JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 36, 1315-1337 (2020)
DOI: 10.6688/JISE.202011_36(6).0012

Scheduling for the QOE Optimization of DASH Streaming*

YU-CHI LIU[1], MENG-HUANG LEE[3,+], SHIN-HUNG CHANG[4], WEI-CHUN CHUNG[2], KUAN-JEN WANG[4], JAN-MING HO[2] AND RAY-I CHANG[1]
[1]*Department of Engineering Science and Ocean Engineering*
National Taiwan University
Taipei, 116 Taiwan
[2]*Institute of Information Science*
Academia Sinica
Nankang, 115 Taiwan
[3]*Department of Information Technology and Management*
Shih Chien University
Taipei, 104 Taiwan
[4]*Department of Computer Science and Information Engineering*
Fu Jen Catholic University
New Taipei City, 242 Taiwan

DASH (Dynamic Adaptive Streaming over HTTP) is now the most widely used standard in video streaming. To support a DASH video transmission over a residential network with a small variation in bandwidth (such as a DSL-based network for IPTV), we designed an algorithm for generating the optimal transmission schedule, *L2H*. Given the transmission rate and initial delay, the schedule can optimize the quality of experience (QoE) metrics such as rebuffering, the lexicographically maximal resolution, the number of resolution switching events, and normalized average quality. We further present $L2H_B$ to consider the usage of the system buffer when applying *L2H*. $L2H_B$ demonstrates a benefit compared with algorithms proposed by previous studies based on objective and subjective QoE evaluations. In addition, by introducing a system buffer size constraint, the proposed algorithm can control the transmission schedule to enable the segments with the highest resolution to appear as soon as possible, encouraging viewers to continue watching.

Keywords: DASH, IPTV, quality of experience (QoE), scheduling, quality of service (QoS)

1. INTRODUCTION

Mobile data traffic has been dominated by video streaming [11] and therefore many content providers have deployed various solutions for the efficient transmission of multimedia streams to end-users. Particularly for IPTV, an Internet application has the potential to overwhelm the Internet backbone and residential broadband access networks [7]. In addition, as noted by Driscoll *et al.* [22], IPTV streams operate on a stand-alone basis, and the network uses a switch-video mode of operation. Therefore, the constant bitrate (CBR) technique is used for carrying video streams over DSL-based networks.

To improve the user experience for clients watching on-demand videos, content providers have introduced various streaming technologies into their infrastructures. Microsoft Media Services (MMS) and the Real Time Streaming Protocol (RTSP) are widely used solutions for video streaming services. However, solutions based on HTTP are usually

Received January 22, 2019; revised May 15, 2019; accepted October 1, 2019.
Communicated by Haiying Shen.
+ Corresponding author: meng@mail.usc.edu.tw.
* This is a major revised version of a preliminary conference report that was presented in [31].

preferred over those based on TCP because of the following advantages [8]. First, this technology is less expensive to deploy into the existing HTTP infrastructure. Second, it is able to penetrate firewalls and network address translators. Third, it is easier to deploy over a content delivery network.

Dynamic Adaptive Streaming over HTTP (MPEG-DASH, or DASH) [25, 26] is a pull-based solution for providing uninterrupted video streaming regardless of the network conditions and device capabilities. The technology has gained popularity among multimedia content providers, including Netflix [2], Hulu [1], and YouTube [3]. A video encoded in the DASH format consists of a series of segments. Each segment contains video content of the same playback duration, *e.g.*, a few seconds or tenths of a second in length. Each video segment is presented at several different resolutions. A DASH server is a standard HTTP server. A client application can choose the resolution of the next segment to be retrieved based on the state of its buffer occupancy and a real-time measurement of the network performance quality. However, the DASH standard does not specify how a streaming service may adapt to network dynamics for delivering the optimum transmission schedule without introducing or reducing visual quality. To the best of our knowledge, most studies have supported DASH video streaming services through the best-effort Internet model (*e.g.*, OTT service), but have not comprehensively investigated IPTV services that use residential broadband access networks.

In this paper, we present the architecture and algorithms for supporting DASH CBR-encoded video streaming services [10, 18, 22, 29] over residential networks with small bandwidth variations (such as a DSL-based network for IPTV [22]). The first algorithm, denoted as *L2H*, was designed to generate a transmission schedule for a given transmission rate and initial delay, optimizing the quality of experience (QoE) metrics. QoE refers to subjective user opinions regarding the quality of service, which are difficult to quantify. In this study, we employed several QoE metrics for quantitatively modeling the notion of QoE, enabling its analysis. Specifically, we considered QoE metrics such as rebuffering, the lexicographically maximal resolution, the minimal number of resolution switching events, and the smoothness of the change in resolution. We also present *L2H* with system buffer size constraints, $L2H_B$. In contrast to the traditional notion of introducing system buffer size constraints for modeling a limited amount of memory embedded in a user device, we show that, by introducing system buffer size constraints, one may control the transmission schedule such that the higher-resolution segments appear as soon as possible, encouraging viewers to continue watching. We also show through simulations that the proposed algorithms achieve a higher QoE than that of previously reported algorithms.

The remainder of this paper is as follows. Related works are described in Section 2. In Section 3, the DASH scheduling problems and QoE metrics are presented. The proposed algorithms are then discussed in Section 4. Simulation and evaluation results are next presented in Section 5. Finally, some concluding remarks and areas of future study are listed in Section 6.

2. RELATED WORKS

Most previous works on DASH video streaming services have focused on OTT (*i.e.*, dynamic changes in network bandwidth) [32]. Qadir *et al.* [23] reviewed different mechanisms proposed for the QoE optimization of video traffic. Examples include resolution adaptation, cross-layer mechanisms, scheduling, and content and resource management.

The majority of the proposed resolution adaptation algorithms can be divided into two major categories. One is based on the estimation of system buffer occupancy [9, 13, 14, 20, 21, 30, 33, 34, 36] at the client end, and swaps video resolutions using various thresholds. The other is based on the network bandwidth [15, 19, 35] switch resolutions depending on the bandwidth; however, the throughput may not be optimal in an unstable network.

Buffer-based Algorithms: Yuming *et al.* [9] proposed a QoE-friendly resolution adaptation method that can achieve fewer resolution switching events and a smooth change in resolution. Muller *et al.*'s work [21], which is referred to as the *BufferLevel* algorithm in the remaining part of this paper, sets a 30-s buffer (*i.e.*, 15 segments) to compensate the high bandwidth fluctuations. The *BufferLevel* algorithm determines the next read segment resolution using the segment retrieving cycle according to the state of the buffer occupancy. When the buffer occupancy is at a lower level, the segment with a higher resolution will be read during the next segment retrieving cycle; when the buffer occupancy is at a higher level, the segment with a lower resolution will be read during the next segment retrieving cycle. It was found in our experiments that when the system is under a steady state, the average bit rate of the segments read by the system is near or close to the network bandwidth of the system, whereas the network bandwidth utilization depends on the distance between the algorithm's occupancy level and the distribution of the segment resolutions.

Bandwidth-based Algorithms: Liu *et al.* [19] proposed a novel resolution adaptation algorithm which is referred to as the *RateAdaptation* algorithm in the remaining part of this paper for DASH streaming that detects bandwidth changes using smoothed network throughput. The measurement of which is based on the segment fetch time (SFT); however, the algorithm does not consider changes in the resolution amplitude, resolution switching events, or high-resolution video. The RA algorithm determines the next segment resolution read through the segment retrieving cycle according to the bandwidth size. When the bandwidth is bigger, the segment with a higher resolution will be read, and when the bandwidth is smaller, the segment with a lower resolution will be read. However, when the bandwidth is abruptly enlarged, a buffer overflow may easily occur when the input of the buffer is larger than its output. Therefore, the algorithm needs to wait for a period of time before reading the segments to avoid a buffer overflow. It was found in our experiments that under a steady network environment, the segments read by the system are slower than the network bandwidth of the system. In addition, to avoid a buffer overflow, the algorithm reads the segments while pausing according to the changes in the network bandwidth, and thus the network bandwidth cannot be fully utilized. To address this problem, the gap between the two bitrates that the video server provides should be lower to control the value to be smaller. Therefore, the normalized average quality is higher than before. If it is larger, the threshold when switching upward will be higher. To address this problem, the gap between the two bitrates provided by the video server should be lower to control the value to be smaller. Therefore, the normalized average quality is higher than before.

