

Evaluation of the Benefits of Variable Segment Durations for Adaptive Streaming

Susanna Schwarzmann*, Thomas Zinner*, Christian Sieber† and Stefan Geissler*

*Chair of Communication Networks, University of Würzburg, Am Hubland, 97074 Würzburg, Germany

Email: {susanna.schwarzmann,zinner,stefan.geissler}@informatik.uni-wuerzburg.de

†Chair of Communication Networks, Technical University of Munich, 80333 Munich, Germany

Email: c.sieber@tum.de

Abstract—HTTP Adaptive Streaming (HAS) is the de-facto standard for video delivery over the Internet. It enables the dynamic adaptation of video quality by splitting the video clip into small segments and providing multiple quality levels per segment. Current HAS streaming services typically utilize segments of equal durations. However, this leads to video encoding overhead as segments have to start with I-frames, independently of the encoded video content. In this paper we evaluate the prospects of variable segment durations, where video segments are aligned to the video characteristics. We evaluate the reduction of the encoding overhead and investigate its impact on the stalling probability using a theoretical model. It turns out that the variable approach outperforms the fixed approach in 86% of the evaluated cases with respect to video stalls.

Index Terms—Adaptive streaming, QoE, Segment duration, DASH

I. INTRODUCTION

On-demand video consumption by Internet video delivery has become the prevalent way of video consumption and a large fraction of the global Internet traffic can be attributed to on-demand video content [1]. MPEG dynamic adaptive streaming over HTTP (DASH) [2] is a widely adopted standard for Internet video delivery and allows the adaptation of the video quality to the available throughput and client capabilities. For DASH, the content is split typically into segments of 2 to 10 seconds length and encoded into multiple quality levels [3]. The segments are provided via HTTP and the location and properties of the segments are summarized in an XML-based media presentation description (MPD) file. A DASH client, e.g. a set-top box or browser, first requests the MPD file and afterwards downloads and displays the segments in a quality dictated by the client's internal adaptation strategy. The adaptation strategy considers a combination of parameters, like the client's device screen size, user preferences, measured throughput or current buffer level, to decide which quality level to choose for which segment. The objective of the adaptation strategy is to maximize the Quality of Experience (QoE) of the user. It does so by downloading the best possible quality while at the same time avoiding playback interruptions due to an empty playback buffer (stallings) and frequent changes in quality.

In video coding, a sequence of pictures is encoded by replacing some of the pictures with motion vectors which only describe the changes compared to the previous picture.

An encoded picture which contains the full original picture is called an I-frame, while the motion vectors are denoted as P- and B-frames. In DASH, a segment is self-contained and therefore has to start with an I-frame, followed by multiple I-, P-, or B-frames. While the DASH standard allows fixed or variable segment durations [4], current implementations utilize a fixed [3] segment duration, usually between 2 and 10 seconds. As shorter the segment duration, more of the expensive I-frames are required. I-frames are also required when changes between frames can not be expressed as motion vectors, e.g. due to scene changes.

In this paper we investigate the impact of video coding and content segmentation on the performance of DASH. Instead of setting a fixed segment duration and forcing the encoder to place expensive I-frames, we let the encoder determine the best I-frame placement and adapt the segment duration to the I-frames. This way the content encoding efficiency is increased, resulting in a reduced bitrate per video clip. However, the segment duration can show large variations depending on the content characteristics, which may affect the streaming behavior of DASH heuristics in a negative way. The contribution of this paper is threefold. Firstly, we quantify the reduction of segmentation overhead with variable segment durations compared to fixed segment durations. Secondly, we identify appropriate encoder settings for variable segment durations. Thirdly, we compare the performance of variable segment duration and fixed segment duration on the stalling probability using a generic video buffer model from literature.

The paper is structured as follows. Chapter II presents background and related work. Chapter III discusses the methodology of the evaluation. Chapter IV presents the evaluation results and Chapter V concludes this paper.

