Evaluation Of Bitrate Ladders For Versatile Video Coder

Reda Kaafarani*, Mederic Blestel*
Thomas Maugey†, Michael Ropert*, Aline Roumy†
**Mediakind France, Rennes*, Email: firstname.lastname@mediakind.com
†*Inria Rennes, France*, Email: firstname.lastname@inria.fr

Abstract—**Many video service providers take advantage of bitrate ladders in adaptive HTTP video streaming to account for different network states and user display specifications by providing bitrate/resolution pairs that best fit client's network conditions and display capabilities. These bitrate ladders, however, differ when using different codecs and thus the couples bitrate/resolution differ as well. In addition, bitrate ladders are based on previously available codecs (H.264/MPEG4-AVC, HEVC, etc.), i.e. codecs that are already in service, hence the introduction of new codecs e.g. Versatile Video Coding (VVC) requires re-analyzing these ladders. For that matter, we will analyze the evolution of the bitrate ladder when using VVC. We show how VVC impacts this ladder when compared to HEVC and H.264/AVC and in particular, that there is no need to switch to lower resolutions at the lower bitrates defined in the Call for Evidence on Transcoding for Network Distributed Video Coding (CfE).**

Index Terms—**bitrate ladder, VVC, HEVC, H.264/AVC**

I. Introduction

Nowadays, video streaming accounts for most of the internet's traffic. Indeed, it is expected that the percentage of video content of the total traffic will be around 80% (business and consumer combined) in 2021 compared to 67% in 2016 [1]. This amount of video consumption created a need for increasing compression efficiency in order to provide better video quality while lowering the bandwidth needed. This need was answered by the introduction of Advanced Video Coding (AVC) which is also referred to as H.264/MPEG4-AVC [2] in 2003, followed by the introduction of High Efficiency Video Coding (HEVC) [3] in 2013 that provides about 50% better data compression while maintaining a similar video quality when compared to H.264/AVC. Finally, Versatile Video Coding (VVC) [4] is the latest standard to be introduced jointly published by ITU-T and ISO/IEC. It was finalized in July of 2020, aiming for around 50% better compression than HEVC while having a similar perceivable video quality. In addition, the increasing need of video consumption motivated the development of enhanced and faster networks that are able to provide large bandwidth and lower latency like fiber optics, as well as broadband cellular network technologies like 4G [5] and more recently, 5G [6].

In order to offer the best experience to the users, video content providers rely on adaptive bitrate streaming. More precisely, technologies like Dynamic Adaptive Streaming over HTTP (DASH) [7], HTTP Live Streaming (HLS) [8], HTTP Dynamic Streaming (HDS) [9] and HTTP Smooth Streaming (HSS) [10] were introduced in order to enable seamless playback by offering various decoding solutions of the same content. Adaptive streaming technology consists in processing various representations of a content, i.e. different couples of resolution and bitrate, also known as *bitrate ladder*. Such processing provides the end-user devices the ability to select the representation that best fits its display capabilities and network conditions.

However, the traditional use of a fixed bitrate ladder across content is not optimal for every type of content in Video on Demand (VoD) nor in live streaming. To that end, different methods were proposed in order to find new ways to define bitrate ladders for various scenarios. In [11], Netflix introduced a per-title encoding optimization approach where they create bitrate ladders that are content dependent, meaning for a certain video, depending on the complexity of its scenes, the optimal encoding scheme (resolution and bitrate) is determined. In addition, authors in [12] introduced a content-driven approach to implement VMAF-based bitrate ladder estimation using machine learning models. It's worth mentioning, however, that improving the bitrate ladder is not the only concern. The introduction of a new codec technology alters this ladder and hence forces a new study of the optimal rates for different resolutions. In [13], an approach that provides optimal ladders for different types of networks as well as types of content was implemented. The work was on three subsets of mobile devices that can either decode H.264/AVC, HEVC or both. The focus was to design encoding profiles and to define multiple streams encoded by both codecs to achieve an optimal quality (best possible reconstruction of the videos) out of the system. To summarize, the focus of previous works was to improve a fixed bitrate ladder, which leaves it's evolution through different codecs somehow untouched.

