BBGDASH: A Max-Min Bounded Bitrate Guidance for SDN Enabled Adaptive Video Streaming

Ali Edan Al-Issa[§], Abdelhak Bentaleb[*], Thomas Zinner[†], Is-Haka Mkwawa[§], Bogdan Ghita[§]
[§]Centre for Security, Communications and Network Research, [*]School of Computing, [†]INET: Internet Network Architectures,
[§]Plymouth University, [*]National University of Singapore, [†]TU Berlin,
E-mail: ali.alissa,is-haka.mkwawa,bogdan.ghita@plymouth.ac.uk, bentaleb@comp.nus.edu.sg, zinner@inet.tu-berlin.de

***Abstract*—The increase in video traffic and the end-users demands for higher quality video have triggered academia and industry to find novel mechanisms for media distribution. Among the available streaming services, HTTP adaptive streaming (HAS) is being the de facto standard for multi-bitrate streaming. Recent studies show that the bitrate adaptation of client-driven HAS applications is challenging due to the fact that they based on distributed, and local decisions for adapting the quality on a per-client basis. Software-defined networking (SDN) has emerged as a new network paradigm to provide centralised management. The programmability and flexibility of the SDN can be utilised o enhance the delivery of video over the Internet. In this paper, we present a novel and scalable network assistance approach (denoted BBGDASH) that identifies the boundary range of the requested bitrate levels while preserving HAS at the end player. Experimental results demonstrate the potential of the proposed approach for delivering the video over SDN-enabled networks.**

***Index Terms*—SDN, QoE; DASH; Fairness; OpenFlow; Stability; Adaptive Video Streaming; Streaming architecture; Bounded Bitrate Guidance; Network-Assistance**

I. INTRODUCTION

The Internet video traffic has rapidly increased over the past few years and it is anticipated to dominate future networks. As reported in the Cisco Visual Networking Index [1], the global IP video traffic will represent 82% of all consumer Internet traffic by 2021, up from 73% in 2016. The increase in video traffic and users' demand for higher quality video has triggered service providers (SPs) and mobile operators to find novel solutions for bandwidth management and better ways of resource utilisation on their existing network infrastructure for optimising the quality of experience (QoE) experienced by the end users. However, achieving a good QoE is a challenging task due to the wide range of client patterns (i.e. devices' and videos' characteristics), and the variability of the network conditions [2]. HTTP Adaptive Streaming (HAS) protocol is the dominant mechanism for video streaming and adopted by most streaming services such as YouTube and Netflix. It covers different implementations that have similar adaptation principles, with Dynamic Adaptive Streaming over HTTP (DASH) [3] being one of the most popular ones.

In DASH, the video content is encoded into different bitrate levels and segmented into small temporal chunks of typically 2 to 10 seconds. Furthermore, each video client is equipped with an adaptive bitrate algorithm (ABR), which based on different approaches and metrics to dynamically adapt the perceived quality to the network conditions.

Despite the great benefits they bring, purely client-based HAS solutions also face challenges relating to quality fluctuations when they compete for a shared link [4]. The first challenge can be raised from the mismatched ON/OFF behaviour of the different players, which can lead to inaccurate throughput prediction. Further, other QoE related metrics such as the impact of the end-user device and the user context are not considered within the ABR algorithms, which might result in an unfair quality distribution among different users. To overcome these issues, network and video providers must exchange information and cooperate. Emerging technologies such as the server and network assisted delivery (SAND) architecture [5], offers standard signalling for network-to-client and network-to-network communication of quality-relevant information. This architecture allows the network elements to apply the appropriate policies that match network condition and user requirements. Software Defined Networking (SDN) [6] is an emerging technology to deploy such architecture, which provides centralized control for efficient and flexible network management.

To overcome the mentioned issues, this paper presents three contributions towards improving the perceived QoE for video streaming. First, it introduces a novel guidance mechanism (BBGDASH) that provides guidance to the client without moving the whole control logic to an additional entity nor relying purely on the client side decision. Second, we implemented a network-assisted video streaming framework that leverages the functionality of SDN to deploy BBGDASH. Third, a use-case based evaluation has been conducted to present the feasibility and the potential of the proposed approach.

The remainder of this paper is structured as follows, Section II presents the related work. Section III introduces the SDN-based architecture of BBGDASH. Section IV models the system and presents the guidance algorithm for computing the boundary bitrate levels for each DASH player. Section V details the experimental setup. Then, the performance evaluations of the proposed approach are outlined in Section VI. Finlay, Section VII concludes the paper with a summary of the achievements and possible avenues for future research.

II. RELATED WORK

The use of HAS for over-the-top (OTT) video services has driven researchers for developing new adaptation approaches that optimise the perceived QoE through dynamically adapting the requested quality according to network conditions. Different adaption approaches have presented in the literature [7]. The majority of these approaches are deployed at the end client, which can be broadly classified into throughput-based, buffer-based, and hybrid-based approach.

Throughput-based mechanisms such as [8] [9] rely on the estimated throughput for adapting the quality of the next video chunk. For example, in [8] the authors proposed a method that compares the segment download time with segment playout time in order to increase or decrease the video bitrate. Albeit, this approach can reach the optimum level quickly, it cannot provide a fair quality distribution between users. In order to address this challenge, Jiang et al. [9], proposed FESTIVE, which applies randomised chunk scheduling to avoid segments overlapping between different players. FESTIVE improves the fairness by 40%, stability by 50% and efficiency by almost 10%, but it does not support QoE fairness among heterogeneous competing DASH players in a shared network. The Cross-Session Stateful Predictor (CS2P) [10] algorithm also falls within the throughput-based category, employing hidden Markov chains to predict the throughput; even though this approach is easy to deploy, it may perform poorly under unstable network conditions.