II. RELATED WORK AND BACKGROUND

Per-chunk bitrate control for consistent-quality is proposed in [5]. Instead of encoding the whole video with a target quality, the complexity of each video chunk is considered during the encoding process. Using the proposed method, all chunks have similar quality along the video, although the complexity varies among the segments, while preserving a capped bitrate.

The impact of segment durations on encoding efficiency and DASH performance is discussed on a large dataset in [6]. The

impact in mobile networks is evaluated in [7]. The analytical and simulative results advise to use segments of shorter durations, as they support a smooth video playback. Since the HTTP signaling increases with shorter chunks, the authors propose to request several segments with one HTTP-GET. A similar approach is presented in [8], where the HTTP/2 push feature is used to deliver several video chunks after one GET-request. The authors propose to use super-short segments, of less than one second duration in live streaming scenarios. They evaluate the optimal segment duration and number of segments to be pushed by the server after a GET-request as a function of the network RTT.

A method to determine the segment duration for optimal accuracy and speed of rate adaptation is presented in [9]. The goal is to produce a smooth TCP rate, which reflects the current network capacity. Hence, the adaptation algorithm does not solely decide about the next segment's bitrate, but also about the segment's duration. The simulation results show that stalling can be reduced and play back quality can be increased by using the proposed adaptation algorithm compared to state-of-the-art heuristics. In [10], the server, where the source videos are available with different segment durations, decides about the segment duration to be delivered to the clients. The server chooses the segment durations so as to reduce spread in packet interarrival times and segment fetch durations, leading to more accurate throughput estimations.

A segment-duration-aware adaptation heuristic is presented in [11]. Similar to [9], [10] the server provides the same video sequence split in chunks of different durations. Based on network conditions, the heuristic chooses both, the next segment's bitrate and duration. The weakness of this approach comes with the constrained possibilities for switching between different representations, as this is only possible, where GoPs are synchronized.

Content-aware video encoding and its feasibility for mobile video streaming is addressed in [12]. However, instead of segmenting the video in a content-aware fashion, the authors propose to allocate bitrate based on spatial and temporal perceptual information. The representation bitrate during encoding is only increased in case it involves an increased perceived quality. Using the proposed method, the generated data traffic can be reduced, while maintaining a stable video quality.

As shown above, some works consider variable segment durations for HAS. However, the videos are not split in segments of variable lengths, rather are they made available in several representations, whereby each representation considers different fixed segment durations. In contrast [13], considers content-aware encoding and segmentation by taking into account scene-cuts and present I-frames. The authors show that the content-based segmentation process can save up to 10% of bandwidth, without quality degradation. Besides [13], no work is presented by now on segmenting videos for HAS in variable segment durations. In this paper, we evaluate the overhead induced by HAS video segments of strictly the same duration on a large scale dataset. We further compare the

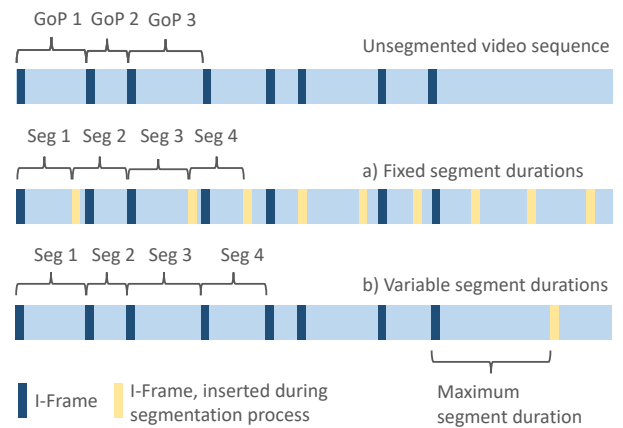


Fig. 1. Fixed segment durations versus variable segment durations, aligned with existing I-frames

stalling probability of variable segment duration streaming with fixed segment duration streaming using a GI/GI/1 model with pq -policy.

III. METHODOLOGY

This section comprises the content preparation, a verification that the content can be used for HAS streaming, and the applied analytical model.

