Average Performance of Adaptive Streaming

Yuriy A. Reznik

Brightcove, Inc.
601 Union Street, Suite 1601
Seattle, WA 98101, USA
yreznik@brightcove.com

***Abstract**:* This paper analyzes average behavior of video streaming systems with adaptation to network bandwidth and player sizes. The main results are formulae for average system performance parameters for given models of codecs, content, players and networks. Derived expressions are used to study performance limits achievable by adaptive streaming systems, and pose several related optimization problems. Numerical simulations, illustrating usefulness of the proposed formulae and techniques are also provided.

1. Introduction

Continuous playback under unknown or dynamically changing network conditions was the first and arguably most fundamental problem that early Internet streaming systems have tried to solve [1-3]. The first satisfactory solution was so-called "SureStream" technology, introduced by RealNetworks in 1998 [3]. The main idea was to encode media at multiple bitrates, and then have a system switching between such streams adaptively, as needed to match network bandwidth, available at each point in time. Same basic concept is well known today as Adaptive Bitrate Streaming, and it provides basis for modern streaming protocols and standards such as HLS and DASH [4,5].

Moreover, in modern days of streaming, there is another important parameter affecting the playback. It is the size of a video player window, or a "viewport" of a webpage embedding it. What causes its variation are user preferences and form factors of their devices.

To illustrate its effects, in Table 1, we list parameters of encoded streams, and in Figure 1, we present a set of playback statistics captured during streaming of USGA US Open event, June 10-16, 2019, streamed online at www.usopen.com.

Table 1. Encoding ladder used for streaming of US Open event.

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

As shown in this figure, the playback was done by a population of players with a particular distribution of window heights. About 11 sizes have been used most frequently. The bottom left subplot, shows the effects of adaptation to player sizes. The bottom right subplot shows adaptation to the network bandwidth. The interplay between both decision mechanisms can be observed in the high-bandwidth regime. Here, instead of using last stream exclusively, as normal bandwidth adaptation logic would, we see that all renditions are being loaded. We see a particular mix, shaped by player-size based decisions.

