On Accounting for Screen Resolution in Adaptive Video Streaming: A QoE-Driven Bandwidth Sharing Framework

Othmane Belmoukadam
Université Côte d'Azur, Inria, France
othmane.belmoukadam@inria.fr

Muhammad Jawad Khokhar
Université Côte d'Azur, Inria, France
muhammad-jawad.khokhar@inria.fr

Chadi Barakat
Université Côte d'Azur, Inria, France
chadi.barakat@inria.fr

***Abstract*—Screen resolution along with network conditions are main objective factors impacting the user experience, in particular for video streaming applications. Terminals on their side feature more and more advanced characteristics resulting in different network requirements for good visual experience [1]. Previous studies tried to link MOS (Mean Opinion Score) to video bit rate for different screen types (e.g., CIF, QCIF, and HD) [2]. We leverage such studies and formulate a QoE-driven resource allocation problem to pinpoint the optimal bandwidth allocation that maximizes the QoE (Quality of Experience) over all users of a provider located behind the same bottleneck link, while accounting for the characteristics of the screens they use for video playout. For our optimization problem, QoE functions are built using curve fitting on data sets capturing the relationship between MOS, screen characteristics, and bandwidth requirements. We propose a simple heuristic based on Lagrangian relaxation and KKT (Karush Kuhn Tucker) conditions for a subset of constraints. Numerical simulations show that the proposed heuristic is able to increase overall QoE up to 20% compared to an allocation with TCP look-alike strategies implementing max-min fairness. Later, we use a MPEG/DASH implementation in the context of ns-3 and show that coupling our approach with a rate adaptation algorithm (e.g., [3]) can help increasing QoE while reducing both resolution switches and number of interruptions.**

I. Introduction

Today, end devices such as mobile phones, tablets, and monitors, are doted with more and more advanced features. In particular, they support different but still limited video formats and resolutions. Apple with more than 20% share of the mobile market presents plenty of products with several characteristics [4]. For example, an iPhone 4S supports up to 640x480 pixels, while an iPhone 8 supports up to 1334x750 pixels. On the other side, mobile operators and network providers give high importance to Quality of Experience (QoE) as a metric to assess users satisfaction and avoid any economic loss. Even though 5G networks promise high connectivity and huge transmission capacity aiming to take users experience to the next level [5], [6], bandwidth sharing is still an important issue for network operators and content providers, especially in view of the exponential rise of video traffic volume. It turns out that the objective aspect of Quality of Experience is tightly correlated to terminal playout characteristics (e.g, size, resolution) but also to network conditions [1], [7], [8].

In light of the previous facts, studies have linked QoE metric MOS (Mean Opinion Score) to bit rate for different screen resolutions in the context of video streaming. By assembling video quality data for screens of different resolutions, Cermak et al. [2] managed to show that small screens (e.g, CIF, QCIF) scale toward high MOS levels at small bit rates. Moreover, they provide a curve for H.246 bit rate required for a desired MOS level for different screen resolutions (Fig. 5 in [2]). On the other side, content providers, such as YouTube, propose content in different resolutions without a careful customization to the screen or the viewport. It follows that an end device can download video at more than its display capacity before transcoding it down to match its display, which consists in a waste of network resources without bringing additional gain to the user in terms of her QoE. In practice, the choice between the different resolutions is managed either statically by the user, or automatically by the player thanks to an HTTP Adaptive Streaming protocol (HAS) in a way to cope with the network conditions [9]. A well known HAS protocol used by major content providers is MPEG DASH. In this context, changing the viewport does not necessarily mean a resolution change as the DASH still downloads the best available video resolution that fits within the available network bandwidth. Less is done to explore how one can leverage the terminal characteristics (e.g., screen resolution) in the allocation of network resources (e.g., bandwidth). The allocation is left to DASH and TCP, which in case of multiple flows sharing a bottleneck link, converge to a fair split of the available bandwidth. For different screen resolutions, or different viewports in general, this latter allocation does not lead to a fair QoE allocation, nor to an optimal overall QoE.

In this paper, we study the problem of resource allocation for several video flows sharing the same bottleneck, and develop an optimization framework to maximize the overall QoE taking into consideration terminal display capabilities. We use two data sets to fit QoE functions that link throughput [10] or video bit rate [2] to a QoE level. We formulate a non-linear optimization problem and propose heuristics to solve it, then we conduct numerical and network simulations under different system settings to validate the interest of a QoE-driven bandwidth allocation approach. Our results show that we can indeed ensure a higher overall QoE if screen resolu-

This work is supported by Inria Project Lab BetterNet and the French National Project ANR BottleNet.

978-3-903176-24-9 © 2019 IFIP

tions are to be taken into consideration. Our contributions can be summarized as follows:

- We formulate a resource allocation optimization problem based on QoE, where QoE functions are built using data sets linking (throughput or bit rate) to MOS.
- We present a relaxation to a non-linear problem by considering continuous video bit rates. Under this relaxation, we develop a simple and greedy heuristic based on Lagrangian multipliers and KKT (Karush Kuhn Tucker) conditions, and prove that our heuristic converges to a state where all gradients are either equal or constraints on the bit rate are reached.
- We use the network simulator ns-3 [11] and an open source implementation of DASH to validate our approach and to propose an implementation of the optimal solution by limiting the subset of visible video representations of a player based on the resolution of its viewport.

The rest of this paper is organized as follows. Sec. II reviews some of the related works on QoE modeling and resource sharing. In Sec. III and IV we present our framework and formulate our optimization problem. Sec. V shows numerical results and evaluates the gain of the proposed solution compared to other allocation strategies. In Sec. VI we illustrate our experimental results using the network simulator ns-3 [3], [12], and evaluate the gain in terms of overall QoE for both the optimal solution and a practical implementation that limits the video bit rate as a function of the client's viewport. Finally, we conclude and provide traces for our future work.