A. Content Preparation

1) *Video Encoding with Variable Segment Durations:* Videos prepared and encoded for online streaming usually consist of several groups of pictures (GoP). Each GoP starts with an I-frame, which contains the complete picture information, followed by several P-frames and B-frames. P-frames and B-frames solely contain information about the differences relative to the I-frame, stored as motion vectors. While I-frames are expensive in terms of Bytes which need to be stored and transmitted, B-frames and P-frames make the video encoding efficient. The placement of I-frames depends on two properties. Firstly, I-frames are needed at scene cuts, where the displayed image usually differs significantly. Secondly, I-frames are inserted in order to reduce errors, once a certain threshold for maximum GoP length is reached.

Conventionally, when preparing and encoding a HAS video, the clip is split into segments of equal length, commonly between 2 and 10 seconds. All of those video segments are self-contained, meaning that they start with an I-frame. Hence, splitting a video into several segments leads to additional overhead due to the extra I-frames, as illustrated in Figure 1.

The upper blue bar represents an encoded, but unsegmented video sequence. I-frames are illustrated as dark blue bars. The middle blue bar shows the same encoded clip, however split into segments of equal durations. Yellow lines indicate the I-frames which have to be inserted additionally at the beginning of each segment. The I-frames which are present in the unsegmented video remain in the video segments.

We propose a variable segmentation approach, as represented by the lowest blue bar. Instead of demanding segments

TABLE I
VIDEO CHARACTERISTICS

Name	Duration (s)	Resolution	Bitrate (kbps)
Big Buck Bunny (bbb)	600	1920x1080	597197
Sita Sings the Blues (sita)	4900	1920x1080	596601
Sintel (sin)	888	4096x1744	2057307
Elephants Dream (ed)	654	1280x720	265421
Tears of Steel (tos)	734	4096x1714	2021917

TABLE II
CONSIDERED CRFs AND SEGMENT DURATIONS

Property	Fix	Variable
(Maximum) segment durations	{0.5:0.5:15}	{1:1:15}
CRF	{17:3:41}	{17:3:41}
Resulting configurations per clip	270	135

of strictly the same duration, we only set a maximum segment duration. The encoder, in our case `ffmpeg`¹, can then dynamically decide where to cut the video and per default aligns the video cuts with the existing I-frames. Consequently, we can save the costly I-frames, as they are only needed if the maximum segment duration would be exceeded.

2) *Video Dataset*: In order to evaluate the potential of the variable approach in terms of reduced segmentation overhead, we create a large data set of segmented and encoded videos. As source material we use five different uncompressed raw video clips which are provided by the Blender Foundation² in the y4m format. All clips have a frame rate of 24 fps, other video characteristics can be found in Table I. Several representations are created for each video clip. The different properties are shown in Table II. For the encoding with fixed length segments, we consider 30 different durations, ranging from 0.5 seconds up to 15 seconds. For the variable approach, the encoder can choose the segment duration between 0 seconds and an upper threshold (maximum duration). We consider for this maximum duration values between 1 second and 15 seconds in steps of 1. In both cases, fixed and variable segmentation, we encode the video with nine different constant rate factors (crf) to obtain nine quality levels. We use as highest quality the FFmpeg recommendation for (nearly) lossless compression with a crf of 17. The crf is decreased in steps of 3, until a crf of 41 (lowest quality) is reached. This results in 270 configurations for the fixed segmentation approach and 135 configurations for the variable segmentation approach. We apply all configurations on all videos and therefore obtain 2025 video representations in total.

