Towards Efficient Multi-Codec Streaming

Yuriy Reznik, Karl Lillevold, Abhijith Jagannath

Brightcove Inc, Boston, MA, USA.

Nabajeet Barman

Brightcove UK Ltd, London, UK.

Written for presentation at the

SMPTE 2022 Media and Technology Summitt

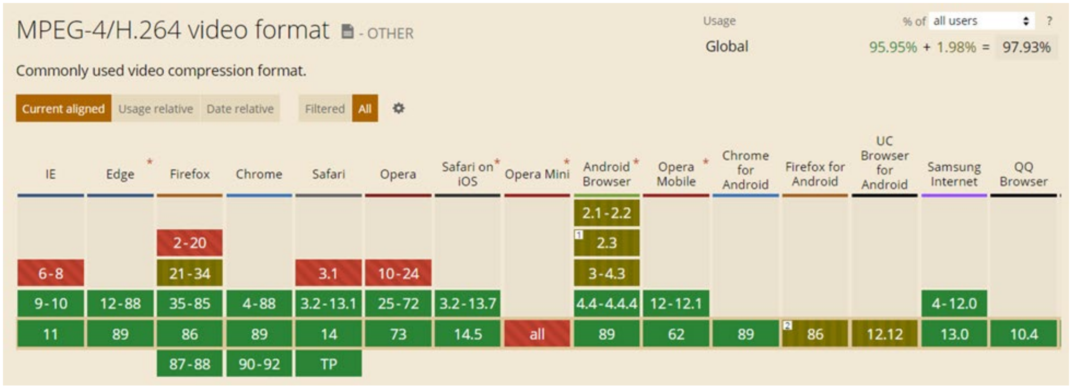
Abstract. *One of the biggest challenges in modern-era streaming is the fragmentation of codec support across receiving devices. For example, modern Apple devices can decode and seamlessly switch between H.264/AVC and HEVC streams. Most new TVs and set-top boxes can also decode HEVC, but they cannot switch between HEVC and H.264/AVC streams. And while most older devices/streaming clients can only receive and decode H.264/AVC streams. With the arrival of next-generation codecs - such as AV1 and VVC, the fragmentation of codec support across devices becomes even more complex. This situation brings a question – how can we serve such a population of devices most efficiently by using codecs delivering the best performance in all cases yet producing the minimum possible number of streams and such that the overall cost of media delivery is minimal? In this paper, we explain how this problem can be formalized and solved at the stage of dynamic generation of encoding profiles for ABR streaming. The proposed solution is a generalization of the context-aware encoding (CAE) class-of techniques, considering multiple sets of renditions generated using each codec and codec usage distributions by the population of the receiving devices. We also discuss several streaming system-level tools needed to make the proposed solution practically deployable.*

.

Keywords. Adaptive bitrate streaming, DASH, HLS, H.264/AVC, HEVC, multi-codec streaming, per-title encoding, context-aware encoding, CAE.

Introduction

During the last decade, most video streams sent over the Internet have been encoded using the ITU-T H.264 / MPEG-4 AVC video codec [1]. Developed in the early 2000s, H.264/AVC has become broadly supported on various devices and platforms. According to www.caniuse.com analytics website [2], the current reach of H.264 across web platforms is approaching an overwhelming 97.93%.



MPEG-4/H.264 video format - OTHER

Usage % of all users

Global 95.95% + 1.98% = 97.93%

Commonly used video compression format.

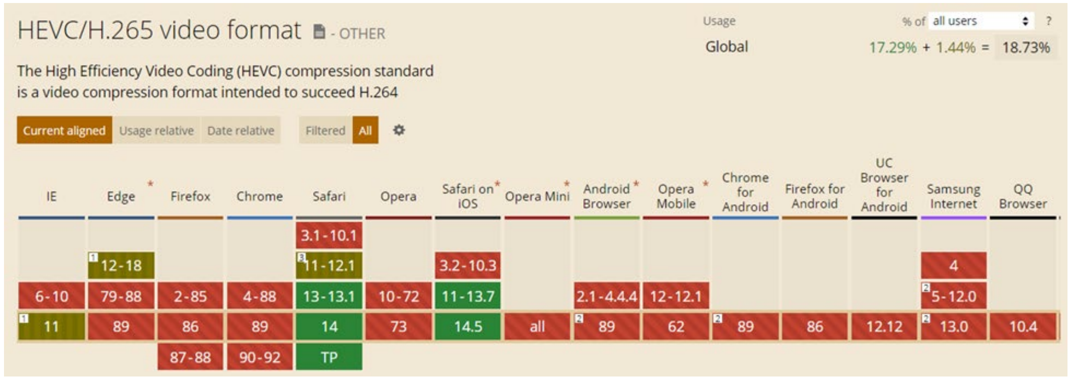
Current aligned | Usage relative | Date relative | Filtered | All

IE	Edge *	Firefox	Chrome	Safari	Opera	Safari on iOS *	Opera Mini *	Android Browser *	Opera Mobile *	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	QQ Browser
								2.1-2.2						
		2-20						[1] 2.3						
6-8		21-34		3.1	10-24			3-4.3						
9-10	12-88	35-85	4-88	3.2-13.1	25-72	3.2-13.7		4.4-4.4.4	12-12.1				4-12.0	
11	89	86	89	14	73	14.5	all	89	62	89	[2] 86	12.12	13.0	10.4
		87-88	90-92	TP										

Figure 1. Support of H.264/AVC codec across different platforms (Source: www.caniuse.com).

However, in terms of technology, H.264/AVC codec is pretty old. In recent years, several more advanced codecs have been introduced. The two best-known ones are: the HEVC [3] from ISO/ISO MPEG and ITU-T committees and AV1 [4] from the Alliance for Open Media. Both technologies claim about 40-50% gain in compression efficiency over the H.264/AVC [6-8]. Even higher gains have been recently reported for an emerging VVC coding standard [5,8,9].

