

# Stepwise Adaptive Video Streaming in the Wireless Mobile Network Using the Temporal-Geo Bandwidth Estimation Method\*

CHUNG-MING HUANG AND RUI-XIAN WEI  
*Department of Computer Science and Information Engineering*  
*National Cheng Kung University*  
*Tainan, 701 Taiwan*  
*E-mail: {huangcm; weirx}@locust.csie.ncku.edu.tw*

This work proposed a method called Stepwise Adaptive Streaming using Temporal-Geo Bandwidth Estimation (SASTGEBE) for the wireless mobile networking environment based on MPEG Dynamic Adaptive Streaming over HTTP (MPEG-DASH). To have better Quality Of Experience (QoE), *e.g.*, no or fewer quality switching, lower bitrate difference between two continuous video segments, no or fewer suspended times, no or shorter paused time, *etc.*, (1) the proposed method considers location, time, and date for estimating the available bandwidth in the future and (2) the proposed stepwise adaptive streaming control scheme considers buffer level, video quality of the most recently downloaded segment, the downloading rate of the most recently downloaded segment and the estimated bandwidth to decide the video quality for the next downloaded segment. The proposed method has been implemented in the Android system for the client side and the Linux system for the server side. The experiments using SASTGEBE in the real environment shown that SASTGEBE has improvement in the performance of bandwidth utilization, suspended times, quality switch percentage, average bitrate difference considering suspending, and standard deviation of bitrate difference over the wireless mobile network.

**Keywords:** mobile video, temporal-geo bandwidth map, MPEG-DASH, adaptive streaming, location-based service (LBS)

## 1. INTRODUCTION

Video streaming over the Hyper Text Transfer Protocol (HTTP) in wireless mobile network has become a hot research issue [1]. To have good adaptive streaming, the main concern is how to predict the network condition to adjust video quality. The estimated bandwidth method that is normally used is based on the exponential weighted moving average (EWMA) way. Using the EWMA way, the weight of the past measured bandwidth is decayed exponentially. That is, the estimated bandwidth at time  $t$   $B_E^t$  is equal to  $(1 - \alpha)BW_E^{t-1} + \alpha BW_{cm}^t$ , for which  $B_E^t$  denotes the estimated bandwidth that is to be used at time point  $t$  and  $BW_{cm}^t$  is the currently measured bandwidth at time point  $t$ . But it also needs to consider the factor of QoE, *e.g.*, no or fewer quality switching, lower bitrate difference between two continuous video segments, *etc.* QoE is one of the major issues to be tackled in video streaming [1, 2]. When users are watching streaming video, not only the low video quality but also the sharp and/or frequent quality switches make users feel awful. That is, a smooth or step by step video quality switch will make users feel better than an abrupt video quality switch. Thus, video streaming needs a reasonable adaptive streaming control

Received January 31, 2019; accepted April 1, 2019.

Communicated by Ren-Hung Hwang.

\* This work was supported by the Ministry of Science and Technology (MOST), Taiwan under the Grant No. MOST 106-2221-E-006-029.

scheme, which should not only consider the measured bandwidth but also QoE to decide which video quality to download for the next video segment.

In order to have better QoE, some work presented some design principles for having adaptive streaming algorithms [3, 4]. These works considered the current buffering situation and past available bandwidth to get better QoE based on the temporal concern. However, these works are not so suitable for the adaptive video streaming in the wireless mobile environment because mobile devices are moving and thus the connected base stations (BSs) are changing and the attached Internet segments are changing, *i.e.*, the location is changing. Thus, the corresponding adaptive video streaming should consider more factors, *e.g.*, the distance between the mobile device and BS, the number of mobile devices that each BS is serving, the capacity of a BS, and the moving speed of mobile devices [5]. The principle is as follows. The recorded time of the past available bandwidth  $X$  is much nearer the time for bandwidth prediction, the more weight  $X$  has for the estimated bandwidth of the current time  $t$ ; the shorter distance between the location of recording the past measured bandwidth  $X$  and the current location of the mobile device is, the more weight  $X$  has for the estimated bandwidth of the current time  $t$ . That is, the past measured bandwidth associated with location can also be considered to calculate the estimated bandwidth to improve the QoE of video streaming in mobile devices.

This work proposes a stepwise adaptive video streaming architecture and control scheme for the wireless mobile networking environment based on MPEG Dynamic Adaptive Streaming over HTTP (MPEG-DASH) [6]. MPEG-DASH makes a single video have multiple representations in order to make the client have different choices to adapt with the network bandwidth fluctuation. MPEG-DASH avoids the traversal issues of the Network Address Translation (NAT) mechanism and firewall because it is over HTTP. Additionally, MPEG-DASH leaves the adaptive streaming control schemes void for customized design. For adapting with the wireless mobile network condition, our proposed method uses location, time, and date concerns for estimating the available bandwidth in the future. The principle is as follows: (i) the more proximate area from the current location of the mobile device is; (ii) the nearer time from the current time is; and (iii) the more recent day from the current day is, the higher weight the corresponding recorded measured bandwidth should have. Besides, our work designs a stepwise adaptive streaming control scheme that considers (i) buffer level; (ii) the video quality of the most recently downloaded segment; (iii) the downloading rate of the most recently downloaded segment and (iv) the estimated bandwidth that considers location, time, and date, to have better QoE, *i.e.*, avoiding abrupt quality increasing and decreasing. The proposed method is called Stepwise Adaptive Streaming using Temporal-Geo Bandwidth Estimation (SASTGBE) hereafter. The main technical issues that are tackled in the proposed SASTGBE are as follows: (1) how to build the temporal-geo bandwidth map? (2) how to calculate the estimated bandwidth? (3) how to derive which video quality the client should choose to download the next segment based on the aforementioned 4 factors?

The rest of this paper is organized as follows. Section 2 presents related works. Section 3 introduces the proposed architecture of the proposed SASTGBE. Section 4 describes the temporal-geo bandwidth estimation method in detail. Section 5 explains the proposed stepwise adaptive streaming control scheme in detail. Section 6 shows the experimental environment and performance analysis. Finally, this work is summarized and concluded in Section 7.

## 2. RELATED WORK

This Section presents related works for geo-adaptive streaming over HTTP.

With the popularity of mobile devices having GPS, different networking capabilities, and various sensors, the work depicted in [7] tried to build the bandwidth map using mobile devices, and some researches [8, 9] tried to consider spatial information, which was collected by mobile devices, for adaptive streaming to get better QoE. Murtaza *et al.* [7] proposed a method that uses opportunistic throughput measurements to create a personal bandwidth map. However, it only considered location; the measured bandwidth records related with time and date were ignored. Besides, it didn't propose an adaptive streaming control scheme because its goal was to create the personal bandwidth map.