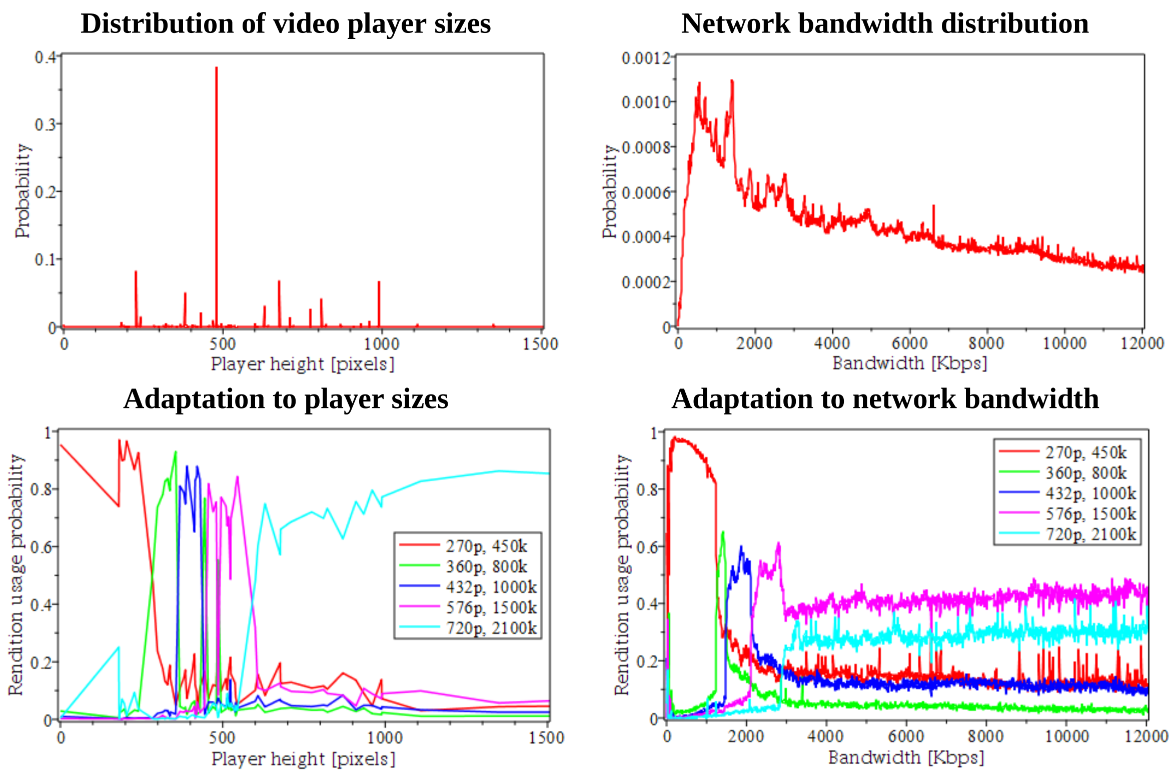

Figure 1: Playback statistics captured during streaming of US Open event.

Much of early research on streaming has been focused on combatting network-related issues: congestions, packet losses, CND cache misses, etc. [1-3,6]. Various improvements in player algorithms have also been proposed [7-9]. The fact that content can be different, requiring different encoding has also been exploited, producing so-called "per-title" [10], "content-aware" [11], and "context-aware" techniques [12-14]. Encoding optimizations based on playback statistics have also been proposed [15].

However, most of such results have been obtained under assumption that streaming clients are adapting only to network bandwidth. Adaptation to player sizes, as we have just observed, brings a new dimension, and it notably complicates the problem.

This paper proposes a set of tools that can be used to analyze such more complex systems. In Section 2, we introduce our main models. In Section 3, we derive expressions for average performance characteristics of the system. In Section 4, we study their limits. In Section 5, we discuss related optimizations problems. In Section 6, we present simulation results, and in Section 7, we draw conclusions.

2. Main models

2.1. Codec and content

Consider a set of points $(H_i, R_i, D_i), i = 1, \dots, m$, where H_i denote video resolution (frame height), R_i denote bitrate, and D_i denote distortion (measured e.g. in PSNR or SSIM [16]) corresponding to the outcomes of m "probe" encodes generated by a given encoder for a given video content. Assume also that this set covers practically relevant operating range:

$(H_i, R_i) \in [H_{\min}, H_{\max}] \times [R_{\min}, R_{\max}]$, and that there exists a *model function* $D(H, R)$ approximately matching distortion values in all these points

$$D(H_i, R_i) \approx D_i, \qquad i=1,\ldots,m. \tag{1}$$

By analogy with the concept of *distortion-rate function* in information theory [17], we will call this function $D(H, R)$ an *empirical distortion-rate function*.

For example, for H.264 [18] codec and SSIM metric, we may use the following model:

$$D(H, R) = \left(1 + \left(\frac{R}{\alpha H^{\beta}}\right)^{-\gamma}\right)^{-\frac{1}{\gamma}}, \tag{2}$$

where α, β, γ are model parameters. In Figure 2, we show the results of fitting of this model to numbers obtained for 3 video sequences, denoted as "Easy", "Medium", and "Complex".