II. Related works

The topic QoE-driven resource sharing has been already investigated in several occasions. For example, in terms of routing, neural networks have been used in both wired and wireless networks to optimize QoE. Previous work tried to select the best path using network-level QoS features as QoE replacement (e.g., loss rate, delay) [13], [14]. Quang et al. illustrate QoE-driven routing as a MILP problem by considering Pseudo-Subjective Quality Assessment (PSQA) as QoE model and propose a heuristic solution [15]. Moreover, Calvigioni et al. take into consideration the HAS flow requirements and study them in conjunction with TCP. They use a linear QoE function in order to express a joint routing and resource allocation problem and propose a dual sub-gradient approach based on Lagrangian relaxation sub-problems to select a single best path upon each request [16].

In [17], the authors express a rate allocation problem to maximize a Two-term Power Series Model over three requested resolutions and under link capacity constraint. The optimal solution is implemented in switches by means of a weighted fair queuing and by using OpenFlow. However, the Utility function used depends on the characteristics of a test video that is too specific and less generic since it requires a mapping per video at each resolution. Moreover, the use of the Structural Similarity Index (SSIM) as a metric to asses quality requires very specific state sharing with the controller, which can be tricky giving the prevalence of encryption-based delivery.

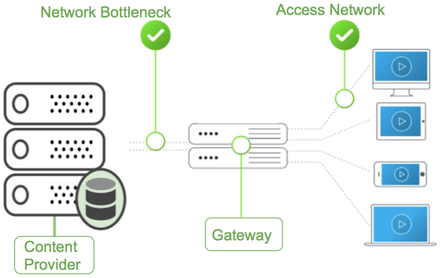


Fig. 1: Framework overview

However, to the best of our knowledge, studying the resource sharing problem from an end device perspective and with trace-driven models for video QoE is very promising. Therefore, we focus on video streaming and leverage QoE functions able to capture the link between video bit rates or throughput and QoE for different screen resolutions. We express a resource sharing problem to maximize the sum of the non-linear QoE functions under linear constraints on the screen resolution and the bottleneck link utilization. We also validate our model and compare the different solutions with a realistic DASH implementation in ns-3.

III. Framework and system model

A. Framework

Consider a set of users with different screen resolutions streaming videos from a server as illustrated in Fig. 1. Videos on the server are encoded into M different representations (i.e., bit rates or resolutions). We assume users are not limited by their access links and are thus able to download any video representation. We assume the bottleneck of the system to be the backhaul link located between the gateway and the video server. In this work, we focus on the problem of QoE-driven bandwidth sharing on the backhaul link, and do not consider the presence of any caching functionality at the gateway. Caching would add another interesting dimension to our problem, and would certainly interact with bandwidth sharing on the backhaul, so we differ its study to a future work dedicated to the topic. For now, one can see our work as specific to those videos that are not cached.

B. System model

We now describe in more detail the model that we consider in this work and introduce our notation. Let F denote the set of video files in our catalog (server) offered to the users. Any video file $f \in F$ is encoded into a set of M representations with f_m being the m-th representation of video f, having an encoding bit rate equal to B_{fm}. For the sake of simplicity, and without loss of generality, we suppose all videos to have the same duration T. Further, we suppose that $\forall f \in F$ and $\forall m \in M$, the B_{fm}'s are the same (i.e., which can be seen as

Notation	Representation
F	Set of videos
M	Set of video representations
S	Set of screen resolutions
R	Set of video resolutions
λ_f	Request rate per Video f
$\lambda_{f,s}$	Request rate per video f and screen resolution s
α	Parameter of the popularity Zipf distribution
C_l	Backhaul link capacity
$B_{M,s}$	Upper bound on bit rate for $s \in S$
X	Bandwidth allocation vector

TABLE I: Notations of our framework

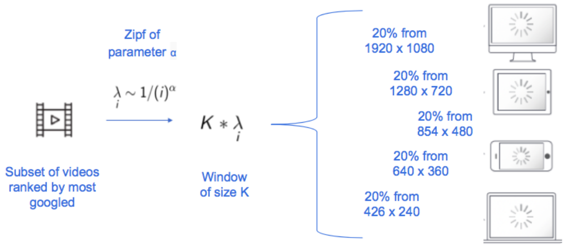


Fig. 2: Traffic generation according to our model

the average over all videos of the catalogue for representation m). Let S be the vector of distinct screen resolutions.

For every $f \in F$ we assign a request rate λ_f (i.e., popularity) according to a Zipf distribution of parameter α. This request rate is total over all screen resolutions. Each request to a video f is assigned a screen resolution according to a given probability distribution over S. In practice, a network operator can obtain the information using the IMEI (International mobile equipment identity) of the end-user devices or by collaborating with the video content providers such as YouTube. Multiplied by λ_f, this gives the request rate per video f and per screen resolution s that we denote $\lambda_{f,s}$. We have $\lambda_f = \sum_{s \in S} \lambda_{f,s}$. Table I summarizes the notation used in our framework, while Fig. 2 illustrates the process of generating requests for the case $|S| = 5$ screens (most common screen resolutions in mobile market).

C. From QoS to QoE

As per prior subjective studies, the QoE of video streaming is a function of application layer QoS features that are either dependent on the video content (e.g., video bit rate) or the playout metrics (e.g., the intial startup delay) [8], [18]; the playout metrics further depend on the underlying network conditions such as the network throughput or delay.