The present study differs from these related works mainly in two ways. First, we focus on DASH video streaming services over residential networks with a small variation in bandwidth (such as a DSL-based network for IPTV). Second, several QoE metrics are presented to yield a quantitative model of QoE. Two versions of the *L2H* algorithm, with and without system buffer size constraints, were designed to achieve a better user experience.

3. PROBLEM STATEMENT

In this section, we formulate the transmission schedule problem for achieving an improved user QoE. The system parameters are provided for further discussions.

Fig. 1 shows an example of an uninterrupted transmission schedule. An initial delay, T_p, was designed to avoid rebuffering by preloading the data before playback. DASH video segments are loaded periodically after every ΔT time point, where ΔT corresponds to the playback duration of a DASH video segment. The steps of the transmission schedule are represented by a zigzag-like shape. In this paper, the proposed algorithms were devised for generating a transmission schedule for a given transmission rate and initial delay, such that the QoE metrics are optimized. QoE refers to subjective user opinions on the quality of service, which are difficult to quantify. Therefore, to enable analysis, we considered QoE metrics for quantitative modeling of QoE, namely rebuffering, the lexicographically maximal resolution, the minimal number of rate switching events, and the smoothness of the resolution change. In addition, we considered an algorithm with system buffer size constraints. In contrast to the traditional notion of introducing system buffer size constraints for modeling the limited amount of memory embedded in a user device, the algorithm controls the transmission schedule, such that higher-resolution segments appear as soon as possible, encouraging viewers to continue watching.

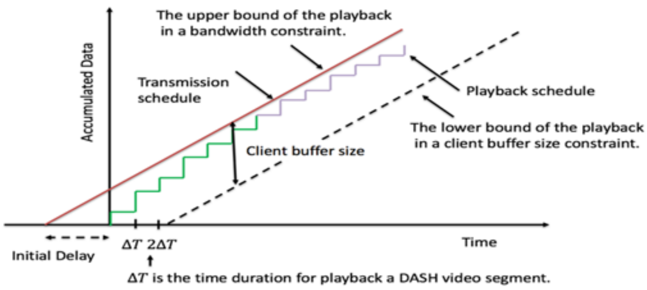


Fig. 1. An example of an uninterrupted transmission schedule.

The system parameters are listed in Table 1. A DASH video is divided into N segments. Each segment is ΔT seconds long, and is encoded at R different resolutions. Thus, a DASH video is formulated as follows:

$$V = \{s_{i,k}\},\ 1 \leq i \leq N,\ 1 \leq k \leq R. \tag{1}$$

When considering CBR encoding, it is assumed that every file at the same resolution in each segment has the same size. That is, $|s_{i,k}| = |s_{j,k}|$, where $1 \leq i, j \leq N$. In addition, a *feasible transmission schedule* of DASH segments is one in which re-buffering at the client side does not occur, and is formulated as follows:

$$S(V) = \{s_{i,k}\},\ 1 \le i \le N. \tag{2}$$

We also define the resolution vector (RV) to compare the QoE levels between the schedules in the lexicographic order. The RV is a vector consisting of n resolutions that are scheduled for playback, and is sorted in order of decreasing resolution. The RV notation is $RV = (n_1, ..., n_R)$. Thus, in our model, schedule $S(V)$ is superior to $S(V)'$ if and only if $S(V)$ is lexicographically greater than $S(V)'$.

Table 1. Notations of the system parameters.

notation	definition and description
$s_{i,k}$	The ith segment of a DASH video in the kth resolution.
$\lvert s_{i,k}\rvert$	The file size of the segment $s_{i,k}$.
V	The set of $s_{i,k}$ segments of a DASH video.
B	The constant network bandwidth.
β	The system buffer size.
R	The number of resolutions of a DASH video segment.
N	The number of segments divided by a CBR DASH video.
T_p	The initial delay.
ΔT	The playback duration of a DASH video segment.
$S(V)$	The transmission schedule of a DASH video V. The playback unit is $s_{i,k}$.
n_k	The number of segments of $S(V)$ in the kth resolution.
RV	The set of n_k, corresponding to the resolution vector in a *feasible schedule* of a DASH video.
m_N	The slope of the start point of the lowest resolution with respect to the last point of the highest resolution.
m_B	The slope of the network bandwidth.
m_R	The slope of the lowest resolution.
BO_i	The buffer occupancy of playing i^{th} segment of a DASH video.

4. FEASIBLE SCHEDULE AND QOE OPTIMIZATION

In this section, we present the architecture of the *L2H* algorithms with and without a buffer size constraint. Examples are also given to illustrate the optimized QoE metrics, and to briefly explain the related performance issues.

4.1 Assumptions

We make the following assumptions in this paper:

- The available network bandwidth is constant (*i.e.*, the network bandwidth has small variations).
- A service client has sufficient computational power to render all of the video quality representations.
- The available network bandwidth between the server and client is always higher than or equal to the lowest video quality level.
- We assume that segments of all levels of quality are provided by the content provider, which means that the average bitrates of the video quality levels are known.

These assumptions are realistic. An IPTV stream operates on a stand-alone basis, and the network uses a switch-video mode operation. Therefore, the CBR technique is used for carrying video streams over DSL-based networks. Content providers provide all video segments with all available bitrate representations for the service providers, which are often responsible for offering metered media services to the end users. Therefore, they can easily obtain the video quality information from the encoding profiles. In this paper, we are only interested in transmitting DASH videos over residential networks with a small variation in bandwidth, instead of other factors such as the computational power of the clients and the operational costs of the service providers (*i.e.*, the caching, transcoding, and bandwidth costs). As a result, the objective of this paper is to find the optimal solution that maximizes the user experience.

4.2 System Architecture

Fig. 2 shows the architecture of our proposed system, which was referenced from the Digital Audio-Visual Council (DAVIC) [12], which defines the distribution of video and multimedia content through communication networks. There are four major rules, *i.e.*, the
4 service client (*i.e.*, DASH client), content provider (*i.e.*, a CBR-MR video server), service provider, and deliver system. To optimize the given QoE metrics, we first focus on transmitting DASH videos over networks with a constant bandwidth. The service provider is designed for analyzing the status of the transmission network for making playback decisions to prompt the users to continue watching.

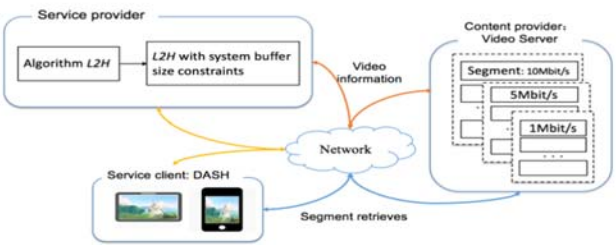


Fig. 2. Architecture of our proposed framework for optimizing the QoE metrics of a DASH video.

4.3 Finding a Resolution Vector

In the following, T_p is a *feasible startup/initial delay* (*time*) that is necessary for a feasible transmission schedule. N is the number of segments in a DASH video, and ΔT represents the constant playback duration of a video segment. The overall time for transmitting a DASH video is $T_p + (N - 1) * \Delta T$. B is assumed to be a given constant network bandwidth for transmission. Thus, the data size transmitted during the video playback can be defined as follows:

$$B * (T_p + (N - 1) * \Delta T). \tag{3}$$

Let n_1 be the number of segments with the highest resolution (each with the size $|s_{i,1}|$),

$n_{i,R}$ be the number of segments with the lowest resolution (each with the size $|s_{i,R}|$), and n_k be the number of segments with intermediate resolutions. The data size of a video with N segments is:

$$n_1 * |s_{i,1}| + \sum_{j=2}^{R-1} n_j * |s_{i,j}| + (N - n_1 - \sum_{j=2}^{R-1} n_k) * |n_R|. \tag{4}$$

To generate a transmission schedule without interruptions, the expression in Eq. (3) should be larger than that in Eq. (4). Thus, we developed a heuristic algorithm (Algorithm 1) to discover a *feasible resolution vector* for which a *feasible transmission schedule* can be obtained. The algorithm iterates through resolutions to find a segment with an appropriate resolution for playback (lines 2 to 12). If buffer underflow occurs, the algorithm stops the search and returns the resolution vector (lines 7 to 10). Therefore, the resolution vector output by the algorithm features the greatest number of highest resolution segments.