The buffer-based approaches in [11] and [12] depend only on the buffer level to adapt the quality. In [11], authors defined two thresholds, B_{min} and B_{max} and three level of the buffer occupancy including B: reservoir ($B < B_{min}$), upper reservoir ($B > B_{max}$) and cushion ($B_{min} < B < B_{max}$). The client requests the lowest or the highest quality if the occupancy of the buffer is in the reservoir or upper reservoir range, respectively. In the Buffer Occupancy based approach (BOLA) [12], the authors formulated the ABR decision as a utility maximisation problem that includes the average video bitrate and the rebuffering duration. Furthermore, Lyapunov optimisation has been employed in BOLA to achieve online control.

Another adaptation approach is the hybrid, which combines both throughput estimation and buffer occupancy to adapt the quality as in Probe AND Adapt (PANDA) [13], which mimics the congestion control at the application layer. In PANDA, the requested bitrate and the arrival time of the consequence subsequent segments are used to select the bitrate of the next segment by taking into account the buffer occupancy to reduce freezing. The evaluation results showed that PANDA could reduce the quality instability by over 75% when compared with other conventional algorithms [8]. A recent game theory-based ABR (GTA) has presented by Bentaleb et al. [14]. GTA leverages the game theory capabilities [15] and formulates the ABR decisions problem as bargaining and consensus problem. The proposed algorithm achieves a good performance where it outperforms its competitors in the average QoE by 38.5% and in quality stability by 62%.

Despite its success, the client-based adaptation approach of HAS streaming may fail to provide a fair quality distribution among heterogeneous users. As different users may have different requirements in term of device screen size and resolution, they should be treated in different ways. In order to overcome such limitations, a large body of research focused on network-based approaches [16] [17] [18] [19]. In [16], Mok et al. deployed a proxy server approach (QDASH) between the client and the server to measure the available bandwidth, round-trip time and loss rate. Based on the measurement, QDASH can guide the end user for the proper bitrate that fits with the network conditions. However, this approach can generate network overhead and cannot be deployed if the traffic is encrypted.

Another research direction is based on a centralised approach (e.g. SDN) to optimise the received QoE and provide a fair allocation among users. SDN provides a global view for the whole network and flexible way to manage the network. On top of that, introducing a standard signalling scheme such as SAND, enables active cooperation between network elements. Kleinrouweler et al. [17] proposed an SDN-based architecture for DASH-aware streaming, with the SDN controller providing network and application level assistance. Performance evaluation tests showed that the approach provides a stable video delivery, but the available bandwidth is divided equally among users, which can lead to unfair QoE, particularly in a heterogeneous environment. Bentaleb et al. [18] presented an SDN-enabled DASH architecture (SDNDASH) to maximise per-client QoE by dynamically allocating network resources and guide each client for the optimal bitrate level for the next chunks to be downloaded. Petrangeli et al. proposed in [20] an ABR algorithm for providing a fair distribution within a multi-client setting. They deployed an in-network system of coordination proxies to allocate the available resource fairly between the clients. The in-network proxies estimate the available bandwidth periodically and send the fair signal to the clients, whereas clients perform the bitrate adaptation based on the received signal. However, there is no consideration for heterogeneity among different users.

Georgopoulos et al. [19] proposed an OpenFlow-assisted QoE Fairness Framework (QFF) that leverages the OpenFlow to provide a fair-QoE among heterogeneous users. QFF is composed of a set of components, including utility and optimisation functions. The utility function (UF) is set to provide a model that maps the bitrate of a video at a particular resolution and the QoE perceived by the user. The Optimisation Function (OF) uses the models provided by the UF to find the optimum set of bitrates that ensures QoE fairness across all DASH players in the network. However, the utility function of QFF does not consider the buffer level occupancy, which could lead to buffer starvation. Further, this approach cannot be deployed within encrypted video streaming.

Another approach named Oboe has presented by Zahaib et al. [21], in which the heuristic-specific parameters can be tuned in a real-time to meet the current network conditions.

Oboe is based on a pre-computed ConfMap to find the best configurations for a given ABR algorithm. The experimental results showed that Oboe significantly improved the client based adaptation approaches like BOLA [12] and FastMPC [22].

Most of the aforementioned works either optimise the adaptation decision locally or move the decision logic to an additional entity. With purely client based HAS applications, the distributed per-client based optimisation might results in an unfair QoE allocation among the involved clients. On the other hand, moving the adaptation decision to a specific element results in harming the principle of HAS at the end player based decision and generating scalability issues due to the deployment of such an approach leads to severe overhead. Overall, it is essential to introduce a hybrid guidance approach that provides guidance to the client without moving the whole control logic to an additional entity nor relying purely on the client side decision.

III. SYSTEM ARCHITECTURE

BBGDASH's architecture leverages the features and capabilities of SDN to dynamically manage the available resources among users in a way that provides a fair QoE allocation among users. As shown in Figure 1, the proposed solution consists of three layers, namely, the application layer, the control layer, and the infrastructure layer. Several functional entities and communication interfaces are distributed within the three layers. A detailed design of BBGDASHs architecture will be discussed in the remainder of this section.

A. Application Layer

This layer is composed of three entities, namely, the DASH server, the DASH player and the SDN-based DASH agent.