In this work, we consider building QoE functions that take as input the network throughput or the video bit rate. The throughput is a function of the available bandwidth, but also of other network performance issues such as the delay and the packet loss rate. To build these QoE functions, we rely on two publicly available datasets that map the QoS to the QoE. The first dataset is built by controlled experiments and links the network throughput to the QoE modeled according to ITU P.1203 recommendation [10], while the second dataset is based on the work of the Video Quality Experts Group (VQEG) [2] and maps the video bit rate to the MOS. On these datasets, we apply curve fitting methods (e.g., non-linear least squares) with the canonical function given in Equation (1) to build our target QoE function taking each time as input the network throughput and the video bit rate respectively. In this Equation (1), x stands for the network throughput or the video bit rate, while index s stands for the screen resolution. Constant β determines the shape of the QoE function.

$$QoE_s \underset{exp}{=} QoE_{max}(1 - e^{-\beta_s x}). \tag{1}$$

1) Network throughput to QoE: The dataset for this model is built by controlled experiments in the lab [10]. It consists of 100k unique YouTube video playouts under different trace-driven emulated network conditions. This dataset maps the network QoS features such as throughput, delay, and packet loss to application level measurements such as join time, stalls and video resolutions. The application QoS data allows to calculate the ITU-T P.1203 subjective MOS [1] for different screen resolutions. We use curve fitting based on Equation (1) to plot the computed MOS according to ITU standard with respect to the sole network throughput for the different screen resolutions as shown in Fig. 3. The screen resolutions given in the figure correspond to the available video resolutions in the traces of the experiments in [10].

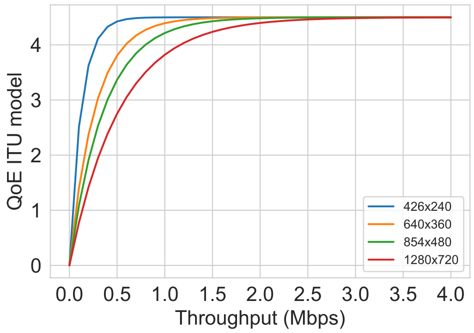


Fig. 3: Fitting QoE function (1) using controlled experiments data from [10]

2) Video bitrate to QoE: Cermak et al. [2] studied the relationship between the video quality, the screen resolution, and the video bit rate using the VQEG datasets. The authors show that for screen resolutions such as CIF, QCIF and HD, different video bit rates are needed to achieve a certain MOS level (see Fig. 5 in [2]). For the main mobile screen resolutions, typically from 426x240 up to 1920x1080, we extrapolate a vector Z where each entry has two values (z_{BR},z_{MOS}) then we use mean square error method to fit curves using Eq. (1).

Note that according to [2], the video bit rates take discrete values, thus making the resource allocation problem an INLP

[1] The ITU-T P.1203 model is a standardised model that takes as input the application QoS to estimate the subjective MOS.

(Integer Non-Linear). However, for this work, we relax the problem to make the bit rate take any real value between its minimum and maximum values given by [2], which has the advantage to transform the resource allocation problem into an NLP (Non-Linear), hence easing the solution. We believe in a real scenario, video content varies considerably making the bit rate take more diverse values than the ones in [2].

IV. QoE-driven bandwidth sharing

A. Problem description

Our problem can be described as follows. Given the different representations of videos, the distribution of screen resolutions and the backhaul link capacity (denoted C_l), we seek how to share the bandwidth of the backhaul link between the multiple video sessions so that the total system utility (defined as the sum of QoE functions as in Section III-C1) is maximized. We want this maximization to account for the screen resolutions and for the constraint on the capacity of the backhaul link. Note here that the best we can hope from TCP and DASH is a fair split of the available C_l over all flows. However, the optimal allocation is nontrivial as fairness at the resource level does not necessarily imply fairness at the QoE level. The fact that small screens require lower bit rate than large screens for same level of QoE is a good illustration.

B. Problem formulation

Let's introduce the vector $X = (x_s), s \in S$, where the s-th element denotes the bandwidth allocated to each of the users with screen resolution s. The QoE-driven bandwidth allocation for video quality improvement can be formulated as an nonlinear program (NLP) as follows:

$$\max_{X} \quad U(X) = \sum_{s\in S} \lambda_s QoE_s(x_s) \tag{2a}$$

$$\text{s.t.}$$

$$\sum_{s\in S} \lambda_s x_s \leq C_l \tag{2b}$$

$$x_s \leq B_{M,s}, \qquad s \in S \tag{2c}$$

$$x_s \geq 0, \qquad s \in S \tag{2d}$$

The global utility function of the system is defined as the sum of weighted QoE functions built in Section III-C1. To do so, we aggregate users with same screen resolution as they are supposed to obtain the same allocation ($\lambda_s = \sum_{f\in F} \lambda_{f,s}$). Constraint (2b) accounts for the backhaul capacity limitation, whereas constraint (2c) upper bounds the allocation for every screen resolution based on the bit rate or throughput needed for excellent video quality according to the data sets we are using. Note here that this upper bound can be removed as it is accounted for indirectly by the QoE functions, but we decided to keep it for clarity of the presentation.

