Design and Performance Evaluation of Network-assisted Control Strategies for HTTP Adaptive Streaming

Design and Performance Evaluation of Network-assisted Control Strategies for HTTP Adaptive Streaming

GIUSEPPE COFANO, Politecnico di Bari, Italy
LUCA DE CICCO, Politecnico di Bari, Italy
THOMAS ZINNER, University of Würzburg, Germany
ANH NGUYEN-NGOC, University of Würzburg, Germany
PHUOC TRAN-GIA, University of Würzburg, Germany
SAVERIO MASCOLO, Politecnico di Bari, Italy

This paper investigates several network-assisted streaming approaches which rely on active cooperation between video streaming applications and the network. We build a Video Control Plane which enforces Video Quality Fairness among concurrent video flows generated by heterogeneous client devices. To this purpose, a max-min fairness optimization problem is solved at run-time. We compare two approaches to actuate the optimal solution in an SDN network: the first one allocating network bandwidth slices to video flows, the second one guiding video players in the video bitrate selection. We assess performance through several QoE-related metrics, such as Video Quality Fairness, video quality, and switching frequency. The impact of client-side adaptation algorithms is also investigated.

CCS Concepts: • **Networks** → **Network management**; *Network control algorithms*; Network experimentation;

General Terms: Design, Experiments, Control

Additional Key Words and Phrases: Adaptive Video Streaming, DASH, network-assistance, Control Plane, Quality of Experience, Fairness

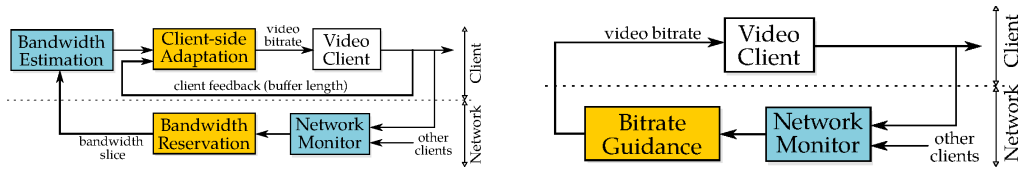
ACM Reference format:
Giuseppe Cofano, Luca De Cicco, Thomas Zinner, Anh Nguyen-Ngoc, Phuoc Tran-Gia, and Saverio Mascolo. 2017. Design and Performance Evaluation of Network-assisted Control Strategies for HTTP Adaptive Streaming. *ACM Trans. Multimedia Comput. Commun. Appl.* 1, 1, Article 1 (April 2017), 23 pages.
DOI: 10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The amount of video content that is being distributed over the Internet is increasing thanks to the wide diffusion of Smart TVs, tablets, and smartphones. Today, video providers leverage the HTTP infrastructure made of servers and CDNs to scale their video delivery system and reach their users. However, scalability is not the only concern for video providers. User-centric objectives such as service costs or Quality of Experience (QoE) significantly impact user engagement. Accordingly, video providers have to satisfy user expectations to avoid user abandonment and the resulting

This work has been partially supported by the Italian Ministry of Education, Universities and Research (MIUR) through the MAIVISTO project (PAC02L1_00061) and by the Apulia Region (Italy) through the Future in Research project no. ACYBEH5. This work has been also partly funded by Deutsche Forschungsgemeinschaft (DFG) under grants ZI 1334/2-1 and TR257/43-1.

DOI: 10.1145/nnnnnnn.nnnnnnn



(a) Bandwidth Reservation (b) Bitrate Guidance

Fig. 1. Network-assisted approaches for adaptive video streaming

revenues losses [27]. One of the main influence factors, which thus has to be improved, is the QoE [4, 27, 33].

Video providers currently rely on the HTTP adaptive streaming (HAS) approach, a technique allowing video quality adaptation on short time scales, to deliver videos to the users. Video clients are equipped with controllers allowing to autonomously change the video bitrate to improve the QoE. These HTTP adaptive streaming algorithms are designed to avoid playback interruptions due to buffer underruns and to maximize the video bitrate – possibly matching the end-to-end bandwidth – while containing the video bitrate switching frequency [4].

The simultaneous presence of several adaptive video streaming flows transmitted via a shared bottleneck link results in a fair bandwidth distribution among the involved flows. However, QoE-relevant influence factors such as the device capabilities or the user context are not taken into account by QoS-based distribution. For instance, users with small screens are served with the same video bitrate as users with large screens, resulting either in bad QoE for users with large screens or in wasted network resources due to the over provisioning of video quality. The resources are fairly shared with respect to the QoS parameters, but not with respect to the user's QoE [14]. To overcome this problem, an interaction between video and network provider may prove beneficial.

A video control plane can leverage the exchanged information to enforce network-assisted streaming strategies. A standard signaling plane is required to enable active cooperation between network elements, such as, f.i., the one proposed by Server And Network Assisted DASH (SAND DASH)[1]. Such architectures allow a network element to trigger a control mechanism such as quality adaptation, flow prioritization or bandwidth reservation, based on network state and client context. Software Defined Networking (SDN) is a viable technology to implement such mechanisms due to the presence of a centralized control element, which is particularly beneficial in the presence of complex topologies [40].

Our work provides a broad investigation of the design space of video control planes by studying and experimentally comparing the performance of three classes of network-assisted strategies. The *Bandwidth Reservation* assigns a bandwidth slice to a video flow (or a group of video flows). Two nested control loops are established as shown in Fig. 1 (a): the *outer control loop* is executed in the network and sets the bandwidth slice, whereas the *inner control loop*, running at the client, autonomously selects the video bitrate based on video client feedback and bandwidth estimates. In this paper, we consider several bandwidth reservation strategies and several client-side adaptation algorithms to assess interactions between the two control loops. Additionally, we take into account the constraints imposed by the capabilities of the current hardware (i.e. limited number of configurable bandwidth slices) by proposing mechanisms to address this issue. The second category, shown in Fig. 1 (b), is named *Bitrate Guidance*: when this approach is employed, a centralized

[1]https://tools.ietf.org/id/draft-begen-webpush-dash-reqs-00.txt

algorithm running in a network element computes the video bitrate that is then enforced by the video client. Finally, we take into account *hybrid strategies* combining Bandwidth Reservation and Bitrate Guidance.

To experimentally compare the performances obtainable with these approaches, we have implemented a testbed in which network-assisted strategies enforce a management policy to maximize Video Quality Fairness (VQF). The testbed is built using an SDN controller and several concurrent video sessions are generated using TAPAS [11]. In the considered network scenario the bottleneck is located at the SDN switch where the VCP enforces the management policy. The bottleneck link can be the one connecting the ISP Access Network to the Home Router or the egress congested link of a CDN. Moreover, since this is the first paper comparing network-assisted approaches for video delivery, the results obtained in this paper can serve as a starting point to further study network-assisted strategies over general topologies.

2 RELATED WORK

In the following, we separately review related work on (i) Quality of Experience for adaptive video streaming, (ii) the use of network-assisted approaches for the delivery of video content.

2.1 QoE of Video Streaming

The concept of Quality of Experience (QoE) explicitly refers to the user-perceived quality by relying on subjective criteria. For classical HTTP video streaming, the essential influence factors on QoE are initial delay and stalling due to buffering [15, 17]. HAS introduces quality adaptations during video playback as additional influence factors and allows to trade-off waiting times and quality switching frequency. The main perceptual QoE influence factors for HAS can be grouped in waiting times, video quality adaptation, and video quality switching frequency.

Waiting Times. According to [15], waiting times can be classified into waiting times before the video playback is started and interruptions during video playback. It is also shown that, in general, initial delays are perceived less disturbing than playback interruptions. Furthermore, subjective evaluation results in [19] showed that, on equal terms of the overall stalling time, the larger the number of stalling events the more detrimental the effect on the QoE. Accordingly, the number of playback interruptions, as well as the total stalling time, should be reduced, even at the expense of other factors such as initial delays or video quality adaptation [33].

Video Quality Adaptation. Video quality adaptation can be performed in three dimensions, namely in the temporal, spatial, and quality dimension. Authors of [33] point out that reducing the video quality too much in any of these dimensions leads to a bad QoE. Further, an adaptation in multiple dimensions is perceived better than a single dimension adaptation. The dimensions to adapt, however, depend on the content type and its characteristics. The video service provider performs the selection of content representations when preparing the content. It is worth noting that current HAS systems typically rely on adaptations in the quality dimension and tend to neglect the other dimensions.

Video Quality Switches. Besides waiting times, another factor influencing the QoE are the video quality switches during playback. In particular, the effect of quality switches can be characterized based on the period, amplitude, and the type of video content [30]. The effect of the switching period is perceived as annoying in the case switching periods are lower than 1 s. Although the effect of the video quality switching period is the dominating one, it has been shown that small amplitudes are not detectable by users. Finally, content plays a significant role in the case the quality switch involves either the temporal or the spatial dimension.

Authors of [18] investigate the influence factors of the adaptation parameters for HTTP adaptive streaming with two quality layers. The results confirm the high impact of the switching amplitude between two played back representations, and that recency effects are negligible if more than two switches occur. The time on highest video quality layer has a significant impact on the QoE, and the number of quality switches can be neglected. These investigations are further extended in [34] by introducing an intermediate layer and highlighting its impact on the QoE. Thereby, the position of the intermediate layer has no significant impact on the QoE. Further, a positive effect of the intermediate layer and a negative effect of the low layer on the QoE were visible. Hence, the quality of each layer and the playback time spent on each layer are essential QoE parameters. The authors show that the average SSIM value for the played back video quality has a high correlation with the MOS values of constant profiles, i.e., SSIM is a good predictor for the MOS.

Summary. Based on the discussed work, the following conclusions can be derived: (i) video stallings, either initial or during playback, have a high impact on the overall video quality and should be avoided; (ii) as long as no flickering effects occur (switches between qualities on short timescales) the number of quality switches has a minor impact on the overall QoE; (iii) in the absence of flickering, averaging the SSIM values for the played back qualities on a per-frame basis yields to a good QoE estimation.