1) DASH Server: HTTP server hosting video content encoded with different bitrate levels and segmented into small chunks. For each video, we provide an objective quality measurement based on the VQM metric [23], which is used as a reference for measuring the perceived quality.

2) DASH Player: dash.js reference player need to be modified to implement the proposed hybrid guidance mechanism. Upon the reception of the MPD file, the DASH player extracts the quality related parameters together with the detected screen resolution and sends them to the DASH manager to define the maximum and the minimum boundary bitrate levels for each user. Each user has a local ABR algorithm as described in Section II. The local ABR algorithm is based on the received signal from the DASH manager for identifying the range of the bitrate levels that can be requested. Besides the streaming service of the DASH player, it also measures the application level Key Quality Indicators (KQI) and stores the results of the probe in a shared database to be accessed by both network and service providers. The collected KQIs include the average bitrate, initial buffering time, number of stalling, stalling duration, along with the number and amplitude of quality switching. The initial buffering time represents the time difference between the video requesting time and the video playback time. The number of stalling is the number of times when user experience playback pauses due to buffer overdraw, while stalling duration is counted as the time consumed during the stalling event. The number of quality switching represents the number of times the user experiences quality level changes.

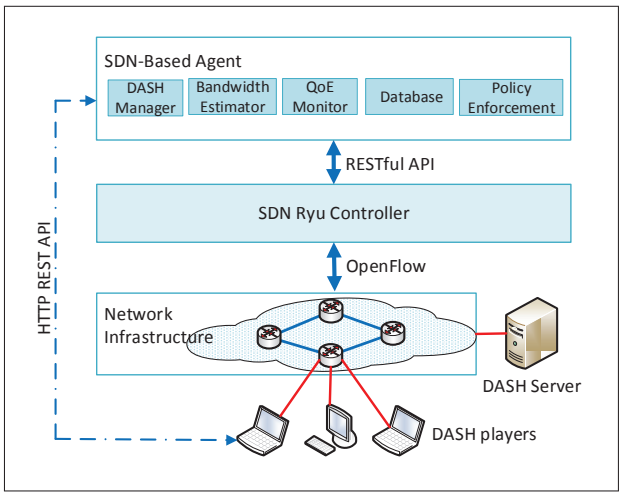


Fig. 1: BBGDASH Architecture

3) SDN-Based DASH Agent: This agent is implemented on top of the control plane, which consists of five modules, namely DASH manager, bandwidth estimation, QoE estimation, database, and policy enforcer. In the following, we describe the details of each module:

DASH manager: This module is aware of the active DASH flows and the available bandwidth through the interaction with other modules in this plane. It is based on the proposed algorithm for allocating the network resources and guiding DASH players for defining the boundary levels of the requested bitrate.

Bandwidth Monitor: An accurate bandwidth estimation is a strict prerequisite for computing the optimal boundary bitrate levels that maximise the perceived quality and the network resources while ensuring fairness. This module calculates the available bandwidth through probing the network statistics at the infrastructure layer.

QoE Monitor: This module estimates the perceived QoE at the end player by considering the four main QoE metrics: received bitrate, initial delay, stalling and video bitrate switching.

Database Module: This module stores the perceived QoE metrics, configuration parameters, estimated bandwidth, and the specifications of the active DASH flows. Further, it acts as an interface between the database engine and the other module of the SDN-based agent. It offers the DASH manager module periodically with the required parameters such as number of players, available bandwidth, and the specification of the requested videos, which are used as inputs to calculate the suitable bitrate ranges.

Policy Enforcement: Based on the output of the proposed algorithm, the policy enforcement module applies the actions on the network and application levels. The network level action is implemented through the SDN controller in the way of network resource slicing, while the application level actions can be applied through sending the recommended maximum and minimum bitrate levels (i.e. bitrate range) for each DASH player.

B. Control Layer

This layer represents the central network component that bounds the application layer to the infrastructure layer. Further, it is designed to offer a set of QoE-related services to the application layer through a northbound application programming interface such as RESTful [24]. It further provides access to the forwarding devices on the infrastructure layer via a southbound protocol, such as OpenFlow [25] to perform the desired network-level actions.

C. Infrastructure Layer

This layer represents the set of OpenFlow-enabled forwarding devices, such as switches and routers. These devices are responsible for packet forwarding and allocating the resources based on the SDN controller policies. In this paper, we used Open VSwitch (OVS) which acts as the forwarding devices of the proposed architecture.

IV. SYSTEM MODEL AND ALGORITHM DESCRIPTION

The main issue that most ABR algorithms aim to address is resource provisioning for N concurrent players that access a shared bottleneck link with a limited capacity of W_t at time t, requiring a potential maximum bandwidth $W_{max} > W_t$. We assume that each player $p_i \in P$ has a specific device resolution r_{p_i}. Further, for each video $v \in V$, the set of players may request different videos regarding the maximum bitrate $l_{p_i}^{max} \in L_{p_i}^v$ or the number of the bitrate representations $L_{p_i}^v$ of video v, such that:

$$\sum_{i=1}^{N} l_{p_i}^{max} > W_t, \forall p_i \in P, i = [1, \ldots, N], \tag{1}$$

$$\frac{\sum_{v=1}^{V}\sum_{i=1}^{N} L_{p_i}^v}{N \times L_{p_i}^v} \neq 1, \qquad \frac{\sum_{i=1}^{N} l_{p_i}^{max}}{N \times l_{p_i}^{max}} \neq 1, \forall v \in V, \tag{2}$$

Where P denote the set of players, V is the set of videos stored in the DASH server, $L_{p_i}^v$ is the complete set of bitrate levels of video v requested by $p_i \in P$, and $l_{p_i}^{max}$ is the maximum available bitrate level of video v requested by $p_i \in P$.
In this context, enforcing the same bitrate allocation for all users may result in an unfair distribution and inefficient of both QoE and network resources. In order to meet the user QoE requirements and achieve a high level of fairness among the users, we develop a network-assisted approach based on the concept of max-min fairness. Our solution aims to provide an efficient and fair allocation for a set of different players sharing the common bottleneck and requesting different videos (e.g., in terms of type, number of bitrate levels, etc.).

