ASTES

Advances in Science, Technology and Engineering Systems Journal
Vol. 6, No. 1, 184-199 (2021)
www.astesj.com
Special Issue on Multidisciplinary Sciences and Engineering

ASTES Journal
ISSN: 2415-6698

QoE-aware Bandwidth Allocation for Multiple Video Streaming Players over HTTP and SDN

Pham Hong Thinh[1,2], Tran Thi Thanh Huyen[3], Nguyen Ngoc Quang[1], Pham Ngoc Nam[4], Truong Cong Thang[2], Nguyen Viet Hung[5], Truong Thu Huong[*,1]

[1]*School of Electronics and Telecommunications, Hanoi University of Science and Technology, 100000, Vietnam*

[2]*Faculty of Engineering and Technology, Quy Nhon University, 820000, Vietnam*

[3]*The University of Aizu, Aizu-Wakamatsu, 965-0006, Japan*

[4]*College of Engineering & Computer Science, VinUniversity, Hanoi, 100000, Vietnam*

[5]*Faculty of Information Technology, East Asia University of Technology, Hanoi, 100000, Vietnam*

ARTICLE INFO

Article history:
Received: 30 October, 2020
Accepted: 06 January, 2021
Online: 15 January, 2021

Keywords:
HTTP Video streaming
Resource allocation
QoE
SDN
Traffic shaping
Adaptive Bitrate Streaming - ABR

ABSTRACT

For many years, the most popular technique for Internet video streaming is hypertext transfer protocol-based adaptive streaming, known as HAS (HTTP Adaptive Streaming). However, a seamless viewing experience can not be just simply guaranteed by HAS only. In the management network, the adaptation of HAS copes with a huge challenge since client-driven schemes lead to unfair share of available bandwidth when multiple players request adaptive bitrates (i.e bandwidth) through a bottleneck network link. Each client's requesting to maximize its needed bandwidth leads to the competition of network resources. This causes great QoE (Quality of Experience) reduction in terms of main metrics for each player: fairness, efficiency, and stability. In this paper, we propose an integration scheme of bitrate adaptation and Software Defined Networking-based resource allocation that can improve the QoE of competing clients. Our experiments show that the proposed scheme increases at least 20% up to 124% in terms of QoE scores compared with some existing methods as well as gains smoother viewing experience than the solutions of the traditional Internet.

1 Introduction

Recently we have observed a huge volume of media content, especially high-definition videos over the Internet. According to the report from [1], video traffic is estimated to occupy around 82% of the overall Internet traffic by 2021. Although the quality of the Internet has been improved significantly thanks to the advanced technologies such as 4G, 5G and Fiber to the Home (FTTH), current technologies are not yet ready to provide sufficient Internet bandwidth for new multimedia types such as 360-degree Videos and 12K Videos. Those new types of multimedia cause an enormous portion of Internet traffic.

The quality of video streaming over the Internet have been enhanced by several approaches such as: Microsoft's Smooth Streaming, Real-time Transmission Protocol, Adobe's Adaptive Streaming, Apple's HTTP Live Streaming, and MPEG-DASH. Especially, with MPEG-DASH, the vendor dependence on previously invented protocols were removed so as content providers can extend their services easily.

Based on this DASH standard emerged a number of solutions in which a source video is segmented into short segments of 2-10 seconds, each of which is encoded at a different bitrate resolution and level. During the playback, each client uses its bitrate adaptation logic to dynamically request and fetch desired encoded segments based on metrics such as average throughput and buffer occupancy. In the DASH technology, most of the current adaptive algorithms that decide the bitrate for the next downloaded segment are based on two parameters at the client side: throughput variation and buffer level. The buffer-based bitrate adaptation methods determine the bitrate mainly based on the characteristics of the buffer. These methods can also consider throughput, but the characteristics of the buffer are the preferred factors. Typically, buffers are divided into

[*]Corresponding Truong Thu Huong. Email: huong.truongthu@hust.edu.vn

multiple ranges, and for each of these ranges, different actions will be applied to determine the bitrate. Note that the specific buffer thresholds depend on the specific adaptation method. Typically, if the buffer level is high, the next segment will be selected with a higher bitrate than the current segment. If the buffer level is average, the selected bitrate will not be too different from the current bitrate. On the contrary, if the buffer level is low, the next segment will be selected with a low bitrate to avoid rebuffering and video freezes.

In addition, throughput-based methods are only based on estimated throughput to define a bitrate for the next segment. The main difference between these types of algorithms is the way for estimating throughput and how to utilize throughput. The simplest way to determine the throughput is based on the ratio of the given segment duration over the download time. The segment delivery time is determined from the time of sending a HTTP request to the time of receiving a HTTP response message that contains that segment.

On the other hand, Software-Defined Networking (SDN) [2] has emerged as a new networking paradigm in which a SDN Controller can make network management more flexibly by eliminating proprietary protocols from hardware vendors. With the network control plane implemented in software, rather than in firmware, network traffic can be managed more dynamically and at a more granular level. SDN switches/routers send and receive information to and from the SDN controller via southbound APIs; and the SDN controller relays information to the applications and business logic via northbound APIs. Within the scope of SDN, the OpenFlow Controller [3] is a type of SDN Controller that uses the OpenFlow protocol to configure SDN switches.

In any network exists always a popular issue called bottleneck link that could be congested when the amount of bandwidth demands go beyond the bottleneck link capacity. In case of bottleneck, video streams for each resource-sharing users could experience a remarkable decrease in perception of the video service, called Quality of Experience (QoE). Therefore, a method to solve the bottleneck problem in video streaming is neccesary.

To extend our previous work [4], in this work, we deploy a streaming architecture which make use of the MPEG-DASH technique and the SDN technology to solve the bottleneck problem. The scheme can increase the overall video quality for all users sharing the same bottleneck point. At the client's side, we develop a throughput-buffer based bitrate adaptation algorithm to utilize most of the available bandwidth in order to increase users' QoE. On the SDN controller, we establish a flow management mechanism which monitors the network to ensure fairness and avoid congestion when the network does not have sufficient bandwidth to serve new incoming stream requests. In the SDN controller, we develop the so-called Media Streaming Multiple Access (MSMA) module to monitor the network, detecting the number of ongoing streams, and deciding a fair share of bandwidth to all clients. Additionally, MSMA can also reallocate bandwidth or reject a new incoming request if no sufficient bandwidth is found; and thereby preventing congestion at the bottleneck point. The proposed adaptation and bandwidth allocation methods aim at the three following goals:

• $Efficiency$: all players joining the system can choose for themselves a level of video quality as high as possible.

• $Fairness$: When there are multiple clients accessing the bottleneck link at the same time, all of the available network resources must be allocated as fairly as possible.

• $Stability$: Each player should avoid video stalling, unnecessary quality switches, which will adversely affect users' QoE.

The proposed solution not only allocates bandwidth fairly and adaptably on the network side, but also adaptive bitrate at the client side to improve video streaming quality. When there are many clients streaming on bottleneck link, the problem of sharing bandwidth for optimization is very important. There are three main approach groups: dividing bandwidth equally, dividing it by groups or choosing which user should be the one to share bandwidth with new coming users or new request of an existing user. Each solution has its own advantages and disadvantages. So, our proposed technique's limitation is to trade off a small QoE degradation of one or a small number of users sharing the bandwidth. However, the advantage of the proposal is to maximize the number of access clients while keeping QoE in a good shape.

The rest of this paper is organized as follows. Section 2 presents related work of typical bitrate adaptation algorithms and bandwidth allocation schemes. Section 3 provides information on throughput estimation and the proposed adaptation algorithm. Our proposed SDN-based dynamic resource allocation is described in Section 4 . A performance evaluation is presented in Section 5. Finally, conclusions and future directions are discussed in Section 6.

2 Related Work

HTTP Adaptive Streaming (HAS) has become the key solution for video streaming. There are a number of recent researches conducted to improve users' QoE in the context of multiple clients requesting for DASH service. As discussed in [5] and [6], the authors proposed a so-called Aggressive method in which a rate adaptation algorithm based on the estimated throughput is implemented where the last segment throughput is simply used as the estimated throughput. The quality level will increase or decrease depending entirely on estimated bandwidth. The Aggressive method is sensitive to cases where the bandwidth increases sharply in a short period of time and then abruptly decreases (short-term bitrate peaks). The client then requires a high-quality version, but the high bandwidth level does not last long enough to load the required segment. This may cause video freezing during playback. The Aggressive algorithm aims to use most of the available bandwidth for the streaming client by always adapting to the highest quality version of video segment that does not surpass the available bandwidth. By doing this, the expectation of streaming users is also met. In [7], [8] the authors proposed a buffer-based bitrate adaptation algorithm for HTTP video streaming, called BBA. BBA is based on the current playback buffer occupancy only by selecting bitrate heuristically without throughput estimation. The goal of BBA is avoiding stalling events and maximizing the average video quality by choosing the highest available bitrate level. However, this scheme shows many limitations such as QoE degradation in case of long-term bandwidth variations and slow bandwidth fluctuation, especially in case of bandwidth competition in a shared network. In the literature, there are some ABR (Adaptive Bitrate) algorithms that take into account both of the estimated throughput and current buffer occupancy to define the most suitable video version

for the next segment. In [9], [10], the authors propose a buffer-based bitrate adaptation algorithm for VBR (variable bitrate) videos. The study in [11] proposes a throughput-buffer based algorithm relying on the buffer target, the playback buffer filling level, and the current bandwidth to determine the next segment's version. Work [12] proposes a segment-aware rate adaptation algorithm (SARA), in which segment size variation, the estimated throughput and the current buffer occupancy are considered to predict precisely the time required to download the next segment. In SARA, the MPD file not only contains the average bit rate information of the segment but also the size of the segment. SARA has expanded and changed the structure of the MPD file to accomplish this. Estimated throughput is based on the measured throughputs in the past and divides the buffer into different levels to determine the bitrate. This assures to download the best possible representation of the video while avoiding video buffer starvation. However, SARA shows low performance and poor QoE in the context of bandwidth sharing competition.