Algorithm 1 Compute allocation vector

```
Output: $Optimal\ value\ of\ X = (x_s), s \in S$
Input: $\lambda_s,\ QoE_s,\ B_{M,s},\ C_l$
    $Initialize\ X\ from\ (W) \begin{cases} \frac{\partial QoE_i}{\partial x_i} = \frac{\partial QoE_j}{\partial x_j}\ \forall (i,j) \in S \\ (\sum_{s\in S} (\lambda_s x_s) - C_l)) = 0 \end{cases}$
while either (2c) or (2d) false do
    if $\exists\ x_i \in X, such\ that\ x_i < 0$ then
        $Set\ Min(x_i \in X|\ x_i < 0) = 0\ for\ (X, W)$;
        $Solve\ (W)$;
        Continue;
    end
    if $\exists\ x_i \in X, such\ that\ x_i > B_{M,i}$ then
        $Set\ Max(x_i \in X|\ x_i > B_{M,i}) = B_{M,i}\ for\ (X, W)$;
        $Solve\ (W)$;
        Continue;
    end
end
```

C. Gradient solution based on Lagrangian relaxation

For the QoE function we consider in this paper (Eq. (1)), our problem is convex and thus possesses a unique solution. One can use well known heuristics such Sequential Least Squares [19] to get an approximation of the optimal. However, given the particular shape of our QoE functions (i.e., monovariate) and constraints, and to get insights on the optimal solution, we propose a simple greedy heuristic that helps approximating the non-linear objective function and efficiently maximizing it. The proposed greedy heuristic considers KKT (Karush Kuhn Tucker) conditions to check if a feasible solution is optimal. We start by writing the partial Lagrangian function obtained by relaxing constraint (2b) :

$$L(X, \gamma) = U(X) - \gamma(\sum_{s\in S}(\lambda_s x_s) - C_l). \tag{3}$$

Constant γ is the Lagrangian multiplier associated to constraint (2b). By supposing constraint (2b) to be set to equality at the optimal solution (otherwise the system is under-utilized), and by differentiating the Lagrangian with respect to allocation vector X, we can prove that a first possible solution could be the one that equalizes all gradients of the QoE functions.

$$\frac{\partial QoE_i(x_i)}{\partial x_i} = \frac{\partial QoE_j(x_j)}{\partial x_j}, \forall (i, j) \in S. \tag{4}$$

This, together with $(\sum_{s\in S} (\lambda_s x_s) - C_l) = 0$, gives a system of equations (W) that we can solve to find our first bid on the optimal allocation vector. This first bid is the optimal allocation if the two other constraints (2c) and (2d) are not violated, otherwise our vector is not the optimal vector and has to be updated. We use the information on the violated constraints to reshape the search space, i.e., we take those violated constraints one by one and at each step, we set the corresponding allocation either to zero or to the upper bound, then we replace them in the Lagrangian (3) and repeat the previous process until converging to an allocation that satisfies all constraints while nullifying the gradient of the Lagrangian. The above algorithm provides further details on our approach.

V. Numerical simulations

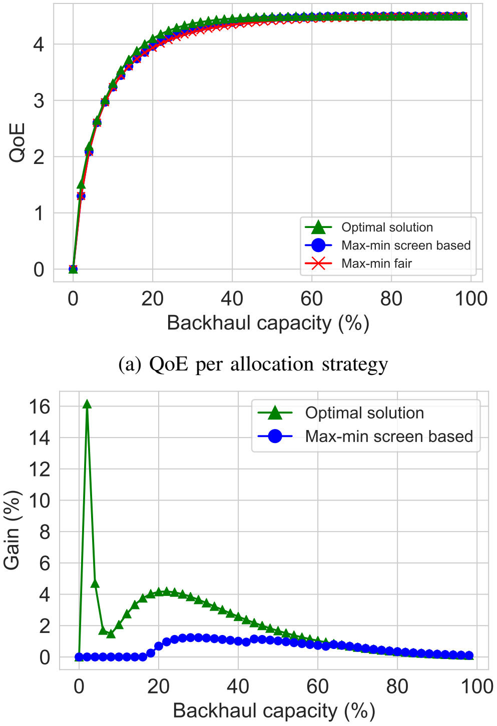
A. Simulation setup

We consider a network where a set of users have different screen resolutions distributed uniformly over S. We consider S to include the five major mobile screen resolutions depicted in Fig. 2. Videos are of equal duration, the allocation vector X stands in this case for the number of bytes each video would require from the network. As reference allocations, we consider two max-min allocations, which model the existing solutions based on TCP and DASH. The first allocation is called *max-min fair* which consists in video flows sharing equally the available bandwidth independently of the characteristics of their screens (i.e., a flow can get more than it can play out). The second allocation is called *max-min screen based* where bandwidth is fairly shared but in the limit of maximum supported bit rate per screen (denoted $B_{M,s}$ in our notation). This consists of a video flow of screen size s fighting for the bandwidth and sharing it fairly with the others as long as the maximum bit rate $B_{M,s}$ is not reached. Once reached, the flow (i.e., DASH) does not ask for higher bit rates even if bandwidth is available. This control can be either implemented at the client, or at the server if information on the screen (or viewport) is made available to it.

In addition to these reference allocations, we use our heuristic to derive the best bandwidth share maximizing the sum of QoE functions over all flows. We show results for the exponential QoE function in Equation (1), but we also discuss an extreme case where QoE grows linearly with the throughput or the bit rate. Note that the optimization with a linear QoE function cannot be solved with our heuristic as it corresponds to a linear program. We solve it instead with CPLEX [20] and provide an intuitive interpretation of the results. Note here that we focus on a snapshot problem where we perform the optimization only once before assigning the resources. As for adapting to dynamic changes, we need the distribution of users and screens which can be computed and tracked in an online fashion.