TABLE I: Notation and Symbols Description.

Symbol	Descriptions
N	Total Number of DASH players
N_{cl}	Number of clusters cl of specific screen resolution r
N_{cl}^p	Number of DASH players in cluster cl_j
$l_{cl_j}^{ini,p_i}$	Initial bitrate selection for player p_i in cluster j
$OPT_l^{cl_i}$	Optimal bitrate for cluster cl_i
S_f	Scaling factor
W_t	The total capacity of the network at time t
$[l_{p_i}^1 : l_{p_i}^{max}]$	Available bitrate levels of video v requested by player p_i
$l_{p_i}^{diff}$	The difference between the current bitrate level and the next level of the video v requested by player p_i
W^{remin}	The remaining bandwidth after the first allocation
V_1	Maximum allowed bitrate level set for the active players
V_2	Minimum allowed bitrate set for the active players
α	The difference between max. and min. selected bitrates

In our approach, the system is modelled as undirected graph $G = (X, Y)$, where X represents the set of nodes x and Y is the set of links y between the nodes. The set of nodes in the proposed architecture includes a number of subsets, encompassing DASH clients P, DASH server S, forwarding devices F, and the SDN controller A. Each client $p \in P$ has at least one link $y \in Y$ connecting it to the access node $f \in F$. As the first assumption that the network has only one bottleneck in the access network with total capacity of W_t at time t. At any time t, each player p_i requires $w_{p_i}[t] \in \mathbb{R}+$ to stream a video v with a bitrate level equal to $l_{p_i} \in L^v$. Furthermore, the total bandwidth allocated to all players must not exceed the total capacity of the shared link, as the following:

$$\sum_{i=1}^{N} w_{p_i}[t] \leq W_t, \forall t \in T \tag{3}$$

The main objective is to maximise the minimum bitrate among users. The objective function f is described as follows:

$$f = \begin{cases} \max\left(\frac{\sum_{i=1}^{N} l_{p_i}}{N}\right) \\ s.t \quad \sum_{i=1}^{N} w_{p_i}[t] \leq W_t, \forall t \in T \\ \qquad w_{p_i}[t] \geq 0, l_{p_i} \in L_{p_i}^v \end{cases} \tag{4}$$

To solve the objective function (Eq. 4), we used a dynamic programming based Algorithm (Bitrate Selection Algorithm 1) to calculate the optimal boundary range for the maximum and minimum bitrate levels for each player. The proposed algorithm is based on the fact that different players having different requirements may request videos with heterogeneous representation levels, therefore dividing the available bandwidth equally among different users result with unfair and inefficient resource allocation, and thus, a bad QoE distribution among them.

Algorithm 1 Bitrate Selection Algorithm

Input: $N, L^v_{p_i}, W_t, N_{cl}, N^p_{cl}, OPT^{cl_i}_l, \alpha$

1: CALCULATE $S_f \leftarrow \frac{W_t}{\sum_{i=1}^{N_{cl_i}} N^p_{cl_j} * OPT^{cl_i}}$
2: **for each** *cluster* $cl \in CL$ **do**
3: $l^{ini,p_i}_{cl} \leftarrow OPT^{cl} * S_f$
4: **for each** *cluster* $cl \in CL$ **do**
5: **for each** *player* $p_i \in P$ **do**
6: FINDMAX$(l^m_{p_i} \in L^v) \mid l^m_{p_i} \leq l^{ini,p_i}_{cl}$
7: INSERT$(l^m_{p_i})$ *into* $V_1[]$
8: **for each** *player* $p_i \in P$ **do**
9: **if** $l^m_{p_i} \neq l^{max}_{p_i}$ **then**
10: INSERT$(l^{m+1}_{p_i})$
11: $l^{diff}_{p_i} = (l^{m+1}_{p_i}) - (l^m_{p_i})$
12: **else**
13: $l^{m+1}_{p_i} = \infty$
14: INSERT$(l^{m+1}_{p_i})$ *into* $V_2[]$
15: INSERT$(l^{diff}_{p_i})$ *into* $V_3[]$
16: CALCULATE$(W^{remin}) = \sum_{i=1}^{N} l^{ini,p_i}_{cl} - l^m_{p_i}$
17: SORT$(V1)$
18: **while** $W^{remin} \geq$ MIN(V_3) **do**
19: **for each** $a \in V_1$ **do**
20: **if** $V_3[a] \leq W^{remain}$ **and** $V_2[a] \neq \infty$ **then**
21: $V_1[a] = V_2[a]$
22: $W^{remin} = W^{remin} - V_3[a]$
23: **else** CONTINUE()
24: **for each** $a \in V1$ **do**
25: $V2[a] = V1[a] - \alpha$
26: RETURN(V_1, V_2)