In a shared network environment, with so many Internet services available, bandwidth resource always gets risk of becoming insufficient to the increasing number of users, despite the maturity of cutting edge technologies nowadays. Therefore, it is always a huge concern for researchers to deal with limited resource allocation in any networking circumstances. Most of the existing researches tend to increase the Quality of Experience of clients while optimizing the transmission capability. Now exist three solution categories to deal with the problem where many clients compete through a bottleneck point. The first category is to solve the issue of sharing video streaming services at the client side as mentioned in [13]–[15]. In [13], an algorithm is proposed to improve efficiency, fairness, and stability named FESTIVE by providing schemes to ensure a trade-off of these factors. FESTIVE is the first known client-side adaptive bitrate algorithm that is designed for the multiple client context specifically. FESTIVE contains three main components: a scheduler that determines the time of starting to download a segment, an algorithm that estimates the throughput based on all previous downloaded segments and a mechanism to select the bitrate of the next segment. However, with FESTIVE, when the number of users grows sharply in a competition, the system's stability could be reduced remarkably since the estimation of the bandwidth is too high, perhaps. In [14], a rate adaptation algorithm for HAS service is proposed based on a "probe-and-adapt" principle at the client side so-called PANDA. "Probe-and-adapt" is a kind of the additive increase and multiplicative decrease principle (AIMD) that is used in TCP. However Probe-and-adapt operates at the application layer and in a longer time scale. PANDA is a HAS client rate adaptation algorithm designed to provide high stability and fast responsiveness to bandwidth variations in the context of multiple HAS clients sharing the same bottleneck links. For detecting the available bandwidth, the network is probed by PANDA which additively increments its sending rate at each adaptation step and multiplicatively decreases its rate if a congestion is detected, in order to adapt its video bitrate accordingly. During the estimation of the available bandwidth, the PANDA probing mechanism tries to increase its sending rate, instead of transmitting auxiliary piggybacking traffic. PANDA also allows HAS clients to respond to bandwidth fluctuation quickly and to achieve stability. But PANDA does not consider the resource allocation strategy in the context of multiple user negotiation. In [15], the streaming framework in VHS (VideoHomeShaper) is designed and implemented to evaluate performance in home's last access hop. In this study, the evaluation of QoE for competing clients takes into account some metrics such as Stability, utilization, and fairness. Overall, these researches mainly focus on adapting bit rate at the client side to improve the parameters that can affect user's QoE.

The second category focuses on the sever side. In this area, servers are normally considered as the only control point responsible for managing video transmission for all clients. The authors of [16] introduce a server-side adaptation scheme, which uses TCP to limit the video bit rate. A server-based traffic shaping solution is proposed in [17] to notably decrease such oscillations, at the expense of a small loss in bandwidth utilization. However, we can only be observe the actual bandwidth congestion problem at the network's side. Consequently, not too many clients might be able to handled by the server.

Finally, the category of in-network methods uses active bandwidth allocation to gain QoE fairness for multiple clients. Fair bandwidth allocation for players at a bottleneck link has been recently suggested by several work. Based on the SDN technology, [18] proposes traffic shaping techniques and do an analysis of ABR video streams when sharing resources through a congested link. In this method, all shared connections at the bottleneck point have the same throughput; and the total bandwidths assigned to requiring clients must not exceed the available bandwidth at the bottleneck. SDN-based traffic shaping in [18] can improve QoE for video streaming in terms of efficiency, fairness, and quality. The authors in [19] also consider the bandwidth sharing within a bottleneck link. However, in the network sharing model, the network sharing is calculated periodically, not in the per-request basis. In adaptive streaming, clients can require changing the bit rate at any time and the SDN controller must calculate the network sharing based on all of requests at the certain time. Authors in [20] proposed a system for increasing QoE of SVC-DASH clients based on SDN. The SDN architecture enables to select different streaming path to transfer different video layer flows between the server and the clients. But this paper does not consider the resource allocation strategy in the context of multiple user negotiation. In [21], the author proposed an OpenFlow-based QoE fairness framework (QFF). QFF manages the player statuses and allocates network resources dynamically to achieve the maximum user-level fairness for all clients that compete in a shared network. A proof-of-concept implementation is provided considering a small number of concurrent flows. However, only the fairness parameter of QoE is considered in this research. In [22], network-assisted strategies are considered to provide service differentiation to concurrent video flows such as the Bitrate Guidance approach (BG), the Bandwidth Reservation approach (BR), and the hybrid approach (BG+BR). A centralized Video Control Plane (VCP) is built to provide Video Quality Fairness (VQF) to concurrent video sessions that are sharing a common bottleneck. Work [23] argue that the network and clients shall collaborate to gain QoE fairness while video streams are encrypted. Placing cDVD in the greater context of QoE delivery methods is instructive. The resource-sharing clients only focus on fairness but not efficiency and stability. In [24], centralized and distributed architectures are proposed for collaboration between video service provider (VSP), network service provider

(NSP), and DASH clients to provide SDN-based NSP-managed or VSP-managed DASH services with quality-of-service (QoS) reserved network slices. The SDN controller computes the shortest-path route and bitrate for each client. This approach is to gain QoE fairness among heterogeneous DASH clients. However, again the resource-sharing clients only focus on fairness but not efficiency and stability. In [25], the author proposed an optimization model aiming at fairly sharing the bottleneck link while maximizing the client's perceived video quality. However, the work focuses only in bandwidth allocation and maximal link capacity of each client without considering how bit rate should be adapted to receive desired quality experience of the clients based on the real condition of the clients. Research [26] proposes a SDN-assisted adaptive streaming solution for tile-based contents using DASH to improve the user's quality of experience. The paper also summarizes the difference in immersive content streaming between SDN-based and traditional networks and introduce a SDN-based framework to support tile-based immersive content streaming. However, this paper does not consider the bandwidth allocation and bitrate adaptation problem. In [27] and [28], the author proposed a SDN based management architecture and dynamic resource allocation for HAS systems aiming to alleviate scalability and improve the QoE of each client. This architecture is in charge of allocating the network resources dynamically for each client depending on the client's QoE expectation. The same authors in [29] propose a SDN-based QoE-aware bandwidth broker for HTTP adaptive streams in an hybrid fiber coax (HFC) network, so-called Bandwidth Management Solution - BMS that dynamically chooses the respective bandwidth allocation decisions and the optimal joint representation in oder to meet the per-session and per-group QoE objectives. In [30], there is a network-driven video streaming architecture is built to design robust mechanisms for multiple players so as to achieve end-user QoE fairness. The SDN controller observes the environment state, computing the reward, and updating Q-value and taking the bandwidth allocation action by forwarding flow tables. However, the author considered how bandwidth should be shared to achieve fairness while QoE is taken into account from the perspective of receiving bit rate that is suitable to the link bandwidth. The work does not consider how QoE of each client can be acquired based on the client's real need in terms of bit rate at a certain moment QoE. In [31], a SDN-based model is built over the DASH protocol to improve the QoE whilst taking into account the variety of video parameters, devices and the network requirements. The proposed scheme comprises two phases: Machine Learning-based estimation phase of available resources, and adaptation and selection phase based on the results of the first phase. The goal of the approach is to have the best quality perceived video. The authors in [32] proposed a SDN-assisted solution to use the network management plan to reinforce policies for improving QoE of clients. However, again, this work does not consider the case of lack of network resource so multiple users need to negotiate for the network share in trade off with QoE. In [33], the authors present SAND/3 that is an SDN-Assisted QoE control for DASH over HTTP/3 for improving the QoE at the client side based on criteria such as buffer size, display size, type of device and subscription plan, and available memory. From the SDN network perspective, SDN tries to find a route for each requested connection from each client based on available resource. But this paper does not consider the resource allocation strategy in the context of multiple user negotiation.

Generally, SDN-based dynamic bandwidth allocation shows a good performance in improving video quality. However, the interaction between the network control and the client-side control in the context of multiple users sharing one bottleneck link has not been well investigated. Therefore, deploying solutions in a large-scale system is still a pending question.

3 Bitrate Adaptation Algorithm

Our proposed framework to the bottleneck problem is a integration of the bitrate adaptation algorithm at the client's side and the centralized SDN-based resource allocation, i.e. a cooperation between the network and video streaming applications. In the streaming network, a flow management mechanism called MSMA (Media Streaming Multiple Access) is proposed (discussed in Section 4.2).

In fact recently, there have been two widely-adopted types of algorithms at a MPEG-DASH Client, which belong to either the throughput-based group or buffer-based group. Therefore, we consider the context of multiple video-on-demand players, provided that the throughput-based adaptation algorithm called Aggressive [6] is deployed at each client.

Besides, we have also proposed a buffer based bitrate adaptation algorithm to determine a video quality level for the next segment. Compared to the throughput-based algorithms, buffer-based algorithms provide smoother bitrate curves in video-on-demand streaming. In the next subsections, the description of the throughput estimation and the proposed adaptation algorithm are presented.

Table 1: Symbols use in this paper

Symbol	Description
B_i	Current buffer level at segment i
B_e^{i+1}	Estimated buffer for segment $i+1$
B_{max}	Maximum buffer size
N	Number of clients
I_{i+1}	The chosen version for the next segment
T_i	The throughput for the current segment
T_{i+1}^e	Estimated throughput for the next segment
r_i	Bitrate of client i
$r_i^{'}$	Bitrate of r_i after the SDN controller is recalculated
$R_{0,\ldots,V-1}$	Bitrate of version 0, 1, ..., V-1
R_{max}	Bitrate corresponds to the highest level of quality
RTT	Round Trip Time
SD	Segment Duration
V	Total number of video versions

3.1 Throughput Estimation

In simple terms, the bitrate adaptation mechanism adapts its requested bitrate based on the estimated throughput. Video segments are delivered from the server to the clients in sequence of consecutive HTTP request-response transactions. In general, throughput is obtained by dividing the data size by the delivery time. Knowing the actual amount of data delivered by a request-response transaction is

important. The segment throughput T_i^e for each delivered segment i is computed using the request-response duration (i.e. from the instant of sending the request to the instant of receiving the last byte of the response). Table 1 lists the symbols used in this paper.

