Towards Understanding of the Behavior of Web Streaming

Yuriy A. Reznik
Brightcove, Inc.
Seattle, USA
yrenik@brightcove.com

Karl O. Lillevold
Brightcove, Inc
Seattle, USA
klillevold@brightcove.com

Abhijith Jagannath
Brightcove, Inc
Seattle, USA
ajagannath@brightcove.com

Xiangbo Li
Brightcove, Inc
Scottsdale, USA
xli@brightcove.com

***Abstract*—We study the behavior of a modern-era adaptive streaming system delivering videos embedded in web-pages. In such an application, the size of videos rendered on the screen may depend on user preferences, such as the position and size of a browser window. Moreover, the stream selection logic in such a system appears to be influenced not only by the available network bandwidth but also by the output video size, which, in many cases, limits the selection of higher quality streams. To explain this behavior, in this paper we introduce a simple analytical model of a client adapting to both bandwidth and player size. Using this model, we then compute stream selection probabilities and show that they are sufficiently close to respective statistics observed in practical experiments. Possible uses of this proposed client model are also suggested. Specifically, we show how it can be used to derive formulae for the average performance parameters of the system and also for posing related optimization problems.**

***Keywords*—ABR streaming, encoding ladder, stochastic models, average-case analysis, non-linear constrained optimization**

I. Introduction

A. The behavior of adaptive streaming

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 address [1-3]. An early example of a satisfactory solution was the so-called “SureStream” technology, introduced by RealNetworks in 1998 [3]. The main idea was to encode media at multiple bitrates and design a system switching between such streams adaptively, as needed to match network bandwidth, observable at each point in time. The same basic concept is well known today as *Adaptive Bitrate Streaming*, providing the basis for modern streaming protocols and standards such as HLS [4] and DASH [5]. In all such recent systems, the decisions about stream selections are usually done by *streaming clients*, running on user devices.

Moreover, in the modern days of streaming, and specifically in streaming to web browsers, there is another important parameter affecting the playback. It is the *size of a video player window* or *video display area* on a webpage. What causes its variation are user preferences, influencing the position and size of the browser window on the screen, as well as form factor/display size of the user device.

To illustrate the significance of this parameter, in Fig. 1, we show playback statistics captured during a large scale web streaming event. In Table 1, we also list the parameters of the

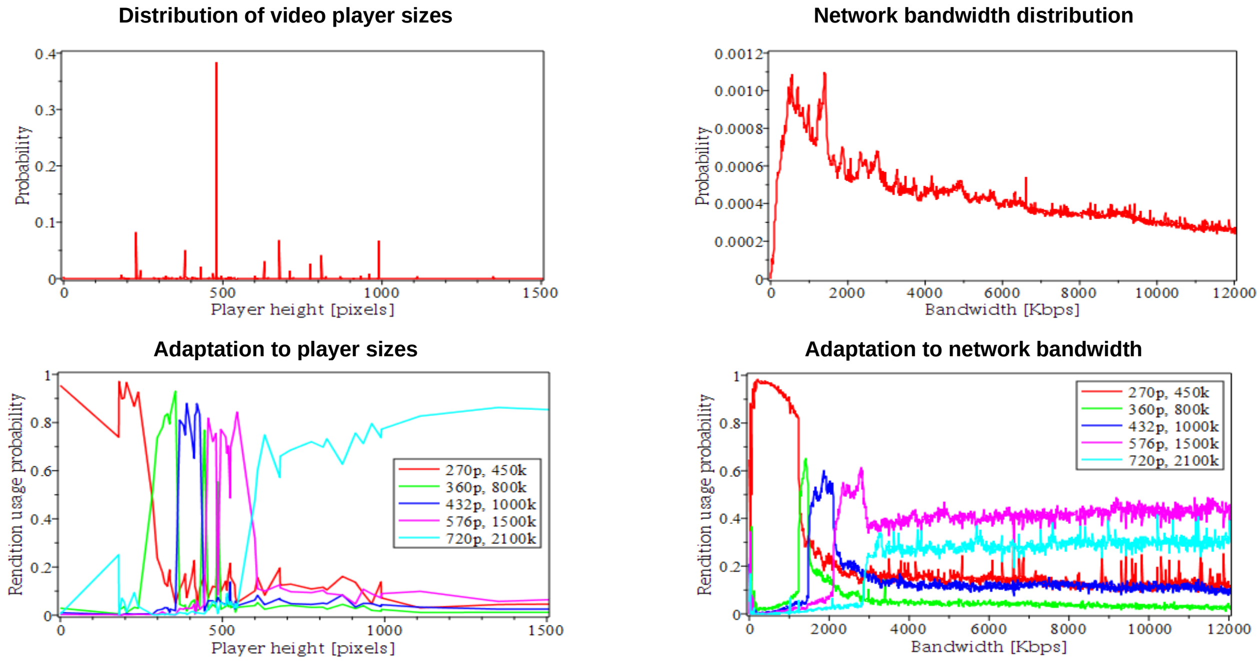

Fig. 1. Playback statistics captured during a large scale live streaming event.

encoded streams used for this event. The playback statistics data used in this example are provided in [8].

