SELECTIVE STORAGE: STORE AND DELIVER ONLY WHAT MATTERS

M. Blestel, J. Le Tanou, M. Ropert

Mediakind, France

ABSTRACT

Video streaming services must satisfy the growing consumer request on having the best video quality, anywhere, anytime. As such, and in order to optimize the end-user video quality for various network and end-device capabilities, Over The Top (OTT) video delivery infrastructure massively relies on adaptive bitrate (ABR) streaming technology. In the common scenario, it consists in processing and storing various compressed representations of the content to stream, i.e. different couples of resolution and bitrate, also known as bitrate ladder. As both processing and storage resources are expensive, each content provider must optimize its operating and infrastructure costs. This paper discusses the existing limitations in the ABR streaming landscape, and introduces an innovative *Selective Storage* algorithm that tackles one of the ABR streaming main challenges: the storage cost. This algorithm reduces up to 30% the amount of required storage for the same video quality and remains compliant with existing optimizations into the OTT ecosystem.

INTRODUCTION

Video consumption has become the dominant traffic on fixed and mobile networks and should continue to grow (1) in the future to meet consumer expectations of getting a better quality of experience, anywhere, anytime.

The video delivery landscape can be divided in two different categories: linear TV and non-linear TV. The first category is the traditional television that is broadcasted with a scheduled program. At the opposite, non-linear TV allows the consumer to access a library of video contents and to watch a specific program whenever she/he wants.

First, Internet TV has initiated the change with set-top boxes, then connected TVs and more affordable subscriptions to VOD platforms have fostered non-linear TV consumption (2). At the same time, thanks to mobile device screen sizes increase, and better network bandwidths video consumption have shifted from home to outdoor entertainment.

Overall, Over-The-Top (OTT) video delivery services, have massively adopted adaptive bitrate (ABR) streaming technology to optimize the end-user video quality for various network and end-device capabilities. In the common scenario, it consists in processing and storing various compressed representations of the content to stream, i.e. different couples of resolution and bitrate, also known as bitrate ladder (3).

Consequently, ABR streaming solutions can offer the best video quality for each user but at the expense of a high processing or storage cost for handling the different coded representations. The higher the resolution and the target bitrate, the greater the storage and bandwidth requirements for delivering content. A significant amount of video segments must therefore be processed and stored on the HTTP server before being delivered to the end-user. As both processing and storage resources are expensive, each content provider must optimize its operating and infrastructure costs. Storage cost optimization is the main purpose of the solution further described in this paper.

First, adaptive streaming principles, benefits, and related ecosystem are recalled. Then, the related limitations and challenges are drawn from and discussed. Finally, it is introduced an innovative *Selective Storage* algorithm that helps to optimize storage costs, by reducing the number of video segments (for all the representations) stored on the HTTP server.

ADAPTIVE STREAMING

Principles and Benefits

Adaptive streaming is a technique to deliver video content to the user with the highest possible quality by adapting the video stream to its viewing conditions: i.e. its end-device resolution capability and network bandwidth. By design, it resolves most of poor video quality and re-buffering issues for the end-user, while saving transmission (i.e. bandwidth) costs for the provider.

Everyone is now able to watch video contents with a resolution that better fits the device capability whatever it’s a TV, a smartphone, a tablet, ...To enable this, instead of streaming contents encoded in one single resolution and bitrate, an adaptive streaming solution provides a content encoded in different resolutions. This guarantees to best fit the device capabilities, offering a better user experience (i.e. without unnecessary up-scale/down-scale), and most importantly saving bandwidth (i.e. not streaming costly high resolutions that would require down-scale by the end-device).

In the other way around, to adapt the video stream to poor-quality network and to increase service eligibility, an adaptive streaming solution also provides content of a given resolution encoded at several bitrates. It enables the device to switch to another bitrate according to its network bandwidth and to avoid playback interruption. Better the network, higher the bitrate.