A video version is chosen based on Equation (1). Bitrate R_j is selected as the maximum value not exceeding the product of the current estimated throughput T_{i+1}^e and safety margin μ (μ varies from 0.0 to 0.5). μ is defined to solve the instability of the Internet connection and delay caused by other streaming processes such as bitrate adaptation calculation and file decoding.

$$R_j \leq (1-\mu) \times T_{i+1}^e \tag{1}$$

Next, we propose to estimate the next throughput based on two previous measured throughputs at the client side (as described in Equation (2)). In the Equation, γ is a dynamic constant range from [0, 1] that allows a client to adapt well with highly bandwidth fluctuations, especially with highly variable bandwidth since it is the main cause of video freezing. γ is usually set to be 0.5. When γ = 1.0, the estimated result will be exactly the same as the one of the aforementioned aggressive method.

$$T_{i+1}^e = \gamma \times T_i + (1-\gamma) \times T_{i-1} \tag{2}$$

Algorithm 1: Proposed Bitrate Adaptation Algorithm

```
Input: T_i, R_n, B_i, RTT, SD, I_i, Δ_B ;
Output: I_{i+1} ;
T^e_{i+1} ← γ × T_i + (1 − γ) × T_{i−1};
I_{i+1} = 0;
for j ← V − 1 to 0 do
    B^e_{i+1} ← B_i + SD − RTT − SD×R_i / T^e_{i+1};
    if j > I_i and B^e_{i+1} ≥ B_i + Δ_B then
        I_{i+1} = j;
        Break;
    end
    if j ≤ I_i and B^e_{i+1} ≥ B_i − Δ_B then
        New request to controller;
        I_{i+1} = j;
        Break;
    end
end
```

3.2 Proposed adaptation algorithm

In this paper, we design a new quality-selecting scheme based on the estimate buffer level so as to avoid video freezing. According to our previous work [34], we use Equation (3) to estimate the next buffer level, provided the client selects version j for segment i where SD is segment duration. RTT is the duration defined from the instant a client sends its request to the instant it receives a response and calculated by timestamps of the westward and eastward packets captured at the network card of the SDN controller. The westward packet is the request packet sent from a client to the controller and the eastward packet is the packet carrying information of calculated BW from the controller to that client.

The main goal of our solution is to choose a suitable quality level that can keep the buffer greater than the current buffer B_i to prevent stalling events. Our method details are shown in Algorithm 1. This algorithm is based on an estimated buffer to determine a new video version for the next segment.

In fact, increasing the quality version has a lower priority than keeping the quality version stable, and the quality should be reduced as little as possible through threshold Δ_B. The larger Δ_B , the less varied the video quality version, but the worse the connection adaptation. On the contrary, the smaller Δ_B, the better the connection adaptation, but more the stability is lacked. This parameter usually ranges from 0.25s to 2s. In this paper, we choose $\Delta_B = 1s$, since this figure is appropriate to trade off the issues above. When there is network fluctuation, a client shall reduce its quality version accordingly. In that case, the client will request the SDN controller to recalculate the bitrate, and sending this bitrate information to all clients that are having ongoing connections so that the quality version of all clients are kept as stably as possible.

$$B_{i+1}^e = B_i + SD - RTT - \frac{SD \times R_j}{T_{i+1}^e} \tag{3}$$

4 Proposed SDN-based Solution

4.1 Bandwidth Allocation Policy

Provided that N clients go through the same bottleneck with bandwidth capacity C. $BW_{i,t}$ is the amount of bandwidth allocated for client i at time t. The bandwidth allocation policy is to map the bitrate requested by each client to the actual bandwidth that could be granted to that client:

$$BW_{i,t} = f(r_{i,t}) \tag{4}$$

where $r_{i,t}$ is the bitrate that client i requests at time t. r_i is the bit rate the clients desire to have at the beginning. But BW_i is the actual bit rate that the clients will adapt to after being reconsidered by the SDN controller. In case the bandwidth resource is sufficient, all clients will have BW_i which are the same to r_i. In case network resource lacks, one or some r_i may be different from BW_i.

If the network resource (bottleneck link capacity) is sufficient for all requested bandwidths of each client, the clients will be allocated the same bandwidth as they request, i.e.:

$$BW_{i,t} = \frac{C}{N}, \forall i \in N, \forall t \tag{5}$$

However, in reality, HTTP networks are not ideal because clients frequently require bitrates that can be higher than the then-actually allocated bandwidth. As described in [35], [36], the bandwidth allocation policy for this non-ideal HTTP network condition is as follows:

. if $r_i = r_j$ then $f_i(r) = f_j(r)$;
. if $r_i > r_j$ then $f_i(r) > f_j(r)$;
. $\frac{\partial f_i(r)}{\partial r_i} > 0$, $\frac{\partial f_i(r)}{\partial r_j} < 0$, where $i \neq j$;
. $lim_{r_i \to \infty} f_i(r) < C, lim_{r_i \to 0} f_i(r) > 0$;
. $f(.)$ is a function of r and value of f is independent of the client order.

In fact, requests do not come to the edge switch at the same time, therefore naturally, a solution to each request is thought appropriate.

When a request comes, the system calcualte to determine if it should accept the request, so as to ensure no congestion occured at the bottleneck point, causing the drop of quality of experience of all N clients.

In DASH, content provider encode videos at different quantity levels of different bit rates and then brake the videos into segments. That is the base to determine the minimum bandwidth needed by each client in order to obtain an uninterrupted video streaming. That minimum bandwidth is equal to the average bitrate of the video with the lowest quality level. This value is denoted as R_0.

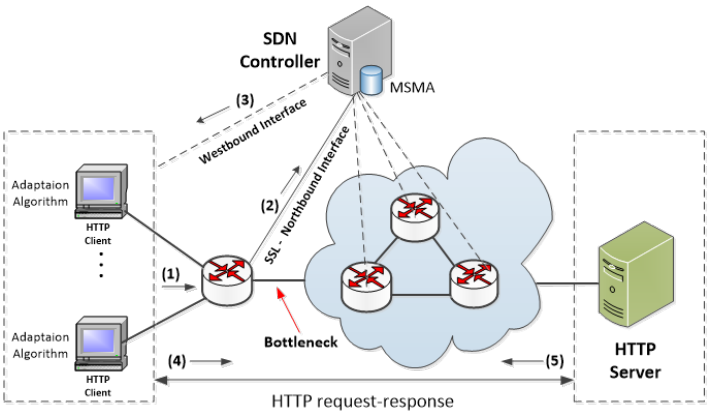


Figure 1: Proposed architecture and operation for the bottleneck problem

4.2 Proposed fair bandwidth allocation solution (MSMA-0)

The proposed architecture is illustrated in Figure 1. As it can be seen, the Aggressive adaptation algorithm and the MSMA module are implemented in each client and the SDN controller respectively. The whole solution is called MSMA-0. In our proposed architecture, the edge switch (OFS - OpenFlow Switch) communicates to the SDN controller through 2 interfaces: a standard Northbound interface of Openflow that connects OFS and the SDN controller via TSL protocol; and a legacy network interface that connect the OFS and controller as 2 legacy network devices called Westbound interface. The operation of the proposed architecture is as follows:

• Phase 1: Negotiation of appropriate bit rates requested by each client

(1) Each client calculates its desired bit rates according to the bit rate adaptation algorithm implemented at client.

(2) Client sends that streaming request (i.e. desired bit rate) to the edge OFS switch. Then, the OFS encodes the request information in a *Packet − In* message and sending it to the SDN controller via the Northbound interface (which is defined in OpenFlow protocol).

(3) The Controller, via the Northbound interface with the OFS edge switch, having already knowledge of link capacity of the network cards of the switch can calculate new BW values, then sending these new information to the clients via Westbound interface (as a legacy network device).

• Phase 2: Requesting the finally negotiated bit rate to the application server

After receiving information on BW from the SDN controller, the clients operate as follows:

(4) N ongoing clients send new requested BW_i to the application server via an open connection.The new client $(N+1)$ sends a request to connect to the application server, the SDN controller will order the edge OFS to install a new flow entry to establish the connection between Client N+1 and the application server.

(5) Upon receiving the request on bit rate (i.e BW_{N+1}), the app server will push a video quality version segments accordingly to all streaming clients.

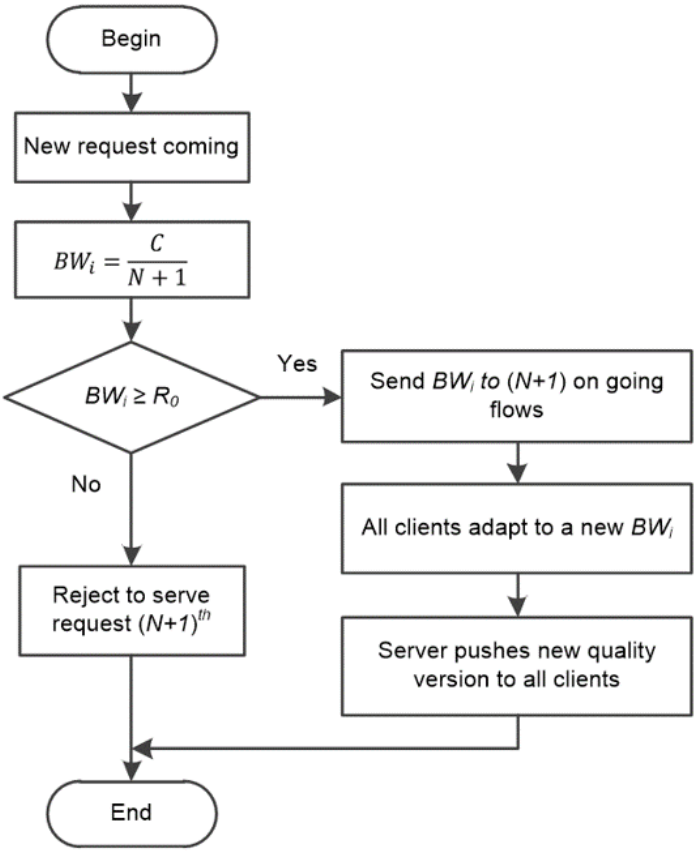


Figure 2: MSMA Flowchart

As the implication of its name, a network bottleneck causes a slow connection speed, limiting the experience of media streaming. Solving the aforementioned issue is the goal of our proposed MSMA algorithm. Figure 2 depicts the flowchart of MSMA deployment with the working principle described as follows:

1) Given that there are N ongoing clients requesting new quality versions for each segment, the average bandwidth BW_i distributed to $(N + 1)$ streams is recalculated by the MSMA module, according to the Equation (6) below:

$$BW_i = \frac{C}{N+1} \tag{6}$$

2) BW_i is then compared with the minimum acceptable bitrate R_0 by the SDN controller that then decides what action to take. Either one of these two actions depending on the result of the comparison will be carried out by the SDN controller.

3) A flow for the incoming stream is added by the SDN controller and BW_i is sent to all on going streams by the controller if $BW_i \geq R_0$ i.e. the bandwidth allocated to each client is higher than the one of the lowest quality version and is sufficient to stream that video version.

4) The incoming request is rejected to serve by the controller if $BW_i < R_0$ i.e. the bandwidth allocated to each client is insufficient to stream even the lowest quality version of the video.

Then after all, all streaming clients shall adapt to BW_i decided by the SDN controller. The clients then request the DASH server for the decided bitrate. The adaptation logic is described in the next section. TheDASH server pushes a new quality version to all streaming clients.

