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

CHUNG-MING HUANG*[1], RUI-XIAN WEI[1]
[1]*Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan*
E-mail: huangcm@locust.csie.ncku.edu.tw;

With the advance of wireless mobile communication technologies, video streaming has advanced much more on these years. However, the wireless mobile network usually suffers fluctuation in available bandwidth because mobile users usually keep moving and playing streaming video simultaneously. To overcome the situation, this work proposed a method called Stepwise Adaptive Streaming using Temporal-Geo Bandwidth Estimation (SASTGBE) 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 SASTGBE in the real environment shown that SASTGBE 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 recently [1]. However, unlike wired network, the wireless mobile networking environment usually suffers fluctuation in available bandwidth. Two video streaming techniques over HTTP are (i) progressive downloading and (ii) adaptive video streaming. Progressive downloading means that the video quality remains the same and can not be adjusted while playing a video. However, it has a significant defect because the bandwidth in the networking environment is varied, unpredictable and uncontrollable. Adaptive video streaming tackles the aforementioned problems of progressive downloading because it can adjust the video quality depending on the currently available bandwidth. Thus, adaptive video streaming over HTTP becomes a more feasible solution to adapt the video quality with network fluctuation. The reason is

This work was supported by the Ministry Of Science and Technology (MOST), Taiwan (R.O.C.) under the grant number MOST 106-2221-E-006-029.

that the adaptive video streaming technique can adjust video quality based on the networking condition. This work concentrates on the adaptive video streaming technique for HTTP video streaming.

To have good adaptive streaming, the main concern is how to predict the network condition to adjust video quality. The rough idea is just based on the past and the currently measured bandwidth of the current streaming session. 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 BW_E^t is equal to $(1-\alpha)BW_E^{t-1}+\alpha BW_{cm}^t$, for which BW_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, no or fewer suspended times, no or shorter paused time, 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. More factors, e.g., playback pause resulted from exhausted buffer, that may affect QoE should also be considered. Thus, adaptive video streaming needs a reasonable adaptive streaming control 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][5]. 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. For example, a mobile device in a high speed moving vehicle usually suffers heavy fluctuation in available bandwidth because of frequent BS switching, which consequently results in the poor QoE.

The work depicted in [6] considered both temporal and spatial factors to resolve the video streaming problems that exist in the wireless mobile network environment, for which mobile devices keep moving and playing streaming video simultaneously. The temporal factor means the time difference between the recorded time of the past measured bandwidth and the time of measuring the current bandwidth; the spatial factor means the distance between the location of recording the past measured bandwidth and the location of measuring the current bandwidth. 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 should have 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 should have for the estimated bandwidth of the current time t. That is, the past measured bandwidth associated with location can also be considered to predict the estimated bandwidth in the future such that the QoE of having

video streaming in mobile devices can be improved. In other words, a mobile device needs to collect some environmental and contextual data, which are used by itself or delivered to a temporal-geo bandwidth map and processing server. The temporal-geo bandwidth map and processing server store records of measured bandwidth associated with the corresponding recorded time and location for calculating the estimated bandwidth of current time and then derive the suitable streaming quality.

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) [11]. 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 increase and decrease. The proposed method can thus have step-wise video quality change and can avoid playback pause resulted from exhausted buffer. 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 on (1) video streaming over HTTP and (2) geo-adaptive streaming over HTTP.

2.1 Video Streaming over HTTP

The work depicted in [12] proposed a new quality adaptation algorithm called the Wireless Quality Adaptation (WQUAD) for the wireless mobile environment. WQUAD

leverages the SVC-DASH framework and gives priority to lower layers over higher ones as long as the playout buffer is not filled by the lower layers. In other words, it lets clients fill up the buffer with the base layer first and then the enhancement layers sequentially. However, it was not compatible with the AVC-DASH framework. The work depicted in [13] revealed that live video streaming over HTTP often exhibits latencies of up to 20 seconds. Thus, it proposed an adaptive algorithm called LOLYPOP (short for low-latency prediction-based adaptation) for HTTP-based live streaming. LOLYPOP was designed to operate with a transport latency of a few seconds over the wireless network environment. The work depicted in [14] proposed a system architecture that uses past available bandwidth to predict available bandwidth in the future, for which a predictor is used to make decisions based on the predicted buffer level. The aforementioned work claimed that the proposed mechanism can dynamically control the rate of the requested video and have intelligent decision-making for downloading video segments. The work depicted in [15] tried to reduce the network load in situations for which the non-optimal viewing condition, i.e., the user distracts himself to do other things, exists or the user does not focus on watching the video. In other words, the work wanted to reduce the network load without harming the QoE. The work depicted in [16][17] proposed a machine learning adaptive streaming algorithm. It claimed that the proposed method can reduce the bandwidth consumption and machine learning can help the client media engine to learn about the network condition and let the adaptive streaming algorithm make future decisions on adaptation without consulting the actual adaptive algorithm.

The aforementioned work [12] tried to get better QoE depending on the intra-session information, i.e., the past and current downloading rates and remaining buffered time length of its own session. They are not suitable for adaptive HTTP streaming in the wireless mobile environment, in which the mobile device is moving and thus the metrics of considering location and the corresponding location's temporal-related records, e.g., experience of other mobile nodes' sessions of different time and locations, can also be adopted for the future estimated bandwidth.

2.2 Geo-Adaptive Streaming

With the popularity of mobile devices that equip the GPS function, different networking capabilities, and various sensors, the work depicted in [17][18] tried to build the bandwidth map using mobile devices, and some researches [6]-[10][9] tried to consider spatial information, which was collected by mobile devices, for adaptive streaming to get better QoE.

