average bitrate, bitrate variation, rebuffering time, and the number of rebuffering events. However, the output of these utility functions does not always directly reflect the QoE because they are not constructed to express the QoE. Therefore, we adopt the mean opinion score (MOS) [11] as a QoE metric. The MOS directly corresponds to the QoE because the MOS-estimation model we use is constructed through subjective quality tests [12]–[14]. Therefore, by using a target MOS as the balancing parameter, users can set it more intuitively than when using the output of the utility functions. On the basis of the target MOS, BANQUET automatically selects a bitrate that reduces the traffic volume while maintaining the target QoE by estimating the throughput and buffer transition.

We evaluate BANQUET through a trace-based simulation using actual 4G LTE throughput data [15] and compare BANQUET to a baseline algorithm adopted in commercial streaming services that has an upper limit for selectable bitrates. The simulation results show that BANQUET reduces the traffic volume by up to 47.0% compared to the baseline algorithm while maintaining the same average estimated MOS.

The remainder of this paper is constructed as follows. First, we describe related work and limitations in Section II. We then give details on BANQUET in Section III. We evaluate BANQUET through a trace-based simulation in Section IV. Finally, we conclude the paper and describe future work in Section V.

II. Related work

A. Bitrate-selection Algorithm

Many bitrate-selection algorithms have been proposed to improve QoE based on the buffer length [6], model-prediction control [7], proportional-integral-differential control [8], automatic ABR parameter tuning [9], and deep reinforcement learning [10]. These algorithms improve the output of utility functions, which are certain weighted linear sums of several performance metrics *e.g.* the average bitrate, bitrate variation, rebuffering time, and number of rebuffering events. Since these algorithms do not consider traffic volume, the volume tends to be high because they attempt to select the highest bitrate that does not cause rebuffering.

To reduce the traffic volume while maintaining the QoE, the statistically indifferent quality variation (SIQV) [16] approach is used to remove a higher bitrate from selectable bitrates if no statistical video quality difference exists between adjacent bitrates. SIQV reduces the traffic volume while maintaining the video quality, but if users do not need high quality, the video player may still select a high bitrate and generate undesirable traffic. Another approach generally used in commercial video-streaming services is to set an upper limit for the selectable bitrates. For example, YouTube [5] implements a function that enables users to select the upper bitrate limit. This reduces traffic volume, but it is difficult for users to set the upper bitrate limit because they do not know the appropriate limit to obtain a sufficient QoE.

B. QoE Estimation

Many studies [6]–[10] on bitrate-selection algorithms use utility functions to represent the QoE. However, these functions are artificially constructed and not constructed through subjective assessment tests. Therefore, they are not appropriate to estimate the QoE.

A MOS-estimation model is constructed through subjective assessment tests. The MOS is an index generally used as a representative QoE indicator [11]. Based on subjective test results, objective MOS-estimation models for ABR-based video streaming have been constructed and standardized [12]–[14]. These models calculate the estimated MOS using measurable indices such as selected bitrate series, number of rebuffering events, and rebuffering time. These models are constructed using an absolute category rating method with a five-grade quality scale (5: Excellent, 4: Good, 3: Fair, 2: Poor, 1: Bad) [17]. Thus, the output of the MOS-estimation model directly reflects the QoE. BANQUET uses the MOS-estimation model to calculate the bitrate. Note that this model estimates the average score without taking into account the difference among contents or users.

III. Proposed Bitrate-selection Algorithm

A. Goals and Approach

The goals for BANQUET are (1) to control the balance between the QoE and traffic volume using an intuitive balancing parameter and (2) to achieve a better QoE with a lower traffic volume. One simple approach to achieve these goals is to maximize a utility function represented as a weighted linear sum of the QoE and traffic volume. However, this approach may result in an unacceptably low QoE in exchange for achieving a lower traffic volume. This is because maximizing the utility function does not necessarily guarantee the QoE. Therefore, assuming that a *target QoE* is given as a balancing parameter, BANQUET selects the bitrate that minimizes the traffic volume while maintaining the QoE above the target. The target QoE is set by a user considering the content type, contract plan, or viewing style. Since the policy for setting the target QoE is independent from BANQUET, a service provider can also set the target QoE considering the service-management policy such as better QoE for premium users.