2.2 Network-assisted Approaches

Bandwidth Reservation. The virtualization of the ISPs access infrastructure using open APIs supported through SDN is proposed in [35]. Content providers can programmatically provision capacity to user devices to guarantee QoE by employing network resources slicing. Moreover, an algorithm is proposed for optimally allocating network resources, leveraging bulk transfer time elasticity and access path space diversity. In [22] an SDN-based application-aware bandwidth allocation approach is used to maximize the QoE of YouTube flows. In [7] a control architecture and a reference implementation of a network control plane for video flows are proposed. The reference implementation is evaluated through numerical simulations. In [28] a new QoE metric is introduced, which takes into account the video resolution and the distance of the user from the screen. Based on this metric, a QoE max-min fairness problem is formulated to enforce a per-flow bandwidth allocation in the Home Network.

Bitrate Guidance. In [14] an OpenFlow-assisted QoE Fairness Framework is proposed to fairly maximize the QoE of multiple competing video clients in a Home Access Network. Authors provide a proof-of-concept implementation considering a small number of concurrent flows. In [26] authors propose to place a HTTP proxy server between client and server (in the gateway or any other network device) to drive the bitrate adaptation of the players by using the *URL rewriting* technique. An analytical model in the form of a Markov process is employed at the proxy to compute the bitrate for each player. In [31] a rate adaptation algorithm is proposed with the goal of providing fairness in a multi-client setting. To the purpose, authors propose to employ an in-network system of coordination proxies to facilitate fair resource sharing among clients. The in-network components provide the clients with feedback, whereas clients perform the bitrate adaptation. The performance evaluation is carried out through ns-2 simulations.

Other Network-Assisted Approaches . Authors of [25] propose two bitrate adaptation assistance mechanisms to overcome performance losses due to ON-OFF patterns: the first, explicitly signaling target bitrates to DASH players and the second performing dynamic traffic control in the network. The paper shows that both the mechanisms improve the streaming performances, but users experience a high and stable video quality only when they are used simultaneously. In [23] a data-driven

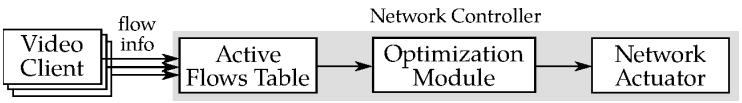


Fig. 2. A block diagram of the control system

prediction model is designed to improve bitrate selection mechanisms. Combining a real-world pilot deployment with a trace-driven analysis, authors show that the proposed prediction model leads to significant improvements, both in terms of stalling due to buffering and average video bitrate. Authors of [13] propose C3, a centralized control platform designed to optimize video delivery. The platform enables (i) per-CDN real-time monitoring of the delivered video QoE, (ii) the prediction of expected performance and (iii) the selection of the CDN and the video bitrate. C3 enforces the bitrate selection by using a centralized *Decision Layer*, which is made aware of the transport networks performance by a prediction algorithm located in an upper layer.

3 THE VIDEO CONTROL PLANE

This Section describes the Video Control Plane (VCP) that we employ to enforce a Video Quality management policy. We have considered a single bottleneck scenario in which resource allocation is enforced at the bottleneck link. The VCP can be used in any of the networks involved in the video delivery, f.i., the link connecting the ISP Access Network to the Home Router or the egress congested link of a CDN [29].

3.1 Control System Architecture

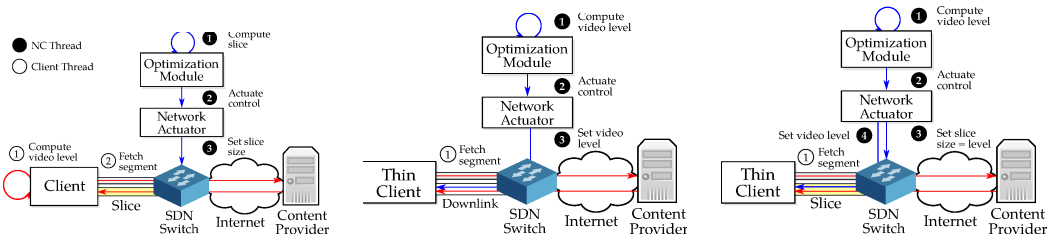
Fig. 2 shows a block diagram of the overall control system that builds on two components: the *Network Controller* (NC) and the *video clients*.

The Network Controller. The NC runs on top of the SDN controller and undertakes the following tasks: 1) it creates and manages bandwidth slices implemented through dedicated queues on the network interfaces; 2) it handles a bidirectional communication pipe with the video clients. The NC consists of three components: the *Active Flows Table*, the *Optimization Module*, and the *Network Actuator*. The Active Flows Table stores information of the currently active video sessions. Each video client provides such information at the beginning of the video session. The Optimization Module takes as input the information provided by the table and periodically computes, each T_s seconds, the bitrate assignment according to the Video Quality management policy. Specifically, the algorithm assigns a bitrate (or bandwidth) to each active video session. Finally, the Network Actuator is the component enforcing the computed bitrates (or bandwidth). The actuation mode depends on the adopted network-assisted approach as described in Section 3.2.

Video Client. The clients undertake the following tasks: 1) set-up/teardown of the video session by sending messages to the NC; 2) download the corresponding segments for the bitrate computed by the bitrate adaptation algorithm.

3.2 Network-assisted Streaming Approaches

In this work we consider three network-assisted strategies to provide service differentiation to concurrent video flows: 1) the *Bandwidth Reservation* approach (BR), the *Bitrate Guidance* approach (BG) and the approach combining *Bitrate Guidance*, and *Bandwidth Reservation* (BG+BR). The control system shown in Fig. 2 can implement such approaches by combining two parallel and independent threads, as depicted in Fig. 3: the *Client thread* and the *NC thread*.



(a) Bandwidth Reservation (b) Bitrate Guidance (c) Bitrate Guidance and Bandwidth Reservation

Fig. 3. The considered network-assisted approaches

Bandwidth Reservation (BR). When this approach is used, the NC reserves dedicated bandwidth slices to the video flows. The NC does not send any explicit information to the video client which independently selects the video bitrate according to its client-side adaptation algorithm. Fig. 3 (a) shows how to implement this approach. The NC thread is composed of three actions repeated in a cycle: ❶ the *Optimization Module* computes the bandwidth slice assignment based on the management policy; ❷ the *Network Actuator* receives the computed bandwidth slice and ❸ creates or updates the dedicated slice for the flow (or the group of flows). The Client thread is made of two actions consecutively run on each video segment download: ① the video bitrate is selected by the client according to its adaptation algorithm; ② the segment is retrieved from the Content Provider.

Bitrate Guidance (BG). In this case, the NC computes the optimal video bitrate according to the Video Quality management policy. Then, the NC sends the computed values to the video clients that download the corresponding video segments. It is important to notice that, when using this approach, all the video flows share the same bandwidth slice. The client shapes the download rate to match the selected bitrate, thus providing service differentiation to the flows sharing the slice. Fig. 3 (b) shows the implementation of this approach. The first two actions of the NC thread ❶ and ❷ are exactly equivalent to the ones executed in the BR approach. The third action ❸ is different: instead of creating bandwidth slices on the network interfaces, the Network Actuator communicates the computed bitrates to the video clients. The Client thread only runs action ①, i.e. it downloads the next video segment based on the video bitrate set by the NC. Accordingly, Fig. 3 (b) labels such a video client as a *Thin Client*. To make this approach scalable, clients do not send any feedback information to the NC. As a consequence, the NC is not aware if the playout buffer is draining and the client is required to download a lower bitrate to fill it again quickly. For this reason, when the playout buffer gets below a threshold, a safety mechanism is activated and the video bitrate is selected by the client ignoring the guidance of the NC.

Bitrate Guidance and Bandwidth Reservation (BG + BR). This approach is enforced by combining the two strategies described above. In particular, the third action of the NC thread is split into two sub-actions: 1) the bandwidth reservation in the network and 2) the bitrate guidance. The client thread is again limited to performing segment downloads, i.e. the client can be considered as a Thin Client exactly as in the case of the BG approach.

3.3 Client-side Adaptation

Client-side algorithms select the video bitrate from a discrete set at each segment download based on parameters such as the estimated bandwidth and the playout buffer length. Such algorithms aim

at improving the QoE by: 1) avoiding rebuffering events; 2) maximizing the video bitrate; 3) keeping the number of video bitrate switches as low as possible.

In general, it is difficult to achieve these goals simultaneously and some trade-offs have to be made. An appropriate classification to our investigation is the following, which makes the distinction between *rate-based* and *level-based* approaches [8]. The first one requires the algorithm to insert pauses (OFF periods) between the downloads of consecutive segments to make the download rate match the selected video bitrate on average. As a consequence, this approach sacrifices bandwidth utilization to reduce video level switches. The second approach downloads the segments back-to-back and the playout buffer is prevented from growing by throttling the video bitrate according to a control law. This approach achieves the full link utilization at the price of a higher number of video level switches [8].

This paper investigates the interactions between the client-side control loop and the network control loop in the case of three algorithms: *Conventional* [39], *PANDA* [39], and *Elastic* [10]. We have decided to consider these algorithms to cover both control approaches.

Conventional. It is a simple *rate-based* algorithm selecting the video bitrate based on bandwidth estimates [39]. In a nutshell, the k-th controller output is equal to a filtered version y_k of the estimated bandwidth x_k. In particular, the k-th estimated bandwidth sample is computed as $x_k = \tau r_{k-1}/T_{k-1}$ where r_{k-1} is the video level rate of the last downloaded segment, τ is the segment duration, and T_{k-1} is the download time. Then, x_k passes through a first-order low-pass filter giving the filtered bandwidth sample y_k computed as $y_k = y_{k-1} - T_{k-1}\alpha(y_{k-1} - x_k)$ where $\alpha > 0$ is the filter parameter. The video level index of the next video segment to be downloaded is a quantized version of y_k (see [39] for more details). Finally, when the system is in *steady state,* i.e., the buffering phase is completed, the controller sets the OFF period length equal to $\max(\tau - T_{k-1}, 0)$. We have considered this algorithm to check if the Bandwidth Reservation strategy performs well even when users employ a very simple client-side adaptation algorithm.

