Perceptually Optimized ABR Ladder Generation for Web Streaming

Yuriy A. Reznik, Karl O. Lillevold, and Rahul Vanam; Brightcove, Inc.; Seattle, WA/USA

Abstract

We discuss the problem of the design of encoding profiles for web streaming. In this application, the video is normally embedded in a web page, and based on user preferences, e.g. browser stretch factor, full-screen mode, etc., the area that it occupies on the screen may be different. When multiple viewers tune to the same web page, this creates a distribution of possible player sizes. The key idea of this paper is to consider such distribution as input to the problem of the design of encoding profiles for web streaming. The objective is to maximize the average quality that can be experienced by a population of viewers. We define this problem mathematically, show that it belongs to a class of non-linear constrained optimization problems, and show how it can be solved practically. Examples of optimal profiles generated for different videos, networks, and player models are also provided. Provided results demonstrate the significance of accounting for player size distributions in the design of encoding profiles for web streaming.

1. Introduction

In Adaptive Bit-Rate (ABR) streaming systems [1, 2, 3], the content is typically encoded at several bitrates, and where each encoded *stream* (or *rendition*) incorporates random access points (e.g. I- or IDR-frames), allowing players to switch between them. During the playback, a player (or *streaming client*) monitors the rate at which encoded content is arriving. If such a rate becomes insufficient for continuous playback, the client switches to a lower bitrate stream. This prevents buffering. On the other hand, if the network rate becomes greater than the bitrate of the current stream, the client may switch to a higher one. This improves quality. Such an adaptation mechanism is now widely adopted, and incorporated in most modern streaming protocols, such as HLS [4], MPEG DASH [5], etc.

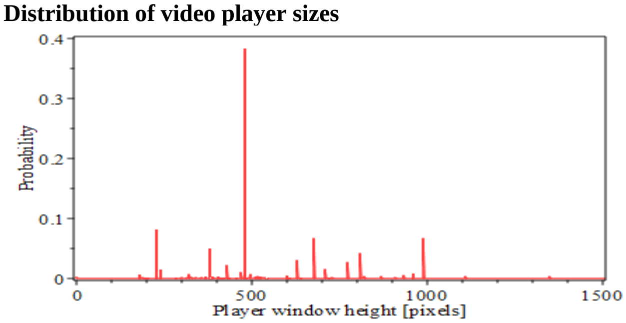
The composition of characteristics of streams used for ABR streaming, such as their bitrates, resolutions, codec constraints, etc. is commonly called an *encoding profile* or a *ladder*. When first ABR streaming systems were deployed, the encoding profiles were very simple: they typically included 28k, 56k, and 128k streams, corresponding to connection speeds achievable by dial-up and ISDN modems [2]. When faster connections become available, the encoding profiles were extended to include a few higher-bitrate streams. Examples of typical for today's practice profiles can be found in HLS deployment guidelines [6].

In recent years, it was also discovered that the performance of ABR streaming systems can be significantly improved by using *dynamic ABR profile generators*, producing encoding profiles customarily for each video asset, by considering rate-distortion characteristics specific to the content and/or properties of networks used for delivery. Such approaches have become known as "per-title", "content-aware encoding", and "context-aware encoding" techniques [7-12].

However, when we consider web streaming, we notice another variable and related statistics, which apparently, have not yet been accounted for in ABR profile optimizations. This specific variable is *the size of a video player window* used to render the decoded video on the user's screen.

To explain the significance of this parameter, in Figure 1, we present the distribution of video player sizes and related playback statistics, collected by using the Brightcove analytics system [13] during the streaming of the US Open event, June 10-16, 2019.

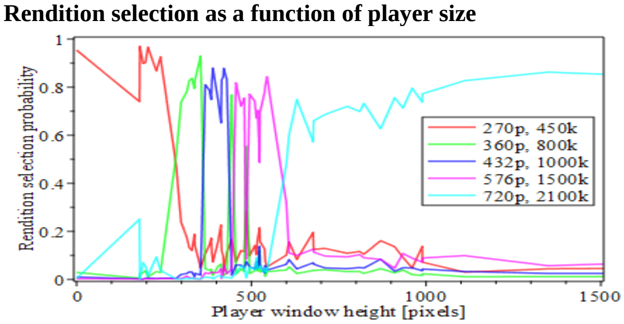
Distribution of video player sizes



Encoding ladder used for streaming

#	Codec	Profile	Resolution	Framerate	Bitrate
1	H.264	Baseline	480x270	23.976	450
2	H.264	Baseline	640x360	23.976	800
3	H.264	Main	768x432	23.976	1000
4	H.264	Main	1024x576	23.976	1500
5	H.264	Main	1280x720	23.976	2100

Rendition selection as a function of player size



Rendition selection as a function of network bandwidth

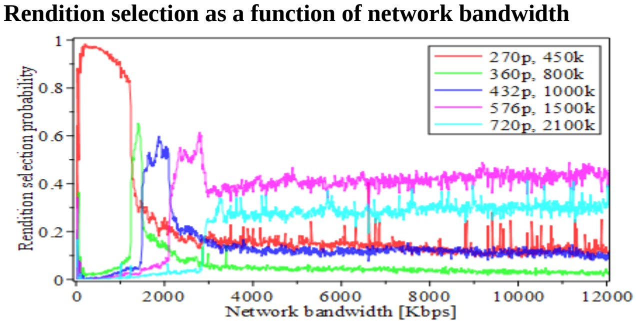


Figure 1. Playback statistics for streaming of US Open event, June 13, 2019, www.usopen.com

As shown in Figure 1, the distribution of player resolutions exhibits several highly distinct peaks, with 480p being the most pronounced. We also see that player sizes do have a significant impact on stream selection logic. E.g., from the bottom left plot, we see that 270p rendition was loaded most frequently when player window sizes were ~300 lines or less. Similarly, 720p rendition was loaded most frequently when player sizes were ~600 lines and beyond. From the bottom right plot, we further see that, as expected, clients also switch streams based on the available network bandwidth. However, in the high-bitrate regime, we notice that it is not the highest bitrate stream (720p, 210Kbps) that becomes used exclusively, but rather a particular mix of all renditions, apparently shaped by the distribution of player sizes.

In other words, we see that player sizes affect the choices of streams delivered by the system. Additionally, and even more importantly, player sizes also affect the *perception of the quality* of videos rendered on the screen. Smaller videos are perceived as having lower quality. This is well known from studies by Westerink and Roufs [15], Lund [14], Barten [16,17], and others [18-22]. These are significant effects, which ought to be accounted for in the analysis and optimization of web streaming systems.

This paper is dedicated to the study of this topic. In Section 2, we offer definitions of all involved variables and models. In Section 3, we derive an expression for the *average quality* delivered by the streaming system. In Section 4, we pose and study the problem of optimal design of encoding profiles. In Section 5, we present and analyze examples of optimal profiles generated. In Section 6, we draw conclusions.

2. Definitions and Main Models

2.1. Parameters of Encoded Videos. Ladders

Hereafter, by letters W and H we will denote video *width* and *height* respectively, by R we will denote the *bitrate* at which video is encoded, and by D we will denote *distortion* introduced by the encoder in the encoding process. Distortion can be measured in PSNR, SSIM [23], or any other codec noise metric.

When a given video sequence is encoded, this effectively produces a point (H, R, D), describing a tradeoff between video resolution, bitrate, and distortion level achieved. When the same video is encoded with several different target resolutions and bitrates, this effectively produces a set of points

$$\mathcal{L} = \{(H_i, R_i, D_i),\quad i = 1, \dots, n\}. \quad (1)$$

We will call such set an *encoding ladder*. Each encoding/point in the ladder we will call a *rendition*. By letter n we will denote the number of renditions in the ladder.

We will say that the encoding ladder $\mathcal{L}$ is *proper* if its rendition bitrates are monotonically increasing:

$$0 < R_1 < \dots < R_n, \quad (2)$$

and if the respective rendition resolutions are also non-decreasing:

$$0 < H_1 \le \dots \le H_n. \quad (3)$$

We will also assume that the aspect ratios W_i/H_i of all renditions in the proper ladder are the same.

