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

G. Cofano*, L. De Cicco†, T. Zinner‡, A. Nguyen-Ngoc‡, P. Tran-Gia‡, S. Mascolo*

*{giuseppe.cofano, mascolo}@poliba.it, †luca.de_cicco@telecom-sudparis.eu,
‡{zinner, anh.nguyen, trangia}@informatik.uni-wuerzburg.de

*Politecnico di Bari, Italy, †Télécom SudParis, France, ‡University of Würzburg, Germany

ABSTRACT

In this paper we investigate 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 the 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. Performance is assessed 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.

Categories and Subject Descriptors

C.2.3 [**Network Operations**]: [Network management]

Keywords

Adaptive Video Streaming; DASH; network-assistance; Control Plane; Quality of Experience; Fairness

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 smart phones [4]. 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 like service costs or Quality of Experience (QoE) significantly impact user engagement. Accordingly, video providers have to satisfy user expectations to avoid user abandonment and thus to increase revenues [15]. One of the main influence factors, which thus has to be improved, is the QoE [1, 15, 20].

Video providers currently rely on HTTP adaptive streaming approaches, a technique allowing video quality adaptation on short time scales, to deliver video clips to the users. Network resource allocation is managed in a distributed way at the end-points by each client. Video clients are equipped with controllers allowing to autonomously change the video bitrate to improve the QoE [22]. 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 [1].

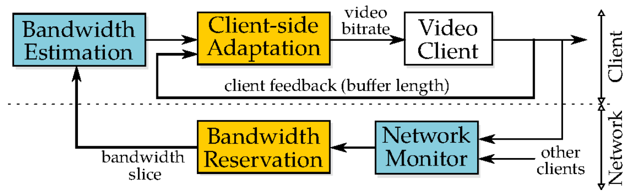
The simultaneous presence of several adaptive video stream flows transmitted via a shared bottleneck link results in a fair bandwidth distribution among the involved flows. QoE-relevant influence factors like the device capabilities or the user context are not taken into account. For instance, users 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 [10]. To overcome this problem, interaction between video and network provider is required.

The exchanged information can be leveraged by a video control plane enforcing network-assisted streaming strategies. To enable active cooperation between network elements, a standard signalling plane is required, such as the one proposed by Server And Network Assisted DASH (SAND DASH)[1]. This allows 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 especially beneficial in the presence of complex topologies [26].

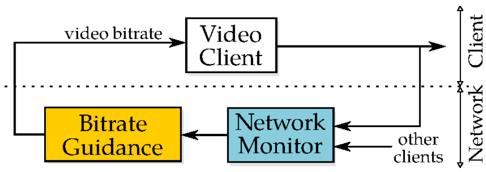
Our work provides a broad investigation of the design space of video control planes by studying several network-assisted strategies. In particular, we compare the performance of three categories of network-assisted approaches. The *Bandwidth Reservation* assigns a bandwidth slice to a video flow (or a group of video flows). Two nested control loops are established as shown in Figure 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 in order to assess interactions between these 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,

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

MMSys '16 May 10-13, 2016, Klagenfurt, Austria.



(a) Bandwidth Reservation



(b) Bitrate Guidance

Figure 1: Network-assisted approaches for adaptive video streaming

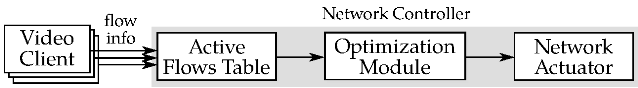


Figure 2: A block diagram of the control system

shown in Figure 1 (b), is named *Bitrate Guidance*: when this approach is employed the video bitrate is computed by a centralized algorithm running in a network element and enforced by the video client. Finally, we also take into account *hybrid strategies* combining Bandwidth Reservation and Bitrate Guidance.

In order to 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 [8].

2. 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, for instance the link connecting the ISP Access Network to the Home Router or the egress congested link of a CDN [17].

2.1 Control System Architecture

Figure 2 shows a block diagram of the overall control system that is made of 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. Such information is provided by each video client 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 accourding 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 2.2.

Video Client The clients undertake the following tasks: 1) setup/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.

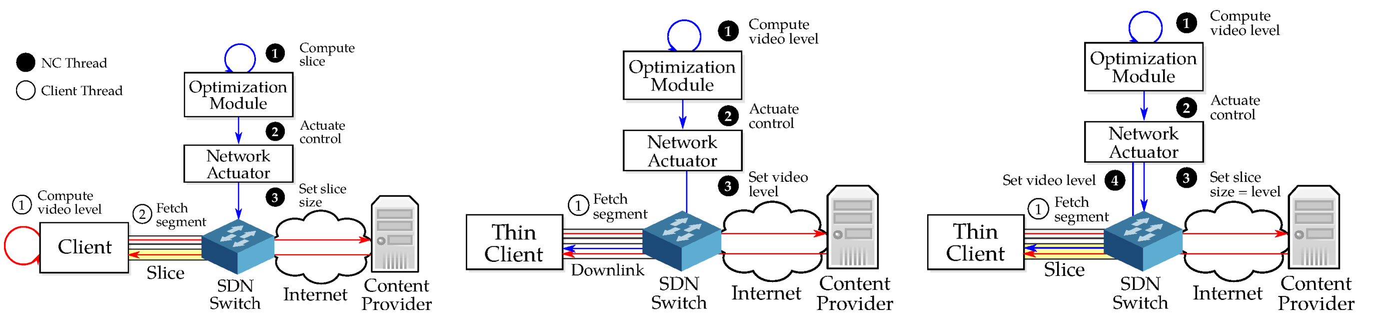
2.2 Network-assisted Streaming Approaches

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). Such approaches can be implemented in the control system shown in Figure 2 by combining two parallel and independent threads, as shown in Figure 3: the *Client thread* and the *NC thread*.