PANDA. It is a *rate-based* algorithm designed to cope with the fairness issues affecting several HAS algorithms [39]. It follows a probe-and-adapt approach, incrementing the bitrate to probe the available bandwidth. In particular, PANDA employs a control mechanism to regulate the OFF periods duration and another control law to adapt the video bitrate. The control law to regulate the OFF periods takes into account the current level of the playout buffer: in a nutshell, when the current playout buffer level is below the playout buffer target, OFF periods are shrunk; conversely OFF periods are increased. Concerning the video bitrate control law, PANDA employs an AIMD probing mechanism similar to the TCP congestion control. The output of the controller y_k, i.e., the video level rate, is additively increased when the last segment download rate x_{k-1} is larger then the previous controller output y_{k-1}. Otherwise, if $x_{k-1} < y_{k-1}$ holds, i.e., the selected video bitrate is higher than the measured download rate, the output is decreased proportionally to $y_{k-1} - x_{k-1}$. Similarly to the *Conventional* controller, PANDA selects the video level index of the next video segment to be downloaded as the output of a quantization function $Q(\cdot)$ having as input y_k (see [39]).

Elastic. It is a *level-based* algorithm employing a feedback control technique known as feedback linearization to control the playout buffer length by varying the video bitrate [10]. The algorithm downloads video segments back to back and sets the video level according to the following control law:

$$l_k = Q\left(\frac{b_k}{1 - k_1 q_k - k_2 q_k^I}\right)$$

where: 1) $l_k \in \mathscr{L}$ is the video level chosen by the controller for the download of the k-th segment, 2) q_k is the measured queue level, 3) q_k^I is the integral of the error $q_T - q_k$, 4) q_T is the queue target, 5) b_k is the bandwidth estimate, 6) $Q : \mathbb{R} \rightarrow \mathscr{L}$ is the quantization function returning the maximum video level $l \in \mathscr{L}$ lower than its input value. To prevent the playout buffer from growing indefinitely, Elastic uses an ON-OFF pattern similar to the one employed by *Conventional* only when the following conditions hold simultaneously: 1) the selected video level l_k is equal to the maximum available level in $\mathscr{L}$; 2) the queue has already grown to a maximum value $q_{\max} \gg q_T$. The OFF periods are inhibited again when at least one of the two conditions above does not hold anymore. It has been shown that Elastic is able to overcome fairness issues affecting the rate-based algorithms. Moreover, since Elastic behaves as a long-lived TCP flow, it avoids the "downward spiral" phenomenon that affects other rate-based algorithms when in the presence of other TCP flows [21].

3.4 The Management Policy

Video distribution platforms require different management policies depending on the application scenario and the employed monetization process. Different policies could consider, for instance, service differentiation based on user classes (premium versus unsubscribed users) or on QoE-related parameters. Before introducing the proposed management policy, it is important to make a clear distinction between *video quality* and *QoE*. The first term refers to metrics only related to the visual quality of the video; the video quality can be assessed with metrics such as SSIM, PSNR, VQM, PEVQ, or MSE (see [32] for a comparison of video quality metrics). On the other hand, the category of QoE-related metrics comprises all the parameters affecting the user experience, including the video quality; other important QoE-related metrics in the case of video streaming are rebuffering ratio, video level switching frequency, start-up latency [33].

Video Quality Fair Allocation. Several papers focusing on the design of video control planes have considered the use of resource allocation based on either video quality [14] or QoE [31], which is indeed more appropriate than fair bandwidth allocation (i.e. QoS fairness) in the context of video delivery.

This work considers, as an example, a management policy aiming at providing fairness to concurrent video streams in terms of *video quality*. A more sophisticated policy could be designed by also taking into account other QoE-related parameters such as rebuffering ratio and video level switches. However, we argue that using only video quality is motivated by two reasons. First, video quality can be computed off-line in the case of VoD. This means that client feedback is not required, which improves scalability. Second, QoE-related metrics are already taken into account by client-side adaptation algorithms. This approach has the merit of decoupling the overall problem in two subproblems, one handled centrally and one handled at the end-points. The interactions between these subproblems are described in detail in [7].

The Optimization Problem. We have considered a simple network composed of a single node, whose egress link is the bottleneck link on which the network-assisted approaches are implemented.

We consider the following scenario using the following notation. N video sessions are active over a channel with capacity C. Each video session $n \in \{1, \ldots, N\}$ streams the video v_n with a client device whose screen resolution is r_n. The video v_n is encoded in several video representations. Each representation is characterized by its bitrate $\bar{l}_i \in \mathscr{L}_n$ and its resolution $\bar{r}_i \in \mathscr{R}_n$. We assume that users do not request video representations with a resolution higher than their screen resolution r_n.

For each video session a utility function $U_n(\cdot)$ can be defined, which associates to each video bitrate in $\mathscr{L}_n$ the corresponding perceived video quality. The next paragraph shows how such

functions are computed. It is worth noting that the utility function depends on the client screen resolution.

We are now ready to formulate the Video Quality Fairness policy as a max-min fairness problem. The issue here is to compute, at each sampling interval and for each active session n, the video bitrate l_n to stream in order to maximize the minimum measured Video Quality over all the video sessions.

Depending on the particular scenario and network-assisted strategy, one of the following optimization problems will be solved. The first, named *Discrete Video Quality fair assignment,* requires the optimal bitrate for each video session to belong to its video level set. This problem can be used to both compute bandwidth slices in the case of the BR approach and to compute video bitrates to guide video clients in the case of the BG approach. The second, the *Continuous Video Quality fair assignment*, in which the bandwidth slice size b_n allocated to the n-th video session can assume any real value between the minimum bitrate and the maximum bitrate of the video level set, i.e., $b_n \in [\min\{\mathscr{L}_n\}, \max\{\mathscr{L}_n\}]$. This optimization problem will be used only in the case of the BR approach to compute bandwidth slices size. In the following, we formulate and briefly discuss the two optimization problems.

Problem 3.1 (Discrete Video Quality fair assignment).

$$\begin{array}{ll} \text{Maximize} & \left[\min_{l_n \in \mathscr{L}_n} U_n(l_n)\right] \\ \text{Subject to} & \sum_{n=1}^{N} l_n \leq C. \end{array} \tag{1}$$

For each active video session n, a bitrate l_n belonging to the video level set $\mathscr{L}_n$ is computed. The constraint imposes that the sum of the bitrates cannot exceed the link capacity.

Problem 3.2 (Continuous Video Quality fair assignment).

$$\begin{array}{ll} \text{Maximize} & \left[\min_{b_n \in \mathbb{R}} \overline{U}_n(b_n)\right] \\ \text{Subject to} & \sum_{n=1}^{N} b_n \leq C, \\ & \min\{\mathscr{L}_n\} \leq b_n \leq \max\{\mathscr{L}_n\} \end{array} \tag{2}$$

Here, the utility function $\overline{U}_n(\cdot)$ is a continuous function mapping the allocated bandwidth b_n to the corresponding video quality.

Both the optimization problems (1) and (2) can be solved with a *progressive filling approach* [5], by starting with all the components in the solution vector being equal to the lowest bitrates and growing all solutions together at the same pace, until either the link capacity limit is hit or all the video sessions have been assigned with the maximum bitrate. In the case of (1), one of the active video sessions is selected at each step and its video level is increased by 1. This procedure requires a criterion to select the next video session to increase at each step. We have resorted to the heuristic of selecting the video session whose level increase maximizes the video quality increment. In the case of (2), instead, a much more efficient approach can be taken due to the fact that the utility functions are strictly monotonically increasing and thus can be inverted: it can be shown that the problem corresponds to finding the root of a univariate equation that can be solved efficiently.

Video Quality Measurement. To solve the optimization problems (1) and (2) we need the mappings $U_n(\cdot)$ and $\overline{U}_n(\cdot)$ for each video session n. To the purpose, a metric to estimate the video quality is needed. We have decided to employ the *Structural SIMilarity* (SSIM) index, an objective reference-based method, to compute the estimate of video quality off-line [37]. Similarly to other metrics available in the literature, the SSIM reflects the subjective user experience only to a certain extent and provides an approximate estimate of the QoE. However, the SSIM has been shown to be a

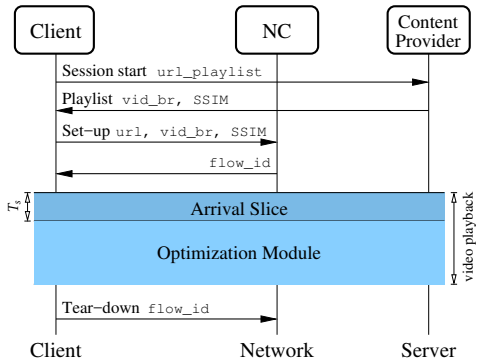


Fig. 4. Video session flow diagram

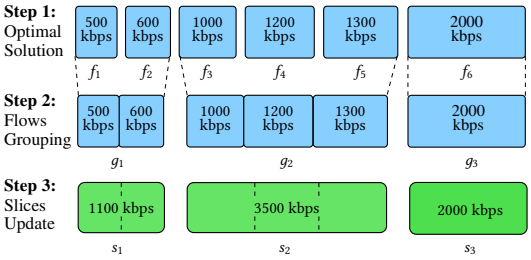


Fig. 5. The Quantized aggregation strategy

valuable tool to obtain an approximation of the user-perceived video quality (see Section 2.1). Notice that, if on one hand the results obtained in this paper depend on the employed QoE metric, on the other hand the described methodology can be applied easily to other QoE metrics.

In the case of videos the SSIM is computed as the average SSIM over all the video frames of a segment.

We define the reference video as the best available video representation of that video clip at the client screen resolution [36]. Thus, given the client resolution, the reference video is chosen from the video level set as the representation with the same resolution and the highest bitrate. Let us consider a video clip encoded into several representations, each of them characterized by a bitrate $\overline{l}_i$ and a resolution $\overline{r}_i$. We denote with $\hat{r}$ the reference resolution. The SSIM of each representation is computed by comparison with the reference video. If the representation to be evaluated has a resolution $\overline{r}_i$ lower than $\hat{r}$, it is upscaled to $\hat{r}$ before being compared. The upscaling is motivated by the fact that the video player also upscales the decoded video to the device resolution[2] when rendering the video during playback.

