Scheduling of Optimal DASH Streaming

Yu-Chi Liu
National Taiwan University
r03525057@ntu.edu.tw

Wei-Chun Chung
Institute of Information Science, Academia Sinica
wcchung@citi.sinica.edu.tw

Meng-Huang Lee
Shih Chien University
meng@mail.usc.edu.tw

Kuan-Jen Wang
Fu Jen Catholic University
jeb20302@gmail.com

Shin-Hung Chang
Fu Jen Catholic University
shchang@csie.fju.edu.tw

Chi-Jen Wu
Institute of Information Science, Academia Sinica
cjw.ntu@gmail.com

Ray-I Chang
National Taiwan University
rayichang@ntu.edu.tw

Jan-Ming Ho
Institute of Information Science, Academia Sinica
hoho@iis.sinica.edu.tw

ABSTRACT

DASH (Dynamic Adaptive Streaming over HTTP) is now the most popular standard in video streaming. For supporting DASH video transmission over residential networks with small bandwidth variation (such as DSL based network for IPTV), we design an optimal transmission schedule, *L2H*. Given a transmission rate and an initial delay, the schedule can optimize the QoE (quality of experience) metrics such as rebuffering, lexicographically maximal resolution, number of rate switching events, and smoothness of the rate change. We further present $L2H_B$ for considering usage of the system buffer when applying the *L2H*. $L2H_B$ comes up with its benefit when comparing with other research work by objective QoE evaluations. Besides, by introducing the system buffer size constraint, the proposed algorithm can control the transmission schedule to let highest-resolution segments appear as soon as possible for prompting the users to stay tuned.

Categories and Subject Descriptors

C.2.3 [**Computer-Communication Networks**]: Network Operations; H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems

Keywords

DASH, Quality of Experience (QoE), scheduling, IPTV

1. INTRODUCTION

Mobile data traffic has been dominated by video streaming [11]. Many content providers deployed solutions for efficient transmission of multimedia stream to end-user. Especially for IPTV, an Internet application has the potential to overwhelm the Internet backbone and residential broadband access networks [7]. Beside, as noted by Driscoll et al. [22], IPTV stream operates on a stand-alone basis and the network uses the switch-video mode of operation. Therefore, the CBR technique is used for carrying video streams over DSL based networks.

To improve user experience of watching on-demand videos, content providers introduce various streaming technologies into their infrastructures. Microsoft Media Services (MMS) and Real Time Streaming Protocol (RTSP) are popular solutions used by the video streaming services. However, solutions based on HTTP over TCP are usually preferred owing to the following advantages [8]. First, this technology is cheaper to deploy into the existing HTTP infrastructure. Second, it is allowed penetrating through firewalls and network address translators (NAT). Third, it is easier to deploy over a Content Delivery Network (CDN).

Dynamic Adaptive Streaming over HTTP (MPEG-DASH, or DASH) [24, 25] is a pull-based solution for providing uninterrupted video streaming regardless of the network conditions and device capabilities. It has gained popularity among multimedia content providers, including Netflix [3], Hulu [2], and YouTube [5]. A video encoded in DASH format consists of a series of segments. Each segment contains video content of the same time, e.g., lasting a few seconds or tenths of seconds. Each video segment is present in several resolutions. A DASH server is a standard HTTP server. A client application may decide which resolution of the next segment to retrieve based on the state of its buffer occupancy and 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 best transmission schedule to the client without introducing annoying interruptions and poor visual quality. However, to the best of our knowledge, most studies deal with supporting DASH video streaming services on top of the best-effort Internet (i.e., OTT service), do not more comprehensive investigate in IPTV service with residential broadband access networks.

In this paper, we present the architecture and algorithms

ICC '16, March 22-23, 2016, Cambridge, United Kingdom
© 2016 ACM. ISBN 978-1-4503-4063-2/16/03…$15.00
DOI: http://dx.doi.org/10.1145/2896387.2896428

for supporting DASH CBR-encoded video [10, 18, 22, 26] streaming services over residential networks with small bandwidth variations (such as 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 QoE metrics. Note that QoE refers to user subjective opinions on the service quality, which is difficult to quantify. In this paper, we propose several QoE metrics for quantitatively modeling the notion of QoE, enabling its further analysis. We considered the QoE metrics such as rebuffering, lexicographically maximal resolution, the minimal number of rate switching events, and the smoothness of the rate change. We also present the *L2H* with the system buffer size constraints, $L2H_B$. Note that unlike the traditional notion of introducing the system buffer size constraints for modeling the limited amount of memory embedded in user devices, we show that by introducing the system buffer size constraints one may control the transmission schedule such that higher-resolution segments will appear as soon as possible, prompting the users to stay tuned. We also show in simulations that the proposed algorithms achieve better QoE than previously reported.

The remainder of this paper is organized as follows. Related work is introduced in Section 2. In Section 3, the MPEG-DASH scheduling problems and the QoE indices are presented. Then the proposed algorithms are discussed in Section 4. Simulation and performance evaluation are presented in Section 5. Finally, conclusions and future work are given in Section 7.