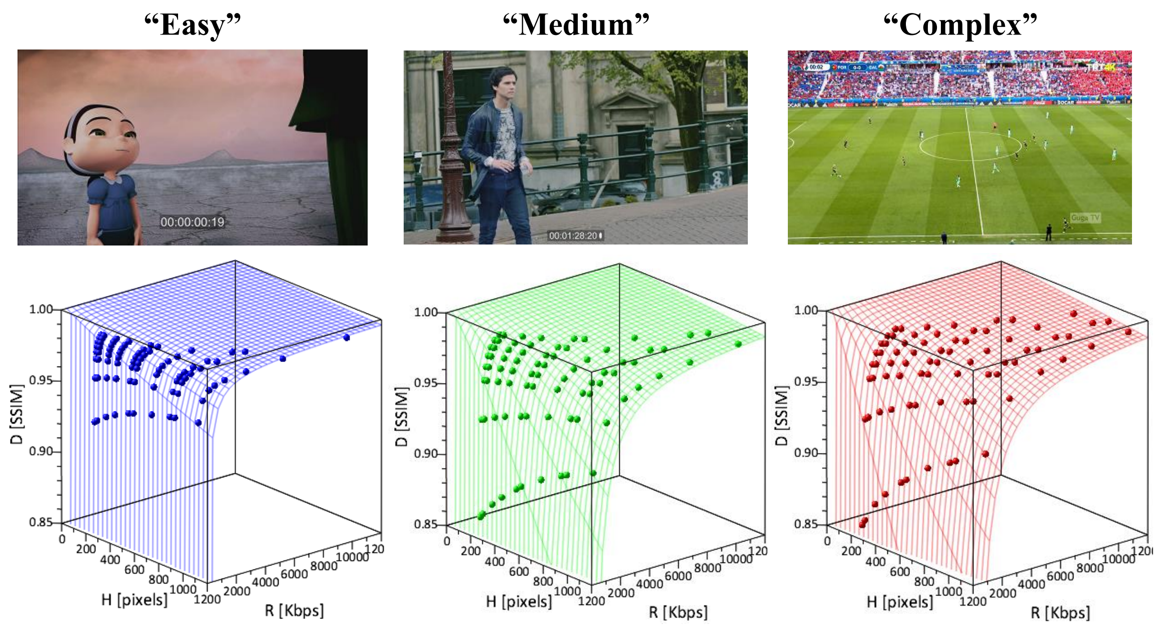

Figure 2: Probe points and empirical distortion-rate models for 3 video sequences.

In this experiment, we used x264 encoder [19], operating in 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 α, β, γ model parameters, as well as RMSE accuracy numbers are shown in Table 2.

Table 2. Parameters of empirical distortion-rate functions for 3 video sequences.

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

2.2. Encoding ladder

By *encoding profile* or *ladder* we will understand is a set of resolutions and bitrates

$$(H_i, R_i), i = 1, \ldots, n \tag{3}$$

at which content is encoded for streaming. Parameter n denotes the number of renditions in the ladder. We will say that the ladder is *proper*, if bitrates are strictly increasing $0 < R_1 < \cdots < R_n$, and resolutions are non-decreasing $0 < H_1 \leq \cdots \leq H_n$ for all points in the ladder. We also assume that aspect ratios W_i/H_i of all renditions in the ladder are the same.

2.3. Streaming client model

We will next assume that adaptation logic of a client with respect to network bandwidth B, and player size H_p can be modeled by a function:

$$i(B, H_p), \tag{4}$$

returning an index of a rendition selected. This model ignores temporal aspect (buffer state, etc.), as our objective is to capture effects in statistics, not dynamics of a particular session.

For example, to model client behavior shown in Figure 1, we may use the following model:

$$i(B, H_p) = \min\{i_R(B),\, i_H(H_p)\}, \tag{5}$$

where

$$i_B(B) = \begin{cases} 1 & if \quad B < T_1^B \\ i & if \quad T_i^B \leq B < T_{i+1}^B, \quad i = 2,..,n-2, \\ n & if \quad B \geq T_{n-1}^B \end{cases} \tag{6}$$

describes selection of rendition based on network bandwidth, and

$$i_H(H_p) = \begin{cases} 1 & if \quad H_p < T_1^H \\ i & if \quad T_i^H \leq H_p < T_{i+1}^H, \quad i = 2,..,n-2, \\ n & if \quad H_p \geq T_{n-1}^H \end{cases} \tag{7}$$

describes selection of rendition based on player size.

These chains rely on following thresholds along bitrates and resolutions, respectively:

$$T_i^B = (1+\delta)R_{i+1}, \quad i = 1..n-1,$$
$$T_i^H = \alpha H_i + (1-\alpha)H_{i+1}, i = 1..n-1$$

where $\delta \geq 0$ is a constant describing how closely client is trying to use available network bandwidth, and $\alpha \in [0,1]$ is a constant describing client's preference towards downscaling vs. upscaling. Good fit for statistics shown in Figure 1 is obtained by using $\delta \approx 0.35$ and $\alpha \approx 0.75$. We show plots of all these functions in Figure 3.