B. Bandwidth allocation and QoE

We compare the previous allocation strategies in terms of the overall QoE while varying the backhaul capacity. We start by considering the dataset available in [2] (see Section III-C2) to calibrate our QoE function. We express the backhaul capacity as a percentage of the worst case scenario where the operator over-dimensions its network to deliver the maximum bit rate to all users independently of their screen sizes. Fig. 4 shows both the overall QoE with respect to C_l and the relative gain of the two strategies *optimal* and *max-min screen based* with respect to the baseline strategy *max-min fair*. We notice how leveraging the QoE function with very limited backhaul capacity can achieve a QoE gain of 16% over the baseline strategy. Furthermore, the optimal and the *max-min screen based* strategies manage to reach the maximum possible QoE earlier than the *max-min fair* one while increasing the backhaul capacity. For small backhaul capacity, *max-min fair* and *max-min screen based* lead to same result (i.e., zero gain) as the maximum bit rate per screen is not reached, which is not the case of the *optimal* which still delivers a better result. We recall that this result is obtained for screen sizes of equal popularity.



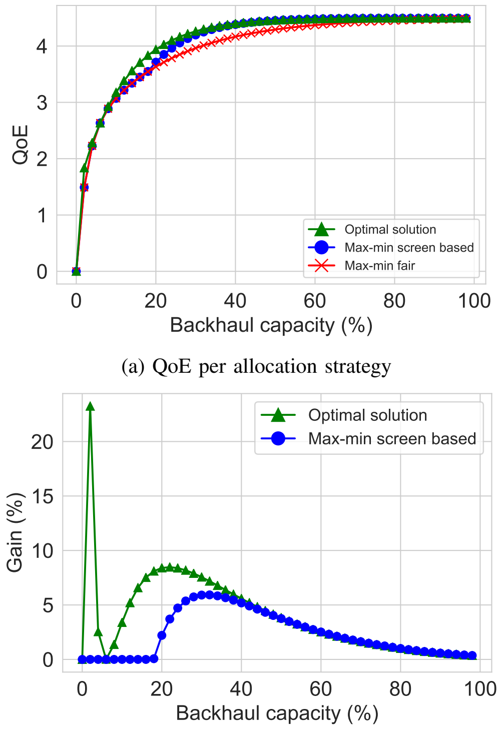
(a) QoE per allocation strategy

(b) Relative gain, uniform screen probabilities

Fig. 4: Comparison of allocation strategies with uniform screen probabilities, QoE from III-C2

To check for the impact of screen popularity, we apply other distributions of screen sizes over S and find that our approach works well for other scenarios. Moreover, we point to cases where the gain can actually be up to 20% compared to a simple fair split of the available capacity. Indeed, we compare in Fig. 5 the different allocation strategies for a scenario of only two screen resolutions, 426x240 (small) and 1920x1080 (large). In Fig. 5(b), we notice how the gain improves and can go above 20%. In this case, we end up with two subsets of users, greedy users (big screens) and users easy to satisfy (small screens). In such scenario, the baseline strategy divides the available bandwidth fairly among all users, which is insufficient especially when C_l is small, as large screens cannot get to an acceptable QoE level with the given allocation while small screens get more than needed.

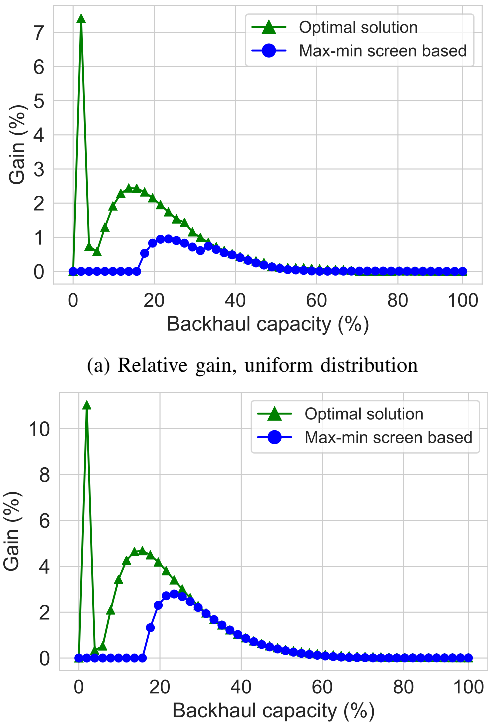
We redo the same numerical simulation using QoE functions fitted on the controlled experiments data in [10] (see Section III-C1). Fig. 6 includes comparison of the different strategies for two distributions of screen sizes, uniform and small/large only. We can notice how the gain for these QoE functions drawn according to ITU-T P.1203 standard spans



(a) QoE per allocation strategy

(b) Relative gain, 426x240 and 1920x1080

Fig. 5: Comparison of allocation strategies with two screen sizes (426x240 and 1920x1080), Qoe from III-C2



(a) Relative gain, uniform distribution

(b) Relative gain, 426x240 and 1920x1080

Fig. 6: Comparison of allocation strategies with different screen size distributions, QoE from III-C1

smaller ranges while maintaining the same shape as with QoE functions from III-C2. In particular, for Fig. 6(a), our heuristic results in a gain in overall QoE up to 7% compared to the baseline TCP fair split (*max-min fair*). On the other hand, *max-min screen based* is only 1% better compared to the baseline. Moreover, when 50% of requests come from 426x240 devices and the rest from 1920x1080 devices, the optimal allocation (Fig. 6(b)) results in a gain of 11% compared to baseline allocation. Again we notice that for small C_l, the *max-min screen based* strategy and the *max-min fair* result in the same allocation leading to the same QoE.