In theory, such gains should lead to a significant reduction in streaming costs. However, in practice, these new codecs can only reach particular subsets of the existing devices or web browsers. For example, for HEVC, www.caniuse.com reports only 18.73% reach, primarily due to Apple devices incorporating hardware HEVC support. The AV1's support among web browsers is higher, but notably, it is not supported by Apple devices and most existing set-top box platforms [2,9].



HEVC/H.265 video format - OTHER

Usage % of all users

Global 17.29% + 1.44% = 18.73%

The High Efficiency Video Coding (HEVC) compression standard is a video compression format intended to succeed H.264

Current aligned | Usage relative | Date relative | Filtered | All

IE	Edge *	Firefox	Chrome	Safari	Opera	Safari on iOS *	Opera Mini *	Android Browser *	Opera Mobile *	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	QQ Browser
				3.1-10.1										
	[1] 12-18			[3] 11-12.1		3.2-10.3							4	
6-10	79-88	2-85	4-88	13-13.1	10-72	11-13.7		2.1-4.4.4	12-12.1				[2] 5-12.0	
[1] 11	89	86	89	14	73	14.5	all	[2] 89	62	[2] 89	86	12.12	[2] 13.0	10.4
		87-88	90-92	TP										

Figure 2. Support of HEVC codec across different platforms (Source: www.caniuse.com).

Moving forward, we can expect additional codecs (VVC, AV2 and others) to be introduced [9] contributing to futher fragmentation of codec support across media devices.

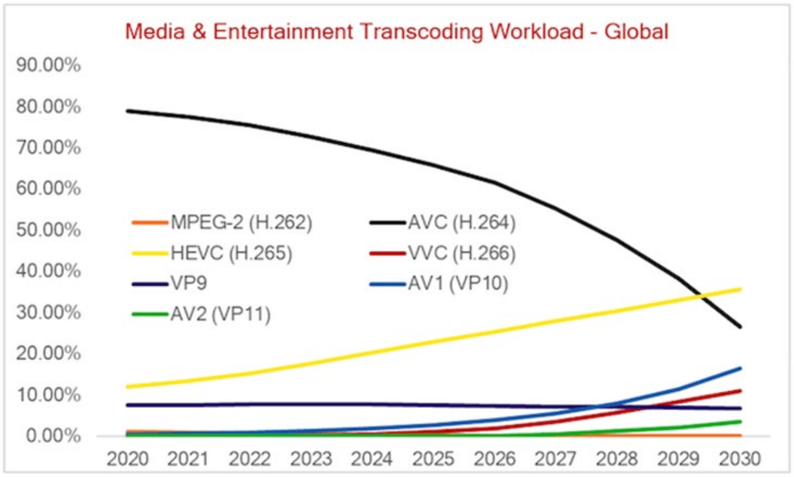


Figure 3. Predicted usage of video codecs in the next 10 years (Source: RethinkTV [9]).

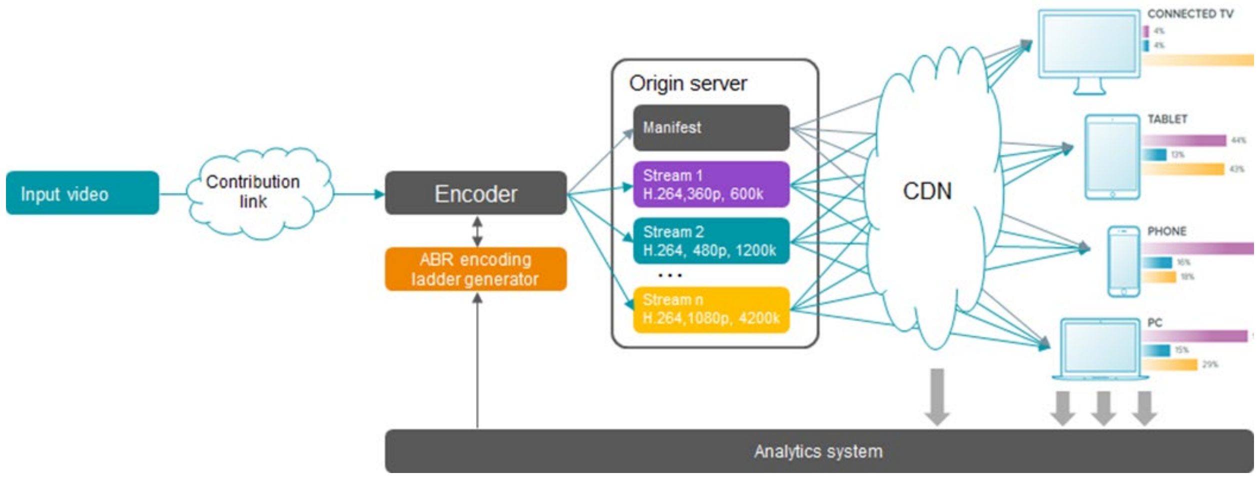
This situation brings a question – how can we serve such a fragmented population of devices most efficiently by using codecs delivering the best performance in all cases, yet producing the minimum possible number of streams, and such that the overall cost of media delivery is minimal?

In this paper, we will explain how this problem can be formalized and solved at the stage of dynamic generation of encoding profiles for ABR streaming. The proposed solution effectively generalizes the per-title or context-aware encoding (CAE) class-of techniques [10-15], considering multiple sets of renditions generated using each codec and codec usage distributions by the population of the receiving devices. The proposed solution also utilizes advanced signaling mechanisms in HLS [16] and MPEG DASH [17], as well as current deployment guidelines and interoperability requirements developed for these standards [18-20]. An example of the practical realization of such a multi-codec streaming system and various additional means for increased robustness and reach will also be described.