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. Figure 3 (a) shows how this approach can be implemented. 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. The computed values are sent 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 in order to match the selected bitrate, thus providing service differentiation to the flows in the shared slice. Figure 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, Figure 3 (b) labels such video client as a *Thin Client*. In order 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 quickly fill it again. 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 in two sub-actions: 1) the bandwidth reservation in the network and 2) the bitrate guidance. The client thread is again limited to



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

Figure 3: The considered network-assisted approaches

performing segment downloads, i.e. the client can be considered as a Thin Client exactly as in the case of the BG approach.

2.3 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 simultaneously achieve these goals and some trade-offs have to be made. A relevant classification to our investigation is the following, which makes the distinction between *rate-based* and *level-based* approaches [6]. The first approach requires the algorithm to insert pauses (OFF periods) between the downloads of consecutive segments in order 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 [6].

The interactions between the client-side control loop and the network control loop are investigated in the case of three algorithms: *Conventional* [25], *PANDA* [25], and *Elastic* [7]. We have decided to employ these algorithms in order to cover both control approaches.

Conventional is a *rate-based* algorithm selecting the video bitrate based on bandwidth estimates [25]. We have considered this algorithm to check if the Bandwidth Reservation strategy is able to perform well even when a very simple client-side algorithm is employed.

PANDA is a *rate-based* algorithm designed to cope with the fairness issues affecting several HAS algorithms [25]. It follows a probe-and-adapt approach, incrementing the bitrate to probe the available bandwidth.

Elastic 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 [7]. It has been shown that Elastic is able to overcome fairness issues affecting the rate-based algorithms.

2.4 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, or MSE. 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 [20].

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 [10] or QoE [19], which is indeed more appropriate than fair bandwidth allocation (i.e. QoS fairness) in the context of video delivery.

In this paper we consider, 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 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 is 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 will be shown 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

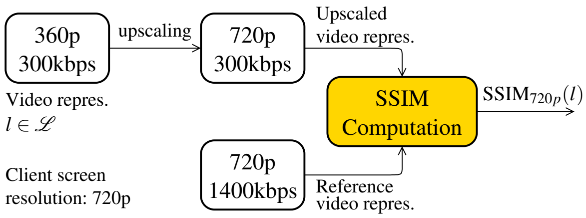


Figure 4: SSIM computation for the video representation l (300kbps, 360p): the video representation is first upscaled to the client resolution 720p and then compared to the corresponding reference video representation (1400kbps, 720p).

Video Representation		SSIM		
Level $\bar{l}_i$	Resolution $\bar{r}_i$	720p	1080p	2160p
400 kbps	640x360	0.931	0.892	0.833
700 kbps	854x480	0.953	0.924	0.851
1000 kbps	1280x720	0.998	0.977	0.898
1400 kbps	1280x720	1	0.985	0.907
2000 kbps	1920x1080	-	1	0.927
4500 kbps	2560x1440	-	-	0.980
8000 kbps	3840x2160	-	-	1

Table 1: SSIM for the video Big Buck Bunny with respect to three different reference video resolutions

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 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 is not higher than the link capacity.

PROBLEM 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* [2], 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: it can be shown that the problem corresponds to finding the root of a univariate equation that can be solved efficiently.

Video Quality Measurement. In order 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. They are generated off-line by means of the Structural SIMilarity (SSIM) index, which is employed as an estimate of the video quality [23].

The SSIM is an objective reference-based method to evaluate the quality of an image, which correlates well with the human perception and also allows an efficient computation. In the case of videos the SSIM is computed as the average SSIM over all the video frames.

We define the reference video as the best available video representation of that video clip at the client screen resolution. 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 a number of representations, each of them characterized by a bitrate $\bar{l}_i$ and a resolution $\bar{r}_i$. 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 $\bar{r}_i$ 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.

As a concrete example, Figure 4 shows how the SSIM is computed for a video representation with $\bar{l}_i = 300$kbps and $\bar{r}_i =$360p when the resolution $\hat{r}$ of the reference video, i.e. the client resolution, is 720p.

The results obtained when evaluating the quality of the encoded versions of the video Big Buck Bunny are reported in the Table 1 as an example. The last three columns of Table 1 show that the same bitrate has three different SSIMs corresponding to the three different resolutions of the reference video (720p, 1080p, 2160p). 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.

3. IMPLEMENTATION

3.1 Video Session Management

Figure 5 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 2.4. 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. Figure 5

[2]Our Video Quality Fairness policy differs in this aspect from the one proposed in [10] that makes the restrictive assumption that for each bitrate several representations with different resolutions are available.

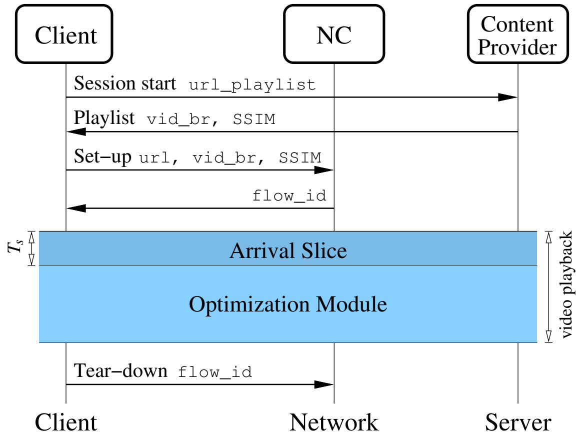


Figure 5: Video session flow diagram

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. In order to avoid a delayed start-up, the NC assigns the flow to the *Arrival Slice*, which is reserved to newly arrived video sessions. Then, at most after T_s seconds, the Optimization Module is executed, the video flow is removed from the *Arrival Slice* 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.