In this paper, we discuss the bitrate ladder for VVC and we analyze how it alters this ladder due to its efficiency when compared to older codecs (H.264/AVC and HEVC). We also show that the number of resolutions can be reduced while keeping the same number of representations.

The rest of the paper is structured as follows. Methodology will be presented in section II. In section III we will provide comparisons between bitrate ladders of different codecs and discuss the results. Finally, a conclusion will be made in section IV.

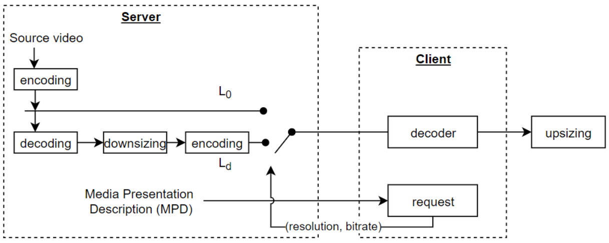


Fig. 1. Example of adaptive bitrate streaming system

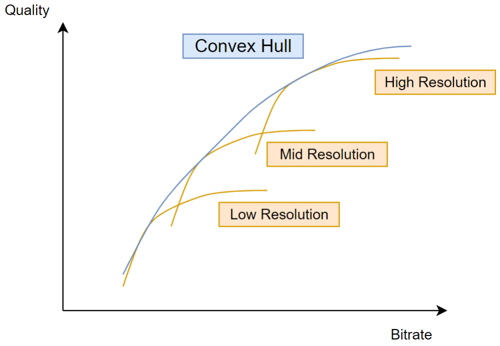


Fig. 2. Example of a convex hull

II. Methodology

A. Bitrate Ladder

Let's suppose a typical streaming scenario as depicted in Fig. 1, where a certain user wants to play a video (originally at resolution L_0) available at the server, the user having a resolution called L_d and a bandwidth limited to a rate $R \leq R_{max}$. The transmission should be adapted to match the client's display resolution L_d as well as his bandwidth R_{max}. The Adaptive Bitrate Streaming (ABR streaming) system's goal is to provide this video with the best quality while matching the client's constraints.

To do so, the ABR does not change the bitrate encoding parameter of the video coder. For some R_{max}, it might be more efficient to decrease the input video resolution before encoding. For instance, encoding a given content into a down-scaled resolution could provide a better objective video quality than encoding it using the native resolution. That could happen when the content complexity is very high compared to the bitrate. In this case, using the native resolution, the encoder will be compelled to increase the quantization step in order to respect the bitrate constraint, resulting in the loss of a lot of high frequency data. Using the same bitrate and a lower resolution could be beneficial as the encoder will use a lower quantization step, resulting in less degradation of the signal and a higher video quality.

Based on that, an ABR system is able to provide the user with a down-scaled video that gets up-scaled (to L_d) at the decoder for playback. Since the resolution on which a user watches a video could differ from L_0, it's worth noting that comparing video qualities for two different resolutions is not obvious and that the evaluation of the quality metric should be done at the desired resolution after up/down-sizing the content. In practice, upsizing and downsizing are not perfect meaning that both these operations introduce irreversible degradations that can not be fixed by the encoding process, and will directly impact the quality metric. So the better the upscaling and downscaling algorithms, the higher the video quality metric. In an ideal world, applying a downscale or an upscale should not modify the video, however, that is not the case and several assumptions need to be made. It should be considered that both up and down scaling algorithms need to be aligned, meaning they should be used under the same assumptions. In an ABR system, as depicted in Fig. 1, the end user is able to select the representation (bitrate/resolution couple) that best fits his network and display capabilities thanks to the Media Presentation Description (MPD) [14], that summarizes all the available representations, i.e. the bitrate ladder. More precisely,

$$(\text{representation}) = \text{bitrate ladder}(\underbrace{L_d}_{\text{displayed resolution}}, \underbrace{R_{max}}_{\text{network bandwith}})$$

In theory, providing bitrate ladders for every possible display resolution results in the best user experience. However, that would lead to a significant amount of processing. Therefore, in practice, only the bitrate ladder for the highest resolution L_0 is provided and the video is up/down-scaled to match the display resolution L_d of the user.