Riiser et al. [17] shown the average value of past available bandwidth and variance versus the path position in different traffic transportation. The goal of the work depicted in [17] is to reveal the trend of available bandwidth and variance in different communication paths. Since the purpose of [17] is to provide a dataset and motivate researchers to know the fact that the available bandwidth is related with the location, the authors did not propose a solution for geo-adaptive streaming over HTTP. Murtaza et al. [18] 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 [9] 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. [10] 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. Riiser et al. [6] shown that past bandwidth records in near locations could be referred while playing a streaming video. Riiser et al. [6] proposed a history-based prediction algorithm that uses past available bandwidth records in near locations to predict bandwidth in the future. However, the prediction method is simply to calculate the average of past bandwidth records within 100 meters of the current location, and they did not consider to have different weights for the past bandwidth records of different distances, time and dates from the mobile device to get smooth effect. Hao et al. [7] 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. [8] 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 [9] 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

This Section introduces the proposed MPEG-DASH-based system architecture and main considered technical issues.

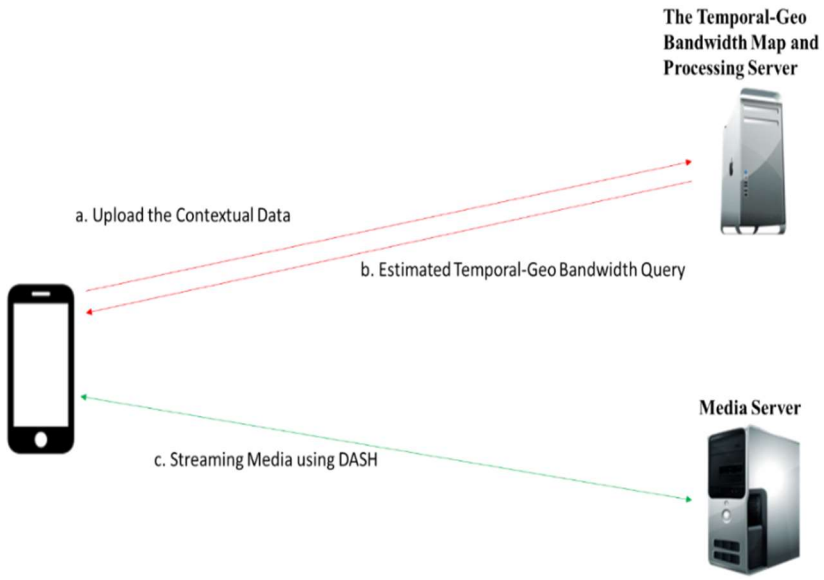

Fig. 1: Brief of the control flow of the proposed system architecture.

Fig. 1 shows the abstract control flow between a client and servers in the proposed system architecture based on MPEG-DASH. Since MPEG-DASH is a pull-based streaming technique, the client should decide which video quality to download. The client records its measured bandwidth which is associated with current location, i.e., current latitude and longitude, current time and date, and then uploads these contextual data to the server called the temporal-geo bandwidth map and processing server. When a client wants to download a segment, it requests the temporal-geo bandwidth map and processing server to calculate the smooth bandwidth of the past measured bandwidth based on the past measured bandwidth records, which are associated with time, day, and location. After the 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 video segment. The client considers buffer level, video quality of the most recently downloaded segment, the downloading rate of the most recently downloaded segment and the currently estimated bandwidth to decide which video quality of the next video segment to be downloaded from the media server.

3.1. System Components

Fig. 2 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.

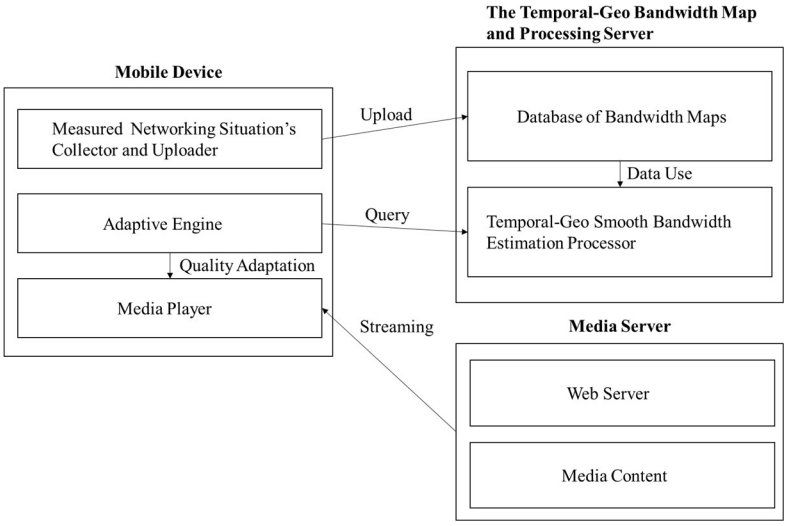

Fig. 2: System components of the proposed streaming architecture.

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 with the aforementioned calculated video quality. The media player supporting the MPEG-DASH standard follows the instruction of the adaptive engine to stream video from the media server.

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 any time and any place. Each record contains the measured bandwidth associated with 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., fitering them and doing some mathematical operations, and then the server sends the smooth bandwidth of the past measured bandwidth to the corresponding mobile device. Time, date, and location are the three axes

of the temporal-geo bandwidth map for filtering the records. That is, the temporal-geo bandwidth map and processing server filters stored records based on the client's current location, time and date, and then derives a smooth bandwidth value for 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 should provide 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 1280x720/1920x1080/ 2560x1440 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.

3.2. Main Technique Issues

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. The MPEG-DASH standard states that a media file is partitioned into one or more segments with a media presentation description (MPD) describing segments' information. Thus, the mobile client must download the MPD at first and then the segments. 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. Time, date, and location are the three axes in the temporal-geo bandwidth map database, and 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 details of the proposed temporal-geo bandwidth estimation method.