TABLE I. ABR LADDER USED FOR STREAMING OF THE 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 Fig. 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 about 300 lines or less. Similarly, 720p rendition was loaded most frequently when player sizes were about 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 significantly affect the choices of streams used by the system. In this paper, we will try to explain the above-observed effects. We will approach this problem mathematically, by first offering a simple deterministic player model, and then by using probabilistic techniques to derive conditional probabilities of rendition loads under certain assumptions about statistical behavior of networks bandwidth and player sizes. We will then study differences between the predicted and the observed behavior. Additional applications of the proposed model will also be discussed.

B. *Prior and related work*

Most early studies on adaptive streaming systems have focused on network-related issues: congestions, packet losses, CND cache misses, etc. [1-3,9]. Rate adaptation algorithms have also been in focus [10-12]. The fact that content can be different has also been exploited, producing so-called “per-title” [13], “content-aware” [14], and “context-aware” techniques [15-18]. Earlier uses of probabilistic techniques and idealized client models can be found in [15,16,18]. However, most of these results have been obtained under the assumption that streaming clients are adapting only to network bandwidth. Adaptation to player sizes, as we have just observed, makes system behavior much more complex. This paper focuses on the derivation of an adequate client model. Applications of this proposed model for the analysis and optimizations of such systems will be discussed in detail in forthcoming papers [19,20].

C. *Outline*

This paper is organized as follows. In Section II we offer definitions of the most involved variables and stochastic models. In Section III, we introduce our model of the client. In Section IV we will use this model to compute rendition load probabilities and compare them to the experimental results. In Section V we will discuss applications of this proposed model. In Section 6, we will offer conclusions.

II. DEFINITIONS

A. *Encoding ladders*

By an *encoding profile* or a *ladder*, we will understand is a set of video resolutions and bitrates at which a given video asset is encoded for streaming:

$$(W_i \times H_i, R_i), \;\; i = 1, \ldots, n. \qquad (1)$$

Here W_i denotes video width [in pixels], H_i denotes video height [in pixels], and R_i denotes bitrate [in Kbps] of each rendition. Parameter n denotes the number of renditions in the ladder. Parameter i denotes rendition index.

For simplicity, we will also assume that the aspect ratios W_i/H_i of all renditions in a ladder are the same. Therefore, specification of a single resolution parameter, e.g., height H_i is sufficient.

We will further 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 renditions in the ladder. Table 1 represents an example of a proper ladder.

B. *Player sizes*

By $W_p \times H_p$ we will denote the width and the height [in pixels] of a player. For simplicity, we will also assume that player size has the same aspect ratio as video, and therefore specification of only player height H_p is sufficient.

When working with a *population* of streaming clients and viewers of the content, we will be dealing with a *set* $\mathcal{H}_p$ of possible player sizes $H_p \in \mathcal{H}_p$ that may be selected by different viewers. Furthermore, to model that such different player sizes may be selected with different probabilities, we will assume that H_p is a *discrete random variable* with a certain probability mass function $q(H_p)$, defined over $\mathcal{H}_p$.

An example of the distribution of player sizes observed in practice is provided in the top left subfigure in Fig. 1.

C. *Network bandwidth*

By B we will denote network bandwidth value [in Kbps] that may be observed (or measured) by a streaming client.

To enable the probabilistic study of the streaming system, we will further assume that B is a *continuous random variable* with a certain given probability density function $p(B)$ and support $[0, \infty)$.

An example of the distribution of network bandwidth values as measured by a population of streaming clients in practice is provided in the top right subfigure in Fig. 1.