Background Information

Adaptive Bitrate Streaming: Main Principles

Let us first briefly review the main principles of operation of modern-era Adaptive Bit-Rate (ABR) streaming systems based on HLS [16] or DASH [17] standards. We show a conceptual diagram of such a system in Figure 4. For simplicity, we only consider the VOD delivery case.



Input video
Contribution link
Encoder
ABR encoding ladder generator
Origin server
Manifest
Stream 1 H.264,360p, 600k
Stream 2 H.264, 480p, 1200k
Stream n H.264,1080p, 4200k
CDN
CONNECTED TV
TABLET
PHONE
PC
Analytics system

Figure 4. Typical architecture of HTTP-based adaptive-bitrate streaming system.

As shown in this figure, when an input video is prepared for ABR streaming, it is typically transcoded into several *renditions* (or *variant streams*). Such streams may have different bitrates, resolutions, and other codecs- and presentation-level parameters.

Once all renditions are generated, they are placed on an HTTP web server called the *origin server*. Along with the set of renditions, the origin server also receives *manifest files* describing the properties of the encoded streams. In the HLS streaming standard [16], such manifests are called "playlists," while in the MPEG-DASH standard [17], they are called "media presentation description" or "mpd" files.

The subsequent delivery of the encoded content to user devices is done over HTTP and by using Content Delivery Network (CDN). The use of CDNs ensures the reliability and scalability of the delivery system.

For playback, user devices use special software called *streaming clients*. For example, a streaming client can be a JavaScript run by a web browser. It may also be a custom application or a video player supplied by the OS. Regardless of the implementation, most streaming clients include the logic for adaptive selection of streams/renditions during the playback.

For example, if the client notices that the observed network bandwidth is too low to support real-time playback of the current rendition, it may switch to a lower bitrate rendition. This switch prevents buffering. Otherwise, the client may switch to a higher bitrate / higher quality rendition if there is sufficient bandwidth. This switch leads to a better quality of experience. This logic is what makes streaming delivery adaptive. It is also the main reason for using multiple (typically 5-10) renditions. For information about the origins of ABR streaming and other benefits of multi-rate design, the reader is referred to publications [21-23]

The system depicted in Figure 4 has two additional components: the *analytics system*, collecting the playback statistics from CDNs and streaming clients, and the *ABR encoding ladder generator*, defining the number of renditions and properties of each rendition to create. For example, in the Brightcove VideoCloud streaming system [24], this block corresponds to Context-Aware Encoding (CAE) [25] module. We will explain the significance of the encoding ladder generation operation next.

Encoding Ladders

Let us now consider an example of an encoding ladder created for streaming. This specific example was created using the Brightcove CAE generator [25] for action-movie content.

Table 1. Example single-codec ladder.

Rendition	**Codec**	**Resolution**	**Framerate**	**Bitrate** [Kbps]	**Quality** [MOS]
1	H.264	384x216	25	261.59	2.178
2	H.264	512x288	25	513.54	2.719
3	H.264	768x432	25	1024.37	3.408
4	H.264	1280x720	25	2075.71	4.215
5	H.264	1920x1080	25	4203.03	4.769

This ladder defines 5 streams, enabling video delivery with resolutions from 216p to 1080p and using from about 260 Kbps to 4200 Kbps in network bandwidth. All streams are encoded using H.264/AVC codec [1]. The last column in this table lists perceived visual quality scores as estimated for playback of these renditions on a PC screen. These values are reported using the standard Mean Opinion Score (MOS) scale [26].

In Figure 5, we plot the (bitrate, quality) points corresponding to all renditions. We also show a trend of the best quality levels achievable by the streaming system with varying network bandwidth. This trend becomes a step function, shown in blue.

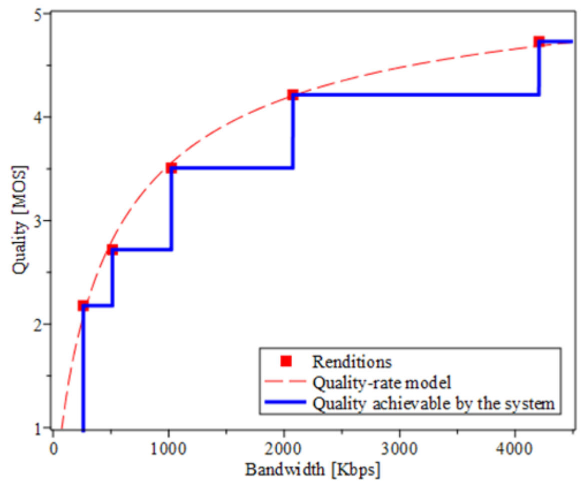


Figure 5. Quality delivered by a streaming system employing a ladder from Table 1.

Figure 5 also plots the so-called quality-rate model function [12-14]. This function describes the quality achievable by encoding the content with the same encoder operating at every possible bitrate point within the same range of bitrates. A dashed red curve shows this function.

As it can be easily grasped, with proper ladder design, the rendition points become a subset of points from the quality-rate model, and the step function describing quality achievable by streaming becomes an approximation of this model. What influences the quality of the streaming system is the number of renditions in the encoding ladder and the placement of renditions along the bandwidth axis. The closer the resulting step function is to the quality-rate model, the better quality the streaming system can deliver.

This example shows that the encoding profiles/ladders for ABR streaming must be carefully designed. This is why most modern streaming systems employ special profile generators to perform this step dynamically by accounting for the properties of the content, networks, and other relevant contexts. Related techniques are known as "per-title," "content-aware," and "context-aware" encoding techniques [10-15].

Multi-Codec Streaming: Encoding Ladders and Adaptation Models

Now that we've explained the key concepts, we can turn our attention to multi-codec streaming. To make this presentation more specific, let us consider an example of an encoding ladder generated using two codecs: H.264/AVC and HEVC.

Table 2. Example encoding ladder for two codecs: H.264/AVC and HEVC.