In the problem (2), where a continuous utility function $\overline{U}_n(\cdot)$ is required, we have employed a linear interpolation between consecutive bitrates to generate a continuous mapping from the discrete one.

4 IMPLEMENTATION

4.1 Video Session Management

Fig. 4 shows the workflow of a video session. A client starts the video session by retrieving the playlist from the video server. We suppose that, in addition to the video level set, the playlist also carries the SSIM values for each video representation computed as shown in Section 3.4. When a video session starts, the client sends the *set-up* message to the NC, which carries the information employed to compute the optimal bitrate distribution. Fig. 4 also shows the information carried by the set-up message: the video content URL, the video level set and its corresponding SSIM extracted from the playlist. The NC stores this information in the Active Flows Table.

The video client starts to download video segments as soon as the set-up message is sent. Since the NC periodically executes the Optimization Module with a sampling time T_s, the video flow cannot be served with differentiated service until the next execution of the Optimization Module.

[2]Our Video Quality Fairness policy differs in this aspect from the one proposed in [14] that makes the restrictive assumption that several representations with different resolutions are available for each bitrate.

To avoid a delayed start-up, the NC assigns the flow to the *Arrival Slice*, which is reserved for newly arrived video sessions. Then, at most after T_s seconds, the Optimization Module is executed, the video flow is removed from the *Arrival Slic*e and served with the differentiated service according to the adopted network-assisted approach.

Finally, when the client decides to terminate the session, it sends a *tear-down* message to the NC, which removes it from the Active Flows Table at the next iteration of the Optimization Module.

4.2 The Flows Aggregation Strategy for BR

The Bandwidth Reservation (BR) approach ideally assigns to each flow one slice whose size is computed by the Optimization Module. However, the number K of available QoS queues on a network interface is usually limited between 4 and 10 [38], which is, in general, much lower than the number N of concurrent video sessions.

Hence, if $K < N$, it is necessary to use a flow aggregation strategy grouping the N video flows into K slices to implement the BR approach with some approximation.

It has to be noticed that rate-based HAS clients employing the ON-OFF traffic pattern [3] may suffer from fairness issues when sharing a bottleneck (or slice) [1]. Unfairness issues affecting this class of HAS clients can be coped in different ways: 1) in the HAS client, by appropriately controlling the idle phases as shown in [39] or by randomizing the scheduling of segment downloads [24]; 2) in the switch or at the server, by means of rate shaping techniques [2, 12, 20]. On the other hand, level-based algorithms, such as f.i. ELASTIC [10], do not insert idle periods between consecutive segment downloads and, as a consequence, are not affected by fairness issues. In any case, we argue that well-designed HAS clients should provide fairness among users sharing a bottleneck. Based on the above, we can make the reasonable assumption that HAS clients are able to fairly share a slice. In such conditions, if flows with similar video bitrate are assigned to the same slice, each of these flows will obtain a bandwidth share close to the one set by the Optimization Module. In the following we describe the two proposed strategies and Section 5 evaluates the impact of the approximation induced by such strategies on the actuation of the optimal solution.

Quantized strategy. A quantization process maps each flow to one of the K slices. Then, each group of flows is assigned with a slice whose size is equal to the sum of the video bitrates belonging to the group.

To explain the proposed strategy, we give an example, shown in Fig. 5. Let us consider the case of $N = 6$ concurrent video flows accessing a network with only $K = 3$ available queues (i.e., slices). First, the ideal slice allocation is computed by solving the optimization problem (1) (or (2)). Let us suppose that the optimal solution is $\bar{l} = [500, 600, 1000, 1200, 1300, 2000]$ kbps (first row in Fig. 5). The flows are then aggregated based on the following quantization thresholds: $\{800, 1400\}$. According to such quantization, three groups of flows are created: $g_1 = \{500, 600\}$, $g_2 = \{1000, 1200, 1300\}$ and $g_3 = \{2000\}$ (second row in Fig. 5). Finally, three slices equal to, respectively, 1100 kbps, 3500 kbps, and 2000 kbps are created (third row in Fig. 5). Thanks to the TCP fairness, the flows in the first slice are expected to obtain on average a bandwidth share equal to 550 kbps, the flows in the second slice 1166 kbps, and the flow in the third slice 2000 kbps, thus achieving an approximation of the optimal solution $\bar{l}$.

Weighted Proportional strategy. In this case, all the video sessions having the same resolution $r \in \mathscr{R}$ are assigned to the same slice. This approach has the advantage of not requiring to solve the optimization problem. The channel capacity C is split based on the following equation:

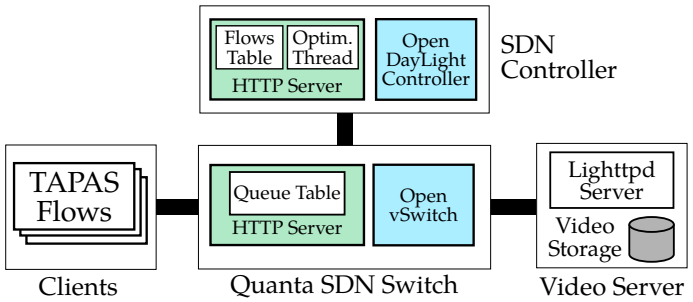


Fig. 6. The implementation of the VCP

$$C = \sum_{r \in \mathscr{R}} \alpha_r N_r b \tag{3}$$

where N_r is the number of clients having a screen resolution equal to r, α_r is the weighting coefficient for the resolution r, and b is the unknown variable. Once b is computed by solving (3), the slices sizes are set equal to $\alpha_r N_r b$.

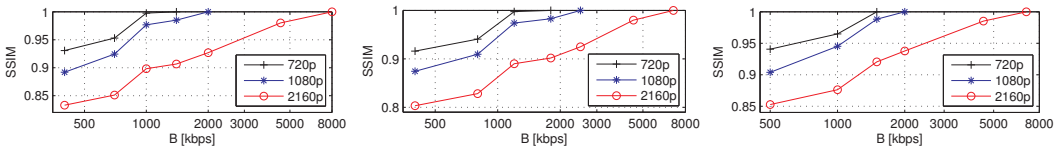
The weighting coefficients α_r are computed according to the following procedure. For each video clip v in the video catalog $\mathscr{V}$ we compute a linear regression of the video quality functions $U_{r,v}(b)$ relative to clients with screen resolution r. With this procedure the video quality functions $U_{r,v}(b)$ are approximated with the following linear function $U_{r,v}(b) = b/\alpha_{r,v}$. The value of α_r is computed by taking the average of $\alpha_{r,v}$ with $v \in \mathscr{V}$.

A safety mechanism is used to ensure that in the case of a large number of sessions, all the flows are provided with at least their lowest bitrates. Compared to *Quantized*, this strategy is expected to be less accurate, but it is cheaper to be implemented and allows for a higher scalability. In particular, since this approach does not require the solution of an optimization problem, the allocation has a very low complexity. The next Section quantifies the performance impact due to the approximation introduced by the Weighted Proportional strategy compared to the Quantization strategy.

4.3 The Testbed Setup

The Video Control Plane has been implemented in the testbed shown in Fig. 6, where three *Intel Core Duo* machines running *Ubuntu 14.04* are connected through a Quanta SDN switch. The client machine generates a configurable number of DASH video flows by means of the TAPAS tool [11]. The server machine hosts the *Lighttpd* HTTP server to send the video segments to the clients. The controller machine hosts the *Opendaylight Hydrogen Release* SDN Controller and the NC. The switch is a *Quanta T1048-LB9*, with *PicOS v2.6* OS and *Open vSwitch 2.3.0* as software switching stack. The bottleneck link is the *GbE* cable between the switch and the client machine. Its capacity is shaped by means of the `tc` Linux tool. In the following we focus on the implementation of: 1) the NC; 2) the TAPAS clients; 3) the video content encoding and the SSIM evaluation.

Network Controller. The NC has been implemented through two communicating HTTP servers, one hosted by the controller machine and one hosted by the switch, as shown in Fig. 6. Both the servers have been written in Python. The HTTP server hosted at the controller maintains the Active Flows Table, executes the Optimization Module in a Python thread, and establishes communication pipes through JSON APIs. In particular, two pipes are handled by the HTTP server: 1) the first with the video clients, which has the task of receiving the *set-up* and *tear-down* messages from the



(a) Big Buck Bunny (b) Tears of Steel (c) Sintel

Fig. 7. Measured SSIM for the considered videos and client resolutions

clients and send them the selected bitrates; 2) the second with the HTTP server at the SDN switch to create, manage, and delete QoS queues. The HTTP server running on the switch maintains the Queue Table and manages the QoS queues through the *Open vSwitch 2.3.0* APIs. A slice is generated by creating a dedicated queue on the network interface. The slice bitrate computed by the Optimization Module is set on the corresponding queue as the minimum guaranteed rate for the flows assigned to it. The employed switch allows to create 8 queues on the Ethernet interface, 7 of which dedicated to the video slices and one to the Arrival Slice. The Optimization thread employs the communication pipes to perform three actions: 1) in the case BG or BG+BR strategies are used, it communicates the selected bitrates to the clients; 2) it handles Openflow rules; 3) it manages the slices size.

The *Arrival Slice* size is dynamically set at each execution of the Optimization Module based on a periodically updated measure of the video traffic arrival statistics. In particular, at each sampling time kT_s the arrival rate of video flows $\hat{\lambda}(kT_s)$ is estimated with an EWMA filter and the *Arrival Slice* is set equal to $T_s b_{min} \hat{\lambda}(kT_s)$, where b_{min} is the minimum bandwidth we want to guarantee to each video flow during the start-up phase.

TAPAS clients. The client machine employs TAPAS (Tool for rApid Prototyping of Adaptive Streaming control algorithms) [11] to generate the video sessions. TAPAS is an open-source video client supporting DASH and HLS written in Python that allows to easily design and carry out experimental performance evaluations of adaptive streaming controllers. The following client-side algorithms, described in Section 3.3, have been implemented using TAPAS: Elastic, PANDA, Conventional, and the Thin Client for the Bitrate Guidance case. In order to run several (up to 50) concurrent video clients on the same client machine, we employ a TAPAS feature that allows to disable video segments decoding. When this feature is used, the obtained playout buffer dynamics is exactly the same that would be obtained if the video segment had been decoded, but with the advantage of remarkably decreasing the CPU and memory usage [11].