B. Selecting QoE Metric

BANQUET adopts the MOS as a QoE metric and not the utility functions used in current bitrate-selection algorithms. There are two reasons the MOS is adopted. The first is that the MOS more directly reflects the actual QoE. As described in Section II-B, the utility function output does not represent the actual QoE, while the MOS-estimation model is constructed based on the subjective evaluation tests. The

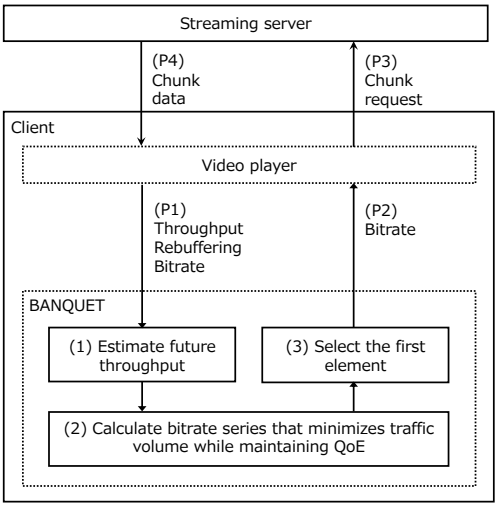

Fig. 1. Overview of video-streaming system using BANQUET.

TABLE I
NOTATIONS

Notation	Meaning
t	Clock time in simulation (s)
b_t	Buffer length at t (s)
$\mathbf{r}$	Candidate bitrate series of solution
R	Set of candidate bitrate series
r_i	Average bitrate of the i-th chunk (bps)
l_i	Chunk length of the i-th chunk (s)
$\tilde{c}_t$	Estimated throughput at t (bps)
y_t	Bits that cannot be downloaded at t
n_t	Number of chunks already started downloading
$g_{i,t}$	Sum of downloaded bits by t at the i-th chunk
$h_{i,t}$	Buffer length at t when the i-th chunk is downloaded not considering buffer consumption (s)
h	Throughput prediction horizon (s)
q_t	MOS at t
T_{high}	Upper limit of buffer (s)
T_q	Target MOS
T_b	Threshold of buffer (s)
T_d	Threshold of length of bitrate-series candidates

second is that the target value can be set more intuitively than when using the utility functions. It is difficult to give an intuitive interpretation to the utility-function output value. On the other hand, the MOS-estimation model output directly corresponds to the subjective rated score on a 1–5 scale [17]. Therefore, the MOS facilitates the setting of the target QoE compared to the utility functions.

C. Overview of BANQUET

Figure 1 shows a video-streaming system using BANQUET, which is implemented inside the video-player application. When the player needs to receive the next chunk, it sends a bitrate-calculation request to BANQUET attached with the already selected bitrates, rebuffering, and measured throughput information (P1). Based on this information, BANQUET calculates the appropriate bitrate and notifies the player (P2). Then, the player requests the chunk with the specified bitrate (P3) and receives the chunk with the appropriate bitrate (P4). This procedure is performed every time the player requests a chunk. In this procedure, BANQUET calculates the bitrate in three steps. The notations are summarized in Table I.

First, BANQUET estimates a future throughput series sequentially from t_0 to $t_0 + h$ where t_0 and h denote the calculation start time and throughput prediction horizon, respectively. Future throughput values are estimated using the harmonic mean of the last five measured throughput values sequentially. We adopt this method for its robustness against outliers and use in previous studies [7], [8]. The measured throughput value is calculated for each chunk by dividing the number of bits by the chunk download time. The estimated throughput series from t_0 to $t_0 + h$ is denoted as $\{\tilde{c}_{t_0}, \tilde{c}_{t_0+1}, \cdots, \tilde{c}_{t_0+h}\}$.

Second, BANQUET calculates the bitrate series that minimizes the traffic volume while maintaining a MOS higher than the target MOS, T_q. The T_q can be set by a user in the player settings. To avoid risk of rebuffering, we add a constraint such that the buffer length at t_0+h must be greater than T_b where T_b is the buffer threshold. The details for this step are given in Section III-D. If no bitrate series satisfies the constraints, BANQUET selects the bitrate series that achieves the highest MOS while satisfying the buffer constraint. Note that we assume zippy [18] as the pacing method so that chunks are always requested unless the buffer length reaches T_{high}.

Third, BANQUET selects and outputs the first element of the calculated bitrate series.