2.2. Perceptual Quality Model

We next consider a situation when playback is done by a web player, projecting received and decoded video into a certain area on a PC screen. By W_p and H_p we will denote the *width* and *height* of the video player respectively and will assume that *display pixel density* ρ, as well as *viewing distance* d are also known. For example, we may assume that ρ=96 [dpi] and that d=24 [in] as typical for PC screens and PC viewing practices.

Given all these parameters, we will next define a model for predicting *quality scores* reported by human observers. For this purpose, we will employ a model of Westerink and Roufs [14] quantifying the impact of video resolution and its projection/scaling on the perceived quality, and we will also use SSIM metric [23] to quantify the impact of codec-introduced noise. The combination of resolution- and noise-based factors in this model is multiplicative, as suggested in [22].

In more exact terms, our proposed model is defined as follows:

$$Q(H, H_p, D) = \alpha\left(\beta + Q_{WR}\left(\phi(H_p),\, u(H, H_p)\right)\right) f(D), \quad (4)$$

where:

$$\phi(H_p) = 2\arctan\left(\frac{H_p\, DAR}{2d\rho}\right), \quad (5)$$

is the *viewing angle* to video on the screen, $DAR = W_p/H_p$,

$$\phi_c(H, H_p) = 2\arctan\left(\frac{H_p/\min(H, H_p)}{d\rho}\right), \quad (6)$$

is the *viewing angle capturing 2-pixels interval* (length of a smallest feasible "cycle") in the projected video,

$$u(H, H_p) = \frac{1}{\phi_c(H, H_p)}, \quad (7)$$

is the *angular resolution* [in cycles per degree] of video projected to the screen,

$$Q_{WR}(\phi, u) = 3.6\log_{10}\left(\phi\frac{\pi}{180}\right) + 2.9 + 4.6\log_{10}(u) + 2.7\log_{10}(u)^2 - 1.7\log_{10}(u)^3, \quad (8)$$

is the Westerink and Roufs model [14], establishing the relationship between ϕ, u and MOS scores, and

$$f(D) = \mathrm{e}^{\gamma D}, \quad (9)$$

is a mapping function, translating the distortion D, measured in SSIM units [23], to the MOS domain [24].

Parameters α, β, and γ are the calibration constants used to fit this model to MOS scores in training data sets. For example, by fitting this model to data present in the Netflix data set [25,26] we arrive at constants α=0.1075, β=−4.859, and γ=2.424467. The RMSE achieved by this model on this dataset is 0.329 on a 1-5 MOS scale, which is reasonable and compares well to other metrics tested with the same dataset [25].

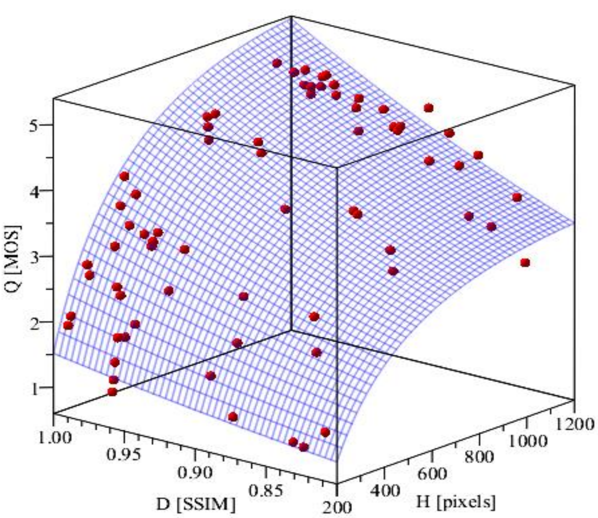


Figure 2. The fit of the proposed model to MOS scores in the Netflix dataset.

Importantly, the proposed model $Q(H, H_p, D)$ computes quality scores based not only on noise and resolution of the encoded video but also based on *player window size* H_p, which is a parameter of particular interest in this study.

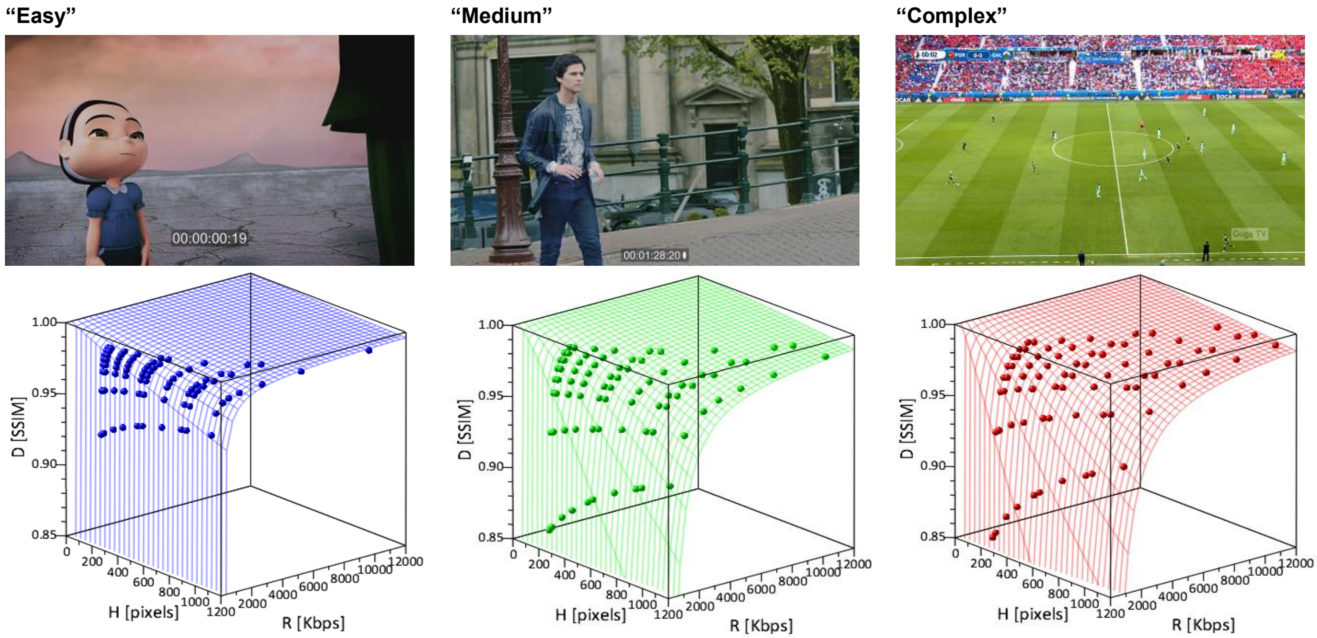


Figure 3. Probe points and distortion-rate functions constructed for 3 video sequences.

2.3. Distortion-Rate and Quality-Rate Functions

Consider now a set of points $(H_i, R_i, D_i), i = 1, \dots, m$, corresponding to outcomes of m "probe" encodes generated by same encoder and same content. Assume also that video resolutions and rates of such encodes cover some practically relevant operating range: $(H_i, R_i) \in [H_{\min}, H_{\max}] \times [R_{\min}, R_{\max}]$.

We next define a model function $D(H, R)$ approximately matching distortion values in all these points:

$$D(H_i, R_i) \approx S_i, \quad i = 1, \dots, m. \tag{10}$$

Specifically, when working with H.264 [27] codec and SSIM metric [23], we will use the following model:

$$D(H, R) = \left(1 + \left(\frac{R}{\alpha H^{\beta}}\right)^{-\gamma}\right)^{-\frac{1}{\gamma}}, \tag{11}$$

where α, β, γ are the parameters providing fit it to probe points.

For example, in Figure 3, we show the results of fitting this model to experimental data obtained for 3 different video sequences. We will call these sequences "Easy", "Medium", and "Complex". In all cases, probe encodes have been done by using the x264 encoder [28], operating using Main profile, level 4, 2sec GOPs, with CRF=[16,18,20,22,24,26,30,36] and target resolutions (heights) H=[270,288,360,432,540,576,720,864,900,1080]. The resulting best-fitting values of α, β, γ model parameters, as well as RMSE accuracy numbers, are shown in Table 1. It can be observed that model (11) predicts true SSIM values reasonably well.

