Proposed an Adaptive Bitrate Algorithm based on Measuring Bandwidth and Video Buffer Occupancy for Providing Smoothly Video Streaming

Saba Qasim Jabbar
School of Electronics and
Information Engineering, Huazhong
University of Science and
Technology, Wuhan, China

Dheyaa Jasim Kadhim
Electrical Engineering Department,
College of Engineering, University
of Baghdad, Baghdad, Iraq

Yu Li
School of Electronics and
Information Engineering, Huazhong
University of Science and
Technology, Wuhan, China

***Abstract*—Dynamic adaptive streaming via HTTP (DASH) has been popular disseminated over the Internet especially under the circumstances of the time varying network, which it is currently the most challenging for providing smoothly video streaming via high quality. In DASH system, after completing the download of one segment, the player estimates the available network bandwidth by calculating the downloading throughput and then adapting the video bitrate level based on its estimations. However, the estimated bandwidth in the application layer is not accurate due to off-intervals appearance during the downloading process. To avoid the unfairness of bandwidth estimation by the clients, this work proposes a logarithmic approach for received network bandwidth, which includes increasing or decreasing this bandwidth logarithmically to converge the fair share bandwidth (estimated bandwidth). After obtaining the measured bandwidth, an adaptive bitrate algorithm is proposed by considering this measured bandwidth in addition to video buffer occupancy. The video buffer model is associated with three thresholds (i.e. one for initial startup and two for operating thresholds). When the video buffer's level stays between the two operating thresholds, the video bitrate will keep unchanged. Otherwise, when the buffer occupancy is too high or too low, an appropriate video bitrate is chosen to avoid buffer overflow/underflow. Simulation results show that the proposed scheme is able to converge the measured bandwidth to the fair share bandwidth very quickly. Also the proposed scheme is compared with conventional scheme, we found that our proposed scheme outperforms in achieving the best performance in terms of efficiency, stability and fairness.**

***Keywords*—DASH; video streaming; video buffer; video adaptive bitrate algorithm, QoE**

I. Introduction

In last few years, HTTP adaptive video streaming has been widely used in Internet video technologies such as Microsoft Smooth Streaming, Netflix, Apples HLS, Adobes HDS and Akamai HD [1], [2], since the use of HTTP over TCP is easy to configure and simplify the traversal of firewalls. Moreover, the standard of HTTP based adaptive video streaming is the Dynamic Adaptive Streaming over HTTP (DASH) [3]. In DASH systems, each video is encoded into multiple representations of different bitrates and each representation is divided into multiple segments (2-10 seconds of video time). At the client side, a client requests and receives video segments from DASH servers that own the segments continuously. One of the most important features in DASH system is adapting the video bitrate to a varying network bandwidth dynamically. In such way, DASH clients' can enjoy with video at maximum quality possible since they receive video segments from different versions each of which being encoded with specific bitrates in a way throttling the visual quality to match the available network bandwidth.

It is important challenging to satisfy the user experience during a whole video session under time-vary network conditions. Without a powerful adaptive bitrate algorithm, DASH's client may face frequent interruptions and which it is degraded his video quality. For example, if the selected video bitrate is higher than the available bandwidth, it will cause network congestion. On the other hand, if the video bitrate is lower than the available bandwidth, the visual video quality would not reach the maximum allowed by the available bandwidth. Besides, smooth video bitrates and stable bitrates switching are preferred through video playback [4]. Due to the On-Off phase which it is a natural phenomenon in DASH system (ON means downloading a segment and OFF means staying idle), the competing clients may face difficultly in estimating their bandwidths which usually leads to several performance problems such as inefficiency, instability, and unfairness [5].

There are many research problems about video streaming for multiple DASH clients over TCP network still open and challenging. As an example, the tradeoff between the stability and efficiency which it is very important issue to have smooth video bitrates. Under time-varying bandwidth condition, requesting a low video bitrate will produce more opportunities for rate selection, and therefore well ensure high stability (smoothness) with continuous video playback but it also causes a low video quality with low bandwidth usage (inefficiency). Another tradeoff challenging is between sensitivity and stability, since the channel bandwidth is inherently different in time, the high sensitivity of the bitrate control technique usually makes the video bitrate identical with the bandwidth and hence leading to high instability. Such these challenges become more troublesome when multiple clients compete on the bottleneck link because each client will try to optimize the quality of the video without looking at others. Besides, the fairness problem arises for multiple clients in DASH technology since they are deployed via HTTP/TCP network. The existing bitrate adaptive algorithms for DASH system are aiming to either achieve the efficiency of high bandwidth usage for video adaptation bitrate to match available bandwidth, or maintain continuous video playback by homogeneity the video bitrate to avoid buffer overflow/underflow such as bandwidth-based approaches discussed in [6]-[8] and buffer-based approaches in [9][10].Recently, the problems with video streaming for multiple DASH clients competing over a common bottleneck have been studied in many researches such as [11]-[13].

In [11] and [12], the authors studied problems of bitrate adaptation and determined the causes of many unwanted interactions that arise as a result of modifying the video bitrate over HTTP. Furthermore, a series of techniques have been developed that have tried to systematically guide the tradeoffs between stability, fairness and efficiency. However, these techniques did not look well in showing the factors that affect the quality of experience, such as the video bitrate oscillation and video playback interruptions. While in [13], a Markov Decision Process (MDP) is used to deal with the stochastic decision problem, which reduces both the number of playback interruptions and the number of quality level switches while increases the quality of experience. Besides rate adaptation, there are some other research works addressing the fairness problem from several aspects such as the work in [14] aims at achieving relative fairness at the packet level through the implementation of the weighted fair list. At [15], a new protocol is proposed in the context of complex multi-server adaptation, aimed at improving the user experience by providing the best fairness, efficiency and stability. The problem of fairness has been addressed by the application of a server on traffic shaping as in [16]. In our work, a bitrate adaptive approach is designed on the basis of stability and fairness of DASH system under the scenario that many clients compete for network resources.