Video Content. The video content has been encoded with the H.264 codec with a frame rate equal to 30 fps and a segment size equal to 4 seconds. The video levels are encoded in VBR with two-passes and by specifying the target bitrate. The SSIM has been computed through the Matlab script released by the SSIM authors.

Finally, we have added a safety margin of 15% to the nominal bitrates when running the optimization in order to take into account the mismatch between the nominal bitrate reported in the video playlists and the real encoded bitrate.

5 EXPERIMENTAL RESULTS

5.1 The Scenario

In this Section, we describe the scenario considered in our experimental evaluation. The video catalog is composed of three videos: Big Buck Bunny[3], Sintel[4] and Tears of Steel[5]. We have considered three classes of client devices, whose screen resolutions are 720p, 1080p, and 2160p. Fig. 7 shows the measured SSIMs.

Each run is identified by a workload and has a duration of 900s. A workload defines for each video session of the run: 1) the starting time, which is generated by a Poisson arrival process with parameter λ; 2) the video, which is chosen from the video catalog according to a discrete uniform distribution; 3) the device resolution, which is chosen from the set of client resolutions according to a discrete uniform distribution. Background non-video traffic has not been considered in our investigation, since we assume that at least one bandwidth slice is dedicated to best-effort traffic. In fact, we have experimentally evaluated[6] that when best-effort background traffic is assigned to a low priority bandwidth slice it does not interfere with video traffic assigned to the video slices managed by the VCP. As such, we denote with C the quota of the link capacity available for video flows.

To generate a configurable link load, we have employed the following approach. The duration of all the video sessions has been set to $D = 300$s. As a consequence, the run is split into two phases. In the first one, lasting D seconds, there are only flow arrivals and no departures; in this phase, the number of active sessions grows with an average pace of λ. Then, during the second phase, the average arrival rate matches the average departure rate and – as a consequence – the average number of active sessions keeps to $N = \lambda D$. During this phase, the average bandwidth fair share for each flow is $C/(\lambda D)$ Mbps. By keeping C fixed and setting different values of λ, we can set the link load for each workload.

Throughout all the experimental evaluation we have set the link capacity C equal to 50 Mbps, a propagation round trip time of 50ms and a queue size equal to the bandwidth delay product. The minimum guaranteed bandwidth $b_{\min}$ of the *Arrival Slice* has been set equal to 1000 kbps. The Optimization Module sampling time T_s, unless otherwise specified, has been set to 30 s. The effect of T_s on performance has been evaluated in [9] and not shown in this paper due to space constraints.

We have employed the following quantization thresholds for the *Quantized Bandwidth Reservation* $\{1200, 1500, 1800, 2100, 2500, 5000\}$ kbps. Finally, in the *Weighted Proportional Bandwidth Reservation* strategy we have computed the following weighting coefficients $\alpha_{720\text{p}} = 1$, $\alpha_{1080\text{p}} = 1.4$, and $\alpha_{2160\text{p}} = 4.7$ based on the procedure described in Section 4.2.

5.2 The Metrics

In each run, we evaluate the following metrics to compare the performance of the investigated strategies.

RMSE. The Root Mean Squared Error is computed as the root of the average squared error between the optimal SSIM for the n-th user $\text{SSIM}_n^\star$, which is set by the Optimization Module, and

[3]http://distribution.bbb3d.renderfarming.net/video/mp4/bbb_sunflower_2160p_30fps_normal.mp4
[4]https://download.blender.org/durian/movies/Sintel.2010.4k.mkv
[5]http://ftp.nluug.nl/pub/graphics/blender/demo/movies/ToS/tearsofsteel_4k.mov
[6]Results are not reported due to space constraints.

Table 1. Considered network-assisted approaches

Symbol	Network-assisted approach
BR_Q	Quantized Bandwidth Reservation
BR_{WP}	Weighted Proportional Bandwidth Reservation
BG	Bitrate Guidance
BG + BR	Hybrid Bitrate Guidance and Bandwidth Reservation

the corresponding measured SSIM, SSIM_n.

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(\text{SSIM}_n - \text{SSIM}_n^{\star})^2}$$

This metric is proportional, through $1\sqrt{N}$, to the ℓ^2 distance between the optimum SSIM allocation vector $[\ \text{SSIM}_1^{\star} \ \cdots \ \text{SSIM}_N^{\star}]^T$ computed by the VCP and the measured SSIM vector $[\ \text{SSIM}_1 \ \cdots \ \text{SSIM}_N]^T$. Thus, by definition the RMSE measures the accuracy of a network-assisted approach in actuating the optimal allocation according to the management policy. This allows an unbiased comparison of the mechanisms regardless of the functional being used in the optimization problem. We stress that, since in this paper the Optimization Module enforces QoE fairness, a lower RMSE indicates a higher QoE fairness, in the sense defined by the functional being optimized.

Switching Frequency. It is computed as the average number of video bitrate switches in a second (measured in Hz). The switching frequency negatively affects the QoE only when it is higher than a threshold, which is on the scale of 0.1 Hz [18, 30].

Download Rate. The client measures it as the downloaded bytes in a given time interval.

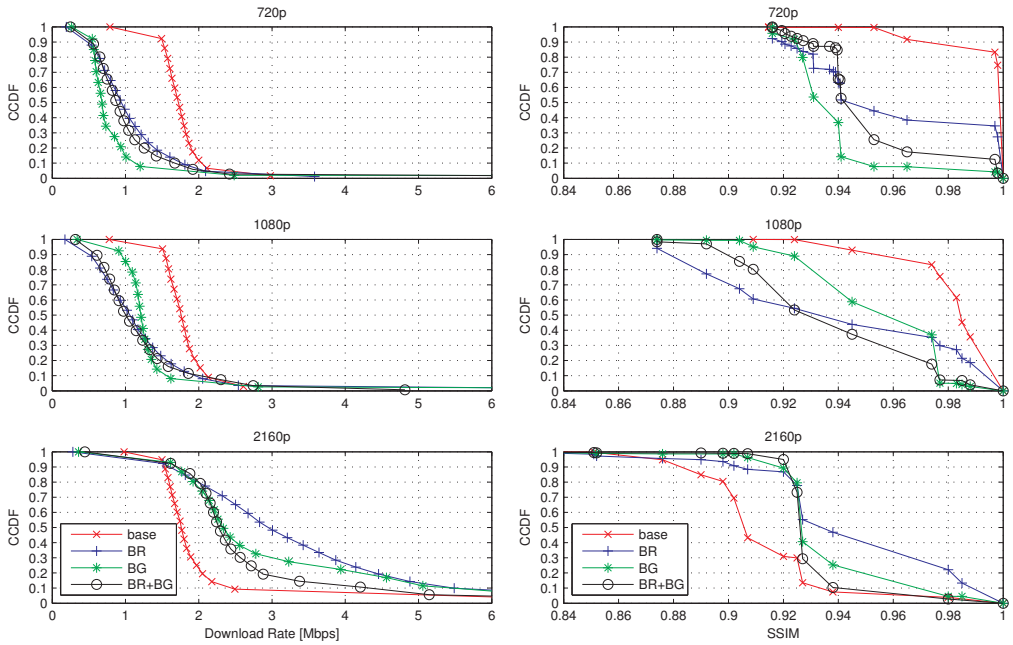
We do not report the rebuffering ratio in the results since it was negligible (lower than 0.5%) in all the experiments. This is arguably due to the fact that the lowest bandwidth fair share tested in the experiments is about 1.4 Mbps, which is much higher than the lowest bitrate for each video [16].

5.3 Results

In this section, we describe the results obtained by the considered network-assisted approaches shown in Table 1.

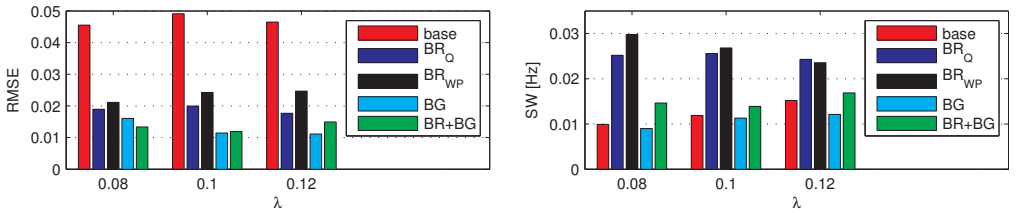
5.3.1 General performance. We start our analysis by comparing the overall performance achieved by the considered strategies. The case in which no Video Control Plane is used is labeled as *baseline* and is employed as a term of comparison. In the case of *baseline* and BR the client-side algorithm Elastic has been used (the impact of the client-side algorithm is separately investigated in Section 5.3.2).

First of all, we evaluate the effectiveness of the considered network-assisted strategies in enforcing the Video Quality Fairness management policy. Towards this end, we consider a single run corresponding to an arrival rate $\lambda = 0.08$ (runs with a different arrival rate exhibit similar qualitative behavior). Fig. 8 (a) and (b) show the complementary CDFs (CCDF) of the download rate and SSIM broken down by video client resolution. Let us consider the CCDFs of the download rate, shown in Fig. 8 (a). In the *baseline* case, the median value is roughly equal to 1.7 Mbps regardless of the client resolution. On the contrary, all the considered network-assisted approaches provide a median download rate that depends on the resolution. In particular, the 2160p flows obtain a higher median bandwidth share compared to the *baseline* case which does not provide service



(a) Download rate (b) SSIM

Fig. 8. Complementary CDFs of the per-resolution download rate and SSIM when $\lambda = 0.08$



(a) RMSE (b) SW

Fig. 9. RMSE and SW obtained by the considered network-assisted approaches as the link load varies.

differentiation. The expected consequence is that flows with smaller resolutions are assigned with a lower bandwidth share. Let us now consider Fig. 8 (b) to check the impact of service differentiation on the obtained SSIM. In the baseline case, users with 720p screen resolution achieve an SSIM with a median higher than 0.99. However, the *baseline* case heavily penalizes 2160p users who obtain a median SSIM of around 0.905. On the other hand, all the considered network-assisted approaches provide a fair Video Quality across different users. Moreover, Fig. 8 (b) shows that the BR approach provides the best SSIM compared to the other considered strategies. In particular, 40% of the 2160p flows experience an SSIM higher than 0.95 in the case of BR whereas BG and BG+BR obtain an SSIM greater than 0.925. This improvement is due to the higher download rate achieved by BR.