4.3 Proposed adaptive bandwidth allocation solution (MSMA-1)

In this work, we propose a new method (called MSMA-1) that combines a bit rate adaptation algorithm at the client side (i.e. Algorithm 1), with modul MSMA deployed in the SDN controller as Algorithm 2. The goal of MSMA-1 algorithm is trying to improve the Quality of Experience of each client on the streaming service by providing the bit rate that a client requires while trying to increase the number of clients that can access to the system within a limited resource.

At the network side, we allocate bandwidth dynamically (i.e. Algorithm 2) for all clients in order to maximize the number of clients that can be given an access to the network and minimize the number of clients who are requested to change their bitrate expectations (bitrate to satisfy the QoE criterion). Provided the system has N clients that have HTTP streaming over a bottleneck point. The capacity of the output link of the bottleneck point is assumed C (Mbps).

At the client side, each client deploys the bitrate adaptation algorithms proposed in Algorithm 1 to find out what is their bitrate expectation in order to achieve desired QoE. Actually, each bitrate is corresponding to a quality version defined by DASH and stored at the server.

In case, the sum of all requested bitrates r_i of N clients (i = 1 to N) is smaller or equal to capacity C, the SDN controller will request the DASH server to push the quality version desired by each client.

In case, the sum of all bitrates r_i (i = 1 to N) is larger than C, i.e:

$$\sum_{i=1}^{N} r_i > C \tag{7}$$

then the SDN controller recalculates bandwidth allocated for each client. The SDN controller operates in the following criteria:

1. The SDN controller tries to keep the bitrates r_i required by the clients as stably as possible with the goal of meeting these clients' QoE requirement.

2. The SDN controller selects a client holding the highest rank, and taking a fraction of its required bandwidth to give to another client. In this manner, the network take a fraction of the network resource of this client but in return it will give an access for an additional client (i.e. client (N + 1)). So, the goal of serving as many clients as possible is fulfilled.

In fact, as in the solution described in section 4.1, the link capacity C is divided equally to all (N + 1) clients, therefore, all (N + 1) clients suffer an undesirable QoE degradation since distributed resource may be lower than their requirements to achieve each client's satisfaction. Therefore, in this new proposed solution, the SDN controller, firstly, ranks all clients in a certain manner. Then the controller will set a policy to allocate resource to all clients by rank. Technically, this ranking function (or objective function) can be defined as a function of multiple objectives. Within the scope of this paper, we define the rank of client by Equation (8). Looking at the Equation (8) shows that the client that have a higher buffer level and a bitrate is likely to have the highest rank function. Therefore, this client will bring in some of the resources available for the new service. In addition, the choice of which client to share the resource will depend on for the two parameters which are bit rate r_i and buffer level B_i such that rank function is highest.

$$Rank(i) = f(B, r) = \alpha \times \frac{B_i}{B_{max}} + \beta \times \frac{r_i}{R_{max}} \tag{8}$$

where α, β: non-negative weighting factors. And r_i is bitrate required by the ongoing client, B_i is current buffer level of client.

The set of $Rank(i)$ is arranged in the decreasing order according to the rank function value.

Details of the proposed solution are described in Algorithm 2. The details of this algorithm are explained as follows:

Firstly, the requested bitrate of a client who has the highest rank is recalculated to share the resource to other client. Provided the client with the highest rank is Client N. Then the bandwidths of client j and client N are reallocated as Equation (9) and (10) accordingly:

$$r'_j = \frac{r_j}{r_j + r_N} \times \Delta C \tag{9}$$

$$r'_N = \frac{r_N}{r_j + r_N} \times \Delta C \tag{10}$$

where r_N is the requested bitrate of client N who has the highest rank value and r_j is the requested bitrate of client who requests a new bitrate. r'_j and r'_N are the varied bitrates of r_j and r_N after the SDN controller recalculate bandwidth distribution. ΔC is the remaining bandwidth after reserving the bandwidth portion for all left clients that have not been shared it's resource. (i.e. all clients that retain their bitrates r_i). ΔC is defined by Equation (11).

$$\Delta C = C - \sum_{i=m}^{N} r_i \tag{11}$$

where m is the number of clients who are sacrificed.

In case if the calculated bitrate of client j is smaller than the smallest acceptable quality version ($r_j < R_0$), continue to take the second highest bitrate requested by an ongoing client and redivide

proportionally for all 3 clients. The process continues until all N clients satisfy a bitrate $r_i \geq R_0$ ($i = 1 \div N$).

Algorithm 2: . MSMA-1 algorithm

Input: r, C, Δ_C
Output: r
for $i \leftarrow$ *to* N **do**
  $r'_{N+1} = \frac{r_{N+1}}{\sum_{k=1}^{i} r_k + r_{N+1}} \times \Delta_C$;
  for $j \leftarrow 1$ *to* i **do**
    $r'_j = \frac{r_j}{\sum_{k=1}^{i} r_k + r_{N+1}} \times \Delta_C$;
  end
  if $min(r') \geq R_0$ **then**
    $r = r'$;
    Break ;
  end
  else if $i = N$ *and* $min(r') < R_0$ **then**
    Reject client $(N+1)^{th}$;
  end
end

5 Performance Evaluation

5.1 Experiment Setup

Bitmovin, a multimedia technology company providing services, which transcode digital video and audio to streaming formats using cloud computing, and streaming media players has developed and maintained a library named libdash [37]. In this paper, QtSamplePlayer is used as the media player of the streaming architecture. QtSamplePlayer is a Qt-based player recommended for the libdash library by Bitmovin. Videos pushed from the HTTP server to the clients are 2 types of videos: constant bit rate CBR and variable bit rate VBR. Besides, traffic from the clients to other devices (i.e. OFS switch and the SDN controller) is only control information on bit rate requests to adapt to required quality version.

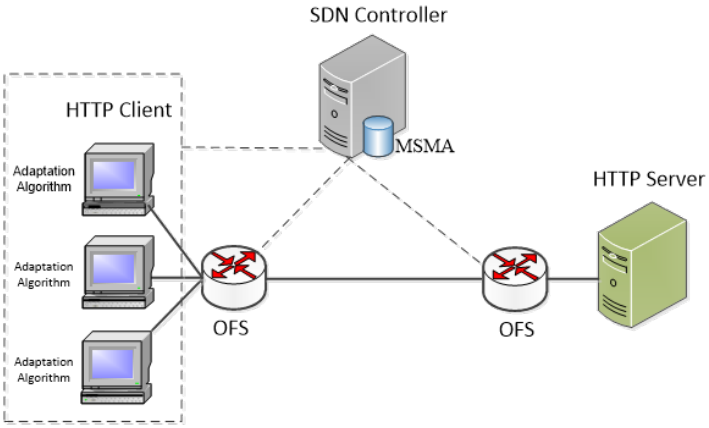


Figure 3: Network topology used for experiment.

Figure 3 shows the network setup for performance evaluation. In our experiment, five machines are used to conduct the experiment: the media server, one machine for emulation of the streaming network including the SDN switches and the SDN controller - Floodlight [38], and 3 video streaming clients. One clients is implemented on a physical PC and all other machines are VMWare machines. Due to limited testbed resource, we just make a proof of concept with 3 clients. In the future, we will improve the testbed to scale up the experimental scenario. The network consists of two OpenFlow switches and a bottleneck link between them. An OFS is connected to the three client machines and the other OFS is connected to the media server ,in which MPD (Media Presentation Description) and video files are stored. The MPD file is an XML (Extensible Markup Language) file that represents the different quality levels of the media content and the individual segments of each quality level with Uniform Resource Locators (URLs). For this evaluation, Mininet [39] is used to create a realistic virtual network.

There are many different types of videos that can be used in evaluation simulations, but all of them focus on two types of video: variable bit variable (VBR) and constant bitrate video (CBR). In this paper, the experiment videos are two short movies named Elephants Dream [40] and Destiny [41] to represent the 2 aforementioned video types. In the Elephants Dream movie, the video is divided into 327 segments with the total duration of 654 seconds (i.e. 2 seconds per segment). This video is encoded in the VBR mode of 12 different versions corresponding to 12 different average bitrates. The version index, Quantization Parameter (QP), and the average bitrate of each version are listed in Table 2. The Destiny short file is a constant bitrate video (CBR) and is divided into 340 2-seconds-long segments of 16 quality levels of different bitrate (50, 100, 250, 400, 600, 800, 1000, 1200, 1600, 2000, 3000, 4000, 6000, 8000, 12000, 16000 Kbps).

Table 2: Version information of the VBR video

Version	QP	Average Bitrate (Kbps)
0	46	354
1	43	472
2	40	638
3	37	882
4	34	1234
5	31	1779
6	28	2588
7	25	3823
8	22	5613
9	19	8028
10	16	11156
11	13	15227

5.2 Experimental Scenarios

Six experiments are deployed to evaluate our proposed system:

• Experiment 1: The link capacity of the bottleneck is 10 Mbps, 7 Mbps and 4 Mbps. The controller uses the aforementioned method MSMA-0. Client 1 starts streaming 110 seconds (55 segments). After the first 110 seconds of Client 1, Client 2 starts streaming in 110 seconds and in the same way with Client 3.

• Experiment 2: The link capacity of the bottleneck is set at 10 Mbps, 7 Mbps and 4 Mbps. The controller uses the aforementioned MSMA-1 algorithm. Each of Client 1, Client 2 and Client 3 starts

streaming in 110 seconds, one after another.

• Experiment 3: The link capacity of the bottleneck is set at 10 Mbps, 7 Mbps and 4 Mbps. The Aggressive, SARA, BBA, PANDA algorithms are used. MSMA is not available. Each of Client 1, Client 2, and Client 3 starts streaming in 110 seconds, one after another.

• Experiment 4: The link capacity of the bottleneck is set at 10 Mbps, 7 Mbps and 4 Mbps. The SARA, BBA algorithms are used. The MSMA module is activated. Each of Client 1, Client 2, and Client 3 starts streaming in 110 seconds, one after another. This experiment simulates methods "SDN-based SARA" and "SDN-based BBA".

• Experiment 5: The link capacity of the bottleneck is set at 500 Kbps. The controller uses the MSMA modul. The clients request for the VBR video an independent time.

• Experiment 6: The link capacity of the bottleneck is 500 Kbps. The controller does not use the MSMA modul. The clients request for the VBR video at independent time.

We use the Aggressive method as the bitrate adaptation algorithm [6] with a safety margin $\mu = 0.2$.

We perform each of the five experiments in three times and record the average results for later comparison. The purpose of Experiment 1, Experiment 2, and Experiment 3 is to determine stability, efficiency, and fairness; The purpose of Experiment 4 and Experiment 5 is for checking the possibility of rejecting new requests when a congestion is possible.