In this work, to avoid the unfair bandwidth that was estimated by the client due to off intervals during the downloading process, an increment method is designed based on estimated bandwidth for converging the measured bandwidth to the estimated bandwidth through increasing scheme. Exclusively, when the measured bandwidth is smaller than the estimated bandwidth (i.e. estimated bandwidth is equal to the size of a segment divided by the time it takes to download this segment), a logarithmic law based increment scheme is designed for converging the measured bandwidth toward share bandwidth in efficient way among competing clients. In contrast, when measured bandwidth is higher than estimated bandwidth, a conservative reducing scheme is designed for reducing the measured bandwidth to avoid congestion. On other side for stable and smooth video bitrate, we propose an adaptive bitrate algorithm that takes the measured bandwidth based on increment scheme and the buffer occupancy level into account for keeping a continuous video playback and selecting the best video bitrate to download the next segment. In our approach video buffer is associated with three predefined thresholds for keeping the buffer in stable state. Using this model, clients will download video segments continuously without the need for OFF interval. Especially, when the share bandwidth is higher than the selected video bitrate (with the condition the video bitrate not exceed the measured bandwidth), the buffer occupancy will increase and converge toward the maximum level. The adaptive algorithm will switch the bitrate in a way preventing buffer overflow, not idle and vice versa. Through simulation results, our approaches outperform the existing algorithms in measuring the fair share bandwidth, achieving fairness, buffer stability and reducing the number of video bitrates switching.

II. System Design

Generally HTTP clients operate the media content that transferred from streaming server via available network bandwidth. The Server-Client model for streaming system is shown in Fig. 1 below, where the network bandwidth represents the data producing rate and video content's bitrate represents the data consuming rate to the video buffer of the client.

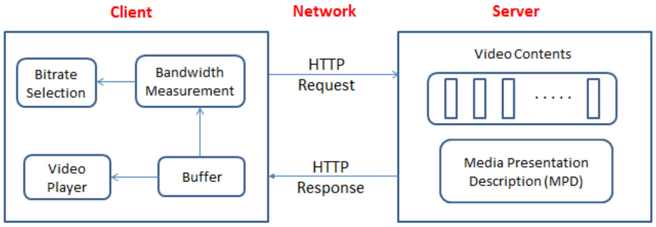


Fig. 1. Server-client model for streaming system.

Assuming at the video playing time, the current duration being playbacked is fully loaded, the next duration's playback time is from t_s to te as shown in Fig. 2, and T_i denotes the network bandwidth. While V_{Ri+1} defines the video bitrate for the next duration i+1 to guarantee there is no playback interruption during the future interval $[t_s, t_e]$.

At the beginning of each download stage, an adaptation algorithm will select the suitable bitrate of the next segment to download $V_{Ri+1} \in V,\ V = \left\{ V_R^1, ..., V_R^n, ..., V_R^L \right\}, 1 \le n \le L$ and $V_R^1 < V_R^2 < ..... < V_R^L$, which it will specify how much time is giving for the current segment download until the next download request, i.e. $t^{est}(i+1)$, which it is the estimated required time for downloading the next segment). Hence the client initiates an HTTP GET request to the server for the segment of sequence number i+1 with V_{Ri+1} then the downloading is starting. Let $t^m=[t_e- t_s]$ be the measured time that is required to complete the download. Assuming that no pipelining of downloading is involved, the next download step starts after the following time that can be expressed as follows:

$$t^{act}(i+1) = \max(t^{est}(i+1), t^m(i+1)) \tag{1}$$

where $t^{act}(i+1)$ is the actual time to download $(i+1)^{th}$ segment. If the download duration t(i+1) is shorter than the $t^{est}(i+1)$ then a client must wait time $[t^{est}(i+1)- t^m(i+1)]$ which is the off-interval before starting the next downloading step (case A), otherwise the client starts the next download step immediately after the current download is completed (case B).

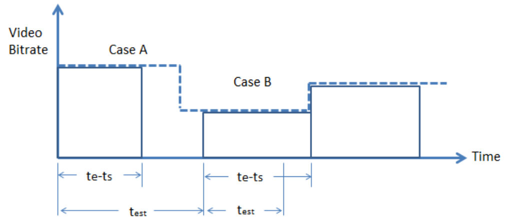


Fig. 2. HTTP segment downloading process.

III. Shared Bandwidth Approach

To explain the effect of bandwidth consumption on adapting the suitable video bitrates, three different scenarios are considered: ideal participated, more participated and less participated which they represent respectively the amount of bandwidth asked by client to be equal to, more than and less than the shared bandwidth. We assume a temporal overlap of the ON-OFF intervals among two competing clients. Suppose the available bandwidth is T and two active connections share this bandwidth fairly, i.e. each has T/2.So T_1 and T_2 are the bandwidth received by client1 and client2 respectively. In ideal participated, the total amount of traffic is requested by the two clients perfectly filling the link, i.e. each client exploits the available bandwidth during its ON period $T_1= T_2 =T/2$. If the shared bandwidth is overestimated, new segments with high video bitrates maybe asked by the clients who lead to congestion. In this case, the current estimated bandwidth is less than the previous estimation, and hence clients will turn to ask segments with lower video bitrates. This fluctuation in video quality level leads to video buffer instability as shown in Fig. 3(a). Fig. 3(b) shows the case of more participated where the ON duration of one client falls in the ON duration of the other client. This happened when the clients are still asking video segments for different video bit-rates. In this situation, the monitored throughput by the clients is $T_1>T/2$ and $T_2 =T/2$, i.e., first client overestimates the fair share of bandwidth.

When only one client overestimates the fair share of bandwidth, the two clients will converge to a stable but unfair equilibrium that the client, who overestimates the fair share of bandwidth, will request a higher video bitrate, causing unfairness. While Fig. 3(c) discusses the problem of imperfect usage of the shared bandwidth when the ON duration for the two clients are aligned, the available bandwidth at the server side is discrete and limited in this case clients estimate the fair share of bandwidth correctly, it still may cause imperfect utilization problem. For example if the available shared bandwidth is not T/2, clients will ask segments with video bitrates smaller than T/2 to avoid buffer underflow. The OFF periods should adopt to solve the buffer overflow case and imperfect utilization problem happens.

Accordingly, the bandwidth oscillation will impact on video bitrates stability when the bandwidth is in more participated case (i.e., the congestion occurs), then the bandwidths are estimated by the clients approximately equal, and fairness can be obtained. When congestion occurs, the video buffer at the client side will be drained, causing playback stop since clients may ask segments with bitrates more than the shared bandwidth, which may degrade the video quality of experience.