2. RELATED WORK

Most previous works only focused on DASH video streaming services on OTT service (i.e., dynamically changing network traffic). Qadir et al. [23] review the proposed mechanisms for the QoE optimization of video traffic. Examples include rate adaptation, cross-layer mechanisms, scheduling, and content and resource management. The majority of proposed rate adaptation algorithms are heuristic. Buffer-based adaptation algorithms [9, 13, 14, 20, 21, 27] estimate the system buffer size at the client end, and swap video presentations by using various thresholds. Yuming et al. [9] proposed a QoE friendly rate adaptation method which can achieve less rate switching events and the smoothness of the rate change. Muller et al. [21] is referred to as the BL algorithm in the remaining part of the paper, sets a 30 second buffer (i.e., 15 segments) to compensate high bandwidth fluctuations. Bandwidth-based [15, 19] algorithms switch resolutions depending on bandwidth; however, the throughput may not be optimal in an unstable network. Liu et al. [19] proposed a novel rate adaptation algorithm for adaptive HTTP streaming that detects bandwidth changes using a smoothed HTTP throughput measured based on the segment fetch time (SFT) but also does not consider resolution change amplitude, resolution switching event, and high resolution video. Lee et al. [17] proposed a QoE-aware scheduling approach in which feedback is collected from users for indicating network utilization. Then, a sigmoid-like user-centric QoE function is applied for improving the average user experience. Draxler et al. [12] proposed an anticipatory download scheduling algorithm for heuristic scheduling of playback sequences by estimating the user rate of data exchange; however, playback interruption still occurred.

This paper clearly differs from those related works mainly in two folds. First, we focus on DASH video streaming services over residential networks with small bandwidth variation (such as DSL based network for IPTV). Second, we present several QoE metrics to yield a quantitative model of QoE. Then, two versions of algorithm *L2H*, with and without the system buffer size constraints are designed for achieving better user experience.

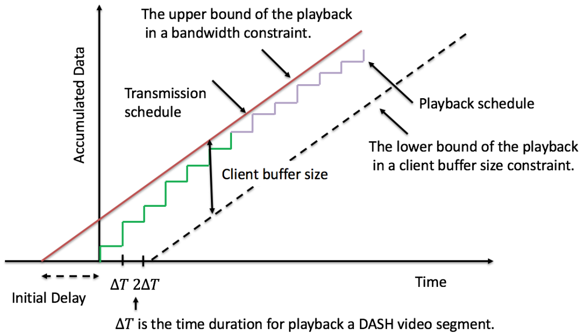
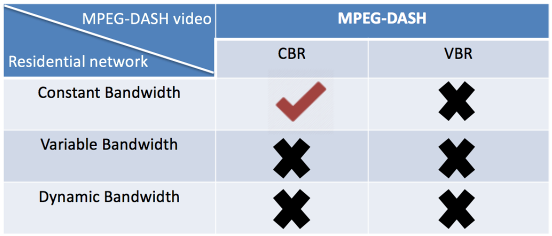

The upper bound of the playback in a bandwidth constraint.
Accumulated Data
Transmission schedule
Playback schedule
The lower bound of the playback in a client buffer size constraint.
Client buffer size
Initial Delay
ΔT 2ΔT
Time
ΔT is the time duration for playback a DASH video segment.

Figure 1: An example of uninterrupted playback schedule



MPEG-DASH video / Residential network	MPEG-DASH	
	CBR	VBR
Constant Bandwidth	✓	✗
Variable Bandwidth	✗	✗
Dynamic Bandwidth	✗	✗

Figure 2: The specification analysis of residential network and MPEG-DASH video

3. PROBLEM STATEMENT

In this section, we formulate the transmission schedule problem for achieving better user QoE. System parameters are also provided for further discussions.

3.1 Problem definition

Figure 1 shows an example of an uninterrupted transmission schedule. An initial delay T_p is designed for avoiding rebuffering by preloading data before playback. DASH video segments are loaded periodically after every ΔT time point. ΔT corresponds to the playback duration of a DASH video segment. A zigzag-like shape formed by the steps represents the transmission schedule. In this paper, we propose the architecture and algorithms for supporting DASH video streaming services over residential networks with small bandwidth variations (such as DSL based network for IPTV), with constant bit-rate DASH video segment playback (see Table 2). The algorithms are devised for generating a transmission schedule for a given transmission rate and initial delay, such that the QoE metrics are optimized. Note that QoE refers to user subjective opinions on the quality of service, which is difficult to quantify. Therefore, we considered the QoE metrics for quantitative modeling of QoE, to enable further analysis. The considered QoE metrics were rebuffering, lexicographically maximal resolution, the minimal number of rate switching events, and the smoothness of the rate change. In addition, we also considered the algorithm with the system buffer size constraints. Unlike the traditional notion of introducing the system buffer size constraints for modeling the limited amount of memory embedded in user devices, the algorithm controls the transmission schedule, such that higher-resolution segments appear as soon as possible, prompting the users to stay tuned.

Table 1: Notation of system parameters

Notation	Definition and description
s_{ik}	The i th segment at resolution k of a DASH video.
$\|s_{ik}\|$	The file size of segment s_{ik}.
V	The set of s_{ik} 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 time duration for playback a DASH video segment.
$S(V)$	The playback schedule of a DASH video V. The playback unit is s_{ik}.
n_k	The number of segments of $S(V)$ at resolution k.
RV	The set of n_k which is the resolution vector in a feasible schedule of a DASH video.
m_N	The slope of the start point of lowest resolution to the last point of highest resolution.
m_B	The slope of the network bandwidth.
m_R	The slope of the lowest resolution.

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

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

When considering CBR encoding, it is assumed that every file at the same resolution in each segment has the same size. That is, $|s_{ik}| = |s_{j,k}|$, where $1 \leq i, j \leq N$. In addition, a feasible playback schedule is denoted as follows.

$$S(V) = \{s_{ik}\}, \quad 1 \leq i \leq N. \tag{2}$$

We also define *resolution vector* (RV) to compare QoE level between schedules by using lexicographic order. A RV is a vector consists of the number of each resolutions n scheduled for playback, and is sorted by resolution k in descending order. The notation of RV is formulated as $RV = (n_1, ..., n_R)$. Thus, in our model, a schedule $S(V)$ is better than $S(V)'$ if and only if $RV_{S(V)}$ is lexicographically greater than $RV_{S(V)'}$.