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

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

where α is a parameter and $0 \le \alpha \le 1$, 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.

$$G_d^t = (1\text{-}\beta) * G_{d-1}^t + \beta * X_d^t \qquad (2)$$

where $0 \le \beta \le 1$ and X_d^t is the historical records belonging to the same day. After expanding the above formula and substituting the start day, the following formula can be derived:

$$G_d^t = (1\text{-}\beta)[(1\text{-}\beta)G_{d-2}^t + \beta X_{d-1}^t] + \beta X_d^t = (1\text{-}\beta)^2 G_{d-2}^t + (1\text{-}\beta)\beta X_{d-1}^t + \beta X_d^t = (1\text{-}\beta)^3 G_{d-3}^t + (1\text{-}\beta)^2 \beta X_{d-2}^t + (1\text{-}\beta)\beta X_{d-1}^t + \beta X_d^t = (1\text{-}\beta)^k X_{d-k}^t + (1\text{-}\beta)^{k-1} X_{d-(k-1)}^t + \cdots + (1\text{-}\beta)^2 \beta X_{d-2}^t + (1\text{-}\beta)\beta X_{d-1}^t + \beta X_d^t = (1\text{-}\beta)^k X_{d-k}^t + \sum_{i=d-k+1}^{d} (1\text{-}\beta)^{d-i} \beta X_i^t \qquad (3)$$

The aforementioned formula shows that the more past date it is, the smaller weight it has. For convenient understanding, Fig. 3 shows the tabular representation of Eq. (3).

index	d-k	d-(k-1)	…	d-2	d-1 (Yesterday)	d (Today)
Symbol	X_{d-k}^t	$X_{d-(k-1)}^t$	…	X_{d-2}^t	X_{d-1}^t	X_d^t
coefficient	$(1-\beta)^k$	$(1-\beta)^{k-1}\beta$	…	$(1-\beta)^2\beta$	$(1-\beta)\beta$	β
$G_d^t =$	$(1-\beta)^k X_{d-k}^t +$	$(1-\beta)^{k-1}\beta X_{d-(k-1)}^t$	$+\cdots+$	$(1-\beta)^2\beta X_{d-2}^t$	$+(1-\beta)\beta X_{d-1}^t$	$+\beta X_d^t$

Fig. 3: Tabular representation for Eq. (3).

X_d^t is the estimated temporal-geo bandwidth based on the proximate area and the recent time period of a specific day. For example, X_d^t denotes today's part, X_{d-1}^t denotes yesterday's part, X_{d-2}^t denotes the past second day's part, X_{d-i}^t 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^t. Let it consider the past k days. Then X_{d-k}^t is the start date for G_d^t, e.g., when k = 9, the start date is X_{d-9}^t

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).

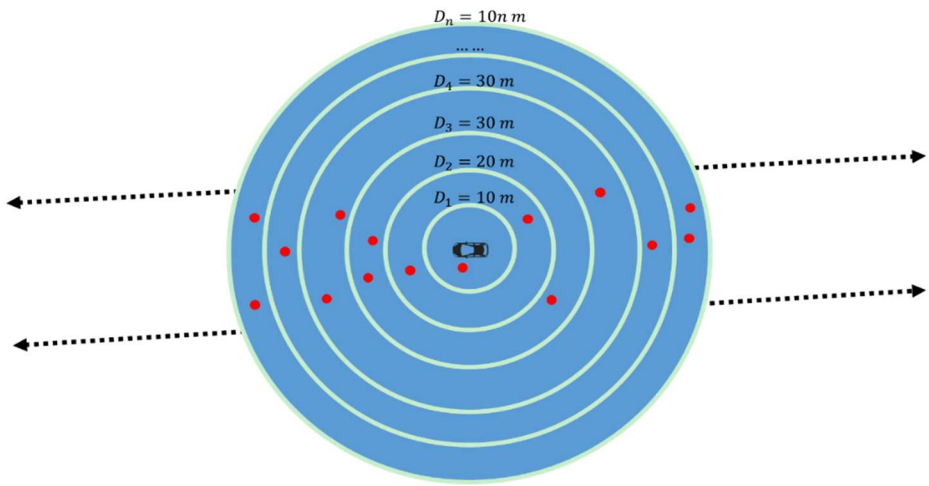

Fig. 4: The illustrated distance level for deriving X_d^t.

X_d^t 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^t 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 means to consider the situations of the previous 30 minutes and the 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 as follows: $D_1, D_2, D_3, \ldots, D_p$, where $0 < D_1 < D_2 <, \ldots, < D_p$. For example, $D_1 = 10\ m, D_2 = 20\ m, D_3 = 30\ m, \ldots, D_p = 10 * p\ m$. Fig. 4 shows an illustrated distance level for deriving X_d^t. Then the surrounding area R_i is defined as $D_i < R_i \leq D_{i+1}$. (3) There are several temporal-geo bandwidth records in area R_i, $D_i < R_i \leq D_{i+1}$. (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 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 $m, D_2 = 20$ m. Since 10 (D_1) < 17 < 20 (D_2), the record should belongs to D_2.

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

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

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 $\frac{D_p}{\sum_{j=1}^{p} D_j}$. 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, using the formula of X_d^t will result in a fatal error. Besides, it should use records with 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 stored in the server based on the time zone, for which the temporal-geo bandwidth records that are in the same time zone have the same weight. Fig. 5 demonstrates an example with the following condition: Let the time zone be 10 minutes, the time periofd 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. 5 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.

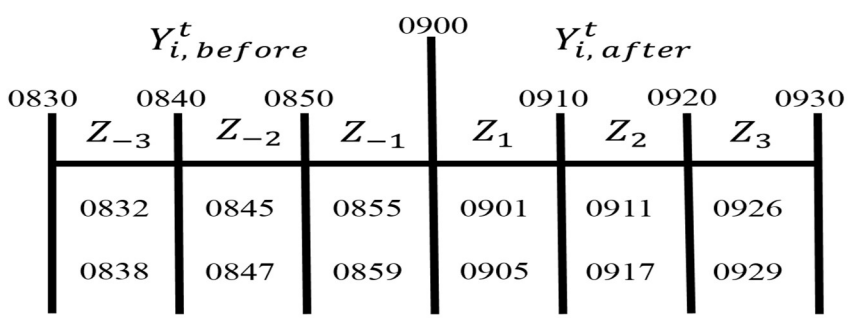

Fig. 5: An example of grouping Y_i^t.

$Y_{i,\ before}^t$ 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 $S^{[t+m*n,\quad t+m*(n+1))}$. Finally, Y_i^{t} is derived as follows:

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

However, $Y_i^{t} = Y_{i,\ befor}^{t}$ 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 design principles and details of the proposed stepwise adaptive streaming algorithm.