3) *Docker Container for Video Encoding*: In order to easily distribute the encoding and evaluation tasks on several compute nodes in a local OpenStack³ Cloud, the tasks are encapsulated within a Docker⁴ container. Each docker instance obtains one task, constituted by the source video id and a combination of the parameters described above. When the video segmentation and encoding is completed, the container determines several characteristics of the resulting video rep-

¹<http://ffmpeg.org/>

²<https://www.blender.org/>

³<https://www.openstack.org/>

⁴<https://www.docker.com/>

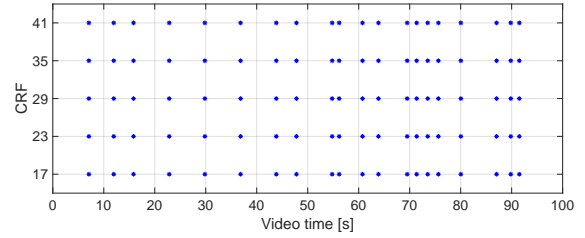


Fig. 2. Resulting segment durations for a variable segmentation between 0 and 7 seconds of the Big Buck Bunny clip

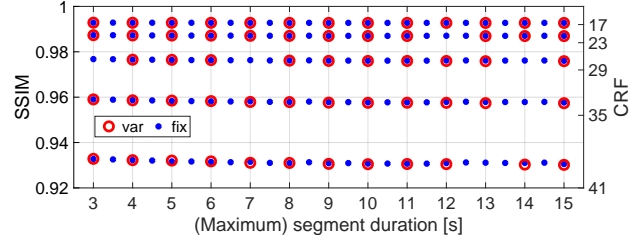


Fig. 3. Resulting SSIM values for several fixed and variable segmentations of the Big Buck Bunny clip

resentation. This includes the video segments' bitrates and segment durations, as well as the respective mean values, standard deviations, minima and maxima. Furthermore, the structural similarity metric (SSIM), statistics and video frame types, as well as the total video file size are evaluated. Besides those encoding- and segmentation-dependent metrics, basic video characteristics like frames per second and resolution are returned as well. The Docker container is publicly available on GitHub.⁵

B. Quality and DASH-feasibility Validation

In order to be able to use videos with variable segment durations for DASH, it must be ensured that the segment structure does not change throughout the different qualities. That means that the k -th segment of quality level l must have the same length as the k -th segments of all other quality levels. Otherwise, it is not possible to switch between the video qualities. Figure 2 exemplary shows for the Big Buck Bunny clip with a maximum segment duration of 7 seconds that this requirement is full-filled. The x-axis shows the video time in seconds, the y-axis denotes the nine different CRF values applied during encoding. The dots represent the cumulative segment duration on each quality layer. As the dots are vertically aligned, the level can be changed after each segment without changing the video position.

The variable and fixed segmentation approach can only be compared with each other in a fair manner if the quality remains the same among different video segmentations of one quality representation. Figure 3 shows the SSIM values resulting from different constant rate factors and segment durations for the Big Buck Bunny clip. For the lower qualities, there is a slight tendency for decreasing SSIM with increasing (maximum) segment durations. Nonetheless, there is no significant difference between the SSIM values resulting from fixed

⁵<https://github.com/linfo3/docker-video-encoding>

TABLE III
MAXIMUM DIFFERENCE AND STANDARD DEVIATION OF SSIM FOR
DIFFERENT SEGMENTATIONS OF EQUAL CRF.

CRF	17	20	23	26	29	32	35	38	41
diff	.0001	.0002	.0003	.0001	.0002	.0010	.0014	.0014	.0023
std	.0001	.0002	.0002	.0004	.0005	.0007	.0008	.0010	.0012

segmentation approach and variable segmentation. Table III shows the maximum difference of SSIM values between representations that differ with regard to the segmentation, but are encoded using the same CRF. Furthermore, the standard deviation of the SSIMs of all segmentations of a specific CRF are shown.

C. Analytical Model for Analyzing DASH Streaming Behavior

A model for investigating the stochastic properties of the buffer level distribution is introduced in [14]. The model is based on a GI/GI/1 queue with pq -policy and allows the computation of relevant performance metrics like the stalling probability, the stalling duration or the buffer utilization based on distributions for the segment duration, the segment size and the bandwidth. The variables p and q represent thresholds for pausing and continuing the video download. They thus enable to keep the buffer level between both thresholds, if enough resources are available. We use the outlined approach in this paper to determine the impact of fixed and variable segmentation on the stalling probabilities. Hereby, we utilize the characteristics of the different encodings of a video clip as input parameters and compute the corresponding performance metrics for specific bandwidth distributions.