The algorithm starts with initialising the input variables including N, $L^v_{p_i}$, W_t, N_{cl}, N^p_{cl}, $OPT^{cl_i}_l$ and α which are defined with other corresponding notations in table 1. In (line 1), the algorithm computes the clusters scaling factor S_f since each cluster represents a set of players of having a specific screen resolution r_{p_i}. Based on the computed S_f, the algorithm computes the initial optimal bitrate level $l^{ini,p_i}_{cl_j}$ for each player p_i in cluster cl_j (lines 2-3). Next, for each player p_i within cluster cl_j, the algorithm finds $l^m_{p_i}$ as the higher bitrate level less than or equal to l^{ini,p_i} to be stored in V_1 lines (4-7). Lines (8-15) compute the required bandwidth for each player to stream with the next bitrate $l^{m+1}_{p_i}$. The amount of unused bandwidth W^{remin} from the initial allocation is calculated in line (16). Lines (17-23) fairly redistribute the unused bandwidth among players in a way that insures a fair QoE distribution and efficient resource utilisation. Based on the value of α the algorithm defines the lower bound for the selected bitrates in lines (24-25). The algorithm ends with sending the minimum and maximum bitrate level for each player.

V. EXPERIMENTAL SETUP

The SDN architecture proposed in Section III was implemented in a testbed environment in order to evaluate the performance of BBGDASH. In this section, we explain the testbed and methodology used for the experiment.

A. Evaluation Testbed

The set of experiments were conducted on the proposed SDN-based solution, which was implemented on three virtual machines (VMs) running Linux (Ubuntu V16.04 LTS) as shown in Figure 2. Our testbed is divided into three planes: data plane, control plane, and application plane. The data plane is implemented on Mininet[1] V2.3 network emulator and it consists of three DASH users (U1, U2, and U3) and two OpenFlow switches. The control plane is implemented using Ryu SDN controller [26] and employs the OpenFlow v1.3 protocol as the southbound interface to apply the required QoS configurations and to pull network statistics periodically. Also, RESTful API is used as the northbound interface to provide communication between the Ryu controller and the application plane which hosts dash.js [27] based players running on Google-Chrome. In addition, we attached an Apache server to the Mininet network to provide the video access for the DASH players. Furthermore, in order to provide different content in the video server, we used Big Buck Bunny video with 600 seconds long and encoded using FFmpeg at 4 different resolutions (360p, 480p, 720p, and 1080p) with varying bitrate representations, then each bitrate has segmented into set of 6 seconds chunk using GPAC MP4Box [28].

We used the sflow agent deployed in the OpenVSwitch (OVS) to estimate the available bandwidth. At the client side, the dash.js player [27], was extended by building an HTTP communication channel with the database server (MySQL V5.7) to report the KQIs and the content information in a real time manner. Further, the database server provides an access for the SDN applications to inquire the QoE related parameters and configure the available network resources based on the acquired information.

B. Experiment Design

The implemented testbed was used to provide an access for three DASH players that are sharing a common access network for streaming the video. This replicates the scenario of a moderate congested residential network with multiple users aiming to stream videos concurrently [29]. In this scenario, the first DASH player is requesting a video encoded at 8 bitrate levels L1 = {128, 255, 320, 500, 780, 1000, 1200, 1460} Kbps, the second player is requesting a video encoded at 13 bitrate levels L2 = {128, 255, 320, 500, 780, 1000, 1200, 1460, 2000, 2400, 2900, 3300, 3800} Kbps, while the third player is requesting a video encoded into 11 bitrate levels L3 = {128, 217, 373, 573, 779, 1200, 1460, 2000, 2900, 3500, 3800} Kbps.The bitrate levels were selected based on the DASH dataset [30]. In order to create a shared bottleneck between

[1] http://mininet.org/

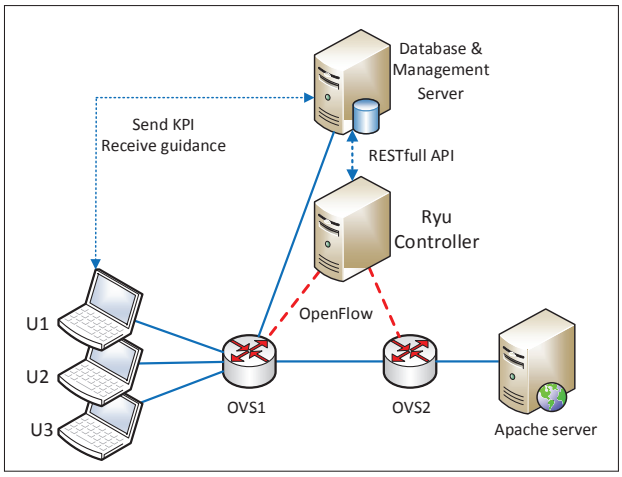


Fig. 2: Experimental Testbed

players, the link between OVS1-OVS2 has been shaped to 7 Mbps using tc tool for bandwidth shaping. The evaluation included a set of four experiments.

***First Experiment*:** The goal of the first experiment was to evaluate the stability of the received video and the efficiency of utilising the available resources when multiple DASH players stream video without application/network-assistance. In this experiment, each player relies only on the local ABR algorithm to adapt the quality of the received video to the network condition.

***Second Experiment*:** The second experiment replicates the work of Kleinrouweler et al. [17], which enabled an application/network assistance approach to provide a stable video delivery. The purpose of the second experiment is to show the feasibility of allocating the available resources equally without considering device or video specifications.