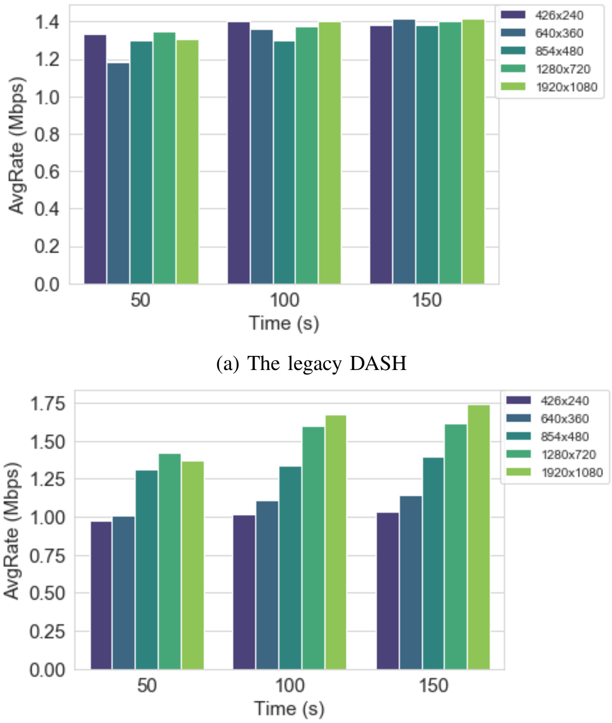
C. Linear QoE

Instead of exponential QoE function, one can imagine an extreme case where QoE grows linearly with the network throughput or the video bit rate, within the range $[0, B_{M,s}]$ for screen resolution s. Even though not realistic, this type of QoE functions is interesting because of its implication on the optimal allocation and the way it is to be implemented. Indeed, the sum of linear QoE functions transforms the NLP to an LP that can no longer be solved with our heuristic. Instead, one can use CPLEX to solve it [20]. In such case, we found that more important gains can be reached. More interestingly, and because slopes of QoE functions are constant but no longer the same for all screens (fast slopes for small screens, slow slops for big screens), the optimal allocation would simply consist in giving full bandwidth priority to small screens on large screens. So small screens are served first in the limit of their $B_{M,S}$, then larger screens, and so on, until all screens are served. Such allocation, as it requires full priority, cannot be simply implemented on an end-to-end basis, but requires the intervention of the network operator as well.

VI. Network simulation

For the validation of our numerical results, we use the simulation software ns-3 [11]. We work with an implementation of MPEG/DASH proposed by [12], [21] that supports the Smooth Video Adaptation Algorithm (SVAA) designed in [3]. DASH being a standard issued by MPEG in 2012 for HAS, different rate adaptation algorithms are proposed in the literature to figure out the resolution of the next segment to download so as to minimize the number of switches and stalls. The SVAA algorithm we consider has shown its efficiency in preventing resolution switches and interruptions [12], [21].

Our simulation setup consists of multiple terminals (fifteen in total) acting as DASH clients and downloading videos from a DASH server. Clients are connected to a router via access links of $5Mbps$ and $2ms$ delay simulating ADSL access links. Their traffic is routed toward the central server by the mean of a wired link of fixed capacity $30Mbps$ and of fixed delay $6ms$. In terms of video content, we use an animated YouTube video called the Elephants Dream. The video is divided in 2s segments, each of which is available in several representations produced using traces from [22].



(a) The legacy DASH

(b) The QoE based DASH

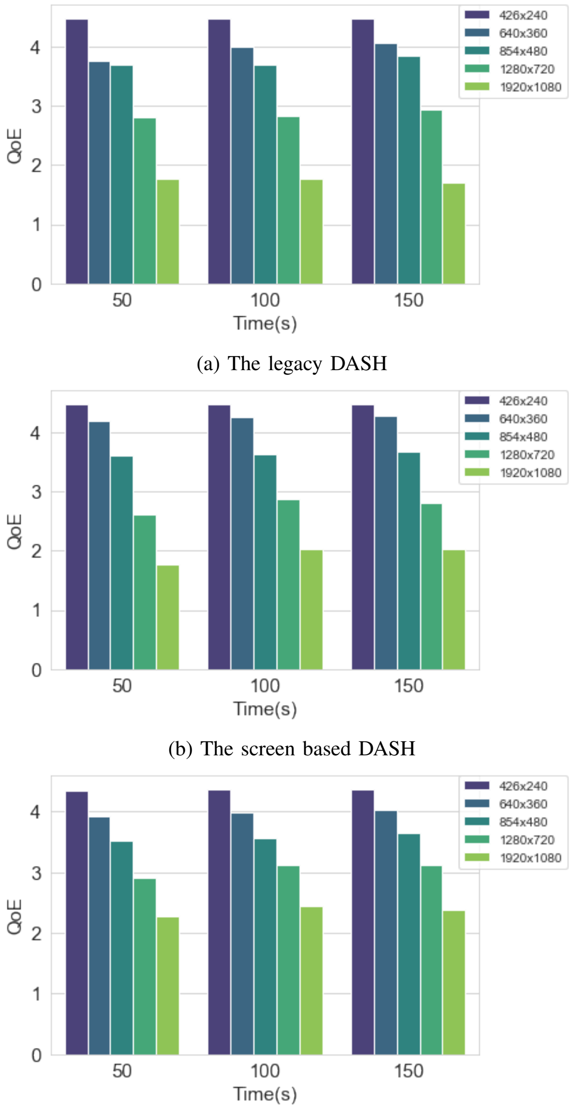
Fig. 7: Average download bit rate per screen resolution over the shared link $C_l = 30Mbps$

In addition to the standard implementation of DASH/SVAA, we propose two other implementations illustrating the different aspects of our approach. To simulate the optimal solution (see Section IV-C), we find the optimal allocation of a video flow using our heuristic then limit the view of the client to video representations not exceeding this allocation. Moreover, we implement screen based max-min by changing the DASH/SVAA client so that the maximum downloadable representation is the ceil of the maximum bit rate for which the given screen resolution attains the maximum QoE (with respect to dataset presented in Section III-C2).