Rendition	**Codec**	**Resolution**	**Framerate**	**Bitrate** [Kbps]	**Quality** [MOS]
1	H.264	384x216	25	261.59	2.178
2	HEVC	512x288	25	300	2.529
3	H.264	512x288	25	513.54	2.719
4	HEVC	768x432	25	607.89	3.260
5	H.264	768x432	25	1024.37	3.408
6	HEVC	1024x576	25	1166.03	3.793
7	H.264	1280x720	25	2075.71	4.215
8	HEVC	1600x900	25	2362.74	4.549
9	H.264	1920x1080	25	4203.03	4.769
10	HEVC	1920x1080	25	4203.45	4.915

The plots of rendition points, quality-rate models, and quality achievable by streaming clients decoding only sets of H.264 and HEVC streams are presented in Figure 6.

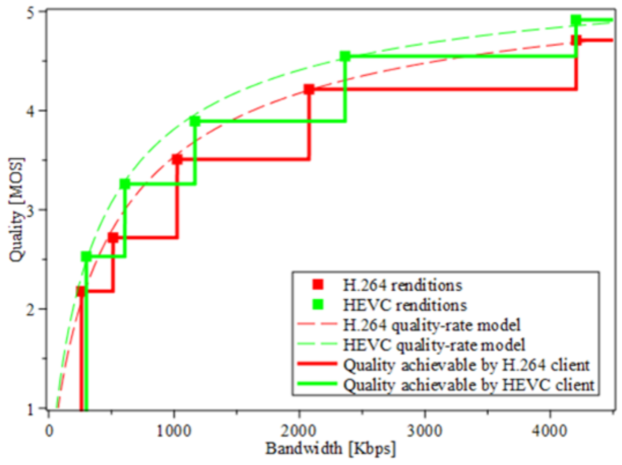


Figure 6. Quality delivered by streaming clients using H.264 and HEVC ladders from Table 2.

As easily observed, the quality-rate model function for HEVC is consistently better than the quality-rate model for H.264/AVC. By the same token, HEVC renditions should also deliver better quality-rate tradeoffs than renditions encoded using H.264/AVC encoder.

However, considering that there are typically only a few rendition points, and they may be placed sparsely, and in an interleaved pattern, this may create regions of bitrates, where H.264/AVC renditions may deliver better quality than the nearest HEVC rendition of smaller or equal bitrate. Such areas in the figure above are seen when step functions for H.264/AVC clients go above the same for HEVC clients.

This observation implies that decoding only HEVC-encoded streams in such a ladder does not automatically result in the best possible quality! Clients that selectively and intelligently switch between H.264/AVC and HEVC streams may achieve even better quality. We illustrate the quality achievable by such "2-codec clients" in Figure 7

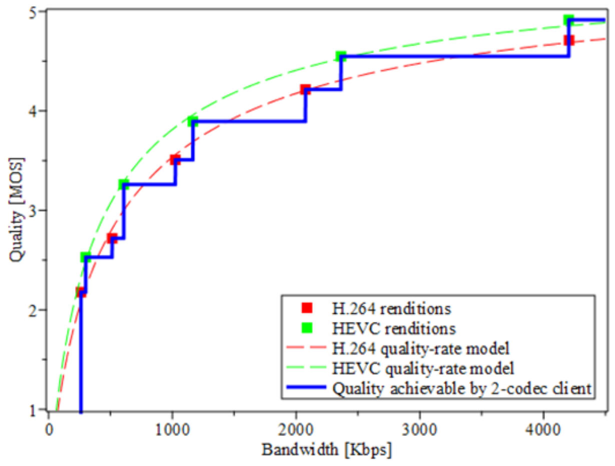


Figure 7. Quality delivered by a streaming system employing a "2-codec client" and H.264+HEVC ladders from Table 2.

In this example, the 2-codec client can make 9 adaptation steps instead of just 5 in HEVC or H.264-only ladders. This enables better utilization of the available network bandwidth and delivery of better quality overall. This also allows fewer renditions using both H.264 and HEVC codecs to be generated, as both subsets are effectively used for adaptation.

Multi-Codec Support in Existing Devices

As we observed, the ability of the streaming client not only to decode but also intelligently and seamlessly switch between H.264/AVC and HEVC streams is extremely important. This leads to better quality and allows the reduction in the total number of streams, reducing streaming costs.

Perhaps the best-known examples of existing clients supporting codec switching are native players in most new Apple devices: iPhones, iPads, Mac computers, etc. They can decode and seamlessly switch between H.264/AVC and HEVC streams. Recent versions of Chrome and Firefox web browsers have added support for so-called *changeType()* method, which allows

JavaScript-based streaming clients to implement switching between codecs. Using this method we may expect most web-browser-based clients to add codec switching capability in the future.

But there are indeed some other platforms, such as some SmartTVs, set-top boxes, etc., that can only decode either H.264/AVC or HEVC streams and won't switch to another codec during a streaming session. And naturally, many legacy devices can only decode H.264/AVC encoded streams.

This fragmented space of streaming clients and their capabilities must be understood and accounted for at the stages of the generation of encoding ladders, properly defining HLS and DASH manifests, and the design of the delivery system for multi-codec streaming.

Optimal Generation of Multi-Codec Encoding Profiles

In this section, we will discuss the problem of optimal design of encoding ladders for multi-codec streaming. Our presentation will follow the notations and concepts introduced earlier in [12,13].

Assumed Multi-Codec System

Consider a system with two codecs and three types of clients, with a conceptual processing diagram presented in Figure 8.