The work depicted in [8] revealed that more recent information has higher correlation. The temporal factor can be spitted into two parts, the date part and the time part. Ignoring the date part may skip the wireless mobile network upgrade, *e.g.*, BSs upgrade from 3G to 4G, and then make inaccurate available bandwidth prediction because using the past bandwidth records in the old cellular network. The time part is noticeable because there is difference between rush hour and off-hour. Rush hour implies heavy traffic network, and off-hour implies smooth traffic network. Thus, even if it is on the same day and location, it should make different results of available bandwidth prediction in heavy traffic network and smooth traffic network. Yao *et al.* [9] adopted a bandwidth map that divides a long road into many segments and calculated an estimated bandwidth value for each segment. Nevertheless, the proposed method did not consider to have different weights for the past available bandwidth records of different distances, time and dates from the mobile device to get the smooth effect. Besides, they didn't have an exquisite adaptive streaming algorithm that considers downloading rate and remaining buffered time length.

Hao *et al.* [10] used the  $k$  nearest neighbors Inverse Distance Weighted ( $kNN$ -IDW) interpolation method to predict available bandwidth in the future, for which the 1-predict algorithm and the  $N$ -predict algorithm were proposed. The 1-predict algorithm is for the next video segment and the  $N$ -predict algorithm is for the next  $N$  video segments. The method gave different weighting factors for different distances between past available bandwidth records and the mobile device's current location; but it ignored the effect of different time's and dates' past bandwidth records.

Dubin *et al.* [11] shown their geo-predictive adaptive logic (GPAL) algorithm that can predict future bandwidth and choose a video quality to download. However, only some rough concepts for bandwidth prediction were shown. That is, they didn't reveal the formal scheme of the prediction method. Bayan Taani and Roger Zimmermann [8] analyzed the effect of location, time, and date for bandwidth estimation. They used the Kriging method, which is a method of interpolation in statistics to predict available bandwidth in the future. However, they didn't design a complete solution because they only proposed the method for bandwidth prediction but lacked an exquisite adaptive streaming algorithm.

## 3. THE PROPOSED ARCHITECTURE AND MAIN ISSUES

Fig. 1 shows the system components of the proposed architecture. Three main components are (1) mobile devices; (2) the temporal-geo bandwidth map and processing server,

and (3) the media server.

Three functional parts in a mobile device are (1) measured network situation's collector and uploader; (2) adaptive engine; and (3) media player. The measured network situation's collector and uploader record the contextual data of the mobile device, and then uploads the record each time when the mobile device has downloaded one video segment. The adaptive engine can query the smooth bandwidth of the past measured bandwidth from the temporal-geo bandwidth map and processing server and then derive the currently estimated bandwidth, and thereafter calculates the quality of the next downloaded video segment based on buffer level, video quality of the most recently downloaded segment, the downloading rate of the most recently downloaded segment and the currently estimated bandwidth. Then, the adaptive engine requests the media player to download the next segment of the calculated video quality. Then, the media player supporting the MPEG-DASH standard streams the corresponding video segment from the media server.

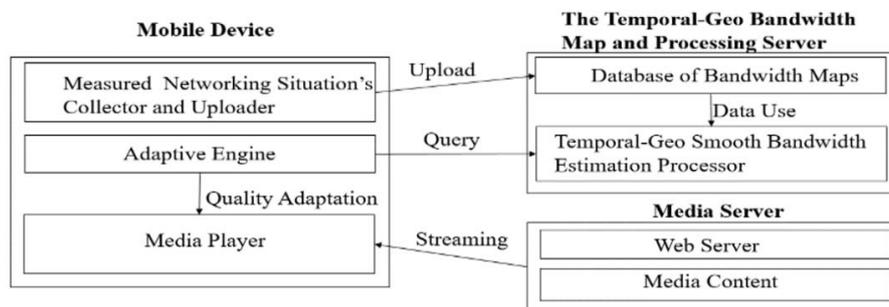


Fig. 1. System components of the proposed streaming architecture.

Two functional parts of the temporal-geo bandwidth map and processing server are (1) the database storing the bandwidth map and (2) the temporal-geo smooth bandwidth estimation processor. The database storing the bandwidth map is the storage of the measured bandwidth's records that were uploaded by mobile devices. Each record contains the measured bandwidth associated with the three axes of location (latitude, longitude), time, and date. The temporal-geo smooth bandwidth estimation processor is responsible for calculating the smooth bandwidth of the past measured bandwidth using the records stored in the database, *i.e.*, filtering them based on the client's current location, time and date and doing some mathematical operations, and then the server sends the smooth bandwidth of the past measured bandwidth to the corresponding mobile device to decide the follow-up streaming quality.

The media server is the storage of media content and is responsible for providing adaptive video streaming over HTTP. It is a web server that is compatible with MPEG-DASH and stores video files with multiple representations. The media server provides multiple representations for a single video in order to make mobile devices have different choices to adapt with the wireless mobile network bandwidth fluctuation while streaming. For example, one video file can have three representations, which are the resolution of 1280×720/1920×1080/2560×1440 with bitrate of 2Mbps/10Mbps/18Mbps, of a video file in its video track. The media server is a media content provider and can transfer the media presentation description file and media segments of a video to the mobile device.

Three main technique issues that are tackled in this work are as follows: (1) how to build the temporal-geo bandwidth map? (2) how to calculate the estimated bandwidth, and (3) how to derive which video quality the client should choose to download based on the estimated bandwidth and other contextual situation? In other words, they are the adaptive logic that needs to be devised for video quality adaptation.

The first issue is how to build the temporal-geo bandwidth map. That is, how to make records. When the mobile client completely downloads one segment, it can produce a record reporting the currently measured bandwidth, which is associated with its current location, time point and date, and then uploads it to the temporal-geo bandwidth map and processing server to build the temporal-geo bandwidth map.

The second issue is bandwidth estimation. The temporal-geo bandwidth map and processing server filters stored records based on the mobile client's current location, time and date, and then derives the smooth bandwidth of the past measured bandwidth for the corresponding mobile client. After the mobile client receives the smooth bandwidth of the past measured bandwidth calculated by the temporal-geo bandwidth map and processing server, it uses this data and the currently measured bandwidth to calculate the currently estimated bandwidth for the next segment. The basic concept for assigning different weights is as follows: (1) Location: the more proximate area it is, the higher weight it has; (2) Time: the much nearer time point it is, the higher weight it has; (3) Date: the more recent day it is, the higher weight it has.

The third issue is how to design the adaptive streaming control scheme for the wireless mobile network environment. For having better QoE, the design principle is as follows: (1) labelling the buffered presentation time length (not buffered data's volume) as multiple ranges, and use different streaming strategies in different ranges; (2) according to the current buffered presentation time length, select the corresponding streaming strategy and then choose the quality level for the next segment to be downloaded depending on the video quality of the most recently downloaded segment, the downloading rate of the most recently downloaded segment and the currently estimated bandwidth.

#### 4. TEMPORAL-GEO BANDWIDTH ESTIMATION

This Section presents the proposed temporal-geo bandwidth estimation method.