A. Simulating QoE-driven DASH

We assign to our devices screen resolutions from a subset of five major mobile screen resolutions (e.g., from 426x240 to 1920x1080) using uniform probability distribution. We simulate the users' behavior with each described implementation, then we plot the average attainable download bit rate over the shared link per screen resolution and calculate the total corresponding QoE using the functions fitted in Section III-C2. We repeat every simulation several times and average outputs to reduce the bias effect and smooth results.

Fig. 7(b) illustrates the effect of QoE based DASH, where we can see the attainable download bit rate proportional to the screen resolution as needed for good video quality. This is contrary to Fig. 7(a) where all users grab approximately the same share of the available capacity as expected.



(a) The legacy DASH

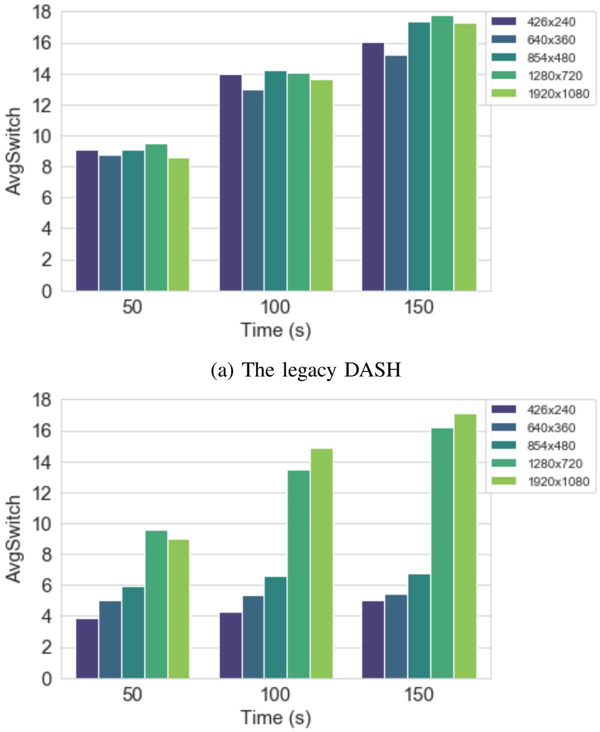
(b) The screen based DASH

(c) The QoE based DASH

Fig. 8: QoE per screen resolution as function of average download bit rate, multiple DASH implementations

In Fig. 8 we compute the average QoE per screen resolution for the three implementations: (a) legacy DASH, (b) screen based DASH, and (c) QoE based DASH. We notice through the three figures how the QoE is rearranged in between the different screen resolutions. Thanks to screen size consideration, we manage to enhance the average QoE for greedy users (big screen resolution) while maintaining a good QoE level for the others, all this without exceeding the backhaul budget.

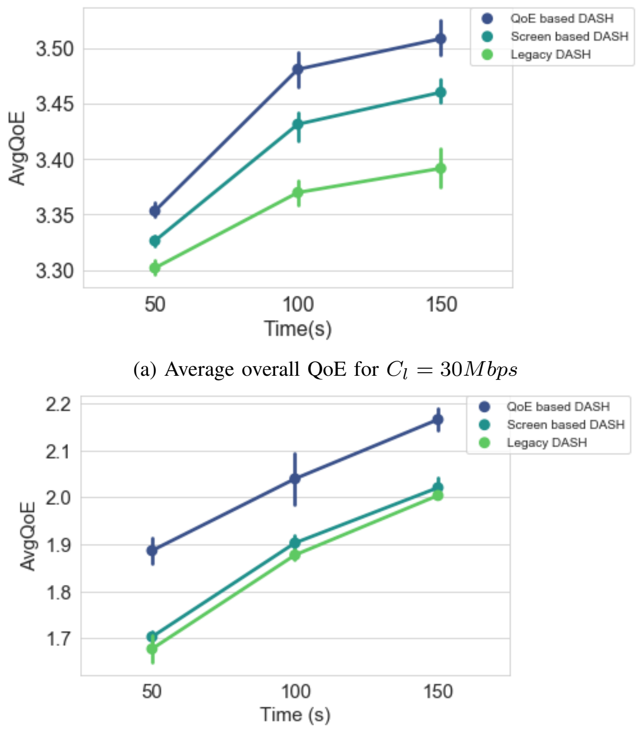
We also compare the different implementations in terms of main application-level QoS factors (e.g., stalls, resolution switches and interruptions) that could impact the subjective QoE [8], [18]. We focus on the quality switches as they occur only in adaptive video streaming sessions making their evaluation important. We can see in Fig. 9(b) how QoE based DASH manages to reduce the number of switches per screen



(a) The legacy DASH

(b) The QoE based DASH

Fig. 9: Average resolution switches per screen resolution



(a) Average overall QoE for $C_l = 30Mbps$

(b) Average overall QoE for $C_l = 10Mbps$

Fig. 10: Average overall QoE vs. backhaul capacity

resolution during a watching session compared to legacy DASH (Fig. 9(a)). We made the same observation regarding the other application-level QoS metrics of number and duration of interruptions.