Algorithm 1: Finding a resolution vector

Input: Constant network bandwidth, B; Segment information of a DASH video, $s_{i,k}$
Output: Resolution vector, RV
1. *netCumuTransData* $= B * (Tp + (N-1) * \Delta T)$;
2. **for** each $k \in [1, R]$ do
3.   *cumuPlaybackData* $= \sum_{j=1}^{k} n_j * |s_{i,j}|$;
4. $n_k = \lfloor(netCumuTransData - cumuPlaybackData) - (N - \sum_{j=1}^{k} n_j) * |s_{i,R}|)/(|s_{i,k}| - |s_{i,R}|)\rfloor$;
5. $n_R = N - \sum_{i=1}^{k} n_i$;
6. $RV \leftarrow n_k, n_R$;
7.   **if** (*netCumuTransData* − *cumuPlaybackData*) < $|s_{i,R}|$
8.   **then**
9.   break and output RV;
10.   **end_if**
11.   return RV;
12. **end_for**

4.4 Feasible Transmission Schedule of Algorithm *L2H*

L2H (Algorithm 2) was designed to generate a *feasible transmission schedule* based on the rearrangement of the resolution vector. Fig. 3 shows a possible *feasible schedule* generated by *L2H*. We assume that the *feasible schedule* advances from the lowest resolution to the highest resolution. The schedule is feasible if and only if the amount of the transmitted data is lower than the bandwidth during playback. Therefore, a schedule with optimal QoE metrics can be defined through the following theorems:

Algorithm 2: *L2H*

Input: A resolution vector, RV
Output: A *feasible transmission schedule*, $S(V)$
1. $i \leftarrow 1$;
2. **for** each $j \in [1, R]$ do
3. $n_j \leftarrow RV$
4.   **for** $x \leftarrow 1$ to n_j do
5. $S(V) \leftarrow s_{i,j}$

```
6.            i++;
7.        end_for
8.    end_for
```

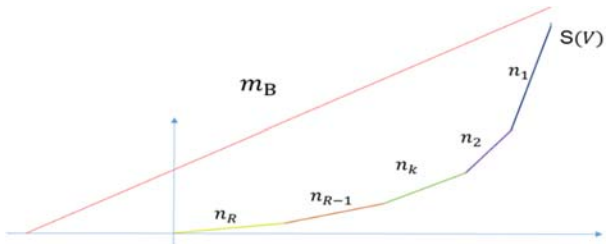


Fig. 3. An example illustrating *L2H*.

Theorem 1: The feasible transmission schedule generated by *L2H* has the optimal QoE with respect to the lexicographic order.

***Proof*:** We prove this theorem through a contradiction.

Assume that *L2H* satisfies the following constraint: $B * (T_p + (N - 1) * \Delta T) \geq \sum_{j=1}^{R} n_j |s_{i,j}|$, and that the resolution vector $RV(n_1, n_2, \ldots, n_R)$ that maximizes the given QoE metrics. Suppose that there exists a resolution vector $RV(n'_1, n'_2, \ldots, n'_R)$ for which $n'_1 > n_1$ or $n'_2 > n_2$, *etc*. The definition of the scheduling feasibility prohibits the overall amount of data consumed at the client end from exceeding the overall amount of transmitted data. However, the vector RV' violates this definition, (*i.e.*, $B * (T_p + (N - 1) * \Delta T) \leq \sum_{j=1}^{R} n_j |s_{i,j}|$).

Theorem 2: The feasible transmission schedule generated by *L2H* guarantees jitter-free playback.

***Proof*:** We prove this theorem through induction. The proof includes the following steps:

Step 1: We prove that the playback duration of the segments with resolution n_R guarantees jitter-free playback.

To form a virtual dashed line, as shown in Fig. 4, we first connect the start point with the lowest resolution and the end point with the highest resolution. The line m_N is given

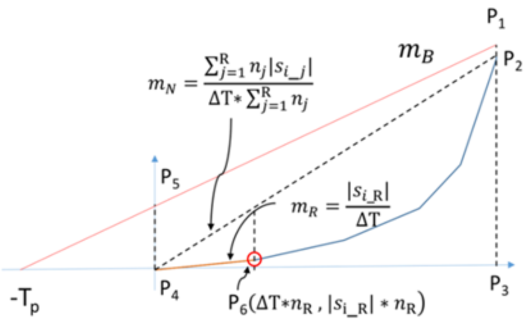


Fig. 4. Schematic representation of the proof of Step 1 in Theorem 2.

by $\left(\frac{\sum_{j=1}^{R} n_j * |s_{i,j}|}{\Delta T * \sum_{j=1}^{R} n_j}\right)$, which is lower than $\left(\frac{|s_{i,R}| * \sum_{j=1}^{R} n_j}{\Delta T * \sum_{j=1}^{R} n_j}\right)$. However, m_R is equal to $\left(\frac{|s_{i,R}|}{\Delta T}\right)$ and $\left(\frac{|s_{i,R}| * \sum_{j=1}^{R} n_j}{\Delta T * \sum_{j=1}^{R} n_j}\right)$, therefore, $m_N > m_R$.

In a specific region (*e.g.*, ■ $P_1P_3P_4P_5$), assume that $\overline{P_1P_5}$ does not intersect $\overline{P_2P_4}$. In this case, our schedule satisfies $B * (T_p + (n_k - 1)) * \Delta T \geq \sum_{j=1}^{R} n_j * |s_{i,j}|$, therefore, $m_B > m_N > m_R$. Thus, we proved that the playback duration of segments with resolution n_R guarantees jitter-free playback.

Step 2: We prove that if the playback duration of the segments with the resolution $n_k + 1$ guarantees jitter-free playback, and thus the playback duration of segments with the resolution n_k also guarantees jitter-free playback.

We first draw two parallel lines ($\overline{P'_1P'_4}$ and $\overline{P_1P_4}$) that intersect the X axis at right angles (see Fig. 5). We then extend the segments s_{ik} to obtain the intersection points P_3 and P'_3 on the lines $\overline{P_1P_4}$ and $\overline{P'_1P'_4}$, respectively. To verify that the data consumption in each time slot does not exceed the transmitted data, we calculate the slopes of $m_{\overline{P_3P_2}}$ and $m_{\overline{P_3P'_3}}$. Here, $m_{\overline{P_3P_2}}$ is equal to $\left(\frac{\sum_{j=k}^{1} n_j * |s_{i,j}|}{\Delta T * \sum_{j=k}^{1} n_j}\right)$, which is lower than $\left(\frac{|s_{i,k}| * \sum_{j=k}^{1} n_j}{\Delta T * \sum_{j=k}^{1} n_j}\right)$. However, the slope $m_{\overline{P_3P'_3}}$ is equal to $\left(\frac{|s_{i,k}| * \sum_{j=k}^{1} n_j}{\Delta T * \sum_{j=k}^{1} n_j}\right)$. Therefore, we state that $m_{\overline{P_3P_2}} > m_{\overline{P_3P'_3}}$.

In a specific region (*e.g.*, ■ $P'_1P'_3P_3P_1$), assume that $m_{\overline{P'_1P_1}}$ does not intersect $m_{\overline{P_3P'_3}}$. In this case, our schedule satisfies $B * (T_p + (n_k - 1)) * \Delta T \geq \sum_{j=1}^{R} n_k * |s_{i,j}|$. Therefore, $m_B > m_{\overline{P_2P_3}} > m_{\overline{P_3'P_3}}$, and we proved that the playback duration of the segments with the resolution n_k guarantees jitter-free playback.