Table 1: Parameters of empirical rate-distortion functions

Sequence	α	β	γ	RMSE
"Easy"	0.7844e-3	1.2281	0.7463	0.3404e-2
"Medium"	0.8278e-2	1.3217	0.9593	0.2792e-2
"Complex"	0.07316	1.0957	1.0336	0.1153e-2

By analogy with the concept of *distortion-rate function* in information theory [29], we will call model $D(H, R)$ an *empirical distortion-rate function*. Unlike true distortion-rate function, $D(H, R)$ is not derived analytically, but it serves a similar objective. It describes the space of achievable distortion-rate tradeoffs when encoding a given video sequence by a given video encoder.

Next, given empirical distortion-rate function (11), and reproduction quality model (4), we can define a model describing the *space of achievable quality-rate tradeoffs* when working with a particular encoder, content, and a player:

$$Q(H, H_p, R) = Q(H, H_p, D(H, R)). \tag{12}$$

We will call this function a *quality-rate model.* We show plots of this function for our "Complex" sequence and player sizes $H_p \in \{270, 540, 1080\}$ in Figure 4.

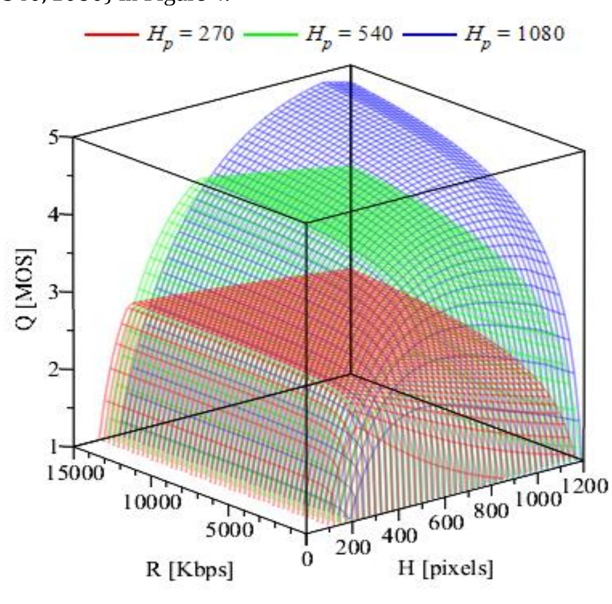


Figure 4. 3D projections of $Q(H, H_p, R)$ model for the "Complex" sequence.

This model will play a key role in our definition of *quality-optimal encoding ladders*, and related optimization problems.

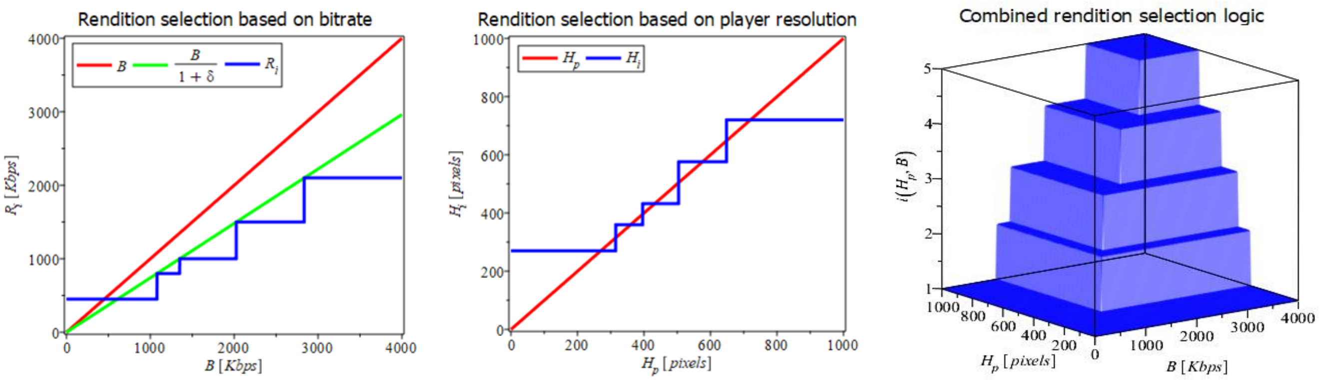


Figure 5. Construction of streaming client model. Left: rendition selection based on the available network bandwidth B. Middle: rendition selection based on player window size H_p. Right: the combined rendition selection logic.

2.4. Streaming Client Model

As we have already seen in Figure 1, streaming clients make decisions about streams to use based at least on two parameters: available network bandwidth, and player window size. We will first propose adaptation models for each variable and then will offer a combined model.

To describe client adaptation to network bandwidth, we will use the following model:

$$i_B(B) = \begin{cases} 1 & if \quad B < T_1^B \\ i & if \quad T_i^B \leq B < T_{i+1}^B, \quad i = 2, .., n-2, \\ n & if \quad B \geq T_{n+1}^B \end{cases} \tag{13}$$

$$T_i^B = (1+\delta) R_{i+1}, \quad i = 1..n-1, \tag{14}$$

where $i_B(B)$ denotes the index of rendition selected, B is the available network bandwidth, R_i are ladder bitrates, T_i^B are bandwidth decision thresholds, and where $\delta \geq 0$ is a "bandwidth overhead" constant, used to make bandwidth-based decisions more conservative. E.g. based on statistics shown in Figure 1, it appears that $\delta \approx 0.35$. We show the plot of this model function in the left subfigure of Figure 5.

To describe client adaptation to player window, we will use the following model:

$$i_H(H_p) = \begin{cases} 1 & if \quad H_p < T_1^H \\ i & if \quad T_i^H \leq H_p < T_{i+1}^H, \quad i = 2, .., n-2, \\ n & if \quad H_p \geq T_{n-1}^H \end{cases} \tag{15}$$

$$T_i^H = \alpha H_i + (1-\alpha) H_{i+1}, i = 1..n-1 \tag{16}$$

where $i_H(H_p)$ denotes the index of rendition selected, H_p is the player height, H_i are the heights of renditions in the ladder, T_i^H are the player resolution-based decision thresholds, and where $\alpha \in (0,1)$ is a constant describing preference towards downscaling vs. upscaling. Based on Figure 1, it appears that $\alpha \approx 0.5$. We show a plot of this model in the middle subfigure of Figure 5.

Finally, when adapting to both network and player-size parameters, we will assume that player logic will be to pick the "safer" choice:

$$i(B, H_p) = \min \{ i_R(B), i_H(H_p) \}. \tag{17}$$

As easily observed, with a proper ladder, this logic results in the selection of renditions with rates always below the available network bandwidth, and resolutions below decision points based on player window size. We plot this model function in the right subfigure in Figure 5.

In passing, we must note that the proposed model is indeed an extreme simplification of the logic that may be implemented by practical streaming clients. It intentionally ignores all temporal effects, client buffer size and state, pre-roll duration, specifics of bandwidth estimation algorithm, and other design aspects. But despite its simplicity, it provides the logic that is feasible and adequate for modeling key effects needed for the average-case analysis of streaming systems.

3. Average Performance of Web-Streaming

Given all models defined in the prior section, we will next derive the *average performance characteristics of the streaming system.*

For this purpose, we will need to consider two input distributions:

- *network bandwidth distribution*: $p(B)$, where B is assumed to be a continuous random variable in $[0, \infty)$, and
- *distribution of player sizes*: $q(H_p)$, where player height H_p is assumed to be a discrete random variable, taking values from a certain set: $H_p \in \mathcal{H}_p$.

In practice, both distributions are usually known or can be measured by streaming analytics systems. Of interest, indeed, are steady-state statistics, observed over a considerable time window, and which are specific to a particular region, network and CDN operators, event, web page, and audience.

Given all these ingredients, we now can write an expression for the *average quality* delivered by a streaming system using an n-point ladder with rendition bitrates $R_1, .., R_n$ and resolutions $H_1, \dots, H_n$ respectively:

$$\bar{Q}(H_1, \dots, H_n, R_1, .., R_n, p, q) = \int_0^\infty p(B) \sum_{H_p \in \mathcal{H}_p} q(H_p)\, Q\left(H_{i(B,H_p)}, H_p, R_{i(B,H_p)}\right) dB. \tag{18}$$

In this expression, the averaging across network bandwidth B is done by the first integral, then the averaging across all possible player resolutions H_p is done by the following sum, and where the reproduction quality value $Q(H_i, H_p, R_i)$ for each pair of values B, H_p is computed by retrieving the index of the selected rendition $i = i(B, H_p)$, and its bitrate R_i and resolution H_i parameters.