IV. EVALUATION

A. Reduction of Average Bitrate

In the following, we compare the average bitrate that results from fixed-duration segmentation with the average bitrate resulting from variable segmentation. The comparison is performed in two different ways. Firstly, the approach with segments of fixed duration x is evaluated against the variable segmentation with a maximum duration of x . In the following, we denote this as *equal-max*. Secondly, the fixed-duration segmentation is compared with the variable segmentation that results in an average duration of (nearly) x . We refer to this comparison as *nearest-average*. Figure 4 shows the relative bitrates required for the variable segmentations compared to the invariant segment durations for the video Elephants Dream, encoded with a constant rate factor of 26. The y-axis shows the invariant segment durations, the x-axis shows the range within which the encoder can decide about the segment durations for variable splitting. Each column has one highlighted box, which identifies the fixed segmentation that is nearest the average duration of variable segmentation, i.e. the nearest-average. For example, a variable segmentation between 0 and 5 seconds results in segments of 4.1957 seconds on average. Hence, in terms of average segment duration, the variable segmentation [0–5] is compared to the segments of strictly 4 seconds length.

The heatmap’s values follow the ratio $\frac{abr_{var}}{abr_{fix}}$, where abr is the average bitrate. Values greater 1 denote that more bits per second are required for encoding a video with segments of variable durations than for a video with a static segment length. We only observe this case if the upper threshold for variable segments is shorter than the fixed duration. For example, a higher bitrate is required for segments that range from zero to three seconds, than for segments with a static length of four seconds. Whereas, bitrate can be economized in all cases where the upper duration threshold of the variable segmentation is larger than the fixed segment length. Most notably is the fact, that along the diagonal from the upper left corner to the lower right corner, all values are below 1. This means that bitrate can always be saved by choosing variable segmentation between 0 and x seconds compared to a static segment duration of x seconds.

The aggregated results for all five videos and all nine qualities are shown in Figure 5. The x-axis shows the relative bitrate required for the variable segments compared to the bitrate of the video with fixed-duration segments. On the y-axis, the CDF is depicted. The dashed lines denote the equal-max comparison. That means we compare the segmentation in video portions of length between 0 and x with the segmentation of fixed duration x .

The variable approach has the highest potential when applied to the Big Buck Bunny video. Its bitrate share compared to fixed segmentation is not exceeding 0.973 and can even reach 0.932. Hence, almost seven percent can be saved. Among the evaluated video clips, the variable approach performs the poorest when applied to the Tears of Steel video. The relative bitrate ranges between 0.998 and 0.978.

The solid lines represent the nearest-average comparison. That means we compare the invariant segmentation with length x with the variable segmentation resulting in an average segment duration of x . For the Big Buck Bunny clip, the relative bitrate share lies between 0.873 and 0.95. In about 50% of the evaluated cases, 10% of bitrate can be saved when segmenting variably, compared to the conventional fixed duration approach. For the Tears of Steel video, the values for relative bitrate range between 0.96 and 0.998.

Please note that the variable approach always reduces the overall bitrate in any of the observed cases. Further, although the encoding complexity is increased, it is still bounded by the maximum segment duration.

B. Stalling Probability

The invariant and variable segmentation approaches influence adaptive video streaming with respect to the buffer filling behavior. While in the first case, the same video portion is added to the playout buffer with each segment arrival, the playtime added to the buffer in the latter case is fluctuating. Furthermore, as the variable-length segments have different durations compared to the fixed-length segments, their download durations differ as well. In the following, we use the model described in Section III-C to evaluate the impact of variable segment durations on the stalling probability.