Thus, from Steps 1 and 2, using mathematical induction, we proved that the *feasible transmission schedule* generated by the *L2H* algorithm guarantees jitter-free playback.

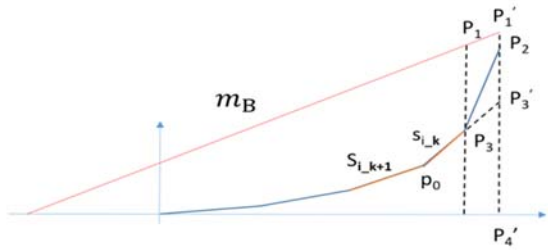


Fig. 5. Schematic representation of the proof of Step 2 in Theorem 2.

Theorem 3: The feasible transmission schedule generated by *L2H* features a small number of switching events.

Let us use the resolution vector for calculating the number of switching events as the number of instances in which the resolutions of two adjacent segments are different. The definition is as follows:

$$g^* = \sum_{i=2}^{N} g(s_{i,k}) \text{ where } g(s_{i,k}) = \begin{cases} 1, & \text{if } f(s_{i-1,k}) \neq f(s_{i,k}) \\ 0, & \text{else} \end{cases}.$$

Note that $f(s_{i,k})$ corresponds to resolution k of the ith segment, R denotes the number of resolutions, and n_k is the number of segments with resolution k of a *feasible transmission schedule*. Assuming that each resolution is used at least once, we have $n_k > 0$ and $1 \leq k \leq R$. The number of switching events g^* for *L2H* is $R - 1$. In the following, we prove that *L2H* generates the lowest number of switching events.

***Proof*:** We prove this theorem through a contradiction.

Because $n_k > 0$ and $1 \leq k \leq R$, we have $g^* = R - 1$. Suppose that $\exists g^* = R - 1$. From the definition of R in Table 1, $g^* = R - 1$ uses fewer than R resolutions. That is a contradiction and thus the theorem is proved.

4.4 *L2H* with System Buffer Size Constraints (*L2H$_B$*)

Algorithm 1 outputs a resolution vector that serves as the input to *sL2H*. Without considering the system buffer size constraints, *L2H* yields a *feasible transmission schedule* to serve as a feasible transmission schedule, ordered from the lowest- to the highest-resolution segments, for a given transmission rate and initial delay. By introducing the system buffer size constraints, we developed Algorithm 3, denoted as $L2H_B$, to obtain a transmission schedule according to the output of Algorithm 1 so that the transmission schedule also guarantees that optimal QoE metrics are obtained. $L2H_B$ starts to control the transmission schedule from low-resolution segments (lines 8 to 11). $L2H_B$ switches to the transmission of high-resolution segments when buffer overflow occurs. By contrast, $L2H_B$ does not switch back to the transmission of low-resolution segments for the transmission schedule until buffer underflow occurs (lines 15 to 18). On the basis of the resolution vector, the output of Algorithm 1, $L2H_B$ continuously inputs video segments into transmission schedule $S(V)$. In $L2H_B$, we use a variable d to control the segment selection order. The initial value of the variable d is zero, therefore, $L2H_B$ inputs video segments into $S(V)$, starting with low-resolution. If system buffer overflow occurs, then $L2H_B$ switches the segment selection order from low- to high-resolution by changing the value of the variable d from zero to one (line 15) and vice versa (line 17).

Theorem 4: $L2H_B$ yields a feasible transmission schedule if the system buffer size constraint is $\beta \geq B * \Delta T + |s_{i,1}| - |s_{i,R}|$.

***Proof*:** A transmission schedule is called feasible when system buffer overflow and system buffer underflow will not occur in a DASH video transmission. Therefore, to prevent these two undesired situations from occurring, the system buffer occupancy of the ith playback duration, BO_i, is constrained by Eq. (5):

$$0 \leq BO_i \leq \beta - B \times \Delta T. \tag{5}$$

While playing the ith segment of a DASH video, $B \times \Delta T$ is calculated as the buffer data increment during each playback duration, ΔT. Additionally, β is defined as the system buffer size constraint. Therefore, it is obvious that Eq. (6) should be constrained and BO_i is equal to $BO_{i\text{-}1} + B \times \Delta T - |s_{i,k}|$, where $1 \leq k \leq R$.

$$0 \leq BO_{i\text{-}1} + B \times \Delta T - |s_{i,k}| \leq \beta - B \times \Delta T \tag{6}$$

In the DASH video streaming, different resolution video segments will be adjusted to send to DASH clients. When $L2H_B$ transmits the highest-resolution segment, $|s_{i1}|$, if buffer underflow is avoided at the playback of the ith video segment, then Eq. (7) should be constrained.

$$|s_{i,1}| - B \times \Delta T \leq BO_{i\text{-}1} \tag{7}$$

On the contrary, when $L2H_B$ transmits the lowest-resolution segment, $|s_{i,R}|$, if the buffer underflow is avoided at the playback of the i-th video segment, then Eq. (8)

$$BO_{i\text{-}1} + B \times \Delta T - |s_{i,k}| \leq \beta - B \times \Delta T. \tag{8}$$

Eq. (9) is derived by substituting $|s_{i,1}| - B \times \Delta T$ from Eq. (7) into $BO_{i\text{-}1}$ of Eq. (8).

$$|s_{i,1}| - |s_{i,R}| + B \times \Delta T \leq \beta \tag{9}$$

Because $|s_{i,1}|$ and $|s_{i,R}|$ are the largest and the smallest video segment size distinctly, $L2H_B$ transmits a video segment of resolution k, $|s_{i,k}|$, whose size must be between $|s_{i,1}|$ and $|s_{i,R}|$, at the ith playback duration. Therefore, $|s_{i,R}| \leq |s_{i,k}| \leq |s_{i,1}|$, where $1 \leq k \leq R$. Constrained by Eq. (9), $\beta \geq B \times \Delta T + |s_{i,1}| - |s_{i,R}|$, $L2H_B$ yields a feasible transmission schedule.

Algorithm 3: *L2H* with the system buffer size constraints (*L2H*)

Input: A resolution vector, $RV = (n_1 \ldots n_R)$; Residential network bandwidth, B, playback duration ΔT; A system buffer size constraint, β.
Output: A *feasible transmission schedule*, $S(V)$.

```
if (β < |s_i,1| − |s_i,R| + B × ΔT)
    return null;
end_if
S(V) ← ∅;
d ← 0;
for i ← 1 to N do
    switch (d)
        case 0: tr ← max{k|n_k > 0 and 1 ≤ k ≤ R};
            break;
        case 1: tr ← min{k|n_k > 0 and 1 ≤ k ≤ R};
            break;
    end_switch
    if (s_it causes buffer overflow or buffer underflow)
        switch(d)
            case 0: d ← 1; /*switch order from low to high*/
                break;
            case 1: d ← 1; /*switch order from high to low*/
                break;
        end_switch
    end_if
    S(V) ← s_i,t;
    n_t ← (n_t − 1);
end_for
return S(V);
```

5. EVALUATION

In Section 4, *L2H* or *L2H* was used to generate a feasible transmission schedule and some of the QoE metrics were proved to be optimal for the schedule. In this section, we consider two types of benchmarks, a buffer-based algorithm [21], *BufferLevel*, and a bandwidth-based algorithm [19], *RateAdaptation*, which are discussed in more detail in Section 2. We further conduct objective and subjective QoE evaluations to show that $L2H_B$ outperforms other previous methods.

5.1 Objective QoE Evaluation

The following three evaluation metrics were used for the evaluation. The first is from the 3GPP DASH specifications TS 26.247 [6]. The other two are defined in the present paper.