Such average quality is achievable for a specific encoder and content as described by the quality-rate model $Q(H, H_p, R)$ employed in (18). It is also specific to a network model $p(B)$ and player size distribution $q(H_p)$ used to compute it.

Similarly, we can also compute many other average performance parameters of the streaming system. For example:

- *the average resolution* of the video, as delivered:

$$\bar{H}(H_1,\dots,H_n,R_1,\dots,R_n,p,q) = \int_0^\infty p(B) \sum_{H_p \in \mathcal{H}_p} q(H_p)\ H_{i(B,H_p)} dB, \quad (19)$$

- *average distortion* in the video, as delivered:

$$\bar{D}(H_1,\dots,H_n,R_1,\dots,R_n,p,q) = \int_0^\infty p(B) \sum_{H_p \in \mathcal{H}_p} q(H_p)\, D_{i(B,H_p)} dB, \quad (20)$$

- *average player size*:

$$\bar{H}_p(q) = \sum_{H_p \in \mathcal{H}_p} q(H_p)\, H_p, \quad (21)$$

- *average network bandwidth used for streaming*:

$$\bar{R}(H_1,\dots,H_n,R_1,\dots,R_n,p,q) = \int_0^\infty p(B) \sum_{H_p \in \mathcal{H}_p} q(H_p)\ R_{i(B,H_p)} dB, \quad (22)$$

- *average available network bandwidth:*

$$\bar{B}(p) = \int_0^\infty p(B)\, B\ dB, \quad (23)$$

and so on.

4. Design of Quality-Optimal Ladders

4.1. The Problem

Given that we already know how to compute average quality (18), and we see that it is effectively a function of parameters $H_1,\dots,H_n$ and $R_1,\dots,R_n$ of the encoding ladder, we can next pose the following problem.

For a given codec, video content, as well as player and network distributions p and q, find parameters $\hat{H}_1,\dots,\hat{H}_n$ and $\hat{R}_1,\dots,\hat{R}_n$ of an encoding ladder, such that the average quality $\overline{Q}(\hat{H}_1,\dots,\hat{H}_n,\ \hat{R}_1,\dots,\hat{R}_n,p,q)$ delivered by an ABR streaming system is maximal:

$$\overline{Q}(\hat{H}_1,\dots,\hat{H}_n,\ \hat{R}_1,\dots,\hat{R}_n,p,q) = \max_{\substack{R_{\min} \le R_1 < \dots < R_n \le R_{\max} \\ H_{\min} \le H_1 \le \dots \le H_n \le H_{\max} \\ R_1 \le R_{1,\max},\ H_1 \le H_{1,\max}}} \overline{Q}(H_1,\dots,H_n,\ R_1,\dots,R_n,p,q)\ . \quad (24)$$

The added constraints include the following:

- the requirement for the ladder to be proper: $R_1 < \dots < R_n$ and $H_1 \le \dots \le H_n$
- restriction of ladder bitrates to a certain operating range: $R_{\min} \le R_i \le R_{\max}$
- restriction of ladder resolutions to a certain operating range: $H_{\min} \le H_i \le H_{\max}$
- upper limits on bitrate and resolution of the first rendition: $R_1 \le R_{1,\max}$ and $H_1 \le H_{1,\max}$.

The last constraints (bounds for rate and resolution of 1st rendition) effectively establish the lowest quality operating point that system must be able to support. They also influence some other characteristics, such as start-up time, buffeting probability, etc.

In practice, many additional constraints may also be added. For example, the video resolutions may be required to be part of a certain allowed set $H_i \in \mathcal{H}$, such resolutions may also be required to be distinct (i.e. strictly growing across the ladder): $H_1 < \dots < H_n$, the granularity of rate and resolution changes from one stream to another may also be bounded, and so on.

4.2. Analysis of The Problem

We first observe that problem (24) belongs to a class of non-linear constrained optimization problems. To solve it, we may indeed try to use existing numerical optimization techniques, such as e.g., *sequential quadratic programming* [30], but before, we must explore if it can be simplified, or at least rewritten in a more convenient form.

With this objective, let us now revisit the client decision model (17), and define two inverse mappings: from rendition indices to applicable ranges of bitrates or sets of player resolutions:

$$B(H_p,j) = \{B:\ i(B,H_p) = j\}, \quad j = 1,\dots,n. \quad (25)$$

$$\mathcal{H}_p(B,j) = \{H_p \in \mathcal{H}_p:\ i(B,H_p) = j\}, \quad j = 1,\dots,n. \quad (26)$$

Using these sets, we can write two alternative expressions for the average quality (18):

$$\bar{Q}(H_1,\dots,H_n,R_1,\dots,R_n,p,q) = \sum_{i=1}^{n} \sum_{H_p \in \mathcal{H}_p} q(H_p)\, Q(H_i,H_p,R_i) \int_{B \in B(H_p,i)} p(B)\, dB \quad (27)$$

$$\bar{Q}(H_1,\dots,H_n,R_1,\dots,R_n,p,q) = \sum_{i=1}^{n} \int_0^\infty p(B) \sum_{H_p \in \mathcal{H}_p(B,i)} q(H_p)\, Q(H_i,H_p,R_i)\, dB. \quad (28)$$

These expressions are somewhat easier to understand than (18). Thus, from (27) we can extract direct formulae for probabilities of each rendition:

$$\Pr(i|H_p) = \int_{B \in B(H_p,i)} p(B)\, dB. \quad (29)$$

From (28) we also see that in certain regions $H_p \in \mathcal{H}_p(B,i)$ our quality expression $Q(H_i,H_p,R_i)$ collapses to something simpler:

$$\bar{Q}_{H_p \in \mathcal{H}_p(B,i)}(H_i,R_i) = \textstyle\sum_{H_p \in \mathcal{H}_p(B,i)} q(H_p)\, Q(H_i,H_p,R_i) \quad (30)$$

However, this does not yield some average quantity expression across all player resolutions:

$$\bar{Q}(H_i,R_i) = \sum_{H_p \in \mathcal{H}_p} p(H_p)\, Q(H_i,H_p,R_i), \quad (31)$$

as ranges of the respective sums in (28) are different, and in (27) a similar-looking sum also involves integrals depending on H_p. This indicates that full quality model $Q(H_i,H_p,R_i)$ is needed and that the impact of player resolutions H_p cannot be trivialized or otherwise removed from this problem.

Next, let us check if optimal choices of rendition resolutions $H_1,\dots,H_n$ can be directly derived based on best choices of rates $R_1,\dots,R_n$. Indeed, by looking at shapes of quality-rate functions (see Figure 4), we notice, that they exhibit maxima points for H. Hence, one may think that optimal ladder resolutions $\hat{H}_i$ could be computed by simply taking optimal values of H_i for quality-rate function:

$$Q(\hat{H}_i,H_p,R_i) = \max_{H_i} Q(H_i,H_p,R_i). \quad (32)$$

This was a key argument put forward in Netflix's "per-title" approach [7].

However, if we look closely at average quality expression (27), we notice, that resolutions $H_1,\dots,H_n$ affect not only the average quality values $Q(H_i,H_p,R_i)$, but also rendition probabilities (29), since integral ranges $B \in B(H_p,i)$ are also depend on parameters $H_1,\dots,H_n$! So, the resolutions found by (32) may not be optimal in the context of a problem that we are solving.

In other words, we see that this problem cannot be trivially reduced to only n dimensions $R_1,\dots,R_n$. Rendition resolutions

$H_1, \ldots, H_n$ in this problem act as additional degrees of freedom, and must be treated as such by the solution finder.

4.3. Finding the Solution

As already observed, problem (24) belongs to a class of non-linear constrained optimization problems. It can be possibly be solved by using existing numerical optimization techniques [30]. However, we must also note that the objective function (18) in this case is fundamentally not a "smooth" function. It is affected by the decision model of a client (17), which introduces discontinuities. Hence, we may expect convergence problems.