4. FEASIBLE SCHEDULE AND QOE OPTIMIZATION

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

4.1 System architecture

Figure 3 shows the architecture of our proposed system. There are three major roles, i.e., the DASH client, the CBR-MR video server (i.e., HTTP server), and the *L2H* Scheduling Agent. To optimize the given QoE metrics, we first focus on transmitting DASH videos over networks with constant bandwidth. The *L2H* Scheduling Agent is designed for analyzing the status of the transmission network for making playback decisions to prompt the users to stay tuned.

4.2 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 a constant playback duration of a video segment. The overall time for transmitting a DASH video is $T_p + N * \Delta T$. B is assumed a given constant network bandwidth for transmission. Thus, the data size transmitted during the video playback can be defined as follows:

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

Let n_1 stand for the number of segments with the highest resolution (each with the size $|s_{i1}|$), n_{iR} for the number of segments with the lowest resolution (each with the size $|s_{iR}|$), and n_k for the number of segments with intermediate resolutions. The data size of a video with N segments is:

$$n_1 * |s_{i1}| + n_k * |s_{ik}| + (N - n_1 - n_k) * |s_{iR}|. \tag{4}$$

To generate a transmission schedule without interruptions, the expression in Eq. (3) should be larger than the expression in Eq. (4). Thus, we developed a heuristic algorithm (Algorithm 1) to discover a *feasible resolution vector* that a *feasible transmission schedule* can be obtained from this resolution vector. The algorithm iterates through resolutions to find a segment with an appropriate resolution for playback (lines 2 to 9). If buffer underflow occurs, the algorithm stops the search and returns the resolution vector (lines 7 to 9). Therefore, the resolution vector output by Algorithm features the most number of highest resolution segments.

4.3 Feasible Transmission Schedule of Algorithm L2H

The *L2H* (Algorithm 2) is designed to generate a *feasible transmission schedule* based on the rearrangement of the resolution vector. Figure 4 shows a possible feasible schedule generated by *L2H* . We assume that a *feasible schedule* advances from the lowest resolution n_R to the highest resolution n_1. Note that a 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 by the following theorems:

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

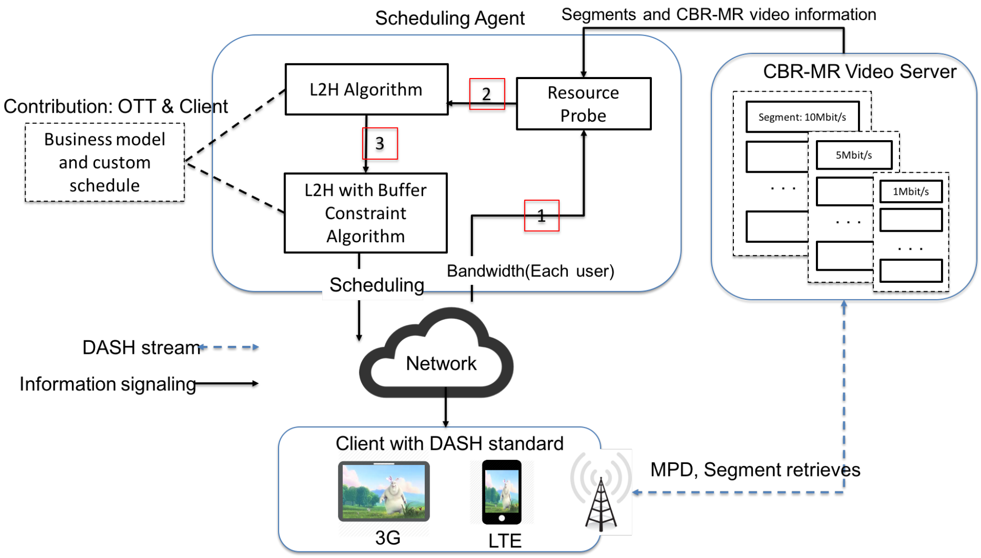


Figure 3: The architecture of our proposed framework for optimizing QoE indices of MPEG-DASH video

Algorithm 1: Finding a resolution vector

```
Input: Residential network bandwidth, $B$; Segment information of a MPEG-DASH video, $s_{ik}$
Output: A resolution vector, $RV$
$totalTranData = R*(T_p + N*\Delta T)$;
for each $k \in [1,R]$ do
    $totalComsumeData = \sum_{j=1}^{k} n_j * |s_{ij}|$;
    $n_k = \lfloor (totalTranData - totalComsumeData - (N - \sum_{j=1}^{k} n_j) * |s_{iR}|)/(|s_{ik}| - |s_{iR}|) \rfloor$;
    $n_R = N - \sum_{i=1}^{k} n_i$;
    $RV \leftarrow n_k, n_R$ ;
    if $(totalTranData - totalComsumeData) < |s_{iR}|$ then
        Break and output $RV$ ;
    end
end
return $RV$;
```

Algorithm 2: *L2H* algorithm

```
Input: A resolution vector, $RV$
Output: A feasible playback schedule, $S(V)$
$i \leftarrow 1$;
for each $j \in [1,R]$ do
    $n_j \leftarrow RV$;
    for $x \leftarrow 1$ to $n_j$ do
        $S(V) \leftarrow s_{ij}$ ;
        $i{+}{+}$;
    end
end
return $S(V)$;
```

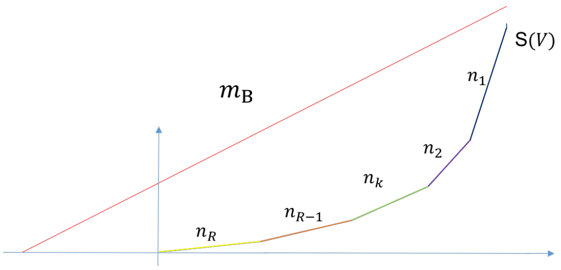


Figure 4: An example to illustrate the *L2H* algorithm

PROOF. We prove this theorem by contradiction.