Thus, by encoding contents in various representations, i.e. couples of resolution-bitrate, known as bitrate ladder, a content provider can enlarge its audience as network conditions and devices capabilities can fluctuate, while saving bandwidth and transmission costs.

Adaptive Streaming Ecosystem

As for terrestrial or satellite distribution, ABR streaming encompasses content processing, content storage, and networking, but each function is significantly different.

ABR over HTTP has enabled the advent of OTT platforms. Contrary to other ecosystems, OTT operates as a pull system. Using the Media Presentation Description (MPD), the OTT device is aware of all available representations of the content and can request the specific one according to its display and network capabilities. This particularity puts pressure on the infrastructure as the video service may have to deal with many customers’ requests and needs to deliver thousands of different files at the same time.

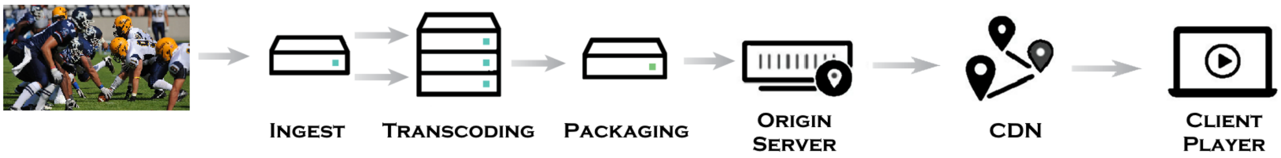


Figure 1: Overview of an OTT ecosystem

In an OTT ecosystem (Figure 1), the operator needs efficient storage with smart ingest and categorization to provide a personalized viewing experience, including dynamic Ad insertion (DAI) and Digital Rights Management (DRM). Once all representations have been encoded, a packager chops them into multi-second segments (generally between 2 and 10 seconds) and wraps it in different packages like HLS (HTTP Live Streaming) (4), DASH (Dynamic Adaptive Streaming over HTTP) (5) to be available on as many devices as possible (e.g. Android, Apple, etc.). Then all ABR streams will be moved to the central storage, where there will be accessible by the origin server to be streamed through the Content Delivery Network (CDN).

The CDN is composed of groups of servers, geographically distributed to faster delivery of the video contents. Its main purpose is to provide caching, which consists of storing copies of files in a temporary location. Then, on a customer request, the closest server will be able to deliver the content more quickly and to decrease the origin server load by reducing the number of requests, and consequently the origin server internet traffic.

Adaptive streaming is now the standard way of streaming content compared to single constant bitrates so that the viewing experience is improved even in networks with highly variable performance. However, video delivery services based on adaptive streaming will have to overcome several challenges to continue growing as today.

ADAPTIVE STREAMING LIMITATIONS AND CHALLENGES

Glass to Glass Latency

As already mentioned, ABR streaming enhances the video quality and end-user experience for all but can considerably increase the glass to glass latency. During the latest Super Bowl (6), the delay between real-time and various streaming sources was measured around 40 seconds, leading to spoilers and a reduced user experience, with respect to legacy IPTV or broadcast delivery. This delay can be explained by the traditional segmentation mechanism and the usage of third-party IP networks. In the general case, the closer the user from the origin server the shorter the delay.

First, the latency can come from the CDN itself. If the content is already in cache, the CDN latency will be optimal. If not, it needs to pull content segments from the origin server, delaying the delivery.

At the end-user side, the device can buffer several multi-second video segments to protect the decoding process and playback from any interruption. Buffering a 6 seconds video segment will create at least 12 seconds delay, if ever the device starts decoding a segment only when completely downloaded. Reducing the buffer size and starting to decode the chunk while being downloaded are common relevant delay optimizations for the playback.

More recently, using CMAF (7) file format, in its low latency version, over MPEG-DASH (5) allows to chop the segments into smaller fragments (~100-200 ms), therefore reducing the buffering size and the decoding start delay. As opposed to about 30 seconds end-to-end latency for legacy ABR streaming delivery, combining both CMAF-LL+DASH lowers the end-to-end latency down to few seconds.