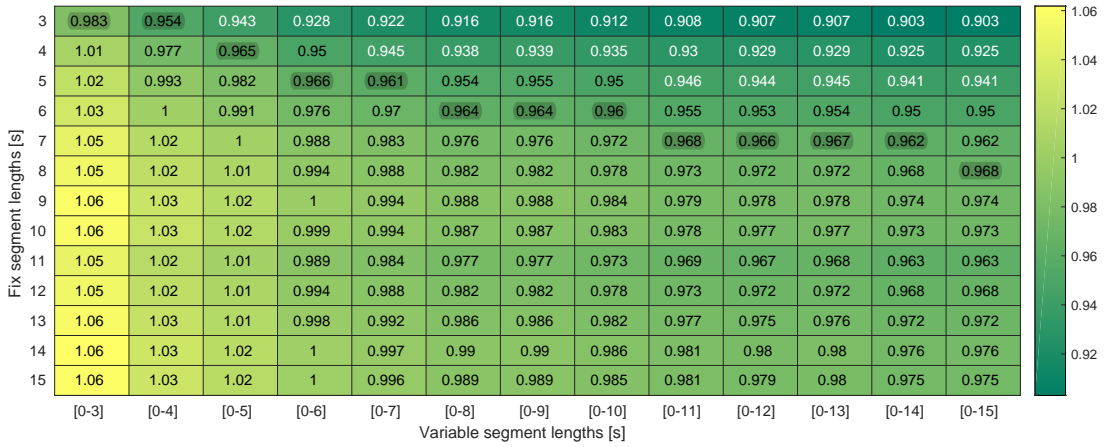


Fig. 4. Relative bitrates needed for variable segmentation for the Elephants Dream video, encoded with a CRF of 26

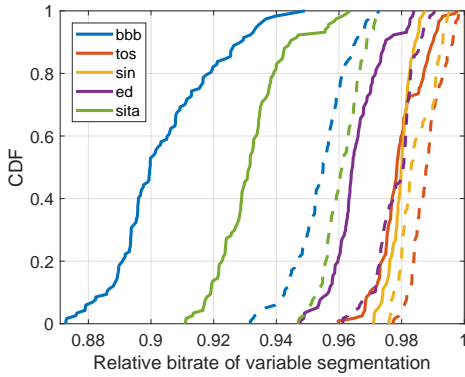


Fig. 5. Relative bitrate required for variable segmentation compared to bitrate required for segments of equal duration. Solid lines indicate a comparison based on nearest-average, dashed lines equal-max, respectively.

When comparing a variable-length video representation with the respective fixed-length video representation, the average bitrate abr_{fix} of the video with invariant segment duration is used as a reference. Using this reference, the average available bandwidth is determined as the $\{0.8 : 0.1 : 1.6\}$ -fold of abr_{fix} . We refer to this ratio as the bandwidth provisioning factor a . Besides different bandwidth provisioning factors, we also consider various values for the coefficient of variation $cvar$ of the available bandwidth. The model's buffer thresholds p and q are set to 30 and 40, respectively.

Figure 6 illustrates the stalling probabilities obtained with the model for a subset of the representations of the Big Buck Bunny video. The x-axis denote the bandwidth provisioning factor a , the y-axis represent the stalling probability. The video was encoded with a constant rate factor of 26. Results are depicted for coefficients of variation $cvar = 0.1$ and $cvar = 0.5$. As shown in Figure 6(a), for $a = 0.8$ and $cvar = 0.5$, the stalling probability can be reduced from 0.37 to 0.27, if the video is variably split into segments between 0 and 4 seconds duration, instead of using segments of strictly 4 seconds. Figure 6(b) shows the results for segments with a (maximum) duration of 10 seconds. For $a = 0.8$ and $cvar = 0.5$, the stalling probability can be reduced by 0.18 with the variable approach. Furthermore, in the depicted

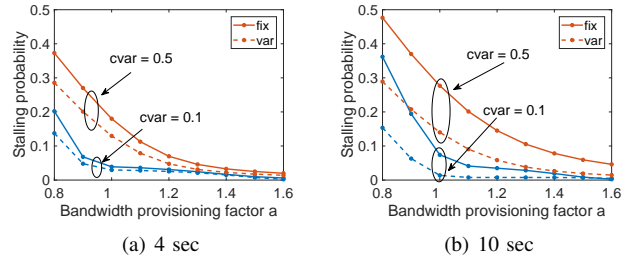


Fig. 6. Stalling probability of variable segmentation and fixed segmentation approach compared to equal-max for the Big Buck Bunny clip encoded with a CRF of 26.

results, the variable approach always performs better.