The mobile device calculates the currently estimated bandwidth based on the EWMA way using Eq. (1)

$$B_E^t = (1 - \alpha) * B_{measured}^t + \alpha * G_d^t \quad (1)$$

where  $\alpha$  is a parameter and  $0 \leq \alpha \leq 1$ ,  $B_E^t$  is the currently estimated bandwidth at time  $t$ ,  $B_{measured}^t$  is the currently measured bandwidth, which is measured by the mobile device and is derived from the size of the most recently downloaded segment dividing by the downloading time of the most recently downloaded segment, and  $G_d^t$  is the smooth bandwidth of the past measured bandwidth considering the past  $k_d$  days, the surrounding area in  $k_s$  meters, and the recent time of  $k_m$  minutes.  $G_d^t$  is calculated in the temporal-geo bandwidth map and processing server using the temporal-geo bandwidth map stored in the server based on the current location, time and date of the mobile device, and then is sent to the mobile device.

$$G'_d = (1 - \beta) * G'_{d-1} + \beta * X'_d \tag{2}$$

where  $0 \leq \beta \leq 1$  and  $X'_d$  is the historical records belonging to the same day. Expanding the above formula and substituting the start day, the following formula is derived:

$$\begin{aligned} G'_d &= (1 - \beta)[(1 - \beta)G'_{d-2} + \beta X'_{d-1}] + \beta X'_d = (1 - \beta)^2 G'_{d-2} + (1 - \beta)\beta X'_{d-1} + \beta X'_d = \\ &(1 - \beta)^3 G'_{d-3} + (1 - \beta)^2 \beta X'_{d-2} + (1 - \beta)\beta^2 X'_{d-1} + \beta X'_d = (1 - \beta)^k X'_{d-k} + (1 - \beta)^{k-1} \beta X'_{d-(k-1)} \tag{3} \\ &+ \dots + (1 - \beta)^2 \beta X'_{d-2} + (1 - \beta)\beta X'_{d-1} + \beta X'_d = (1 - \beta)^k X'_{d-k} + \sum_{i=d-k+1}^d (1 - \beta)^{d-i} \beta X'_i \end{aligned}$$

The aforementioned formula shows that the more past date it is, the smaller weight it has.

$X'_d$  is the estimated temporal-geo bandwidth based on the proximate area and the recent time period of a specific day. For example,  $X'_d$  denotes today's part,  $X'_{d-1}$  denotes yesterday's part,  $X'_{d-2}$  denotes the past second day's part,  $X'_{d-i}$  denotes the past  $i$ th day's part. Since it is impossible to consider an infinite number of past days, it needs to set a start date for  $G'_d$ . Let it consider the past  $k$  days. Then  $X'_{d-k}$  is the start date for  $G'_d$ , e.g., when  $k = 9$ , the start date is  $X'_{d-9}$ . Note that the past  $k$  days denote the same type of  $k$  days, i.e., the past  $k$  weekdays or the past  $k$  weekend days and holidays. Since the database may lack data in some days, e.g.,  $(d - 2)$  and  $(d - 4)$  are void. Then, just skip these data. For example, if it needs data for the past 9 days, it can use data in  $d, (d - 1), (d - 2), (d - 3), (d - 4), (d - 5), (d - 6), (d - 7), (d - 8),$  and  $(d - 9)$ ; but if the database doesn't have data in  $(d - 2)$  and  $(d - 4)$ , it uses data in  $d, (d - 1), (d - 3), (d - 5), (d - 6), (d - 7), (d - 8), (d - 9)$ .

$X'_d$  is the estimated temporal-geo bandwidth based on the proximate area and the recent time period of a specific day, for which the nearer/farther area it is, the higher/lower weight it has, and the more recent/past time period it is, the higher/lower weight it has. The procedure for calculating  $X'_d$  is as follows: (1) Select the data between  $t - m$  and  $t + m$  on a specific day  $d$ , where  $m$  is a parameter in minutes. For example, if  $m$  is 30 minutes, it considers the situations of the previous and next 30 minutes, then the time period is one hour; (2) The near/farther area is defined based on the distance level. The distance level can be  $D_1, D_2, D_3, \dots, D_p$ , where  $0 < D_1 < D_2 < D_3 < D_p$ . For example,  $D_1 = 10 m, D_2 = 20 m, D_3 = 30 m, \dots, D_p = 10 * pm$ . Fig. 2 shows an illustrated distance level for deriving  $X'_d$ . Then the surrounding area  $R_i$  is defined as  $D_i < R_i < D_{i+1}$ ; (3) There are several temporal-geo bandwidth records in area  $R_i, D_i < R_i < D_{i+1}$ ; (4) It can group the temporal-geo bandwidth records stored in the server based on the distance level, for which the temporal-geo bandwidth records that are in the same distance level have the same weight; (5) Deciding

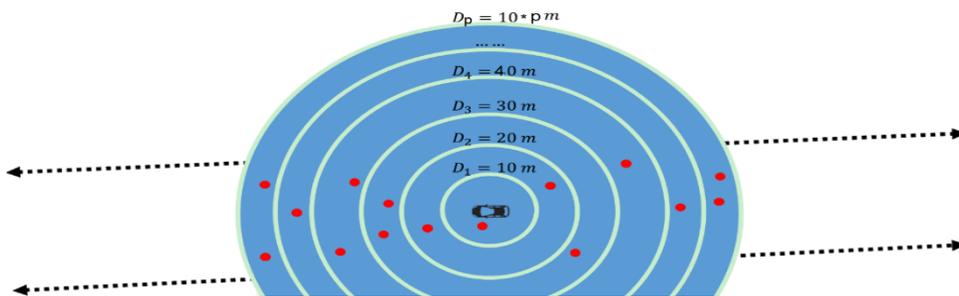


Fig. 2. The illustrated distance level for deriving  $X'_d$ .

which distance level the selected temporal-geo bandwidth record should belong to is based on its distance to the mobile device. For example, if a record is 17 meters from the mobile device, and  $D_1 = 10\text{m}$ ,  $D_2 = 20\text{m}$ . Since  $10 (D_1) < 17 < 20 (D_2)$ , the record should belong to  $D_2$ .

Let the weight value of subset  $D_i$  be  $Y_i$ ,  $i = 1, 2, \dots, p$ . Eq. (4) is derived as follows:

$$X_d^t = \sum_{i=1}^p \frac{D_{p-j+1}}{\sum_{j=1}^p D_j} Y_i^t = \frac{D_p}{\sum_{j=1}^p D_j} Y_1^t + \frac{D_{p-1}}{\sum_{j=1}^p D_j} Y_2^t + \dots + \frac{D_1}{\sum_{j=1}^p D_j} Y_p^t. \quad (4)$$

Here  $Y_i^t$  is the estimated temporal-geo bandwidth in the proximate area  $R_j$ , where  $D_j < R_j \leq D_{j+1}$ , based on  $t$ 's previous/next time periods of a specific day. Eq. (4) implies that the nearest set  $Y_1$  has the highest coefficient  $D_p/\sum_{j=1}^p D_j$ . Since a distance level may have no record, it should be set to zero, *i.e.*,  $D_k = 0$ , when there is no record for the proximate area  $R_{k-1}$ ; otherwise, the formula of  $X_d^t$  will result in a fatal error. Besides, it should use records of the same operator as the corresponding client and choose suitable distance levels.

