Quality of Experience-based Routing of Video Traffic for Overlay and ISP Networks

Giacomo Calvigioni, Ramon Aparicio-Pardo, Lucile Sassatelli
Universite Cote d'Azur, CNRS, I3S
Email: {calvigioni,raparicio,sassatelli}@i3s.unice.fr

Jeremie Leguay, Paolo Medagliani, Stefano Paris
France Research Center, Huawei Technologies Co. Ltd
Email: {name.surname}@huawei.com

***Abstract*—The surge of video traffic is a challenge for service providers that need to maximize Quality of Experience (QoE) while optimizing the cost of their infrastructure. In this paper, we address the problem of routing multiple HTTP-based Adaptive Streaming (HAS) sessions to maximize QoE. We first design a QoS-QoE model incorporating different QoE metrics which is able to learn online network variations and predict their impact on representative classes of adaptation logic, video motion and client resolution. Different QoE metrics are then combined into a QoE score based on ITU-T Rec. P.1202.2. This rich score is used to formulate the routing problem. We show that, even with a piece-wise linear QoE function in the objective, the routing problem without controlled rate allocation is non-linear. We therefore express a routing-plus-rate allocation problem and make it scalable with a dual subgradient approach based on Lagrangian relaxation where subproblems select a single path for each request with a trivial search, thereby connecting explicitly QoE, QoE and HAS bitrate. We show with ns-3 simulations that our algorithm provides values for HAS QoE metrics (quality, rebufferings, variation) equivalent to MILP and better than QoS-based approaches.**

I. Introduction

IP video traffic will represent 82% of all consumer Internet traffic by 2021, up from 73% in 2016 [1]. Therefore, delivering high-quality video services is a key challenge for Internet Service Providers (ISPs). In this context, this paper studies the optimization of network resources in order to maximize the QoE (Quality of Experience) perceived by end-users. Such a QoE-based network optimization is also of high interest to Content Distribution Networks (CDNs) for the overlay network they operate [2].

The growing demand in video services has been enabled by HTTP-based streaming. Specifically, HTTP Adaptive Streaming (HAS) standardized into MPEG-Dynamic Adaptive HTTP Streaming (DASH) [3], splits the video into temporal segments, each available in different qualities, i.e. encoding rates. The quality of each segment (or chunk) to download can be chosen based on the network and client state. The manifold of existing DASH adaptation policies aim at providing the client with the best QoE while absorbing the network variations. The concept of QoE encompasses the metrics the viewer is directly sensitive to, in particular, as defined in [4]: the visual quality (for which metrics, possibly PSNR-based, exist such as SSIM or VQM [5]), the frequency and duration of re-buffering events (a.k.a. stalls or interruptions), the startup delay, the amplitude and frequency of quality variations. These metrics can be contradictory (e.g. a low startup delay may be achieved at the expense of a low buffer level and hence incur more stalls), and each adaptation logic has its own trade-off. These logics can however be cast into two major categories, called Rate-Based (RB) and Buffer-Based (BB, [6]) adaptations, which base their decisions on the sensed throughput or the buffer level. A number of hybrid approaches exist, possibly based on an explicit formulation of the optimization problem [4].

In this paper, we address the problem of resource sharing in routing for several unicast HAS sessions in an ISP or a CDN overlay network, to maximize the sessions' QoE while taking into account the specifics of HAS flows. While the problem of QoE-based routing has been investigated in some works targeted at wireless mesh or ad hoc networks (e.g., [7], [8]), the QoS-QoE model was not fitted for HAS flows in wired networks. In wired networks, the works aiming at improving the routing of HAS sessions (e.g., [9], [10]) mostly resorted to QoS-based optimization of path selection based on Lagrange Relaxation Based Aggregated Cost (LARAC)-like approaches [11].

- **First contribution**: We first design a QoS-QoE model incorporating different QoE metrics which is able to take into account the impact of network variations on HAS adaptation logics. This model is able to learn online network variations, and predicts their impact on three representative classes of adaptation logics (RB, BB or hybrid), video types (cartoon, low motion and high motion movies) and terminal resolutions (360p, 720p and 1080p). Different QoE metrics (not only the visual quality) are then combined into a QoE score based on ITU-T Rec. P.1202.2 [12], [13].

Additionally, a number of works (see [14] and references therein) have shown that the client-side adaptations are prone to incur unfairness, under-utilization and instability when HAS clients share a same bottleneck. Two main reasons for these phenomena are (i) the intricate interplay between the HAS and TCP control loops [15], and (ii) the fact that with a perfect TCP, the best one can hope for is a perfect bandwidth sharing, which does not correspond to a QoE fairness when certain sessions (e.g., with higher resolutions and/or motion) need more rate to achieve the same perceived quality [16]. Our model takes these two factors into account through the modeling of the TCP performance under HAS into the QoE function.

- **Second contribution**: We then consider the framework of

Network Utility Maximization [17] to formulate the routing problem. Owing to the efficiency of solving techniques for linear or integer linear programming, we aim at expressing the routing problem linearly. We show that, even with a piece-wise linear QoE function in the objective, the routing problem without controlled flow allocation and assuming perfect TCP has a non-linear expression. We therefore express a routing-plus-flow allocation problem as a Mixed Integer Linear Program (MILP). To ease the handling of massive video request arrivals, a Lagrangian decomposition is devised so that each request can be served immediately with a proper resource share. The Karush Kuhn Tucker (KKT) conditions of each primal subproblem allow to select a single path for each request as well as the optimal bandwidth allocation found with a simple dichotomic search based on the HAS representations. This allows to make explicit the connection between bandwidth, path delay, QoE and HAS bitrate.

• **Third contribution**: We carry extensive simulations with a centralized controller deployed within ns-3. We compare our QoE-based routing with MILP and Lagrangian relaxation to QoS-based routing, namely congestion minimization, and minimum delay routing. We show that our relaxed approach provides values for HAS QoE metrics (quality, re-bufferings, variation) equivalent to MILP and better than the QoS-based approaches, while maintaining a fast solving time. We consider both static and dynamic request arrivals, and study the sensitivity to re-optimization frequency.

The novelty of this work lies in building a routing strategy:

- on a refined QoS-QoE model able to incorporate rebufferings, visual quality, TCP defects under HAS, and learning of current bandwidth variations;
- which optimally computes the resource allocation instead of using approximation path search techniques based on reinforcement learning or QoS routing constraints.
- which is independent of fine-grained information of the client state, thus seamlessly working with standard HAS clients.

Relevant related works are discussed in Sec. II. The QoS to QoE model is detailed Sec. III. Sec. IV formulates the optimization model and derives the low-complexity Lagrangian relaxation explicitly connecting QoS metrics to path selection and flow allocation. Numerical assessments are shown in Sec. V. Sec. VI concludes this paper.

II. Related works

QoE-based routing has been first investigated within wireless networks. For wireless mesh networks, Matos et al. in [7] solve the problem of routing using reinforcement learning (Q-learning) to choose progressively the best path based on a mapping of downloading rate, loss rate and delay onto a QoE score, which only accounts for visual quality. Quang et al. in [8] consider ad hoc networks and a linearization of the Pseudo-Subjective Quality Assessment (PSQA) score to express QoE-based routing as a MILP and then derive a heuristic. PSQA is based on Random Neural Network predicting the mean opinion score.