1) *Resolution switching events*: These agre defined as the number of resolution switching events in a transmission schedule. For instance, if a transmission schedule is $\{s_{1,1}, s_{2,2}, s_{3,2}\}$, the number of resolution switching events is 1 (*i.e.*, $s_{1,1}$ to $s_{2,2}$).
2) *Maximal highest-resolution*: This is presented in Theorem 1 in Section 4.
3) *Normalized average quality*: This refers to the ratio of *cumuPlaybackData* to *netCumuTransData*. For example, we assume that the video length is 20 s, the network bandwidth is 200 Kbps, the playback resolution vector is (n_1 = 1, n_2 = 4, n_3 = 5), the total number of segments is 4, and the bitrate of $s_{1,1}$, $s_{2,2}$, $s_{3,3}$ are 300, 200, 100 Kbps, respectively. Each segment playback time is 2 second, and the initial delay is 1 second. Therefore, the average normalized quality is $\frac{300 *2\ s*1+200 *2\ s*4+100 *2\ s*5}{200 *[1+(10-1)*2\ s]}$ *100%=84%.

Figs. 6-9 show the performances of $L2H_B$ (yellow curve), *BufferLevel* (black curve), and *RateAdaptation* (cyan curve). The x-axis is the video playback time (s), and the y-axis is the cumulative data (bits) received on the client side. Six different bitrates (*i.e.*, 100, 400, 900, 1500, 2500, and 3000 Kbps) of the content bitrate version are used for the profile [17]

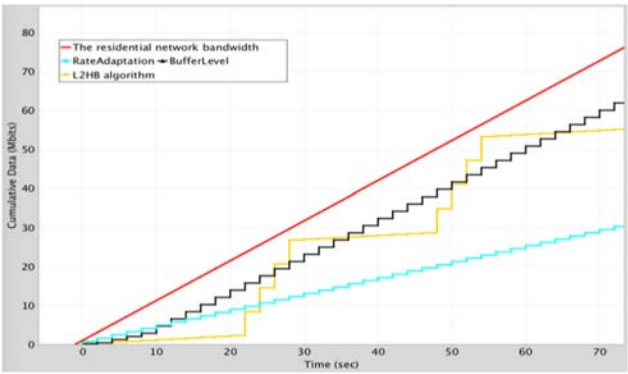


Fig. 6. Performance comparison of $L2H_B$, *BufferLevel*, and *RateAdaptation* when the network bandwidth is 1 Mbps.

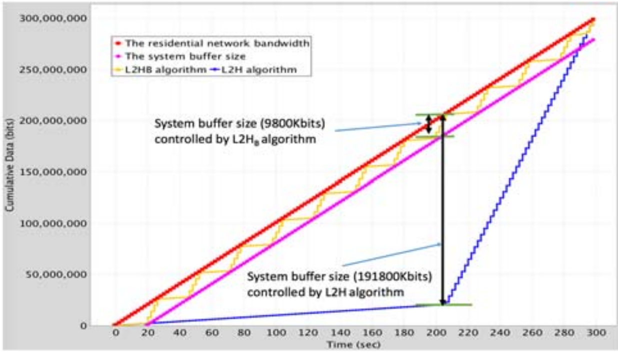


Fig. 7. The effectiveness of *L2HB* on reducing the system buffer size.

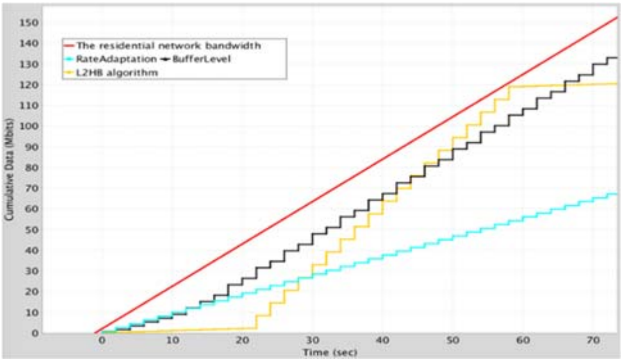


Fig. 8. Performance comparison of *L2HB*, *BufferLevel*, and *RateAdaptation* when the network bandwidth is 2 Mbps.

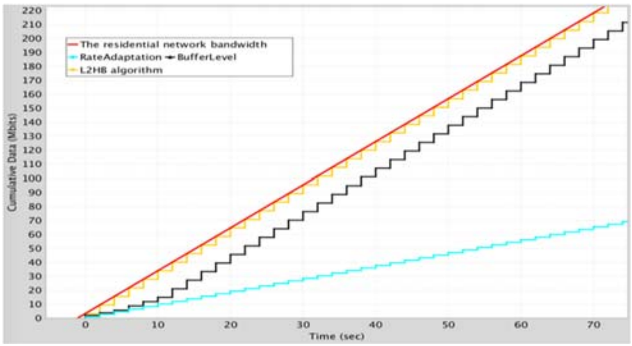


Fig. 9. Performance comparison of *L2HB*, *BufferLevel*, and *RateAdaptation* when the network bandwidth is 3 Mbps.

listed in Table 2. Each video is divided into small fixed-length segments with a 2-s playback duration. A 10-min long video (*i.e.*, 300 segments) was tested and the initial delay was set up to 1s. We set the system buffer to 20s (*i.e.*, ten segments). Finally, the network bandwidth was set to 1, 2, and 3 Mbps to simulate the ADSL bandwidth provided by the Internet service providers (ISPs) in Taiwan [27, 28].

Table 2. Bitrate level in real adaptive streaming system.

Resolution	Bitrate (Kbps)
320 × 240	100
480 × 360	400
720 × 480	900
720 × 576	1500
1280 × 720	2500
1920 × 1080	3000

Some of the observations from this experiment are as follows. First, $L2H_B$ nearly exhausted the available network bandwidth during the transmission time. We also found that the resolution switching events decrease as the network bandwidth increases. This reduction can be attributed to the increased number of highest resolutions in the resolution vector, which are calculated using Algorithm 1. Second, *RateAdaptation* has the lowest normalized average quality compared to the other algorithms because it uses a conservative stepwise switch-up and aggressive switch-down strategy to avoid a playback interruption, despite the fact that the network bandwidth is sufficient for transmitting a higher bitrate segment. Moreover, it also uses the idle time calculation algorithm (*i.e.*, waiting a period of time to fetch a segment) to prevent a client buffer overflow. Therefore, under the scenario of a small variation in bandwidth, *RateAdaptation* is shown too conservative to select a higher resolution segment, leading to a lower normalized average quality.

Finally, as the network bandwidth increases, the resolution switching events decreases compared with $L2H_B$, whereas the one compared with *BufferLevel* does not show a specific behavior. To understand this phenomenon, we looked at the transmission time of one segment under various network bandwidths for *BufferLevel*, as shown in Table 3. Specifically, when the average steady-state is more closed to the segment duration (*i.e.*, 2 s), the number of resolution switching events will be lower. In our case, the steady-state under 1-, 2- and 3-Mbps network bandwidths was 1.8 s (*i.e.*, the segment size is divided by the network bandwidth 1800 Kbits/1000 Kbps), and 3, 1.5 and 2.5, and 2 s, respectively. Thus, a network bandwidth of 2 Mbps has the largest number of resolution switching events when compared with 1 and 3 Mbps. Moreover, the average steady-state, the initial delay, and the system buffer size also effect the normalized average quality. These results show that the normalized average quality is the highest at 3 Mbps, and the lowest at 1 Mbps because the network bandwidth can be utilized as the steady-state is more closed to the segment duration.

Table 3. Transmission time of one segment under various network bandwidths for *BufferLevel* algorithm (values in bold indicate the time when the system is under a steady state).

Quality level	Bitrate (Kbps)	Segment size (Kbits)	Network bandwidth (Mbps)		
			1	2	3
k_1	100	200	0.2	0.1	0.067
k_2	800	1600	0.8	0.4	0.267
k_3	900	1800	**1.8**	0.9	0.6
k_4	1500	3000	**3**	**1.5**	1
k_5	2500	5000	5	**2.5**	1.267
k_6	3000	6000	6	3	**2**

Table 4. QoE metrics for the three algorithms, for a network bandwidth of 1, 2, and 3 Mbps.