Fixed, Content Driven or Dynamic Ladder?

To deploy an ABR channel, the OTT operator defines a bitrate ladder (8), that consists in defining a set of multiple “resolution-bitrate” couples. This bitrate ladder needs to be well defined to deliver good subjective video quality and a sustainable stream, decodable in various network conditions. As an example, Apple provides requirements for both VOD and Live video content using HLS (4).

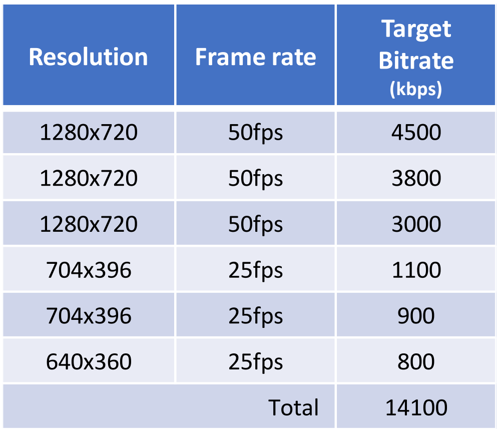
Given the variety of contents' complexity, a bitrate ladder cannot be optimal for every content; a high bitrate for easy content will lead to waste bandwidth, while a too low bitrate for complex content will result in poor video quality. It’s then obvious that a bitrate ladder defined for a sports content would be sub-optimal for a movie, and reciprocally.

An optimal solution would be to have a dynamic bitrate ladder that would be adapted according to the scene complexity. Such solution (3) can be easily used as an encoding strategy but the whole ABR framework needs to support it. Furthermore, dynamic bitrate ladders generate a supplemental delay due to the content pre-analysis. In the case of live events, this delay comes in addition of the conveying chain. Using Content-Aware-Encoding (CAE) (9-10), a content provider can reduce the playback bandwidth while maintaining the same quality of experience. On a per-scene encoding basis, it will adapt the bitrate for each representation, then consequently it will also help to reduce the storage bit cost.

Video Segments Storage

Even for live events delivery, storage capacity requirement is significant to enable OTT use cases. According to the playback options (e.g. time shifting, replay, start over, etc.) offered by the OTT operators, and the available primary packaging formats (e.g. HLS, HDS, DASH, and MSS), the amount of storage could increase considerably. Each packaged video segment needs to be distributed across all edge cache locations to improve the delivery process; more popular videos will be stored as close as possible to the end device to offload the traffic between the origin server and caches.

Let’s consider an OTT operator which gives its customer a 7 days’ time-shifted viewing window on a single ABR stream using the bitrate ladder of Table 1, and which uses 3 different packaging formats to address as many customers/devices as possible. Then, the operator would need more than 3.2 Terabytes of storage on the CDN. Additionally, to reduce the latency, the stream would be distributed across many edge cache locations, multiplying the storage by as many caches exist. Increasing the number of streams and increasing the time-shift period will proportionally increase the amount of required storage.



Resolution	Frame rate	Target Bitrate (kbps)
1280x720	50fps	4500
1280x720	50fps	3800
1280x720	50fps	3000
704x396	25fps	1100
704x396	25fps	900
640x360	25fps	800
	Total	14100

Table 1: OTT bitrate ladder example

In order to reduce the amount of storage, several strategies can be designed. In the general case, all contents are stored as it is streamed to the consumer, i.e. each representation is pre-encoded and prepared within different packaging formats to be ready to deliver on request. To reduce the storage overhead, an intermediate packaging format for the storage can be defined. Then, Just-In-Time-Transcoding (JITT) (11) and Just-In-Time Packaging (JITP) (12) solutions could be applied. It consist in transcoding and packaging video assets on-the-fly to the requested format. However, JITT is not well deployed as it causes additional processing delay and requires a pretty high processing cost relatively to storage cost.