Let us now consider Fig. 9 that shows the measured RMSE for several arrival rates λ. The figure shows that all the network-assisted approaches achieve a lower RMSE compared to the *baseline* case.

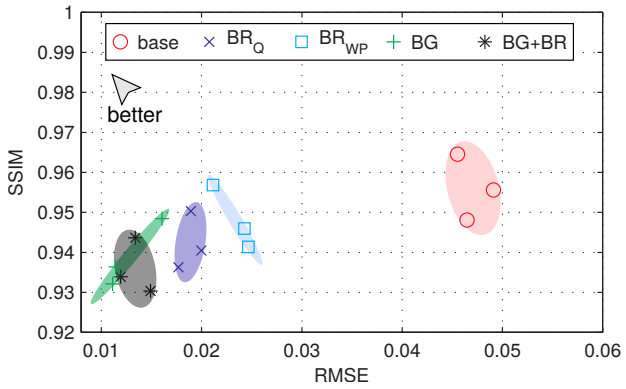


Fig. 10. The trade-off between Video Quality Fairness (RMSE) and average video quality (SSIM)

The *baseline* provides an RMSE higher than 0.045 for all the considered link loads, whereas all the network-assisted strategies are in general able to keep the RMSE below 0.025. The BG and BG+BR approaches outperform both the BR approaches. Adding bandwidth reservation to bitrate guidance (BG+BR) does not offer a distinct advantage compared to the BG strategy. The BR_Q is slightly more accurate in actuating the Video Control Plane decisions compared to the BR_{WP} strategy due to the higher granularity of its slicing mechanism. BR_{WP} balances the loss of accuracy with its lower implementation costs. Finally, the performances of all the investigated strategies are insensitive to the link load.

Fig. 9 shows the measured Switching Frequency as a function of the arrival rate λ. The figure confirms that Bandwidth Reservation increases the Switching Frequency. In particular, both the bandwidth reservation strategies provide Switching Frequencies up to three times higher than the ones of BG and BG+BR. However, it is important to notice that even the highest measured Switching Frequency, i.e. 0.03Hz, does not significantly affect the perceived QoE [30].

Fig. 10 shows a scatter plot which clearly represents the existing trade-off between the Video Quality Fairness, measured through the RMSE, and the video quality expressed in terms of SSIM.

The higher RMSE shown by the *baseline* corresponds to an SSIM between 0.95 and 0.97, whereas the SSIM of the network-assisted approaches is in the range between 0.93 and 0.96. Thus, we can conclude that the proposed management policy trades the Video Quality Fairness for the average Video Quality. This trade-off is unavoidable in resource allocation problems and goes under the name of the *"price of fairness"* [6].

Summary: All the considered network-assisted approaches provide a fair Video Quality across the video sessions compared to the baseline case in which no VCP is used. Moreover, VCP trades off a higher Video Quality Fairness for lower average video quality. Regarding Video Quality Fairness, Bitrate Guidance provides the best results, whereas Bandwidth Reservation slightly improves the video quality but with a higher Switching Frequency.

5.3.2 The impact of the client-side algorithm on the Bandwidth Reservation approach. We now investigate the impact of the considered client-side algorithms, namely Conventional, PANDA, and Elastic, on the performance of the BR approach. To this purpose, we investigate two types of scenarios. In the first one, that we name the *homogeneous case*, all the clients employ the same bitrate selection algorithm. In the second type of scenario, called the *heterogeneous case*, the three algorithms are concurrently used. In such a case, we assign each video client with a bitrate selection

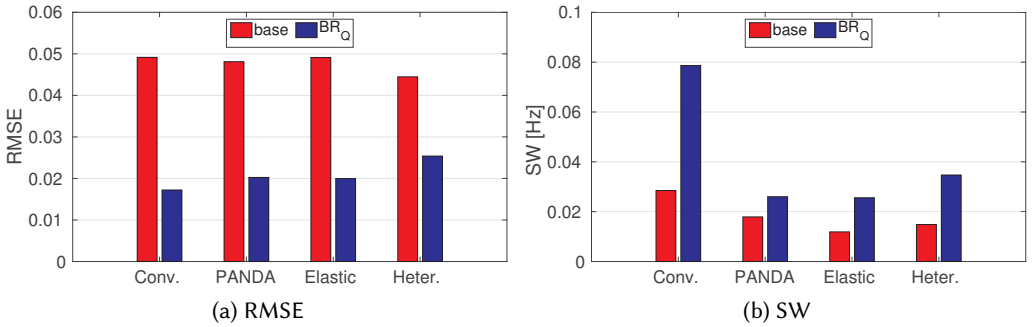
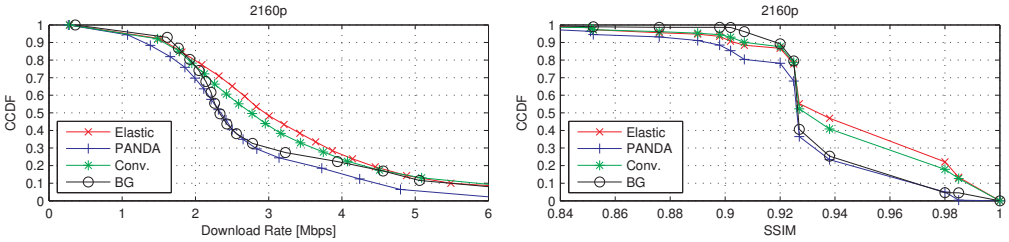


Fig. 11. The impact of client-side algorithms in the *homogeneous clients* scenario (first three columns groups) and in the *heterogeneous clients* scenario (last columns group) in the case of the baseline and the BR strategy ($\lambda = 0.08$)

algorithm drawn from a discrete uniform distribution. We consider the case where the VCP is not employed as the *baseline* term of comparison for the performance of the client-side algorithms.

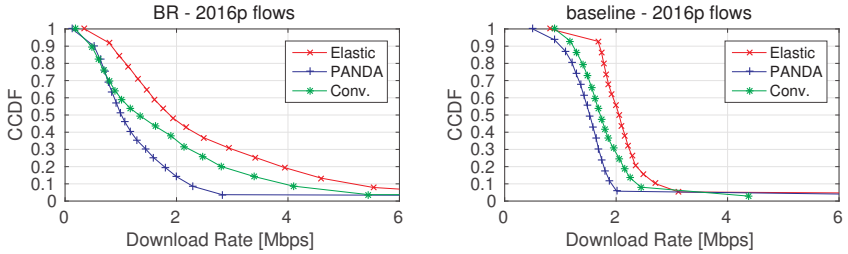
Fig. 11 (a) shows the RMSE when the arrival rate λ is set to 0.08. We now focus on the three homogeneous scenarios whose results are represented by the first three bar groups labeled "Conventional," "PANDA," and "Elastic." The RMSE is roughly insensitive to the employed client-side algorithm regardless the VCP is used or not. We now consider the CCDFs of the download rate and the SSIM video sessions ($\lambda = 0.08$). In Fig. 12 we show only 2160p sessions since performance differences are more remarkable for these clients. In this figure, BG is the term of comparison since it does not employ a client-side bitrate adaptation (see Section 3). Let us focus on Fig. 12 (a). The first important difference is that Conventional and Elastic always provide a higher bandwidth share to 2160p video sessions with the BR strategy compared to the case in which BG is used. In particular, the median bandwidth share for Elastic, Conventional, and BG are roughly 3 Mbps, 2.8 Mbps, and 2.3 Mbps respectively. On the other hand, PANDA does not exploit the advantage provided by BR and provides a lower bandwidth share to 2160p users compared to BG. This issue is due to the PANDA's sluggishness in tracking the time-varying available bandwidth [10]. As a consequence, Fig. 12 (b) shows that PANDA obtains a lower SSIM. Despite the fact that SSIM medians are roughly equal since the measured RMSE are similar (Fig. 11 (a)), Elastic clearly provides the best results whereas PANDA obtains the same SSIM as BG.

We now consider the heterogeneous scenario. Fig. 13 (a) shows the complementary CDFs of the download rates grouped by client-side algorithm. In general, with both *baseline* (i.e., no control plane is used) and BR, Elastic obtains higher download rates, followed in order by Conventional and PANDA. Authors of [10] have shown that, compared to Conventional and PANDA, Elastic provides a better bandwidth utilization when several video flows share a channel. The slice aggregation mechanism used by BR further increases this performance difference. Elastic obtains a larger bandwidth share when BR is used (Fig. 13 (a) on the left) compared to the *baseline* case (Fig. 13 (a) on the right). In particular, the 80-th percentile of the download rate obtained with BR and *baseline* are equal to 4 Mbps and 2.4 Mbps, respectively. The higher bandwidth utilization results in a higher SSIM, especially for Elastic. We now turn our attention to the video quality fairness measured through the RMSE and shown in Fig. 11 (a). Compared to the homogeneous scenarios, the RMSE is larger in the heterogeneous case confirming that fairness worsens in this last scenario. The QoE fairness worsens due to the slice aggregation mechanism that bases on the assumption that video flows fairly share the bandwidth. However, BR improves the RMSE compared to the *baseline*.