More generally for wired networks, in [18] shortest path routing is performed with edge weights set as a combination of residual bandwidth and delay, to maximize the quality obtained by Scalable Video Coding (SVC) flows. Within an OpenFlow (OF) framework, different weights are used to route the base or improvement layers. The same idea is leveraged in [10], while [9], [19] and [20] consider constrained shortest path routing under several QoS constraints (bandwidth, path loss, delay, jitter), resort to LARAC path computations and show that QoS-based routing outperforms best-effort routing in terms of obtained bitrate and PSNR.

The above works however hardly take into account the specifics of HAS flows and their complex competition additionally impacted by their interplay with TCP. In [15], the impact of losses on TCP-carried HAS flows has been dissected to explain why the goodput TCP (Cubic) may be substantially below the available bandwidth (or throughput). The authors show that the ON/OFF behavior corresponding to successive segment transfers, generates regular bursty short-lived flows. Despite the connection is usually in persistent mode, the impact of the starting and ending phase, i.e., initial burst and ACK-trailing phase, can be substantial. Some solutions (such as [21], [22]) have been designed to alleviate these problems.

Leveraging on the centralized control and easy switch reconfigurations brought by the concept of Software Defined Networking (SDN), network assistance has been investigated to ensure quality-level fairness between contending HAS flows in a single bottleneck. In [16], the authors express the rate allocation problem to maximize the sum of the qualities over the different requested resolutions and under link capacity constraint. The SSIM metric is considered as quality and the flow rate is enforced at the OpenFlow switch by a weighted fair queuing. In [23], still assuming the bottleneck is the last access router, the authors devise a controller able to track the clients' buffers states and move the flow of the clients in danger of stalling to a high-priority queue. In both strategies above, only queuing is leveraged, without routing or client modification.

While the network has a general view and can provide bandwidth reservation at routers, clients are in charge of the final decision and can receive recommendations. Coupling network-assistance with coordination from the client is proposed within MPEG-DASH SAND. Examples of such proposals are [24]–[26]. However, these approaches have major drawbacks of (i) requiring client's logic modification or (ii) fine-grained client's state disclosure to the network controller. Having (i) may be difficult as the adaptation logic is often proprietary with the content provider not necessarily eager to modify its client application subject to several ISPs or telcos. Assuming (ii) on the other hand may be tricky owing to the ubiquitous use of HTTPS and encryption-based delivery (central in HTTP/2), as recently outlined in [27]. With these obstacles in mind, very recently, two major articles [28], [29] have investigated the trade-offs of having different levels of network-assistance, client state disclosure and client modification.

In our work, we design a refined QoE model incorporating different QoE metrics and able to learn the network variations

in order not to require client/network active cooperation yet accounting for the behavior of other clients sharing bottlenecks.

III. QoS-QoE model

Our goal is to optimize routing so that HAS sessions get the best QoE. Two fundamental questions therefore arise: (i) How to define the QoE score of a session, and (ii) How is QoE expressed as a function of the QoS parameters.

As we want to avoid the need for fine-grained knowledge of the clients' states (either bitrate decisions or buffer levels), we do not work with an optimization where the objective is only a function of the visual quality while the other metrics such as re-bufferings are considered as constraints (e.g., [28, Eq. 10]. Instead, we need a function incorporating the different QoE metrics into a score, to be truly reflective of the user experience of watching an HAS video stream. The QoE score must be a function of the optimization variable, i.e., the allocated bandwidth over the selected path. This function must be parameterized based on the characteristics of the video streams that we can assume accessible.

Fig. 1 presents an overview of the QoS-QoE model we used in the QoE-based routing algorithm we developed in Sec. IV. The next paragraphs describe all the steps in details.

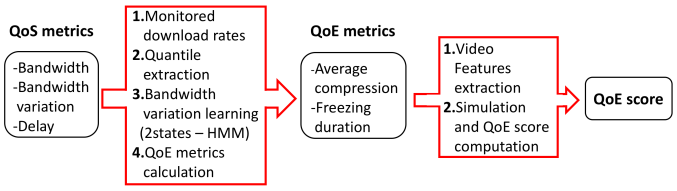


Fig. 1. QoS-QoE model

Disclosure assumptions: In this work, we assume the video type (cartoon, low motion or high motion), player adaptation (RB, BB or hybrid) and resolution can be known upon the session launch. Owing to the generalized use of HTTP over TLS [30], a network operator not providing the content itself is blind to these data, except for the adaptation which we assume can be inferred from the provider. Then we assume a system such as cDVD [27] to provide the network with this information. When we consider the problem of overlay routing in a CDN however, the CDN is the physical provider of the content, therefore has visibility into the HTTPS content established with one of its own servers.

From the QoE metrics to the QoE score: First, the concept of QoE for streamed videos has been investigated in numerous works, both in the networking community [31] and in the multimedia community (within the Video Quality Expert Group - VQEG [32]). A number of user studies have aimed at determining a QoE score (like Mean Opinion Score - MOS) from these QoE metrics (stated in Sec. I). For instance [24], [31] considers a weighted linear combination. More refined QoE models have been proposed in the multimedia community, often based on supervised machine learning models. However, to carry out a network optimization based on such a model, a strong constraint is that we need an explicit function connecting the decision variables with the end QoE score, therefore preventing from using, e.g., artificial neural networks or decision trees [31], [33]. We have therefore chosen the reference model of ITU-T Rec. P.1202.2 [12], [13], which incorporates the exponential relations of QoE to stalls and encoding rate into an explicit log-logistic regression model. The QoE for each user n, given the allocated bandwidth, writes as follows:

$$QoE_n = \frac{1}{1+\alpha\left(a_{c_n} z_{c_n}^{b_{c_1}} x_{c_n}^{b_{c_0}} + a_{f_n} z_{f_n}^{b_{f_1}} x_{f_n}^{b_{f_0}}\right)^{\beta}} \quad (1)$$

where x_c and x_f are respectively the average compression of the video and the total freezing duration (i.e., main factors), z_c is the content unpredictability, z_f is the motion homogeneity (i.e., two co-factors), and a, b, α, and β are parameters used to maintain the logistic shape of the curve. These parameters are calculated using a set of pre-distorted videos (see [13]).

From the QoS metrics to the QoE metrics: Second, the QoE metrics, namely the average compression and freezing duration, must be expressed from the decision variable, that we choose as the average bandwidth allocated to the session. We make two main observations: (i) for the streaming of stored videos, the major factor impacting QoE is the variations of available bandwidth during the streaming [4], and (ii) the QoE metrics obtained depend on the player's adaptation mechanism, which mainly aims at absorbing the bandwidth variations. Predicting the QoE metrics obtained from allocating a certain average bandwidth therefore requires a network variation model, detailed below. Once the variation model is learned, each point on the QoE score versus average bandwidth curves is obtained by modeling the performance of a 100 chunks video session undergoing the bandwidth variation generated from the learned model, for each combination of video type (e.g., high motion, slow motion), client resolution (e.g., 720p, 1080p) and HAS adaptation logics (e.g., RB, BB). Fig. 2 depicts such curves for adaptation logics from the extended version of LibDASH [1] of the Adaptive Multimedia Streaming Simulator Framework (AMust) [34] and 3 reference videos.