III. THE PROPOSED CLIENT MODEL

As we have already seen in Fig. 1, it appears that practical web streaming clients make decisions about which streams to use based at least on two parameters: available network

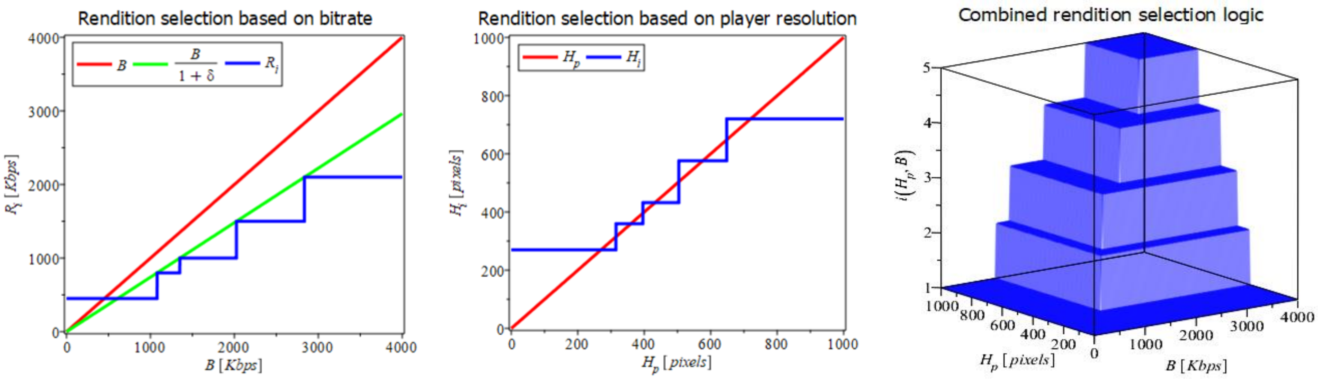

Fig. 2. Construction of streaming client model. Left: rendition selection based on the available network bandwidth B. Middle: rendition selection based on player window size H_p. Right: the combined rendition selection logic.

bandwidth B, and player window size H_p. To capture this behavior, we will first propose adaptation models for each of these variables separately and then offer a combined model.

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

$$i_B(B) = \begin{cases} 1 & if \quad B < T_1^B \\ i & if \quad T_i^B \le B < T_{i+1}^B, \quad i = 2,..,n-2, \\ n & if \quad B \ge T_{n+1}^B \end{cases} \quad (3)$$

$$T_i^B = (1+\delta) R_{i+1}, \quad i = 1..n-1, \quad (4)$$

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 $\delta \ge 0$ is a "bandwidth overhead" constant, used to characterize the extent to which a client is trying to utilize all available bandwidth.

We show the plot of this model function in the left subfigure of Fig. 2. The plot is rendered for $\delta = 0.35$.

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

$$i_H(H_p) = \begin{cases} 1 & if \quad H_p < T_1^H \\ i & if \quad T_i^H \le H_p < T_{i+1}^H, \quad i = 2,..,n-2, \\ n & if \quad H_p \ge T_{n-1}^H \end{cases} \quad (5)$$

$$T_i^H = \alpha H_i + (1-\alpha) H_{i+1}, i = 1..n-1 \quad (6)$$

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 the client's preference towards downscaling vs. upscaling.

We show the plot of this model function in the middle subfigure of Fig. 2. The plot is rendered for $\alpha = 0.5$.

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

$$i(B, H_p) = \min \{ i_R(B), i_H(H_p) \}. \quad (7)$$

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 Fig. 2.

In passing, we must note that the proposed client model (7) is indeed extremely simple. It is simply a function of two parameters: bandwidth B and player size H_p. This model has no state and no dependencies on buffer size, current buffer fullness, and various additional parameters that all normal implementations of streaming clients would have. However, as we will show in the next section, even such a basic and simple model is sufficient to predict several key effects that we have observed in the real-world playback statistics with web players.

IV. Model Fitting and Accuracy Analysis

A. Rendition load probabilities

Given out introduced client model $i(B, H_p)$, as well as assumed probabilistic behavior of player sizes H_p and network bandwidth B, we can next write formulae for conditional probabilities of loading of k-th rendition:

$$P(k \mid H_p) = \int_{B:\, i(B,H_p)=k} p(B)\, dB, \qquad k = 1, \dots, n \quad (8)$$

$$P(k \mid B) = \sum_{H_p:\, i(B,H_p)=k} q(H_p), \qquad k = 1, \dots, n \quad (9)$$

In these formulae, the ranges of integration or summation are the regions of values of B or H_p such that the index of rendition that becomes selected is k.

The derived expressions conditional probabilities (8,9) define possible analytical models for the same quantities that we empirically measured and shown earlier in lower sub-figures of Fig. 1.