D. Details of Second Step of BANQUET

Algorithm details for calculating an appropriate bitrate series are described here. To calculate the appropriate bitrate series, we must calculate the MOS, buffer length, and traffic volume at $t_0 + h$ for each candidate series. BANQUET estimates the buffer transition for each series, and derives these three elements.

Algorithm 1 is a pseudo code that calculates the appropriate bitrate series. Lines 1 to 3 correspond to initialization steps. In line 1, candidate solution $\hat{\mathbf{r}} = \{r_i\}$ is initialized with the already selected bitrate series for $1 \leq i \leq n_{t_0} - 1$ and r_{max} for $n_{t_0} \leq i \leq n_{t_0} + T_d$. Term i corresponds to the chunk index and r_{max} is the highest bitrate among the selectable bitrates. Then, set of candidate bitrate series $\mathcal{R}$ is initialized with all the bitrate patterns up to the next T_d chunks in line 3. Threshold T_d limits the length of the bitrate series candidates. The candidate bitrate series are initialized with the already selected bitrate series for $1 \leq i \leq n_{t_0}$ and all the bitrate patterns for $n_{t_0} \leq i \leq n_{t_0} + T_d$. In lines 5 to 12, the buffer transition for each $\mathbf{r} \in \mathcal{R}$ is estimated. More specifically, BANQUET calculates buffer length b_t by updating n_t and y_t, which are the number of chunks that has already started downloading and the number of bits that cannot be downloaded at time t, respectively. We give the details on update procedures (line 7) of b_t, n_t, and y_t later. In line 13, BANQUET calculates MOS q_t based on candidate bitrate series $\mathbf{r}$ and the buffer transition information estimated

Algorithm 1 Bitrate-series calculation algorithm

Require: $\{\tilde{c}_{t_0}, \tilde{c}_{t_0+1}, \cdots, \tilde{c}_{t_0+h}\}$, T_q, T_d, T_b
1: Initialize $\hat{r}_i$
2: $t \leftarrow t_0$
3: $\mathcal{R} \leftarrow$ all the bitrate patterns up to the next T_d chunks
4: **for each** $\mathbf{r} \in \mathcal{R}$ **do**
5: **while** $t \leq t_0 + h$ **do**
6: **if** $b_{t-1} < T_{high}$ **then**
7: Update b_t, n_t, y_t
8: **else**
9: $b_t \leftarrow b_{t-1} - 1$
10: **end if**
11: $t \leftarrow t + 1$
12: **end while**
13: Calculate q_t
14: **if** $(q_t \geq T_q) \wedge (b_t \geq T_b) \wedge (\sum r_i l_i \leq \sum \hat{r}_i l_i)$ **then**
15: $\hat{\mathbf{r}} \leftarrow \mathbf{r}$
16: **end if**
17: **end for**
Ensure: $\hat{\mathbf{r}}$

in the previous step. From lines 14 to 17, BANQUET updates the best candidate, $\hat{\mathbf{r}}$, with the lowest traffic volume that satisfies the MOS and buffer length constraints. Since this algorithm is constructed based on a brute-force search, the calculation time may drastically increase. Thus, we evaluate the calculation time and consider its reduction in Section IV-C.

We explain updating b_t, n_t, and y_t in line 7. First, we calculate n_t. If the chunk that was being received from the previous time slot was not completely received, $n_t = n_{t-1}$. Otherwise, the player receives as many chunks as possible unless the buffer length exceeds T_{high}. Therefore, n_t is calculated as

$$n_t = \begin{cases} n_{t-1} & y_{t-1} \geq \tilde{c}_t \\ \max\{m \mid g_{m-1,t} < \tilde{c}_t, h_{m-1,t} \leq T_{high}\} & y_{t-1} < \hat{c}_t \end{cases} \quad (1)$$

where $g_{i,t}$ and $h_{i,t}$ indicate the number of received bits and buffer length if the player receives the i-th chunk in this time slot, respectively. Thus, $g_{i,t}$ and $h_{i,t}$ are respectively formulated as

$$g_{i,t} = y_{t-1} + \sum_{k=n_{t-1}+1}^{i} r_k l_k \quad (2)$$

$$h_{i,t} = b_{t-1} + \sum_{k=n_{t-1}+1}^{i} l_k, \quad (3)$$

where l_k denotes the length of the k-th chunk.