With Experiment 5, since $R_0 = 354Kbps$, only the connection request of the first client will be accepted by the controller; and video with the minimum average bitrate will be streamed by the client. The evaluation results illustrates that the number of video freezes is 5 in case the segments contain a high bitrate compared to the representative bitrate (i.e. average bitrate). Our proposed system does not accept to provide services to requests from client 2 and client 3. Experiment 6 shows the scenario of a traditional network where the controller does not operate with the MSMA modul. In that scenario, the 3 requests from 3 clients are accepted in turn.

So, because the available bandwidth is not sufficient for all 3 clients sharing bandwidth and asking an acceptable bandwidth, all 3 clients are frozen immediately after being connected and also experience buffering most of the time during the whole course of their streaming section. This proves that at the circumstance of poor bandwidth, MSMA-0 is still able to serve the modest number of players, rather than the other methods where no user will be served.

5.3 Simulation Results and Discussion

To evaluate the performance of the proposed solution, we use three metrics of fairness, efficiency and stability as presented in [42] and [13] and a QoE metric. In the following subsections, these metrics will be described in detail.

5.3.1 Fairness, efficiency and stability metrics

In this subsection, the 3 metrics of stability, fairness, and efficiency are presented. In particular, the fairness score is measured by Equation (12)

$$Fairness = \frac{|\sum_{i=1}^{N} r_{i,t}|^2}{N\sum_{i=1}^{N} {r_{i,t}}^2} \tag{12}$$

where $r_{i,t}$ is the video bitrate allocated to client i at time t. The fairness score should be in the range of [0 to 1]. As the matter of fact, the closer to 1 the fairness score is, the fairer the system is.

The efficiency score of the bandwidth distribution is computed using Equation (13), where W is the available bandwidth of the shared connection. The closer to 1 the Efficiency score is, the better the system efficiency is.

$$Efficiency = \frac{|\sum_i r_{i,t}|}{W} \tag{13}$$

Equation (14) defines the stability score for client i at time t. This score is calculated based on the absolute amplitudes of the quality fluctuations (i.e. bitrate switches).

$$Stability = 1 - \frac{\sum_{d=0}^{k-1} |r_{i,t-d} - r_{i,t-d-1}| \times \omega(d)}{\sum_{d=1}^{k} r_{i,t-d} \times \omega(d)} \tag{14}$$

where: $\omega(d) = k - d$: weight function; and k: 20 seconds (10 segments) in this experiment. The closer to 1 the Stability score is, the more stable the system will be.

5.3.2 QoE Metric of Video Streaming

In this section, we first present parameters that have influences on the QoE of video streaming. Then, a QoE metric taking into account the parameters is provided. According to [43], the most important parameters are as follows:

Initial Delay (D_t): Initial Delay is computed as the duration from the point the first segment of a video is requested until the point the video is played out. For an effective video streaming system, the buffer time required for this parameter must be much smaller than the maximum client-side buffer level. According to [27], the initial delay should be less than 6 seconds.

$$D_t \ll B_{max} \tag{15}$$

Average Version Quality (AVQ): In the DASH system, the quality of the segments received at the client is variable. This parameter is defined as the average version of all the downloaded segments. Assume a video stream consists of K segments and each segment is encoded with V bit rate corresponding to V quality level, where v represents a specific quality level .

$$AVQ = f(i, v) = \frac{1}{K}\sum_{i=1}^{K} q_{i,v} \tag{16}$$

where $q_{i,v}$ is the quality value of the i^{th} segment at the v quality session level, for example $q_{i,v_{max}}$ is the quality value of the i^{th} segment with the highest quality level.

Number of Version Switch-downs (NVS): is the number of times that a downloaded segment has a lower bitrate than the previous segment. We can use this paremeter along with Average Video Quality to provide quantitative inferences of QoE. If video flows have similar Average Video Quality, a flow with the lower number

of Version Switch-downs will be perceived,by users, to be the better one .

$$NVS = \frac{1}{K-1}\sum_{i=1}^{k-1}|q_{i+1,v_{i+1}} - q_{i,v_i}| \quad (17)$$

Number of Video Stalling (S_t): Video streaming will be stalled if the buffer at the client side is empty. In other words, if the playback buffer size is reduced to the minimal buffer level but has not downloaded the current segments, then a video stalling will occur. Number of Video Stalling can be calculated as the total number of stalls in a video playback time period.

$$S_t = \frac{1}{K}\sum_{i=1}^{K} S_i, \forall 0 \le B_i \le B_{min} \quad (18)$$

To evaluate the performance of our bitrate adaptation scheme, we use an QoE metric based on the above parameters and also from the reference model provided by A. Bentaleb et al. [27] as follows:

$$QoE_i = \alpha_1 \times AVQ - \alpha_2 \times NVS - \alpha_3 \times S_t - \alpha_4 \times D_t \quad (19)$$

Both S_t and D_t are in seconds. ($\sum_{n=1}^{4}\alpha_n = 1$) are non-negative tunable values and selected based on [13] to test in our experiments. We convert the QoE score to a normalized QoE (N-QoE) accordingly to Table 3 to represent a user's satisfaction MOS (mean opinion score) from 1 to 5.

Table 3: Normalized QoE

QoE_i	N-QoE	Quality
$QoE_i \le 1$	1	Bad
$1 < QoE_i \le 2$	2	Below Average
$2 < QoE_i \le 3$	3	Average
$3 < QoE_i \le 4$	4	Good
$QoE_i > 4$	5	Excellent

5.3.3 Performance Evaluation

In this section, we compare our proposals (MSMA-0 and MSMA-1) with the existing methods such as PANDA, BBA, SARA, Aggressive (called AGG), SDN-based SARA and SDN-based BBA. In which, the SDN-based BBA and SDN-based SARA methods are the BBA and SARA methods that are combined with the fair bandwidth allocation of the MSMA module. To understand more about the existing solutions, let us to highlight some main points of the existing methods:

1. AGG only based on the throughput of the previous segment to estimate the throughput for the next segment. This method has advantages in fast adaptability to the strongly fluctuating bandwidth conditions, but has disadvantages in the stable bandwidth conditions compared to other methods.

2. For SARA method, the bitrate adaptation algorithm of SARA is both throughput-based and buffer based to determine which quality level would be chosen. SARA has shown a number of advantages over the AGG algorithm as it is able to maintain a higher quality level for longer time under stable bandwidth conditions because of having the segment capacity information, thus it could predict download time.

3. BBA is the buffer-based method only to select a version quality without estimating throughput.

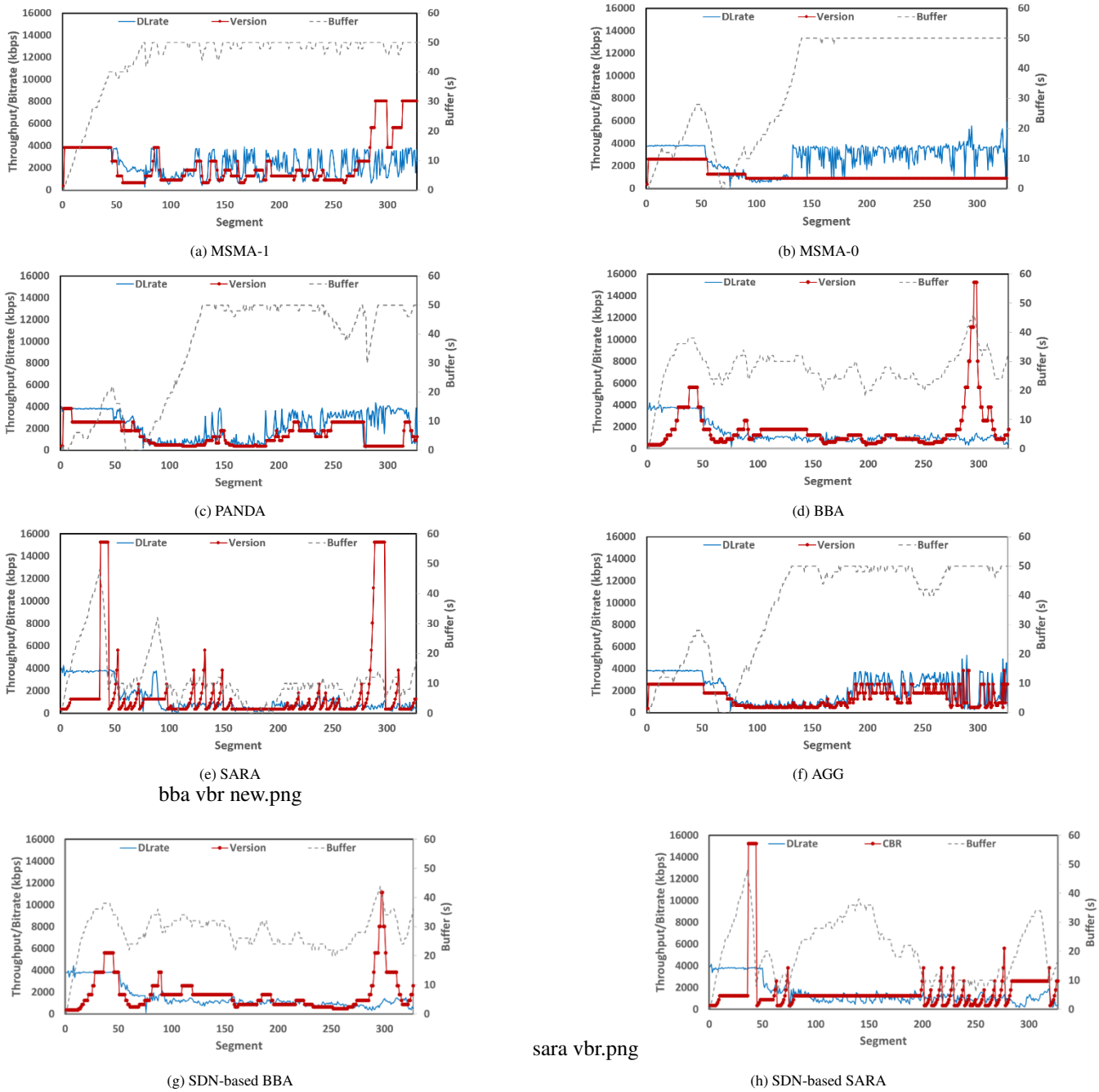
4. PANDA uses "probe-and-adapt" to build the rate adaptation algorithm and fair-share bandwidth for players. This solution achieves good stability but the averrage video quality is not high. On the other hand, PANDA's resource allocation is in a distributed rather than centralized manner like our proposed one so fairness is not high.