3.2 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 [24], 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.

We have designed and tested two strategies, both exploiting the fairness property of the video flows provided by the TCP. In fact, if flows with similar video bitrate are assigned to the same slice, TCP fairness will guarantee that, with some approximation, each of the flows will obtain a bandwidth close to the one set by the Optimization Module. In the following we describe the two proposed strategies.

***Quantized* strategy.** Each flow is mapped to one of the K slices through a quantization process. 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.

In order to explain the proposed strategy, we give an example, shown in Figure 6. Let us consider the case of $N = 6$ concurrent video flows accessing a network with only $K = 3$ available queues (slices). First, the ideal slice allocation is computed by solving the optimization problem (1) (or (2)). Let us suppose that the

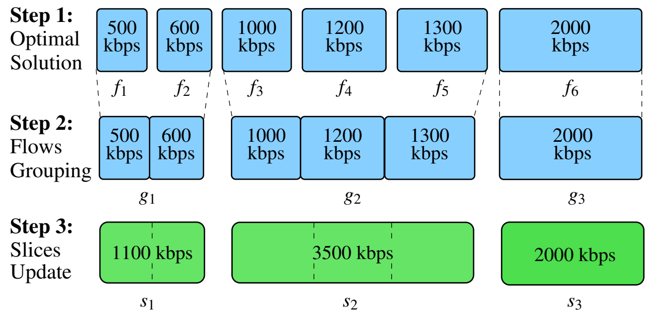


Figure 6: The Quantized aggregation strategy

optimal solution is $\bar{l} = [500, 600, 1000, 1200, 1300, 2000]$kbps (first row in Figure 6). 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 Figure 6). Finally, three slices equal to, respectively, 1100kbps, 3500kbps, and 2000kbps are created (third row in Figure 6). Thanks to the TCP fairness, the flows in the first slice are expected to obtain on average a bandwidth share equal to 550kbps, the flows in the second slice 1166kbps, and the flow in the third slice 2000kbps, 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:

$$C = \sum_{r \in \mathscr{R}} \alpha_r N_r b \qquad (3)$$

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 through a linear regression of the video quality functions. 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 strategy does not require the solution of an optimization problem, the allocation has a very low complexity. In the next Section we will quantify the impact on performance of the approximation introduced by the Weighted Proportional strategy with respect to the Quantization strategy.

3.3 The Testbed Setup

The Video Control Plane has been implemented in the testbed shown in Figure 7, 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 [8]. 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

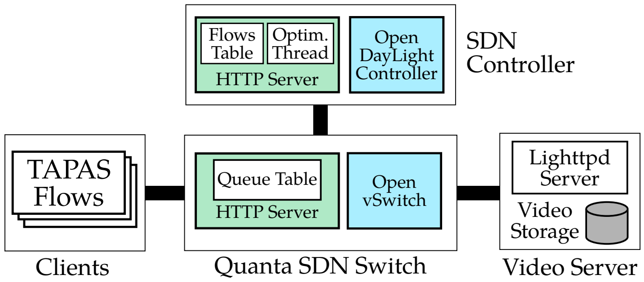


Figure 7: The implementation of the VCP

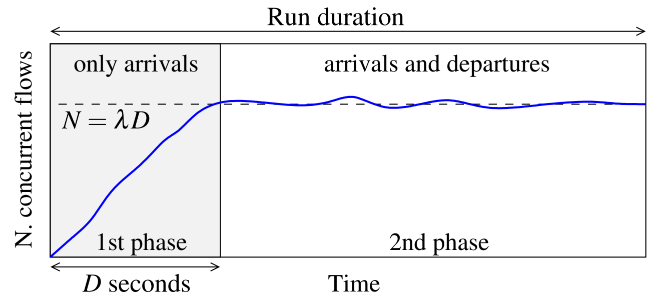


Figure 9: Number of concurrent video sessions over time

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 Figure 7. 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 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 Slic*e 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) [8] 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 2.3, 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 [8].

Video Content. The video content has been encoded with the H.264 codec with a frame rate equal to 30 fps. The segment size has been fixed to 4 seconds.

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.

4. EXPERIMENTAL RESULTS

4.1 The Scenario

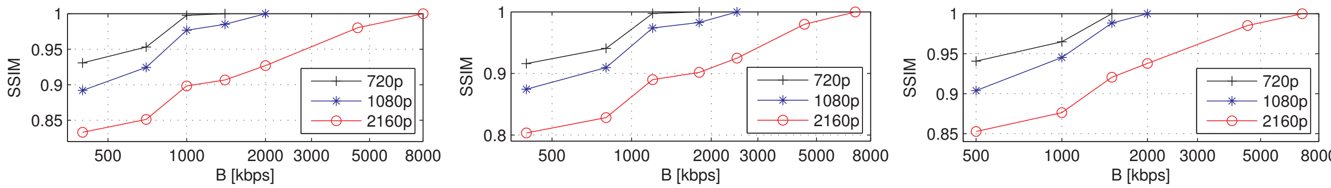
In this Section we describe the scenario considered in our experimental evaluation. The video catalog is composed of three videos: Big Buck Bunny[3], Sintel[4] and Tears of Steel[5]. We have considered three classes of client devices, whose screen resolutions are 720p, 1080p, and 2160p. The measured SSIM are shown in Figure 8.