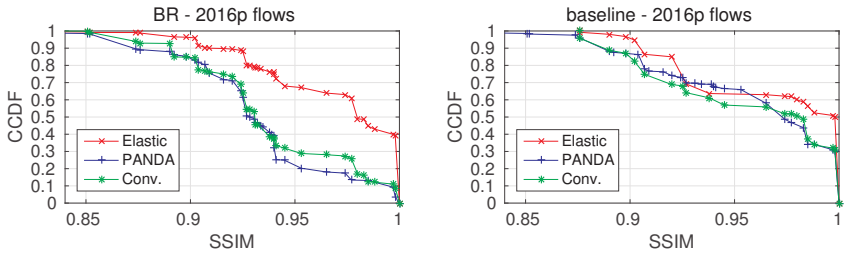


(a) Download rate (b) SSIM

Fig. 12. Complementary CDFs of download rate and SSIM with different client-side algorithms



(a) Download Rate

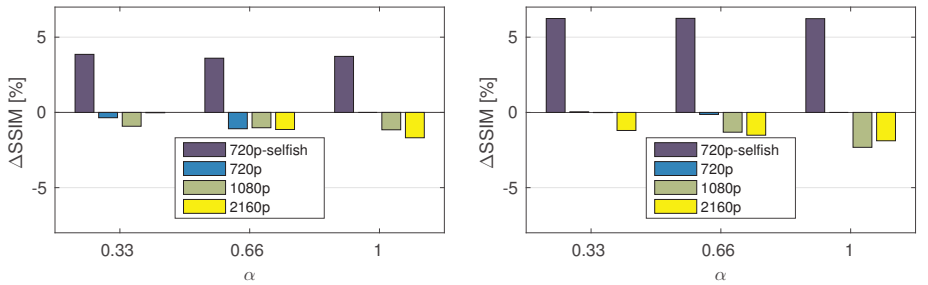


(b) SSIM

Fig. 13. Complementary CDFs of 2160p users with heterogeneous client-side algorithms: Download Rate (a) and SSIM (b)

These results suggest that confining video flows using the same client-side algorithm to the same slice is beneficial for fairness. In heterogeneous scenarios, we argue that per-flow rate-shaping performed either at the video server or at the switch could be used to alleviate fairness issues [2, 12]. The performance evaluation and the scalability issues due to the deployment of such an approach require a separate study and are outside the scope of this paper.

To conclude, we analyze the performance regarding the Switching Frequency obtained by the considered client-side algorithm. Fig. 11 (b) shows that Conventional exhibits the worst performance both with the *baseline* and the BR due to its very aggressive bitrate adaptation strategy. With the BR it reaches 0.08Hz, i.e. roughly one switch each three video segments. PANDA and Elastic, instead, provide similar results and show a slight increase of the switching frequency due to the BR compared to the *baseline* case. In the heterogeneous scenario, the Switching Frequency is below 0.04Hz both with baseline and BR.



(a) BR (b) BG

Fig. 14. Impact of selfish clients on the video quality perceived by the video clients, which are grouped by resolutions: BR (a) and BG (b)

Summary: The Video Quality Fairness, measured through the RMSE, is insensitive to the client-side algorithm employed in conjunction with the BR strategy. However, Elastic and Conventional provide higher SSIM values compared to PANDA. Conventional provokes a high switching frequency that might be detrimental to QoE. VQF performance worsens when different client-side algorithms share the same bandwidth slice.

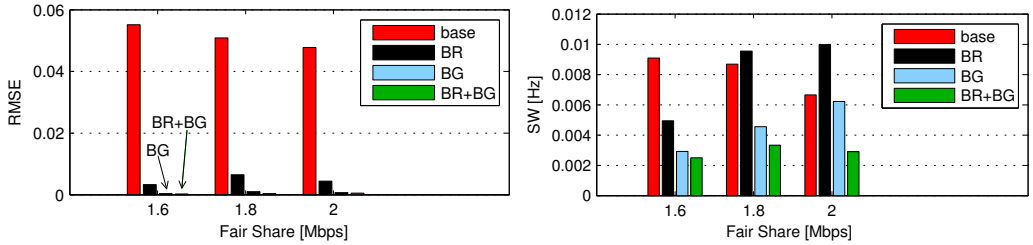
5.3.3 The impact of selfish clients. In this Section, we investigate the robustness of the proposed network-assisted approaches to the presence of *selfish clients*, which advertise a false (higher) screen resolution to the Control Plane with the intent of obtaining a higher bandwidth share. This is a relevant scenario to be considered since the bitrate selection control logic typically runs in the web browser in a Javascript application. Thus, even authenticated clients could advertise a false screen resolution and behave selfishly.

To the purpose, we have considered a scenario in which we allow a fraction α of the clients with 720p screen resolution to advertise a screen resolution equal to 2160p. We have investigated the performances of the BR and BG strategies as α varies in $\{0.33, 0.66, 1\}$. Clients that advertise their true screen resolution are defined as *legitimate users* to distinguish them from selfish users. Fig. 14 shows the SSIM percentage variation compared to the case without selfish clients function of α. Users are grouped by screen resolution. Fig. 14 (a) depicts the percentage variation for the BR strategy. The selfish clients obtain an SSIM that is roughly 4% higher than in the scenario without selfish clients regardless of α. At the same time, the legitimate clients suffer from SSIM losses between 1% and 2%. Fig. 14 (b) shows the percentage variation in the case of the BG strategy. In this case, the video quality gain for the selfish clients is more pronounced, namely higher than 6%. As a consequence, legitimate clients incur in a higher SSIM percentage loss: when $\alpha = 1$, both 1080p and 2160p legitimate clients lose more than 2%. It is worth to notice that BG is more sensitive than BR to the presence of selfish clients.

Summary: The presence of selfish clients advertising a false resolution to obtain a larger bandwidth share, and consequently a higher quality, worsens the performances of the proposed network-assisted strategies. BG is more sensitive compared to the BR approach.

5.3.4 The impact of per-flow queuing. In this paragraph, we investigate the impact of the grouping strategy to implement the slicing in the BR approach. Since the number of queues is limited to 8 in our testbed, the only way to do it is to consider a different scenario where a number of flows lower than 8 is generated. In this way, we can dedicate a single slice to each flow, without using the flow aggregation strategies presented in Section 4 in the case of the BR and the BG+BR

approaches. In this scenario, we generate 5 flows according to a Poisson arrival process. To consider several link loads, the link capacity has been set to 8, 9 and 10 Mbps, corresponding to a bandwidth fair share of 1.6, 1.8, and 2 Mbps respectively.



(a) RMSE (b) Switching Frequency

Fig. 15. RMSE and Switching Frequency in the case of 5 video flows with per-flow queuing

Fig. 15 (a) and (b) show respectively the RMSE and the Switching Frequency. If on one hand, the RMSE in the *baseline* case is comparable to the one obtained in the other scenario, on the other hand, all the network-assisted approaches remarkably improve the RMSE compared to the case where flow aggregation strategies are used to implement bandwidth slicing. In particular, the RMSE obtained by BG and the BG+BR is close to 0, whereas BR provides an RMSE below 0.01.

Similar considerations hold for the Switching Frequency. The *baseline* shows similar behavior compared to the other scenario (around 0.01 Hz), whereas BG and BG+BR are able to keep it below 0.005 Hz. Although BR obtains the worst performance, it provides an improvement compared to the other scenario. This improvement is due to the per-flow queuing that avoids video sessions to share the same slice.

Summary: The experimental results show that Video Quality Fairness and Switching Frequency are improved when per-flow queuing is used by network-assisted strategies.

5.3.5 Discussion. We conclude this Section with a brief discussion of the overall features of the considered strategies in the light of the experimental results presented above.

The *Bandwidth Reservation* approach requires no control communication after the video session is established, which is beneficial for scalability to support a large number of concurrent clients. Moreover, no information on the network state is exposed to the clients, which independently select the video bitrate according to the client-side adaptation algorithm. At the same time, a modest control effort due to bandwidth slices management has to be taken into account. The main drawback of this strategy is its sensitivity to the client-side algorithm employed to select the bitrate. In fact, experimental results have shown that both the Video Quality and the Switching Frequency depend on the employed client-side control algorithm. Moreover, VQF worsens when different client-side bitrate selection algorithms are used on the same slice.This issue is due to the different client-side control algorithms performance concerning bandwidth utilization. A way to tackle this problem would be to confine flows employing the same client-side algorithm to the same slice, with the added complexity of increasing the number of bandwidth slices. Finally, BR is moderately sensitive to the presence of selfish clients advertising a false screen resolution.

On the other side, the *Bitrate Guidance* approach provides the best accuracy in enforcing the management policy at the expense of a higher amount of communication (the VCP sends a message to each active video client each T_s seconds) and exposure of information reflecting the network state (i.e., the suggested bitrate). Furthermore, BG requires mutual trust between network and clients

and, as such, is sensitive to the presence of selfish clients which can obtain a higher bandwidth share by advertising a false (higher) screen resolution. Finally, our experimental results indicate that combining the two strategies does not provide a clear performance advantage compared to the BG strategy despite the higher control effort involved.

6 CONCLUSIONS

In this work, we have experimentally investigated several network-assisted strategies to actuate the decisions of a centralized Video Control Plane (VCP) whose goal is to provide Video Quality Fairness (VQF) to concurrent video streaming sessions sharing a common bottleneck. As a general result, we have found that all the considered network-assisted approaches provide a remarkable improvement in terms of obtained VQF compared to the case in which no VCP is employed. Concerning the VQF, the Bitrate Guidance approach provides the best results, whereas Bandwidth Reservation (BR) might improve the average video quality depending on the client-side algorithm. Regarding the impact of the client-side adaptation algorithm, we have found that the VQF is roughly insensitive to the employed client-side algorithm in the *homogeneous clients* case. However, Elastic and Conventional provide higher SSIM compared to PANDA. Conventional provokes a high switching frequency that might be detrimental to QoE. VQF worsens in the *heterogeneous clients* case, i.e., when different client-side algorithms are concurrently used on the same slice. Finally, we have investigated the sensitivity of the considered network-assisted approaches to the presence of selfish clients advertising false screen resolution to obtain a larger bandwidth share. Results indicate that the BG strategy is more sensitive compared to BR which better protects the legitimate clients from selfish clients. Finally, we have shown that VQF and switching frequency improve when BR employs per-flow queuing.

REFERENCES

[1] S. Akhshabi, L. Ananthakrishnan, A. C. Begen, and C. Dovrolis. 2012. What Happens When HTTP Adaptive Streaming Players Compete for Bandwidth?. In *Proc. ACM NOSSDAV*. 9–14.