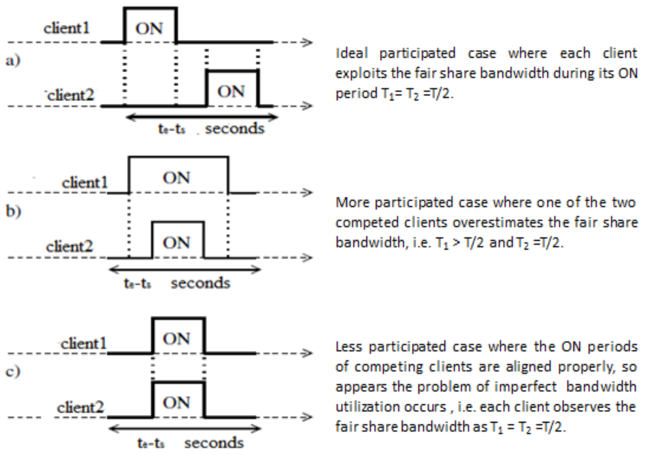

a)
client1
ON
client2
ON
te-ts . seconds
Ideal participated case where each client exploits the fair share bandwidth during its ON period T1= T2 =T/2.
b)
client1
ON
client2
ON
te-ts seconds
More participated case where one of the two competed clients overestimates the fair share bandwidth, i.e. T1 > T/2 and T2 =T/2.
c)
client1
ON
client2
ON
te-ts seconds
Less participated case where the ON periods of competing clients are aligned properly, so appears the problem of imperfect bandwidth utilization occurs , i.e. each client observes the fair share bandwidth as T1 = T2 =T/2.

Fig. 3. Two clients are competing shared bandwidth during download a segment.

When the bandwidth is in less participated case, the requested video bitrate is smaller than the available bandwidth, and OFF periods is needed to stop the transmission so as to avoid buffer overflow. In this case, when the client buffer reaches the maximum level Bmax, this will lead to bandwidth overestimation and the video bitrates will oscillate accordingly. But the available shared bandwidth cannot continue with the same video bitrate, then video buffer falls below Bmax, so the client returns to the buffer growing mode and the off-intervals disappear, so the link again is going to be in more participated case and the measured throughput starts to converge to the fair-share bandwidth. Finally, due to the quantization impact, the requested video bitrate falls below the fair-share bandwidth, and the client buffer starts growing again, completing one oscillation cycle.

IV. Video Bitrate Adaptive Approach

In this section, we will describe a four stages model for DASH bitrate adaptation algorithm, which includes both the conventional algorithms (e.g., [17], [18]) and our proposed bitrate adaptation algorithm in this work, which will serve as a benchmark. Essentially, a bitrate adaptation algorithm proceeds in the following four stages:

***a)* Estimating Stage:** The algorithm begins the first step in adaptation process by estimating the network throughput which be used in selecting the video bitrate. The current shared bandwidth equates to the previous TCP throughput that monitors during the ON interval, i.e.:

$$\hat{T}_i = \tilde{T}_{i-1} \qquad (2)$$

***b)* Filtering Stage:** Then a filtering process is done by using a smoothed function fs() with the aim of removing outliers. The algorithm takes the measured history as an input i.e.:

$$\tilde{T}_i = f_s(\{\hat{T}_j : j \le i\}) \qquad (3)$$

***c)* Adapting Stage:** The continuous measured throughput is adapted to get the suitable video bitrate. In this step a discrete video bitrate is selected based on side history information such as video buffer occupancy, previous selected video bitrate V_{Rj}, …, etc. Then the current video bitrate is given by:

$$V_{Ri+1} = f_a(\tilde{T}_{i+1};\ V_{Rj}, B(j) : B(j) < B_{max}, j \le i) \qquad (4)$$

d) **Next Segment Downloading Stage:** A scheduling time for download the next segment is determined by the adaptive algorithm as a mechanical function based on video buffer size. If the current buffer occupancy B(i) is less than the maximum level B_{max}, then the estimated time for next download is set to 0 and the download time for next segment begins after the current download is finished. Then the estimated time for next down is given by: But the time is τ (. i.e. :

$$t^{est} = \begin{Bmatrix} 0, & B(i) < B_{max} \\ \tau, & B(i) = B_{max} \end{Bmatrix} \qquad (5)$$

where τ is the video segment period to stop the video buffer from growing.

In general, all of today's commercial DASH clients implement the first stage and last stage of the rate adaptation algorithm in a similar way, though they may differ in their implementation of the second stage and third stage of the algorithm. In this work we make some modifications to these stages since the off intervals may become a source of ambiguity for a client to correctly perceive its fair share of the network bandwidth, thus preventing the client from making accurate rate adaptation decisions. To ensure fairness, we focus on preventing the happenstance of OFF phenomenon, that is, the bandwidth is either in the ideal participated or more participated. In the case of more participated, fairness is guaranteed, while in the ideal of participated, there is an unlimited number of bandwidth-sharing modes. It is worth to note that although it is easy to avoid OFF phenomenon by keeping on downloading without break with them, it is impossible even to distinguish if the bandwidth is in more participated or in the ideal participated.

Moreover, because the available video bitrate is limited and discrete, instead of using the estimated bandwidth, we suggest an effective bandwidth increment scheme to guide the adaptive video bitrate. The stability is taken into account through proposing double threshold of buffer level, and then the frequency of filling the buffer occupancy and the impact of bandwidth variations on the adaptive video bitrate can be reduced. As per during the rate adaptation process, the smoothness is also taken into account and it will be discussed deeply in next section. In DASH system, the video bitrate can be changed only when a segment is completely downloaded. Then, the video bitrate for the next segment to be downloaded is calculated accordingly, we first estimate the network bandwidth and filter it. Then, with this estimated bandwidth, we update the incremented bandwidth which will be used to guide the rate adaption. According to the buffer occupancy, the video bitrate is adapted when the buffer occupancy is between the two predefined thresholds to avoid buffer overflow and underflow.

V. Smoothing Network Bandwidth

In this section, we will describe the principles of our approach to smooth the network bandwidth estimation and how filtering it, also we will describe our Trial increment approach based on bandwidth estimation and then we will explain in details our proposed stable and smooth adaptive bitrate algorithm.

A. Network Bandwidth Estimation and Filtering

Assume the encoding of a video segment is done into different versions of L, with playback of different video bitrates $V = \{V_R^1, ..., V_R^n, ..., V_R^L\}$ in V set. All video versions are divided into segments of equal length, each of which consumes the same playing time τ. For each client, the streaming process is divided into sequential segment downloading steps i=1, 2, 3… then, the required bandwidth to download (i+1)th segment, is estimated as [6]:

$$\hat{T}_{i+1} = \frac{\tau\, V_{Ri+1}}{t_m^{i+1}} \qquad (6)$$

where τ is segment duration. The downloaded segments are stored in the video buffer. Furthermore, to remove the impact of noise and interference during the receiving bandwidth estimation, a noise-filter step is adopted [19]. Then, we have:

$$Y_{i+1} = \partial \hat{T}_i + (1-\partial) Y_i \qquad (7)$$

where Y_{i+1} is the available modified bandwidth after filtering step. ∂ is a smooth coefficient factor that reflects the changing network conditions; it is measured as:

$$\partial = \left| \frac{x_i}{z_i} \right| \qquad (8)$$

where x_i and z_i represent the absolute and smoothed values of the throughput deviation (e_i) respectively and they are given by:

$$\begin{aligned} x_i &= \rho e_i + (1-\rho)\, x_{i-1} \\ z_i &= \rho |e_i| + (1-\rho)\, z_{i-1} \end{aligned} \qquad (9)$$

where ρ is a constant with $0 \le \rho \le 1$. At any instant e_i is determined as:

$$e_i = \hat{T}_i - Y_i \qquad (10)$$

where e_i is difference between the measured value and adjusted value. If ∂ is a small value, the previous adjusted bandwidth may play a more important role in predicting the bandwidth.