5.1. Main Concerns

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 thresholds in the buffered data's presentation time length (in seconds) are set in the proposed stepwise adaptive streaming algorithm: τ_{vlow} is the very low buffer threshold, τ_{low} is the low buffer threshold, τ_{high} is the high buffer threshold, and τ_{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. 6 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.

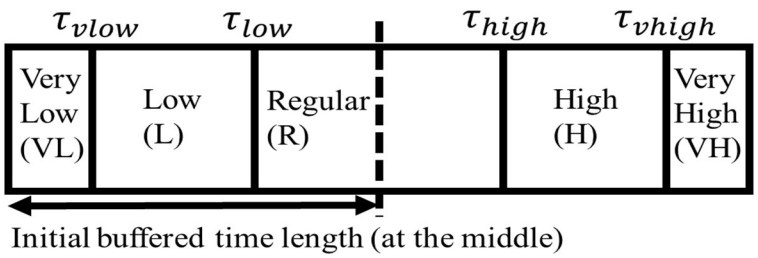

Fig. 6: Five ranges of the buffered data's presentation time length.

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

τ	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

5.2. Description

Table 1 shows the symbols that are used in the proposed stepwise adaptive streaming algorithm. Fig. 7 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 τ_{high}.

Stepwise adaptive streaming algorithm

```
switch condition do
  case τ < τ_vlow
    q ← q_l
  case τ > τ_vhigh
    Suspend downloading until the current buffered data's presentation
    time length goes back to threshold τ_high
  case τ_vlow ≤ τ < τ_low
    if B_E > r_last then
      // Low buffer but the networking situation is becoming better
      q ← q_last
    else
      if q_est > q_last then
        q ← q_last
      else if q_est = q_last then
        q ← max(q_last − p_1, q_l)
      else
        q ← max(q_last − p_2, q_l)
      end if
    end if
  case τ_high < τ ≤ τ_vhigh
    if B_E ≥ r_last then
      // High buffer and the networking situation is still becoming better
      q ← min(q_last + p_3, q_h)
    else
      if q_est > q_last then
        q ← min(q_last + p_4, q_h)
      else if q_est = q_last then
        q ← min(q_last + p_5, q_h)
      else
        q ← q_last
      end if
    end if
  case default (τ_low ≤ τ ≤ τ_high)
    if q_est > q_last + p_6 then
        q ← q_last + p_6
    else if q_est < q_last − p_7 then
        q ← q_last − p_7
    else
        q ← q_last
    end if
```

Fig. 7: The stepwise adaptive streaming algorithm.

When the buffered data's presentation time length is in the range of Low (L), i.e.,

$\tau_{vlow} \leq \tau < \tau_{low}$, there are two conditions: (1) If $B_E > r_{last}$, it means that the network situation is improving. But, 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. Statments $q \leftarrow \max(q_{last} - p_1, q_l)$ on line 15 and $q \leftarrow \max(q_{last} - p_2, q_l)$ on line 17 of Fig. 7 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., $\tau_{high} < \tau \leq \tau_{vhigh}$, there are two conditions: (1) If $B_E \geq r_{last}$, it means that the network situation is becoming better. But, 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. However, 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. 7 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 [7], 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. 8 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. 8: The experimental path used for performance analysis.

Sintel [19], which is an open content that is 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. Besides, the difference of 2 neighboring levels is about 1 Mbits/s and each quality level uses a fixed segment length of 2 seconds, 24 frames per second, and 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:

$$\int (video\ \ bitrate * dt)\ \ /\ \ \int (Available\ \ bandwidth * dt)$$

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 total segments minus 1. Higher values represent very 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:

$$\sqrt{\frac{\sum_{i=1}^{k}(i^{th}\ \ birate\ \ difference)^2}{quality\ \ switch\ \ times}}$$

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.

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

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

$$\frac{\sum_{i=1}^{k}|i^{th}\ \ birate\ \ difference|}{quality\ \ switch\ \ times}$$

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).

6.3. The General Trend of Experimental Results

For measuring the general trend of the experiment, many trips were experienced in the experimental path. To have the performance comparison, many experimental sessions are repeatedly executed to get average, standard deviation and 99% confidence level of the aforementioned evaluation metrics.

Table 2: Statistical results of using SASTGBE and the traditional adaptive algorithm.

Metric	Method	Bandwidth Usage %	Pause Time %	Suspended Time %	Suspended Times Per 20 Minutes	Quality Switch %	Average Bitrate Difference Considering Suspending (Mbps)
Average	SASTGBE	74.07	1.36	0.10	0.60	52.68	5.79
Average	Traditional Adaptive Algorithm	83.55	5.73	4.07	12.36	68.58	5.84
STD	SASTGBE	3.19	0.41	0.14	0.65	5.80	0.88
STD	Traditional Adaptive Algorithm	3.06	2.38	2.27	6.11	7.56	1.02
99% Confidence level	SASTGBE	1.61	0.20	0.07	0.33	2.93	0.44
The average of population	SASTGBE	74.07±1.61	1.36±0.20	0.10±0.07	0.60±0.33	52.68±2.93	5.79±0.44
99% Confidence level	Traditional Adaptive Algorithm	1.55	1.20	1.15	3.09	3.82	0.51
The average of population	Traditional Adaptive Algorithm	83.55±1.55	5.73±1.20	4.07±1.15	12.36±3.09	68.58±3.82	5.84±0.51