To solve this problem practically, and with high certainty that the global maximum point is not missed, we can also try to turn it into a discrete domain:

- use fixed set of standard resolutions that can be allowed as rendition resolution choices: $H_i \in \mathcal{H}$
- use fixed lattice of bitrate points, e.g. by employing a 1% increase chain: $R_i \in R = \{R_{\min} \cdot 1.01^k, k = 0,1, \ldots, k_{\max}\}$

and then apply combinatorial enumeration of all possible ladder points $(H_1, \ldots, H_n, R_1, \ldots, R_n)$ and maximum selection. With modern computers and limited ladder sizes (e.g. n=1…7), it may be solvable in a practically acceptable time.

This is the method that we have implemented for producing experimental results shown in the next section.

5. Experimental Results

5.1. Test Conditions

To test the effectiveness of our modeling and profile optimization technique, we will next consider several test conditions, exhibiting:

- differences in the "encoding complexity" of video content,
- differences in networks,
- differences in players.

To test adaptability to video content we will use 3 video sequences, already considered in Section 2.3. We will call them "Easy", "Medium", and "Complex" respectively. Their empirical distortion-rate functions are shown in Figure 3, and Table 1 lists parameters of models used to describe them.

To test streaming system behavior under different networks, we will consider 2 models shown in Figure 6.

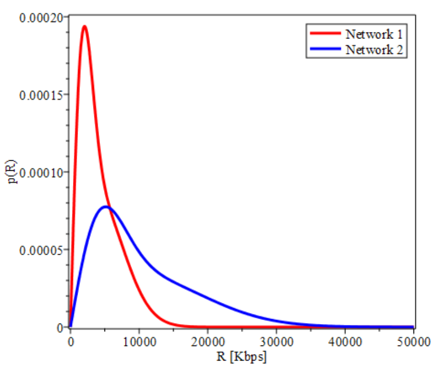


Figure 6. Network models used in the tests.

Both network distributions are realizations of the following model:

$$p(B) = \alpha\, f(B, \sigma_1) + (1 - \alpha)\, f(B, \sigma_2), \quad (33)$$

where

$$f(x, \sigma) = \frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}} \quad (34)$$

is the Rayleigh distribution, and $\alpha, \sigma_1, \sigma_2$ are the model parameters. Specific values of these parameters in cases of Networks 1 and 2 are shown in Table 2.

Table 2. Parameters of network models used in the tests.

Network	α	σ_1	σ_2	$\bar{B}$
"Network 1"	0.4287	1802.2	4499.28	4189.87
"Network 2"	0.4287	4,505.5	11,248.2	10474.7

Table 2 also lists the effective average bandwidth values $\bar{B}$ achievable by these networks.

To test system performance under different players we will use 2 player models with different possible player sizes $H_p \in \mathcal{H}_p$ and related distributions $q(H_p)$, as listed in Table 3.

Table 3. Player models.

Player	Player sizes $\mathcal{H}_p$	Player size probabilities $q(H_p)$
"1080p player"	{1080}	{1.0}
"Web player"	{228, 240, 380, 430, 480, 630, 678, 710, 774, 810, 990}	{0.103188906, 0.017734224, 0.062664264, 0.026945508, 0.480776451, 0.038259368, 0.083865235, 0.018247353, 0.033203174, 0.051450527, 0.08366499}

In the above table, the "1080p player" is a model of a player that always stretches video to 1080p window during playback, while the "Web player" is a model representing a population of web players having 11 possible resolutions and respective probabilities of each. This model is based on real-world playback statistics shown in Figure 1.

5.2. Ladders Generated and Parameters Measured

For all combinations of the above listed (content, network, player) models we have subsequently generated optimal encoding profiles. The following constraints have been imposed in all cases:

- $R_{\min}$ = 100 [Kbps]: minimum allowed bitrate
- $R_{\max}$ = 5050 [Kbps]: maximum allowed bitrate
- $R_{1.\max}$ = 180 [Kbps]: maximum allowed bitrate for the first rendition
- $H_{1.\max}$ = 480 [pixels]: maximum allowed resolution (height) for first rendition
- $\mathcal{H}$ = {216, 270, 288, 360, 432, 480, 540, 576, 720, 900, 1080}: allowed list of resolutions
- $i < j \Rightarrow R_i < R_j$: strictly increasing bitrates
- $i < j \Rightarrow H_i < H_j$: strictly increasing resolutions
- $n = 1,\ldots,5$: numbers of renditions generated.

For each profile the following average performance parameters have been computed and reported:

- $\bar{Q}$: the average quality [MOS] achieved by the system (18),
- $\bar{R}$: the average network bandwidth [Kbps] used (22),
- $\bar{H}$: the average encoded resolution [height] of video streams delivered to viewers (19),
- $\bar{D}$: the average distortion [SSIM] in encoded video streams delivered to viewers (20)
- $\bar{H}_p$: the average player size [height] used by a population of viewers (21).

For comparison, all same performance parameters have also been computed for a reference encoding profile, shown in Figure 1.

Table 4.1: Optimal ladders for "Easy" Content, Network 1, 1080p player

Optimal ladders:										
n	**Rendition 1**	**Rendition 2**	**Rendition 3**	**Rendition 4**	**Rendition 5**	$\bar{H}$	$\bar{H}_p$	$\bar{D}$	$\bar{Q}$	$\bar{R}$
1	854x480 180k					480.0	1080	0.9629	3.230	180.0
2	854x480 180k	1920x1080 899k				1043.1	1080	0.9754	4.843	854.8
3	854x480 180k	1600x900 427k	1920x1080 1440k			1047.7	1080	0.9805	4.942	1288.9
4	854x480 180k	1280x720 228k	1600x900 584k	1920x1080 1557k		1044.2	1080	0.9817	4.954	1385.1
5	854x480 167k	1024x576 173k	1280x720 277k	1600x900 607k	1920x1080 **1557k**	1043.7	1080	0.9819	**4.955**	**1388.4**
Reference:										
5	480x270 450k	640x360 800k	768x432 1000k	1024x576 1500k	1280x720 2100k	647.8	1080	0.9910	**4.075**	**1846.5**

Table 4.2: Optimal ladders for "Medium" Content, Network 1, 1080p player

Optimal ladders:										
n	**Rendition 1**	**Rendition 2**	**Rendition 3**	**Rendition 4**	**Rendition 5**	$\bar{H}$	$\bar{H}_p$	$\bar{D}$	$\bar{Q}$	$\bar{R}$
1	854x480 180k					480.0	1080	0.8466	2.436	180.0
2	854x480 180k	1920x1080 1440k				992.6	1080	0.9227	4.186	1256.4
3	854x480 180k	1600x900 865k	1920x1080 2305k			1000.3	1080	0.9416	4.431	1819.8
4	854x480 180k	1280x720 584k	1600x900 1280k	1920x1080 2697k		983.4	1080	0.9501	4.496	2061.6
5	854x480 180k	1024x576 410k	1280x720 769k	1600x900 1384k	1920x1080 **2804k**	975.4	1080	0.9534	**4.512**	**2130.5**
Reference:										
5	480x270 450k	640x360 800k	768x432 1000k	1024x576 1500k	1280x720 2100k	647.8	1080	0.9718	**3.891**	**1846.5**

Table 4.3: Optimal ladders for "Complex" Content, Network 1, 1080p player

Optimal ladders:										
n	**Rendition 1**	**Rendition 2**	**Rendition 3**	**Rendition 4**	**Rendition 5**	$\bar{H}$	$\bar{H}_p$	$\bar{D}$	$\bar{Q}$	$\bar{R}$
1	854x480 180k					480.0	1080	0.7533	1.943	180.0
2	854x480 180k	1920x1080 1557k				980.0	1080	0.8911	3.911	1327.4
3	854x480 180k	1600x900 973k	1920x1080 2493k			987.9	1080	0.9199	4.217	1911.5
4	854x480 180k	1280x720 657k	1600x900 1440k	1920x1080 2917k		969.9	1080	0.9327	4.310	2164.6
5	854x480 180k	1024x576 480k	1280x720 899k	1600x900 1619k	1920x1080 **3155k**	954.4	1080	0.9392	**4.337**	**2288.0**
Reference:										
5	480x270 450k	640x360 800k	768x432 1000k	1024x576 1500k	1280x720 2100k	647.8	1080	0.9585	**3.774**	**1846.5**

5.3. The Results

5.3.1. Optimal Ladders for 1080p Players

In tables 4.1 through 4.3 we show optimal ladders constructed by assuming that playback is done by using a 1080p player. It is also assumed that delivery is done over Network 1. Separate profiles are generated for "Easy", "Medium", and "Complex" sequences.