Net. Bandwidth (Mbps)	Algorithm	Average playback bitrate (Kbps)	Lexicographic order (*resolution vector*)	NAQ (%)
1	*L2H_B*	998.33	1 (92, 1, 0, 0, 1, 206)	100
	BufferLevel	889.67	2 (0, 0, 40, 255, 3, 2)	89.12
	RateAdaptation	400	3 (0, 0, 0, 0, 300, 0)	40.13
2	*L2H_B*	1996.67	1 (196, 0, 0, 0, 2, 102)	100
	BufferLevel	1971.3	2 (0, 146, 148, 4, 2, 0)	98.73
	RateAdaptation	896.66	3 (0, 0, 0, 299, 1, 0)	44.98
3	*L2H_B*	2995	1 (299, 0, 1, 0, 0, 0)	100
	BufferLevel	2964	2 (294, 0, 3, 3, 0, 0)	98.96
	RateAdaptation	896.66	3 (0, 0, 0, 299, 1, 0)	29.98

Table 4 shows the results of the QoE metrics under different network bandwidths for the three algorithms. We found that $L2H_B$ almost fully utilizes the available bandwidth, and obtains the highest normalized average quality and lexicographic order among all of the algorithms. In these cases, *RateAdaptation* has the lowest number of resolution switching events because it has nearly no changes in resolution. Although *BufferLevel* has a higher normalized average quality than *RateAdaptation*, *BufferLevel* has the highest number of resolution switching events. From the above observation, we found that $L2H_B$ can result in an enjoyable experience regarding the transmission schedule for the user. *RateAdaptation* and *BufferLevel* maintain an uninterrupted playback, but do not consider a high average quality.

Remarks regarding *L2H* and *L2H_B*: Based on the present results obtained in Sections 3 and 4, *L2H* yields a feasible playback schedule that was proved to be optimal in many QoE metrics without considering the influence of the system buffer size constraint. An inspection of the results in Fig. 7 reveals that *L2H* fully utilizes the residential network bandwidth when yielding a transmission schedule. Therefore, *L2H* has the most significant advantage by providing clients with highest-resolution video segments and the mean bitrate of a transmission schedule due to *L2H* is higher than that of the other two algorithms proposed by previous researchers. With the above results, we now turn to a discussion on the system buffer size constraint. Fig. 7 clearly shows that the system buffer size of the transmission schedule is the largest when using *L2H*. To solve this problem, we enhanced *L2H* and propose $L2H_B$ to yield a transmission schedule by using a constrained system buffer size. In Section 5, we showed that $L2H_B$ is more beneficial regarding the objective and subjective QoE.

By introducing a system buffer size constraint, we showed the effectiveness of reducing the system buffer size of $L2H_B$ when streaming a 5-min long DASH video containing 150 segments over a residential network bandwidth of 1000 Kbps. $L2H_B$ yields the same resolution vector (97, 1, 0, 0, 1, 51) as *L2H*. It is obvious that $L2H_B$ not only selects the highest-resolution video segments but also significantly reduces the system buffer size of the yielded transmission schedule, from 191,800 to 9,800 Kbits. In addition, a system buffer underflow and a system buffer overflow do not occur in the transmission schedule when the system buffer size β is at least $B \times \Delta T + |s_{i,1}| - |s_{i,R}|$ (*i.e.*, 2000 Kbits × 2s + 3000 Kbits × 2s − 100 Kbits × 2s = 9800 Kbits).

Table 5. Quality level of videos "Jeremy Lin 2014-2015 Highlights" and "Pixar Short Films #7 For the Birds 2000."

Video	Quality level	Resolution	Bitrate (Kbps)	Segment size (Kbits)
Jeremy Lin 2014-2015 Highlights	k_1	1280 × 720	3293	6586
	k_2	854 × 480	1118	2236
	k_3	640 × 360	612	1224
Pixar Short Films #7 for the Birds 2000	k_1	1280 × 720	2193	4386
	k_2	854 × 480	1085	2179
	k_3	640 × 360	593	1186

5.1 Subjective QoE Evaluation

Section 5.1 shows that $L2H_B$ is better than the buffer-based algorithm [21], *BufferLevel*, and the bandwidth-based algorithm [19], *RateAdaptation*, in terms of the three QoE object metrics. In this section, we further describe the subjective QoE evaluation conducted to show the benefit of $L2H_B$.

To emulate a more realistic environment, we obtained two short video clips from YouTube, and used the video quality provided. One was a sport clip called "Jeremy Lin 2014-2015 Highlights," and the other was an animation clip titled "Pixar Short Films #7 For the Birds 2000." The clip durations were 271 and 205 s, respectively. The video clips were themselves taken from various sources, including sporting events and animation. We subsequently down-sampled the source video clips to other quality levels. Table 5 shows the profiles of all quality levels, *i.e.*, (a) $k1$, 1280 × 720, at 3,293 Kbps, (b) $k2$, 854 × 480, at 1118 Kbps, and (c) $k3$, 640 × 360, at 612 Kbps. In addition, we set the playback duration of a segment to 2s (*i.e.*, is 2s). The file size of a segment is equal to a multiple of the video bitrate, and the level of quality of each segment file size is 6,586, 2,236, and 1,224 Kbits, respectively.

1) *Descriptive Statistics*: Before starting this subjective experiment, every participant was asked to complete a short demographic survey. The participants were mainly from Asia, and 84.7% were between the ages of 21 and 25, with the age groups 26-30 and 15-20 making up 3.8% and 11.5%, respectively. All of the test subjects are students: 51.9% are undergraduate students, 44.2% are graduate students, and 3.8% are doctoral candidates. For their network connection, 53.8% use a wired network, whereas 46.2% use a mobile network. Finally, 88.5% of the participants wear glasses. The resolution distribution of the participants' display monitors is shown in Table 6.

Table 6. The resolution distribution of the participants' display monitors in the subjective QoE experiment.

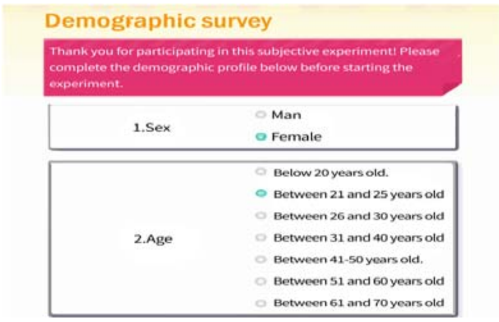
Resolution	# of Participants	Resolution	# of participants
2048 × 1152	1	1440 × 900	1
1920 × 1080	21	1366 × 768	3
1680 × 1050	1	1280 × 800	2
1600 × 900	1	1024 × 768	3

2) *Experiment Setup*: To generate the test videos, we first compressed each video into three different resolutions. We used the YouTube video dataset (see [17]) and a residential bandwidth of 3,000 Kbps to simulate the objective results shown in Section 5.1. We then split and spliced the video segments using the best result of the resolution vectors: (24, 0, 4) for $L2H_B$, (0, 30, 0) for *BufferLevel*, and (8, 12, 10) for *RateAdaptation*.

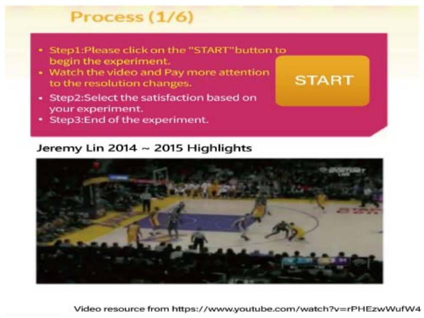
We designed the experiment process as follows: a demographic survey was conducted, followed by the viewing of the test video, and concluding with a questionnaire regarding the user experience (see Fig. 10). First, we gave the participants a short introduction explaining the conditions of the experiment. We then asked them questions while they viewed the videos. In the final step of experiment, the participants were asked to rate their level of satisfaction regarding each video using a five-point rating scale with various options (see Table 7). The results collected from the participants were then mapped to the corresponding MOS scores for a statistical analysis.

Table 7. Evaluation scale (mean opinion score, or MOS).

MOS	Quality	Impairment
1	Bad	Very annoying
2	Poor	Annoying
3	Fair	Slightly annoying
4	Good	Perceptible but not annoying (noticeable, but not significant)
5	Excellent	Imperceptible (did not notice any changes)



(a) A demographic snapshot of the subjective experiment.



(b) A video capture of the subjective experiment.

Fig. 10. Website designed for conducting the subjective measurements.