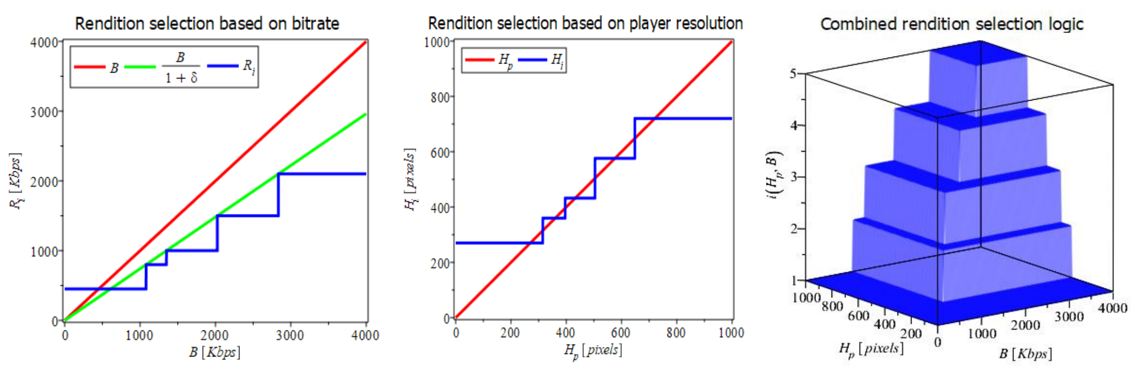

Figure 3. Example client model. Left: selection based on network bandwidth (6). Middle: selection based on player resolution (7). Right: combined selection logic (5).

2.4. Reproduction quality model

Next, we will need a model

$$Q(H, H_p, D), \tag{10}$$

translating video resolution H, codec-introduced distortion D, and player size H_p into a figure that has a good correlation with subjective quality scores reported by viewers.

For this purpose, we may employ the following model:

$$Q(H, H_p, D) = \alpha\left(\beta + Q_{\phi,u}\left(\phi(H_p),\ u(H, H_p)\right)\right) Q_D(D), \tag{11}$$

where:

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

is the *viewing angle* to video projected on the screen, $DAR = W_p/H_p$, d is a *viewing distance*, and ρ is a *display pixel density*,

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

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

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

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

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

is a Westerink-Roufs model [20,21], predicting MOS scores based on ϕ and u to, and

$$Q_D(D) = \mathrm{e}^{\gamma D},$$

is a function, translating the distortion D, measured in SSIM [16], to MOS domain [22].

The combination of impacts of resolution/projection- and distortion- based quality factors in this model is multiplicative, as earlier suggested in [23].

Parameters α, β, and γ in this model are calibration constants used to fit it to MOS scores in reference data set. For example, by fitting this model to Netflix data set [24,25] 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 1-5 MOS scale, which is quite reasonable and compares well to other metrics tested with same dataset [24].

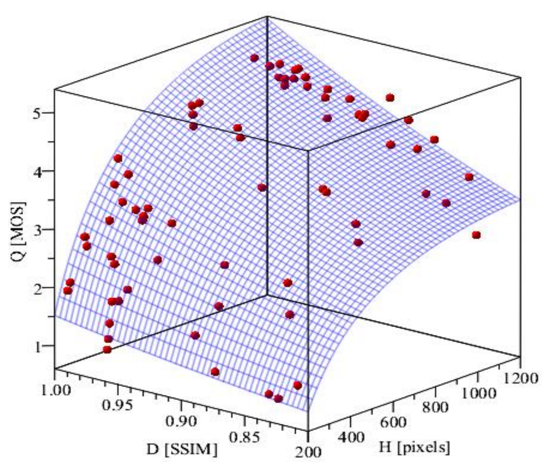

Figure 4. Fit of the proposed quality model to MOS scores in Netflix dataset.