Table 2 depicted the average, standard deviation and 99% confidence level of evaluation metrics from many experimental sessions for SASTGBE and the traditional adaptive algorithm. The traditional adaptive algorithm is much worse in pause time percentage, suspended time percentage, suspended times, quality switch percentage, average bitrate difference considering suspending, and standard deviation of bitrate difference than SASTGBE. It means that using the traditional adaptive algorithm would meet the problem of pausing playback, quality switches frequently, and high quality difference between neighboring segments, which result in poor QoE, because the traditional adaptive algorithm doesn't mind how long the buffered presentation time length remains and thus tends to download the highest quality level which it could support. Besides, the traditional adaptive algorithm has no single quality change limitation and therefore it can make the problem of high quality difference between neighboring segments. The traditional adaptive algorithm is better than SASTGBE in bandwidth usage percentage because it tends to download the highest quality level that it could support rather than avoids exhausting buffer. Besides, the traditional adaptive algorithm's stand deviations of pause time percentage, suspended time percentage, suspended times, quality switch percentage, average bitrate difference considering suspending, and standard deviation of bitrate difference are much higher than those of SASTGBE. It means that the traditional adaptive algorithm is worse to absorb the fluctuation of available bandwidth than SASTGBE. Table 2 also depicts the 99% confidence level of evaluation metrics, for which

the values are very small. The small values of the 99% confidence level of evaluation metrics shown that the experimental results are not dispersed and thus are credible. The average of population with the 99% confidence level also shown that SASTGBE is better than the traditional adaptive algorithm in pause time percentage, suspended time percentage, suspended times, quality switch percentage, average bitrate difference considering suspending, and standard deviation of bitrate difference. The average of population with the 99% confidence level shown that the traditional adaptive algorithm is better than SASTGBE in bandwidth usage percentage because the traditional adaptive algorithm tends to download the highest quality level that it could support rather than avoids exhausting buffer and it also tends to make playback suspending instead of switching to a lower quality level. The aforementioned trends of the traditional adaptive algorithm harm QoE a lot.

Table 3: Statistical results of using SASTGBE and kNN IDW with 1-predict.

Metric	Method	Bandwidth Usage %	Pause Time %	Suspended Time %	Suspended Times Per 20 Minutes	Quality Switch %	Average Bitrate Difference Considering Suspending (Mbps)	Standard Deviation of Bitrate Difference (Mbps)
Average	SASTGBE	74.79	1.64	0.17	0.77	53.19	5.89	5.51
Average	kNN IDW with 1-predict	69.13	1.75	0.02	0.30	77.09	6.21	5.86
STD	SASTGBE	2.21	0.49	0.34	0.88	5.18	0.89	0.87
STD	kNN IDW with 1-predict	5.46	0.43	0.04	0.50	3.77	2.14	2.06
99% Confidence level	SASTGBE	1.12	0.25	0.17	0.44	2.62	0.45	0.44
The average of population	SASTGBE	74.79±1.12	1.64±0.25	0.17±0.17	0.77±0.44	53.19±2.62	5.89±0.45	5.51±0.44
99% Confidence level	kNN IDW with 1-predict	2.76	0.22	0.02	0.25	1.91	1.08	1.04
The average of population	kNN IDW with 1-predict	69.13±2.76	1.75±0.22	0.02±0.02	0.30±0.25	77.09±1.91	6.21±1.08	5.86±1.04

Table 3 depicted the average and standard deviation of evaluation metrics from many sessions for SASTGBE and kNN IDW with 1-predict. SASTGBE is 5.66 percent more in the bandwidth usage percentage, 23.09 percent less in the quality switch percentage, 0.32 Mbits/s less in the average bitrate difference considering suspending, and 0.35 Mbits/s less in the standard deviation of bitrate difference than kNN IDW with 1-predict. SASTGBE makes higher bandwidth utilization, lower quality switching, lower average bitrate difference considering suspending, and lower standard deviation of bitrate difference than kNN IDW with 1-predict by paying a little price in the suspended time percentage and suspended times. SASTGBE is a little worse than kNN IDW with 1-predict in the suspended time percentage and suspended times. The reason that SASTGBE is a little worse than kNN IDW with 1-predict in the suspended time percentage and suspended times is that SASTGBE limits one single quality change to a lower bound and an upper bound, which makes SASTGBE can't take a strong action while a large and long decrease of available bandwidth occurs. Besides, the kNN IDW with 1-predict's standard deviations of the bandwidth usage percentage, average bitrate difference considering suspending, and standard deviation of bitrate difference are much higher than that of SASTGBE. It means that kNN IDW with 1-predict is more unstable in bandwidth utilization, average bitrate difference considering suspending, and standard deviation of bitrate difference than

SASTGBE. Table 3 also depicts the 99% confidence level of evaluation metrics. The average of population with the 99% confidence level also shown that SASTGBE is better than kNN IDW with 1-predict in bandwidth usage percentage, pause time percentage, quality switch percentage, average bitrate difference considering suspending, and standard deviation of bitrate difference. The average of population with the 99% confidence level shown that kNN IDW with 1-predict is better than SASTGBE in suspended time percentage and suspended times because SASTGBE limits one single quality change to a lower bound and an upper bound, which makes SASTGBE can't take a strong action while a large and long decrease of available bandwidth occurs.

6.4. Experimental Results of an Individual Session

After analyzing the general trend of the experiments, an individual session's situation is shown hereafter. Table 4 and Figs. 9 to 13 are for comparing SASTGBE and the traditional adaptive algorithm. Table 5 and Figs. 14 to 18 are for comparing SASTGBE and kNN IDW with 1-predict.

Table 4 shows the measured data for the individual 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. Figs. 9 and 10 depict the estimated bandwidth and measured bandwidth in the individual session.

Table 4: SASTGBE versus the traditional adaptive algorithm.

Metric	SASTGBE	Traditionally 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
Average Bitrate Difference Considering Suspending (Mbps)	5.63	5.89
Standard Deviation of Bitrate Difference (Mbps)	5.23	5.63

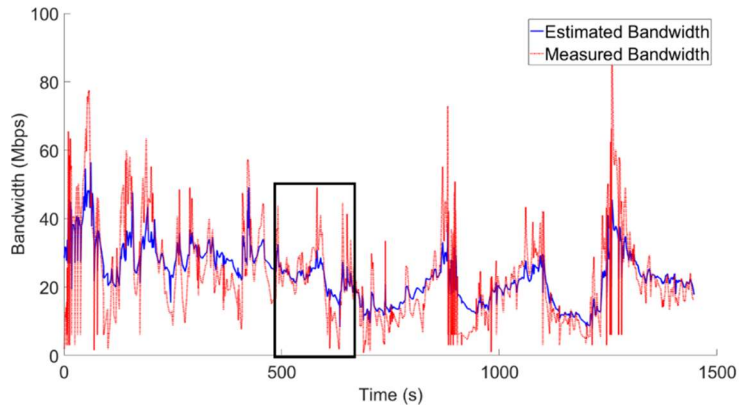

Fig. 9: Estimated bandwidth and measured bandwidth of SASTGBE.