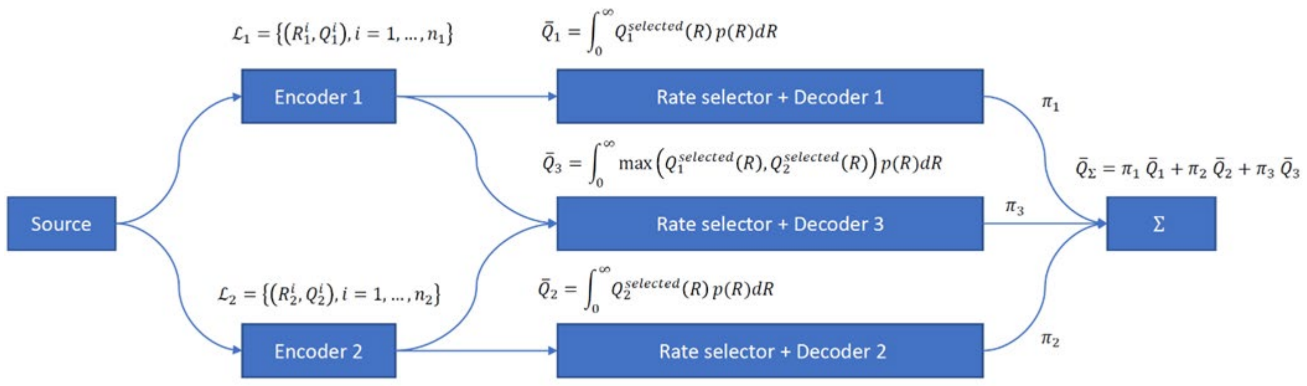

$\mathcal{L}_1 = \{(R_1^i, Q_1^i), i = 1, \dots, n_1\}$
$\bar{Q}_1 = \int_0^\infty Q_1^{selected}(R)\, p(R) dR$
Encoder 1
Rate selector + Decoder 1
π_1
$\bar{Q}_3 = \int_0^\infty \max\left(Q_1^{selected}(R), Q_2^{selected}(R)\right) p(R) dR$
$\bar{Q}_\Sigma = \pi_1 \bar{Q}_1 + \pi_2 \bar{Q}_2 + \pi_3 \bar{Q}_3$
Source
π_3
Rate selector + Decoder 3
Σ
$\mathcal{L}_2 = \{(R_2^i, Q_2^i), i = 1, \dots, n_2\}$
$\bar{Q}_2 = \int_0^\infty Q_2^{selected}(R)\, p(R) dR$
Encoder 2
Rate selector + Decoder 2
π_2

Figure 8. Conceptual diagram of an ABR streaming system with two encoders and three types of decoders/clients. Decoders 1 and 2 can only receive and decode streams encoded with Encoders 1 and 2, respectively. A decoder of type 3 can decode and switch between streams produced by both encoders.

Translating this system to practice, we can think that codec 1 is H.264/AVC, codec 2 is HEVC, client 1 is the one that can only decode H.264/AVC streams, client 2 is the one that can decode HEVC streams, and client 3 is the one that can decode both H.264/AVC or HEVC streams and can switch between them during the streaming session.

By $\mathcal{L}_1$ and $\mathcal{L}_2$ we denote *encoding ladders* produced for codecs of type 1 and 2, respectively. Mathematically, we will assume that each ladder is presented by a set of (rate, quality) points, corresponding to the bitrate and quality characteristics of its renditions:

$$\mathcal{L}_1 = \left\{\left(R_1^i, Q_1^i\right), i = 1, \dots, n_1\right\}, \quad \text{and} \quad \mathcal{L}_2 = \left\{\left(R_2^i, Q_2^i\right), i = 1, \dots, n_2\right\}$$

Here, by R_j^i we denote bitrates and by Q_j^i we denote the quality values of each rendition. The sub-indices indicate codec type. The upper indices indicate the rendition number in each set. The values $n_1$and n_2 denote the number of renditions in each set and $n = n_1 + n_2$ denotes the total number of all renditions used for streaming.

As in reference [12], we further assume that the performance of each codec is modeled by *quality-rate functions*: $Q_1(R)$ and $Q_2(R)$, and that the above (quality, rate) points can be understood as samples taken from these functions:

$$Q_1^i = Q_1\left(R_1^i\right), i = 1, \dots, n_1, \quad \text{and} \quad Q_2^i = Q_2\left(R_2^i\right), i = 1, \dots, n_2.$$

Also, as in reference[12], the behavior of the *network bandwidth* during the streaming session is modeled as a *continuous random variable* R with known (or empirically measured) probability density function $p(R)$.

On the receiver-end, we assume that clients follow the *idealized client model* [12], selecting the maximum ladder rate R^i that is less or equal than available network bandwidth R:

$$R_1^{selected}(R) = \max_{\substack{i=1,..,n_1 \\ R_1^i \le R}} R_1^i\,, \quad \text{and} \quad R_2^{selected}(R) = \max_{\substack{i=1,..,n_2 \\ R_2^i \le R}} R_2^i\,.$$

Assuming monotonically increasing quality-rate models, this also implies that:

$$Q_1^{selected}(R) = \max_{\substack{i=1,..,n_1 \\ R_1^i \le R}} Q_1\left(R_1^i\right), \quad \text{and} \quad Q_2^{selected}(R) = \max_{\substack{i=1,..,n_2 \\ R_2^i \le R}} Q_2\left(R_2^i\right).$$

Average Quality Delivered by the Streaming System

Next, given all the above definitions, we write expressions for the *average quality* values achievable by clients/decoders of each kind. For instance, the *average quality* achieved by client 1 when working with a ladder $\mathcal{L}_1$ and network with bandwidth distribution $p(R)$ becomes:

$$\bar{Q}_1 = \int_0^\infty Q_1^{selected}(R)\, p(R) dR.$$

Similarly, the *average quality* achieved by client 2, working with ladder $\mathcal{L}_2$, becomes:

$$\bar{Q}_2 = \int_0^\infty Q_2^{selected}(R)\, p(R) dR.$$

Finally, for client 3, which is capable of working with both ladders $\mathcal{L}_1$ and $\mathcal{L}_2$, including switching between all such streams, the expression for the average quality becomes:

$$\bar{Q}_3 = \int_0^\infty \max\left(Q_1^{selected}(R), Q_2^{selected}(R)\right) p(R) dR.$$

The $\max(.)$ function in the above expression reflects the capability of a dual-codec client to select the best streams across renditions encoded by both codecs.