More details on the calculation of the QoE score and QoE metrics can be found in the following technical report [35]. The reader can find how we calibrated the model on 3 open movies commonly used for testing video codecs and recommended by the DASH Industry Forum. The technical report also describes how we linearize QoE functions for the sake of the optimization in Sec. IV.

Learning the bandwidth variation model: Estimating the network parameters to optimize multimedia transmission has been investigated in different contexts. For instance, the selection of the right Forward Error Correction (FEC) overhead under varying path dynamics was studied in [36], while in [37] path selection strategies are shown to be best for QoE (VQM)

[1] https://github.com/ChristianKreuzberger/AMuSt-libdash

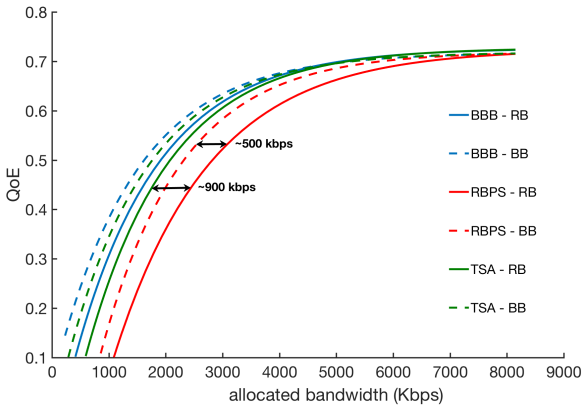


Fig. 2. Examples of QoE extracted for 3 videos (namely BBB, TSA and RBPS - see Sec. V) and 2 adaptation logics, Rate Based (RB) and Buffer Based (BB) at resolution 1080p under high bandwidth variations ($\pm 20\%$ around the mean).

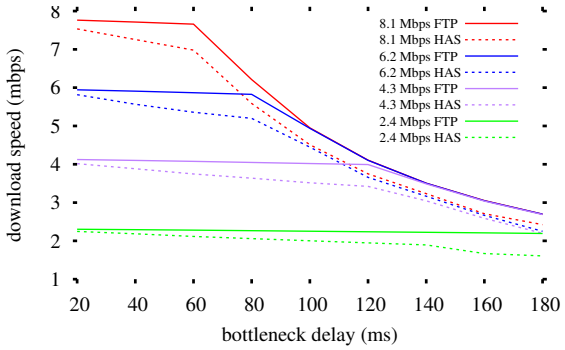


Fig. 3. TCP Cubic performance in ns-3 with FTP transfers and HAS flows, for several bottleneck capacities and end-to-end one-way latencies.

when the considered bandwidth is the 10th percentile rather than the time average.

We therefore need a model generating bandwidth samples for each average bandwidth value, that reflects enough the current dynamics in the network while being simple enough to be learned easily. We consider as generating Markov model, as various studies (e.g., [38]) showed that the pattern of Internet packet loss can be captured by Markov models. If other studies have shown that more refined or other models are more accurate [39], to keep our model simple we opt for a two-state hidden Markov model (HMM) for the bandwidth variations, with states C (Congestion) and NC (Non-Congestion). In each state, we consider 2 percentile thresholds p_{low} and p_{high}. We leverage network monitoring to periodically analyze the bandwidth obtained for the download of chunks in different video sessions, each sample being replaced by 1, 2 or 3 depending on its position w.r.t. p_{low} and p_{high} The complete parameters of the corresponding HMM are then learned with a classical HMM training method (we use the Baum-Welch algorithm) on the aggregate of the collected traces.

Modeling TCP performance with HAS: It has been shown in [15] that HAS TCP flows behave as short-lived TCP flows (with some specificities, such as no systematic reset of the window). The TCP throughput formula is not accurate to model (i) this behavior and (ii) Cubic (as it was designed for AIMD versions of TCP). The ratio of goodput-to-throughput obtained by a single HAS flow has been shown to depend heavily on the RTT (owing to the prominence of the losses in the initial and ending phase) [15, Fig. 1]. In order to estimate this ratio in a bandwidth reservation setting we are considering, the TCP Cubic code from [40] has been ported in ns-3 (the simulator used in Sec. V), and the downloading rates simulated for various reserved bandwidth and end-to-end latencies (one-way), both for long-lived FTP transfers and HAS transfers, as depicted in Fig. 3. Let us specify that the drop in utilization after a certain latency threshold cannot be accounted for by the Linux TCP Cubic algorithm, but is rather due to this specific implementation. We therefore model the goodput y^d served by TCP to a HAS demand d as a ratio of the available bandwidth x^d. This ratio decreases proportionally with the path round trip delay RTD_p. We therefore model this relationship as $y^d = \alpha_p(RTD_p) \cdot x^d$, where $\alpha_p(RTD_p)$ is a coefficient (between 0 and 1) computed from the RTD of path p (the smaller the RTD is, the closer to 1 α is). Specifically, we extract for this ns-3 platform (can be extracted from measurements on any other platform): $\alpha_p = -1.279.10^{-3} + 1.0011 RTD_p$.

IV. Control plane for QoE-based routing

We formulate the problem of routing several HAS demands to maximize the sum of their QoE (additional inclusion of a fairness function is straightforward [28]). We first show that the problem, named Maximal HAS QoE-based Routing (Max-HQR), where the flow rates are not controlled but determined by a perfect TCP fair-share of bottleneck capacity is not linear. It hence cannot be efficiently solved with Mixed Integer Linear Programming (MILP) solvers. We therefore first propose a MILP reformulation by introducing the bandwidth allocation decisions. To make the problem scalable, we then derive a Lagrangian relaxation based on a dual sub-gradient approach, which we thoroughly analyze and which allows to make explicit the connection between QoS and QoE in path selection and bandwidth allocation.

A. System assumptions

Let $\mathcal{G}(\mathcal{N}, \mathcal{E})$ be a network graph, where $\mathcal{N}$ is the set of nodes and $\mathcal{E}$ is the set of unidirectional links. The capacity of link $e \in \mathcal{E}$ is c_e traffic units (i.e. $Mbps$). We denote by $\mathcal{D}$ the set of all (unicast) video HTTP Adaptive Streaming (HAS) demands d, served from source node $a(d)$ to end node $b(d)$. Each $d \in \mathcal{D}$ has a Quality of Experience (QoE) (or utility) function $\mathcal{U}_d(y_d)$, parameterized by the video type, resolution and adaptation logic of d and depending on the goodput y_d TCP effectively serves HTTP with. We assume that the utility function $\mathcal{U}_d(.)$ is strictly concave in y_d, for all $d \in \mathcal{D}$ (see

TABLE I
INPUT PARAMETERS

Name	Description
$a_k^d \in \mathbb{R}_{\geq 0}$	Slope of utility piece $k \in \mathcal{K}_d$ of demand $d \in \mathcal{D}$
$b_k^d \in \mathbb{R}_{\geq 0}$	y-intercept of piece $k \in \mathcal{K}_d$ of demand $d \in \mathcal{D}$
$r_d^{min} \in \mathbb{R}_{\geq 0}$	Min. rate of DASH manifest for $d \in \mathcal{D}$
$r_d^{max} \in \mathbb{R}_{\geq 0}$	Max. rate of DASH manifest for $d \in \mathcal{D}$
$c_e \in \mathbb{R}_{\geq 0}$	Capacity of link $e \in \mathcal{E}$
$C \in \mathbb{R}_{\geq 0}$	Reference link capacity