As ∂ gets larger, the adjusted bandwidth is closer to the current measured value. Thus, if the variation of bandwidth is large, we should decrease ∂, and if the variation of bandwidth is small, we can increase ∂ to more accurately reflecting the change of the network conditions.

B. Trial Increment Approach

Due to the on-off phenomenon in the DASH system, the bandwidth estimated by the client is discretionary and cannot be used directly for the adjustment rate. Additionally, only when bandwidth is under more participated case, the congestion happens, so bandwidth estimated by client equals to a fair-share bandwidth since all clients see almost the same bandwidth as available. On the other hand, when congestion occurs, the required video bitrate cannot be supported by bandwidth, and video freezing may occur. Thus, how to get a free bandwidth quota without congestion is a critical issue to improve the performance of the rate adaptation for DASH system.

In this work, we propose a logarithmic increase scheme based on received network bandwidth, which includes increasing the bandwidth logarithmic and decrease it if the increment in bandwidth more than the fair share bandwidth for making the measured bandwidth quickly converge to the fair share bandwidth. The aim behind this scheme is that the estimated bandwidth by a client is always the upper bound of the fair share bandwidth due to the off intervals. Thus, during the increment phase, the measured bandwidth will continuously increase until it exceeds the estimated bandwidth expressed in (7). Then, when it is higher than the estimated bandwidth, the client will switch to the decrease phase for avoiding congestion.

Let $\tilde{Y}_{i+1}$ be the measured bandwidth which would be used for adapting the suitable video bitrate to (i+1)th segment and it is initialized to zero that $\tilde{Y}_{init} = 0$. Whenever the downloaded process of segment is completed, the estimated bandwidth is update according to expression (7). Then, the measured bandwidth is updated as follows:

1) Increment Phase:

$$\tilde{Y}_{i+1} = \tilde{Y}_i + \max\left(\frac{Y_i - \tilde{Y}_i}{2}, \wp\right), \quad \text{if } \tilde{Y}_i < Y_i \qquad (11)$$

where $\wp$ is a constant to avoid slow convergence, the measured bandwidth will quickly approach to the estimated bandwidth in the logarithmic way. However, when the gap between $\tilde{Y}_i$ and Y_i is small, i.e. $\left(\frac{Y_i - \tilde{Y}_i}{2} < \wp\right)$, the measured bandwidth will be additively increased by $\wp$ rather than by logarithmic approach. Otherwise when the gap between the measured bandwidth and the fair share bandwidth is large, an aggressive way should be employed to increase the measured bandwidth quickly. Practically, the client fair share bandwidth in practice cannot get, so instead its upper bound bandwidth (i.e., the estimated bandwidth) is used.

2) Decrement Phase:

$$\tilde{Y}_{i+1} = \tilde{Y}_i + \beta\left(Y_i - \tilde{Y}_i\right), \quad \text{if } \tilde{Y}_i \geq Y_i \qquad (12)$$

where β is a positive constant satisfying β > 1. When the measured bandwidth exceeds the estimated bandwidth, a conservative way should be employed to control $\tilde{Y}_i$ to be not higher than Y_i i.e. when the measured bandwidth is higher than the fair share bandwidth, congestions may happen, so leading to playback freezing. In this case, a conservative way is designed to decrease the measured bandwidth guaranteeing that no congestion happens.

C. Proposed Adaptive Bitrate Algorithm

At the client aspect, the video player uses an adaptive algorithm to determine the suitable bitrate to be selected for download the next segment. Every segment that is downloaded placed in a buffer of maximum size B_{max}. The buffer is associated with three thresholds (B_0, B_{low}, B_{high}) where $B_0 < B_{low} < B_{high}$ and V represents the set of available bitrates as cleared in Fig. 4. After getting the measured bandwidth in the previous section, to obtain a smooth and stable video bitrate with high bandwidth utilization a bitrate adaptation scheme is designed based on the buffer occupancy with the measured bandwidth. In this work, the buffer occupancy is denoted by the buffered video duration considering that the client buffer may contain segments from different versions, i.e., different video bitrates and there is no longer a direct mapping between the buffered video size and the buffered video duration.

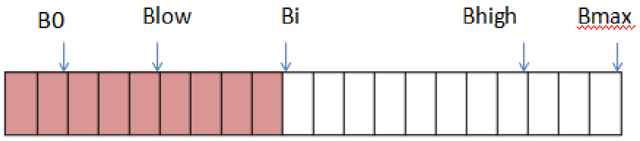


Fig. 4. Client video buffer.

The smooth bitrate adaptation scheme is designed with the aim to provide a smooth video quality and avoid buffer overflow/underflow. The impact of short term bandwidth oscillations on video bitrates can be eliminated by using thresholds for buffer occupancy. At the beginning of the video session (or after an interruption due to buffer empty state), the number of segments in the buffer is below B_0 and the video playback begins during which the lowest bitrate is selected. This step is necessary because reducing the startup time is important to prevent the user from giving away the video session.