At the encoding step, and as mentioned in the previous section, a more suitable solution consists in using the right number of bits according to the content complexity. The nature of networks in addition to the bitrate switching mechanism makes it possible to get rid of the legacy CBR model. Using a unique or a limited number of bitrate ladders, the encoding step can be optimized using Constant Video Quality (CVQ) (13) algorithm that aims at adapting the bitrate (subject to a max-bitrate constraint) to guarantee to the end-user a constant subjective quality. Applied for each representation, such model saves bits on low complexity scenes and improves subjective quality on complex scenes with respect to legacy CBR-based encodings. This kind of solution reduces up to 20 % the encoding bit rate, then decreases the storage proportionally.

Even if some solutions exist to reduce the bit cost of packaged video segments, the overall storage capacity requirement remains huge. The proposed algorithm introduced in next section can help on further optimizing the storage bit cost while being fully compatible with previous approaches.

SELECTIVE STORAGE ALGORITHM, AN INNOVATIVE SOLUTION

Video Segment Pruning

Selective storage is a simple and pragmatic approach which takes advantage of the video content chunking/segmentation mechanism by reshuffling the video segments. In short, for every timestamp, it reduces the number of video segments before the packaging by pruning non-relevant video segments with respect to a given Rate-Distortion (R-D) criterion. By reducing the number of stored video segments, the *Selective storage* solution deflates not only the storage but also the amount of processing in the whole chain of transmission, i.e. at the packaging and the caching stages, reducing the global transmission cost.

As mentioned, the patented pruning algorithm can be driven by various R-D criteria, such it is able to prune in real-time the non-relevant segments without degrading the end-user subjective video quality.

For each timestamp defined in the MPD by a start time and a duration, the algorithm estimates the relevance of each video segment with respect to other video segments from close representations by minimizing a given R-D criterion. Once the estimation is done, either both segments are estimated as essential and are stored, either one segment is identified as irrelevant and is deleted.

The segment relevance estimation is mainly taking benefits from two situations.

First situation, the algorithm estimates the video quality difference with respect to the bitrate difference between segments of two representations. In several cases, the video quality gap between both segments is not noticeable whereas the bitrate increases. This case occurs when the bitrate for the current resolution is higher than needed for the current content complexity. In this case, as the bitrate surplus does not provide a significant video quality change, the video segment with the higher bitrate is deleted. Figure 2 shows the deletion of the 3rd segment of the P_4 representation (noted as P_{43}) since the segment with the same timestamp of the P_3 representation (i.e. P_{33}) provides a similar video quality with a smaller bit cost.

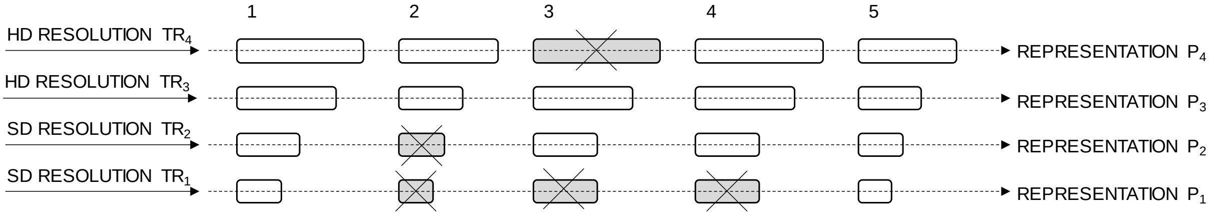


Figure 2: Example of video segments pruning for 4 different representations.