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.

In order to generate a configurable link load we have employed the following strategy. The duration of all the video sessions has been set to $D = 300$s. As a consequence, the run is split in two phases as shown in Figure 9. In the first one, lasting D seconds, there are only flow arrivals and no departures; thus, 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 are able to set the link load for each workload.

Throughout all the experimental evaluation we have set the link capacity C equal to 50Mbps. The minimum guaranteed bandwidth $b_{\min}$ of the *Arrival Slic*e has been set equal to 1000kbps. The Optimization Module sampling time T_s, unless otherwise specified,

[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`



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

Figure 8: Measured SSIM for the considered videos and client resolutions

has been set to 30s (the effect of T_s on performance is separately assessed in Section 4.3.4).

We have employed the following quantization thresholds for the *Quantized Bandwidth Reservation*:

$$\{1200, 1500, 1800, 2100, 2500, 5000\}\text{kbps}$$

Finally, in the *Weighted Proportional Bandwidth Reservation* strategy the following coefficients have been employed: $\alpha_{720\text{p}} = 1$, $\alpha_{1080\text{p}} = 1.4$, and $\alpha_{2160\text{p}} = 4.7$.

4.2 The Metrics

In order to compare the performance of the investigated strategies, we have evaluated the following metrics in each run.

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 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 measures the accuracy of the network-assisted approach in enforcing the optimal allocation according to the management policy. Thus, since we enforce a Video Quality Fair allocation, the lower the RMSE the higher the achieved fairness.

Switching Frequency. It is computed as the average number of video bitrate switches in a second (measured in Hz). It has been shown that the switching frequency negatively affects the QoE only if it is higher than a threshold, which is on the scale of 0.1 Hz [12, 18].

Download Rate. It is measured by the client as the downloaded bytes in a given time interval.

We do not report the rebuffering ratio in the results since it was negligible (less 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 1.4Mbps, which is much higher than the lowest bitrate for each video [11].

4.3 Results

In this section we describe the results obtained by the considered network-assisted approaches shown in Table 2.

4.3.1 General performance

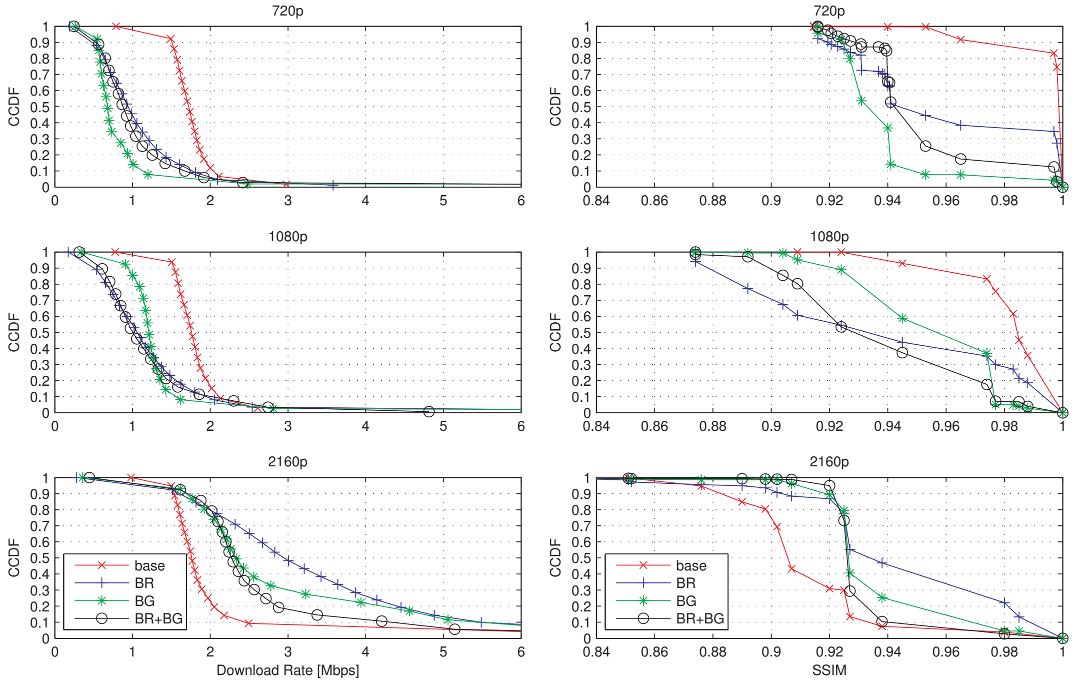
In this Section we compare the overall performance achieved by the considered strategies. The case in which no Video Control Plane is used is labeled as *baseline* and is employed as a term of comparison. In the case of *baseline* and BR the client-side algorithm Elastic has been employed (the impact of the client-side algorithm is separately investigated in Section 4.3.2).

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

Table 2: Considered network-assisted approaches

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). Figure 10 (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 Figure 10 (a). In the *baseline* case the median value is roughly equal to 1.7Mbps regardless of the client resolution. On the contrary, all the considered network-assisted approaches are able to provide a median download rate that depends on the resolution. In particular, the 2160p flows obtain a higher median bandwidth share with respect to the *baseline* case which does not provide service differentiation. This occurs, as expected, at the expense of the flows with smaller resolutions. Let us now consider Figure 10 (b) to check the impact of service differentiation on the obtained SSIM. The *baseline* case provides users with 720p screens a SSIM with a median higher than 0.99, but it heavily penalizes 2160p users which– achieve a median SSIM of around 0.905. On the other hand, all the considered network-assisted approaches are able to provide a fair Video Quality across different users. Moreover, Figure 10 (b) shows that the BR approach provides the best SSIM with respect to the other considered approaches. In particular, 40% of the 2160p flows experience a SSIM higher than 0.95 in the case of BR whereas BG and BG+BR obtain a SSIM higher than 0.925. This improvement is due to the higher download rate achieved by BR.