TABLE II
DECISION VARIABLES

Name	Description
$u^d \in \mathbb{R}_{\geq 0}$	Utility value (QoE) of demand $d \in \mathcal{D}$
$x_p^d \in \mathbb{R}_{\geq 0}$	Bandwidth on path $p \in \mathcal{P}$ to serve $d \in \mathcal{D}$
$z_p^d \in \{0,1\}$	1, if path $p \in \mathcal{P}$ serves $d \in \mathcal{D}$. 0, otherwise.
$z_{dpi} \in \{0,1\}$	1, if $i \in \mathbb{Z}_{>0}$ competitive videos share the path $p \in \mathcal{P}$ serving $d \in \mathcal{D}$. 0, otherwise.
$n_e \in \mathbb{Z}_{\geq 0}$	N of competitive videos sharing link $e \in \mathcal{E}$

Sec. III). The set of candidates paths p is denoted as $\mathcal{P}$. The set of paths traversing a link $e \in \mathcal{E}$ is denoted as $\mathcal{P}_e$. The set of paths between $a(d)$ and $b(d)$ is denoted as $\mathcal{P}_d$, for all $d \in \mathcal{D}$.

B. Combinatorial problem formulation

We first express Max-HQR as Eq. 2 where we assume that TCP fairly shares the bandwidth between demands on the same bottleneck, i.e., $x_d = c_e/i$, where c_e is the capacity of the bottleneck link e and $i \in \mathbb{Z}_{>0}$ is the number of competing TCP flows. Hence we define a set of QoE values normalised with respect to $C = \min c_e$, such as $U_{dpi} = \mathcal{U}_d\left(\alpha_p \cdot (C/i)\right)$ for each triplet ($d \in \mathcal{D}, p \in \mathcal{P}_d, i \in \mathbb{Z}_{>0}$). Tables I and II gather the notations.

$$\max_{\{\boldsymbol{z},\boldsymbol{n}\}} \sum_{\substack{d\in\mathcal{D}\\ p\in\mathcal{P}_d\\ i\in\mathbb{Z}_{>0}}} U_{dpi} \cdot z_{dpi} \tag{2a}$$

$$\text{s.t.} \sum_{p\in\mathcal{P}_d} z_p^d = 1, \qquad d \in \mathcal{D} \tag{2b}$$

$$\sum_{\substack{d\in\mathcal{D}\\ p\in\mathcal{P}_e}} z_p^d = n_e, \qquad e \in \mathcal{E} \tag{2c}$$

$$z_p^d \cdot n_e \leq \frac{c_e}{C} \cdot \sum_{i\in\mathbb{Z}_{>0}} i \cdot z_{dpi}, \quad d \in \mathcal{D}, p \in \mathcal{P}_d, e \in \mathcal{E}_p \tag{2d}$$

$$\sum_{i\in\mathbb{Z}_{>0}} z_{dpi} = 1, \qquad d \in \mathcal{D}, p \in \mathcal{P}_d \tag{2e}$$

$$z_p^d \in \{0,1\}, \qquad d \in \mathcal{D}, p \in \mathcal{P} \tag{2f}$$

$$n_e \in \mathbb{Z}_{\geq 0}, \qquad e \in \mathcal{E} \tag{2g}$$

$$z_{dpi} \in \{0,1\}, \qquad d \in \mathcal{D}, i \in \mathcal{I} \tag{2h}$$

Constraints (2b) force path uniqueness, (2c) count the number of competing videos flows in each link, (2d) and (2e) identify the number of flows on the bottleneck of demand d. Constraints (2d) are not linear since we need to know the exact routing of d to compute its bottleneck load. A classical Integer Linear Programming (ILP) solver is therefore not suitable to address this problem.

C. MILP problem formulation

This subsection reformulates the Max-HQR problem as a MILP model (3) by adding the bandwidth allocation to the routing decisions. The QoE utility $\mathcal{U}_d(.)$ is approximated by a piecewise-linear function made of a set $\mathcal{K}_d$ of pieces (see Sec. III).

$$\max_{\{\boldsymbol{x},\boldsymbol{z},\boldsymbol{u}\}} \sum_{d\in\mathcal{D}} u_d \tag{3a}$$

$$\text{s.t.} \sum_{\substack{d\in\mathcal{D}\\ p\in\mathcal{P}_e}} x_p^d \leq c_e, \qquad e \in \mathcal{E} \tag{3b}$$

$$a_k^d \left(\sum_{p\in\mathcal{P}_d} \alpha_p \cdot x_p^d \right) + b_k^d \geq u_d \qquad d \in \mathcal{D}, k \in \mathcal{K}_d \tag{3c}$$

$$\sum_{p\in\mathcal{P}_d} \alpha_p \cdot x_p^d \geq r_d^{min} \cdot z_p^d \qquad d \in \mathcal{D} \tag{3d}$$

$$\sum_{p\in\mathcal{P}_d} \alpha_p \cdot x_p^d \leq r_d^{max} \cdot z_p^d \qquad d \in \mathcal{D} \tag{3e}$$

$$\sum_{p\in\mathcal{P}_d} z_p^d = 1 \qquad d \in \mathcal{D} \tag{3f}$$

$$x_p^d \in \mathbb{R}_{\geq 0}, \qquad d \in \mathcal{D}, p \in \mathcal{P} \tag{3g}$$

$$z_p^d \in \{0,1\}, \qquad d \in \mathcal{D}, p \in \mathcal{P} \tag{3h}$$

$$u_d \in \mathbb{R}_{\geq 0}, \qquad d \in \mathcal{D} \tag{3i}$$

Constraints (3b) ensure that link capacities are not violated, (3c) model the utilities linearization, (3d) and (3e) limit the allocated bandwidth to a range depending on TCP performance (α_p) and the minimum and maximum bitrates of the DASH representations, and (3f) force path uniqueness. This MILP is a Single Path Allocation problem, proven to be $\mathcal{NP}$-complete in [41], and therefore not tractable for large instances. To remedy this limitation, we propose a Dual subgradient based on Lagrangian relaxation (DGLR) algorithm to devise an online control procedure able to scale with the problem size.