When the share bandwidth link is in more participated case, i.e., the congestion occurs, the bandwidths estimated by the clients are approximately equal, and then fairness can be obtained. When congestion occurs, the video buffer at the client side will be underflow, causing playback stop since clients may ask segments with bitrates more than the share bandwidth, which degrades video quality of experience. So the rate adaptation scheme decreases the video bitrate to minimum level. Once the video buffer exceeds B_0 and lower than B_{low}, to avoid buffer draining and provide a continuous playback, the video bitrate should be selected no higher than the measured bandwidth. Similarly, when the buffer occupancy is higher than B_{high}, to ensure no buffer overflow occurs while achieving high bandwidth utilization, a high video bitrate can be selected. When the video buffer occupancy falls between the two thresholds B_{low} and B_{high}, the risk of buffer overflow/underflow is low, and the video bitrate should be maintained unchanged so that smooth quality is provided. Consider a video content is coded into L representations with different video bitrates. The set of video bitrates is denoted by $V = \left\{ V_R^1, \ldots\ldots, V_R^n, \ldots., V_R^L \right\}, 1 \leq n \leq L$. B_i is the current buffer occupancy after download ith segment and measured bandwidth $\tilde{Y}_{i+1}$ for next segment. The video bitrate for (i+1)th segment is selected as:

$$V_{Ri+1} = \begin{cases} \min_{n=1}\{V_R^n \mid V_R^n \in V,\ V_R^n < \tilde{Y}_{i+1}\} & \text{if } B_i \leq B_0 \\ \max_{1\leq n\leq L} \{V_R^n \mid V_R^n \in V, \frac{\tau V_R^n}{\hat{T}_{i+1}} \leq B_i - B_0, V_R^n \leq \tilde{Y}_{i+1}\} & \text{if } B_i < B_{low} \\ \min_{1\leq n\leq L} \left\{ V_R^n \mid V_R^n \in V, \frac{\tau V_R^n}{\hat{T}_{i+1}} \leq B_i - B_{low}, V_R^n \geq \tilde{Y}_{i+1} \right\} & \text{if } B_i > B_{high} \\ V_{Ri} & \text{if } B_{low} < B_i < B_{high} \end{cases} \quad (13)$$

At the starting of the video session, the number of segments in the buffer is below B_0 and the rate adaptation scheme decreases the video bitrate to minimum level. Once the buffer level goes beyond B_0, the amount of time required to load the next segment is less than the B_i - B_0, the maximal video bitrate which is no higher than the measured bandwidth is selected so as to guarantee a continuous video playback. When the buffer occupancy is higher than the B_{high}, the minimal video bitrate which is no lower than the measured bandwidth is selected so as to improve video quality; otherwise when the buffer occupancy between B_{low} and B_{high} the current bit rate is preserved.

VI. Simulation Results

The proposed scheme is evaluated using ns-3 network simulator and the topology implemented in this work is shown in Fig. 5 below. The topology consists of an HTTP server, two or more HTTP clients and a pair of network elements (i.e. x>=2). To achieve adaptive streaming, the HTTP server offers the client seven levels of representations to adapt the video rates these are V={ 356, 500, 800, 1200, 1500, 2400 and 3500Kbit/s}. The length of video segment and video buffer is 2s and 35 sec, respectively. The values of other parameters B_{low}, B_{high}, B_0, β, ρ, ℘ are 15s, 30s, 5s, 1.25, 0.5, 32kbps respectively.

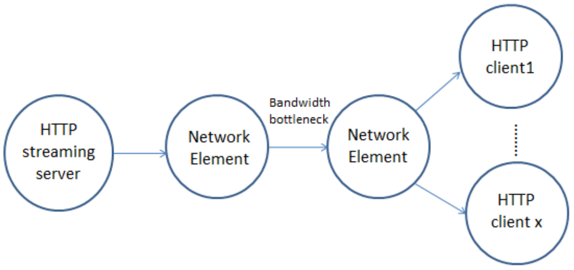
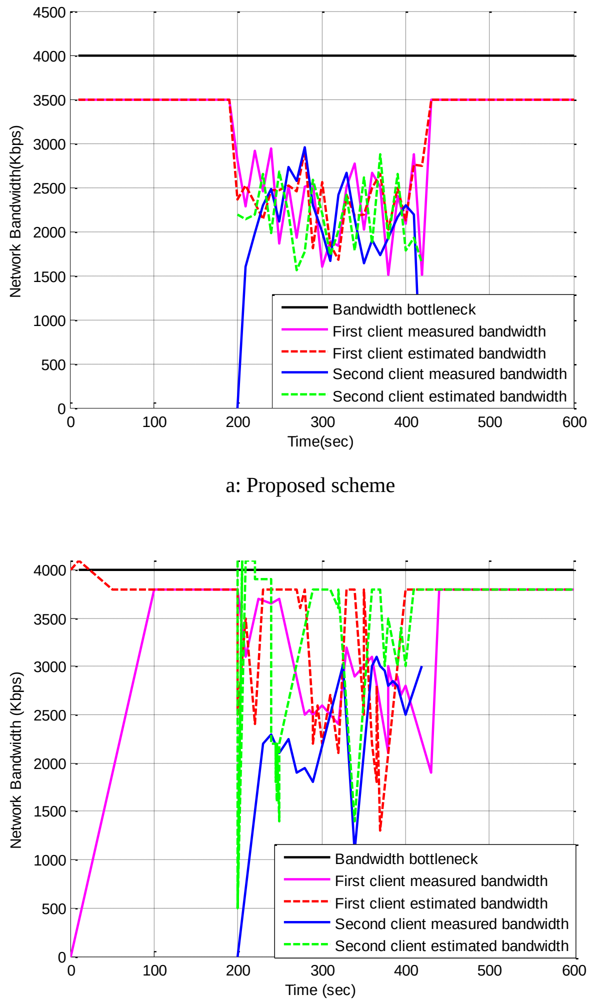


Fig. 5. Network topology configuration in the simulation.

To evaluate the performance of the proposed scheme, we compare it with a conventional scheme [6] as in Fig. 6 and 7 in term of measured bandwidth, video bitrate and buffer occupancy, respectively. The available network bandwidth is equal to 4Mbps with two clients compete on it. The first client enters at the time 0s while the second client enters the network at 200s and leaves at 400s. Fig. 6(a) and 6(b) show the ability of increment scheme in converging the measured bandwidth to the estimated bandwidth (share bandwidth) quickly through the accurate logarithmic law based increment scheme.

We can note from Fig. 6(a) and 6(b), when the second client enters the system, it needs only about 10s to follow the fair share bandwidth well, the first client detects the change in the available network bandwidth such that their measured bandwidth converge to the fair share bandwidth quickly. While the other scheme spends a long time to converge the measured bandwidth to the estimated bandwidth in linear increasing scheme based on estimated bandwidth. The figures also demonstrate that when two clients compete with each other, the compared scheme couldn't track the fair-share bandwidth well because of the slow start and frequent fluctuations, when the bandwidth changes.