As a second situation, the video segment bit cost of a given representation could be lower than the target bit budget of a lower representation. Indeed, for low motion complexity scenes (e.g. cartoons, talking heads, advertising, etc.), encoders can achieve a high compression efficiency, then provide a very good video quality using fewer bits than the target bit budget. This specific situation is illustrated for P_{22}, P_{12}, P_{13}, and P_{14} in Figure 2. Therefore, these video segments can be deleted from the central storage and replaced into the playlist with the segment of a higher representation, here P_{32}, P_{23}, and P_{24}, respectively. The storage saving is less compared to the first situation though, due to its lower occurrence, and due to the lower size of the considered video segments.

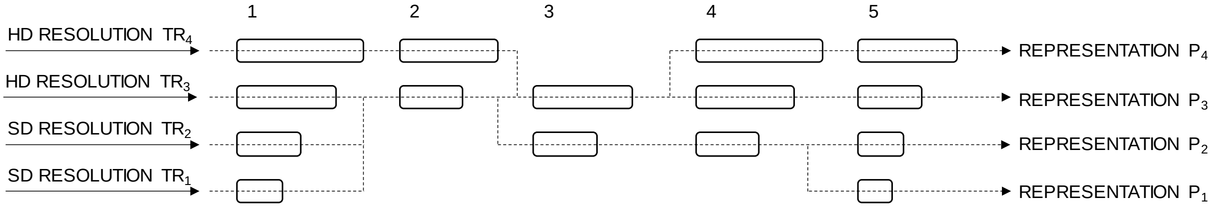


Figure 3: Example of remaining video segments after *Selective Storage* algorithm.

Figure 3 illustrates the remaining video segments out of the pruning algorithm and the resulting segment mapping for each representation if requested by the end-device. Figure 4 further depicts an example of the *Selective Storage* pruning algorithm based on the incoming scene complexity. In this example, 4 contents with various complexities are used,

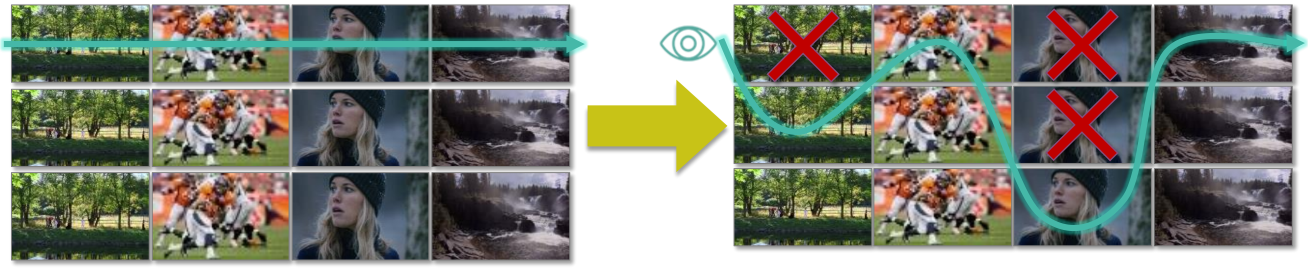


Figure 4: *Selective Storage* pruning according to the input content/scene complexity.

from high (the 2nd scene that is sport and the 4th that is a waterfall), to low complexity (the 3rd scene, a movie). Considering the high complexity scenes, no pruning is done since keeping high bitrate representation will prevent from compression video artifacts. For scenes with a medium complexity (here, the 1st scene), the algorithm can prune segments that provide a limited video quality gain relatively to the increase in bitrate. Finally, for the low complexity scenes, the algorithm can achieve higher pruning and storage saving, deleting most of the video segments from the high bitrate representations.

Segment Mapping

While packaging the ABR streams, the packager also generates a manifest file (MPD). The manifest file is an xml file that describes all the information (i.e. segment location, codec type, start time, duration, etc.) on existing video segments and the bandwidths they are associated with for one or multiple time-periods.

When *Selective Storage* is applied, several segments are removed for saving storage bit cost. Consequently, the MPD needs to consider these deletions to avoid referring to non-existing segment files. The manifest file is then processed such the references to the deleted segment files are mapped to the selected/preserved video segments. This is possible since multiple entries in the MPD can refer to the same segment file. If the algorithm is applied on an existing VOD library, all MPDs that refer to deleted video segments must be updated. In a Live context, by processing the algorithm before the packaging, no additional processing is required as the packager will generate right away the proper MPD.