D. Dual subgradient based on Lagrangian relaxation (DGLR)

We consider the partial Lagrangian function obtained by relaxing constraint (3b).

$$L(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\lambda}) = -\sum_{d\in\mathcal{D}} u_d + \sum_{\substack{d\in\mathcal{D}\\ p\in\mathcal{P}_d}} \lambda_p \cdot x_p^d - \sum_{e\in\mathcal{E}} \lambda_e \cdot c_e \tag{4}$$

where $\lambda_e, e \in \mathcal{E}_p$ are the Lagrangian multipliers associated to Eq. (3b) and $\lambda_p = \sum_{e\in\mathcal{E}_p} \lambda_e$, for all $p \in \mathcal{P}$, are the "synthetic" Lagrangian multipliers associated to path $p \in \mathcal{P}$. The dualization of Eq. (3b) yields the so-called Lagrangian

Relaxed Problem (5) where the Lagrangian function is minimized subject to the non-dualized constraints:

$$D(\boldsymbol{\lambda}) = \min_{\substack{\{\boldsymbol{x},\boldsymbol{u}\} \\ \text{s.t. (3c),(3d),(3e),(3f)}}} L(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\lambda}) \tag{5}$$

where $D(\boldsymbol{\lambda})$ is the dual function, with the dual problem expressed as:

$$\max_{\{\boldsymbol{\lambda}\}} D(\boldsymbol{\lambda}) \tag{6a}$$

$$\text{s.t. } \lambda_e \in \mathbb{R}_{\geq 0}, \qquad e \in \mathcal{E} \tag{6b}$$

The exploration of the dual variable space can be carried out through a heuristic subgradient optimization method (the dual function is concave, see Sec. IV-D2). Therefore, the overall method is based on successive iterations consisting of a minimization step of the Lagrangian Relaxed Problem (*primal step*), and a maximization step of the Dual Problem via a subgradient optimization (*dual step*).

1) Primal step: Problem (5) can be clearly separated into a set of independent subproblems for each video demand $d \in \mathcal{D}$ which can be computed in parallel:

$$\max_{\{\boldsymbol{x_d},\boldsymbol{z_d},u_d\}} u_d - \sum_{p\in\mathcal{P}_d} \lambda_p \cdot x_p^d \tag{7a}$$

$$\text{s.t. (3c) to (3i) applied to } d \text{ only} \tag{7b}$$

where $\boldsymbol{x_d} = \left\{x_p^d, p \in \mathcal{P}\right\}$ and $\boldsymbol{z_d} = \left\{z_p^d, p \in \mathcal{P}\right\}$ is the set of bandwidth allocations and routing path selection, respectively, among the paths $p \in \mathcal{P}$ to serve demand d. These subproblems are still modeled with ILP, but they can be trivially solved without resorting to an ILP solver (like CPLEX) as shown below.

Let us consider the path selection. Let $p^* \in \mathcal{P}_d$ be an optimal path for demand d (then, $x_p^d = z_p^d = 0, p \in \mathcal{P}_d \setminus p^*$). By multiplying objective (7a) and constraints (3c), (3d), (3e) by a constant $Q = \lambda_{p^*}/\alpha_{p^*}$, and making the variable change: $u'_d = Q \cdot u_d$ and $x_p'^d = \lambda_p \cdot x_p^d, p \in \mathcal{P}_d$, we see that any path $p \in \mathcal{P}_d \setminus p^*$ candidate to be optimal with a ratio $\lambda_p/\alpha_p > Q$ worsens the solution. Therefore, the optimal path(s) $p^* \in \mathcal{P}_d$ correspond(s) to that (those) with the smallest quotient $Q = \lambda_{p^*}/\alpha_{p^*}$.

Let us now consider the bandwidth allocation. To find the optimal flow value x_p^{*d} for the optimal path p^*, we introduce the solution $x_p^d = z_p^d = 0, p \in \mathcal{P}_d \setminus p^*$ in the subproblems (7), leading to a simpler Linear Programming (LP) problem. Then, we carry out an analysis of the optimal basis of this LP problem. The standard form of the problem adds $|\mathcal{K}_d| + 2$ slackness variables to the constraints (3c), (3d) and (3e) to get equality constraints. The overall number of variables is hence $|\mathcal{K}| + 4$ (one u_d variable plus the non-null x_p^{*d} variable plus $|\mathcal{K}_d| + 2$ slackness variables) and the number of basic variables corresponds to the number of constraints: $|\mathcal{K}_d| + 2$. If the optimal flow value is neither at the minimal bit rate r_d^{min} nor at the maximal bit rate r_d^{max}, it is at the middle of the piecewise linearization between r_d^{min} and r_d^{max}. In such a case, at least four variables are larger than zeros (i.e. they are in the basis): the two slackness variables associated to constraints (3d) and (3e), the u_d variable and the x_p^{*d} variable, which implies still room for $|\mathcal{K}_d| + 2 - 4 = |\mathcal{K}| - 2$ basic variables. From the piecewise linearization, we know that optimum has to be either in the intersection of two straight-line sections ($|\mathcal{K}_d| - 2$ non-null slackness variables associated to constraints (3c)), or at the middle of a straight-line section ($|\mathcal{K}_d| - 1$ non-null slackness variables associated to constraints (3c)). Obviously, only the first situation is possible. Therefore, in this case, optimal bandwidth allocation and path routing is at one of the intersection points between straight-line sections.

From the study of the dual problem of these LP problems and the KKT optimality conditions, we can draw the same conclusions about their optimality, that we sum up here:

1) **Path selection**. One optimal path is a path p^* with the smallest quotient $Q = \lambda_{p^*}/\alpha_{p^*}$.
2) **Bandwidth allocation**. The optimal flow can be found by comparing the minimal ratio $Q = \lambda_{p^*}/\alpha_{p^*}$ with the slopes $a_k, k \in \mathcal{K}_d$.

These conclusions are the basis to build the trivial Algorithm 1 that optimally solves the independent subproblems.

QoS-QoE connection: Therefore the allocated bandwidth allowing to reach, under the learned bandwidth variations, a certain encoding rate while ensuring a certain level of QoE (hence rebufferings) is chosen at the point where the ratio between congestion and delay can be the tangent's slope of the QoE curve. The lower the ratio, the higher the allocated bandwidth.

Path Selection: In classical networking algorithms where the optimal λ_e multipliers are used as edge weights, any efficient Shortest Path Algorithm is able to find the shortest (lightest) path, without knowing the whole path set. Unfortunately, we cannot use this strategy since our optimal path is the path p with the smallest ratio λ_p/α_p, which means that α_p must be known in advance. We resort to a K-shortest paths algorithm with link RTDs as edge weights to find the paths with highest α_p. Among the K paths, the one with the smallest ratio λ_p/α_p is selected. If several paths have the same ratio λ_p/α_p, the path with the highest α_p is selected. While this approach does not guarantee finding the optimal path, it is likely to be sufficient in practical networking scenarios where link congestions (λ_e) and RTDs (α_p) are correlated.

Bandwidth allocation: The key is to compare the minimal ratio $Q = \lambda_{p^*}/\alpha_{p^*}$ with the slopes $a_k, k \in \mathcal{K}_d$, as detailed in Algo 1. The exploration of the $|\mathcal{K}_d|$ straight line sections requires $\mathcal{O}(log_2(|\mathcal{K}_d|))$ comparisons with the dichotomic search. For comparison, standard Simplex algorithm will perform: (i) *an initial phase to build an initial feasible basis* composed by at least one Simplex iteration, and (ii) *a main phase* composed by $[1, |\mathcal{K}_d|]$ Simplex iterations. The complexity order of a Simplex iteration can be estimated as $\mathcal{O}(mn) = \mathcal{O}(|\mathcal{K}_d|^2)$ (where $m = |\mathcal{K}_d| + 2$ is the number of constraints, and $n = |\mathcal{K}_d| + 4$ is the number of variables in standard form). Therefore, a classical dichotomic search for this problem

(when optimal path p^* is known) is more efficient than the Simplex method.