Let us now consider Figure 11 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. The *baseline* provides an RMSE higher than 0.045 for all the 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 clear advantage with respect to the BG strategy. The BR_Q is slightly more accurate in actuating the Video Control Plane decisions with respect to the BR_WP strategy due to the higher granularity of its slicing mechanism. The loss of accuracy exhibited



(a) Download rate (b) SSIM

Figure 10: Complementary CDFs of the per-resolution download rate and SSIM when $\lambda = 0.08$

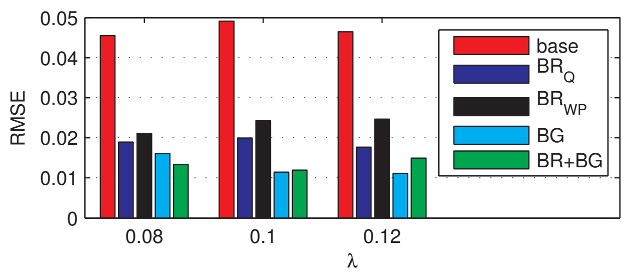


Figure 11: RMSE obtained by the considered network-assisted approaches as the arrival rate λ varies.

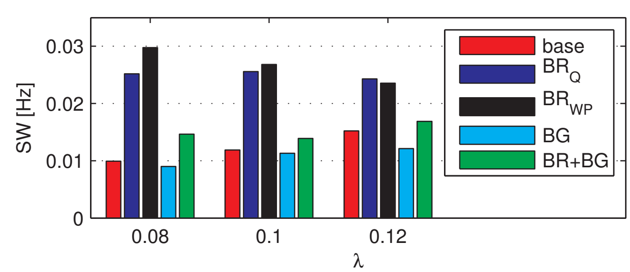


Figure 12: Switching Frequency of the investigated network-assisted approaches as the link load varies.

by BR_{WP} is balanced by its lower implementation costs. Finally, the performances of all the investigated strategies are insensitive to the link load.

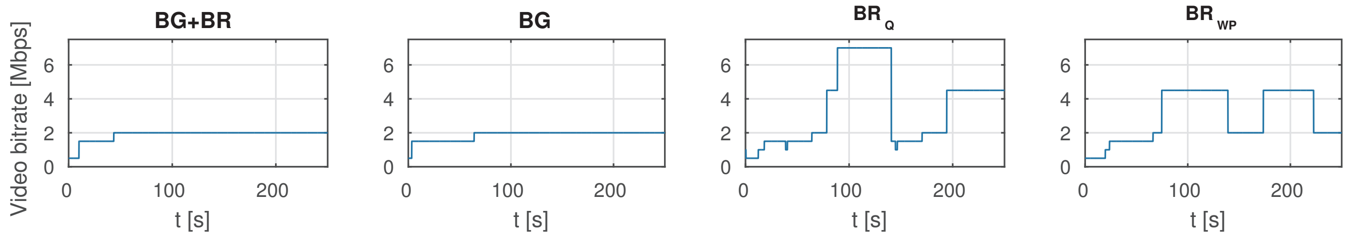
To get a further insight, we compare the video bitrate dynamics of corresponding video sessions when using the same workload for all the considered approaches. As an example, Figure 13 shows the 40-th and 45-th video session in the case of $\lambda = 0.1$. The figure shows that the BR strategies provoke several switches due to the client-side adaptation. On the other hand, BG and BG+BR provoke less switches since the video bitrate is directly set by the Optimization Module.

Figure 12 shows measured the 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 [18].

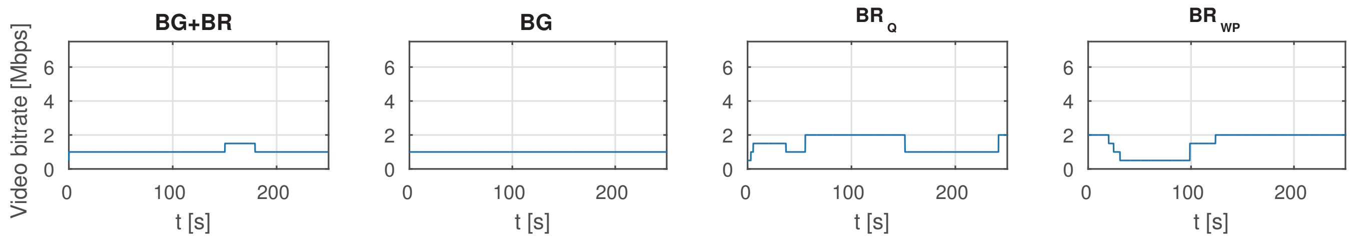
Figure 14 shows a scatter plot which clearly represents the existing trade-off between the Video Quality Fairness, measured through the RMSE, and the video quality expressed in terms of SSIM.

The higher RMSE shown by the *baseline* corresponds to a SSIM between 0.93 and 0.95, whereas the SSIM of the network-assisted approaches is in the range between 0.91 and 0.94. Thus, we can conclude that with the proposed management policy the Video Quality Fairness is obtained at the expense of the average Video Quality. This is an unavoidable trade-off in resource allocation problems that goes under the name of the *"price of fairness"* [3].