The average quality achieved by the entire population of clients becomes:

$$\bar{Q}_\Sigma = \pi_1\, \bar{Q}_1 + \pi_2\, \bar{Q}_2 + \pi_3\, \bar{Q}_3,$$

where π_1, π_2, π_3 denote normalized ($\pi_1 + \pi_2 + \pi_3 = 1$) population counts of clients of each kind.

Optimal Multi-Codec Ladder Design

By considering all the above definitions and by observing that average quality value $\bar{Q}_\Sigma$ can be understood as a function of network bandwidth density $p(R)$, client distribution π, the number of ladder points n_1, n_2, and the sets of rates used in the ladder, we are now ready to define the following ladder optimization problem.

Given:

- the total number of ladder points n,
- limits for all rate points: $R_{\min}$, $R_{\max}$;
- maximum limits for first rate points: $R^1_{\max}$,
- quality-rate functions for both codecs and content $Q_1(R)$, $Q_2(R)$
- network bandwidth density $p(R)$, and
- distribution of clients π,

Find:

- numbers $\hat{n}_1, \hat{n}_2$, such that $\hat{n}_1 + \hat{n}_2 = n$, and
- ladder rates $\hat{R}^1_1, \dots, \hat{R}^{\hat{n}_1}_1$ and $\hat{R}^1_2, \dots, \hat{R}^{\hat{n}_2}_2$

such that overall quality $\bar{Q}_\Sigma$ delivered by the streaming system is maximal:

$$\bar{Q}_\Sigma\left(p, \pi, n, \hat{R}^1_1, \dots, \hat{R}^{\hat{n}_1}_1, \hat{R}^1_2, \dots, \hat{R}^{\hat{n}_2}_2\right) = \max_{\substack{n_1 + n_2 = n \\ R_{\min} \le R^1_1 \le \dots \le R^{n_1}_1 \le R_{\max} \\ R_{\min} \le R^1_2 \le \dots \le R^{n_2}_2 \le R_{\max} \\ R^1_1, R^1_2 \le R^1_{\max}}} \bar{Q}_\Sigma\left(p, \pi, n, R^1_1, \dots, R^{n_1}_1, R^1_2, \dots, R^{n_2}_2\right). \quad (*)$$

As easily noticed, this problem folds into a class of non-linear constrained optimization problems. Additional details about the design of a numerical solver for this problem and examples of solutions can be found in [13].

Additional Constraints and Considerations

We first note that all constraints introduced in the above problem setting (*) are essential for practical applications. For example, the maximum rate limit $R_{\max}$ is needed to prevent the allocation of bitrates beyond those that are physically achievable. The minimum rate limit $R_{\min}$ is usually related to the minimum quality level at which streaming service is feasible. The limit on the maximum first bitrate in the ladder $R^1_{\max}$ is typically used to limit *start-up latency* and *buffering probability* of the clients, etc. In practice, several additional constraints may also be imposed.

We also note that while the problem setting (*) does not explicitly operate with the *video resolutions* for each stream, it assumes that the best choices of video resolutions are already absorbed in the definition of quality-rate models $Q_1(R)$, $Q_2(R)$.

In other words, given a set of allowed resolutions $\mathcal{S}$ and quality-rate models $Q_1(S,R)$, obtained for each specific resolution $S \in \mathcal{S}$, we will assume that the final quality rate model $Q_1(R)$ is defined such that

$$Q_1(R) = \sup_{S \in \mathcal{S}} Q_1(S,R).$$

This way, the ladder optimization problem becomes effectively reduced to the choice of bitrates needed for each rendition.

Practical Implementations

The described mathematical problem and its solution finder form the basis for the design of the profile generator in the Brightcove CAE system [25].

As shown in Figure 9, the Brightcove CAE system includes several "ingest profiles" with the "Multiplatform Extended HEVC (CAE) mixed-codec" profile corresponding to a mode where both H.264 and HEVC codecs are employed. Our example profiles, shown in Tables 1 and 2, have been generated using this system.

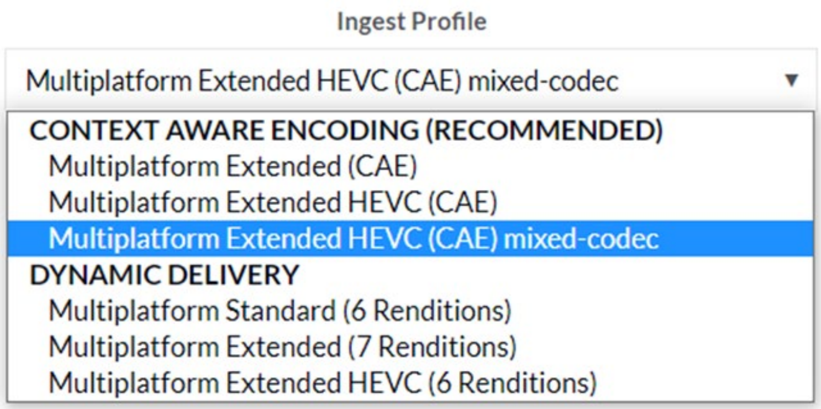

Ingest Profile
Multiplatform Extended HEVC (CAE) mixed-codec
CONTEXT AWARE ENCODING (RECOMMENDED)
Multiplatform Extended (CAE)
Multiplatform Extended HEVC (CAE)
Multiplatform Extended HEVC (CAE) mixed-codec
DYNAMIC DELIVERY
Multiplatform Standard (6 Renditions)
Multiplatform Extended (7 Renditions)
Multiplatform Extended HEVC (6 Renditions)

Figure 9. Multi-codec streaming options in Brightcove VideoCloud system [24,25].