For streaming applications delivering videos to PCs, we may assume that ρ=96 [dpi] and d=24 [in], as typical for PC monitors and viewing practices.

2.5. Quality-rate model

Finally, given empirical distortion-rate function (2), and reproduction quality model (11), we can define a model describing the space of achievable quality-rate tradeoffs when working with a particular encoder, content, and a player:

$$Q(H, H_p, R) = Q(H, H_p, D(H, R)). \quad (12)$$

We will call this function quality-rate model. We show plots of this model for our "Complex" sequence and player sizes $H_p \in \{270, 540, 1080\}$ in Figure 5.

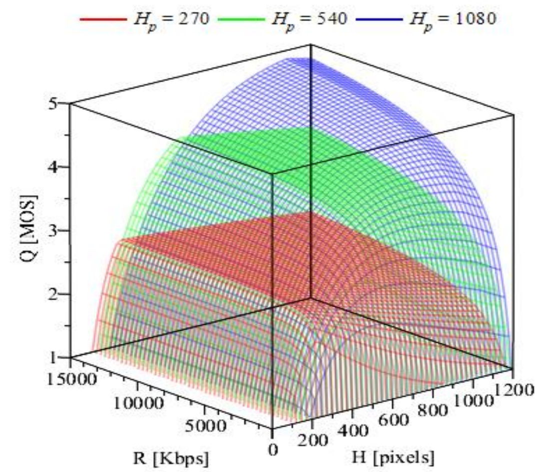

Figure 5. 3D projections of $Q(H, H_p, R)$ model constructed for "Complex" sequence.

Quality-rate model will play a key role in our subsequent analysis of average performance of the system and also in setting of the related optimization problems.

3. Average performance of adaptive streaming

We will next characterize average performance of the system. We will need 2 distributions:

- network bandwidth distribution: $p(B)$, where bandwidth 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: $H_p \in \mathcal{H}_p$.

Given all these ingredients, we now can compute average quality delivered by the system:

$$\bar{Q}(H_1, \dots, H_n, R_1, \dots, R_n, p, q) = \int_0^\infty p(B) \sum_{H_p \in \mathcal{H}_p} q(H_p)\, Q\left(H_{i(B,H_p)}, H_p, R_{i(B,H_p)}\right) dB. \quad (13)$$

Here, the averaging across bandwidth B is done by first integral, the averaging across player resolutions H_p is done by the following sum. The reproduction quality $Q(H_i, H_p, R_i)$ for each pair (B, H_p) is computed by pulling index of the selected rendition $i = i(B, H_p)$, and then retrieving its bitrate R_i and resolution H_i parameters.

The derived average quality expression (13) holds a for specific encoder and content, as described by the quality-rate model $Q(H, H_p, R)$, and an n-point encoding ladder with resolutions $H_1, \ldots, H_n$ and bitrates $R_1, \ldots, R_n$.

In similar fashion, we can also express:

- average resolution of video, as delivered:

$$\bar{H}(H_1, \ldots, H_n, R_1, \ldots, R_n, p, q) = \int_0^\infty p(B) \sum_{H_p \in \mathcal{H}_p} q(H_p)\ H_{i(B,H_p)} dB, \tag{14}$$

- average distortion (codec noise) in video, as delivered:

$$\bar{D}(H_1, \ldots, H_n, R_1, \ldots, R_n, p, q) = \int_0^\infty p(B) \sum_{H_p \in \mathcal{H}_p} q(H_p)\, D\left(H_{i(B,H_p)}, R_{i(B,H_p)}\right) dB, \tag{15}$$

- average player *size*:

$$\bar{H}_p(q) = \sum_{H_p \in \mathcal{H}_p} q(H_p)\, H_p, \tag{16}$$

- average network bandwidth used by streaming system:

$$\bar{R}(H_1, \ldots, H_n, R_1, \ldots, R_n, p, q) = \int_0^\infty p(B) \sum_{H_p \in \mathcal{H}_p} q(H_p)\ R_{i(B,H_p)} dB, \tag{17}$$

- average network bandwidth:

$$\bar{B}(p) = \int_0^\infty p(B)\, B\ dB, \tag{18}$$

and so on.