As expected, we notice that optimal profiles generated for different content are different. The top rendition in the 5-point ladder for "Easy" content uses only 1557Kbps, while the same rendition for "Complex" content uses 3155Kbps.

In all cases, we also notice that at n=5 our optimal profiles deliver significantly better quality than the profile used as a reference. E.g. for "Easy" content we see an 0.92 increase on a 1-5 MOS scale.

We also notice that with the 1080p player model used as a target, the choices of resolutions in renditions are heavily biased up. For example, in 2-point profiles, all second renditions receive 1080p resolution, despite quite low bitrates and SSIM numbers. This is to be expected, as with reproduction being done at 1080p output, using more pixels and lower SSIM often produces better quality.

5.3.2. Optimal Ladders for Web-Players

We now repeat the same experiments as in the previous section, but by using a Web player as a receiver. The results are presented in tables 5.1 through 5.3.

As with profiles for 1080p players, here we also see that bitrates assigned to different videos are different. The top bitrates assigned in 5-point ladders look almost identical to ones we've seen before.

However, the choices of resolutions assigned to renditions targeting Web players are different. They are all much lower, striking a different balance between pixel-count and SSIM. All SSIM values are much higher in this case. The top renditions are also stopped at 900p and not 1080p as earlier.

The reason this is happening is indeed the mix of player sizes assumed by the web-player model. On average, it boils down to only 538.1 lines, and with the highest player resolution being used is 900p. It is also a very different playback model, where renditions are played at resolutions closely matching resolutions at which they are encoded. In this context, low distortion / high SSIM values become more important.

In all cases, we also notice that at n=5 our optimal profiles deliver considerably better quality than reference profile. In fact, for "Easy" content we see that even n=2 with an optimal assignment of rates and resolutions is sufficient to achieve a better quality than with 5 renditions in the reference profile.

5.3.3. Optimal Ladders for a Different Network

In this section, we will repeat experiments from section 5.3.2, but now by using "Network 2" as a network model. The results are presented in tables 6.1 through 6.3.

Table 5.1: Optimal ladders for "Easy" Content, Network 1, Web player

Optimal ladders:										
n	**Rendition 1**	**Rendition 2**	**Rendition 3**	**Rendition 4**	**Rendition 5**	$\bar{H}$	$\bar{H}_p$	$\bar{D}$	$\bar{Q}$	$\bar{R}$
1	854x480 180k					480.0	538.1	0.9629	3.310	180.0
2	854x480 180k	1280x720 935k				549.2	538.1	0.9690	3.567	397.6
3	768x432 180k	854x480 899k	1280x720 1052k			535.5	538.1	0.9818	3.666	756.1
4	768x432 180k	854x480 865k	1280x720 899k	1600x900 1557k		557.5	538.1	0.9816	3.705	773.8
5	512x288 180k	768x432 365k	854x480 935k	1280x720 973k	1600x900 **1557k**	537.7	538.1	0.9850	**3.719**	**1094.5**
Reference:										
5	480x270 450k	640x360 800k	768x432 1000k	1024x576 1500k	1280x720 2100k	465.2	538.1	0.9903	**3.563**	**1151.8**

Table 5.2: Optimal ladders for: "Medium" Content, Network 1, Web player

Optimal ladders:										
n	**Rendition 1**	**Rendition 2**	**Rendition 3**	**Rendition 4**	**Rendition 5**	$\bar{H}$	$\bar{H}_p$	$\bar{D}$	$\bar{Q}$	$\bar{R}$
1	854x480 180k					480.0	538.1	0.8466	2.496	180.0
2	640x360 180k	1024x576 1280k				510.4	538.1	0.9427	3.229	945.7
3	480x270 180k	854x480 973k	1280x720 1752k			500.5	538.1	0.9574	3.388	1019.0
4	480x270 180k	768x432 632k	854x480 1497k	1280x720 1895k		497.0	538.1	0.9625	3.444	1236.0
5	480x270 180k	768x432 632k	854x480 1497k	1280x720 1619k	1600x900 **2697k**	515.9	538.1	0.9617	**3.473**	**1262.3**
Reference:										
5	480x270 450k	640x360 800k	768x432 1000k	1024x576 1500k	1280x720 2100k	465.2	538.1	0.9701	**3.395**	**1151.8**

Table 5.3: Optimal ladders for "Complex" Content, Network 1, Web player

Optimal ladders:										
n	**Rendition 1**	**Rendition 2**	**Rendition 3**	**Rendition 4**	**Rendition 5**	$\bar{H}$	$\bar{H}_p$	$\bar{D}$	$\bar{Q}$	$\bar{R}$
1	768x432 180k					432.0	538.1	0.7748	2.008	180.0
2	384x216 180k	960x540 1183k				471.6	538.1	0.9351	3.049	971.3
3	480x270 180k	854x480 1094k	1280x720 2049k			493.0	538.1	0.9340	3.210	1130.2
4	480x270 180k	768x432 739k	854x480 1684k	1280x720 2216k		489.5	538.1	0.9426	3.289	1370.9
5	480x270 180k	768x432 739k	854x480 1684k	1280x720 1970k	1600x900 **3155k**	506.1	538.1	0.9420	**3.316**	**1407.7**
Reference:										
5	480x270 450k	640x360 800k	768x432 1000k	1024x576 1500k	1280x720 2100k	465.2	538.1	0.9522	**3.258**	**1151.8**

As easily observed, the use of Network 2 has caused all bitrates to increase. This network has 2.5x more bandwidth, and so more bits can be used to achieve better quality. What we see is reasonable and consistent with earlier results on ABR ladder designs, reported in [9,11,12].

5.3.4. Effects of Mistargeting the Player

Finally, in this section, we will compare the performance of ladders constructed for 1080p and Web-players, when the actual playback is done by Web players. Naturally, we would expect profiles designed for Web players to perform better. But the question is: by how much? Tables 7.1 through 7.3 provide the results.

As we can see from these tables, the difference in the average performance delivered by these profiles is quite dramatic! For instance, for complex content, we see a difference of over 0.8 MOS. We also notice a significant drop in network bandwidth usage: over a factor of 2, from 1407Kbps to 532Kbps. This happens when renditions in both profiles are covering about the same range of bitrates and in similar increments.

The obvious cause of this effect is a misallocation of bits and resolutions. In profiles generated for 1080p players, resolutions of all renditions are pushed up. They start at 480p, encoded very aggressively, and only at the final 1080p rendition they reach bitrate and SSIM level providing high-fidelity reproduction. When the 1080p players are used for playing them this all seems reasonable. Such players will always try to pull such highest bitrate renditions. However, when Web players are directed to play such content, their windows in the majority of cases are much smaller, centered around 400-500 pixels in height. With such player sizes, the highest quality renditions in this profile may never be selected! The decision logic will stop at rendition with the closest resolution. As a result, what will be selected with high probabilities are those few starting renditions that were encoded very aggressively. This way the delivered quality will stay poor, regardless of network capacity.

In other words, we see that using the right player models for the design of encoding profiles is extremely important. Failure to account for player's adaptation to their window sizes may lead to the creation of profiles where players may simply fail to deliver satisfactory quality even when network resources are no longer a constraining factor.

6. Conclusions

We have studied the problem of the design of encoding profiles for streaming systems with adaptation not only to network bandwidth but also to player sizes.