In order to deeper investigate the potential of stalling probability reduction with variable segments, we apply the model with the following parameters. As above, the bandwidth provisioning factor a ranges from 0.8 to 1.6 in steps of 0.1. The coefficient of variation is set as 0, 0.1, 0.25, and 0.5. For each video, we consider the 9 quality levels and all available video representations in terms of (maximum) segment durations.

Figure 7 presents the results for all investigated videos compared on an equal-max duration basis. The y-axis represents the cumulative distribution function. The x-axis shows the reduction of stalling probability, i.e. $p_{st,fix} - p_{st,var}$, that can be achieved by applying the variable approach compared to fixed segment durations. Negative values denote that the variable approach results in an increase of the stalling probability, positive values denote that less stallings occur when streaming the video with variable-duration segments. The solid black line represents the results of all video clips. In the worst case, the stalling probability is increased by 12.77% when applying the variable approach and in 9.64% of all the observations, the stalling probability is increased by using variable segment durations. However, in 87.97% of all cases, the proposed approach outperforms the fixed duration segmentation and in 12.64%, the stalling probability can even be reduced by at least 10%. In the best case, the stalling probability can be reduced by 39.6%.

The results when comparing on nearest-average basis are

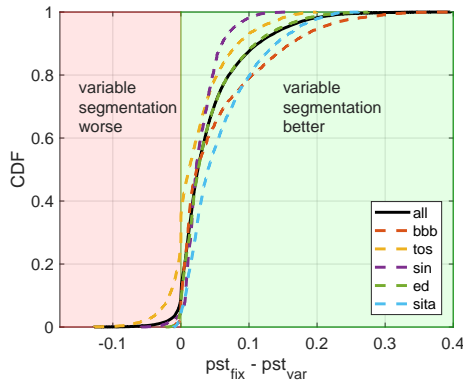


Fig. 7. CDF of stalling probability distance $pst_{fix} - pst_{var}$ for all evaluated video clips and parameter settings, based on equal-max

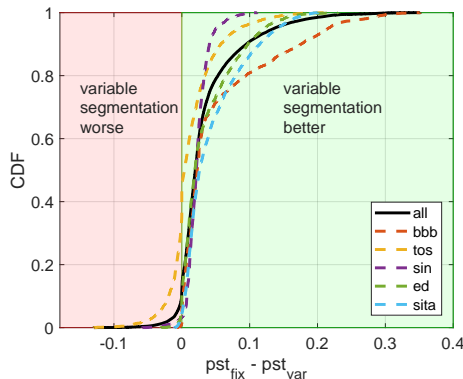


Fig. 8. CDF of stalling probability distance $pst_{fix} - pst_{var}$ for all evaluated video clips and parameter settings, based on nearest-average

shown in Figure 8. The values only slightly differ from the results on max-duration basis. In 10.66% percent of all observations, the conventional approach performs better. In roughly 2.42%, there is no difference between the variable-duration and fixed-duration approach in terms of stalling probability. The proposed approach outperforms the conventional segmentation approach in 86.91%.

V. CONCLUSION

In this paper, we investigated the benefits of variable segment durations on adaptive video streaming. We showed that our proposed approach is more efficient in terms of encoding bitrate compared to the conventional approach with fixed segment durations. Furthermore, we investigated the stalling probability for both approaches using a discrete time analysis model. It turned out that the proposed approach has high potential to reduce video stallings during playback, because the same video quality can be achieved with lower bitrate. As stallings, which are one of the main influence factors of video QoE, can be reduced, the users' satisfaction with video services can be increased without additional costs. In case all potential stallings can be eliminated by the heuristic initiating a switch to lower quality, at least the same quality can be transmitted with less bandwidth consumption.