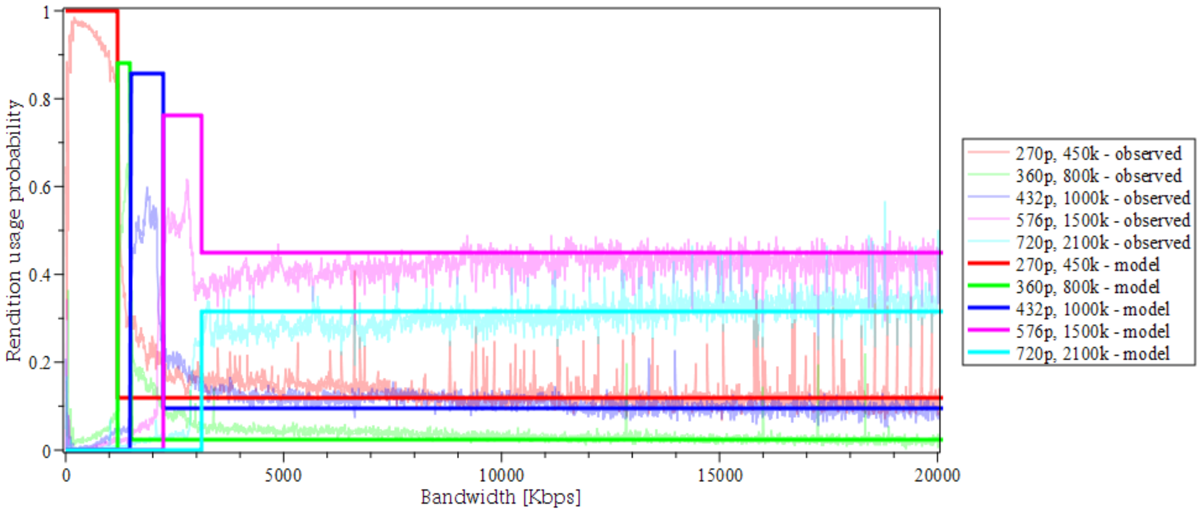

Fig. 3. Observed vs model-predicted conditional probabilities of loading of each rendition.

B. *Streaming data set. Empirical load probabilities.*

In [8], we provide a set of playback statistics collected during a large-scale streaming event. For streaming of this event, Brightcove VideoCloud system [6] was used, employing players built using the open-source *video.js* player framework [7]. VideoCloud Analytics system [6] was used to capture the playback statistics.

Specific data provided in the repository [8] include:

- encoding ladder information (cf. Table 1),
- raw player logs, reporting payer parameters reported at each 10sec events during playback,
- as well as derived empirical distributions (histograms):
  - ○ $\hat{p}(B)$ – network bandwidth histogram,
  - ○ $\hat{q}(H_p)$ – histogram of player sizes,
  - ○ $\hat{P}(k \mid B)$ – conditional load probabilities of each rendition w.r.t. network bandwidth, and
  - ○ $\hat{P}(k \mid H_p)$ – conditional load probabilities of each rendition w.r.t. player sizes.

Overall, the dataset [8] provides records of over 200M of client events, which we feel is sufficient for reliable estimation of rendition load probabilities and other performance statistics.

C. *Fitting the player model to empirical statistics*

Recall, that our proposed player model (7) uses 2 tuning parameters: α and δ. In order to find them, we will need to minimize the differences between the model probabilities $P(k \mid B) = P(k \mid B, \alpha, \delta)$ and corresponding empirical values $\hat{P}(k \mid B)$ provided in the dataset [8].

We use the following objective function:

$$\Phi(\alpha,\delta) = \sum_{B} \hat{p}(B) \sum_{k=1}^{n} \left| P(k \mid B, \alpha, \delta) - \hat{P}(k \mid B) \right| \quad (10)$$

where the summation is done over distinct bandwidth values B as reported in the dataset [8], and where $\hat{p}(B)$ represents empirically measured probability of occurrence of bandwidth value B. This objective function (10) can be understood as the average variational distance between empirical and model-derived distributions. We note that in the computation of model-based probabilities (9) we used the support set $\widehat{\mathcal{H}}_p$ and empirical player size distribution $\hat{q}(H_p)$ as provided by the same dataset [8].

In order to solve the optimization problem:

$$\Phi(\alpha^*, \delta^*) = \min_{\alpha,\delta} \Phi(\alpha, \delta). \quad (11)$$

we used brute force enumeration, which with a precision of 1e-3 has yielded $\alpha^* = 0.723$, and $\delta^* = 0.45$.