We have defined this problem mathematically, by introducing several models, accounting for the distribution of player resolutions, distribution of network bandwidth, and other factors influencing

Table 6.1: Optimal ladders for "Easy" Content, Network 2, Web player

Optimal ladders:										
N	**Rendition 1**	**Rendition 2**	**Rendition 3**	**Rendition 4**	**Rendition 5**	$\bar{H}$	$\bar{H}_p$	$\bar{D}$	$\bar{Q}$	$\bar{R}$
1	854x480 180k					480.0	538.1	0.9629	3.310	180.0
2	640x360 180k	1024x576 1684k				524.5	538.1	0.9867	3.598	1325.5
3	480x270 180k	854x480 1138k	1280x720 1970k			522.1	538.1	0.9888	3.725	1252.9
4	480x270 180k	854x480 1138k	1280x720 1822k	1920x1080 2697k		550.1	538.1	0.9886	3.766	1277.7
5	480x270 180k	768x432 657k	854x480 1895k	1280x720 1970k	1920x1080 **2697k**	545.9	538.1	0.9898	**3.781**	**1608.6**
Reference:										
5	480x270 450k	640x360 800k	768x432 1000k	1024x576 1500k	1280x720 2100k	486.5	538.1	0.9904	**3.653**	**1232.6**

Table 6.2: Optimal ladders for "Medium" Content, Network 2, Web player

Optimal ladders:										
n	**Rendition 1**	**Rendition 2**	**Rendition 3**	**Rendition 4**	**Rendition 5**	$\bar{H}$	$\bar{H}_p$	$\bar{D}$	$\bar{Q}$	$\bar{R}$
1	854x480 180k					480.0	538.1	0.8466	2.496	180.0
2	384x216 180k	960x540 1895k				488.2	538.1	0.9721	3.399	1620.8
3	480x270 180k	854x480 1752k	1280x720 3281k			512.5	538.1	0.9712	3.557	1922.9
4	480x270 180k	768x432 1094k	854x480 2917k	1280x720 3412k		508.6	538.1	0.9748	3.595	2399.2
5	480x270 180k	768x432 1052k	854x480 2804k	1280x720 2917k	1600x900 **4856k**	530.0	538.1	0.9741	**3.630**	**2421.0**
Reference:										
5	480x270 450k	640x360 800k	768x432 1000k	1024x576 1500k	1280x720 2100k	486.5	538.1	0.9704	**3.482**	**1232.6**

Table 6.3: Optimal ladders for "Complex" Content, Network 2, Web player

Optimal ladders:										
n	**Rendition 1**	**Rendition 2**	**Rendition 3**	**Rendition 4**	**Rendition 5**	$\bar{H}$	$\bar{H}_p$	$\bar{D}$	$\bar{Q}$	$\bar{R}$
1	768x432 180k					432.0	538.1	0.7748	2.008	180.0
2	384x216 180k	960x540 2131k				485.0	538.1	0.9565	3.287	1799.8
3	480x270 180k	854x480 1970k	1280x720 3690k			508.6	538.1	0.9548	3.442	2128.6
4	384x216 180k	768x432 1183k	854x480 3281k	1280x720 3838k		497.8	538.1	0.9646	3.498	2641.6
5	384x216 180k	768x432 1183k	854x480 3155k	1280x720 3281k	1600x900 **5050k**	519.1	538.1	0.9638	**3.531**	**2635.8**
Reference:										
5	480x270 450k	640x360 800k	768x432 1000k	1024x576 1500k	1280x720 2100k	486.5	538.1	0.9532	**3.347**	**1232.6**

average quality delivered by the streaming system. We have shown, that the resulting problem belongs to a class of non-linear constrained optimization problems, and offered a practical approach towards solving it.

The optimal profiles and performance numbers computed by using our approach, have also allowed us to observe several interesting phenomena. Thus, we've observed that encoding profiles generated for web-players exhibit very different tradeoffs between pixel count and codec-introduced distortion values as compared to profiles generated for players always stretching videos to full-screen. We have also shown, that mistargeting player models in the design of encoding profiles may have very significant consequences for the performance of streaming systems.

Overall, this work confirms that optimal design of encoding profiles for web streaming must be done by utilizing distributions of player sizes, and brings forward a set of mathematical tools enabling such implementations.

References

[1] D. Wu, Y.T. Hou, W. Zhu, Y-Q. Zhang, and J.M. Peha, "Streaming video over the internet: approaches and directions," IEEE Trans. Circ. Syst. Video Technology, vol. 11, no. 3, pp. 282–300, 2001.

[2] G. J. Conklin, G. S. Greenbaum, K. O. Lillevold, A. F. Lippman, and Y. A. Reznik, "Video coding for streaming media delivery on the internet," IEEE Trans. Circ. Syst. Video Technology, vol. 11, no. 3, pp. 269–281, 2001.

[3] B. Girod, M. Kalman, Y.J. Liang, and R. Zhang, "Advances in channel-adaptive video streaming," Wireless Comm. and Mobile Comp., vol. 2, no. 6, pp. 573–584, 2002.

[4] R. Pantos, and W. May, "HTTP live streaming, RFC 8216," https://tools.ietf.org/html/rfc8216, August 2017.

[5] ISO/IEC 23009-1:2012, "Information technology – Dynamic adaptive streaming over HTTP (DASH) – Part 1: Media presentation description and segment formats," February 2012.

[6] Apple Inc., "HLS authoring specification for Apple devices," https://developer.apple.com/ documentation/http_live_streaming/hls_authoring_specification_for_apple_devices, Sept. 2018.

[7] A. Aaron, Z. Li, M. Manohara, J. De Cock, and D. Ronca, "Per-title encode optimization," https://medium.com/netflix-techblog/per-title-encode-optimization-7e99442b62a2, Dec. 15, 2015.

[8] Ultra HD Forum, "Ultra HD Forum phase B guidelines," https://ultrahdforum.org/wp-content/uploads/Ultra-HD-Forum-Phase-B-Guidelines-v1.0.pdf, April 2018.

Table 7.1: Comparison of ladders for “Easy” Content, Network 1, Web player

Optimal ladders for 1080p player:										
n	**Rendition 1**	**Rendition 2**	**Rendition 3**	**Rendition 4**	**Rendition 5**	$\bar{H}$	$\bar{H}_p$	$\bar{D}$	$\bar{Q}$	$\bar{R}$
1	854x480 180k					480.0	538.1	0.9629	3.310	180.0
2	854x480 180k	1920x1080 899k				556.1	538.1	0.9646	3.480	271.2
3	854x480 180k	1600x900 427k	1920x1080 1440k			570.1	538.1	0.9646	3.525	297.8
4	854x480 180k	1280x720 228k	1600x900 584k	1920x1080 1557k		590.0	538.1	0.9636	3.577	309.4
5	854x480 167k	1024x576 173k	1280x720 277k	1600x900 607k	1920x1080 **1557k**	584.4	538.1	0.9630	**3.569**	**306.3**
Optimal ladders for Web player:										
1	854x480 180k					480.0	538.1	0.9629	3.310	180.0
2	854x480 180k	1280x720 935k				549.2	538.1	0.9690	3.567	397.6
3	768x432 180k	854x480 899k	1280x720 1052k			535.5	538.1	0.9818	3.666	756.1
4	768x432 180k	854x480 865k	1280x720 899k	1600x900 1557k		557.5	538.1	0.9816	3.705	773.8
5	512x288 180k	768x432 365k	854x480 935k	1280x720 973k	1600x900 **1557k**	537.7	538.1	0.9850	**3.719**	**1094.5**

Table 7.2: Comparison of ladders for “Medium” Content, Network 1, Web player

Optimal ladders for 1080p player:										
n	**Rendition 1**	**Rendition 2**	**Rendition 3**	**Rendition 4**	**Rendition 5**	$\bar{H}$	$\bar{H}_p$	$\bar{D}$	$\bar{Q}$	$\bar{R}$
1	854x480 180k					480.0	538.1	0.8466	2.496	180.0
2	854x480 180k	1920x1080 1440k				549.3	538.1	0.8568	2.695	325.4
3	854x480 180k	1600x900 865k	1920x1080 2305k			564.3	538.1	0.8615	2.773	383.7
4	854x480 180k	1280x720 584k	1600x900 1280k	1920x1080 2697k		582.8	538.1	0.8707	2.894	457.5
5	854x480 180k	1024x576 410k	1280x720 769k	1600x900 1384k	1920x1080 **2804k**	576.4	538.1	0.8737	**2.915**	**485.6**
Optimal ladders for Web player:										
1	854x480 180k					480.0	538.1	0.8466	2.496	180.0
2	640x360 180k	1024x576 1280k				510.4	538.1	0.9427	3.229	945.7
3	480x270 180k	854x480 973k	1280x720 1752k			500.5	538.1	0.9574	3.388	1019.0
4	480x270 180k	768x432 632k	854x480 1497k	1280x720 1895k		497.0	538.1	0.9625	3.444	1236.0
5	480x270 180k	768x432 632k	854x480 1497k	1280x720 1619k	1600x900 **2697k**	515.9	538.1	0.9617	**3.473**	**1262.3**