Some notes for calculating  $Y_i^t$  are as follows: (1)  $Y_i^t$  has two subsets:  $Y_{i,before}^t$  and  $Y_{i,after}^t$ ; (2) Subset  $Y_{i,before}^t / Y_{i,after}^t$  contains the records on which their recorded time is prior/after the target time  $t$ ; (3)  $Y_i^t$  is derived based on the records that are from different time zones; (4) Each record belongs to a time zone based on its recorded time; (5) It can group the temporal-geo bandwidth records based on the time zone, for which the temporal-geo bandwidth records that are in the same time zone have the same weight. Fig. 3 demonstrates an example with the following condition: Let the time zone be 10 minutes, the time period of  $X_d^t$  be 1 hour, and the target time  $t$  be 09:00, then there are 6 time zones: 08:30-08:40, 08:40-08:50, 08:50-09:00, 09:00-09:10, 09:10-09:20, 09:20-09:30.  $Z_i$ ,  $i = 1, 2, 3$  and  $-1, -2, -3$ , in Fig. 3 is a time zone, and there are 12 records, which are recorded on time points 08:32, 08:38, 08:45, 08:47, 08:55, 08:59, 09:01, 09:05, 09:11, 09:17, 09:26, and 09:29, that belong to different time zones.

$Y_{i,before}^t$			0900	$Y_{i,after}^t$		
0830	0840	0850		0910	0920	0930
$Z_{-3}$	$Z_{-2}$	$Z_{-1}$		$Z_1$	$Z_2$	$Z_3$
0832	0845	0855	0901	0911	0926	
0838	0847	0859	0905	0917	0929	

Fig. 3. An example of grouping  $Y_i^t$ .

$Y_{i,before}^t$  is derived as follows,

$$Y_{i,before}^t = (1 - \gamma)Y_i^{t-m} + \gamma S^{(t-m,t]}, \quad (5)$$

$$Y_i^{t-m} = (1 - \gamma)Y_i^{t-2*m} + \gamma S^{(t-2*m,t-m]}, \quad (6)$$

$$Y_{i,before}^t = (1 - \gamma)^n Y_i^{t-m*n} + \sum_{u=1}^{n-1} (1 - \gamma)^{u-1} \gamma S^{(t-m*u,t-m*(u-1)]}. \quad (7)$$

In Eq. (7),  $n$  is the number of time zones of  $Y_{i,before}^t$  and  $m$  is the size of a time zone.  $S^{(t-m,t]}$  is the arithmetic mean of the temporal-geo bandwidth records in the time zone  $(t -$

$m, t]$  of the proximate area  $R_j$ , where  $D_j < R_j \leq D_{j+1}$  on a specific day. Since it is impossible to consider an infinite number of time zones, it needs to set a start time zone for  $Y_{i,before}^t$ . Let it consider  $n$  time zones. Then  $Y_i^{t-m*n}$  is the start time zone and its value is  $S^{(t-m*(n+1), t-m*n)}$ .  $Y_{i,after}^t$  is derived as follows,

$$Y_{i,after}^t = (1-\gamma)Y_i^{t+m} + \gamma S^{[t,t+m]}, \quad (8)$$

$$Y_i^{t+m} = (1-\gamma)Y_i^{t+2*m} + \gamma S^{[t+m,t+2*m]}, \quad (9)$$

$$Y_{i,after}^t = (1-\gamma)^n Y_i^{t+m*n} + \sum_{u=1}^{n-1} (1-\gamma)^{u-1} \gamma S^{[t+m*(u-1), t+m*u]}. \quad (10)$$

In Eq. (10),  $n$  is the number of time zones of  $Y_{i,after}^t$  and  $m$  is the size of time zone;  $S^{[t, t+m]}$  is the arithmetic mean of the temporal-geo bandwidth records in the time zone  $[t, t+m]$  of the proximate area  $R_j$ , where  $D_j < R_j \leq D_{j+1}$ , on a specific day. Since it is impossible to consider an infinite number of time zones, it needs to set a start time zone for  $Y_{i,after}^t$ . Let it consider  $n$  time zones. Then  $Y_i^{t+m*n}$  is the start time zone and its value is  $S^{[t+m*n, t+m*(n+1)]}$ . Finally,  $Y_i^t$  is derived as follows:

$$Y_i^t = 0.5Y_{i,before}^t + 0.5Y_{i,after}^t. \quad (11)$$

However,  $Y_i^t = Y_{i,before}^t$  for the current day because the temporal-geo bandwidth records for the future time of today are unknown. Additionally, there may be some time zones which lack records, for which condition it needs to skip them; otherwise, it results in a fatal error.

## 5. THE STEPWISE ADAPTIVE STREAMING ALGORITHM

This Section presents the proposed stepwise adaptive streaming algorithm in details.

The proposed stepwise adaptive streaming algorithm relies on four factors: (i) the buffered presentation time length, which refers to the presentation time length of the buffered data; (ii) the video quality of the most recently downloaded segment; (iii) the downloading rate of the most recently downloaded segment; and (iv) the currently estimated bandwidth.

The playout buffer is an important component for video streaming because it can absorb the disparity of network delay, *i.e.*, jitter, and the disparity between the available bandwidth and the video quality. Two amounts implied by playout buffer are (i) buffered data volume and (ii) buffered data's presentation time length. Playout pause harms QoE as much as low video quality does while playing a video stream. Thus, instead of using the buffered data volume, the buffered data's presentation time length is observed and used to control adaptive streaming when monitoring the playout buffer for having smooth QoE.

Three concerns for maintaining the suitable buffered data's presentation time length are as follows: (1) the overfull buffered data's presentation time length is a waste of resource because if the user switches to the other video, then the overfull buffer of this video will be dropped; (2) The short buffered data's presentation time length will be consumed quickly in the heavy fluctuation situation of available bandwidth, which may result in playout pause; (3) To avoid dropping video quality of experience too sharply, the video quality should be changed gradually regardless of increasing or decreasing quality. Four thresh-

holds in the buffered data's presentation time length (in seconds) are set in the proposed stepwise adaptive streaming algorithm:  $\tau_{vlow}$  is the very low buffer threshold,  $\tau_{low}$  is the low buffer threshold,  $\tau_{high}$  is the high buffer threshold, and  $\tau_{vhigh}$  is the very high buffer threshold. As a result, the buffered data's presentation time length is divided into five ranges based on these four thresholds, and there are different adaptive strategies for different ranges. Fig. 4 shows the five ranges and the initial buffered time length. The goal of this stepwise adaptive streaming algorithm is to keep the playout buffer data's presentation time length stay in range  $R$ .

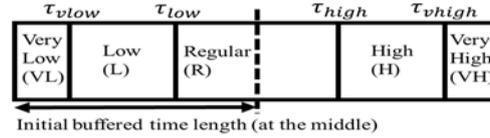


Fig. 4. Five ranges of the buffered data's presentation time length.

Table 1 shows the symbols that are used in the proposed stepwise adaptive streaming algorithm. Fig. 5 depicts the stepwise adaptive streaming algorithm. The stepwise adaptive streaming algorithm has different adaptive strategies for different ranges. When the buffered data's presentation time length is in the range of Very Low (VL), *i.e.*,  $\tau < \tau_{vlow}$ , the playout buffer is to be exhausted. Thus, downloading the next segment with the lowest quality can fill the playout buffer quickly. If it is in the range of Very High (VH), *i.e.*,  $\tau < \tau_{vhigh}$ , the playout buffer is to be full. Thus, suspend downloading until the buffered data's presentation time length goes down to threshold  $\tau_{high}$ .