We employed a one-way analysis of variance (ANOVA) [16] test to identify the statistically different mean opinion scores between the algorithms. We then used the *p*-value to determine whether the variance between and/or within the algorithms was statistically significant. Thus, the null hypothesis (H_0) was set such that the three algorithms have no significant effects. The criterion for determining the statistical significance is $p < 0.05$, which is a considerable threshold for rejecting the null hypothesis, and suggesting whether the difference between the algorithms is statistically significant. In addition, we also used

Bonferroni-corrected *t*-tests to analyze the significance between the algorithms, and to thus identify the best algorithm that achieves a better user experience. Note that the criterion of *the* Bonferroni-corrected *t*-tests is set as $p < \frac{0.05}{c_2^3} = 0.017$.

3) *QoE Results for Subjective Experiment*: Our proposed $L2H_B$ obtained a MOS score of 168, and showed the best results compared to the other algorithms. We also obtained the best-quality results with MOS scores of 3 to 5 (*i.e.*, fair, good, and excellent quality) against the other algorithms (Table 8).

Table 8. MOS scores obtained for the test cases of the three algorithms.

Types of videos	MOS	$L2H_B$	*BufferLevel*	*RateAdaptation*
Sport	1	3	7	14
	2	7	15	19
	3	22	19	14
	4	14	8	3
	5	5	3	2
	SUM	168	141	116
	AVG	3.23	2.71	2.23
	S.D.	1	1.15	1.08
Animation	1	0	2	4
	2	5	7	7
	3	5	17	17
	4	26	33	17
	5	16	4	17
	SUM	209	175	170
	AVG	4.01	3.36	3.27
	S.D.	0.8	0.9	1.46

* A higher score indicates a more confident user experience.

Table 9. Results of a one-way ANOVA.

Types of videos	Source of Variation	Sum of square (SS)	Degree of freedom (df)	Mean of sum (MS)	*F*	*P*-Value	*F* crit
Sport	Between Category Exemplar	26.01	2	1	12.05	$1.38*10^{-5}$	3.06
	Within Category Exemplar	165.13	153	1.15			
	Total	191.15	155	1.08			
Anima-tion	Between Category Exemplar	17.32	2	8.66	1.05	$4.08*10^{-4}$	
	Within Category Exemplar	161.27	153	1.05			
	Total	178.59	155	–			

We used the mean and standard deviations of the MOS scores of the algorithms as the inputs for the ANOVA test (see Table 8). The results of the one-way ANOVA test show that the three algorithms have statistically significant differences because the *p-value* of the algorithms is lower than 0.05, which refutes the null hypothesis (see Table 9). The Bonferroni-corrected *t*-tests show that $L2H_B$ has a better user-experience satisfaction

Table 10. Results of Bonferroni-corrected *t*-test for the three algorithms using MOS scores.

Type of video	Algorithms	t	t crit	p-value
Sport	$L2H_B$ vs *BufferLevel*	2.55	1.98	0.012
	$L2H_B$ vs *RateAdaptation*	4.99		$2.48*10^{-6}$
	RateAdaptation vs *BufferLevel*	2.32		0.022
Animation	$L2H_B$ vs *BufferLevel*	3.6		$4.8*10^{-4}$
	$L2H_B$ vs *RateAdaptation*	3.59		$2.5*10^{-4}$
	RateAdaptation vs *BufferLevel*	0.28		0.77

[*] A statistical significance is $p < 0.017$.

against the other algorithms because the p-values are lower than 0.017 (Table 10). Moreover, there is no significant difference between *RateAdaptation* and *BufferLevel* algorithms because the p-value is greater than 0.017.

6. CONCLUSION AND FUTURE WORKS

In this paper, we presented a transmission schedule algorithm, *L2H*, for streaming DASH videos over residential networks with a small variation in bandwidth (such as a DSL-based network for IPTV). *L2H* was proved to be optimal based on many QoE metrics for capturing the intuition of QoE for an additional analysis, such as the maximal highest-resolution, the minimal number of resolution switching events, and the smoothness of the change in resolution. We then further improved *L2H* as $L2H_B$ by considering the system buffer constraints. Using $L2H_B$, high-resolution segments are transmitted as soon as possible to prompt users to continue watching. $L2H_B$ also outperformed the other algorithms based on many objective and subjective QoE user experience metrics. As future work, we intend to apply *L2H* and $L2H_B$ over a network with dynamic bandwidth variations (such as WiFi). Because the network bandwidth cannot be predicted in advance, some heuristics may be introduced in *L2H* or $L2H_B$.

ACKNOWLEDGMENT

This research is partially supported by Ministry of Science and Technology of Taiwan under grant MOST 102-2221-E-001-013-MY3, 102-2410-H-002-170-MY3, 104-2627-E-002-005, and Shih Chien University under grant USC-106-08-01001.

The Authors would like to thank Dr. Chi-Jen Wu for his helpful advice on various issues examined in this paper.

REFERENCES

1. Hulu, http://www.hulu.com.
2. Netflix, http://www.netflix.com.
3. YouTube, http://www.youtube.com.
4. Jeremy Lin 2014-2015 Highlight, https:// www. youtube. com/watch?v=rPHEzwWufW4, 2015.
5. Pixar Short Films #7 For the Birds, https://www.youtube.com/watch?v=tjUDblaW-Vc,

2000.
6. G. T. 26.247. Transparent end-to-end packet switched streaming service (PSS); Progressive download and dynamic adaptive streaming over HTTP (3GP-DASH), 2015.
7. V. K. Adhikari, Y. Guo, F. Hao, M. Varvello, V. Hilt, M. Steiner, and Z.-L. Zhang, "Unreeling netflix: Understanding and improving multi-CDN movie delivery," in *Proceedings of INFOCOM*, 2012, pp. 1620-1628.
8. A. C. Begen, T. Akgul, and M. Baugher, "Watching video over the web: Part 1: Streaming protocols," *Internet Computing*, Vol. 15, 2011, pp. 54-63.
9. Y. Cao, X. You, J. Wang, and L. Song, "A QoE friendly rate adaptation method for DASH," in *Proceedings of IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*, 2014, pp. 1-6.
10. X. Che, B. Ip, and L. Lin, "A survey of current youtube video characteristics," *IEEE MultiMedia*, Vol. 22, 2015, pp. 56-63.
11. Cisco visual networking index: Forecast and methodology, 2014-2019, 2015.
12. A. Donnelly and C. Smythe, "A tutorial on the digital audio-visual council (DAVIC) standard activity," *Electronics & Communication Engineering Journal*, Vol. 9, 1997, pp. 46-56.
13. J. Hao, R. Zimmermann, and H. Ma, "Gtube: Geo-predictive video streaming over http in mobile environments," in *Proceedings of the 5th ACM Multimedia Systems Conference*, 2014, pp. 259-270.
14. T. Hofeld, M. Seufert, C. Sieber, T. Zinner, and P. Tran-Gia, "Identifying QoE optimal adaptation of http adaptive streaming based on subjective studies," *Computer Networks*, Vol. 81, 2015, pp. 320-332.
15. J.-M. Jeong and J.-D. Kim, "Effective bandwidth measurement for dynamic adaptive streaming over http," in *Proceedings of International Conference on Information Networking*, 2015, pp. 375-378.
16. P. Kortum and M. Sullivan, "The effect of content desirability on subjective video quality ratings," *Human Factors: Journal of The Human Factors and Ergonomics Society*, Vol. 52, 2010, pp. 105-118.
17. S. Lederer, C. Mueller, C. Timmerer, C. Concolato, J. Le Feuvre, and K. Fliegel, "Distributed dash dataset," in *Proceedings of the 4th ACM Multimedia Systems Conference*, 2013, pp. 131-135.
18. Z. Li, A. C. Begen, J. Gahm, Y. Shan, B. Osler, and D. Oran, "Streaming video over http with consistent quality," in *Proceedings of the 5th ACM Multimedia Systems Conference*, 2014, pp. 248-258.
19. C. Liu, I. Bouazizi, and M. Gabbouj, "Rate adaptation for adaptive http streaming," in *Proceedings of the 2nd Annual ACM Conference on Multimedia Systems*, 2011, pp. 169-174.
20. R. K. Mok, X. Luo, E. W. Chan, and R. K. Chang, "Qdash: a QoE-aware DASH system," in *Proceedings of the 3rd Multimedia Systems Conference*, 2012, pp. 11-22.
21. C. Muller, S. Lederer, and C. Timmerer, "An evaluation of dynamic adaptive streaming over http in vehicular environments," in *Proceedings of the 4th ACM Workshop on Mobile Video*, 2012, pp. 37-42.
22. G. O'Driscoll, *Next Generation IPTV Services and Technologies*, John Wiley & Sons, 2008.
23. Q. M. Qadir, A. Kist, and Z. Zhang, "Mechanisms for QoE optimization of video traf-