The download rate (DLrate), buffer level, and selected versions of all the methods are shown in Figure 4 (with VBR video) and Figure 5 (with CBR video). In experiments for the 10 Mbps, 7 Mbps, and 4 Mbps cases, the results show similar performances in terms of version and download rate. Therefore we only count in the 4 Mbps case as the representative for this type of evaluation. As Figure 4 demonstrate, in case of using the SDN controller with MSMA-0 and MSMA-1, video versions of the client are stable during the whole course without requesting any new connection.

Especially, MSMA-0 is very stable, however this method has average video bitrate much lower than of MSMA-1. The other methods have relatively the same average video bitrate. For MSMA-1 and BBA, due to buffer preestimation, the buffer does not go to zero. This means that the video will not be stalled. Other methods have almost video stalling for some times. For MSMA-0, the buffer is dropped low in segments of 68-78. In PANDA, the video is interrupted from segment 60 to segment 78. In SARA, the average buffer is very low, and there are 2 video interruptions, one at around segment 110 and one from segment 174 to 198. In the Aggressive method, video quality is fluctuated continuously which can be annoying to viewers; and video is also stalled from segment 66 to segment 78. With the support in bandwidth sharing of the SDN controller, the SDN-based BBA and SDN-based SARA methods have better performance in comparison with the BBA and SARA, as described in Figure 4g and Figure 4h. As we can see, the SDN-based SARA method has more stable quality levels compared to SARA in the duration from segment 80 to 200. This is because the fair bandwidth allocation of the MSMA module incorporates the segment-aware rate adaptive algorithm of SARA when multiple clients access the system. With good buffer-based bitrate adaptation, SDN-based BBA as well as BBA can avoid video freezing.

As Figure 5 illustrates, the status of clients in the case of the CBR video is more stable than the one of the VBR case, since the segment sizes of the CBR video are quite similar while segment sizes of the VBR video are varied and unpredictable. Average version of SDN-based BBA is increased by about 3.2% compared to BBA. SDN-based SARA had similar average quality levels to SARA, but the variation in quality levels was significantly reduced. This is quite an important parameter in QoE evaluation.

For the performance evaluation, the time when all clients begin to stream at the same time is selected. It take 20 seconds (10 segments) to to calculate results of the performance. Figure 6 compares the performance of our proposed solutions (MSMA-0 and MSMA-1) with PANDA, BBA, SARA and AGG using the three



(a) MSMA-1

(b) MSMA-0

(c) PANDA

(d) BBA

(e) SARA

(f) AGG

bba vbr new.png

(g) SDN-based BBA

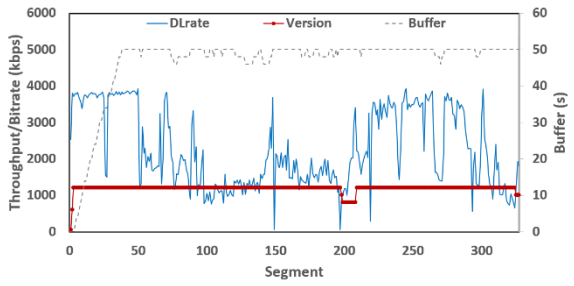
sara vbr.png

(h) SDN-based SARA

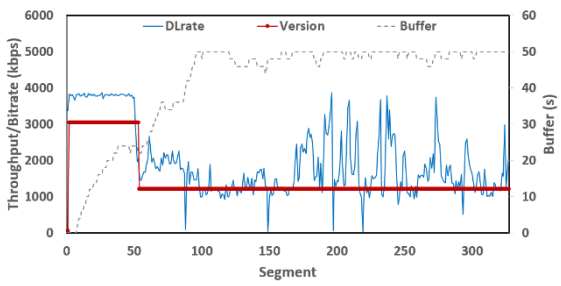
Figure 4: Download speed, buffer and version quality for all methods at a bottleneck link of 4 Mbps with VBR video

evaluation parameters: *Fairness*, *Efficiency* and *Stability*. For a more accurate evaluation, in this Figure 6, we get the average for all the three cases with the link capacities of 4 Mbps, 7 Mbps, and 10 Mbps. In real time, we set the mean value over three runs for 3 players. The results show that the fairness, efficiency, and stability obtained using the existing methods are much lower than those obtained using our proposed solution. However, due to the segment aware rate adaptation mechanism, in Figure 6a, the fairness between segments 4 and 7, SARA performs better than all of the others. This scheme uses all of the segment sizes, the estimated path bandwidth and current buffer occupancy to predict the amount of time needed to download the next segment. Therefore this method guarantees the fairness for all players. Using the existing streaming solutions as PANDA, BBA, SARA and AGG, the fairness in these cases varies from 0.634 to 0.860 and has an average of 0.699, 0.723, 0.77 and 0.687, respectively. The experiment also shows that the proposed architecture has improved the fairness score significantly, with scores ranging from 0.791 to 0.863 that results in an average of 0.816 which is very high. The efficiency of bandwidth allocation in the current solutions also has a considerable deterioration with values from 0.488 to 0.761 (i.e. average of 0.654, 0.655, 0.602 and 0.604, respectively). Whilst the efficiency of MSMA-1 varies from 0.716 to 0.970 (i.e average of 0.830).

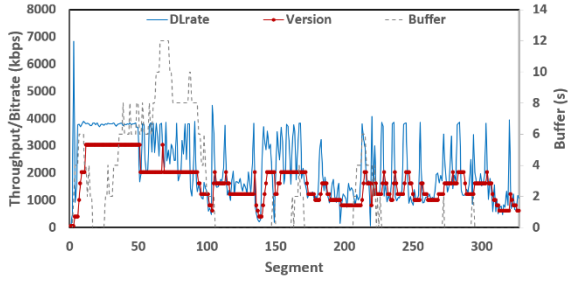
Figure6.c illustrates the comparison of the average stability of



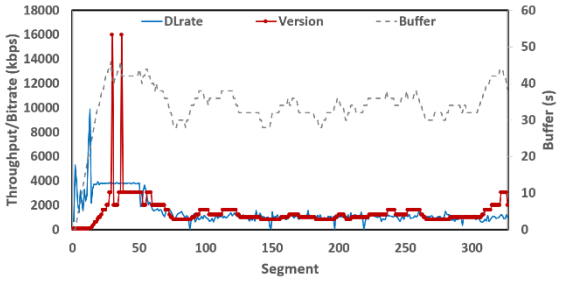
(a) MSMA-1



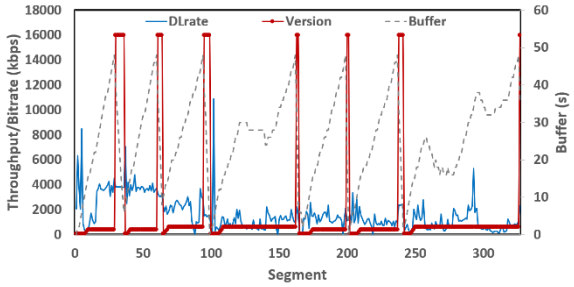
(b) MSMA-0



(c) PANDA

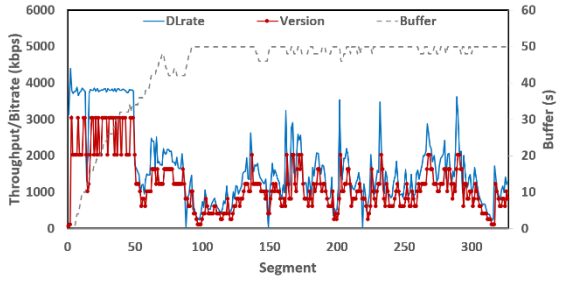


(d) BBA



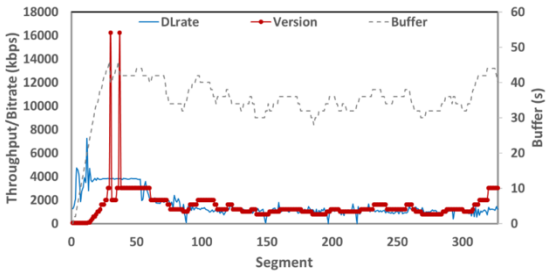
(e) SARA

bba cbr new.png

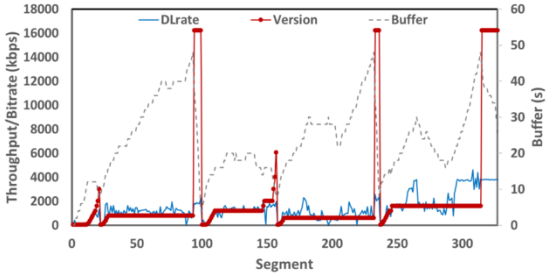


(f) AGG

sara cbr new.png



(g) SDN-based BBA



(h) SDN-based SARA

Figure 5: Download speed, buffer and version quality for all methods at a bottleneck link of 4 Mbps with CBR video

the 3 clients in all methods. It can be seen that the stability of our proposal is better than other schemes. Table 4 and Table 5 show the average fairness, efficiency and stability of the 3 clients in case of using the existing methods and our method. The shown results are taken from 20 seconds (i.e 10 segments) when all players are streaming at the same time.