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

□

THEOREM 2. *The feasible transmission schedule generated by the L2H guarantees jitter-free playback.*

PROOF. Proof by induction. The proof includes the following steps:

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

We first connect the start point with the lowest resolution and the end point with the highest resolution, to form a virtual dash line m_N as shown in Figure 5. The line m_N is given

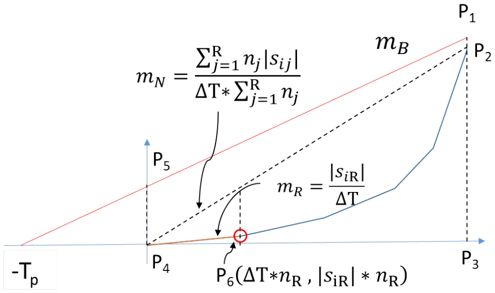

P₁
m_B
P₂
$m_N = \frac{\sum_{j=1}^{R} n_j |s_{ij}|}{\Delta T * \sum_{j=1}^{R} n_j}$
P₅
$m_R = \frac{|s_{iR}|}{\Delta T}$
-T_p
P₄
P₆(ΔT∗n_R , |s_iR| ∗ n_R)
P₃

Figure 5: Schematic diagram for the proof of Step 1 in Theorem 2

by $(\frac{\sum_{j=1}^{R} n_j |s_{ij}|}{\Delta T * \sum_{j=1}^{R} n_j})$ which is lower than $(\frac{|s_{iR}| \sum_{j=1}^{R} n_j}{\Delta T * \sum_{j=1}^{R} n_j})$. However, m_R is equal to $(\frac{|s_{iR}|}{\Delta T})$, also as $(\frac{|s_{iR}| \sum_{j=1}^{R} n_j}{\Delta T * \sum_{j=1}^{R} n_j})$. Therefore, we can say that $m_N > m_R$.

In a specific region (e.g., $\square P_1 P_3 P_4 P_5$), assume that $\overline{P_1 P_5}$ does not intersect $\overline{P_2 P_4}$. In this case, our schedule satisfies $B * (T_p + (n_k - 1)) * \Delta T \geq \sum_{j=1}^{R} n_j |s_{ij}|$. Therefore, $m_B > m_N > m_R$ and we proved that the playback duration of the segments with the resolution n_R guarantees jitter-free playback.

Step2: We prove that if the playback duration of the segments with the resolution n_{k+1} guarantees jitter-free playback, then the playback duration of the segments with the resolution n_k guarantees jitter-free playback as well.

We first draw two parallel lines ($\overline{P_1' P_4'}$ and $\overline{P_1 P_4}$) that intersect the X axis at right angles (see Figure 6). Then, we extend the segments s_{ik} to obtain the intersection points P_3 and P_3' on the lines $\overline{P_1 P_4}$ and $\overline{P_1' P_4'}$, respectively. To verify that the data consumption in each time slot does not exceed the transmitted data, we calculate the slopes of $m_{\overline{P_3 P_2}}$ and $m_{\overline{P_3 P_3'}}$. $m_{\overline{P_3 P_2}}$ is equal to $(\frac{\sum_{j=k}^{1} n_j |s_{ij}|}{\Delta T * \sum_{j=k}^{1} n_j})$, which is lower than $(\frac{|s_{ik}| \sum_{j=k}^{1} n_j}{\Delta T * \sum_{j=k}^{1} n_j})$. However, the slope $m_{\overline{P_3 P_3'}}$ is equal to $(\frac{|s_{ik}| \sum_{j=k}^{1} n_j}{\Delta T * \sum_{j=k}^{1} n_j})$. Therefore, we can say that $m_{\overline{P_3 P_2}} > m_{\overline{P_3 P_3'}}$.

In a specific region (e.g., $\square P_1' P_3' P_3 P_1$), assume that $\overline{P_1' P_1}$ does not intersect $\overline{P_3 P_3'}$. . In this case, our schedule satisfies $B * (T_p + (n_k - 1)) * \Delta T \geq \sum_{j=1}^{R} n_k |s_{ij}|$. Therefore, $m_B > m_{\overline{P_2 P_3}} > m_{\overline{P_3' P_3}}$ 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, by using mathematical induction, we proved that the feasible transmission schedule generated by the *L2H* algorithm guarantees jitter-free playback.