Summary: All the network-assisted approaches are able to provide a fair Video Quality across the video sessions compared to the baseline case in which no VCP is employed. Moreover, VCP trades off a higher Video Quality Fairness for lower average video quality. Bitrate guidance provides the best results in terms of Video Quality Fairness, whereas bandwidth reservation slightly improves the video quality but with a higher Switching Frequency.



(a) 40-th flow of the workload



(b) 45-th flow of the workload

Figure 13: Video bitrate dynamics of two flows of the run with $\lambda = 0.1$ with the considered approaches

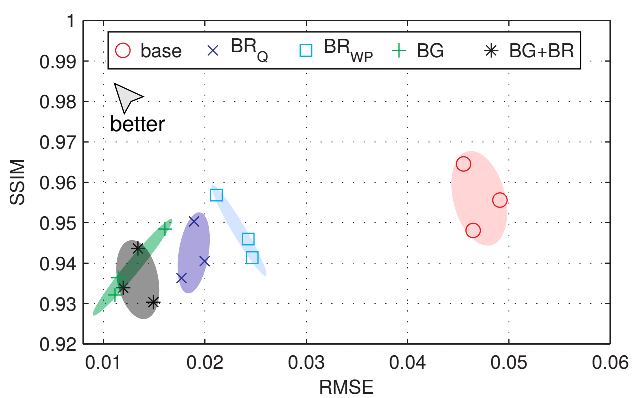
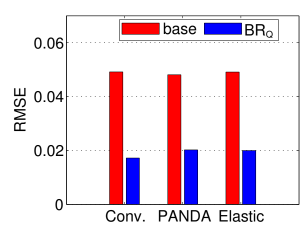
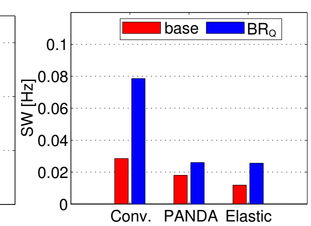


Figure 14: The trade-off between Video Quality Fairness (RMSE) and average video quality (SSIM)



(a) RMSE



(b) Switching Frequency

Figure 15: Comparison of the performance achieved with the client-side algorithms Conventional, PANDA, and Elastic in the case of the baseline and the BR strategy ($\lambda = 0.08$).

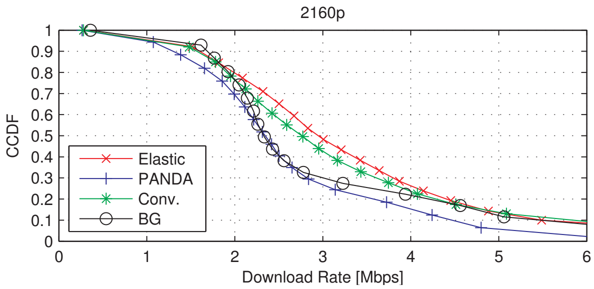
4.3.2 *The impact of the client-side algorithm on the Bandwidth Reservation approach*

We now investigate the impact of the considered client-side algorithms Conventional, PANDA, and Elastic on the performance of the BR approach. We consider the performance of the client-side algorithms in the *baseline* case where no VCP is used as a term of comparison. Figure 15 (a) shows the RMSE when the arrival rate λ is set to 0.08. The RMSE is insensitive to the employed client-side algorithm regardless the VCP is used or not. However, other QoE-related parameters are impacted by it.

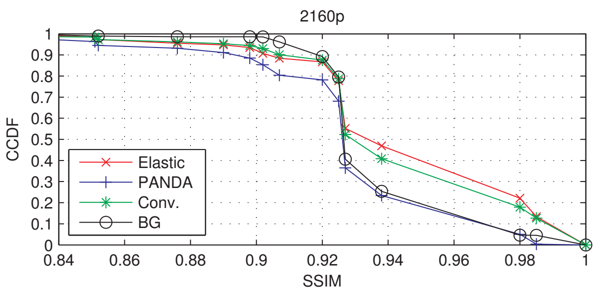
To the purpose, let us consider the CCDFs of the download rate and the SSIM video sessions ($\lambda = 0.08$). In Figure 16 we show only the case of 2160p sessions since performance differences are more remarkable. The figure shows the complementary CDFs obtained with BG as a term of comparison since it does not employ a client side bitrate adaptation (see Section 2). Let us focus on Figure 16 (a). The first important difference is that when Conventional and Elastic are used with the BR strategy they always provide a higher bandwidth share to 2160p video sessions with respect to the case in which BG is used. In particular, the median for Elastic, Conventional, and BG are roughly 3 Mbps, 2.8 Mbps, and 2.3 Mbps respectively. On the other hand, the advantage provided by BR is not exploited by PANDA which provides a lower bandwidth share to 2160p users with respect to BG. This issue is due to the fact that PANDA is slow in tracking the time-varying available bandwidth [7]. This performance loss is reflected in the obtained SSIM shown in Figure 16 (b). In particular, even though the medians of SSIM are roughly equal due to the fact that the measured RMSE are similar (Figure 15 (a)), it is clear that Elastic provides the best results whereas PANDA obtains the same SSIM as BG.

To conclude, we analyze the performance in terms of Switching Frequency obtained by the considered client-side algorithm. Figure 15 (b) shows that Conventional has 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 3 video segments. PANDA and Elastic, instead, provide similar results and show a slight increase of the switching frequency due to the BR.

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 compared to PANDA. Finally, Conventional provokes a high switching frequency that might be detrimental for QoE.