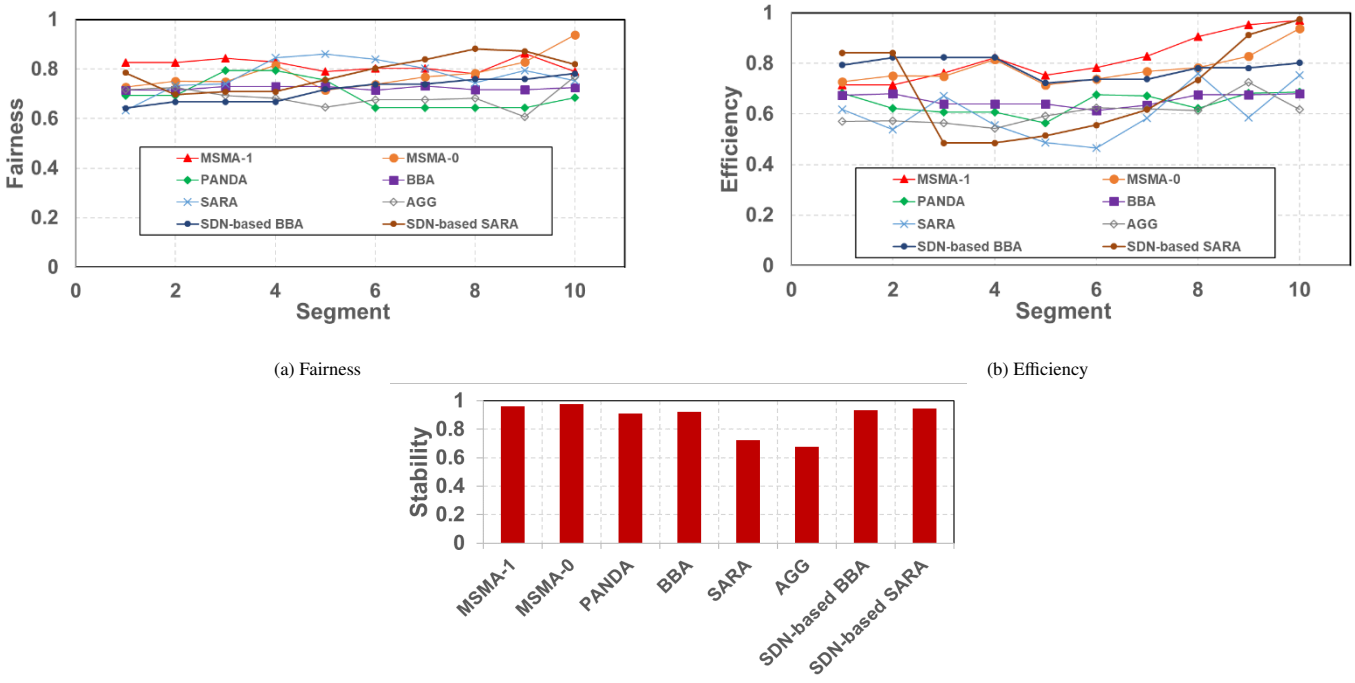
For VBR video Elephants Dream, in the following summary list, the consecutive numbers represent the results for MSMA-0, PANDA, BBA, SARA and AGG, respectively.

1) MSMA-1 outperforms the existing schemes in terms of fairness by an increase of 3.9%, 16.7%, 12.9%, 5.3% and 18.8%.

2) MSMA-1 improves bandwidth utilization in terms of efficiency by 3.3%, 26.9%, 26.7%, 37.8%, and 37.4%.

3) MSMA-1 increases the video stability by -1.5%, 5.6%, 4.6%, 33.1%, and 42.5%.

MSMA-1 improves two parameters: fairness and efficiency compared to MSMA-0. However, MSMA-0 has better stability than MSMA-1 because the clients in MSMA-1 are free to adapt their bitrates based on the throughput. And only when the throughput does not meet the requirements, the client will send the request to the controller. The SDN controller is then responsible for reallocating bandwidth to the clients accessing through that node, so that



(a) Fairness

(b) Efficiency

(c) Stability

Figure 6: Comparison average fairness, efficiency and stability of all methods

Table 4: Performance Comparison with VBR video

Metrics	MSMA-1	MSMA-0	PANDA	BBA	SARA	AGG	SDN-based BBA	SDN-based SARA
Fairness	0.816	0.785	0.699	0.723	0.775	0.687	0.714	0.807
Efficiency	0.830	0.781	0.654	0.655	0.602	0.604	0.783	0.697
Stability	0.962	0.977	0.911	0.920	0.723	0.675	0.933	0.945

the requesting clients receive the new bandwidth that can keep the most appropriate video version. As for MSMA-0, due to the use of the aggressive adaptive bitrate algorithm based on the bandwidth that has been divided evenly from the controller, it is still stable in choosing the bitrate or quality version of the next segment.

Similarly, with statistics in Table 5 (i.e. for CBR video "Destiny"), we can see that the two proposed methods (MSMA-1 and MSMA-0) outperform the rest. On another side, compared with traffic of the VBR video, traffic of the CBR video has higher Stability. The fairness of SDN-based BBA is almost similar to BBA, while the one of SDN-based SARA is slightly increased compared to SARA, by only 4%. SDN-based BBA effectively increases the bandwidth usage by 19.6% compared with BBA and SDN-based SARA increases by 15% compared to SARA. Thanks to the SDN-based fair bandwidth allocation, BBA and SARA have much better stability than the non-SDN solution.

Table 6 shows statistics from the experiments. As shown in the table, MSMA-1 outperforms the other methods. At first, with a reasonable throughput and buffer-based estimation mechanism, MSMA-1 is able to avoid stalling conditions whilst keeping the highest average value of buffer occupancy (i.e. 44.6s).

Secondly, MSMA-1, with the dynamic bandwidth allocation mechanism, can achieve the highest average bitrate of 2445.6 Kbps, which is an improvement of about 103.8%, 76%, 36.7%, 42.4% and 67.3% compared to MSMA-0, PANDA, BBA, SARA and AGG, respectively. Based on the QoE formula in Equation (19), the final QoE scores of MSMA-1 and MSMA-0 are calculated in comparison with 4 other solutions such as PANDA, BBA, SARA, AGG (Table 6). In fact, MSMA shows to achieve the highest QoE score compared to other competitors.

For the AGG algorithm, by requiring the highest bitrate according to the throughput without relying on the current buffer, the average video quality is low and the number of quality level switches-down is largest among all the methods. On the other hand, AGG also has the lowest QoE (i.e. 1.9).

SARA is the mixed throughput-buffer based algorithm, having the average video quality and QoE higher than AGG, but it is only medium compared to the other algorithms. BBA is the buffer-based algorithm, BBA successfully avoids stalling events which are very annoying to video viewers. The average bitrate of BBA is medium at only 1788.7 Kbps and its QoE stays at the good level (3.6).

With the estimation of the available bandwidth, a probing mech-

Table 5: Performance Comparison with CBR video

Metrics	MSMA-1	MSMA-0	PANDA	BBA	SARA	AGG	SDN-based BBA	SDN-based SARA
Fairness	0.971	0.965	0.764	0.713	0.716	0.873	0.932	0.917
Efficiency	0.879	0.796	0.904	0.665	0.726	0.708	0.812	0.803
Stability	0.975	0.957	0.931	0.949	0.826	0.703	0.941	0.955

Table 6: Quality Statistics of the solutions

Criteria	MSMA-1	MSMA-0	PANDA	BBA	SARA	AGG	SDN-based BBA	SDN-based SARA
Average Bitrate (kbps)	2445.6	1199.6	1389.5	1788.7	1717.4	1461.0	1812.3	1645.1
Average Buffer (s)	44.6	35.8	34.2	27.7	11.6	36.4	28.2	18.4
Number of Stalling	0	4	12	0	17	11	0	0
Average Version Index	6	4.5	5.1	5	5.2	5	5.4	5.1
Number of Version Switch-downs	21	3	7	17	20	40	16	15
N-QoE	4.25	3.6	3.54	3.55	2.8	1.9	3.96	3.74

anism is leveraged by PANDA to additively increase its sending rate, whilst decreasing sending rates multiplicatively, when congestion occurs. PANDA has a medium average bitrate at 1389.5 kbps and a relatively low number of switch-downs (7 times). The QoE score of PANDA also reaches 3.6.

For MSMA-0, the average bitrate is the lowest due to the use of the Aggressive bitrate adaptive algorithm on the client side, and MSMA makes its best to fairly allocate the bandwidth afor all clients. However, MSMA-0 has the lowest number of switch-downs (3 times). Besides QoE of MSMA-0 also reaches good quality (i.e. score of 3.6).

At the bottom lines, it's difficult to improve for all QoS parameters as well as the Quality of Experiment (QoE)- related parameters of video streaming. In fact, all of the proposed solutions need a tradeoff among these parameters to improve video streaming quality. However, our proposed solutions MSMA-0 and MSMA-1 have significantly improved average video quality, ensuring the highest average QoE compared to existing methods such as PANDA, BBA, SARA and AGG. Especially, MSMA-1 increases the QoE score by 18%, 20%, 20%, 51%, and 124% compare with MSMA-0, PANDA, BBA, SARA and Aggressive, respectively. As we can see, MSMA-0 does not SDN-based BBA and SDN-based SARA in terms of QoE, but our proposed MSMA-1 outperforms all compared methods.

Overall, our existing innovative algorithm is a combination of bitrate adaptation at the client side and resource allocation at the network side. The proposed rate adaptation method is to choose the best quality version on the current network condition and to avoid video stalling events. At the SDN controller, the resource allocation handles fair and adaptive bandwidth sharing in order to maximize the number of access clients, while trying to minimize the number of clients who got QoE degraded.

6 Conclusion and Future Work

In this work, we have presented an integrated solution of centralized SDN-based resource allocation at the network side and bitrate adaptation at the client side to resolve the problem of multi clients streaming video through a bottleneck connection. This causes several problems such as low quality of service, congestion, delay, and especially quality of experience of users. The experiments show that our proposed solution outperforms the predecessors in terms of Quality of Experience.The proposed scheme ensures bandwidth fairness, efficiency and stability. Another contribution of our work is to ensure QoE of as many ongoing clients as possible. Additionally, the method can prevent congestion at bottleneck point when there is insufficient bandwidth by rejecting a new incoming request

Our future work will be building the whole SDN-based architecture to allocate network resources and to maximize QoE for each client, while avoiding scalability issues among many competing clients in a shared environment. Multiple heterogeneous scenarios with multiple SDN controllers and various clients will also be taken into account.

Acknowledgment

This paper is an extension of work originally presented in the ICTC conference [4]. This work was funded by Vingroup and supported by Vingroup Innovation Foundation (VINIF) under project code VINIF.2020.DA03.

References

[1] "Cisco Visual Networking Index: Forecast and Methodology, 2016-2021"; San Jose, CA, USA, Cisco, White Paper, 2016.

[2] Open Networking Foundation (ONF), "Software Defined Networking: the new norm for network,"; White paper, 2012.

[3] N. McKeown et al., "OpenFlow: Enabling Innovation in Campus Networks," ACM SIGCOMM Comput. Commun. Rev., **38**(2), 69–74, 2008, doi: 10.1145/1355734.1355746.

[4] P. H. Thinh. P. N. Nam, N. H. Thanh, T. T. Huong, "SDN – based Dynamic Bandwidth Allocation for Multiple Video-on-Demand Players over HTTP," proceeding 2019 Int. Conf. Inf. Commun. Technol. Converg. (ICTC), 16-18 Oct. 2019, Jeju, Korea, 163–168, 2019, doi: 10.1109/ICTC46691.2019.8939834.

[5] L. R. Romero, "A Dynamic Adaptive HTTP Streaming Video Service for Google Android," M.S. Thesis, R. Inst. Technol. (KTH), Stock., 148, 2011.