B. Convex Hull

In practice, the bitrate ladder is estimated thanks to the *convex hull*. Suppose we have some content encoded at three different resolutions (high, mid and low) on a range of bitrates spanning very low to very high bitrates like in Fig. 2. We can see that for each of the available resolutions, there is a bitrate range where each resolution performs better than the other ones. These ranges for all the resolutions form a boundary that is called the *convex hull*.

It's worth noting that, whatever the quality metric (SSIM [15], PSNR, VMAF [16] etc.), it's always evaluated at the highest resolution for all of the available resolutions (low, mid and high) of a certain content. In an ideal scenario, the streaming system should work exactly on the *convex hull*, but that is not feasible since it takes only a select set of resolutions. Therefore, video providers work as close to the *convex hull* as possible by selecting the closest bitrate-quality pair to the curve [11]. By computing the *convex hull*, a streaming system is able to select the resolution that will provide the best video quality. As expected, this *convex hull* will differ from a codec to the other, since the more efficient the codec, the less bitrate it needs to encode at a certain resolution with a good quality. That has "sliding" effect on the rate-distortion curves where the points of the curve slide towards the left the more efficient the codec gets. Depending on the efficiency of a certain codec, there may be less need to encode content at lower resolutions as this codec may be able to encode very efficiently using the highest possible resolutions in a larger bitrate range. If that is not the case, encoding at lower resolutions and then up-scaling to the highest resolution will be better in terms of quality for the same bitrate.

III. Results

A. Codec configuration

To compare the impact of H.264/AVC, HEVC and VVC codec on the bitrate ladder, the following encoders and configurations were used:

- Regarding VVC codec, Fraunhofer Versatile Video Encoder (VVenC) [17] version 0.2.1.0 was used. It includes all VVC main features and is a fast and efficient implementation. From all available presets, the predefined preset ”medium” was used in the experiment as it provides a good trade-of between run-time and compression efficiency. Note that using a higher preset would have increased the compression efficiency but providing a limited impact on the bitrate ladder.
- For HEVC, the reference software HM [18] (Encoder version 16.2) was used.
- Considering H.264/AVC, the reference software JM [19] was used (version 19.0).

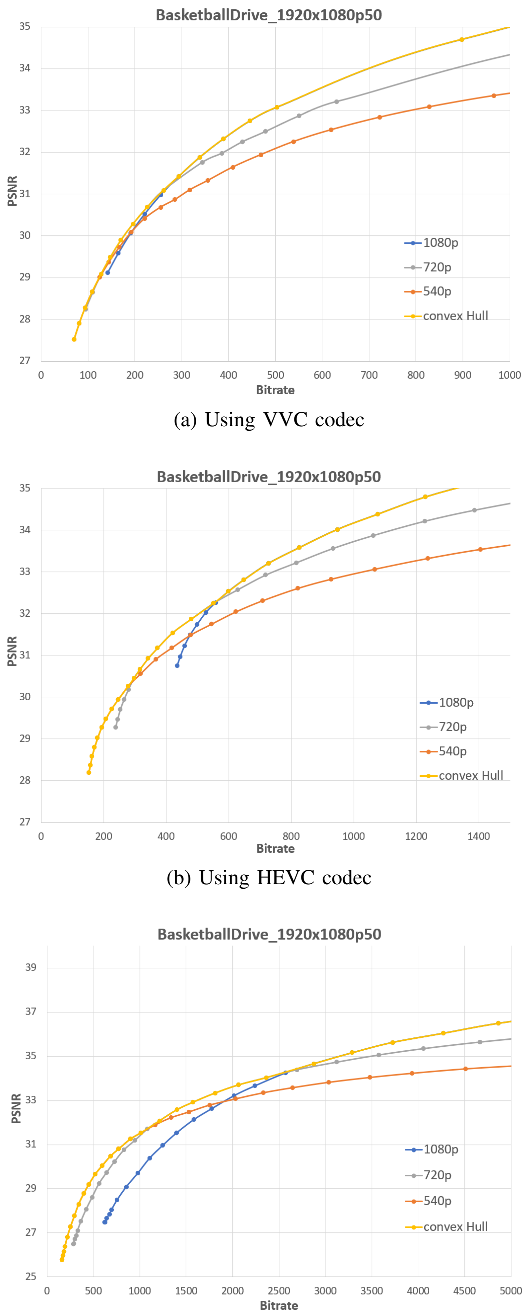
For the experiments, all encoders were configured using a Random Access scenario derived from their respective common test conditions [20]. Common parameters were:

- Intra/Key frame period: all INTRA periods are set to 24, 32, 48 and 64 according to the content frame rate, 24, 30, 50 and 60 fps, respectively.
- Constant QP configuration.
- Bit depth: the internal coding bit depth follows the input content bit depth.
- Referencing structure: default referencing structures were used for each codec.

B. Calculating the convex hull

The tests were carried on a total of 5 sequences which are classB videos from the common test conditions, i.e. videos with a 1920x1080 resolution, also referred in the paper as 1080p. The sequences are the following: *BasketballDrive_1920x1080p50, BQTerrace_1920x1080p60, Cactus_1920x1080p50, Kimono1_1920x1080p24 and ParkScene_1920x1080p24.*

Using FFmpeg [21], each of these sequences are downscaled from 1080p to 720p and 540p which are a subset of the resolutions provided in the Call for Evidence (CfE) on Transcoding for Network Distributed Video Coding (NDVC) [22]. Then for each sequence at each resolution, encodings are done using the three codecs mentioned in III-A over all QPs in the range of 22 to 51. Once all the encodings are done, the PSNR is computed with the reference being the original uncompressed video. For our paper, we decided to work with PSNR for the *convex hull* quality metric. It’s worth noting that the lower resolution encodings are upsampled back to the native resolution, here 1080p, using FFmpeg before computing the metrics. In order to calculate the *convex hull* for a given sequence (using a specific codec), all the data from all the resolutions for that specific sequence are presented in curves like the ones shown in Fig. 2. After that, the data points for each resolution that outperform the next resolution (higher quality with same or lower bitrate) are selected in order to present the *convex hull*. These points can also be easily selected based on the intersection points of curves. Starting from the lowest resolution, we select all the points up until the intersection with the curve of the next resolution, then we select the points of the next resolution until the next intersection and so on.



(a) Using VVC codec

(b) Using HEVC codec

(c) Using H.264/AVC codec

Fig. 3. Convex hull for BasketballDrive sequence using 3 different codecs

Fig. 3 shows 3 different *convex hull*s for the sequence BasketballDrive using the 3 codecs mentioned in III-A. The *convex hull* is shown using a bright orange curve where it can be seen covering the bitrate range where each resolution is

TABLE I
BITRATE SWITCH POINTS TO LOWER RESOLUTIONS FOR EACH SEQUENCE (IN KBPS)

Sequence	1080p to 720p			720p to 540p		
	VVC	HEVC	H.264/AVC	VVC	HEVC	H.264/AVC
BasketballDrive_1920x1080p50	261.36	552.02	2360.3	108.98	278.64	1013.3
BQTerrace_1920x1080p60	118.12	288.84	853.95	55.22	137	295.98
Cactus_1920x1080p50	275.63	555.49	2000.22	96.58	271.58	615.81
Kimono1_1920x1080p24	184.32	909.45	3190.24	127.28	296.86	1642.07
ParkScene_1920x1080p24	129.69	336.26	1386.32	82.97	130.2	534.21

TABLE II
CALL FOR EVIDENCE (CFE) BITRATE LADDER VS VVC OPTIMAL RESOLUTION

	Resolution	
Bitrate (Kbps)	CfE	VVC optimal resolution
5600	1920x1080	1920x1080
4000	1920x1080	1920x1080
2800	1280x720	1920x1080
2200	1280x720	1920x1080
1800	960x540	1920x1080
1500	960x540	1920x1080

better than the one directly above it.