**Table 1. Symbols used in the stepwise adaptive streaming algorithm.**

$\tau$	The current buffered data's presentation time length
$B_E$	The estimated bandwidth
$r_{last}$	The downloading rate for the most recently downloaded segment
$q$	The proposed quality level
$q_{last}$	The quality level of the most recently downloaded segment
$q_l$	The lowest quality level
$q_h$	The highest quality level
$q_{est}$	The maximum quality level for which the needed bandwidth is less than or equal to the estimated bandwidth
$p_1, p_2, p_3, p_4, p_5, p_6, p_7$	The parameters for quality level control

---

#### Stepwise adaptive streaming algorithm

---

- 1: **switch** condition **do**
  - 2:   **case**  $\tau < \tau_{vlow}$
  - 3:      $q \leftarrow q_l$
  - 4:   **case**  $\tau > \tau_{vhigh}$
  - 5:     Suspend downloading until the current buffered data's presentation
  - 6:     time length goes back to threshold  $\tau_{vhigh}$
  - 7:   **case**  $\tau_{vlow} \leq \tau < \tau_{low}$
  - 8:    **if**  $B_E > r_{last}$  **then**
-

---

```

9:      // Low buffer but the networking situation is becoming better
10:      $q \leftarrow q_{last}$ 
11:     else
12:       If  $q_{est} > q_{last}$  then
13:          $q \leftarrow q_{last}$ 
14:       else if  $q_{est} = q_{last}$  then
15:          $q \leftarrow \max(q_{last} - p_1, q_i)$ 
16:       else
17:          $q \leftarrow \max(q_{last} - p_2, q_i)$ 
18:       end if
19:     end if
20:   case  $\tau_{high} < \tau \leq \tau_{vhigh}$ 
21:     if  $B_E \geq r_{last}$  then
22:       // High buffer and the networking situation is still becoming better
23:        $q \leftarrow \min(q_{last} + p_3, q_h)$ 
24:     else
25:       if  $q_{est} > q_{last}$  then
26:          $q \leftarrow \min(q_{last} + p_4, q_h)$ 
27:       else if  $q_{est} = q_{last}$  then
28:          $q \leftarrow \min(q_{last} + p_5, q_h)$ 
29:       else
30:          $q \leftarrow q_{last}$ 
31:       end if
32:     end if
33:   case default ( $\tau_{low} \leq \tau \leq \tau_{high}$ )
34:     if  $q_{est} > q_{last} + p_6$  then
35:        $q \leftarrow q_{last} + p_6$ 
36:     else if  $q_{est} < q_{last} - p_7$  then
37:        $q \leftarrow q_{last} - p_7$ 
38:     else
39:        $q \leftarrow q_{last}$ 
40:     end if

```

---

Fig. 5. The stepwise adaptive streaming algorithm.

When the buffered data's presentation time length is in the range of Low (L), *i.e.*,  $\tau_{vlow} \leq \tau < \tau_{vlow}$ , there are two conditions: (1) If  $B_E > r_{last}$ , it means that the network situation is improving. Since the buffering situation is still low, it can keep the same quality of the most recently downloaded segment for downloading the next one such that the buffer can be filled quickly; (2) If  $B_E \leq r_{last}$ , it means that the network situation is becoming bad; (i) If  $q_{est} > q_{last}$ , it sets the quality level of the next downloading segment as  $q_{last}$  because the estimated bandwidth is still higher than the bitrate of the lastly downloaded segment and thus keeping downloading the  $q_{last}$  level can fill the playout buffer; (ii) If  $q_{est} = q_{last}$ , the quality level for the next downloading segment is set to  $q_{last} - p_1$ , where  $p_1$  is a parameter, in order to fill the playout buffer quickly; (iii) If  $q_{est} < q_{last}$ , it sets the quality level of the next downloading segment as  $q_{last} - p_2$ , where  $p_2$  is a parameter, because sharply changing the video quality will drop the QoE too much. However, if changing to  $q_{last} - p_1$  or  $q_{last} -$

$p_2$  is lower than the lowest quality level, then just changing to the lowest quality level  $q_l$ . Statements  $q \leftarrow \max(q_{last} - p_1, q_l)$  on line 15 and  $q \leftarrow \max(q_{last} - p_2, q_l)$  on line 17 of Fig. 5 denote the aforementioned principle.

When the buffered data's presentation time length is in the range of Regular ( $R$ ), *i.e.*,  $\tau_{low} \leq \tau \leq \tau_{high}$ , it means that the playout buffer is in the target range. Thus, the adaptation relies on the quality level of the most recently downloaded segment and the maximum quality level for which the needed bandwidth is less than or equal to the estimated bandwidth to decide the quality level of the next downloading segment. Instant altering the video quality too much would change the QoE sharply. To avoid this situation, video quality changing should be limited with an upper bound and a lower bound: (1) if  $(q_{last} - p_7 \leq q_{est} \leq q_{last} + p_6)$ , then  $q \leftarrow q_{est}$ , where  $p_6$  and  $p_7$  are parameters; (2) if  $(q_{est} > q_{last} + p_6)$ , then  $q \leftarrow q_{last} + p_6$ ; (3) if  $(q_{est} < q_{last} - p_7)$ , then  $q \leftarrow q_{last} - p_7$ .

When the buffered data's presentation time length is in the range of High ( $H$ ), *i.e.*,  $\tau_{high} < \tau \leq \tau_{high}$ , there are two conditions: (1) If  $B_E \geq r_{last}$ , it means that the network situation is becoming better. Since the buffering situation is better than the regular one, it sets the quality level of the next downloading segment as  $q_{last} + p_3$ , where  $p_3$  is a parameter, for adjusting the playout buffer to range  $R$  gradually. If increasing the quality level by  $p_3$  exceeds the highest quality level, then just download the highest one; (2) If  $B_E < r_{last}$ , it means that the network situation is becoming bad. Since the goal is to make the buffered presentation time length go back to range  $R$  gradually; (i) if  $q_{est} > q_{last}$ , it sets the quality level of the next downloading segment as  $q_{last} + p_4$ , where  $p_4$  is a parameter; (ii) if  $q_{est} = q_{last}$ , it sets the quality level of the next downloading segment as  $q_{last} + p_5$ , where  $p_5$  is a parameter, to consume the playout buffer's data quickly; (iii) if  $q_{est} < q_{last}$ , it sets the quality level of the next downloading segment as  $q_{last}$  to go back to range  $R$ . If changing to  $q_{last} + p_4$  or  $q_{last} + p_5$  is higher than the highest quality level, then just changing to the highest quality level  $q_h$ . Statements (1)  $q \leftarrow \min(q_{last} + p_4, q_h)$  and (2)  $q \leftarrow \max(q_{last} + p_5, q_h)$  in Fig. 5 denote the above principle.