□

THEOREM 3. *The feasible transmission schedule generated by the 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

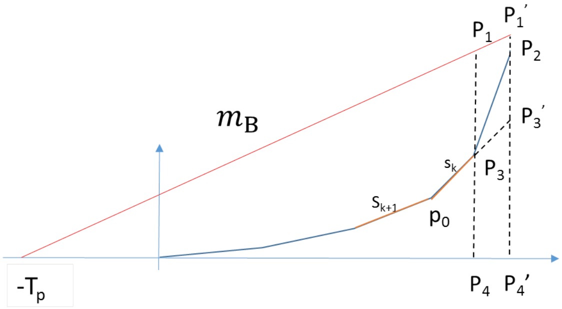


Figure 6: Schematic diagram for the proof of Step 2 in Theorem 2

the resolutions of two adjacent segments are different. The definition is:

$$g^* = \sum_{i=2}^{N} g(s_{ik}) \quad g(s_{ik}) = \begin{cases} 1 & \text{if } f(s_{i-1k}) \neq f(s_{ik}) \\ 0 & \text{else} \end{cases} \tag{5}$$

Note that $f(s_{ik})$ corresponds to the resolution k of i-th segment, R denotes the number of resolutions, and n_k is the number of segments in the 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 for the *L2H* is $R-1$. Below, we prove that the *L2H* generates the least number of switching events.

PROOF. We prove this theorem by 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, $g^* - 1 < R$ uses less than R resolutions. It is a contradiction and thus the theorem is proved.

□

4.4 Algorithm L2H with the System Buffer Size Constraints

Algorithm 1 outputs a resolution vector that serves as the input to *L2H*. Without considering the system buffer size constraints, *L2H* yields a *feasible transmission schedule* for the feasible transmission schedule, ordered from the lowest- to 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 will also guarantee obtaining optimal QoE metrics. $L2H_B$ 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, output of Algorithm 1, $L2H_B$ continuously selects 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 so $L2H_B$ selects video segments into $S(V)$ from low-resolution first. If the system buffer overflow occurs, then

$L2H_B$ will switch the segment selection order from low- to high-resolution by setting 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 $\beta \geq B*\Delta T+|s_{i1}|-|s_{iR}|$.*

PROOF. A transmission schedule without buffer overflow and buffer underflow is called feasible.

$$0 \leq BO_i \leq \beta - B*\Delta T \qquad (6)$$

where BO_i indicates the system buffer occupancy while playing the i^{th} segment of a DASH video, $B*\Delta T$ is the buffer data increment during each playback duration, and β is the system buffer size constraint. In addition, we know that BO_i is equal to $BO_{i-1}+B*\Delta T-|s_{ik}|$, where $1 \leq k \leq R$. Therefore, Eq. (7) is derived:

$$0 \leq BO_i + B*\Delta T - |s_{ik}| \leq \beta - B*\Delta T \qquad (7)$$

Assume that $L2H_B$ transmits the highest-resolution segment, $|s_{i1}|$, and the buffer underflow is avoided at the playback of the i^{th} video segment. Then, Eq. (8) can be derived from the left-hand side of Eq. (7).

$$|s_{ik}| - B*\Delta T \leq BO_{i-1} \qquad (8)$$

Assume that $L2H_B$ transmits the highest-resolution segment, $|s_{iR}|$, and the buffer underflow is avoided at the playback of the i^{th} video segment. Then, Eq. (9) is derived from the right-hand side of Eq. (7).

$$BO_{i-1} + B*\Delta T - |s_{iR}| \leq \beta - B*\Delta T \qquad (9)$$

Substituting $|s_{i1}| - B*\Delta T$ of Eq. (8) into BO_{i-1} of Eq. (9), Eq. (10) is derived:

$$|s_{i1}| - |s_{iR}| + B*\Delta T \leq \beta \qquad (10)$$

$L2H_B$ will not transmit a video segment whose size is larger than $|s_{i1}|$ or smaller than $|s_{iR}|$ while playing the i^{th} video segment, because $|s_{iR}| \leq |s_{ik}| \leq |s_{i1}|$, where $1 \leq k \leq R$. Therefore, the $L2H_B$ yields a *feasible transmission schedule* when the system buffer size constraint is $\beta \geq B*\Delta T + |s_{i1}| - |s_{iR}|$.

□

4.5 An example for L2H

The different bitrates and resolutions of the dataset are chosen from a real adaptive streaming system [16], as shown in Table 2 where for each segment length the content was encoded at 6 different video representations, ranging from 100 Kbps at 320x240 up to 3 Mbps at 1920x1080. The playback duration of the segment is 2 seconds. Therefore, the file sizes for the different segments are 200, 800, 1,800, 3,000, 5,000, 6,000 Kbits. We also assume the *feasible initial delay* of 1 second and the video length of 1 minute (i.e., 30 segments). Thus, the maximal amount of data is 147,500 Kbits (i.e., 2500 Kbps*(1 s +(30-1)* 2 s)= 147500 Kbits). To compute n_1, we start by considering the video size to be

Algorithm 3: *L2H* with buffer size constraint ($L2H_B$)

```
  Input: A resolution vector, $RV = (n_1 ... n_R)$;
         Residential network bandwidth, $B$; Playback
         duration, $\Delta T$; A buffer size constraint, $\beta$ ;
  Output: A feasible transmission schedule, $S(V)$
1  if $\beta < |s_{i1}| - |s_{iR}| + B*\Delta T$ then
2      return null ;
3  end
4  $S(V) \leftarrow \emptyset$ ;
5  $d \leftarrow 0$ ;
6  for $i \leftarrow 1$ to $N$ do
7      switch $d$ do
8          case 0:
9              tr ← $max\{k|n_k > 0$ and $1 \leq k \leq R\}$ ;
10             break;
11         end
12         case 1:
13             tr ← $min\{k|n_k > 0$ and $1 \leq k \leq R\}$ ;
14             break;
15         end
16     endsw
17     if $s_{i,t}$ causes buffer overflow or buffer underflow
       then
18         switch $d$ do
19             case 0:
20                 d ← 1 ;
21                 break;
22             end
23             case 1:
24                 d ← 0 ;
25                 break;
26             end
27         endsw
28     end
29     S(V) ← $s_{i,t}$ ;
30     $n_k \leftarrow (n_k - 1)$ ;
31 end
32 return S(V) ;
```