C. VVC lowers the intersection points

Thanks to Fig. 3 (a), we can note that the intersection between the 1080p and 720p curves in case of VVC occurs at around 270 Kbps. It can be noticed that this point is more than twice lower than that in the case of HEVC (Fig. 3 (b)) and almost 10 times lower than that of H.264/AVC (Fig. 3 (c)). This can be explained by the fact that VVC is more efficient than HEVC and H.264/AVC. Indeed, new tools in VVC like the increase of the number of possible partitions or the multiple transform set, allow for a better adaptation to the content characteristics. Table I shows the different bitrates at which we switch from the higher resolution to the next lower one, i.e. from 1080p to 720p, and from 720p to 540p for all the sequences we used. It's worth noting that the "switch" bitrates simply mean that at this bitrate or at a lower bitrate, encoding using a lower resolution, and then upscaling to the highest resolution provides a higher quality video than encoding at the highest resolution. We can see that all sequences exhibit a similar behaviour of the one seen in *BasketballDrive* where the switch from 1080p to 720p happens at more than 2 times lower bitrates in case of VVC when compared to HEVC. When compared to H.264/AVC, this switch happens at bitrates that are lower by 7 times or more. The switch from 720p to 540p shows close results to those seen in the previous case.

D. The need for lower resolutions in VVC

Looking at Fig. 3 (a), it can be seen that the 1080p resolution has a wider range of bitrates where it performs better than lower resolutions. Users will be able to watch 1080p VVC videos at low bitrates, while lower resolutions would be used when encoded using HEVC or H.264/AVC codecs. As a consequence, switching between resolutions will only happen at very low bitrates.

TABLE III
AVERAGE BD PSNR AND SSIM DIFFERENCE WHEN USING 1080P ONLY

	Quality Metric	
Sequence	SSIM	PSNR
BasketballDrive	-21.19%	-23.41%
BQTerrace	-30.12%	-31.33%
Cactus	-14.32%	-19.42%
Kimono1	-15.26%	-16.63%
ParkScene	-25.74%	-24.04%
Average	**-21.33%**	**-22.97%**

To that end, a test was conducted on the 5 sequences mentioned in III-B, resolutions and their respective bitrates were taken from the CfE document and are presented in Table II. Results are shown in Table III by means of Bjøntegaard Delta (BD) [23] metric for all the sequences over 6 points for each sequence. The table compares using 1080p resolution for all the points (bitrates) as defined by the *convex hull* presented in III-B rather than using 1080p, 720p and 540p defined in the CfE. Using representations close to the *convex hull* provides around -21% in SSIM and around -23% in PSNR bitrate savings when compared to the bitrate ladder defined in the CfE. This result motivates the introduction of the "VVC optimal resolution" column in Table II as users will then benefit from the native resolution (here 1080p) on a wider bitrate range even in more challenging conditions like with a limited bandwidth.

IV. CONCLUSION

In video streaming services, bitrate ladders are used to provide content representations to best serve the client based on his network and device capabilities. In this paper, we analyse the performance of VVC when compared to older codecs, which is seemingly an untouched subject in the literature. By computing the *convex hulls*, we show that VVC allows for better use of the higher resolution than the other codecs, by providing a wider range of bitrates that are able to stream that resolution. In addition, our analysis could be supported by activating new tools in VVC, like Reference Picture Resampling (RPR) which allows adapting the encoding resolution on a per-frame basis, reducing the number of samples to be encoded. VVC codec will then completely reshuffle existing bitrate ladders by reducing the number of required resolutions, while providing up to -23% of additional bandwidth savings compared to an un-reviewed bitrate ladder.