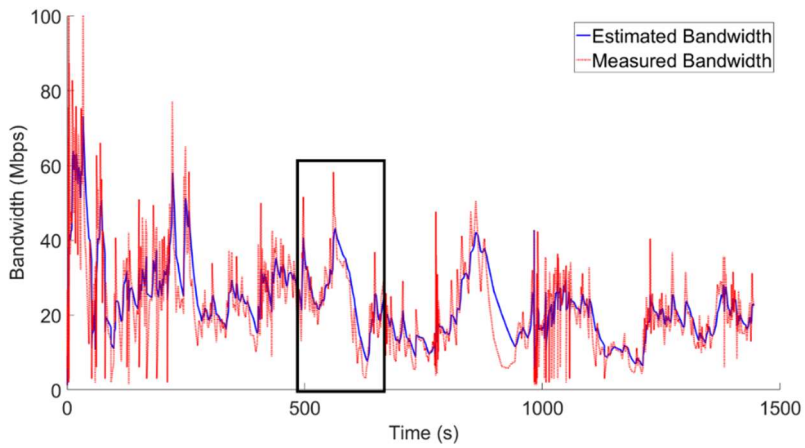

Fig. 10: Estimated bandwidth and measured bandwidth of the traditional adaptive algorithm.

Figs. 9 and 10 depict that the wireless mobile network environment has high measured bandwidth but suffered heavy fluctuation. Figs. 9 and 10 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, in the real world, 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 4 also tells the fact. The black boxes in Fig. 9 and 10 show that SASTGBE has smoother suggestion of estimated bandwidth than the traditional adaptive algorithm.

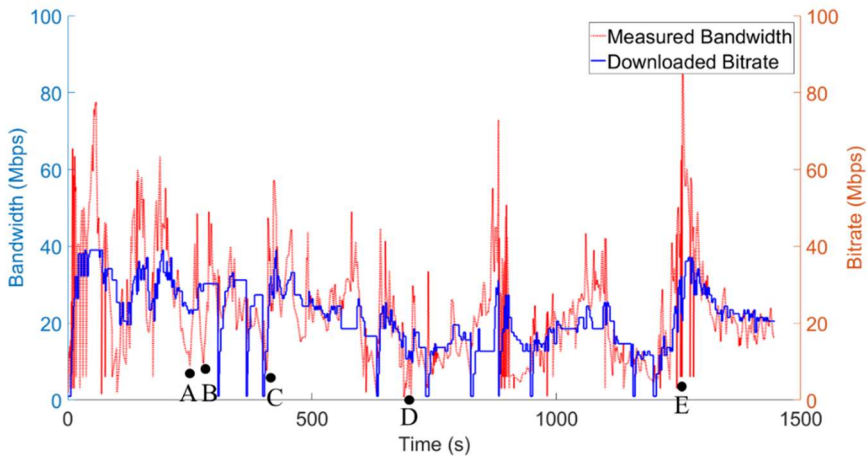

Fig. 11: Measured bandwidth and downloaded bitrate of SASTGBE.

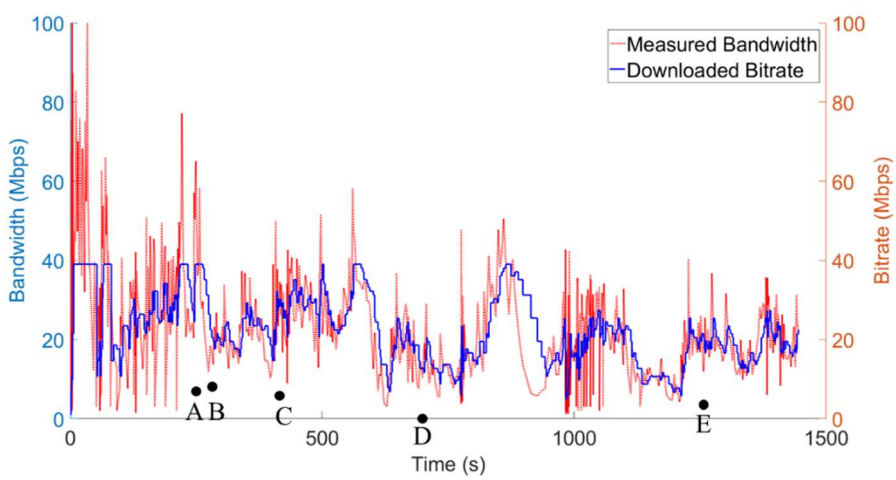

Fig. 12: Measured bandwidth and downloaded bitrate of the traditional adaptive algorithm.

Figs. 11 and 12 show the measured bandwidth and the corresponding downloaded bitrate. Fig. 11 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. 11 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. 12 shows that the traditional adaptive algorithm's downloaded segments depend only on the estimated bandwidth and therefore it would usually run into exhausted buffer, which results in frequent playback suspending (86.459 seconds and 16 times in Table 4). Besides, Fig. 12 shows that the traditional adaptive algorithm has better available bandwidth than SASTGBE in points A, B, C, D, and E.

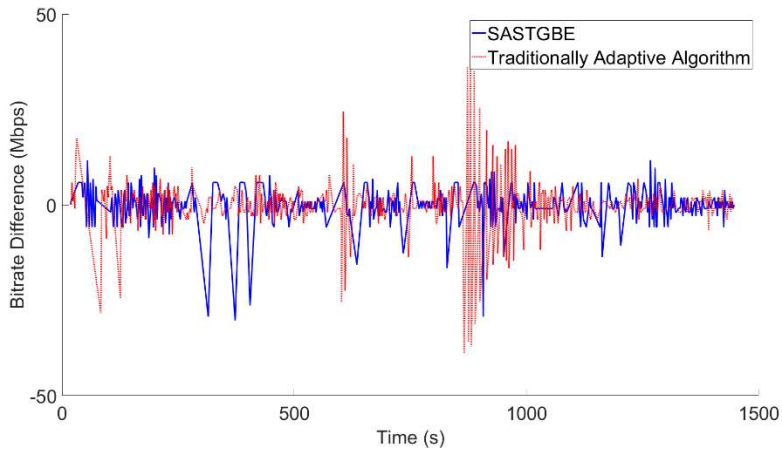

Fig. 13: Presentation bitrate difference considering playback suspending.