$n_1 * 6000Kbits + n_k * |s_k| + (30 - n_1 - n_k) * 200Kbits$. To avoid rebuffering, the total size of the streaming video should be lower than the maximal amount of the transmitted data: n_1*6000 Kbits+n_k*$|s_k|$+ (30$-n_1-n_k$)* 200 Kbits $\leq$ 147500 Kbits. The number of the highest resolution n_1 is 24 (i.e. $\lfloor$(147500 $-$ 200 Kbits $*$ 30)/(6000 Kbits $-$ 200 Kbits)$\rfloor$ = 24). Note that the intermediate resolution n_k is set to 0, and gets the most number of n_1. The n_R of the lowest resolution is 6 (i.e., 30 âĹŠ 24 = 6). However, the difference between the transmission data (i.e., 147500 Kbits) and the size of the transmitted data (i.e., 24∗6000 Kbits + 6∗200 Kbits = 145200 Kbits) is 2,300 Kbits. It is larger than the segment size of the lowest resolution n_R. It implies that an intermediate resolution can be used in the schedule. Then, we iterate over a lower resolution n_4 whose value is equal to 1 (i.e., $\lfloor$(147500 $-$6000Kbits∗24 $-$ 200Kbits $*$ (30-24))/ (1800Kbits $-$ 200Kbits)$\rfloor$ = 1), and resolution n_5 whose value is equal to 1 (i.e., $\lfloor$(147500 $-$6000Kbits∗24 $-$ 1800Kbits∗1 $-$ 100Kbits $*$ (30-25))/ (800Kbits $-$ 200Kbits)$\rfloor$ = 1). Here, the resolution vector is (24, 0, 0, 1, 1, 4). Finally, the transmission schedule produced by the *L2H* is $\{s_{16}$, s_{26}, s_{36}, s_{46}, s_{55}, s_{64}, s_{71}, s_{81}, s_{91}, s_{101}, s_{111}, s_{121}, s_{131}, s_{141}, s_{151}, s_{161}, s_{171}, s_{181}, s_{191}, s_{201}, s_{211}, s_{221}, s_{231}, s_{241}, s_{251}, s_{261}, s_{271}, s_{281}, s_{291}, s_{301} $\}$.

Table 2: bitrate level in real adaptive streaming system.

video type	bitrate
320x240	100 kbps
480x360	400 Kbps
720x480	900 Kbps
720x576	1500 Kbps
1280x720	2500 Kbps
1920 x 1080	3000 Kbps

5. PERFORMANCE EVALUATION

In Section 4, a *feasible transmission schedule* is generated by *L2H* or $L2H_B$. Some QoE metrics are proved to be optimal for the feasible transmission schedule. In this section, we further perform the objective QoE evaluation and subjective QoE evaluation to show that the $L2H_B$ is outperformed than other research work.

The QoE metrics for DASH have been defined in 3GPP DASH specification TS26.247 [6], to be measured by the client upon activation by the server. Noted that these metrics are specific to DASH over the HTTP, and therefore differ from QoE metrics for traditional push-based streaming protocols. The following performance evaluation metrics were used for the evaluation. The first metric is from 3GPP DASH specification TS 26.247. The other two metrics are defined by this paper.

1. *Average playback bitrate per user* : it is defined as the total segments bitrate divided by the total number of segments. For example, the playback resolution vector is $\{n_1 = 1, n_2 = 2, n_3 = 1\}$, the total of segments is 4 ($N, 1 + 2 + 1 = 4$), and s_1,and s_2, and the bitrate of s_3 are 300, 200, and 100Kbps, respectively. Therefore, the average playback time is $\frac{\sum_{j=1}^{4} n_j s_{ij}}{N} = \frac{300Kbps + 200Kbps*2 + 100Kbps}{4} = \frac{800Kbps}{4} = 200Kbps$.

2. *Lexicographic* comparison between resolution vector: denote resolution vectors of two feasible schedules $S_a(V)$ and $S_b(V)$ of CBR DASH video V as $A = (n_1, n_2, ..., n_R)$ and $B = (n'_1, n'_2, ..., n'_R)$. If A is lexicographically greater than B, then $S_a(V)$ is said to give better QoE than does $S_b(V)$.

3. *Ratio of bandwidth utilization*: it refers to the ratio of the total bitrate of selected quality representation to the actual network bandwidth. For example, the video length is 10sec, the network bandwidth is 500Kbps, the playback resolution vector is ($n_1 = 1$,$n_2 = 4$,$n_3 = 5$), s_1, s_2, and s_3 bitrate are 300, 200,and 100Kbps, respectively. The playback time for each segment is 2 seconds. Therefore, bandwidth utilization is $\frac{300*1*1+200*1*4+100*1*5}{500*1*10} * 100\% = 32\%$.

6. OBJECTIVE QOE EVALUATION

Figure 6 investigates the performance of $L2H_B$ (curve indicated in yellow), *BufferLevel* (curve indicated in black), and *RateAdaptation* (curve indicated in cyan). The x-axis indicates video playback time (sec) and the y-axis is the cumulative data (bits) received on the client side. We use 6 different bitrates (i.e., 100, 400, 900, 1500, 2500, 3000 Kbps) of content bitrate version as the profile in Table 3. Each video is divided into segments of small fixed-length with playback duration of 2 seconds playback duration. We test a 10 minutes long video (i.e., 300 segments) and set up the initial delay to 1 second. We set the system buffer for 20 seconds (i.e., 10 segments). Moreover, the system buffer size constraint is set to $B * \Delta T + |s_{i1}| - |s_{iR}|$. For instance, if the network bandwidth is 1 Mbps, the system buffer size is $\frac{1000Kbps*2sec+6000Kbits-200Kbits}{1000Kbps} = 7.98sec$. The network bandwidth is set as 1 Mbps, 2 Mbps, and 3 Mbps to simulate the ADSL bandwidth provided by current Internet Service Providers (ISPs) in Taiwan (e.g., HiNet [1], So-net [4], etc.).