When operating this system, the users can customize some overall ladder parameters – such as the limits on the numbers of renditions, ranges of resolutions and bitrates, assumed network and usage distributions, etc. But the final choices for the number of renditions, and their parameters (resolutions, bitrates, codec HRD controls, etc.) are all made automatically by the CAE system.

Using Multi-Codec Features of HLS and DASH

In designing multi-codec streaming systems, particular attention must be paid to the proper generation of HLS and DASH manifests.

One of the essential components of such design is the inclusion of *quality attributes* needed to guide the proper selection of renditions by the streaming clients. This is important because in mixed-codec ladders, the increase in stream bitrates may no longer mean an increase in quality. In our example ladder in Figure 6, the monotonicity of bitrate-quality relations is preserved by proper choices of codecs and bitrates, but more generally, it may not be the case. An HEVC stream with a lower bitrate may have a better quality than H.264/AVC stream using more bits. *Hence quality-related attributes are essential.*

Additionally, there is always the need to make sure manifests are generated in a backward-compatible fashion such that older generation clients, which can only recognize a subset of declarations, would still operate properly.

There are also specific advanced features of both HLS [16] and DASH [17] standards, as well as limitations imposed by HLS deployment guidelines [18] and DASH-IF interoperability requirements [19], that become important to account for in such designs.

To explain some of these nuances, in Figures 10 and 11, we show sketches of the HLS and DASH manifests constructed for a composition of H.264 and HEVC streams from the ABR ladder in Table 2.

```
#EXTM3U
…
#EXT-X-STREAM-INF:BANDWIDTH=267868,CODECS="avc1.4d401e",RESOLUTION=384x216,SCORE=1, ...
Rendition1.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=307200,CODECS="hvc1.1.6.L90.90",RESOLUTION=512x288,SCORE=2, ...
Rendition2.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=525864,CODECS="avc1.4d401e",RESOLUTION=512x288,SCORE=3, ...
Rendition3.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=75378,CODECS="hvc1.1.6.L90.90",RESOLUTION=768x432,SCORE=4, ...
Rendition4.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=1048954,CODECS="avc1.4d401e",RESOLUTION=768x432,SCORE=5, ...
Rendition5.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=1194014,CODECS="hvc1.1.6.L93.90",RESOLUTION=1024x576,SCORE=6, ...
Rendition6.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=2125527,CODECS="avc1.640028",RESOLUTION=1280x720,SCORE=7, ...
Rendition7.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=2419445,CODECS=”hvc1.1.6.L120.90",RESOLUTION=1600x900,SCORE=8, ...
Rendition8.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=4303902,CODECS="avc1.640028",RESOLUTION=1920x1080,SCORE=9, ...
Rendition9.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=4304332,CODECS="hvc1.1.6.L120.90",RESOLUTION=1920x1080,SCORE=10, ...
Rendition10.m3u8
…
```

Figure 10. HLS master playlist (.m3u8) file created for ABR ladder from Table 2.

```
MPD xmlns="urn:mpeg:dash:schema:mpd:2011" minBufferTime="PT1.500S" type="static" … >
<Period duration="PT0H12M14.167S">
<SupplementalProperty schemeIdUri="urn:mpeg:dash:qr-equivalence:2019" value="1,2" />
<AdaptationSet id="1">
  <SupplementalProperty schemeIdUri="urn:mpeg:dash:adaptation-set-switching:2016" value="2" />
  <Representation id="1" mimeType="video/mp4" codecs="avc1.42001e" bandwidth="267868"   qualityRanking="10" …/>
  <Representation id="2" mimeType="video/mp4" codecs="avc1.42001e" bandwidth="5525864" qualityRanking="8" …/>
  <Representation id="3" mimeType="video/mp4" codecs="avc1.42001e" bandwidth="1048954" qualityRanking="6" …/>
  <Representation id="4" mimeType="video/mp4" codecs="avc1.640028" bandwidth="2125527" qualityRanking="4" …/>
  <Representation id="5" mimeType="video/mp4" codecs="avc1.640028" bandwidth="4303902" qualityRanking="2" …/>
</AdaptationSet>
<AdaptationSet id="2"">
  <SupplementalProperty schemeIdUri="urn:mpeg:dash:adaptation-set-switching:2016" value="1" />
  <Representation id="1" mimeType="video/mp4" codecs="hvc1.1.6.L90.90" bandwidth="307200"   qualityRanking="9" …/>
  <Representation id="2" mimeType="video/mp4" codecs="hvc1.1.6.L90.90" bandwidth="75378"     qualityRanking="7" …/>
  <Representation id="3" mimeType="video/mp4" codecs="hvc1.1.6.L93.90" bandwidth="1194014" qualityRanking="5" …/>
  <Representation id="4" mimeType="video/mp4" codecs="hvc1.1.6.L120.90" bandwidth="2419445" qualityRanking="3" ./>
  <Representation id="5" mimeType="video/mp4" codecs="hvc1.1.6.L120.90" bandwidth="4304332" qualityRanking="1" ./>
</AdaptationSet>
</Period>
</MPD>
```

Figure 11. DASH media presentation description (mpd) file created for ABR ladder from Table 2.

As shown in these figures, in the HLS system, all HEVC and H.264/AVC renditions can be included in the natural order in the master playlist.

However, in MPEG DASH, mixed-codec renditions *must be listed separately*, in *different adaptation sets*, and defined independently for each codec. To enable switching between HEVC and AVC renditions, the "adaptation-set-switching:2016" SupplementalProperty descriptors must also be included in each adaptation set.