***Third Experiment*:** In the third experiment, the proposed network assistance approach for the DASH streaming was applied to provide each player with maximum allowed bitrate level that is calculated based on the current mix of clients and network conditions. This experiment investigates how the proposed solution can outperform the first and the second approaches when different players request videos with varying bitrate representations. This experiment also discloses the efficiency of guiding dash players with only the maximum allowed bitrate level and without bounding the requested bitrate levels.

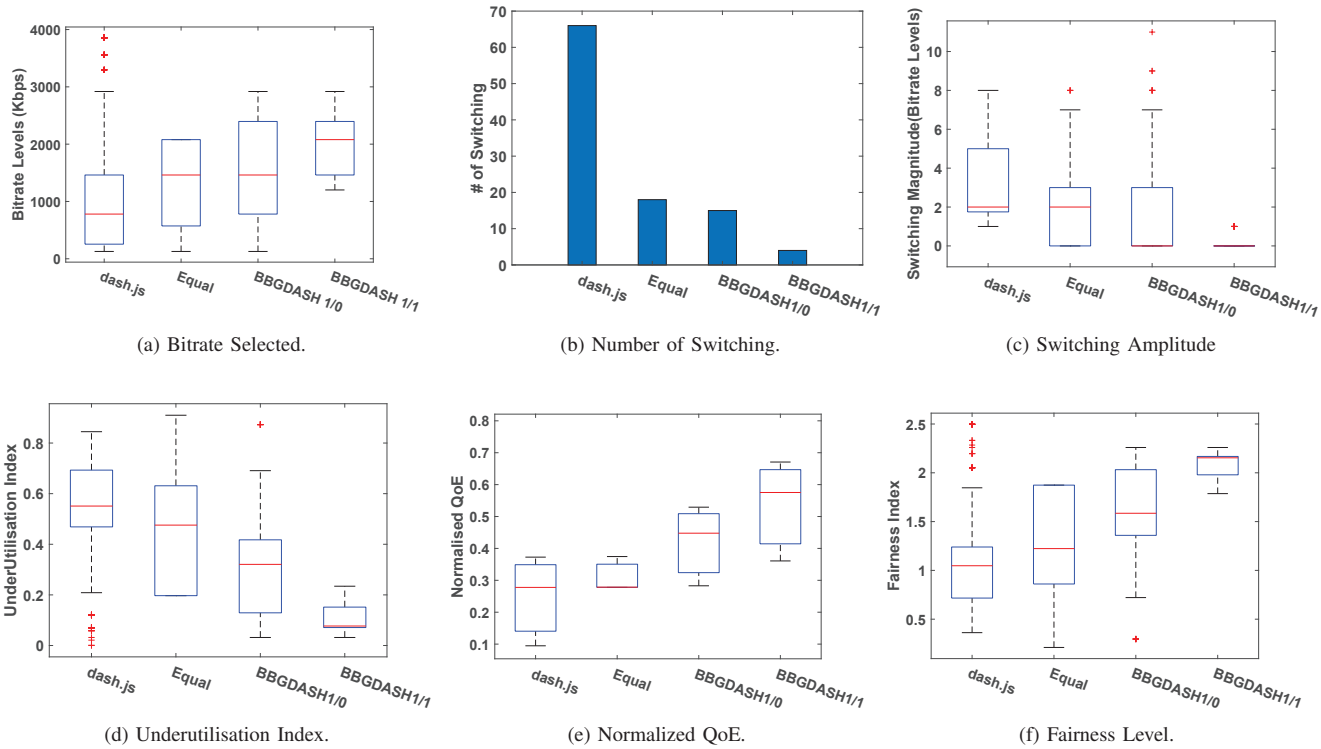
***Fourth Experiment*:** A novel bitrate guidance approach was investigated in the fourth experiment, in which each player receives the maximum and the minimum allowed bitrate levels that define a range of the requested bitrate. The aim of this experiment is to investigate whether bounding the interval of bitrate values can have additional benefits in comparison with specifying only the upper boundary.

In order to compare the performance of four approaches, we adopted a set of metrics [31] [32] [22] [33] for measuring the stability of the received video, the efficiency of utilising the available resources, the perceived QoE, and the fairness level among different players. For the purpose of investigating the stability of the received video, the number and the amplitude of the bitrate switching [31] were used as performance metrics. The Underutilisation index metric, as introduced in [32], was adopted for measuring the efficiency of the proposed approach for utilising the available network resources. The received QoE was calculated using the model presented in [22], which based on the four main QoE metrics (i.e. average bitrate, bitrate switching, video stalling, initial delay) for calculating the perceived QoE. Further, Max ratio [33] was used in our evaluation to examine the fairness level among player.

VI. EXPERIMENTAL RESULTS

This section shows the potential of the proposed solution by comparing the obtained results from the four experiments. The evaluation starts with comparing the average bitrate of all users. Figure 3a demonstrates the stability of the received video in term of average bitrate for the different approaches. As the figure shows, without application/network assistance it is not efficient to provide a stable video delivery when multiple players share a common bottleneck. This inefficiency comes from the fact that local ABR algorithms adapt their quality in a selfish way, trying to stream using the highest video quality. This greedy behaviour results in frequently switching to the lowest bitrate level to avoid stalling. Applying the application/network assistance approach of Kleinrouweler et al. [17] for guiding the users to identify the maximum allowed bitrate level results with boosting the bitrate level of the streamed video. Figure 3a shows that the median bitrate level of all users went up from 780 Kbps without assistance to 1462 Kbps with the equally allocated upper bound guidance [17]. However, as the different players may have different characteristics (i.e. screen resolution) and they may request videos with varying bitrate representations. Therefore, treating the users equally based on Kleinrouweler et al. [17] approach could lead to inefficient resources allocation and poor QoE distribution among different players. Figure 3a depicts the gain of applying the proposed algorithm based on the upper bound guidance (BBGDASH1/0) in term of maximising the received bitrate.