Fig. 13 shows presentation bitrate difference considering playback suspending, in which some illustrated the presentation bitrate difference of the traditional adaptive algorithm vibrates heavily are in the 600th-640th, the 850th-1000th, and the 1030th-1100th 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. 13 depicts that SASTGBE has sharp quality decrease for 3 times in the 300th - 500th 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. 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 300th - 500th seconds because it still has buffered data to play in this example.

Table 5 depicts the measured data for an individual 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 5: 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

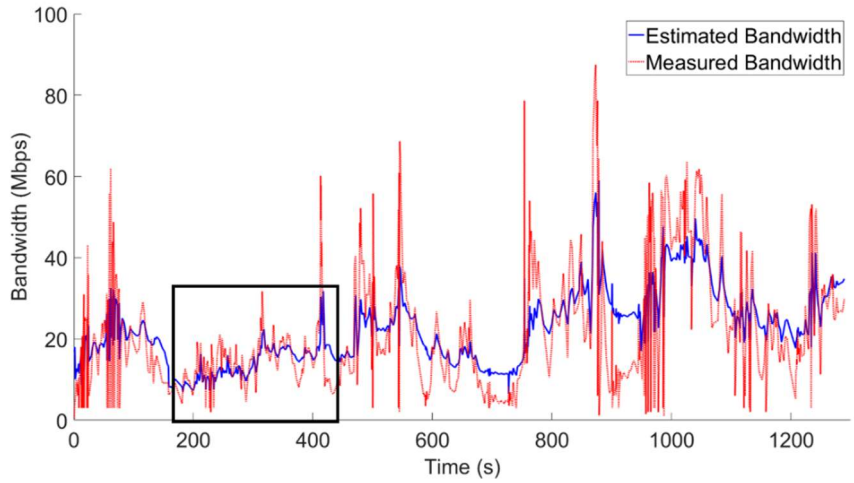

Fig. 14: Estimated bandwidth and measured bandwidth of SASTGBE.

Figs. 14 and 15 depict the estimated bandwidth and measured bandwidth in the individual session and reveals that in the real world, even if the operator has Quality of Service (QoS) control, which results in the measured bandwidth has the same general trend, there are still some little difference. The average measured bandwidth value in Table 5 also tells the fact. The black boxes in Fig. 14 and 15 show that SASTGBE has smoother suggestion of estimated bandwidth than kNN IDW with 1-predict.

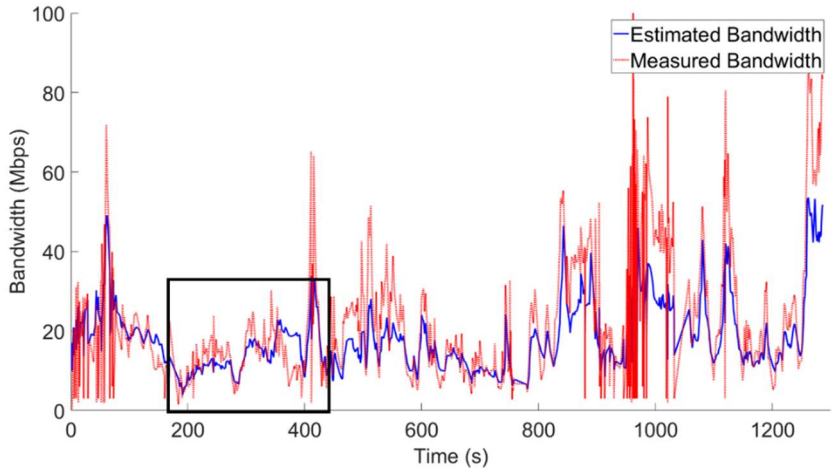

Fig. 15: Estimated bandwidth and measured bandwidth of kNN IDW with 1-predict.

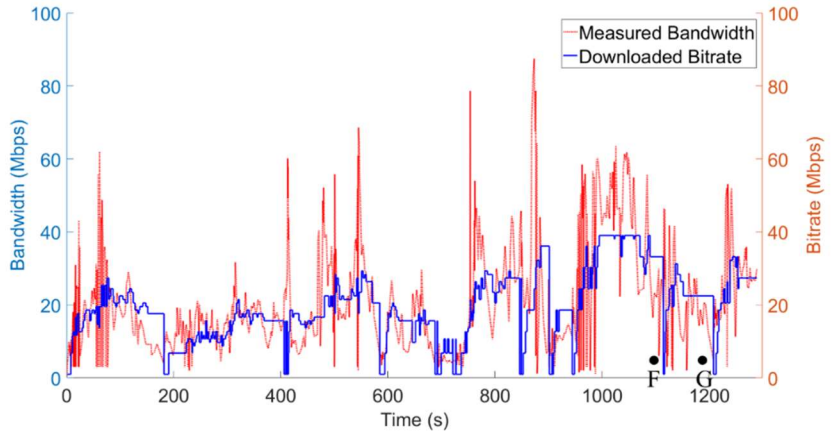

Fig. 16: Measured bandwidth and downloaded bitrate of SASTGBE.

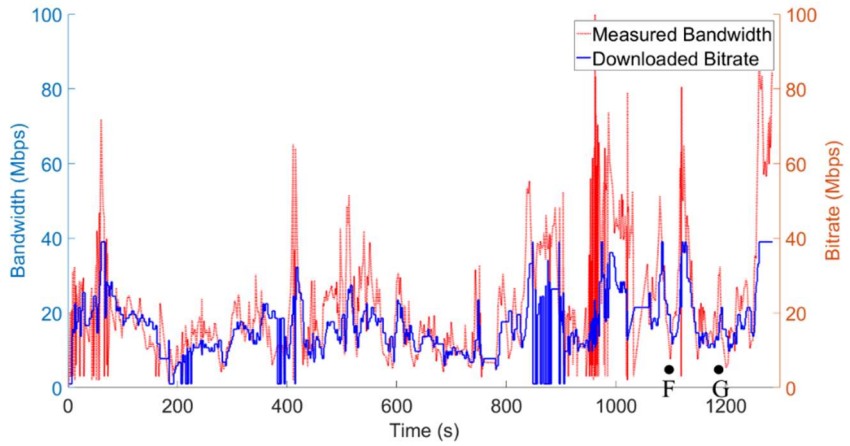

Fig. 17: Measured bandwidth and downloaded bitrate of kNN IDW with 1-predict.