## 6. PERFORMANCE ANALYSIS

In order to evaluate the performance, the client software using the proposed SASTGBE method was implemented in Android 6.0.1. In the server side, an Ubuntu server that installs the MySQL database service and the Apache HTTP service with the customized scripts and APIs is the temporal-geo bandwidth map and processing server; the Ubuntu server is also the media server to simplify the experimental environment. The proposed method is compared with two other methods, for which one is called the kNN IDW with the 1-predict algorithm that was proposed in [10], and the other one is the traditional adaptive algorithm. The traditional adaptive algorithm uses past measured bandwidth records of its own session to predict the available bandwidth in the future, and then downloads the segment whose video quality is smaller than or equal to, *i.e.*, closet to, the predicted bandwidth. The formula of bandwidth prediction using the traditional adaptive algorithm is as follows,

$$B_r^t = \kappa * B_{instant}^t + (1 - \kappa) * B_r^{t-1} \quad (12)$$

where  $0 \leq \kappa \leq 1$ ,  $B_r^t$  is the estimated bandwidth at time  $t$ , and  $B_{instant}^t$  is the measured bandwidth of the mobile device at time instant  $t$ .

## 6.1 Datasets

Fig. 6 depicts the experimental path, whose total length is about 5.8 kilometers, used for the performance analysis. The path starts at the urban area, then passes through the main train station's area, and finally stops at the suburban district of Tainan city. Since the path includes different environments, mobile devices connect to different BSs and suffer fluctuation in available bandwidth. The experiment started from late June to early September 2017, and the tested time period was from 6 pm to 8 pm because it is from rush hour to off-hour. Testers rode motorcycles along the experimental path at the speed of 40 km/h, which is a reasonable speed in the rush hour.



Fig. 6. The experimental path used for performance analysis.

Sintel [12], which is an open content of an animated movie and has much higher video quality source, is used for testing. The Sintel content is encoded to 40 quality levels, in which the lowest level is 1031191 bits/s and the highest level is 40936235 bits/s; the difference of 2 neighboring levels is about 1 Mbits/s; the fixed segment length is 2 seconds; the playout rate is 24 frames per second; video frames are encoded in H.264/AVC.

## 6.2 Evaluation Metrics

To compare the performance of the proposed Stepwise Adaptive Streaming using Temporal-Geo Bandwidth Estimation (SASTGBE) method with the other two methods, eight evaluation metrics are as follows:

- (1) Bandwidth usage percentage: The value integrating the video bitrate function  $b(t)$  over time  $t$  divides the value integrating the available bandwidth function  $B(t)$  over time  $t$ . The following formula is its mathematic form:

$$\int (\text{video bitrate} * dt) / \int (\text{available bandwidth} * dt)$$

- (2) Suspended time percentage: The suspended time resulted from exhausted buffer and rebuffering divides the session time length, excluding the start-up delay time.
- (3) Pause time percentage: The sum of (i) the start-up delay time and (ii) the suspended time divides the session time length.
- (4) Suspended times per 20 minutes: The time for each trip was not fixed because it would depend on the traffic condition. Therefore, suspended times need to be normalized.

Since each session is between 21 and 25 minutes, 20 minutes is chosen as the base.

- (5) Quality switch percentage: The quality switch times divides the number of segments minus 1. Higher values represent more frequent switching, which results in poor QoE.
- (6) Average bitrate difference considering suspending: the root square of the sum of square of bitrate difference of neighboring segments dividing the total quality switch times. This metric regards the playback suspending as a zero bitrate case in order to make an adaptive algorithm pay the bill for suspending. The square gives higher bitrate difference higher penalty. The following formula is its mathematic form:

$$\sqrt{\frac{\sum_{i=1}^k (\text{ith bitrate difference})^2}{\text{quality switch times}}}$$

Higher values indicate that the quality switching is across a large range of bitrate and results in poor QoE.

- (7) Standard deviation of Bitrate Difference: the standard deviation of bitrate difference of neighboring segments. Higher values indicate that bitrate difference values are discrete from each other. That is, sometimes there is a lower quality difference, but sometimes there is a higher quality difference. The aforementioned phenomenon disturbs users and harms QoE, so the lower value is better.

$$\sqrt{\frac{\sum_{i=1}^k (\text{ith bitrate difference} - \text{average bitrate difference})^2}{\text{quality switch times}}}$$

where the average bitrate difference means the arithmetic mean of the absolute value of bitrate differences. The following formula is the mathematic form:

$$\frac{\sum_{i=1}^k |\text{ith bitrate difference}|}{\text{quality switch times}}$$

- (8) Confidence level: the proportion of possible intervals that contain the average of population. For example, if the average of bandwidth usage percentage is 74.07 and the 99 percent confidence level of bandwidth usage percentage is 1.61, then it means that if the measurement is repeated again and again, then 99% of the average of bandwidth usage percentage will be in the range from (74.07 – 1.61) to (74.07 + 1.61). Additionally,  $\alpha$ ,  $B$ ,  $\tau$  and  $k$  in Eqs. (1), (3), (4) and (12) is set as 0.25.

### 6.3 Experimental Results

Table 2 and Figs. 7 to 9 are for comparing SASTGBE and the traditional adaptive algorithm. Table 3 and Figs. 10 to 11 are for comparing SASTGBE and kNN IDW with 1-predict.

Table 2 shows the measured data for a streaming session, which indicates that SASTGBE outperforms the traditional adaptive algorithm except bandwidth usage percentage. The traditional adaptive algorithm is very poor in suspended time and suspended times (86.459 seconds and 16 times), which results in poor QoE. Fig. 7 depicts the estimated bandwidth and measured bandwidth in the streaming session. Fig. 7 shows that the wireless

**Table 2. SASTGBE versus the traditional adaptive algorithm.**

Metric	SASTGBE	Traditional Adaptive Algorithm
Bandwidth Usage %	74.93	87.29
Average Measured Bandwidth (Mbps)	25.84	23.94
Average Presentation Bitrate (Mbps)	21.23	22.87
Pause Time %	1.32	7.35
Initial delay Time (s)	19.067	19.823
Suspended Time (s)	0	86.459
Suspended Times	0	16
Quality Switch %	57.7	72.2
Quality Switch Times	412	483
Bitrate Difference Per Segment (Mbps)	1.94	2.81
Average Bitrate Difference Considering Suspending (Mbps)	5.63	5.89
Standard Deviation of Bitrate Difference (Mbps)	5.23	5.63

mobile network environment has high measured bandwidth but suffered heavy fluctuation. Figs. 7 (a) and (b) indicate that using two mobile devices of the same specification in the same location and same time would have the same trend in measured bandwidth but could get different measured bandwidth values because the BS may make different decisions for different mobile devices. That is, though the operator has Quality of Service (QoS) control, which results in the measured bandwidth having the same general trend, there are still some little difference. The average measured bandwidth value in Table 2 also tells the fact. The black boxes in Figs. 7 (a) and (b) show that SASTGBE has smoother suggestion of estimated bandwidth than the traditional adaptive algorithm.

Fig. 8 shows the measured bandwidth and the corresponding downloaded bitrate. Fig. 8 (a) reveals that SASTGBE's downloaded segments depend on not only measured bandwidth but also the remaining buffer presentation length. Points A, B, C, D, and E of Fig. 8 (a) show that there were heavy fluctuation occurrences, which make SASTGBE be in the very low buffer level. Thus, SASTGBE downloaded the lowest quality level. Fig. 8 (b) shows that the traditional adaptive algorithm's downloaded segments depend only on the estimated bandwidth and thus it usually runs into exhausted buffer, which results in frequent playback suspending (86.459 seconds and 16 times in Table 2). Besides, Fig. 8 (b) shows that the traditional adaptive algorithm has better available bandwidth than SASTGBE in points A, B, C, D, and E. The traditional adaptive algorithm does not mind playback suspending and keep the downloaded bitrate as the same as its estimated bitrate. The traditional adaptive algorithm is a gambler, and it was lucky in the 300<sup>th</sup>-500<sup>th</sup> seconds because it still has buffered data to play in this example.