For future work, we plan to validate the results with simulations and measurements reflecting realistic network conditions. Furthermore, we will investigate whether state-of-the-art HAS heuristics can cope with segments of variable durations. Depending on the results, we plan to design a segment-aware HAS heuristic. Moreover, we will provide further statistics, including the correlation of segment duration and segment size, as well as the distribution of segment durations. We intend to find relationships between those statistics and to estimate the potential of variable video segmentation for a given video by its characteristics, e.g. its spatial and temporal complexity. For that reason, we also plan to extend the dataset and add more videos of different genres and consider several resolutions per video.

ACKNOWLEDGMENT

This work has been supported by the DFG Grant “SDN-enabled Application-aware Network Control Architectures and their Performance Assessment” (ZI1334/2-1, TR257/43-1, KE1863/6-1).

REFERENCES

- [1] C. V. N. Index, “The zettabyte era—trends and analysis,” *Cisco white paper*, 2017.
- [2] T. Stockhammer, “Dynamic adaptive streaming over http—: standards and design principles,” in *Proceedings of the second annual ACM conference on Multimedia systems*. ACM, 2011, pp. 133–144.
- [3] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hossfeld, and P. Tran-Gia, “A survey on quality of experience of http adaptive streaming,” *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 469–492, 2015.
- [4] I. Sodagar, “The mpeg-dash standard for multimedia streaming over the internet,” *IEEE MultiMedia*, vol. 18, no. 4, pp. 62–67, 2011.
- [5] J. De Cock, Z. Li, M. Manohara, and A. Aaron, “Complexity-based consistent-quality encoding in the cloud,” in *Image Processing (ICIP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1484–1488.
- [6] S. Lederer, C. Müller, and C. Timmerer, “Dynamic adaptive streaming over http dataset,” in *Proceedings of the 3rd Multimedia Systems Conference*. ACM, 2012, pp. 89–94.
- [7] Y.-T. Lin, T. Bonald, and S. E. Elayoubi, “Impact of chunk duration on adaptive streaming performance in mobile networks,” in *Wireless Communications and Networking Conference (WCNC), 2016 IEEE*. IEEE, 2016, pp. 1–6.
- [8] J. van der Hoof, S. Petrangeli, T. Wauters, R. Huysegems, T. Bostoen, and F. De Turck, “An http/2 push-based approach for low-latency live streaming with super-short segments,” *Journal of Network and Systems Management*, pp. 1–28, 2017.
- [9] C. Liu, I. Bouazizi, and M. Gabbouj, “Segment duration for rate adaptation of adaptive http streaming,” in *Multimedia and Expo (ICME), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1–4.
- [10] B. J. Villa and P. E. Heegaard, “Group based traffic shaping for adaptive http video streaming by segment duration control,” in *Advanced Information Networking and Applications (AINA), 2013 IEEE 27th International Conference on*. IEEE, 2013, pp. 830–837.
- [11] I. Irondi, Q. Wang, and C. Grecos, “Optimized adaptation algorithm for hevc/h. 265 dynamic adaptive streaming over http using variable segment duration,” in *Real-Time Image and Video Processing 2016*, vol. 9897. International Society for Optics and Photonics, 2016, p. 98970O.
- [12] S. Wilk, D. Stohr, and W. Effelsberg, “A content-aware video adaptation service to support mobile video,” *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 12, no. 5s, p. 82, 2016.
- [13] V. Adzic, H. Kalva, and B. Furht, “Optimizing video encoding for adaptive streaming over http,” *IEEE Transactions on Consumer Electronics*, vol. 58, no. 2, 2012.
- [14] V. Burger, T. Zinner, L. Dinh-Xuan, F. Wamser, and P. Tran-Gia, “A generic approach to video buffer modeling using discrete-time analysis,” *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 2018, preprint on webpage goo.gl/UB6XL8.