B. Changing the backhaul capacity

From the above results, screen based max-min allocation seems to be an efficient allocation easy to implement and providing close gain to the optimal allocation. However, we expect such allocation to deviate from the optimal when the stress on the backhaul link increases (either more traffic or less bandwidth). Indeed, for more congested scenarios, the fair share of bandwidth of a flow of some resolution s becomes likely less than the maximum bit rate for that resolution $B_{M,s}$, which makes the limit on the bit rate driven by the screen resolution less effective. We expect therefore *max-min screen based* to be closer to *max-min fair* and farther from *optimal*. To illustrate this observation, we redo the above ns-3 simulations this time using a shared capacity of $C_l = 10Mbps$ while maintaining the delay to $6ms$. Fig. 10 shows the total QoE for the two cases: (a) $C_l = 30Mbps$ and (b) $C_l = 10Mbps$, and this is for the different allocations. In addition, we show 90% confidence intervals for the observed results. In both cases, QoE based DASH outperforms the other implementations achieving higher overall QoE. However, for limited backhaul capacity (Fig 10(b)) and as expected, screen based DASH gives approximately same results as legacy DASH. This results confirms our intuition that allocating based on QoE always helps shaping boundaries of bandwidth space and improving overall QoE regardless of the available capacity, which is not the case of the other two allocations.

VII. Conclusions

In this paper, we study the problem of bandwidth allocation for multiple video streaming sessions over a shared link. The goal is to maximize the average QoE by leveraging screen resolution. In a first place, we revisit previous studies, able to link either video bit rate [2] or throughput [10] to QoE level for a given screen resolution. We then formulate an optimization problem trying to maximize the overall QoE under linear constraints. To that aim, we propose a Lagrangian based solution to approximate the optimal allocation. Later, we show through numerical and network simulations how leveraging screen characteristics leads to overall QoE improvement in the context of a QoE-driven bandwidth allocation framework. In addition, accounting for screen resolution reduces both switches and interruptions over a watching session.

As future work, we plan on studying other aspects of content management in the Internet such as caching. The caching part can give another dimension to our framework. In addition, the work will continue towards building a collaborative framework, a new labeling system, where content provider can tag packets by quality levels using the more advanced QoE models. The given feedback can be of high interest to network provider as it can help them well dimension the network and maximize the user experience.

REFERENCES

[1] A. Elmnsi, N. Osman, and I. Mkwawa, *The Impact of Mobile Device Preference on the Quality of Experience*. IInternational Journal of Computing Information Sciences, 2016.

[2] G. Cermak, M. Pinson, and S. Wolf, *The Relationship Among Video Quality, Screen Resolution, and Bit Rate*. IEEE Transactions on Broadcasting, 2011.

[3] G. Tian and Y. Liu, *Towards agile and smooth video adaptation in dynamic HTTP streaming*. Proc. 8th Int. CoNEXT, 2012.

[4] *http://gs.statcounter.com/vendor-market-share/mobile*.

[5] . A. M. Polese, M. Giordani, M. Mezzavilla, S. Rangan, and M. Zorzi, *Improved Handover Through Dual Connectivity in 5G mmWave Mobile Networks*. IEEE Journal on Selected Areas in Communications, 2017.

[6] . A. M. Giordani, M. Mezzavilla, S. Rangan, and M. Zorzi, *Multi-connectivity in 5G mmWave cellular networks*. 2016 Mediterranean Ad Hoc Networking Workshop, 2016.

[7] O. Belmoukadam, T. Spetebroot, and C. Barakat, *ACQUA: A user friendly platform for lightweight network monitoring and QoE forecasting*. The 3rd International Workshop on Quality of Experience Management, 2019.

[8] M. Khokhar, N. Saber, T. Spetebroot, and C. Barakat., *An Intelligent Sampling Framework for Controlled Experimentation and QoE Modeling*. Computer Networks,, 2018.

[9] T. Stokhammer, *Dynamic Adaptive Streaming over HTTP Design Principles and Standards*. ACM MMSys, 2011.

[10] M. Khokhar, T. Ehlinger, and C. Barakat., *From Network Traffic Measurements to QoE for Internet Video*. IFIP Networking Conference, 2019.

[11] *http://www.nsnam.org*.

[12] D. J. Vergados, A. Michalas, A. Sgora, and D. D. Vergados, *FDASH: A Fuzzy-Based MPEG/DASH Adaptation Algorithm*. IEEE Systems Journal, 2016.

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

[14] . A. S. Laga, T. V. Cleemput, F. V. Raemdonck, F. Vanhoutte, N. Bouten, M. Claeys, , and F. D. Turck, *Optimizing scalable video delivery throughOpenFlow layer-based routing*. IEEE NOMS,, 2014.

[15] . A. P.T.A.Quang, K.Piamrat, K.D.Singh, and C.Viho, *Videostreaming over ad hoc networks: A qoe-based optimal routing solution*. IEEE Tran. on Veh. Tech, 2017.

[16] G. Calvigioni, R. Aparicio-Padro, and L. Sassateli, *Quality of Experience-based routing of video traffic for overlay and ISP networks*. IEEE INFOCOM, 2018.

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

[18] A. Asan, W. Robitza, E. I. I. Mkwawa, L. Sun, and A. Raake, *IMPACT OF VIDEO RESOLUTION CHANGES ON QoE FOR ADAPTIVE VIDEO STREAMING*. IEEE International Conference on Multimedia and Expo ICME, 2017.

[19] *https://docs.scipy.org/doc/scipy-0.18.1/reference/optimize.minimize-slsqp.htmloptimize-minimize-slsqp*.

[20] *https://www.ibm.com/products/ilog-cplex-optimization-studio*.

[21] D. J. Vergados, A. Michalas1, and A. Sgora, *A control-based algorithm for rate adaption in MPEG-DASH*. 5th International Conference on Information, Intelligence, Systems and ApplicationsM, 2014.

[22] *http://www-itec.uni-klu.ac.at/ftp/datasets/DASHDataset2014/ Elephants-Dream/*.