

Content Aware Segment Length Optimization for Adaptive Streaming over HTTP

Ondrej ZACH, Martin SLANINA

Dept. of Radio Electronics, Brno University of Technology, Technicka 12, 616 00 Brno, Czech Republic

ondrej.zach@phd.feec.vutbr.cz, slaninam@feec.vutbr.cz

Submitted December 20, 2017 / Accepted April 26, 2018

Abstract. *HTTP adaptive streaming is a widely used method for delivering the video content to its final recipients. The visual quality of the streamed video content is being adaptively changed according to network conditions to offer the user a smooth playback, which is even more important for mobile connection like LTE. In this paper, we focus on the encoding of the video content and on the segmentation of the video to be used in DASH based service. We used long sequences with duration up to 2.5 hours to simulate a real life situation. We investigate the influence of the GOP length on the final DASH segment size and evaluate the performance of AVC and HEVC when used in DASH. We used several fixed values of GOP length and one special case of scene change based GOP creation. Our results showed, that such an adaptive segmentation mode brings up to 11% bitrate savings with preserving comparable quality and lower fluctuations in absolute size of the DASH segments.*

Keywords

Adaptive segmentation, GOP, group of pictures, H.264/AVC, H.265/HEVC, HTTP adaptive streaming, PSNR, video encoding

1. Introduction

As in 2017, the video content already represents the majority of the world's overall IP traffic and according to forecast in [1], its share is expected to grow up to 82 % by 2021. Moreover, the traffic generated by Video-on-Demand services (VoD) is expected to be doubled in 2021 as compared with 2016 according to the same study. Hence, delivering the video content to its the viewers efficiently is a crucial task.

Video streaming services put great demands on the quality of the broadband connection of the viewers. Currently, the H.264/Advanced Video Coding (AVC) standard is widely used to encode the video content available online. As the users tend to require better visual quality (high resolution, video with high frame rate), this leads to the need of us-

ing either increased bitrate to encode the video content or employing more efficient video encoding algorithms such as H.265/High Efficiency Video Coding (HEVC).

Other issue that has to be taken into account when considering video streaming are variable properties of the network connection of the user. This is even more important for mobile connections like LTE where the bitrate and latency of connection attainable by the user depend on various conditions, e.g. the distance from the eNodeB, channel interference, number of users in a cell, scheduling etc. [2]. In case when the video would be streamed at a constant bitrate, these fluctuations tend to cause impairments like stalling, which is generally perceived as annoying by the users, [3]. To overcome this, methods of adaptive streaming are widely used.

In this paper, we focus on evaluation and analysis of different video encoding strategies for use in video streaming under variable conditions as video encoding has a significant impact on the overall operation of HAS, including overhead, buffering etc.

The remainder of the paper is organized as follows. Section 2 presents current state of the art and related work. In Sec. 3, the proposed experiment is described and the results are discussed. Finally, Section 4 concludes the paper.

2. State of the Art

In this section, the currently used methods for creating content for adaptive streaming are introduced, as well as the related work.

2.1 High Efficiency Video Coding

Currently, the majority of the video content online is encoded using AVC standard, [4]. AVC performs very well in HD (720p) and Full HD (1080p) scenarios. However, for upcoming challenges regarding Ultra High Definition (UHD) its performance (namely quality vs bitrate ratio) might not be sufficient. To face these challenges, a successor of AVC has been created - the High Efficiency Video Coding (H.265/HEVC), [5].

HEVC builds on the same principles as AVC but improvements in specific coding tools bring bitrate savings up to 50% with preserving comparable perceived quality, [6]. These improvements include allowing for bigger size of the basic processing unit (up to 64x64 pixels), adding more directional modes for intra-picture prediction (33 compared to previous 8 in AVC) or improved Content-Adaptive Binary Arithmetic Coding (CABAC) to name a few. Furthermore, Sample Adaptive Offset filter (SAO) is introduced for better reconstruction of the amplitudes of the signal, [8].

2.2 HTTP Adaptive Streaming

The quality of the link between the server storing the multimedia content and its user may vary over time. As a consequence, streaming of the content at fixed bitrate may not usually bring acceptable results regarding the final Quality of Experience (QoE) as perceived by the users (e.g. perceivable stalling effects). Therefore, an adaptive system of streaming the content to the user is often implemented to combat channel variability. HTTP Adaptive Streaming (HAS) is such a technology. In HAS, the multimedia content on the server is stored at several bitrate levels and split into segments of short duration. The client then measures the current throughput and adaptively requests the segments of the content at corresponding bitrate level to avoid stalling and to offer the best quality the user is able to receive at that moment.

There are several main implementations of HAS available:

- HTTP Live Streaming (HLS) by Apple, [9],
- HTTP Dynamic Streaming (HDS) by Adobe, [10],
- Silverlight Smooth Streaming (MSS) by Microsoft, [11],
- MPEG Dynamic Adaptive Streaming over HTTP (DASH), [12].