D. *Model accuracy*

In Fig 3. we present the superimposed plots of real-word and model-predicted conditional probabilities $P(k \mid B)$ computed by using our model with optimal choice of model parameters α and δ. The fit accuracy is further investigated and reported by using 4 different distance metrics [19] in Table 2. Such metrics were computed for each bandwidth value B, and subsequently averaged across the range of bandwidth values and by using density $\hat{p}(B)$.

TABLE II. MODEL FIT ACCURACY.

Metric	**Accuracy**
Average L1 norm (variational distance)	0.176283
Average L2 norm	0.101642
Average Information Divergence $D(\hat{P}\|\|P)$	0.205428
Average Kolmogorov-Smirnov test	0.073857

Based on plots in Fig. 3, it can be observed that the proposed client model, despite its extreme simplicity, predicts several key phenomena reasonably well. Thus, it can be observed, that the rendition switch positions along bandwidth B are well aligned with the corresponding changes in the client-reported statistics. We also notice, that in the high-bandwidth regime this model captures very well the behavior of the real-world system. In other words, this model explains sufficiently well why web-clients may not use high bitrate renditions even in the excess of bandwidth available.

By studying plots in Fig. 3, we also notice several additional effects that our simple client model does not capture. For example, we notice that real-world rendition selection windows not only do not have sharp vertical boundaries, but they are not even symmetric! The switch-down transition in player statistics seems to be more spread. Indeed, most of such extra detail can easily be captured by making the client model non-deterministic, i.e., treating it as a random process, with a particular density over (B, H_p) space. However, moving along this path is unlikely to achieve much from a methodological standpoint. The simple model that we have now is both intuitive and performs reasonably well to support some useful applications.

V. Applications

A. Average performance analysis of streaming systems

The proposed client model can be easily used to derive the precise expressions for many average performance parameters of streaming systems. For example, the *average bitrate* consumed by the streaming system can be expressed as follows:

$$\bar{R} = \int_0^\infty p(B) \sum_{H_p \in \mathcal{H}_p} q(H_p)\ R_{i(B,H_p)} dB, \qquad (12)$$

where R_i denotes bitrate of rendition selected given each combination of bandwidth B and player size H_p and where integration and summation are done for both. Similarly, we can produce expression for the *average quality* achieved in the system:

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

where $Q(H_i, H_p, R_i)$ is a model of quality for encoded and delivered stream, given its encoded resolution H_i, bitrate R_i, and player size H_p. Again, our client model (9) is utilized here to obtain both selected stream resolution and bitrate. Additional details about suitable for this purpose quality metric and performance analysis of streaming systems can be found in [19].

B. Design of optimal ladders for web streaming

Given the above-derived expressions for the average quality delivered by the streaming system (13), the problem of finding parameters of an encoding ladder: $H_1, \dots, H_n$ and $R_1, \dots, R_n$ maximizing such average quality can be posed. Additional details about the setting of such an optimization problem and finding its solution can be found in [20].

VI. Conclusions

In this paper, we have studied the behavior of a streaming system delivering videos embedded in web pages. We have noticed that client adaptation logic in such systems is influenced not only by the available network bandwidth but also by the size of the player window (or viewport of a web page) on the screen. To study this phenomenon, we have introduced a simple analytical model of streaming clients using adaptation to both of these parameters, and then studied its accuracy relative to statistics of player behavior observed in practice. This study has shown that the proposed model approximates real player behavior reasonably well. Possible uses of this proposed client model for the analysis and optimization of streaming systems have also been discussed.

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. CSVT, 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. CSVT, 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, 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] Brightcove VideoCloud platform https://www.brightcove.com/en/online-video-platform

[7] Video.js open source project, https://github.com/videojs/video.js

[8] Brightcove data set. https://github.com/brightcove/streaming-dataset

[9] 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, 2014, pp. 31-36.

[10] 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.

[11] 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.

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

[13] Netflix, "Per-title encode optimization," https://medium.com/netflix-techblog/per-title-encode-optimization-7e99442b62a2, Dec. 15 2015.

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

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] Y. Reznik, "Average Performance of Adaptive Streaming," Proc. Data Compression Conference (DCC'21), Snowbird, UT, 23-26 March 2021.

[20] Y. Reznik, K. Lillevold, R. Vanam, "Perceptually optimized ABR ladder generation for Web streaming," *Proc. IS&T Electronic Imaging*, San Francisco, CA, January 18-21, 2021.

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