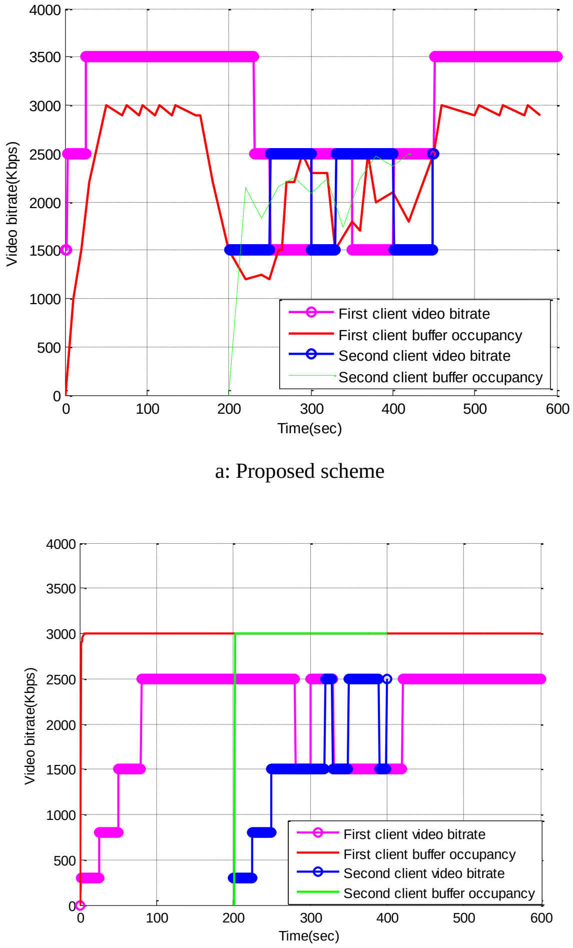


a: Proposed scheme

b: Conventional scheme

Fig. 6. Network bandwidth computed by two clients.

In Fig. 7(a) and 7(b) the video bitrate under the proposed scheme is smooth and stable than the second scheme. The conventional scheme uses the linear increasing method in measuring the share bandwidth which is used for adapting the video bitrate hence this leads to video bitrate oscillation for matching the measured bandwidth. In addition the video buffer occupancy is always high to ensure continuous video playback. Since the video bitrate is separate, the selection of suitable bitrate is usually lower than the measured bandwidth which leads to low bandwidth utilization efficiency. While the video bitrate of our scheme is smoother due to many reasons; at the beginning the adaptation of bitrate isn't depending on the measured bandwidth only, it associates with the buffer occupancy. The video buffer is related with three operating thresholds for avoiding buffer overflow/underflow and compacting the short term bandwidth oscillation that effects on video bitrates. Beside the bitrate adaptive algorithm takes many factors into account that impact on the QoE for video streaming, involving maximize the video bitrates and minimize the video start time, the number of buffering events, and the number of bitrate switching events. So from figure 8a the bitrate curve is smoother than the other scheme. Finally, no buffer overflow/underflow occurs, i.e., video playback without break is guaranteed.



a: Proposed scheme

b: Conventional scheme

Fig. 7. Buffer occupancy and video bitrate of two clients under bandwidth bottleneck.

In Fig. 8, we compare our scheme with other scheme in term of efficiency, stability, and fairness based on the following metrics [11]:

***a)* Instability metric:** Clients are likely to be sensitive to frequent and important video bitrate switches as indicated by some studies. The instability metric is defined as:

$$M_{\text{instability}} = \frac{\sum_{a=0}^{z-1} \left|V_{Ri,i-a} - V_{Ri,i-a-1}\right| \omega(a)}{\sum_{a=1}^{z} V_{Ri,i-a}\,\omega(a)} \qquad (15)$$

The instability metric equals to the weighted sum of all bitrate switch steps monitored within the last 10 segments divided by the weighted sum of bitrates in the last z=10 segments. The weight function $\omega(a)$ = i-a.

***b)* Unfairness metric:** At time t, the unfairness metric is defined as:

$$M_{\text{unfairness}} = \sqrt{1 - \text{jainFair}_t} \qquad (16)$$

where JainFairt is the index of Jain fairness at time t and is calculated based on the bitrates V_{Ri} over all clients.

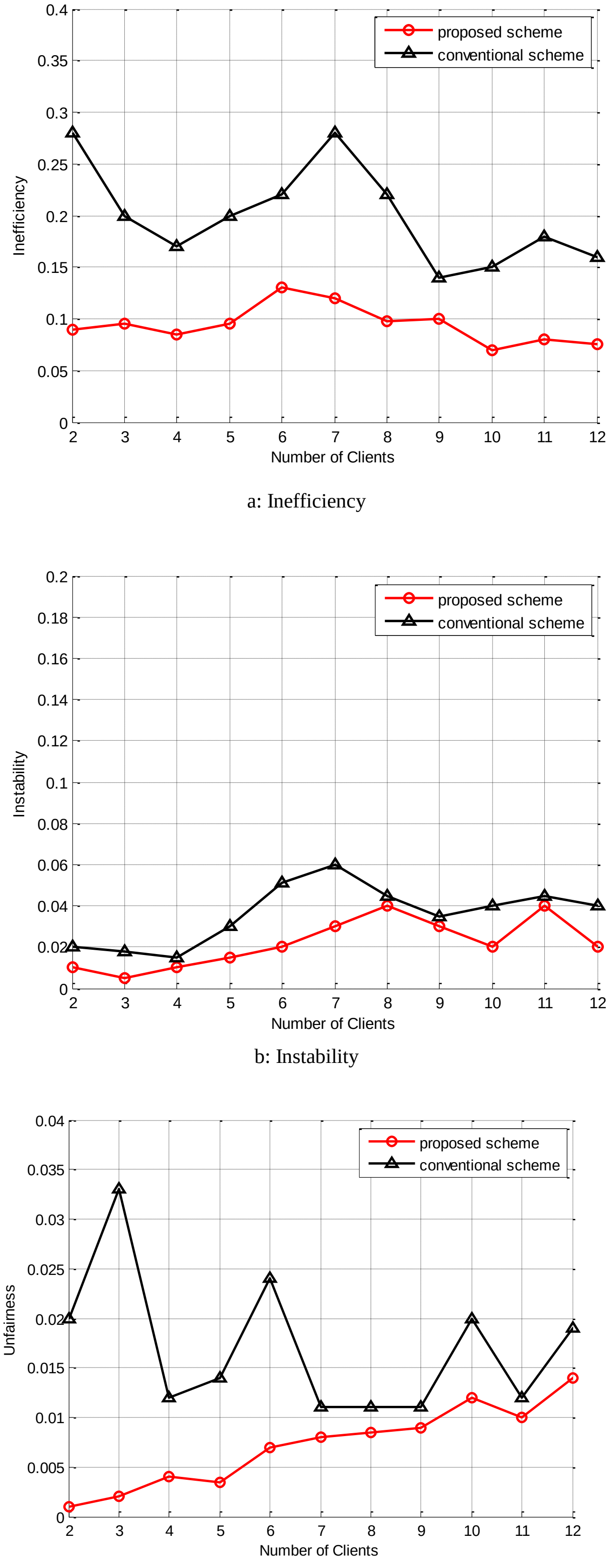
***c)* Inefficiency metric:** This metric is calculated as:

$$M_{\text{inefficiency}} = \left| \frac{\sum_j V_{Ri,j}}{T} \right| \qquad (17)$$

where $V_{Ri,j}$ is the video bitrate of i^{th} segment for j^{th} client and T is the available bandwidth. A value close to zero means that the clients in average are using as high an average video bitrate as possible to improve video quality.