4. Performance limits

If we next consider an extreme case: a ladder with infinite number of points capturing all possible values of bitrates and resolutions in encoded content, then we would have

$$R_{i(B,H_p)} \to \frac{B}{1+\delta}, \qquad H_{i(B,H_p)} \to H_p,$$

and so, the average quality delivered by such system becomes:

$$\bar{Q}_\infty(p, q) = \int_0^\infty p(B) \sum_{H_p \in \mathcal{H}_p} q(H_p)\, Q\left(H_p, H_p, \frac{B}{1+\delta}\right) dB. \tag{19}$$

By same argument

$$\bar{R}_\infty(p) = \frac{1}{1+\delta} \bar{B}(p). \tag{20}$$

These are the ultimate performance limits for a streaming system operating with a given encoder, content, network, and distribution of player sizes.

5. Related optimization problems

Given that we now know how to compute average performance parameters of the streaming system, we may next pose few related optimization problems.

For example, we can pose a problem of finding an encoding ladder maximizing quality:

$$\overline{Q}\left(\widehat{H}_1, \dots, \widehat{H}_n,\ \widehat{R}_1, \dots, \widehat{R}_n, p, q\right) = \max_{\substack{R_{\min} \le R_1 < \cdots < R_n \le R_{\max} \\ H_{\min} \le H_1 \le \cdots \le H_n \le H_{\max} \\ R_1 \le R_{1,\max},\ H_1 \le H_{1,\max}}} \overline{Q}(H_1, \dots, H_n,\ R_1, \dots, R_n, p, q)\,. \tag{21}$$

The added constraints in this problem include limits on ladder resolutions and bitrates. The limits on first rendition bitrate and resolutions are practically needed to establish the lowest quality operating point. This is a generalization of a problem considered in [12].

Same problem can be further constrained by limit on average bitrate:

$$\overline{Q}\left(\widehat{H}_1, \dots, \widehat{H}_n,\ \widehat{R}_1, \dots, \widehat{R}_n, p, q\right) = \max_{\substack{R_{\min} \le R_1 < \cdots < R_n \le R_{\max} \\ H_{\min} \le H_1 \le \cdots \le H_n \le H_{\max} \\ R_1 \le R_{1,\max},\ H_1 \le H_{1,\max} \\ \overline{R}(H_1,\dots,H_n, R_1,\dots,R_n) \le \overline{R}_{\max}}} \overline{Q}(H_1, \dots, H_n,\ R_1, \dots, R_n, p, q) \tag{22}$$

or, alternatively, we can also try to find a ladder minimizing average bitrate while delivering a given limit for average quality:

$$\overline{R}\left(\widehat{H}_1, \dots, \widehat{H}_n,\ \widehat{R}_1, \dots, \widehat{R}_n, p, q\right) = \min_{\substack{R_{\min} \le R_1 < \cdots < R_n \le R_{\max} \\ H_{\min} \le H_1 \le \cdots \le H_n \le H_{\max} \\ R_1 \le R_{1,\max},\ H_1 \le H_{1,\max} \\ \overline{Q}(H_1,\dots,H_n, R_1,\dots,R_n) \ge \overline{Q}_{\min}}} \overline{R}(H_1, \dots, H_n,\ R_1, \dots, R_n, p, q)\,. \tag{23}$$

The last is a generalization of a problem considered in [15].

Many other optimization problems may be posed by considering not only choices of ladder parameters, but also parameters of client adaptation logic, introducing more complex (e.g. 2-state) network models, etc.

6. Simulation results

In this section, we will consider several example systems, operating with different content, networks, and player models, and compute their respective performance parameters.

As content, we will use 3 video sequences, already considered in Section 2.1, and with distortion-rate model parameters summarized in Table 2.