Next, we calculate y_t. If the buffer length reaches T_{high} while receiving chunks, $y_t = 0$. Otherwise, y_t is calculated as the difference between the received bits in this time slot and the network bandwidth. Thus, y_t is formulated as

$$y_t = \max(g_{n_t,t} - \tilde{c}_t, 0). \quad (4)$$

Using $h_{n_t,i}$, we derive b_t on the basis of the decrements in the buffer. Term b_t is formulated as

$$b_t = \begin{cases} h_{n_t,i} & \text{playback is not started} \\ h_{n_t,i} - 1 & \text{playback is started.} \end{cases} \quad (5)$$

Finally, we explain the calculation method of q_t in line 13. This calculation is based on the objective MOS estimation model proposed by Yamagishi [14]. Since we focus on the bitrate-selection algorithm in this paper, we only give an overview of this model. In short, this model outputs the estimated MOS from the video bitrate, audio bitrate, video resolution, and rebuffering information. This model is closed-form and includes 17 parameters that depend on the client type or the range of network quality.

IV. Evaluation

A. Simulation Settings

We evaluated BANQUET using a trace-based simulation employing actual 4G LTE throughput data [15] collected from two mobile operators across different mobility patterns *i.e.* static, pedestrian, car, tram, and train. The data included the average throughput records at a granularity of one sample per second. We divided each trace into 300-s intervals and obtained 427 traces. To avoid trivial results, we excluded traces with the average throughput of 0.3 Mbps or lower that represent throughput series that were excessively low and not suitable for video viewing.

The video bitrate was set based on a commercial video-streaming service for a realistic setting. We watched approximately 100 videos on YouTube [5] and recorded the bitrate for each resolution (240p, 360p, 480p, 720p, and 1080p) for each video. We then calculated the average bitrate for each resolution and set the results as the video bitrate. The details of the video parameters are given in Table II. The audio bitrate was fixed at 128 kbps. The chunk and video lengths were fixed at 5 s and 180 s, respectively.

The BANQUET parameters were set to $T_b = 5$ s, $h = 30$ s, and $T_d = 5$. We set T_{high} and the playback-starting threshold to 20 s and 5 s as the simulated-player parameters, respectively. To determine the MOS-estimation model coefficients, we used a smartphone to conduct a subjective assessment test. Except for the client terminal, the test settings were the same as those in [12].

We compared BANQUET to a bitrate selection algorithm used in commercial video-streaming services as the *baseline*. The *baseline* selected the highest bitrate among the selectable bitrates, which was α times less than the estimated throughput. Term α is a safety margin to avoid rebuffering and was set to 0.9. The selectable bitrate has an upper limit set to the bitrate corresponding to 240p, 360p, 480p, 720p, and 1080p, as shown in Table II. Depending on the upper limit, we refer to the *baseline* as "*baseline* (240p)", "*baseline* (360p)", and so on. For example, we can select 253 kbps or 501 kbps for *baseline* (360p). To perform the evaluation under fair conditions, the *baseline* estimated the throughput using harmonic mean, which is also used in BANQUET.

TABLE II
VIDEO-BITRATE PARAMETERS OF VIDEO ENCODING

Resolution	Bitrate (kbps)	Framerate (fps)
240p	253	30
360p	501	30
480p	961	30
720p	1771	30
1080p	3352	30

B. BANQUET vs. Baseline

Figure 2 shows the relationship between the average of the estimated MOS and the traffic volume. The estimated MOS is calculated using the objective MOS-estimation model described in Section IV-A. Note that the estimated MOS may differ from T_q due to throughput fluctuation. The average estimated MOS is calculated by averaging the estimated MOS for each throughput series described in Section IV-A. The blue line signifies the MOS results of BANQUET when T_q is changed from 1.0 to 5.0 in increments of 0.1. The red plots represent the results of the *baseline* when the upper limit is changed from 240p to 1080p. The figure shows that BANQUET significantly reduces the traffic volume compared to the *baseline* while maintaining the same average estimated MOS. For example, if we focus on the average estimated MOS of BANQUET, which is almost the same as that of *baseline* (1080p), BANQUET reduces the traffic volume by 47.0% compared to that for *baseline* (1080p). For the traffic volume of BANQUET, which is also almost the same as *baseline* (1080p), BANQUET increases the average estimated MOS by 0.255 points compared to that for *baseline* (1080p).