In Fig. 8, the number of clients is changed from two to twelve with the available network bandwidth fixed at 8Mbps in order to evaluate the performance of our scheme with other scheme. From the figure the proposed scheme is outperforming due to the increment scheme based on bandwidth measurement also video buffer model is associated with three operating thresholds for preventing buffer underflow /overflow and the robust of the adaptive algorithm based on estimating the download time of next segment based on segment size.

In Fig. 9, our scheme always achieves the lowest inefficiency, instability, and unfairness, i.e., compared with conventional scheme. The proposed scheme can provide higher and smoother video bitrate, and besides, it can also better guarantee the fairness between the clients. As the available bandwidth increases, the performance does not linearly increase or decrease. This is because the video bitrate is discrete so that for a given available bandwidth, if the fair-share bandwidth is approximately equal to the video bitrate, generally better performance can be achieved, and vice versa.

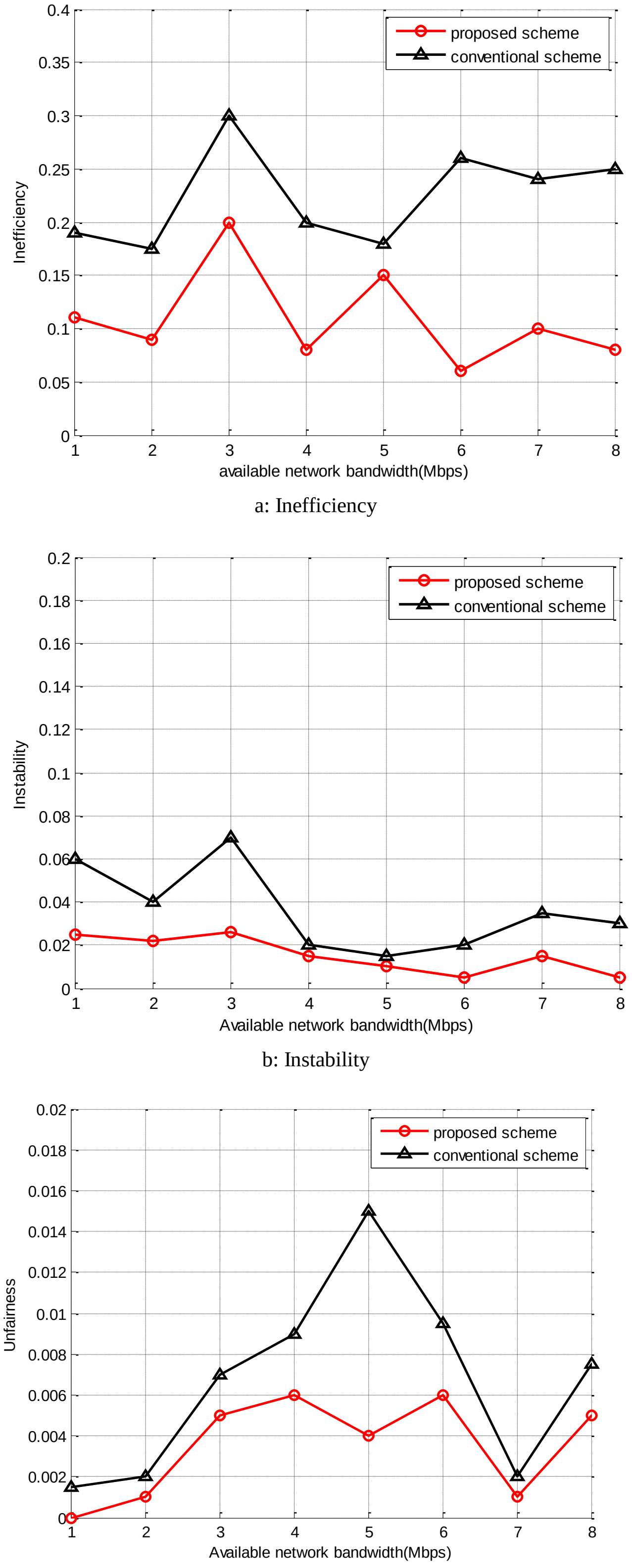


a: Inefficiency

b: Instability

c: Unfairness

Fig. 8. Performance evaluation when the available network bandwidth is fixed at 8 Mbps, and the number of competing clients varies from 2 to 12.