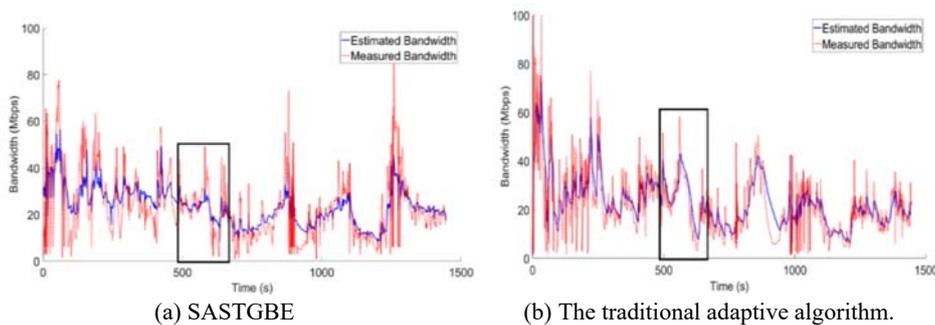


Fig. 7. Estimated bandwidth and measured bandwidth of (a) SASTGBE and (b) the traditional adaptive algorithm.

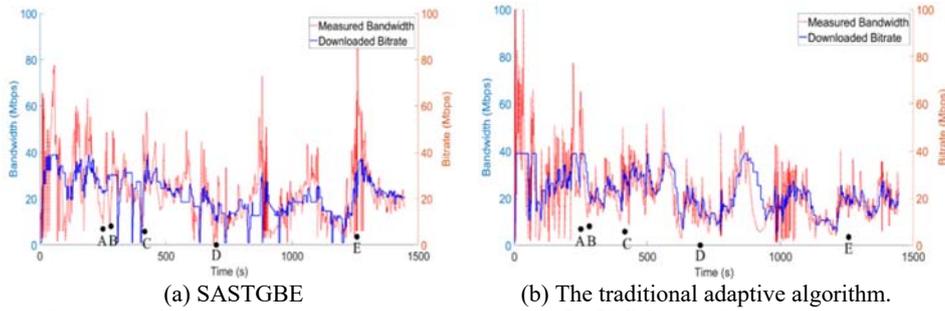


Fig. 8. Measured bandwidth and downloaded bitrate of (a) SASTGBE and (b) of the traditional adaptive algorithm.

Table 3 depicts the measured data for the streaming session of SASTGBE versus kNN IDW with 1-predict. It reveals that SASTGBE outperforms kNN IDW with 1-predict in bandwidth usage percentage, pause time percentage, quality switch times, quality switch percentage, average bitrate difference considering suspending, and standard deviation of bitrate difference.

**Table 3. SASTGBE versus kNN IDW with 1-predict.**

Metric	SASTGBE	kNN IDW with 1-predict
Bandwidth Usage %	72.51	64.74
Average Measured Bandwidth (Mbps)	23.29	23.90
Average Presentation Bitrate (Mbps)	18.41	17.13
Pause Time %	2.06	2.60
Initial delay Time (s)	23.056	31.187
Suspended Time (s)	3.536	2.225
Suspended Times	2	1
Quality Switch %	49.92	78.91
Quality Switch Times	315	494
Average Bitrate Difference Considering Suspending (Mbps)	6.12	6.90
Standard Deviation of Bitrate Difference (Mbps)	5.75	6.59

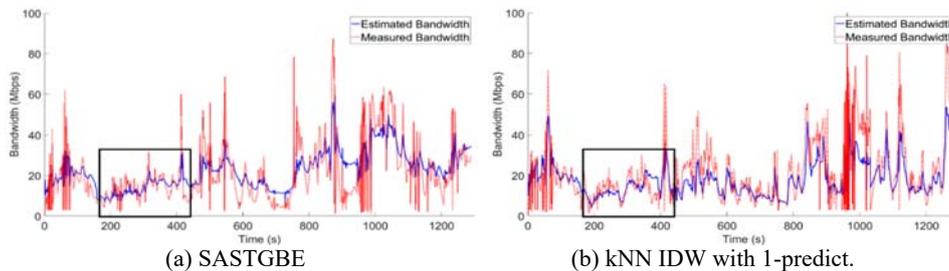


Fig. 9. Estimated bandwidth and measured bandwidth of (a) SASTGBE and (b) of kNN IDW with 1-predict.

Figs. 9 (a) and (b) depict the estimated bandwidth and measured bandwidth in the streaming session and reveals that, even if the operator has Quality of Service (QoS) control, which results in the measured bandwidth having the same general trend, there are still some little difference. The average measured bandwidth value depicted in Table 3 also tells the fact. The black boxes in Figs. 9 (a) and (b) show that SASTGBE has smoother suggestion of estimated bandwidth than kNN IDW with 1-predict. Fig. 10 depicts the mea-

sured bandwidth and the corresponding downloaded bitrate and show that SASTGBE and kNN IDW with 1-predict avoid exhausted buffer, but kNN IDW with 1-predict has frequent quality switch and high downloaded bitrate difference in the 210<sup>th</sup>–230<sup>th</sup> seconds and the 850<sup>th</sup>–920<sup>th</sup> seconds. Points F and G in Fig. 10 (a) show that there were heavy fluctuation occurrences, which makes SASTGBE in the very low buffer level and thus SASTGBE downloaded the lowest quality level. However, there is no such network fluctuation in the same time of Fig. 10 (b).

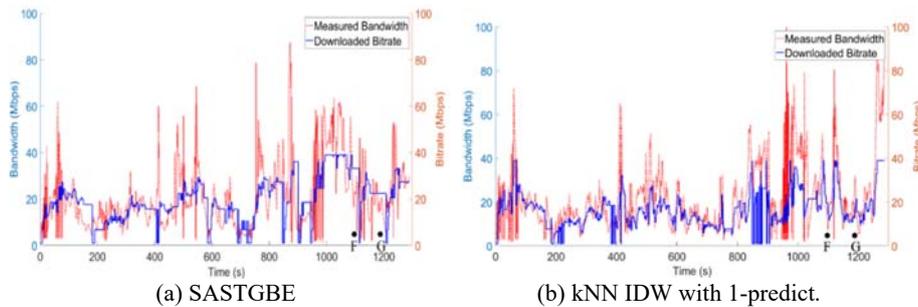


Fig. 10. Measured bandwidth and downloaded bitrate of (a) SASTGBE and (b) of kNN IDW with 1-predict.