To clarify why BANQUET reduces the traffic volume while maintaining the same average estimated MOS, we focus on the difference in estimated MOS distributions. Figure 3 shows the estimated MOS cumulative distribution functions (CDFs) for BANQUET and *baseline* (1080p). To compare them under fair conditions, we focus on when T_q is 3.9, which results in almost the same average estimated MOS as that for *baseline* (1080p). We find that BANQUET concentrates the estimated MOS onto T_q for approximately 70% of the results while that for the *baseline* does not. Such a concentrated distribution is advantageous in terms of reducing the traffic volume because in general, increasing the QoE requires the large traffic volume. More specifically, increasing the QoE from a higher QoE requires more traffic than when increasing the QoE from a lower QoE for the same increase in QoE. Thus, suppressing the estimated MOS when above T_q reduces the traffic volume drastically, and increasing the estimated MOS below T_q maintains the average estimated MOS. These two factors contribute to reducing the traffic volume while maintaining the average estimated MOS.

Next, we compare the traffic volume when the estimated MOSs are the same. Figure 4 shows the average traffic volume per view for each estimated MOS. The T_q and upper limit for the *baseline* are the same as in Fig. 3. We find that when the estimated MOS ranges from 1.5 to 2.5, the

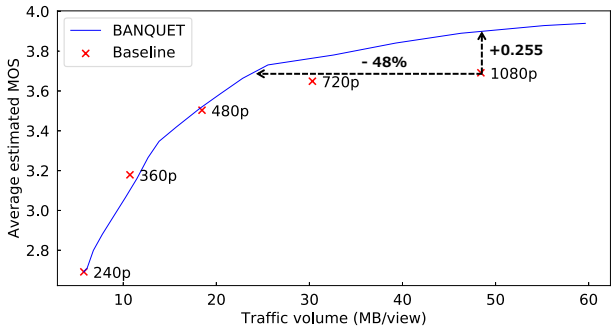

Fig. 2. Relationship between average estimated MOS and average traffic volume of BANQUET and *baseline*. Scatter plots represent results of *baseline*. Line plot represents results of BANQUET when T_q is changed from 1.0 to 5.0.

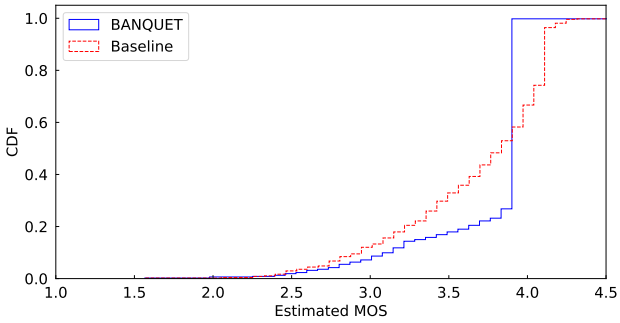

Fig. 3. Estimated MOS CDFs of BANQUET ($T_q = 3.9$) and *baseline* (1080p).

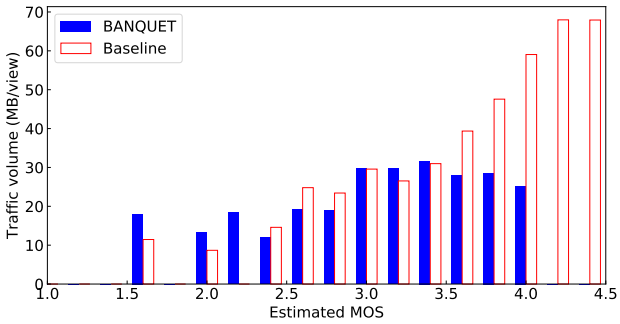

Fig. 4. Average traffic volume of BANQUET ($T_q = 3.9$) and *baseline* (1080p) for each estimated MOS.

BANQUET traffic volume is higher than that for the *baseline* because BANQUET selects a higher bitrate than the *baseline* to achieve the target QoE causing rebuffering. However, the overall effect of this traffic increase is very small because the number of views in this range is also very small, as shown in Fig. 3. When the estimated MOS ranges from 2.5 to 3.5, the traffic volume is almost the same for BANQUET and the *baseline* because the throughput series is relatively low to select the higher bitrate for BANQUET and the *baseline*. In contrast, BANQUET reduces the traffic volume more than the *baseline* when the estimated MOS ranges from 3.5 to 3.9, at which many views are concentrated. To clarify the

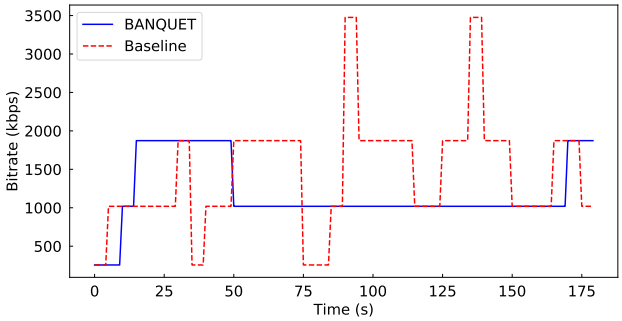

Fig. 5. Sample bitrate series of BANQUET and *baseline* when estimated MOSs are almost the same.