[2] S. Akhshabi, L. Ananthakrishnan, A. C. Begen, and C. Dovrolis. 2013. Server-Based Traffic Shaping for Stabilizing Oscillating Adaptive Streaming Players. In *Proc. of ACM NOSSDAV*. 19–24.

[3] S. Akhshabi, S. Narayanaswamy, A. C. Begen, and C. Dovrolis. 2012. An experimental evaluation of rate-adaptive video players over HTTP. *Signal Processing: Image Communication* 27, 4 (2012), 271 – 287.

[4] A. Balachandran, V. Sekar, A. Akella, S. Seshan, I. Stoica, and H. Zhang. 2013. Developing a Predictive Model of Quality of Experience for Internet Video. In *Proc. ACM SIGCOMM*. 43, 339–350.

[5] D. Bertsekas and R. Gallager. 1992. *Data Networks*. Prentice-Hall.

[6] D. Bertsimas, V. F. Farias, and N. Trichakis. 2011. The price of fairness. *Operations research* 59, 1 (2011), 17–31.

[7] G. Cofano, L. De Cicco, and S. Mascolo. 2014. A Control Architecture for Massive Adaptive Video Streaming Delivery. In *Proc. VideoNext*. 7–12.

[8] G. Cofano, L. De Cicco, and S. Mascolo. in press 2016a. Modeling and Design of Adaptive Video Streaming Control Systems. *IEEE Trans. on Control of Network Systems* (in press 2016). DOI: http://dx.doi.org/10.1109/TCNS.2016.2631452

[9] G. Cofano, L. De Cicco, T. Zinner, A. Nguyen-Ngoc, P. Tran-Gia, and S. Mascolo. 2016b. Design and Experimental Evaluation of Network-assisted Strategies for HTTP Adaptive Streaming. In *Proc. of ACM MMSys '16*. Article 3, 12 pages. DOI: http://dx.doi.org/10.1145/2910017.2910597

[10] L. De Cicco, V. Caldaralo, V. Palmisano, and S. Mascolo. 2013. ELASTIC: a Client-side Controller for Dynamic Adaptive Streaming over HTTP (DASH). In *Proc. Packet Video Workshop*. 1–8.

[11] L. De Cicco, V. Caldaralo, V. Palmisano, and S. Mascolo. 2014. TAPAS: A Tool for rApid Prototyping of Adaptive Streaming Algorithms. In *Proc. VideoNext*. 1–6.

[12] L. De Cicco and S. Mascolo. 2014. An Adaptive Video Streaming Control System: Modeling, Validation, and Performance Evaluation. *IEEE/ACM Transaction on Networking* 22, 2 (2014), 526–539.

[13] A. Ganjam, F. Siddiqui, J. Zhan, X. Liu, I. Stoica, J. Jiang, V. Sekar, and H. Zhang. 2015. C3: Internet-Scale Control Plane for Video Quality Optimization. In *Proc. USENIX NSDI*. 131–144.

[14] P. Georgopoulos, Y. Elkhatib, M. Broadbent, M. Mu, and N. Race. 2013. Towards Network-wide QoE Fairness Using Openflow-assisted Adaptive Video Streaming. In *Proc. ACM SIGCOMM Workshop on Future Human-centric Multimedia*

Networking. 15–20. DOI: http://dx.doi.org/10.1145/2491172.2491181

[15] T. Hoßfeld, S. Egger, R. Schatz, M. Fiedler, K. Masuch, and C. Lorentzen. 2012. Initial delay vs. interruptions: between the devil and the deep blue sea. In *Proc of QoMEX Workshop*.

[16] T. Hoßfeld, R. Schatz, and U. R. Krieger. 2014. QoE of YouTube Video Streaming for Current Internet Transport Protocols. In *Proc. of Measurement, Modelling, and Evaluation of Computing Systems and Dependability and Fault Tolerance International Conference*. 136–150. DOI: http://dx.doi.org/10.1007/978-3-319-05359-2_10

[17] T. Hoßfeld, R. Schatz, M. Seufert, M. Hirth, T. Zinner, and P. Tran-Gia. 2011. Quantification of YouTube QoE via Crowdsourcing. In *IEEE Workshop on Multimedia Quality of Experience - Modeling, Evaluation, and Directions (MQoE)*.

[18] T. Hoßfeld, M. Seufert, C. Sieber, and T. Zinner. 2014. Assessing effect sizes of influence factors towards a QoE model for HTTP adaptive streaming. In *Proc. QoMEX*. 111–116.

[19] T. Hossfeld, D. Strohmeier, A. Raake, and R. Schatz. 2013. Pippi Longstocking calculus for temporal stimuli pattern on YouTube QoE: 1+ 1= 3 and 1· 4≠ 4· 1. In *Proceedings of the 5th Workshop on Mobile Video*. 37–42.

[20] R. Houdaille and S. Gouache. 2012. Shaping HTTP Adaptive Streams for a Better User Experience. In *Proc. of ACM MMSys*. 1–9. DOI: http://dx.doi.org/10.1145/2155555.2155557

[21] T.Y. Huang, N. Handigol, B. Heller, N. McKeown, and R. Johari. 2012. Confused, timid, and unstable: picking a video streaming rate is hard. In *Proc. ACM IMC*.

[22] M. Jarschel, F. Wamser, T. Höhn, T. Zinner, and P. Tran-Gia. 2013. SDN-based Application-Aware Networking on the Example of YouTube Video Streaming. In *Proc. EWSDN*. 87–92.

[23] J. Jiang, V. Sekar, H. Milner, D. Shepherd, I. Stoica, and H. Zhang. 2016. CFA: A Practical Prediction System for Video QoE Optimization. In *Proc. of USENIX NSDI*.

[24] J. Jiang, V. Sekar, and H. Zhang. 2012. Improving fairness, efficiency, and stability in HTTP-based adaptive video streaming with FESTIVE. In *Proc. ACM CoNEXT*. 97–108.

[25] J. W. Kleinrouweler, S. Cabrero, and P. Cesar. 2016. Delivering Stable High-quality Video: An SDN Architecture with DASH Assisting Network Elements. In *Proc. of ACM MMSys*.

[26] J. W. Kleinrouweler, S. Cabrero, R. van der Mei, and P. Cesar. 2015. Modeling the Effect of Sharing Policies for Network-assisted HTTP Adaptive Video Streaming. *ACM SIGMETRICS Perf. Evaluation Review* 43, 2 (2015), 26–27.

[27] S. S. Krishnan and R. K. Sitaraman. 2013. Video stream quality impacts viewer behavior: inferring causality using quasi-experimental designs. *IEEE/ACM Trans. on Networking* 21, 6 (2013), 2001–2014.

[28] A. Mansy, M. Fayed, and M. Ammar. 2015. Network-layer Fairness for Adaptive Video Streams. In *Proc. IFIP/IEEE Networking*. 42–48.

[29] M. K. Mukerjee, D. Naylor, J. Jiang, D. Han, S. Seshan, and H. Zhang. 2015. Practical, Real-time Centralized Control for CDN-based Live Video Delivery. In *Proc. ACM SIGCOMM*. 311–324. DOI: http://dx.doi.org/10.1145/2785956.2787475

[30] P. Ni, R. Eg, A. Eichhorn, C. Griwodz, and P. Halvorsen. 2011. Flicker effects in adaptive video streaming to handheld devices. In *Proc. ACM MultiMedia*. 463–472.

[31] S. Petrangeli, J. Famaey, M. Claeys, S. Latré, and F. De Turck. 2015. QoE-Driven Rate Adaptation Heuristic for Fair Adaptive Video Streaming. *ACM Trans. on Multimedia Computer Communication Applications* 12, 2, Article 28 (2015), 24 pages. DOI: http://dx.doi.org/10.1145/2818361

[32] K. Seshadrinathan, R. Soundararajan, A. C. Bovik, and L. K. Cormack. 2010. Study of Subjective and Objective Quality Assessment of Video. *IEEE Trans. on Image Processing* 19, 6 (June 2010), 1427–1441.

[33] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hoßfeld, and P. Tran-Gia. 2015a. A Survey on Quality of Experience of HTTP Adaptive Streaming. *IEEE Communications Surveys and Tutorials* 17, 1 (2015), 469–492.

[34] M. Seufert, T. Hoßfeld, and C. Sieber. 2015b. Impact of intermediate layer on quality of experience of HTTP adaptive streaming. In *Network and Service Management (CNSM), 2015 11th International Conference on*. 256–260.

[35] V. Sivaraman, T. Moors, H. Habibi Gharakheili, D. Ong, J. Matthews, and C. Russell. 2013. Virtualizing the Access Network via Open APIs. In *Proc. ACM CoNEXT*. 31–42. DOI: http://dx.doi.org/10.1145/2535372.2535381

[36] L. Toni, R. Aparicio-Pardo, K. Pires, G. Simon, A. Blanc, and P. Frossard. 2015. Optimal Selection of Adaptive Streaming Representations. *ACM Trans. Multimedia Comput. Commun. Appl.* 11, 2s, Article 43 (Feb. 2015), 26 pages.

[37] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. 2004. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Trans. on Image Processing* 13, 4 (2004), 600–612.

[38] C. Wilson, H. Ballani, T. Karagiannis, and A. Rowtron. 2011. Better Never Than Late: Meeting Deadlines in Datacenter Networks. In *Proc. ACM SIGCOMM*. 50–61. DOI: http://dx.doi.org/10.1145/2018436.2018443

[39] L. Zhi, X. Zhu, J. Gahm, R. Pan, H. Hu, A. C. Begen, and D. Oran. 2014. Probe and Adapt: Rate Adaptation for HTTP Video Streaming At Scale. *IEEE JSAC* 32, 4 (2014), 719–733. DOI: http://dx.doi.org/10.1109/JSAC.2014.140405

[40] T. Zinner, M. Jarschel, A. Blenk, F. Wamser, and W. Kellerer. 2014. Dynamic application-aware resource management using Software-Defined Networking: Implementation prospects and challenges. In *Proc. IEEE NOMS*. 1–6.