2) Dual step: The solution space of the Dual Problem (6) is explored using a subgradient descent algorithm as:

$$\lambda_e(t+1) = \left[\lambda_e(t) + \gamma \left(\sum_{\substack{d \in \mathcal{D} \\ p \in \mathcal{P}_e}} x_p^{*d}(t) - c_e \right) \right]_0 \quad (8)$$

where t is the iteration index of the primal-dual iteration. These multipliers reflect the link congestions: λ_e increases if the link load exceeds the nominal link capacity.

Dual-primal iterations: the multipliers update (8) (i.e. the *dual step*) is traditionally only performed after solving all the $|\mathcal{D}|$ *primal step* subproblems, that is after finding the bandwidth allocation for all demand $d \in \mathcal{D}$. In this work, we run multipliers update between two *primal* subproblems corresponding to two consecutive video demands (sorted by their arrival times). The rationale behind that is to promote that demands with common source-destination pairs (i.e. the same set $\mathcal{P}_d$) but different utilities $\mathcal{U}_d$ take different optimal paths (the minimal $Q = \lambda_{p^*}/\alpha_{p^*}$ trivially changes between two consecutive *primal step* subproblems). We therefore consider a main iteration as $|\mathcal{D}|$ alternations between a *primal step* and a *dual step*.

Stopping criterion: after a minimal number of main iterations (N_{min}), iterations are stopped if the ratio of change over the last N_{last} iterations is below a pre-defined threshold. If not, the algorithm is stopped after a maximal number of iterations (N_{max}).

Algorithm 1: Bandwidth allocation for demand d

Data: Path set $\mathcal{P}_d$ with ratios λ_p/α_p
Straight line sections $\mathcal{K}_d$ with slopes a_k and the bandwidth corresponding to the intersection points.
Bandwidth r_d^{min} and r_d^{max} corresponding to minimum and maximum bitrates, respectively.
Result: Optimal value of x_p^{*d}
Path selection: Find the path $p \in \mathcal{P}_d$ with the minimum ratio λ_p/α_p;
Bandwidth allocation: Explore the linear pieces $\mathcal{K}_d$:
if $\lambda_p/\alpha_p \geq a_0$ **then**
  $x_p^{*d} = r_d^{min}$;
else if $a_{|\mathcal{K}_d|-1} \geq \lambda_p/\alpha_p$ **then**
  $x_p^{*d} = r_d^{max}$;
else
  $t = 1$;
  $i(t) = 0$;
  $j(t) = |\mathcal{K}_d| - 1$;
  while $i(t) \neq j(t)$ **do**
    if $a_{i(t)} \geq \lambda_p/\alpha_p \geq a_{j(t)}$ **then**
      $i(t) = i(t-1)$;
      $j(t) = floor\{j(t-1)/2\}$;
      $x(t)_p^d$ is the bandwidth at the intersection of the pieces $i(t)$ and $j(t)$;
      $t = t + 1$;
    else
      $i(t) = ceil\{j(t-1)/2)\}$;
      $j(t) = j(t-1)$;
      $t = t + 1$;
  $x_p^{*d} = x(t)_p^d$;

Complexity: Each main iteration of the DGLR algorithm ($|\mathcal{D}|$ *primal-dual* alternations) has complexity $O\left(|\mathcal{D}|^2|\mathcal{N}|^2K\right)$, where the complexity, for each demand, of one *primal-dual* alternation is dominated by the dual step and equal to $O\left(|\mathcal{N}|^2|\mathcal{D}|K\right)$. The path selection of the primal step in a *primal-dual* alternation, consisting of finding the path out of $K = |\mathcal{P}_d|$ with minimum ratio $\frac{\sum_{e \in p} \lambda_e}{\alpha_p}$, requires at most $\mathcal{O}\left(|\mathcal{N}|K\right)$ operations (where the longest path in the network has $|\mathcal{N}| - 1$ edges). Once such a path has been found, the flow allocation requires $\mathcal{O}(log_2(|\mathcal{K}_d|))$ comparisons. In contrast, the dual step, consisting of the Lagrangian multiplier update for each edge of the network according to Eq. (8), requires the sum over all demands and over all pre-computed paths of the allocated flow: $\mathcal{O}\left(|\mathcal{E}||\mathcal{D}|K\right)$ operations. Since K, $\mathcal{K}_d$ are constants, and $|\mathcal{E}| \leq |\mathcal{N}^2|$, the complexity of one *primal-dual* alternation is basically the dual step complexity: $O\left(|\mathcal{N}|^2|\mathcal{D}|K\right)$.

V. Simulation results

This section presents extensive simulations results produced in a fully controllable simulation environment at network and HAS streaming levels.

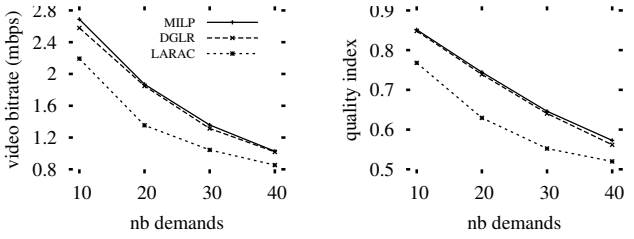
Evaluation methodology. Our simulation platform is based on the Adaptive Multimedia Streaming Simulator Framework (AMust) [34] in ns-3 which implements an HTTP client and server for LibDASH, one of the reference software of ISO/IEC MPEG-DASH standard. We extended the simulator with SDN capabilities to finely control routes for each video session. The routing module is implemented in Matlab and called when a video session starts or the network needs to be re-optimized. The source code of our simulation platform is available in [42].

We compare our Lagrangian relaxation based algorithm (DGLR) to the MILP solved with CPLEX. As QoS routing benchmark against these QoE-based routing algorithms, we have used LARAC [11], a widely used algorithm to efficiently compute constrained minimum delay paths under QoS constraints (e.g., packet loss, jitter).

To compare the different approaches, we measure the following QoE metrics:

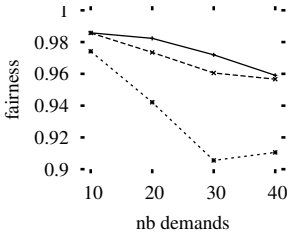
- *Average video bitrate* of the downloaded chunks.
- *Average video quality:* Average on all downloaded chunks of a normalized quality index indicating to which representation they belong. It evolves between 0 and 1 for r^{min} and r^{max}, respectively (see values in [35]).
- *Average quality fairness:* Jain's index over the average quality index of all video sessions. The index evolves in the interval $[0; 1]$ with 1 indicating perfect QoE fairness.
- *Average quality variation:* the standard deviation of the quality index which quantifies quality changes over the different downloaded chunks.
- *Re-buffering ratio*: freezing (or stalling) time over the duration of the video session.

Simulation setup. As streaming content, we have chosen 3 representative open movies commonly used for testing video

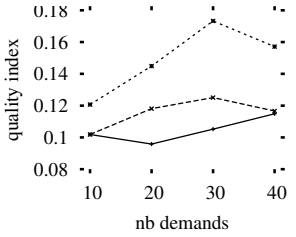


(a) Avg. video bitrate.

(b) Avg. video quality.