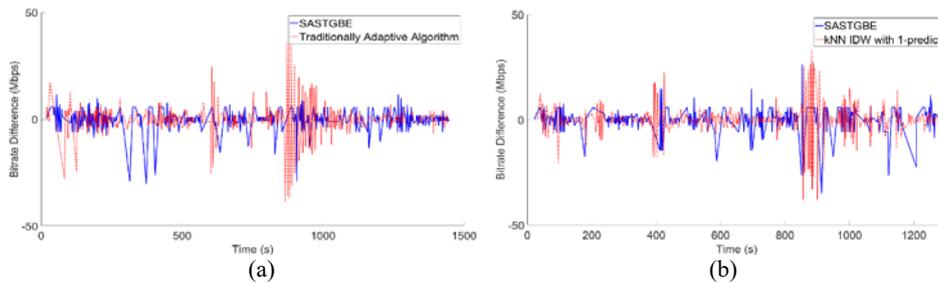


Fig. 11. Presentation bitrate difference considering playback suspending; (a) SASTGBE vs. the traditional method; (b) SASTGBE vs. the kNN IDW with 1-predict method.

Fig. 11 shows presentation bitrate difference considering playback suspending. In Fig. 11 (a), the presentation bitrate difference of the traditional adaptive algorithm vibrates heavily in the 600<sup>th</sup>–640<sup>th</sup>, the 850<sup>th</sup>–1000<sup>th</sup>, and the 1030<sup>th</sup>–1100<sup>th</sup> seconds while the presentation bitrate difference of SASTGBE doesn't. The reason is that SASTGBE sets the upper bound and the lower bound for quality increasement and decreasement. Fig. 11 (a) depicts that SASTGBE has sharp quality decrease for 3 times in the 300<sup>th</sup>–500<sup>th</sup> seconds because SASTGBE suffered heavy network fluctuation but the traditional adaptive algorithm didn't and thus SASTGBE changed its downloaded bitrate to the lowest level to avoid playback suspending.

Fig. 11 (b) reveals that kNN IDW with 1-predict has frequent quality switch and high presentation bitrate difference in the 210<sup>th</sup>–230<sup>th</sup> seconds and the 850<sup>th</sup>–920<sup>th</sup> seconds. However, SASTGBE avoids the aforementioned phenomenon because it sets an upper bound and a lower bound for both quality increasement and decreasement, which plays an im-

portant role in the high measured bandwidth but high network fluctuation wireless mobile environment. Fig. 11 (b) also shows that SASTGBE has sharp quality decrease for 2 times in the 1100<sup>th</sup>–1250<sup>th</sup> seconds because SASTGBE suffered heavy network fluctuation but kNN IDW with 1-predict didn't and thus SASTGBE changed its downloaded bitrate to the lowest level to avoid playback suspending.

## 7. CONCLUSION AND FUTURE WORK

In this work, the network condition in the 4G wireless mobile network has been investigated, in which it reveals that the available bandwidth is high but suffers heavy network fluctuation. Besides, it has shown that the available bandwidth in different time periods of the same type of days, *i.e.*, week days or weekends, is also varied a lot from the observation. The method called SASTGBE has been proposed in this work to tackle the aforementioned conditions. SASTGBE considers location, time, and date for estimating the available bandwidth in the future; the stepwise adaptive streaming control scheme of SASTGBE considers buffer level, video quality of the most recently downloaded segment, the downloading rate of the most recently downloaded segment and the estimated bandwidth to decide the video quality for the next downloaded segment. To avoid altering the video quality too much, which harms QoE, SASTGBE limits quality change with an upper bound and a lower bound. In the performance analysis, the experiments using SASTGBE have been tested many times in the 4G wireless mobile networks. The experimental results have shown that SASTGBE considering both temporal and spatial factors and adapting with the contextual situation can improve the performance of bandwidth utilization, suspended times, quality switch percentage, average bitrate difference considering suspending, and standard deviation of bitrate difference over the 4G wireless mobile network and thus get better QoE in the high network fluctuation environment. The main future work is to intergate the Mobile Edge Computing (MEC) technique, for which some cloud servers are allocated in the edge of wireless mobile network, to offload the computing from the remote site to the local site to have the more real-time response.

## REFERENCES

1. M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hobfeld, and P. Tran-Gia, "A survey on quality of experience of HTTP adaptive streaming," *IEEE Communications Surveys & Tutorials*, Vol. 17, 2015, pp. 469-492.
2. T. Zhao, Q. Liu, and C. W. Chen, "QoE in video transmission: A user experience-driven strategy," *IEEE Communications Surveys & Tutorials*, Vol. 19, 2017, pp. 285-302.
3. K. Miller, E. Quacchio, G. Gennari, and A. Wolisz, "Adaptation algorithm for adaptive streaming over HTTP," in *Proceedings of the 19th International Packet Video Workshop*, 2012, pp. 173-178.
4. G. Tian and Y. Liu, "Towards agile and smooth video adaptation in dynamic HTTP streaming," in *Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies*, 2012, pp. 109-120.
5. H. Riiser, T. Endestad, P. Vigmostad, C. Griwodz, and P. Halvorsen, "Video streaming using a location-based bandwidth-lookup service for bitrate planning," *ACM Tran-*

- sactions on Multimedia Computing, Communications, and Applications*, Vol. 8, 2012, pp. 1-19.
6. ISO/IEC 23009-1, "Dynamic adaptive streaming over HTTP(DASH)-Part1: Media presentation description and segment formats," 2014.
  7. G. Murtaza, A. Reinhardt, M. Hassan, and S. S. Kanhere, "Creating personal bandwidth maps using opportunistic throughput measurements," in *Proceedings of the 49th IEEE International Conference on Communications*, 2014, pp. 2454-2459.
  8. B. Taani and R. Zimmermann, "Spatio-temporal analysis of bandwidth maps for geo-predictive video streaming in mobile environments," in *Proceedings of the 24th ACM on Multimedia Conference*, 2016, pp. 888-897.
  9. J. Yao, S. S. Kanhere, and M. Hassan, "Improving QoS in high-speed mobility using bandwidth maps," *IEEE Transactions on Mobile Computing*, Vol. 11, 2012, pp. 603-617.
  10. 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.
  11. R. Dubin, A. Dvir, O. Pele, O. Hadar, I. Katz, and O. Mashiach, "Adaptation logic for HTTP dynamic adaptive streaming using geo-predictive crowdsourcing for mobile users," *Multimedia Systems*, 2016, pp. 1-13.
  12. <https://durian.blender.org/>.



**Chung-Ming Huang** received the B.S. degree in Electrical Engineering from National Taiwan University on 1984/6, and the M.S. and Ph.D. degrees in Computer and Information Science from The Ohio State University in 1988 and 1991 respectively. He is currently a Distinguished Professor in Department of Computer Science and Information Engineering (CSIE), National Cheng Kung University (NCKU), Tainan, Taiwan. He was (i) Chair of Department of CSIE and Director of Institute of Medical Informatics, NCKU and (ii) served as the Chair of IEEE Vehicular Technology Society Tainan Chapter. His research interests include wireless and mobile network protocol design and analysis, green computing and communication, media processing and streaming, and innovative network applications and services. He is a senior member of IEEE and ACM.



**Rui-Xian Wei** received the B.Sc. degree from Department of Engineering Science and Ocean Engineering, National Taiwan University in June 2012. He got the M.S. degree in Computer Science and Information Engineering, National Cheng Kung University, Taiwan, in January 2018. His research interests include MPEG-DASH, video streaming, location-based service (LBS).