REFERENCES

[1] Cisco, "VNI Complete Forecast Highlights," 2021. [Online]. Available: https://www.cisco.com/c/dam/m/en_us/solutions/service-provider/vni-forecast-highlights/pdf/Global_2021_Forecast_Highlights.pdf

[2] J.-R. Ohm and G. Sullivan, "MPEG-4 Advanced Video Coding," Jul. 2005. [Online]. Available: https://mpeg.chiariglione.org/standards/mpeg-4/advanced-video-coding

[3] J.-R. Ohm and G. Sullivan, "High Efficiency Video Coding," Jan. 2011. [Online]. Available: https://mpeg.chiariglione.org/standards/mpeg-h/high-efficiency-video-coding

[4] MPEG, "Versatile Video Coding." [Online]. Available: https://mpeg.chiariglione.org/standards/mpeg-i/versatile-video-coding

[5] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); Relay architectures for E-UTRA (LTE-Advanced)," TR36.806, Jan. 2015.

[6] ETSI, "Fifth Generation Fixed Network (F5G);F5G Generation Definition Release #1," GR F5G 001, Dec. 2020.

[7] MPEG, "Dynamic Adaptive Streaming Over HTTP," ISO/IEC 23009. [Online]. Available: https://mpeg.chiariglione.org/standards/mpeg-dash

[8] Apple, "HLS Authoring Specification for Apple Devices." [Online]. Available: https://developer.apple.com/documentation/http_live_streaming /hls_authoring_specification_for_apple_devices

[9] "HTTP Dynamic Streaming, an overview." [Online]. Available: https://www.encoding.com/http-dynamic-streaming-hds/

[10] "Microsoft Smooth Streaming, Microsoft's Answer to Adaptive Bitrate Streaming." [Online]. Available: https://www.encoding.com/microsoft-smooth-streaming/

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

[12] A. V. Katsenou, K. S. Fan Zhang, M. Afonso, J. Sole, and D. R. Bull, "VMAF-based Bitrate Ladder Estimation for Adaptive Streaming ," arXiv:2103.07564, Mar 2021.

[13] Y. A. Reznik, X. Li, K. O. Lillevold, A. Jagannath, and J. Greer, "Optimal Multi-Codec Adaptive Bitrate Streaming," in *2019 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, 2019, pp. 348–353.

[14] R. Garrison, "Structure of a MPEG-DASH MPD – A Detailed Guide," April 2021. [Online]. Available: https://ottverse.com/structure-of-an-mpeg-dash-mpd/

[15] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[16] Z. Li, A. Aaron, I. Katsavounidis, A. Moorthy, and M. Manohara, "Toward A Practical Perceptual Video Quality Metric," Jun. 2016.

[17] Fraunhofer, "Fraunhofer Versatile Video Encoder (VVenC)." [Online]. Available: https://github.com/fraunhoferhhi/vvenc

[18] Fraunhofer, "HEVC Test Model." [Online]. Available: https://vcgit.hhi.fraunhofer.de/jvet/HM

[19] Fraunhofer, "Joint Test Model." [Online]. Available: https://vcgit.hhi.fraunhofer.de/jvet/JM

[20] K. Sharman and K. Suehring, "Common Test Conditions," Jan. 2017, document JCTVC-Z1100.

[21] FFmpeg, "FFmpeg Documentation." [Online]. Available: https://ffmpeg.org/ffmpeg.html

[22] MPEG, "Call for Evidence on Transcoding for Network Distributed Video Coding," Jul. 2017. MPEG document N17058.

[23] G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves," *document VCEG-M33 of ITU-T Video Coding Experts Group (VCEG)*, Apr. 2001.