Storage Savings of the Proposed Algorithm

To evaluate the performance of the *Selective Storage* algorithm, we set up an OTT line-up, close to existing deployments, and used the bitrate ladder defined in Table 1. The input stream is based on a variety of test contents, including sports, advertising, news, and documentary. For each timestamp, once all video segments are available, the algorithm, running in real-time, processes the pruning if needed. Note that in this simulation, the pruning algorithm was not processed between segments of different resolutions as no multi-resolution criterion was available.

Profile	Resolution	Frame rate	Target Bitrate (kbps)	Storage Saving (%)	Average bitrate (Kbps)
P_1	1280x720	50fps	4500	74,2%	3968
P_2	1280x720	50fps	3800	0,1%	3792
P_3	1280x720	50fps	3000	0,9%	2999
P_4	704x396	25fps	1100	46,6%	1004
P_5	704x396	25fps	900	0,3%	899
P_6	640x360	25fps	800	0,0%	800
Total			14100	27,5%	13462

Table 2: *Selective Storage* bit saving

Table 2 reports the benefits in storage saving and transmission rate when processing the *Selective Storage* algorithm. The storage saving on the origin server is measured for each representation of the ABR stream and computed as the relative change in Megabytes when the proposed algorithm is processed. Using the set-up of Table 1, the algorithm reduces by 27.5% the total required storage, and by ~4.5 % the transmission rate. As discussed earlier, the storage saving is unequally distributed over the representations, such by design usually most of the pruned video segments are from the highest target bitrate representations. In the evaluated scenario, it reduces by 74.2 % and 46.6% the storage of profiles P_1 and P4, respectively. We remark that P1 video segments deletion contributes to 85% of the global storage reduction. Regarding other profiles (P_2, P_3, P_5, P_6) the storage reduction is quite limited (< 1%) and questions the usage of a multi-resolution criterion to prune segments with different resolutions. Indeed, when analyzing the storage reduction for the P3 and P5 profiles (less than 1%), we can expect that a multi-resolution pruning algorithm may have a very limited impact due to the large bitrate difference and a very low occurrence probability while requiring additional processing to measure the video quality.

Considering the transmission bitrate saving, P_1 average bitrate is reduced by 11.8 % to 3968 kbps and P_4 transmission bitrate is reduced by 9.1 % to 1004 kbps. P_1 and P_4 average bitrate decrease are explained by the mapping and transmission of video segments from representations of lower bitrates.

Subjective Evaluation

In addition to the objective bitrate saving analysis, and in order to assess the impact of the *Selective Storage* algorithm on the end-user video quality (VQ), a paired comparison methodology derived from (14) was performed. A total of 20 subjects from Mediakind's employees were asked to choose their preference between two encodings of the same content: one out of a legacy ABR streaming set-up (i.e. without pruning) and the other out of the *Selective Storage* algorithm (i.e. with pruning and segment mapping). To properly assess the VQ impact on different resolutions, the streams were viewed in two different sessions on two different devices: a SONY 4K OLED 55A1 consumer display and a Xiaomi 9 Pro smartphone. For both devices, the videos were displayed in their native resolution without scaling, using an OTT player. We point out that the subjective evaluation included some representations of higher resolution that those defined from Table 1, especially 1080p resolutions.

The analysis of the results has shown that for both sessions (one for each device), no visible differences were identified between the two encoding scenarios. In most of the cases, the viewers had no preference between the legacy stream and the processed stream out of the *Selective Storage* algorithm. Those results validate the algorithm principle and efficiency in terms of storage saving for the same end-user video quality.