As network models, we will use mixtures or 2 Rayleigh distributions:

$$p(B) = \alpha\, f(B, \sigma_1) + (1-\alpha)\, f(B, \sigma_2), \tag{24}$$

where $f(x,\sigma) = \frac{x}{\sigma^2} \exp\left(-\frac{x^2}{2\sigma^2}\right)$, and $\alpha, \sigma_1, \sigma_2$ are model parameters. Specific values of these parameters, as well as resulting average bandwidth values are shown in Table 3.

Table 3. Network models.

Network	$\boldsymbol{\alpha}$	$\boldsymbol{\sigma_1}$	$\boldsymbol{\sigma_2}$	$\overline{\boldsymbol{B}}$
"Network 1"	0.4287	1802.2	4499.28	4189.87
"Network 2"	0.4287	4,505.5	11,248.2	10474.7

As players, we will consider 2 player models with different possible sets of player sizes $\mathcal{H}_p$ and distributions $q(H_p)$, as listed in Table 4. Specifically, "1080p player" is a model of a player that always stretches video to a 1080p window, and "Web player" is a model with 11 possible window size matching statistics shown in Figure 1.

Table 4. Player models.

Player	**Player sizes $\mathcal{H}_p$**	**Player size probabilities $q(H_p)$**	$\bar{H}_p$
"1080p player"	{1080}	{1.0}	1080
"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}	538.1

For all networks, players, and content we will consider the use of two types of encoding profiles: 1) reference encoding ladder shown in Table 1, and 2) respective quality-optimal encoding ladders, shown in Table 5. These ladders have been generated by solving optimization problem (21) with following constraints applied: $R_{1.\max}$ = 180, $H_{1.\max}$ = 480, $R_{\max}$ = 5050, and $\mathcal{H}$ = {216,270,288,360,432,480,540,576,720,900,1080}.

Table 5. 5-point quality-optimal ladders generated for all cases.

Network	**Player**	**Content**	**Rendition 1**	**Rendition 2**	**Rendition 3**	**Rendition 4**	**Rendition 5**
Network 1	1080p player	Easy	854x480 167k	1024x576 173k	1280x720 277k	1600x900 607k	1920x1080 1557k
		Medium	854x480 180k	1024x576 410k	1280x720 769k	1600x900 1384k	1920x1080 2804k
		Complex	854x480 180k	1024x576 480k	1280x720 899k	1600x900 1619k	1920x1080 3155k
	Web Player	Easy	512x288 180k	768x432 365k	854x480 935k	1280x720 973k	1600x900 1557k
		Medium	480x270 180k	768x432 632k	854x480 1497k	1280x720 1619k	1600x900 2697k
		Complex	480x270 180k	768x432 739k	854x480 1684k	1280x720 1970k	1600x900 3155k
Network 2	1080p player	Easy	853x480 180k	1024x576 203k	1280x720 410k	1600x900 1052k	1920x1080 3033k
		Medium	853x480 180k	1024x576 540k	1280x720 1138k	1600x900 2305k	1920x1080 5050k
		Complex	853x480 180k	1024x576 607k	1280x720 1280k	1600x900 2493k	1920x1080 5050k
	Web Player	Easy	480x270 180k	768x432 657k	854x480 1895k	1280x720 1970k	1920x1080 2697k
		Medium	480x270 180k	768x432 1052k	854x480 2804k	1280x720 2917k	1600x900 4856k
		Complex	384x216 180k	768x432 1183k	854x480 3155k	1280x720 3281k	1600x900 5050k

The resulting average system performance parameters computed for in all cases are shown in Table 6. Average quality $\bar{Q}$ is reported in MOS, average bitrate $\bar{R}$ in Kbps, average height $\bar{H}$ in pixels, and distortion $\bar{D}$ in SSIM.

Presented results exhibit many effects that are well known in practice. Thus, with fixed ladders, we see that complex content may be delivered at low quality, for easy content bits can be wasted, better networks can deliver better quality, and full-screen players pull more bits and deliver better quality as compared to web players.