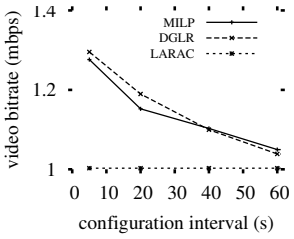


(c) Avg. quality fairness.

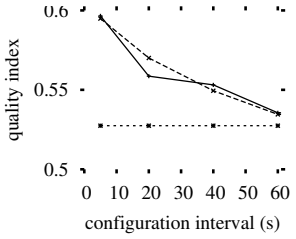


(d) Avg. quality variation.

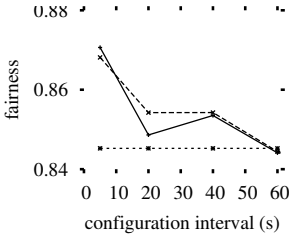
Fig. 4. Static traffic scenario on Geant.



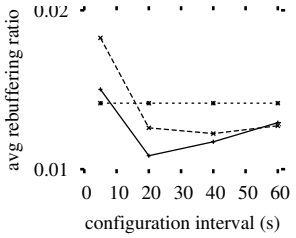
(a) Avg. video bitrate.



(b) Avg. quality.



(c) Avg. quality fairness.



(d) Avg. re-buffering ratio.

Fig. 5. Dynamic traffic scenario on Geant.

codecs and streaming protocols: Big Buck Bunny (BBB), a cartoon with a mix of low and high motion scenes, Swiss Account (TSA), a sport documentary with regular motion scenes and Red Bull Play Street (RBPS), a sport show with high motion scenes. More details on all the representations we used in simulations can be found in our technical report [35].

Our simulations are based on GEANT [43], an academic network composed by 22 nodes and 36 links. As packet-based simulations in ns-3 take a long time, we downsized the capacity of links to 10 Mbps. One-way latency of links are uniformly distributed in $[1, 10]$ ms. Regarding DASH clients, we set the video buffer size to 30s and picked at random the resolution in [360p, 720p, 1080p], the video type in [BBB, TSA, RBPS] and the adaptation logic in [BB, RB, hybrid]. We select one node as the HAS server and attach randomly clients to other nodes. All points in the following results are average over 5 simulation runs.

Simulation results. Fig. 4 shows results for *static* scenarios with a varying number of demands. The network is optimized once at the beginning and all streaming sessions start at the same time for a duration of 100s.

We observe that QoE-based approaches significantly improve user experience with respect to LARAC that considers only QoS metrics. Specifically, Fig. 4(a) shows that MILP and DGLR increase the average bitrate of HAS connections by up to 27% and 26%, respectively. The higher bitrate results in higher quality gains: up to 15% for MILP and 15% for DGLR as illustrated in Fig. 4(b). Furthermore, even without explicitly considering QoE fairness in our optimization, Fig. 4(c) shows improvement up to 6% and 5% of the average quality fairness. While QoE-based routing increases the video quality, it also helps to stabilize the quality as depicted by Fig. 4(d). All these results also show that our DGLR finds paths, almost as good as for MILP, which carefully load balance demands over the network considering the different representations that HAS clients can select.

We turn now our attention to the dynamic traffic scenarios with arrivals and departures of video sessions, which call for periodic re-optimization of the network. To this end MILP and DGLR are executed periodically by ns-3. We neglect the execution time of algorithms to accurately measure the QoE gain. For a fair comparison, the path selected at each session arrival is computed using LARAC (before re-optimization). In practice, we can use dual variables as link weights to compute a minimum cost path, but we leave this improvement for future work. We generated 60 demands with mean duration of 30s according to a Poisson process for a total simulation time of 160s. Fig. 5 shows performance results for different re-optimization intervals. We can observe that reconfiguring the network more often improves average video bitrate and quality (Fig. 5(a) and Fig. 5(b)). The re-buffering ratio is also improved but starts to increase when the configuration interval is set to 1 s (Fig. 5(d)). Indeed, modifying the routing for (potentially) all demands at a high frequency can lead to harmful throughput variations for DASH clients. As for the static scenario, MILP and DGLR improves also fairness although our QoE function does not handle it (Fig. 5(c)).

Convergence speed of DGLR. The constant stepsize γ used by DRLG accelerates the convergence but causes the oscillation around the optimal lower bound. Therefore, in order to shed light on the computational speed, we have analysed the number of iterations required to reach the neighborhood of the optimal lower bound (1%). Specifically, we varied the number of streams in $\{20, 40, 60, 80\}$, obtaining $\{8.6, 12.8, 5.4, 3.2\}$ iterations in average over 5 experiments. In practice, we can quickly stop DRLG after the very first iterations without incurring any significant performance loss.

VI. Conclusion

We addressed the problem of routing several HAS sessions to maximize QoE by taking into account the specific nature

of each video stream. We first design a QoS-QoE model incorporating different QoE metrics and able to take into account the impact of network variations on HAS adaptation logics. This model is able to learn online network variations, and predicts their impact on considered three representative classes of adaptation logic, video motion and client resolution. We express a routing-plus-rate allocation problem and make it scalable with a dual subgradient approach based on Lagrangian relaxation so that each request can be served immediately with a proper resource share. We show with ns-3 simulations that our relaxed approach provides values for HAS QoE metrics (quality, rebufferings, variation) equivalent to MILP and better than the QoS-based approaches.

We conjecture that DGLR and MILP can further improve QoE in networks where a minimum bandwidth can be guaranteed to each stream. Indeed, we did not enforced the bandwidth allocations decided by the algorithms and leave this extension, more complicated to deploy, for future work.

REFERENCES

[1] Cisco, "Cisco Visual Networking Index: Forecast and Methodology, 20162021," June 2017.

[2] R. K. Sitaraman, M. Kasbekar, W. Lichtenstein, and M. Jain, "Overlay networks: An Akamai perspective," *Advanced Content Delivery, Streaming, and Cloud Services*, vol. 51, no. 4, pp. 305–328, 2014.

[3] T. Stockhammer, "Dynamic Adaptive Streaming over HTTP: Standards and Design Principles," in *ACM MMSys*, 2011.

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

[5] "VQM software." [Online]. Available: http://www.its.bldrdoc.gov/n3/video/vqmsoftware.htm

[6] T.-Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson, "A buffer-based approach to rate adaptation: Evidence from a large video streaming service," *ACM SIGCOMM CCR*, vol. 44, no. 4, 2015.

[7] R. Matos, N. Coutinho, C. Marques, S. Sargento, J. Chakareski, and A. Kassler, "Quality of experience-based routing in multi-service wireless mesh networks," in *IEEE ICC*, June 2012, pp. 7060–7065.

[8] P. T. A. Quang, K. Piamrat, K. D. Singh, and C. Viho, "Video streaming over ad hoc networks: A qoe-based optimal routing solution," *IEEE Tran. on Veh. Tech.*, vol. 66, no. 2, pp. 1533–1546, Feb 2017.

[9] H. E. Egilmez, S. Civanlar, and A. M. Tekalp, "An Optimization Framework for QoS-Enabled Adaptive Video Streaming Over OpenFlow Networks," *IEEE Trans. on Multimedia*, vol. 15, no. 3, pp. 710–715, April 2013.

[10] A. Gangwal, M. Gupta, M. S. Gaur, V. Laxmi, and M. Conti, "Elba: Efficient layer based routing algorithm in sdn," in *IEEE ICCCN*, 2016.