[6] T. C. Truong, D. Q. Ho, J. W. Kang, A. T. Pham, and S. Member, "Adaptive Streaming of Audiovisual Content using MPEG DASH," IEEE Trans. Consum. Electron., **58**(1), 78–85, 2012, doi: 10.1109/TCE.2012.6170058.

[7] T.-Y. Huang, R. Johari, M. Watson, M. Trunnell, and N. McKeown, "A buffer-based approach to rate adaptation: Evidence from a large video streaming service," ACM SIGCOMM Comput. Commun. Rev., **44**(4), 187–198, 2015.

[8] T.-Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson, "Using the Buffer to Avoid Rebuffers: Evidence from a Large Video Streaming Service," arXiv Prepr. arXiv1401.2209, 2014, https://arxiv.org/abs/1401.2209.

[9] Y. Zhou, Y. Duan, J. Sun, and Z. Guo, "Towards simple and smooth rate adaption for VBR video in DASH," 2014 IEEE Vis. Commun. Image Process. Conf. VCIP 2014, 9–12, 2015, doi: 10.1109/VCIP.2014.7051491.

[10] H. N. Nguyen, T. Vu, H. T. Le, N. N. Pham, and T. C. Truong, "Smooth quality adaptation method for VBR video streaming over HTTP," 2015 Int. Conf. Comput. Manag. Telecommun. ComManTel 2015, 184–188, 2016, doi: 10.1109/ComManTel.2015.7394284.

[11] S. Latré, J. Famaey, F. De Turck, M. Claeys, and S. Petrangeli, "QoE-Driven Rate Adaptation Heuristic for Fair Adaptive Video Streaming," ACM Transaction Multimedia Computing Communication Application, **12**(2), 1–24, 2015, https://doi.org/10.1145/2818361.

[12] P. Juluri, V. Tamarapalli, and D. Medhi, "SARA: Segment aware rate adaptation algorithm for dynamic adaptive streaming over HTTP," 2015 IEEE Int. Conf. Commun. Work. ICCW 2015, 1765–1770, 2015, 10.1109/ICCW.2015.7247436.

[13] J. Jiang, "Improving Fairness , Efficiency , and Stability in HTTP-based Adaptive Video Streaming with FESTIVE," Proc. 8th Int. Conf. Emerg. Netw. Exp. Technol., 97–108, 2012, https://doi.org/10.1145/2413176.2413189.

[14] Z. Li et al., "Probe and Adapt: Rate Adaptation for HTTP Video Streaming At Scale," IEEE J. Sel. Areas Commun. **32**(4), 719–733, 2014, doi: 10.1109/JSAC.2014.140405.

[15] A. Mansy, "Network-layer Fairness for Adaptive Video Streams," IEEE, IFIP Netw. Conf. (IFIP Networking), pp. 1–9, 2015, doi: 10.1109/IFIPNetworking.2015.7145310.

[16] M. Ghobadi and M. Mathis, "Trickle: Rate Limiting YouTube Video Streaming," Present. as part 2012 USENIX Annu. Tech. Conf. (USENIX-ATC 12), 191–196, 2012, https://www.usenix.org/conference/atc12/technical-sessions/presentation/ghobadi.

[17] S. Akhshabi, L. Anantakrishnan, C. Dovrolis, and A. C. Begen, "Server-Based Traffic Shaping for Stabilizing Oscillating Adaptive Streaming Players," Proceeding 23rd ACM Work. Netw. Oper. Syst. Support Digit. audio video, 19–24, 2013, doi: 10.1145/2460782.2460786.

[18] J. J. Quinlan, A. H. Zahran, K. K. Ramakrishnan, and C. J. Sreenan, "Delivery of Adaptive Bit Rate Video: Balancing Fairness , Efficiency and Quality," 21st IEEE Int. Work. Local Metrop. Area Networks, 1–6, 2015, doi: 10.1109/LANMAN.2015.7114736.

[19] S. Ramakrishnan, X. Zhu, F. Chan, and K. Kambhatla, "SDN Based QoE Optimization for HTTP-Based Adaptive Video Streaming," 2015 IEEE Int. Symp. Multimed., 120–123, 2015, doi: 10.1109/ISM.2015.53.

[20] C. Cetinkaya, Y. Ozveren, and M. Sayit, "An SDN-assisted System Design for Improving performance of SVC-DASH," 2015 Fed. Conf. Comput. Sci. Inf. Syst. (FedCSIS). IEEE, **5**, 819–826, 2015, doi: 10.15439/2015F333.

[21] P. Georgopoulos, Y. Elkhatib, M. Broadbent, M. Mu, and N. Race, "Towards Network-wide QoE Fairness Using OpenFlow-assisted Adaptive Video Streaming," Proc. 2013 ACM SIGCOMM Work. Futur. human-centric Multimed. Netw., 15–20, 2013, doi: 10.1145/2491172.2491181.

[22] G. Cofano, L. De Cicco, T. Zinner, and S. Mascolo, "Design and Experimental Evaluation of Network-assisted Strategies for HTTP Adaptive Streaming," Proc. 7th ACM Multimed. Syst. Conf. (MMSys 2016), 1–12, 2016, doi: 10.1145/2910017.2910597.

[23] J. Chen, M. Ammar, M. Fayed, and R. Fonseca, "Client-Driven Network-level QoE fairness for Encrypted 'DASH-S'," Proc. 2016 Work. QoE-based Anal. Manag. Data Commun. Networks, 55–60, 2016, doi: 10.1145/2940136.2940144.

[24] K. T. Bagci, S. Member, K. E. Sahin, S. Member, and A. M. Tekalp, "Compete or Collaborate: Architectures for Collaborative DASH Video over Future Networks," IEEE Trans. Multimed., **19**(10), 2152–2165, 2017, doi: 10.1109/TMM.2017.2736638.

[25] O. El Marai and T. Taleb, "Online Server-side Optimization Approach for Improving QoE of DASH Clients," GLOBECOM 2017-2017 IEEE Glob. Commun. Conf. IEEE, 1–6, 2017, doi: 10.1109/GLOCOM.2017.8254128.

[26] S. Zhao and D. Medhi, "SDN-Assisted Adaptive Streaming Framework for Tile-Based Immersive Content Using MPEG-DASH," 2017 IEEE Conf. Netw. Funct. Virtualization Softw. Defin. Networks, 1–6, 2017, doi: 10.1109/NFV-SDN.2017.8169831.

[27] A. Bentaleb, A. C. Begen, and R. Zimmermann, "SDNDASH: Improving QoE of HTTP Adaptive Streaming Using Software Defined Networking," ACM Multimed., 1296–1305, 2016, doi: 10.1145/2964284.2964332.

[28] A. Bentaleb, A. C. Begen, R. Zimmermann, and S. Harous, "SDNHAS: An SDN-enabled architecture to optimize QoE in HTTP adaptive streaming," IEEE Trans. Multimed., **19**(10), 2136–2151, 2017, doi: 10.1109/TMM.2017.2733344.

[29] A. Bentaleb, A. C. Begen, S. Member, R. Zimmermann, and S. Member, "QoE-Aware Bandwidth Broker for HTTP Adaptive Streaming Flows in an SDN-Enabled HFC Network," IEEE Trans. Broadcast., **64**(2), 575–589, 2018, doi: 10.1109/TBC.2018.2816789.

[30] J. Jiang, L. Hu, P. Hao, and R. Sun, "Q-FDBA: improving QoE fairness for video streaming," Multimed. Tools Appl., **77**(9), 10787–10806, 2018, doi: 10.1007/s11042-017-4917-1.

[31] T. Abar, A. Ben, L. Sadok, and E. Asmi, "Enhancing QoE based on Machine Learning and DASH in SDN networks," 2018 32nd Int. Conf. Adv. Inf. Netw. Appl. Work., 258–263, 2018, doi: 10.1109/WAINA.2018.00095.

[32] A. Ahmad, A. Floris, and L. Atzori, "Towards Information-centric Collaborative QoE Management using SDN," 2019 IEEE Wirel. Commun. Netw. Conf., 1–6, 2019, doi: 10.1109/WCNC.2019.8885412.

[33] L. Guillen, S. Izumi, and T. Abe, "SAND/3: SDN-Assisted Novel QoE Control Method for Dynamic Adaptive Streaming over HTTP/3," Electron. Multidiscip. Digit. Publ. Inst., **8**(8), 1–17, 2019, doi: 10.3390/electronics8080864.

[34] Pham Hong Thinh, Nguyen Thanh Dat, Pham Ngoc Nam, Nguyen Huu Thanh, Nguyen Minh Hien & Truong Thu Huong. "An Efficient QoE-Aware HTTP Adaptive Streaming over Software Defined Networking,"Mobile Networks and Applications, 1-13,2020, https://doi.org/10.1007/s11036-020-01543-1.

[35] T. Huang, B. Heller, and N. Mckeown, "Confused , Timid , and Unstable: Picking a Video Streaming Rate is Hard," Proceedings of the 2012 internet measurement conference.ACM, 225–238, 2012, doi: 10.1145/2398776.2398800.

[36] X. Yin, M. Bartulovi, V. Sekar, and B. Sinopoli, "On the Efficiency and Fairness of Multiplayer HTTP-based Adaptive Video Streaming" 2017 Am. Control Conf. (ACC). IEEE, 4236–4241, 2017, doi: 10.23919/ACC.2017.7963606.

[37] Bitmovin, "Libdash - bitmovin." [Online]. Available: https://github.com/bitmovin/libdash. [Accessed: 25-Nov-2020].

[38] "Floodlight." [Online]. Available: http://www.projectfloodlight.org/floodlight/. [Accessed: 25-Nov-2020].

[39] "Mininet." [Online]. Available: http://mininet.github.io/. [Accessed: 25-Nov-2020].

[40] O. O. M. P. Studio, "Elephants dream," 2009. [Online]. Available: https://orange.blender.org/. [Accessed: 25-Nov-2020].

[41] V. D. Fabien Weibel, Manuel Alligne, Sandrine Wurster, "Destiny," 2012. [Online]. Available: http://fweibel.com/destiny . [Accessed: 05-Nov-2020].

[42] W. R. Jain, Rajendra K and Chiu, Dah-Ming W and Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer system," East. Res. Lab. Digit. Equip. Corp. Hudson, MA, 1984.

[43] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hossfeld, and P. Tran-gia, "A Survey on Quality of Experience of HTTP Adaptive Streaming," Ieee Commun. Surv. Tutorials, **17**(1), 469–492, 2015, doi: 10.1109/COMST.2014.2360940.