There are three prominent insights in this evaluation. First, $L2H_B$ has the highest bandwidth utilization, average playback bitrate, and lexicographic order, among all algorithms (see Table 3, Table 4, and Table 5). Second, *RateAdaptation* has a lower average bitrate as compared to the other algorithms. This is because it uses a conservative stepwise switch-up and aggressive switch-down strategy to avoid playback interruption, despite the fact that the network bandwidth is sufficient for transmitting a higher resolution segment. Therefore, in the scenario of small bandwidth variation, these results show that *RateAdaptation* is too conservative to select a higher bitrate segment, and leads to a lower average bitrate and bandwidth utilization. Third, different DASH video bitrate datasets and the ADSL network bandwidths would affect the playback sequence of *BufferLevel.* In the initial buffering time (i.e., 1 second), the number of segments in a buffer grows before the video starts playing. Based on *BufferLevel*, the selected bitrate of video segment increases as the buffer level increases. It implies that the playback bitrate will be affected by the buffer level and the network bandwidth. From Figure 7 to Figure 9, the cumulative data dramatically increased in the start-up period of *BufferLevel*, then it will enter a steady-state. This steady-state meant that the system fetched the same one or two resolutions during a single slot time. We found that the difference of the network bandwidth will effect the bandwidth utilization of *BufferLevel*. More specifically, if the steady-state is more closed to segment duration, the bandwidth utilization will higher. For an example, the steady-state of 1, 2 and 3 Mbps network bandwidths were 1.8 sec (i.e., a segment size is divided by the network bandwidth; $\frac{1800Kbits}{1000Kbps}$), 1.5 and 2.5 sec, and 2 sec, respectively. The bandwidth utilization of 3 Mbps was highest, and 1 Mbps was smallest.

Table 3: QoE metrics for the three algorithms, for the network bandwidth of 1 Mbps.

Algorithm	**Avg. bitrate (Kbps)**	**Lex. order**	**Bandwidth utilization**
$L2H_B$	998.33	1	100%
BufferLevle	969.67	2	97.13%
RateAdaptation	400	3	40.13%

Table 4: QoE metrics for the three algorithms, for the network bandwidth of 2 Mbps.

Algorithm	**Avg. bitrate (Kbps)**	**Lex. order**	**Bandwidth utilization**
$L2H_B$	1996.67	1	100%
BufferLevle	1971.3	2	98.73%
RateAdaptation	896.66	3	44.98%

Table 5: QoE metrics for the three algorithms, for the network bandwidth of 3 Mbps.

Algorithm	**Avg. bitrate (Kbps)**	**Lex. order**	**Bandwidth utilization**
$L2H_B$	2995	1	100%
BufferLevle	2964	2	98.96%
RateAdaptation	896.66	3	29.98%

7. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed algorithms for transmission of DASH videos over residential networks with small bandwidth variation (such as DSL based network for IPTV). We presented QoE metrics, such as lexicographically maximal resolution, the minimal number of rate switching events, and the smoothness of the rate change, for capturing the intuition of QoE for additional analysis. Unlike the traditional notion of introducing system buffer size constraints for prevention of resource shortage in user devices, we showed that by introducing the system buffer size constraints, a transmission schedule can be controlled, such that high-resolution segments appear as soon as possible, prompting the users to stay tuned. Experimental results show that the proposed algorithms with known constant bandwidth properties of the underlying network might yield better QoE than other research works that either are not based on this knowledge or do not properly utilize the information. The experiments demonstrated that, since the QoE metrics require the algorithm to generate many video segments with the highest possible resolution, users streaming videos may experience dramatic resolution changes at the resolution-switching points.

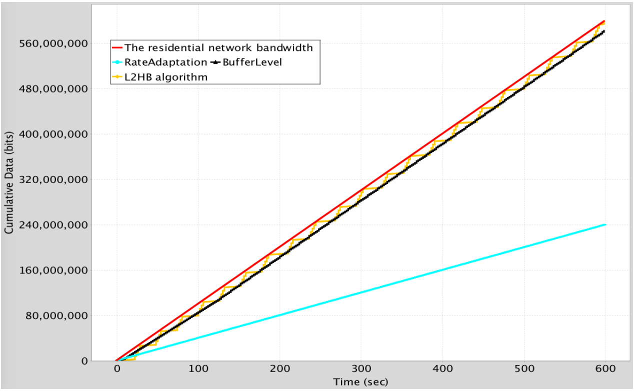


Figure 7: Performance comparison of $L2H_B$, *BufferLevel*, and *RateAdaptation*, for the network bandwidth at 1 Mbps.