Albeit the upper bound guidance approaches seem to improve the stability and the quality of the received video via guiding each player to find the maximum allowed bitrate level that should be requested, there is no guarantee for the players that they can actually reach the recommended bitrate. Figure 3a reveals that based on the upper bound guidance approaches the average bitrate for all players is less than the recommended bitrate by the agent. Further, Figure3b indicates that some players still experience bitrate switching despite deploying the guidance approach. This leads to the conclusion that sending only the maximum allowed bitrate level is not always efficient to achieve a stable video delivery and efficient resources utilisation. In order to solve this issue, we adjust the bitrate guidance scheme by sending minimum and maximum bounds that should be requested. Figure3a also shows that BBGDASH 1/1 boosts the bitrate level of the received video



(a) Bitrate Selected. (b) Number of Switching. (c) Switching Amplitude

(d) Underutilisation Index. (e) Normalized QoE. (f) Fairness Level.

Fig. 3: Experimental Results in Four Experiments.

to (M=2080kbps), compared to (M=1462 kbps) with the upper bound bitrate guidance.

Regarding the stability of the delivered video, Figures 3b and Figure 3c show the stability of the received video for the four approaches in terms of the number and the amplitude of the bitrate switches respectively. In Figure3b, the guidance based approaches outperform the unassisted dash.js in terms of stability. The total number of switching drops from 66 to 18 for equal allocation upper bound based guidance and to 15 for the BBGDASH 1/0. While BBGDASH 1/1 provides more stability by reducing the number of switching to 4. Likewise, the amplitude of bitrate switching has another effect on the user experience, as a switching event with a low amplitude has a negligible impact on degrading the user's watching experience. Figure 3c compares the amplitude of bitrate switching for the four approaches. It is clear that (BBGDASH 1/1) can reduce the amplitude of the switching, as the player should not switch to a bitrate level less than the minimum allowed bitrate level while the buffer level is more than minimum threshold. Figure 3c shows that the maximum switching amplitude value of BBGDASH 1/1 has the value of 1, compared to 7 for the upper bound guidance and 8 for the unassisted dash.js.

Resource utilisation for the different approaches is measured based on the underutilisation index. Figure 3d, depicts the value of the underutilisation index for the examined approaches. It is clear that the BBGDASH 1/1 outperforms other approaches in term of utilising the available resources with a low value of the underutilisation index equal to (M=0.07). In contrast with other approaches performed inefficient resource utilisation with a high value of underutilisation index equal to (M=0.55) for unassisted dash.js, (M=0.47) for the upper-bound assisted dash.js, and (M=0.32) for BBGDASH 1/0.

The main objective of BBGDASH is to provide a scalable solution to fairly maximised the perceived QoE and efficiently utilise the available resources. As the video contents are available in the CDN with varied maximum levels, therefore equally QoE allocation is not always efficient. For this regard, Figure3e, and 3f demonstrate the evaluation of the proposed approach against others in term of perceived N-QoE and the distributed fairness among the different DASH players. As depicted in Figure3e, deploying BBGDASH 1/1 result with boosting the average perceived QoE to 0.6 from 0.27, 0.28 , and 0.45 with dash.js, Equal, and BBGDASH 1/0 respectively. This value however represents the median QoE level among the different players requesting videos with disparate maximum bitrate levels as depicted in section V .

A similar improvement is also visible when comparing the achieved fairness within the compared approaches, as shown in Figure3f . It reveals that BBGDASH 1/1 increased the average value of the achieved fairness to 2.1 compared to 1, 1.2, and 1.5 for dash.js, Equal, and BBGDASH 1/0 respectively. The main reason behind this improvement is BBGDASH 1/1 utilises the remaining bandwidth for boosting the quality of other players and bound the range of the requested bitrate

levels, which ends with maximising the allocating QoE among players.

VII. Conclusion

When using HTTP adaptive streaming, client-driven HAS applications optimise the adaptation decision locally without coordination between each other, which may result with an unfair QoE among the existing clients. Solutions like QoE centric approaches may harm the principle of HAS and generate scalability issues, as adaptation decision is moved to a specific element. To overcome the specific limitations of each approach, we introduce a hybrid guidance mechanism that provides guidance to the client without moving the whole control logic to an additional entity nor relying purely on the client side decision. Results for experimental evaluation of a typical video streaming scenario using a proof-of-concept implementation demonstrate the potential of the proposed approach compared to existing purely client-based solutions. BBGDASH increases network resource utilisation up to (38%), and QoE up to (24%).

References

[1] "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 20162021 White Paper - Cisco."

[2] I. Ahmed and A. Petlund, "Analysing user satisfaction in next generation networks for multimedia multicast transmission," in *2015 Seventh International Workshop on Quality of Multimedia Experience (QoMEX)*, pp. 1–6, IEEE, 5 2015.

[3] "ISO/IEC 23009-1:2014 - Information technology – Dynamic adaptive streaming over HTTP (DASH) – Part 1: Media presentation description and segment formats."

[4] S. Akhshabi, L. Anantakrishnan, A. C. Begen, and C. Dovrolis, "What Happens When HTTP Adaptive Streaming Players Compete for Bandwidth," *Proc. ACM Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV'12)*, p. 6, 2012.