Based on Table 6, we also see that by applying optimizations, performance of the system can be significantly improved. With 1080p players we see that in some cases, quality gains can be 0.8MOS and beyond. With Web players quality gains are also very significant.

We also notice, that optimal ladders generated for different content, player models, and networks look very different. Specifically, we see that optimal ladders generated for web players have very different resolution allocations as opposed to profiles generated to players always stretching videos full screen. This proves that web-streaming needs different profile optimization techniques, and that more generally, accounting for adaptation for player resolutions is important.

Table 6. Average streaming performance for different networks, players, and content.

Network	**Player**	**Content**	**Performance, reference ladder:**				**Performance, optimized ladders**			
			$\bar{H}$	$\bar{D}$	$\bar{Q}$	$\bar{R}$	$\bar{H}$	$\bar{D}$	$\bar{Q}$	$\bar{R}$
Network 1	1080p Player	Easy	647.8	0.9910	4.075	1846.5	1043.7	0.9819	4.955	1388.4
		Medium	647.8	0.9718	3.891	1846.5	975.4	0.9534	4.512	2130.5
		Complex	647.8	0.9585	3.774	1846.5	954.4	0.9392	4.337	2288.0
	Web Player	Easy	465.2	0.9903	3.563	1151.8	537.7	0.9850	3.719	1094.5
		Medium	465.2	0.9701	3.395	1151.8	515.9	0.9617	3.473	1262.3
		Complex	465.2	0.9522	3.258	1151.8	506.1	0.9420	3.316	1407.7
Network 2	1080p Player	Easy	706.3	0.9912	4.343	2076.5	1057.8	0.9892	5.000	2811.0
		Medium	706.3	0.9724	4.150	2076.5	1019.8	0.9747	4.830	4264.4
		Complex	706.3	0.9606	4.034	2076.5	1017.3	0.9670	4.741	4292.1
	Web Player	Easy	486.5	0.9904	3.653	1232.6	545.9	0.9898	3.781	1608.6
		Medium	486.5	0.9704	3.482	1232.6	530.0	0.9741	3.630	2421.0
		Complex	486.5	0.9532	3.347	1232.6	519.1	0.9638	3.531	2635.8

7. Conclusions

The analysis of average performance of streaming systems adapting to both network bandwidth and player window sizes is presented. It is shown that the problem can be formalized by introduction of several models, notably quality-rate and client adaptation models, leading to formulae for average system performance under given statistical models of networks and players. Limits in infinite ladder case are established. Related optimization problems have also been posed. Provided simulation results confirm many effects known from practice, and show that performance of such systems can be significantly improved by applying profile optimizations, accounting to specifics of networks, players, and content.

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] 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.

[3] 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.

[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] D. Lee, C. Dovrolis, A. Begen, “Caching in HTTP Adaptive Streaming: Friend or Foe?,” in Proc. ACM Network and Operating System Support on Digital Audio and Video Workshop, Singapore, March 2014, pp. 31–36.

[7] S. Hesse, “Design of scheduling and rate-adaptation algorithms for adaptive HTTP streaming,” in Proc. SPIE 8856, Applications of Digital Image Processing XXXVI, 88560M, 2013.

[8] C. Zhou, X. Zhang, L. Huo and Z. Guo, "A control-theoretic approach to rate adaptation for dynamic HTTP streaming," in Proc. Visual Comm. Image Processing, San Diego, CA, 2012, pp. 1-6.

[9] K. Spiteri, R. Urgaonkar, R. K. Sitaraman, BOLA: Near-Optimal Bitrate Adaptation for Online Videos. IEEE/ACM Trans. Netw. 28(4): 1698-1711 (2020)

[10] 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.

[11] 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.

[12] Y. 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.

[13] Y. 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.

[14] Y. Reznik, X. Li, K.O. 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.

[15] 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.

[16] 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.

[17] T. Cover, J. Thomas, “Elements of Information Theory”, 2nd Ed., Wiley, Hoboken, NJ, 2006.

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

[19] “x264 encoder project,” https://www.videolan.org/developers/x264.html.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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

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