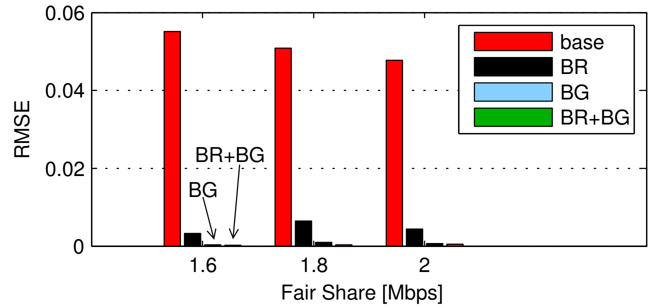


(a) Download rate

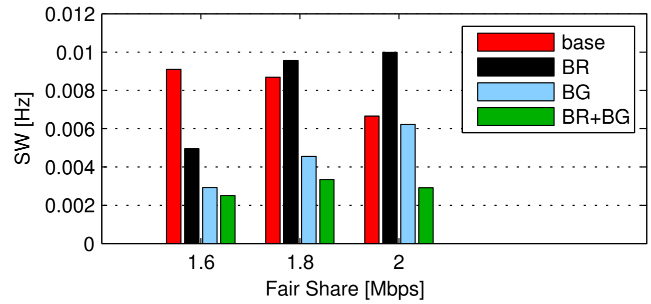


(b) SSIM

Figure 16: Complementary CDFs of download rate and SSIM with different client-side algorithms



(a) RMSE



(b) Switching Frequency

Figure 17: RMSE and Switching Frequency in the case of 5 video flows with per-flow queuing

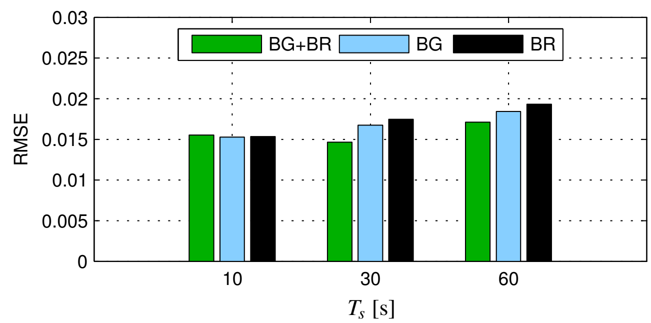


Figure 18: The impact of the Optimization Module sampling time T_s on the RMSE

4.3.3 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 3 in the case of the BR and the BG+BR approaches. In this scenario 5 flows are generated according to a Poisson arrival process. In order to consider several link loads, the link capacity has been set to 8, 9 and 10Mbps, corresponding to a bandwidth fair share of 1.6, 1.8, and 2 Mbps respectively.

In Figure 17 (a) and (b) the RMSE and the Switching Frequency are shown. If on one hand the RMSE in the *baseline* case is comparable to the one obtained in the other scenario (see Figure 11), 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 a 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 with respect to the other scenario. This improvement is due to the fact that in this scenario a per-flow queuing is used and thus video sessions do not share the same slice.

Summary: This section shows that Video Quality Fairness and Switching Frequency is improved when network-assisted strategies can be implemented with per-flow queuing.

4.3.4 The impact of the sampling time

In this Section we assess the impact of the sampling time T_s, at which the optimization is executed, on the performance of the VCP. In Figure 18 the RMSE obtained with BR, BG and BG+BR strategies is compared when T_s is set, respectively, to 10, 30 and 60s. It is worth noting that considering sampling times less than the video segment size is not meaningful since client-side adaptation algorithms are actuated on a per-segment basis. Figure 18 shows that the RMSE increases as the sampling time T_s increases, indicating that Video Quality Fairness degrades when large sampling times are used. The Switching Frequency is negligibly affected by T_s and due to space constraints is not shown.

4.3.5 Discussion

We conclude this Section by briefly discussing the overall characteristics of the considered strategies in the light of the experimental results presented above. For each of the considered approaches Table 3 provides a qualitative summary detailing the features of the considered approaches along with the corresponding obtained performances.

	Control Function.		Performance	
Approach	**Control Comm.**	**Bandwidth Slicing**	**VQF**	**Average Video Quality**
Baseline	-	-	Poor	Very good
BR	-	Yes	Good	Client dependent
BG	NC→Client	-	Very good	Quite Good
BG+BR	NC→Client	Yes	Very good	Quite Good

Table 3: Summary of the control functionalities and performance of the considered network-assisted approaches

BR requires no control communication after establishing the video session, which is beneficial in terms of scalability in the presence of a high number of 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. A drawback of this strategy is that it is sensitive to the client-side algorithm employed to select the bitrate, which can impact QoE-related metrics such as the Video Quality and the switching frequency.

On the other side, BG provides the best performance in terms of accuracy in enforcing the management policy at the expense of a higher amount of communication (a message to each active video client is sent each T_s seconds) and exposure of information reflecting the network state (i.e. the suggested bitrate). Even though reaching the client from the network can be challenging due to the fact that middleboxes have to be traversed, today such information exchange is made possible by employing WebRTC data channels[6]. Differently from BR, BG requires mutual trust between network and clients. Malicious clients, in fact, could obtain a higher bandwidth share by ignoring the suggested bitrate and selecting a higher one.

Finally, BG+BR provides no performance advantage compared to the BG strategy despite the higher control effort.

5. RELATED WORK

In the following we provide an overview of the literature focusing on the use of network-assisted approaches for the delivery of video content.