CONCLUSION

In the coming years, ABR streaming will continue to be the dominant way to deliver video content. The high video consumption growth challenges the related delivery infrastructure and requires further improvements on-top of existing ABR streaming techniques to reduce both storage and transmission costs. Based on the video segmentation mechanism, the OTT ecosystem makes possible to simply discard some video segments in a given bitrate ladder, enabling the proposed *Selective Storage* solution.

We describe in this paper the *Selective Storage* concept, as a simple and pragmatic approach for real-time processing. It relies on a patented pruning algorithm minimizing a Rate-Distortion criterion to identify non-relevant video segments, and to simply delete them from the storage. The mapping of the remaining segments is done by a simple MPD manipulation. No additional processing is required, keeping the conveying chain unchanged.

In comparison to other aggressive techniques like "full transcoding", the processing overhead/cost is negligible. Overall, the *Selective Storage* brings up to 30% storage and 5% transmission bandwidth savings for the same perceived video quality. Besides, it can be easily combined with other optimization techniques, such as CVQ.

Interestingly, and as future work, this technique could be applied on any existing stored video catalog. The only prerequisite would be to have a measure of the distortion per encoded video segment. If not available, it could be estimated from the quantization steps of segments for instance. Besides, *Selective storage* algorithm efficiency could be improved by enabling to prune video segments of different resolutions (e.g. based on a multi-scale distortion estimation).

REFERENCES

1. Nielsen, "The Nielsen total audience report: August 2020", https://www.nielsen.com/us/en/insights/report/2020/the-nielsen-total-audience-report-august-2020/, August 2020.
2. Report Linker, Video on Demand Market – Growth, Trends and Forecasts (2020 – 2025)", https://www.globenewswire.com/news-release/2020/09/23/2098311/0/en/ Video -on-Demand-Market-Growth-Trends-and-Forecasts-2020-2025.html, September 2020.
3. A.V Katsenou et al., « VMAF-based Bitrate Ladder Estimation for Adaptive Streaming", 2021.
4. Apple Inc., "HLS Authoring Specification for Apple Devices", https://developer .apple.com/documentation/http_live_streaming/hls_authoring_specification_for_apple_ devices
5. "MPEG DASH specification (ISO/IEC DIS 23009-1.2)", 2011.
6. Phenix, "Sports Latency Comparison", https://phenixrts.com/en-us/sports-latency.html February 2021.
7. ISO/IEC 23000-19:2018 Information technology –Multimedia application format (MPEG-A) – Part 19: Common media application format (CMAF) for segmented media, 2018, [online] Available: https://www.iso.org/standard/71975.html.
8. The Netflix Tech Blog, "Per-Title Encode Optimization", https://netflixtechblog.com/per-title-encode-optimization-7e99442b62a2, December 2014.
9. Streaming Media European Edition, "Buyers' Guide to Content- and Context-Aware Encoding 2018", https://www.streamingmediaglobal.com/Articles/Editorial/Featured-Articles/Buyers-Guide-to-Content--and-Context-Aware-Encoding-2018-123407.aspx?utm_source=related_articles&utm_medium=gutenberg&utm_campaign=editors_selection

10. Brightcove Video Cloud, “Overview of Context Aware Encoding“, https://apis.support .brightcove.com/general/overview-context-aware-encoding.html#:~:text=Context Aware Encoding uses advanced,bills) while maintaining visual quality.
11. S. Vonog, “Just-in-time transcoding of application content”, US Patent, W02012177779A3, June 2011.
12. Broadband Techology Report, “Just-in-Time Packaging for TV Everywhere”, https://www.broadbandtechreport.com/docsis/headend-hub/article/16446135/justintime-packaging-for-tv-everywhere, May 2020.
13. Mediakind, “Aquila Streaming”, http://www.mediakind.com/wp-content/uploads/2020/ 10/Aquila-Streaming-Datasheet.pdf.
14. ITU-T Rec. P.910, “Subjective Video Quality Assessment Methods for Multimedia”, April 2008.