Figs. 16 and 17 depict the measured 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 210th-230th seconds and the 850th-920th seconds. Points F and G in Fig. 16 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. 17.

Fig. 18 depicts the presentation bitrate difference. Fig. 18 clearly reveals that kNN IDW with 1-predict has frequent quality switch and high presentation bitrate difference in the 210th-230th seconds and the 850th-920th seconds, and this phenomenon results in poor QoE. 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 important role in the high measured bandwidth but high network fluctuation wireless mobile environment. Fig. 18 also depicts that SASTGBE has sharp quality decrease for 2 times in the 1100th - 1250th 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.

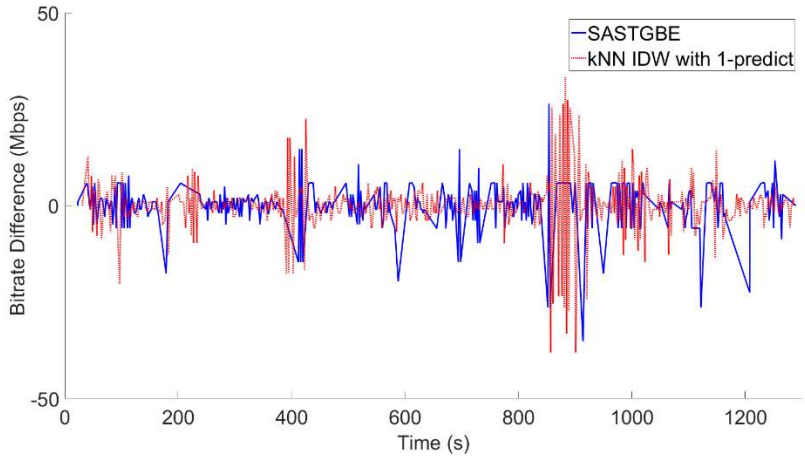

Fig. 18: Presentation bitrate difference considering playback suspending.

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 twofold: to have the real time response, it needs to combine with 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 quick response. The second concern is to transform the static parameters of the temporal-geo bandwidth estimation formulas into a dynamic decision-making mechanism. That is, replacing the static parameters with an exquisite mechanism that can dynamically adjust the weighting factors of the temporal-geo bandwidth estimation formulas according to the difference between the actual value and the predicted value. Since the available bandwidth of a moving device in the wireless mobile network varies a lot, a dynamic decision-making mechanism can be more accurate than using static parameters.

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, no. 1, pp. 469-492, 2015.

[2] T. Zhao, Q. Liu, and C. W. Chen, "QoE in Video Transmission: A User Experience-Driven Strategy," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 285-302, 2017.

[3] J. Jiang, V. Sekar, and H. Zhang, "Improving fairness, efficiency, and stability in HTTP-based adaptive video streaming with FESTIVE," *in Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies*, pp. 97-108, Nice, France, 2012.

[4] 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 (PV)*, pp. 173-178, 2012.

[5] 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*, pp. 109-120, Nice, France, 2012.

[6] H. Riiser, T. Endestad, P. Vigmostad, C. Griwodz, and P. Halvorsen, "Video streaming using a location-based bandwidth-lookup service for bitrate planning," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 8, no. 3, pp. 1-19, 2012.

[7] 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*, pp. 259-270, Singapore, Singapore, 2014.

[8] 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*, pp. 1-13, 2016.

[9] 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*, pp. 888-897, Amsterdam, The Netherlands, 2016.

[10] J. Yao, S. S. Kanhere, and M. Hassan, "Improving QoS in High-Speed Mobility Using Bandwidth Maps," *IEEE Transactions on Mobile Computing*, vol. 11, no. 4, pp. 603-617, 2012.

[11] ISO/IEC 23009-1. (2014) Dynamic adaptive streaming over HTTP(DASH)-Part1: Media presentation description and segment formats.

[12] J. Hwang, J. Lee, and C. Yoo, "Eliminating bandwidth estimation from adaptive video streaming in wireless networks," *Signal Processing: Image Communication*, vol. 47, pp. 242-251, 2016.

[13] K. Miller, A.-K. Al-Tamimi, and A. Wolisz, "QoE-based low-delay live streaming using throughput predictions," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM),* vol. 13, no. 1, pp. 1-24, 2016.

[14] A. Sobhani, A. Yassine, and S. Shirmohammadi, "A video bitrate adaptation and prediction mechanism for HTTP adaptive streaming," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 13, no. 2, pp. 1-25, 2017.

[15] S. Wilk, S. Schönherr, D. Stohr, and W. Effelsberg, "EnvDASH: An Environment-Aware Dynamic Adaptive Streaming over HTTP System," *in Proceedings of the ACM International Conference on Interactive*

Experiences for TV and Online Video, pp. 113-118, Brussels, Belgium, 2015.
[16] A. R. Bhat and S. K. Bhadu, "Machine learning based rate adaptation in DASH to improve quality of experience," *in Proceedings of the 2017 IEEE International Conference on Smart Technologies and Management* for Computing, Communication, Controls, Energy and Materials (ICSTM), pp. 82-89, 2017
[17] H. Riiser, P. Vigmostad, C. Griwodz, and P. Halvorsen, "Commute path bandwidth traces from 3G networks: analysis and applications," *in Proceedings of the 4th ACM Multimedia Systems Conference*, pp. 114-118, Oslo, Norway, 2013.
[18] 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 (ICC)*, pp. 2454-2459, 2014..
[19] https://durian.blender.org/.