fic: A review paper," *Australasian Journal of Information*, *Communication Technology and Applications*, Vol. 1, 2015, pp. 1-18.
24. ITU-T Rec., "Methods for subjective determination of transmission quality," *International Telecommunication Union*, Geneva, 1996.
25. I. Sodagar, "The MPEG-DASH standard for multimedia streaming over the internet," *IEEE Multimedia*, Vol. 4, 2011, pp. 62-67.
26. T. Stockhammer, "Dynamic adaptive streaming over http: standards and design principles," in *Proceedings of the 2nd Annual ACM Conference on Multimedia Systems*, 2011, pp. 133-144.
27. S. E. Taiwan, "So-net ADSL," http://www.so-net.net.tw/, 2015.
28. C. Telecom, http://broadband.hinet.net/fttx.do?s=8, 2015.
29. Y. Xiao, X. Du, and J. Zhang, "Internet protocol television (IPTV): The killer application for the next-generation internet," *IEEE Communications Magazine*, Vol. 45, 2007, pp. 126-134.
30. C. Zhou, X. Zhang, L. Huo, and Z. Guo, "A control-theoretic approach to rate adaptation for dynamic http streaming," in *Proceedings of IEEE International Conference on Visual Communications and Image Processing*, 2012, pp. 1-6.
31. Y.-C. Liu, W.-C. Chung, M.-H. Lee, K.-J. Wang, S.-H. Chang, C.-J. Wu, R.-I. Chang, and J.-M. Ho, "Scheduling of optimal DASH streaming," in *Proceedings of ACM International Conference on Internet of Things and Cloud Computing*, 2016, No. 39.
32. N. Bouten, S. Latré, J. Famaey, W. van Leekwijck and F. de Turck, "Network quality optimization for adaptive video streaming services," *IEEE Transactions on Multimedia*, Vol. 16, 2014, pp. 2281-2293.
33. C. Zhou, C. W. Lin, and Z. Guo, "mDASH: A Markov decision-based rate adaptation approach for dynamic HTTP streaming," *IEEE Transactions on Multimedia*, Vol. 18, 2016, pp. 738-751.
34. A. Bokani, M. Hassan, S. Kanhere, and X. Zhu, "Optimizing HTTP-based adaptive streaming in vehicular environment using Markov decision process," *IEEE Transactions on Multimedia*, Vol. 17, 2015, pp. 2297-2309.
35. I. Ayad, Y. Im, E. Keller, and S. Ha, "A practical evaluation of rate adaptation algorithms in http-based adaptive streaming," *Computer Networks*, Vol. 133, 2018, pp. 90-103.
36. P. Zhao, W. Yu, X. Yang, D. Meng, and L. Wang, "Buffer data-driven adaptation of mobile video streaming over heterogeneous wireless networks," *IEEE Internet of Things Journal*, Vol. 5, 2018, pp. 3430-3441.

Yu-Chi Liu (劉郁琪) was born in Kaohsiung, Taiwan, in 1992. She received the B.E. degree in Software Engineering from National Kaohsiung Normal University, Kaohsiung, Taiwan, in 2010, and the M.E. degree in Engineering Science and Ocean Engineering from National Taiwan University, Taipei, Taiwan, in 2014. In 2012, she joined Institute of Statistical Science, Academia Sinica. Taipei, Taiwan, as a Research Assistant. Her main areas of research interest are scheduling scheme for video stream, and quality of experience (QoE).

Meng-Huang Lee (李孟晃) received his B.S. and M.S. degrees in Electrical Engineering from National Cheng Kung University in 1987 and 1989, respectively, and his Ph.D. degree in Computer Science and Information Engineering from National Taiwan University in 1996. He is currently a Professor at the Department of Information Technology and Management, Shih Chien University. His research interests include multimedia systems, IPTV and computer networks.

Shin-Hung Chang (張信宏) received his Ph.D. and M.S. degrees in Computer Science and Information Engineering from National Taiwan University in 2005 and 1998. He received his B.S. degree in Computer Science and Information Engineering from Fu Jen Catholic University in 1996. He joined the Institute of Information Science (IIS), Academia Sinica as a Research Assistant from 1998-2005 and as a Post Coctor from 2005-2006. Dr. Chang has one Unite State patent and four Taiwan patents in computer screen recording technology. In the IIS, Dr. Chang also handled several technology transferring cases to industry. After his postdoctoral work, Dr. Chang joined an American company, MagnetoX Co. Ltd., as an R&D team manager for leading several multimedia application development projects and handling patent matters from 2006-2007. Dr. Chang joined Fu Jen Catholic University as an Assistant Professor in 2007. Dr. Chang hosted several Ministry of Science and Technology (MOST) research projects in Fu Jen Catholic University. Additionally, Dr. Chang is IEEE member and IEEE computer society member. Dr. Chang's research interests cover the integration of theory and application, including machine learning algorithm, network protocol design, multimedia application, media streaming, media codec algorithm, slot machine game model, and real time strategy (RTS) game AI.

Wei-Chun Chung (鐘緯駿) received the Ph.D. degree in Computer Science and Information Engineering, National Taiwan University, Taiwan. His research interests include big data, bioinformatics, and networking.

Kuan-Jen Wang (王冠仁) received his M.S. and B.S. degrees in Computer Science and Information Engineering from Fu Jen Catholic University in 2016 and 2015. Mr. Wang joined the Institute of Information Science (IIS), Academia Sincia as a research assistant in 2015. During his period of the institute, Mr. Wang joined several research projects of Ministry of Science and Technology (MOST), Taiwan. After earning his M.S. degree, Mr. Wang went to the military service of Taiwan in 2016. Mr. Wang joined a stock quotes company, SysJust Co. Ltd., in Taiwan as a software engineer for developing iOS app in 2017. Currently, he has developed several stock analysis applications for the key product of SysJust. Mr. Wang's research interests cover the media streaming, multimedia application, stock analysis, app development, and algorithm design.

Jan-Ming Ho (何建明) received his Ph.D. degree in Electrical Engineering and Computer Science from Northwestern University in 1989. He received his M.S. at Institute of Electronics of National Chiao Tung University in 1980 and his B.S. in Electrical Engineering from National Cheng Kung University in 1978. Dr. Ho joined the Institute of Information Science, Academia Sinica as an Associate Research Fellow in 1989, and was promoted to Research Fellow in 1994. In 2000-2003, he served as Deputy Director of the Institute. In 2004-2006, he had served as Deputy Director of IIS, Academia Sinica in 2000-2003, and as Director General of the Division of Planning and Evaluation, National Science Council. He co-founded Foresight Taiwan and Germination Initiative and Functional Units with Dr. Eugene Wong in 2007-2012. He has served as reviewer and think tank member of the Germination Initiative ever since. He had also served as advisor of Ministry of Education and its Division of IT education in 2012-2016. Dr. Ho's research interests cover the integration of theory and applications, including combinatorial optimization, multimedia network protocols, financial computing and bioinformatics. Dr. Ho also published results in the field of VLSI/CAD physical design.

Ray-I Chang (張瑞益) is a Professor of the Department of Engineering Science and Ocean Engineering, National Taiwan University, Taiwan. He has published nearly 300 papers in international/national conferences and journals. His current research interests include machine learning, multimedia networking, and real-time scheduling. Dr. Chang is a member of IEEE, CERP, TFTA, and IICM.