Table 7.3: Comparison of ladders for “Complex” Content, Network 1, Web player

Optimal ladders for 1080p player:										
n	**Rendition 1**	**Rendition 2**	**Rendition 3**	**Rendition 4**	**Rendition 5**	$\bar{H}$	$\bar{H}_p$	$\bar{D}$	$\bar{Q}$	$\bar{R}$
1	854x480 180k					480.0	538.1	0.7533	1.991	180.0
2	854x480 180k	1920x1080 1557k				547.6	538.1	0.7720	2.224	335.0
3	854x480 180k	1600x900 973k	1920x1080 2493k			562.6	538.1	0.7808	2.321	400.6
4	854x480 180k	1280x720 657k	1600x900 1440k	1920x1080 2917k		581.1	538.1	0.7980	2.478	484.2
5	854x480 180k	1024x576 480k	1280x720 899k	1600x900 1619k	1920x1080 **3155k**	574.1	538.1	0.8028	**2.513**	**523.4**
Optimal ladders for Web player:										
1	768x432 180k					432.0	538.1	0.7748	2.008	180.0
2	384x216 180k	960x540 1183k				471.6	538.1	0.9351	3.049	971.3
3	480x270 180k	854x480 1094k	1280x720 2049k			493.0	538.1	0.9340	3.210	1130.2
4	480x270 180k	768x432 739k	854x480 1684k	1280x720 2216k		489.5	538.1	0.9426	3.289	1370.9
5	480x270 180k	768x432 739k	854x480 1684k	1280x720 1970k	1600x900 **3155k**	506.1	538.1	0.9420	**3.316**	**1407.7**

[9] Y. A. Reznik, K. O. Lillevold, A. Jagannath, J. Greer, and J. Corley, “Optimal design of encoding profiles for ABR streaming,” in Proc. Packet Video Workshop, Amsterdam, NL, June 12, 2018, pp. 43–47.

[10] C. Chen, Y. Lin, S. Benting, and A. Kokaram, “Optimized transcoding for large scale adaptive streaming using playback statistics,” in Proc. IEEE Int. Conf. Image Proc., Oct 2018, pp. 3269–3273.

[11] Y. A. Reznik, X. Li, K. O. Lillevold, A. Jagannath, and J. Greer, "Optimal Multi-Codec Adaptive Bitrate Streaming," in Proc. IEEE Int. Conf. Multimedia & Expo, Shanghai, China, 2019, pp. 348-353.

[12] Y. Reznik, X. Li, K. Lillevold, R. Peck, T. Shutt, and P. Howard, "Optimizing Mass-Scale Multi-Screen Video Delivery," SMPTE Motion Imaging Journal, vol. 129, no. 3, pp. 26-38, April 2020.

[13] Brightcove Video Analytics, https://www.brightcove.com/en/online-video-platform/analytics

[14] J. Westerink and J. Roufs, "Subjective image quality as a function of viewing distance resolution and picture size," SMPTE Journal, vol. 98, 1989, pp. 113-19.

[15] A. Lund, "The influence of video image size and resolution on Viewing-Distance preferences " SMPTE Journal, vol. 102, 1993, pp. 406-15.

[16] P. G. J. Barten, "Effect of picture size and definition on perceived image quality," IEEE Trans. Electron. Devices, vol. 36, no. 9, pp. 1865-1869, Sept. 1989.

[17] P. G. J. Barten, "Contrast Sensitivity of the Human Eye and Its Effects on Image Quality," SPIE Press, 1999.

[18] L. Jesty, "The relation between picture size viewing distance and picture quality," Proc. IEE, 1958, pp. 425-39.

[19] R. Fish and T. Judd, "A subjective visual comparison of NTSC VHS and Compressed DS1-Compatible video," in Proc. Society for Information Display (SID), vol. 32, no. 2, pp. 157-164, 1991.

[20] S. Pechard, M. Carnec, P. LeCallet, and D.Barba, "From SD to HD television: effects of H.264 distortions versus display size on quality of experience," in Proc. IEEE Int. Conf. Image Proc., Atlanta, GA, 2006, pp. 409-412.

[21] S. Bae, T. Pappas, and B. Juang, "Subjective Evaluation of Spatial Resolution and Quantization Noise Tradeoffs," IEEE Trans. Image Proc., vol. 18, no. 3, 2009, pp. 495-508.

[22] B. Belmudez and S. Moller, "An approach for modeling the effects of video resolution and size on the perceived visual quality", in Proc IEEE Int. Symp. Multimedia, Dana Point CA, 2011, pp. 464-469.

[23] Z. Wang, L. Lu, and A. C. Bovik, "Video quality assessment based on structural distortion measurement," Sig. Proc.: Image Comm., vol. 19, no. 2, pp. 121 – 132, 2004.

[24] U. Engelke, M. Kusuma, H-J. Zepernick, M. Caldera, "Reduced-reference metric design for objective perceptual quality assessment in wireless imaging," Sig. Proc.: Image Comm., vol. 24, no. 7, 2009.

[25] Z. Li, A. Aaron, et al., Toward A Practical Perceptual Video Quality Metric, June 2016, URL: https://netflixtechblog.com/toward-a-practical-perceptual-video-quality-metric-653f208b9652

[26] Netflix dataset: https://drive.google.com/drive/folders/0B3YWNICYMBIweGdJbERlUG9zc0k

[27] ISO/IEC 14496-10:2003, "Information technology –Coding of audio-visual objects – Part 10: Advanced Video Coding," December 2003.

[28] "x264 open-source encoder project," https://www.videolan.org/developers/x264.html.

[29] T. Cover, J. Thomas, "Elements of Information Theory", 2nd Ed., Wiley, Hoboken, NJ, 2006.

[30] J. Nocedal and S.J. Wright, Numerical Optimization, Springer-Verlag, New York, NY, 2006.

Author Biography

Yuriy A. Reznik is a technology fellow and head of research department at Brightcove, Inc., Seattle, WA, USA. Previously he worked at RealNetworks, Seattle, WA (1998-2005), Qualcomm, San Diego, CA, (2005-2011), and InterDigital, San Diego, CA (2011-2016), and contributed to several well-known standards (ITU-T H.264/MPEG-4 AVC, ITU-T H.265/MPEG HEVC, MPEG DASH, etc.) as well as products in the domains of internet streaming and wireless multimedia. He received the Ph.D. degree in computer science from Kyiv University, Kyiv, Ukraine, in 2005. In 2008, he was also a visiting scholar at Stanford University, Stanford, CA. He has co-authored more than 100 papers and co-invented more than 60 granted U.S. patents related to his studies and work.

Karl O. Lillevold is a distinguished engineer at Brightcove, Inc., Seattle, WA, USA. He is leading the work on context-aware encoding and other advanced video compression technologies in Brightcove products. Previously, he worked at Telenor Research, Oslo, Norway (1992-1995), Intel, Hillsboro, OR (1996-1999), and RealNetworks, Seattle, WA (1999-2016), and contributed to several generations of video coding standards (MPEG-2 H.263, H.26L/H.264/AVC), as well as to the family of RealVideo codecs used in pioneering products for internet streaming. He has received the M.S. degree in electrical engineering from the Norwegian Institute of Technology, Trondheim, Norway, in 1992.

Rahul Vanam received the B.E. degree in electronics and communication engineering from Bangalore University Bengaluru, India, in 2000, the M.S.E.E. degree from New Mexico State University, Las Cruces, NM, USA, in 2005, and the Ph.D. degree in electrical engineering from the University of Washington, Seattle, WA, USA, in 2010. From 2010 to 2019, he was with InterDigital, Inc., San Diego, CA, USA, where he was involved in research related to video compression and processing, and the emerging VVC standard. He is currently a Principal Engineer with Brightcove, Inc., San Diego, CA, USA. His current research interests include video compression, and video and image processing.