Such separate placement of renditions in DASH is needed to improve compatibility with legacy players. It is required by DASH-IF Interoperability Guidelines [19]. This practice also complies with CTA WAVE content specification [20], which defines its media profiles on a per-codec level.

Looking at Figures 10 and 11, we next note that the means for adding *quality annotations* in HLS and DASH standards are quite different.

In HLS, quality-related parameters are called *"SCORE"* attributes, with higher values indicating better quality. In MPEG DASH, they are called *"qualityRanking"* attributes, but now with lower values indicating better quality.

Furthermore, DASH's *"qualityRanking"* attributes must indicate the correct relative order of renditions within each (codec-specific) adaptation set and globally across all renditions. To tell the clients that they can rely on such global order of "qualityRanking" attributes, a supplemental property "qr-equivalence:2019" descriptor should also be included in the manifest file.

All other standard content-related requirements and constraints as specified in HLS authoring specification [18] and DASH-IF interoperability guidelines [19] must also be considered in the design of encoding profiles and manifests for multi-codec streaming.

Additional Practical Considerations

While the above-described declarations in HLS and DASH manifest should ensure interoperability across all proper implementations of HLS and DASH clients, in practice, there may still be a risk that some legacy clients may be confused by the presence of redundant renditions or manifest descriptors and will not start playing the stream or will not play it well.

Suboptimal behaviors may also occur with newer devices/clients, which attempt to support new codecs, but with implementations that are not yet mature.

Advanced streaming systems may employ edge-side device detection and manifest filtering to avoid such situations or behaviors. Such logic may use the "user_agent" string in HTTP request headers to detect the type of the device, OS, or web browser and then dynamically decide which renditions or attributes to retain in the manifest.

For example, if the client device is classified as one that can only decode H.264/AVC streams, then only H.264/AVC streams can be retained in the manifest, and the rest of them are pruned.

Such logic may generally help increase the streaming system's robustness, maintainability, and cross-platform reach. In practice, such extra logic is easily combinable with existing edge-level functions that advanced streaming systems employ for CDN selection, multi-format management, redundancy management, and other advanced delivery functions[15,27].

Conclusions

We considered the problem of fragmentation of codec support across streaming devices. We've shown that it can be addressed by creating multi-codec sets of renditions for HLS or DASH streaming systems with the optimized design of the encoding profiles, proper generation of manifests, and manifest filtering functions of the streaming platform.

References

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

[2] "Can I use" web service, https://caniuse.com/, March 18, 2022.

[3] ISO/IEC 23008-2:2013, "Information technology – High efficiency coding and media delivery in heterogeneous environments – Part 2: High efficiency video coding," ISO/IEC, December 2013.

[4] AV1, "AV1 Bitstream and Decoding Process Specification, Version 1.0.0," Alliance for Open Media, January 18, 2019.

[5] ISO/IEC 23090-3:2021, "Information technology — Coded representation of immersive media — Part 3: Versatile video coding," ISO/IEC, February 2021.

[6] T. Laude, Y. G. Adhisantoso, J. Voges, M. Munderloh and J. Ostermann, "A Comparison of JEM and AV1 with HEVC: Coding Tools, Coding Efficiency and Complexity," Picture Coding Symposium (PCS), 2018, pp. 36-40.

[7] P. Topiwala, M. Krishnan, and W. Dai, "Performance comparison of VVC, AV1, and HEVC on 8-bit and 10-bit content," Proc. SPIE 10752, Applications of Digital Image Processing XLI, September 17 2018.

[8] D. Grois et al., "Performance Comparison of Emerging EVC and VVC Video Coding Standards with HEVC and AV1," SMPTE Motion Imaging Journal, vol. 130, no. 4, pp. 1-12, May 2021.

[9] RethinkTV research, "Media & Entertainment Transcoding Workload and Device Royalty Forecast 2020-2030", https://rethinkresearch.biz/reports-category/rethink-tv/#media-entertainment-transcoding-workload-and-device-royalty-forecast-2020-2030, April 30, 2021.

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

[11] 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.

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

[15] 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.

[16] R. Pantos and W. May, "HTTP live streaming, RFC 8216," IETF, 2017.

[17] ISO/IEC 23009-1:2019, "Information technology - Dynamic adaptive streaming over HTTP (DASH) - Part 1: Media presentation description and segment formats," ISO/IEC, August 2019.

[18] Apple, "HTTP Live Streaming (HLS) Authoring Specification for Apple Devices," https://developer.apple.com/documentation/http_live_streaming/http_live_streaming_hls_authoring_specification_for_apple_devices, Apple, November 12, 2021.

[19] DASH-IF, "DASH-IF Interoperability Guidelines, v.3," https://dashif.org/docs/DASH-IF-IOP-v4.3.pdf, DASH-IF, November 15, 2018.

[20] CTA-5001-A, "Web Application Video Ecosystem – Content Specification", https://cdn.cta.tech/cta/media/media/resources/standards/pdfs/cta-5001-a-final.pdf, CTA, December 2018.

[21] D. Wu, Y.T. Hou, W. Zhu, Y-Q. Zhang, and JM Peha, "Streaming video over the internet: approaches and directions," IEEE Trans. CSVT, vol. 11, no. 3, pp. 282-300, 2001.

[22] 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.

[23] 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.

[24] Brightcove VideoCloud platform, https://www.brightcove.com/en/onlinevideo-platform

[25] Brightcove Context-Aware Encoding, https://apis.support.brightcove.com/general/overview-context-aware-encoding.html

[26] ITU-R Rec. BT.500, "Methodology for the subjective assessment of the quality of television pictures," ITU-R, October, 2019.

[27] Y. Reznik, J. Cenzano, B. Zhang, "Transitioning Broadcast to Cloud," Proc. 2020 NAB Broadcast Engineering and Information Technology Conference, Las Vegas, NV, May 13-14, 2020.