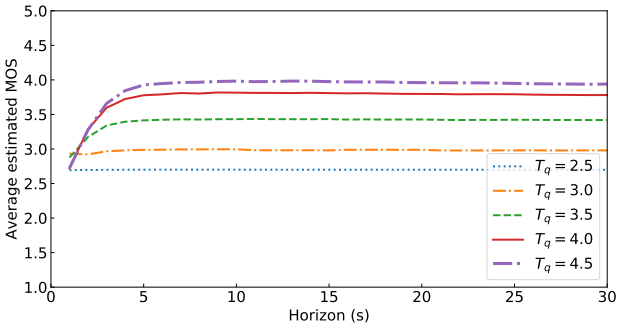

Fig. 6. Average estimated MOS when h is changed.

reason for the reduction, we plot a sample bitrate series in Fig. 5. The *baseline* sometimes selects a very high bitrate or a very low bitrate. BANQUET consistently selects a modest bitrate. Because selecting a higher bitrate significantly increases the traffic volume, there is a difference in traffic-volume reduction even with the same estimated MOS.

In summary, BANQUET reduces the traffic volume by up to 47.0% while maintaining the same average estimated MOS by concentrating the estimated MOS of each view onto T_q. BANQUET also reduces the traffic volume even when viewing with the same estimated MOS, which is achieved by consistently selecting a modest bitrate series.

C. Impact of Prediction Horizon Parameter

BANQUET estimates the future throughput for h and selects the appropriate bitrate based on a brute-force search. If h is set excessively high, the calculation time increases exponentially. If h is excessively low, BANQUET may not select an appropriate bitrate. Thus, we examine how the average estimated MOS changes when h changes.

Figure 6 shows the average estimated MOS when h is changed from 1 s to 30 s and T_q is changed from 2.5 to 4.5. We find that the average estimated MOS gradually increases as h increases, and converges when h is approximately 7 s. This is because inappropriate bitrate selection is corrected when the client selects the next chunk.

The appropriate h may be affected by the network conditions, bitrate setting, and player settings. Taking into account these variable factors, we compare the calculation time when h is set to 10 s (optimized) and 30 s (non-optimized) on an actual smartphone, a Sony Xperia XZ. The evaluation settings are the same as those described in Section IV-A except for h. The results are summarized in Table III, which contains the average and standard deviations for the calculation time when T_q is set to 4.0. The results indicate that by optimizing h, BANQUET calculates the bitrate in 5.93 ms on average, which means that the optimization reduces the calculation time by 96.2%. Since the standard deviation is also sufficiently low, the calculation overhead of BANQUET is sufficiently low to run on an actual smartphone.

TABLE III
CALCULATION TIME OF BANQUET FOR OPTIMIZED AND NON-OPTIMIZED h.

	Calculation Time with Optimized h (ms)	Calculation Time with Non-optimized h (ms)
Mean	5.93	155
Standard Deviation	12.6	99.2

V. CONCLUSIONS AND FUTURE WORK

We proposed BANQUET, a novel bitrate-selection algorithm that enables us (1) to control the balance between the QoE and traffic volume using an intuitive balancing parameter and (2) to achieve a better QoE with a lower traffic volume. Assuming that a target QoE is given as a balancing parameter, BANQUET selects the bitrate that minimizes the traffic volume while maintaining the estimated MOS above the target. To set the target QoE intuitively for users, BANQUET adopts MOS as a QoE metric since it corresponds to a subjective score. We evaluated BANQUET through trace-based simulation using actual 4G LTE throughput data. BANQUET reduced the traffic volume by up to 47.0% compared to that for the *baseline* while maintaining the same average estimated MOS. By optimizing the throughput prediction horizon parameter, BANQUET calculated the bitrate on an actual smartphone in 5.93 ms on average without degrading the achieved MOS.