[5] E. Thomas, M. van Deventer, T. Stockhammer, A. C. Begen, and J. Famaey, "Enhancing MPEG DASH Performance via Server and Network Assistance," *SMPTE Motion Imaging Journal*, vol. 126, no. 1, pp. 22–27, 2017.

[6] D. Kreutz, F. M. V. Ramos, P. Esteves Verissimo, C. Esteve Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-Defined Networking: A Comprehensive Survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2015.

[7] A. Bentaleb, B. Taani, A. C. Begen, C. Timmerer, and R. Zimmermann, "A Survey on Bitrate Adaptation Schemes for Streaming Media over HTTP," *IEEE Communications Surveys and Tutorials*, vol. PP, no. X, p. 1, 2018.

[8] C. Liu, "Rate Adaptation for Adaptive HTTP Streaming," pp. 169–174, 2011.

[9] J. Jiang, V. Sekar, and H. Zhang, "Improving fairness, efficiency, and stability in HTTP-based adaptive video streaming with festive," *IEEE/ACM Transactions on Networking*, vol. 22, no. 1, pp. 326–340, 2014.

[10] Y. Sun, X. Yin, J. Jiang, V. Sekar, F. Lin, N. Wang, T. Liu, and B. Sinopoli, "CS2P: Improving Video Bitrate Selection and Adaptation with Data-Driven Throughput Prediction," *Proceedings of the 2016 conference on ACM SIGCOMM 2016 Conference - SIGCOMM '16*, pp. 272–285, 2016.

[11] 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," 2014.

[12] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman, "BOLA: Near-optimal bitrate adaptation for online videos," *Proceedings - IEEE INFOCOM*, vol. 2016-July, 2016.

[13] Z. Li, X. Zhu, J. Gahm, R. Pan, H. Hu, A. C. Begen, and D. Oran, "Probe and adapt: Rate adaptation for HTTP video streaming at scale," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 4, pp. 719–733, 2014.

[14] A. Bentaleb, A. C. Begen, S. Harous, and R. Zimmermann, "Want to play DASH?," *Proceedings of the 9th ACM Multimedia Systems Conference on - MMSys '18*, pp. 13–26, 2018.

[15] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, *Algorithmic game theory*. Cambridge University Press, 2007.

[16] R. K. P. Mok, X. Luo, E. W. W. Chan, and R. K. C. Chang, "QDASH : A QoE-aware DASH system," *MMSys 2012*, pp. 11–22, 2012.

[17] J. W. Kleinrouweler, S. Cabrero, and P. Cesar, "Delivering Stable High-Quality Video: An SDN Architecture with DASH Assisting Network Elements," 2016.

[18] A. Bentaleb, A. C. Begen, and R. Zimmermann, "SDNDASH: Improving QoE of HTTP Adaptive Streaming Using Software Defined Networking," *Proceedings of the 2016 ACM on Multimedia Conference - MM '16*, pp. 1296–1305, 2016.

[19] P. Georgopoulos, Y. Elkhatib, M. Broadbent, M. Mu, and N. Race, "Towards network-wide QoE fairness using openflow-assisted adaptive video streaming," *2013 5th ACM SIGCOMM Workshop on Future Human-Centric Multimedia Networking, FhMN 2013*, pp. 15–20, 2013.

[20] S. Petrangeli, J. Famaey, M. Claeys, S. Latré, and F. De Turck, "QoE-Driven Rate Adaptation Heuristic for Fair Adaptive Video Streaming," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 12, no. 2, pp. 1–24, 2015.

[21] Z. Akhtar, Y. S. Nam, R. Govindan, S. Rao, J. Chen, E. Katz-Bassett, B. Ribeiro, J. Zhan, and H. Zhang, "Oboe," *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication - SIGCOMM '18*, vol. 1, pp. 44–58, 2018.

[22] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A Control-Theoretic Approach for Dynamic Adaptive Video Streaming over HTTP," *Sigcomm 2015*, pp. 325–338, 2015.

[23] M. H. Pinson and S. Wolf, "A new standardized method for objectively measuring video quality," *IEEE Transactions on broadcasting*, vol. 50, no. 3, pp. 312–322, 2004.

[24] W. Zhou, L. Li, M. Luo, and W. Chou, "Rest api design patterns for sdn northbound api," in *Advanced Information Networking and Applications Workshops (WAINA), 2014 28th International Conference on*, pp. 358–365, IEEE, 2014.

[25] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[26] "RYU SDN Framework. http://osrg.github.io/ryu/."

[27] "DASH Industry Forum. http://www.dashif.org/.."

[28] "GPAC:Multimedia Open Source Project," *What's New*, pp. 0–60.

[29] "Q1 2017 State of the Internet - Connectivity Report — Akamai."

[30] "DASH External Dataset. http://www-itec.uni-klu.ac. at/ftp/datasets/-DASHDataset2014/.."

[31] 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 Communication Surveys & Tutorials*, vol. 17, no. 1, pp. 469–492, 2015.

[32] A. Bentaleb, A. C. Begen, R. Zimmermann, and S. Harous, "SDNHAS: An SDN-enabled architecture to optimize QoE in HTTP adaptive streaming," *IEEE Transactions on Multimedia*, vol. 19, no. 10, pp. 2136–2151, 2017.

[33] T. Lan, D. Kao, M. Chiang, and A. Sabharwal, *An axiomatic theory of fairness in network resource allocation*. IEEE, 2010.