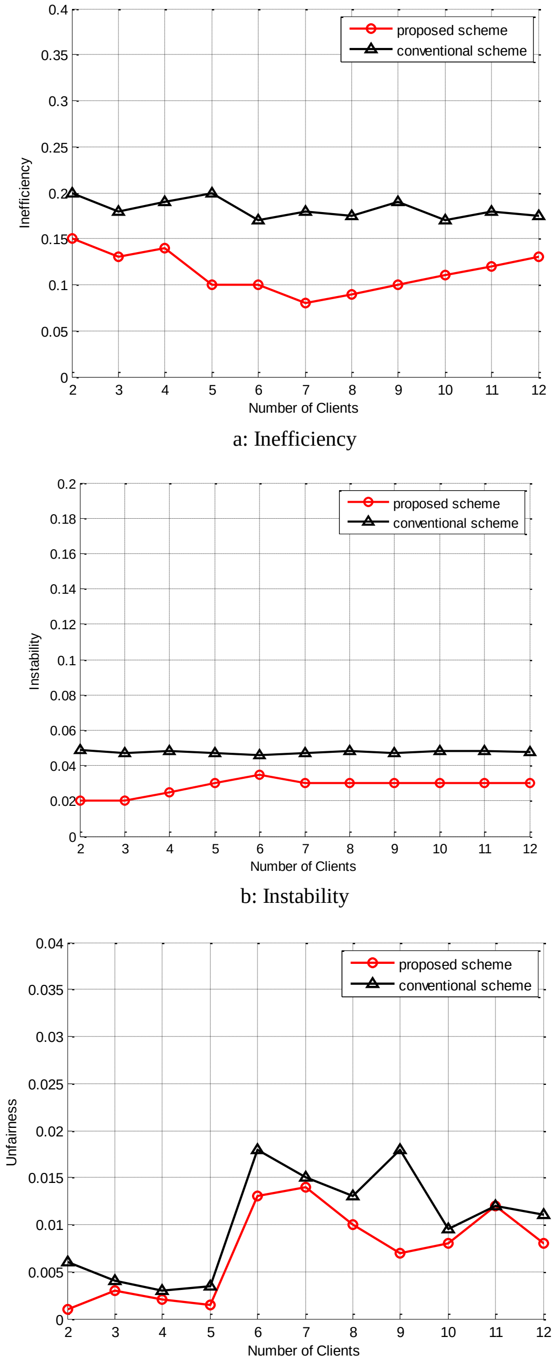
a: Inefficiency

b: Instability

c: Unfairness

Fig. 9. Performance evaluation for the two schemes when the available network bandwidth is changing from 1Mbps to 8Mbps with two competing clients.

From the figure our scheme has better performing in all situations as in (a) and (b) , the other scheme has inefficient bandwidth utilization, instability video bitrate because the decision of video bitrate is based on the measured fair share bandwidth, which is always the same (equal to 1 Mbps) in this experiment . In term of fairness, the other scheme fluctuates with increasing the no. of clients while the proposed scheme performs better since fairness is easy to be impacted with increasing the number of competing clients (Fig. 10).



a: Inefficiency

b: Instability

c: Unfairness

Fig. 10. The performance assessment of two schemes under changing both the number of competing clients and available network bandwidth.

VII. Conclusion

In this work, a bitrate adaptive approach is designed on the basis of stability and fairness in DASH system under the scenario that many clients compete for network resources. The proposed scheme is based on estimated bandwidth to obtain the fair-share bandwidth (i.e. the measured bandwidth) by trial increment mechanism. Then collecting the measured bandwidth with the video buffer occupancy, we propose an adaptive video bitrates algorithm based on time needed for downloading the next segment to achieve smooth, stable video quality and avoid buffer underflow/overflow. From simulation results, our scheme is able to quick converge the measured bandwidth to the fair-share bandwidth even with increasing the number of competing clients, hence achieving fairness bandwidth utility. Besides, the video bitrate is allowed to be higher than the measured bandwidth, thereby achieving a higher bandwidth utilization efficiency and higher average video bit-rate compared with a conventional scheme.

References

[1] Cisco, V.N.I., 2016. Forecast and Methodology, 2015-2020. White Paper, Cisco.

[2] Pantos, R. and May, W., 2010. HTTP Live Streaming draft-pantos-http-live-streaming-05. Apple Inc., IETF draft, 23.

[3] Sodagar, I., 2011. The mpeg-dash standard for multimedia streaming over the internet. IEEE Multimedia, 18(4), pp.62-67.

[4] Kim, J., Caire, G. and Molisch, A.F., 2016. Quality-aware streaming and scheduling for device-to-device video delivery. IEEE/ACM Transactions on Networking, 24(4), pp.2319-2331.

[5] Akhshabi, S., Anantakrishnan, L., Begen, A.C. and Dovrolis, C., 2012, June. What happens when HTTP adaptive streaming players compete for bandwidth?. In Proceedings of the 22nd international workshop on Network and Operating System Support for Digital Audio and Video (pp. 9-14). ACM.

[6] Liu, C., Bouazizi, I. and Gabbouj, M., 2011, February. Rate adaptation for adaptive HTTP streaming. In Proceedings of the second annual ACM conference on Multimedia systems(pp. 169-174). ACM.

[7] Zhou, B., Wang, J., Zou, Z. and Wen, J., 2012, January. Bandwidth estimation and rate adaptation in HTTP streaming. In Computing, Networking and Communications (ICNC), 2012 International Conference on (pp. 734-738). IEEE.

[8] Lin, Q., Liu, Y., Shen, Y., Shen, H., Sang, L. and Yang, D., 2014, December. Bandwidth estimation of rate adaption algorithm in DASH. In Globecom Workshops (GC Wkshps), 2014 (pp. 243-247). IEEE.

[9] Huang, T.Y., Johari, R., McKeown, N., Trunnell, M. and Watson, M., 2015. A buffer-based approach to rate adaptation: Evidence from a large video streaming service. ACM SIGCOMM Computer Communication Review, 44(4), pp.187-198.

[10] Kim, S., Yun, D. and Chung, K., 2016, January. Video quality adaptation scheme for improving QoE in HTTP adaptive streaming. In Information Networking (ICOIN), 2016 International Conference on (pp. 201-205). IEEE.

[11] Jiang, J., Sekar, V. and Zhang, H., 2012, December. 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). ACM.

[12] Yin, X., Bartulović, M., Sekar, V. and Sinopoli, B., 2017, May. On the efficiency and fairness of multiplayer HTTP-based adaptive video streaming. In American Control Conference (ACC), 2017 (pp. 4236-4241). IEEE.

[13] Zhou, C., Lin, C.W. and Guo, Z., 2016. mDASH: A markov decision-based rate adaptation approach for dynamic HTTP streaming. IEEE Transactions on Multimedia, 18(4), pp.738-751.

[14] Ma, H., Hao, J. and Zimmermann, R., 2014, July. Access point centric scheduling for dash streaming in multirate 802.11 wireless network. In

Multimedia and Expo (ICME), 2014 IEEE International Conference on (pp. 1-6). IEEE.

[15] Zhang, S., Li, B. and Li, B., 2015, June. Presto: Towards fair and efficient HTTP adaptive streaming from multiple servers. In Communications (ICC), 2015 IEEE International Conference on (pp. 6849-6854). IEEE.

[16] Dubin, R., Dvir, A., Hadar, O., Shalala, R. and Ahark, O., 2015, January. Video complexity hybrid traffic shaping for HTTP Adaptive Streaming. In Consumer Communications and Networking Conference (CCNC), 2015 12th Annual IEEE (pp. 683-688). IEEE.

[17] Tian, G. and Liu, Y., 2012, December. 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). ACM.

[18] Miller, K., Quacchio, E., Gennari, G. and Wolisz, A., 2012, May. Adaptation algorithm for adaptive streaming over HTTP. In Packet Video Workshop (PV), 2012 19th International (pp. 173-178). IEEE.

[19] Dolinsky, K. and Celikovsky, S., 2012, June. Kalman filter under nonlinear system transformations. In American Control Conference (ACC), 2012 (pp. 4789-4794). IEEE.