[11] A. Juttner, B. Szviatovski, I. Mecs, and Z. Rajko, "Lagrange relaxation based method for the QoS routing problem," in *IEEE INFOCOM*, 2001.

[12] F. Zhang, W. Lin, Z. Chen, and K. N. Ngan, "Additive log-logistic model for networked video quality assessment," *IEEE Trans. on Image Proc.*, vol. 22, no. 4, pp. 1536–1547, April 2013.

[13] ITU-T, "Parametric non-intrusive bitstream assessment of video media streaming quality - Higher resolution application area ," 2013.

[14] J. Jiang, V. Sekar, and H. Zhang, "Improving fairness, efficiency, and stability in http-based adaptive video streaming with FESTIVE," in *ACM CoNEXT*, Dec. 2012.

[15] J. Esteban, S. A. Benno, A. Beck, Y. Guo, V. Hilt, and I. Rimac, "Interactions between HTTP adaptive streaming and TCP," in *ACM NOSSDAV*, Jun. 2012.

[16] P. Georgopoulos, Y. Elkhatib, M. Broadbent, M. Mu, and N. Race, "Towards Network-wide QoE Fairness Using Openflow-assisted Adaptive Video Streaming," in *ACM SIGCOMM Workshop*, 2013.

[17] D. P. Palomar and M. Chiang, "A Tutorial on Decomposition Methods for Network Utility Maximization," *IEEE JSAC*, 2006.

[18] S. Laga, T. V. Cleemput, F. V. Raemdonck, F. Vanhoutte, N. Bouten, M. Claeys, and F. D. Turck, "Optimizing scalable video delivery through OpenFlow layer-based routing," in *IEEE NOMS*, May 2014.

[19] P. Medagliani, S. Paris, J. Leguay, L. Maggi, X. Chuangsong, and H. Zhou, "Overlay Routing for Fast Video Transfers in CDN," in *IEEE Int. Symp. on Integrated Netw. Management*, 2017.

[20] W.-E. Liang and C.-A. Shen, "A high performance media server and QoS routing for SVC streaming based on Software-Defined Networking," in *IEEE ICNC*, Jan 2017.

[21] A. Mansy, B. Ver Steeg, and M. Ammar, "SABRE: A client based technique for mitigating the buffer bloat effect of adaptive video flows," in *ACM MMSyS*, Feb. 2013.

[22] X. Liu, A. Men, and P. Zhang, "Enhancing TCP to Improve Throughput of HTTP Adaptive Streaming," *Int. J. of Future Generation Comm. and Netw.*, vol. 7, no. 1, 2014.

[23] S. Petrangeli, T. Wauters, R. Huysegems, T. Bostoen, and F. De Turck, "Software-defined network-based prioritization to avoid video freezes in HTTP adaptive streaming," *Int. J. of Network Management*, vol. 26, no. 4, pp. 248–268, 2016.

[24] A. Bentaleb, A. C. Begen, and R. Zimmermann, "SDNDASH: Improving QoE of HTTP Adaptive Streaming Using Software Defined Networking," in *ACM Conf. on Multimedia*, 2016, pp. 1296–1305.

[25] J. W. Kleinrouweler, B. Meixner, and P. Cesar, "Improving Video Quality in Crowded Networks Using a DANE," in *ACM NOSSADAV*, 2017.

[26] S. DAronco, L. Toni, and P. Frossard, "Price-Based Controller for Quality-Fair HTTP Adaptive Streaming," in *IEEE ISM*, 2016.

[27] J. Chen, M. Ammar, M. Fayed, and R. Fonseca, "Client-Driven Network-level QoE Fairness for Encrypted 'DASH-S'," in *ACM SIGCOMM workshops*, 2016, pp. 55–60.

[28] X. Yin, M. Bartulovi, V. Sekar, and B. Sinopoli, "On the efficiency and fairness of multiplayer HTTP-based adaptive video streaming," in *American Control Conference (ACC)*, May 2017, pp. 4236–4241.

[29] G. Cofano, L. D. Cicco, T. Zinner, A. Nguyen-Ngoc, P. Tran-Gia, and S. Mascolo, "Design and Performance Evaluation of Network-assisted Control Strategies for HTTP Adaptive Streaming," *ACM Trans. Multimedia Comput. Comm.*, vol. 13, no. 3s, Jun. 2017.

[30] G. Dimopoulos, I. Leontiadis, P. Barlet-Ros, and K. Papagiannaki, "Measuring Video QoE from Encrypted Traffic," in *ACM IMC*, 2016.

[31] A. Balachandran, V. Sekar, A. Akella, S. Seshan, I. Stoica, and H. Zhang, "Developing a predictive model of quality of experience for internet video," *SIGCOMM CCR*, vol. 43, no. 4, pp. 339–350, Aug. 2013.

[32] VQEG, "Video Quality Experts Group." [Online]. Available: https://www.its.bldrdoc.gov/vqeg/vqeg-home.aspx

[33] T. Spetebroot, S. Afra, N. Aguilera, D. Saucez, and C. Barakat, "From network-level measurements to expected quality of experience: The Skype use case," in *IEEE M&N*, Oct 2015.

[34] C. Kreuzberger, D. Posch, and H. Hellwagner, "AMuSt Framework - Adaptive Multimedia Streaming Simulation Framework for ns-3 and ndnSIM," 2016.

[35] R. Aparicio-Pardo, G. Calvigioni, L. Sassatelli, J. Leguay, P. Medagliani, and S. Paris, "Appendices on QoE model and DASH manifest in 'Quality of Experience-based Routing of Video Traffic for Overlay and ISP Networks'," Universite Cote d'Azur, Other, 2018. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01672042

[36] S. R. Kang and D. Loguinov, "Modeling best-effort and fec streaming of scalable video in lossy network channels," *IEEE/ACM Transactions on Networking*, vol. 15, no. 1, pp. 187–200, Feb 2007.

[37] M. Jain and C. Dovrolis, "Path selection using available bandwidth estimation in overlay-based video streaming," *Comput. Netw.*, vol. 52, no. 12, pp. 2411–2418, Aug. 2008.

[38] M. Yajnik, S. Moon, J. Kurose, and D. Towsley, "Measurement and modeling of the temporal dependence in packet loss," in *IEEE INFOCOM*, 1999.

[39] M. Ellis, D. P. Pezaros, T. Kypraios, and C. Perkins, "A two-level Markov model for packet loss in UDP/IP-based real-time video applications targeting residential users," *Computer Networks*, vol. 70, pp. 384 – 399, 2014.

[40] "A TCP CUBIC implementation in ns-3, author=Levasseur, Brett and Claypool, Mark and Kinicki, Robert, booktitle=Workshop on ns-3, year=2014, organization=ACM."

[41] M. Pióro and D. Medhi, *Routing, flow, and capacity design in communication and computer networks*. Elsevier, 2004.

[42] "Simulation platform," 2018. [Online]. Available: https://github.com/sassatelli/QoErouting

[43] S. Uhlig, B. Quoitin, J. Lepropre, and S. Balon, "Providing public intradomain traffic matrices to the research community," *ACM SIGCOMM CCR*, vol. 36, no. 1, pp. 83–86, 2006.