Bandwidth Reservation. The virtualization of the ISPs access infrastructure using open APIs supported through SDN is proposed in [21]. 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 [13] an SDN-based application-aware bandwidth allocation approach is used to maximize the QoE of YouTube flows. In [5] a control architecture and a reference implementation of a network control plane for video flows is proposed. The reference implementation is evaluated through numerical simulations. In [16] 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 [10] 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 [14] authors propose to place a HTTP proxy server between client and server (in the gateway or any another network device) in order to drive the bitrate adaptation of the players by rewriting their HTTP requests. An analytical model in the form of a Markov process is employed at the proxy to compute the bitrate for each player. The proposed model is experimentally validated. In [19] a rate adaptation algorithm is proposed to achieve 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 the bitrate adaptation is performed by the clients. The performance evaluation is carried out through ns-2 simulations.

C3. In [9] authors propose C3, a centralized control platform designed to optimize video delivery. The platform enables per-CDN real-time monitoring of the delivered video QoE, the prediction of expected performance and the selection of the CDN and the video bitrate. The bitrate selection in C3 is enforced by a centralized Decision Layer, which is aware of the performance of the transport networks thanks to the performance prediction made by an upper layer. The Decision Layer is also aware of the current client state thanks to the feedback provided by the client (Thin Client) on a second-level timescale (between 5 and 20s). It is worth to notice that such a control plane is made possible by the use of a large number of decision servers, which are hosted in geographically distributed front-end datacenters as close to the clients as possible.

Our contributions. Both the approaches considered in this paper, the Bandwidth Reservation (BR) and the Bitrate Guidance (BG) have been separately studied in the literature. At the best of authors knowledge, a study comparing these two strategies in the same scenario was still missing. Moreover, concerning the BR approach we have identified two limitations in the published studies: 1) only Home Network scenarios with very few video flows (not more than 6) sharing a common channel have been carried out; 2) the impact of the client-side algorithms on performance was not exposed in previous studies. This paper makes contributions in both the directions, by proposing a Video Control Plane for a more complex network setting and comparing the BR and BG approaches.

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. The impact of client-side adaptation algorithms has also been investigated. 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. Bitrate Guidance provides the best results in terms of VQF, whereas Bandwidth Reservation might improve the average video quality depending on the client-side algorithm. In particular, we have found that the VQF is not impacted by client-side algorithms, but other QoE related metrics, i.e. average video quality and the switching frequency, are affected. In particular, Elastic, and Conventional provide a higher average video quality compared to PANDA, whereas Conventional is affected by a large switching frequency. Finally, we have shown that when per-flow queuing is used, performance in terms of VQF and switching frequency is improved. Extensions of this work

[6] `http://www.w3.org/TR/webrtc/`

will address the design and evaluation of VCPs in multi-bottleneck network scenarios.

7. ACKNOWLEDGEMENTS

This work has been partially supported by the Italian Ministry of Education, Universities and Research (MIUR) through the MAIVISTO project (PAC02L1_00061). This work has been also partly funded by Deutsche Forschungsgemeinschaft (DFG) under grants HO 4770/1-2 and TR257/31-2. Any opinions, findings, conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect the views of the funding agencies.

8. REFERENCES

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

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

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

[4] Cisco. Cisco Visual Networking Index: Forecast and Methodology 2014-2019. May 2015.

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

[6] G. Cofano, L. De Cicco, and S. Mascolo. Characterizing adaptive video streaming control systems. In *Proc. American Control Conference*, pages 2729–2734, 2015.

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

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

[9] A. Ganjam, F. Siddiqui, J. Zhan, X. Liu, I. Stoica, J. Jiang, V. Sekar, and H. Zhang. C3: internet-scale control plane for video quality optimization. In *Proc. USENIX NSDI*, pages 131–144, 2015.

[10] P. Georgopoulos, Y. Elkhatib, M. Broadbent, M. Mu, and N. Race. Towards Network-wide QoE Fairness Using Openflow-assisted Adaptive Video Streaming. In *Proc. ACM SIGCOMM Workshop on Future Human-centric Multimedia Networking*, pages 15–20, 2013.

[11] T. Hoßfeld, R. Schatz, and U. R. Krieger. 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*, pages 136–150, 2014.

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

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

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

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

[16] A. Mansy, M. Fayed, and M. Ammar. Network-layer fairness for adaptive video streams. In *Proc. IFIP/IEEE Networking*, pages 42–48, 2015.

[17] M. K. Mukerjee, D. Naylor, J. Jiang, D. Han, S. Seshan, and H. Zhang. Practical, Real-time Centralized Control for CDN-based Live Video Delivery. In *Proc. ACM SIGCOMM*, pages 311–324, 2015.

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

[19] 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 Computer Communication Applications*, 12(2):28:1–28:24, 2015.

[20] 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 Communications Surveys and Tutorials*, 17(1):469–492, 2015.

[21] V. Sivaraman, T. Moors, H. Habibi Gharakheili, D. Ong, J. Matthews, and C. Russell. Virtualizing the Access Network via Open APIs. In *Proc. ACM CoNEXT*, pages 31–42, 2013.

[22] I. Sodagar. The MPEG-DASH Standard for Multimedia Streaming Over the Internet. *IEEE MultiMedia*, 18(4):62–67, 2011.

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

[24] C. Wilson, H. Ballani, T. Karagiannis, and A. Rowtron. Better never than late: Meeting deadlines in datacenter networks. In *Proc. ACM SIGCOMM*, pages 50–61, 2011.

[25] L. Zhi, 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 JSAC*, 32(4):719–733, 2014.

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