For future work, we plan to address three problems. First, we plan to improve the throughput estimation step in BANQUET because the throughput changes drastically depending on the time, client position, or network type, *e.g.*, wired, cellular, or Wi-Fi. Considering such effects on the throughput estimation may enable more appropriate bitrate selection. Second, we will expand the QoE-estimation model considering various user or content types. Although BANQUET assumes that the average QoE-estimation model is given, the actual QoE may differ depending on the user or content characteristics. Thus, more detailed QoE-estimation models could improve the performance of BANQUET. Third, we will deploy BANQUET in an actual environment and evaluate the performance in the wild.

REFERENCES

[1] "Cisco Visual Networking Index: Forecast and Methodology, 20172022 - Cisco," https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.html.

[2] F. Dobrian, S. Vyas, S. Ion, and Z. Hui, "Understanding the Impact of Video Quality on User Engagement," in *ACM SIGCOMM*, Aug. 2011, pp. 362–373.

[3] S. S. Krishnan and R. K. Sitaraman, "Video Stream Quality Impacts Viewer Behavior: Inferring Causality Using Quasi-Experimental Designs," *IEEE Trans. Netw.*, vol. 21, no. 6, pp. 2001–2014, Dec. 2013.

[4] "Mobile Video: Exposed — Streaming Video Alliance," http://svideoalliance.wpengine.com/mobile-video-exposed/.

[5] "YouTube," https://www.youtube.com.

[6] T. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson, "A Buffer-based Approach to Rate Adaptation: Evidence from a Large Video Streaming Service," in *ACM SIGCOMM*, Aug. 2014, pp. 187–198.

[7] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A Control-Theoretic Approach for Dynamic Adaptive Video Streaming over HTTP," in *ACM SIGCOMM*, Aug. 2015, pp. 325–338.

[8] S. Paris, A. Destounis, L. Maggi, G. S. Paschos, and J. Leguay, "A Control Theoretic Approach to ABR Video Streaming: A Fresh Look at PID-based Rate Adaptation," in *IEEE INFOCOM*, Apr. 2017, pp. 1–9.

[9] Z. Akhtar, Y. S. Nam, R. Govindan, S. Rao, J. Chen, E. Katz-Bassett, B. Ribeiro, J. Zhan, and H. Zhang, "Oboe: Auto-tuning Video ABR Algorithms to Network Conditions," in *ACM SIGCOMM*, Aug. 2018, pp. 44–58.

[10] H. Mao, R. Netravali, and M. Alizadeh, "Neural Adaptive Video Streaming with Pensieve," in *ACM SIGCOMM*, Aug. 2017, pp. 197–210.

[11] ITU-T Recommendation P.800.2, "Mean opinion score interpretation and reporting," May. 2013.

[12] K. Yamagishi and T. Hayashi, "Parametric Quality-Estimation Model for Adaptive-Bitrate Streaming Services," *IEEE Trans. on Multimedia*, vol. 19, no. 7, pp. 1545–1557, Feb. 2017.

[13] ITU-T Recommendation P.1203, "Parametric bitstream-based quality assessment of progressive download and adaptive audiovisual streaming services over reliable transport," Nov. 2016.

[14] K. Yamagishi, "Audio/visual quality estimation device, method for estimating audio/visual quality, and program," WO Patent 2 017 104 416.A1, Jul., 2017.

[15] D. Raca, J. J. Quinlan, A. H. Zahran, and C. J. Sreenan, "Beyond Throughput: A 4G LTE Dataset with Channel and Context Metrics," in *ACM MMSys*, Jun. 2018, pp. 460–465.

[16] B. Rainer, S. Petscharnig, C. Timmerer, and H. Hellwagner, "Statistically Indifferent Quality Variation: An Approach for Reducing Multimedia Distribution Cost for Adaptive Video Streaming Services," *IEEE Trans. Multimedia*, vol. 19, pp. 849–860, Mar. 2017.

[17] ITU-T Recommendation P.913, "Methods for the subjective assessment of video quality, audio quality and audiovisual quality of internet video and distribution quality television in any environment," Mar. 2016.

[18] K. Satoda, H. Yoshida, H. Ito, and K. Ozawa, "Adaptive Video Pacing Method based on the Prediction of Stochastic TCP Throughput," in *IEEE GLOBECOM*, Dec. 2012, pp. 1944–1950.