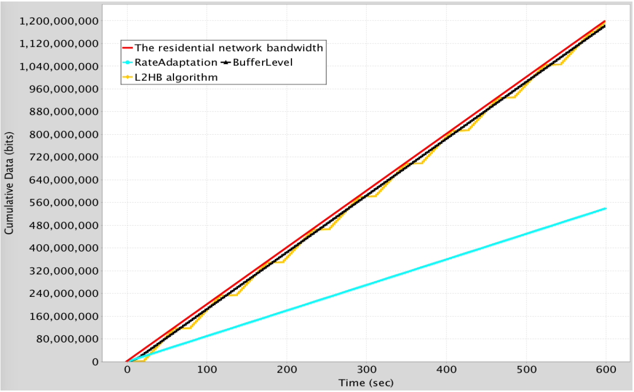


Figure 8: Performance comparison of $L2H_B$, *BufferLevel*, and *RateAdaptation*, for the network bandwidth at 2 Mbps.

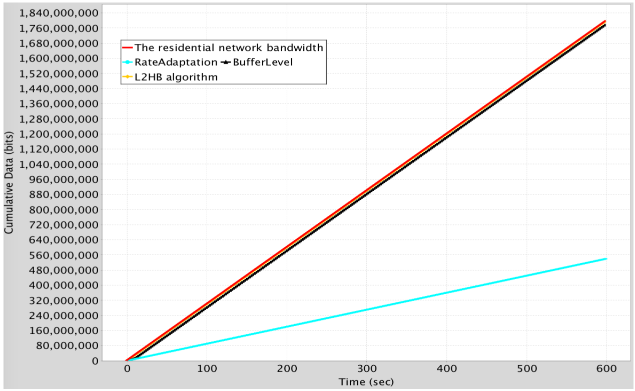


Figure 9: Performance comparison of $L2H_B$, *BufferLevel*, and *RateAdaptation*, for the network bandwidth at 3 Mbps.

However, the number of resolution-switching points and the extent of change in the resolution can be reduced by choosing a subset of resolutions, for matching the given network bandwidth. We plan to extend the proposed algorithms to account for bandwidth variations in residential and mobile networks.

8. 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 Academia Sinica Center for Digital Cultures.

9. REFERENCES

[1] Hinet ADSL. http://broadband.hinet.net/fttx.do?s=8, 2015.

[2] Hulu. http://www.hulu.com, 2015.

[3] Netflix. http://www.netflix.com/, 2015.

[4] So-net. http://www.so-net.net.tw/, 2015.

[5] YouTube. http://www.youtube.com, 2015.

[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. [Online; accessed December-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 *INFOCOM, 2012 Proceedings IEEE*, pages 1620–1628. IEEE, 2012.

[8] A. C. Begen, T. Akgul, and M. Baugher. Watching video over the web: Part 1: Streaming protocols. *Internet Computing, IEEE*, 15(2):54–63, 2011.

[9] Y. Cao, X. You, J. Wang, and L. Song. A qoe friendly rate adaptation method for dash. In *Broadband Multimedia Systems and Broadcasting (BMSB), 2014 IEEE International Symposium on*, pages 1–6. IEEE, 2014.

[10] X. Che, B. Ip, and L. Lin. A survey of current youtube video characteristics. *MultiMedia, IEEE*, 22(2):56–63, 2015.

[11] I. Cisco. Cisco visual networking index: Forecast and methodology, 2014–2019, 2015.

[12] M. Draxler, J. Blobel, P. Dreimann, S. Valentin, and H. Karl. Smarterphones: Anticipatory download scheduling for wireless video streaming. In *Networked Systems (NetSys), 2015 International Conference and Workshops on*, pages 1–8. IEEE, 2015.

[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*, pages 259–270. ACM, 2014.

[14] T. Hoßfeld, M. Seufert, C. Sieber, T. Zinner, and P. Tran-Gia. Identifying qoe optimal adaptation of http adaptive streaming based on subjective studies. *Computer Networks*, 81:320–332, 2015.

[15] J.-M. Jeong and J.-D. Kim. Effective bandwidth measurement for dynamic adaptive streaming over http. In *Information Networking (ICOIN), 2015 International Conference on*, pages 375–378. IEEE, 2015.

[16] 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*, pages 131–135. ACM, 2013.

[17] G. Lee, H. Kim, Y. Cho, and S.-H. Lee. Qoe-aware scheduling for sigmoid optimization in wireless networks. *Communications Letters, IEEE*, 18(11):1995–1998, 2014.

[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*, pages 248–258. ACM, 2014.

[19] C. Liu, I. Bouazizi, and M. Gabbouj. Rate adaptation for adaptive http streaming. In *Proceedings of the second annual ACM conference on Multimedia systems*, pages 169–174. ACM, 2011.

[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*, pages 11–22. ACM, 2012.

[21] C. Müller, S. Lederer, and C. Timmerer. An evaluation of dynamic adaptive streaming over http in vehicular environments. In *Proceedings of the 4th Workshop on Mobile Video*, pages 37–42. ACM, 2012.

[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 optimisation of video traffic: A review paper. *Australasian Journal of Information, Communication Technology and Applications*, 1(1):1–18, 2015.

[24] I. Sodagar. The mpeg-dash standard for multimedia streaming over the internet. *IEEE MultiMedia*, (4):62–67, 2011.

[25] T. Stockhammer. Dynamic adaptive streaming over http–: standards and design principles. In *Proceedings of the second annual ACM conference on Multimedia systems*, pages 133–144. ACM, 2011.

[26] Y. Xiao, X. Du, and J. Zhang. Internet protocol television (iptv): the killer application for the next-generation internet. In *Institute of Electrical and Electronics Engineers*, 2007.

[27] C. Zhou, X. Zhang, L. Huo, and Z. Guo. A control-theoretic approach to rate adaptation for dynamic http streaming. In *Visual Communications and Image Processing (VCIP), 2012 IEEE*, pages 1–6. IEEE, 2012.