These implementations vary in supported codecs for audio and video, in supported length of the segments etc, as can be seen in Tab. 1. In the work presented in this paper, we employ MPEG DASH as it is a standardized solution and offers the widest variety of different settings.

	HLS	HDS	MSS	DASH
Segment duration	10 s	2–8 s	2 s	any
Supported video codecs	H.264	H.264, VC-1	H.264, VP6	any
Supported audio codecs	AAC, AC3, MP3	AAC, MP3	AAC, WMA	any

Tab. 1. Selected parameters of HAS implementations, based on [13].

2.3 Dynamic Adaptive Streaming over HTTP

Dynamic Adaptive Streaming over HTTP (DASH) is a solution for HAS standardized by Motion Pictures Expert Group (MPEG). Compared to other proprietary solutions,

DASH offers much more freedom in choosing different settings to tailor its performance and demands to specific scenarios (VoD, live streaming etc.). In DASH, the multimedia content is split into segments. These segments can have either fixed or variable length and can be stored either in individual files or in a single fragmented file, [13].

The structure of the multimedia content is described by the Media Presentation Description (MPD) file. This file contains all information about the given content: length of the sequence, video resolution, used video codec(s) and bitrates, information about audio track(s) and eventually subtitle track(s). Furthermore, based on a specific method of creation of the MPD, either the concrete URLs to the segments or a naming scheme is described. An example of such an MPD file can be seen in Listing 1. We can see one representation (quality level) of the video content, the duration of the video content is 1:50:11 and the codec used to encode the video is H.265/HEVC. In this case, the MPD file contains the information about the naming convention of the segments, the segments have length of 2 s and the audio track is not present.

```
<?xml version="1.0"?>
<!-- MPD file Generated with GPAC
version 0.7.0-rev0-gbd5c9af-master
at 2017-10-09T14:58:12.753Z-->
<MPD
  xmlns="urn:mpeg:dash:schema:mpd:2011"
  minBufferTime="PT1.500S"
  type="static"
  mediaPresentationDuration="PT1H50M11.958S"
  maxSegmentDuration="PT0H0M2.042S"
  profiles="urn:mpeg:dash:profile:full:2011">
  <Period duration="PT1H50M11.958S">
  <AdaptationSet
    segmentAlignment="true"
    maxWidth="1920"
    maxHeight="804"
    maxFrameRate="24"
    par="1920:804"
    lang="und">
  <SegmentTemplate
    timescale="24000"
    media="video_${Number}.m4s"
    startNumber="1"
    duration="48000"
    initialization="video_init.mp4"/>
  <Representation
    id="1"
    mimeType="video/mp4"
    codecs="hvc1.1.6.L120.90"
    width="1920"
    height="804"
    frameRate="24"
    sar="1:1"
    startWithSAP="3"
    bandwidth="579714">
  </Representation >
  </AdaptationSet >
  </Period >
  </MPD>
```

Listing 1. An example of an MPD file

2.4 Related Work

The methods of HTTP adaptive streaming serve as a de facto standard of video streaming techniques nowadays. Therefore, they have drawn significant attention of researchers. In [14] for instance, the authors present QoE metrics in 3GPP DASH when streaming video using an LTE network. The authors of [15] evaluated the influence of user activities during the playback on the final QoE.

The issue of segment length in HAS has been investigated in [16], [17]. The authors of [16] investigated three different settings and their influence on bandwidth utilization and CPU consumption. However, they used one video content with duration of 3 minutes only and no additional information about the video encoding such as bitrate or target quality was provided. Authors of [17] proposed an experiment to quantify the effect of segment length on the network behavior, e.g. network workload, the influence of RTT etc. The authors evaluated three different segment lengths used id different adaptation algorithms. However, the authors focused on the network performance only and no information about the quality of the video playback is offered. The study in [18] describes an enhanced algorithm for bitrate adaptation to offer a smoother video playback in a case of variable segment size (in KB). However, the issue of the video encoding is neglected in this study.

In this paper, we evaluate the effect of the different video encoding settings on the overall performance of an HAS service. Similar study was performed in [19]. However, the authors used short sequences and AVC as the only video encoding algorithm. In our study, we use long video sequences with duration up to 2.5 hours to mimic a real life situation. To best of our knowledge, this is the first study on the influence of the video encoding on the performance of HAS employing long video sequences.

3. Study Description and Results

In our experiment, four different video contents were used for encoding. These contents were acquired from a high quality blu-ray source and represent the most common movie genres. The *Action* content was a combination of live action movie with computer-generated imagery (CGI), *Drama* content was a pure live action film, *Cartoon* content was created using CGI only and, finally, *Musical* content was a recording of a stage play. As our intention is to simulate the real life situation, we use the complete movies. The corresponding parameters of the movies are shown in Tab. 2. The duration varied from approximately 90 minutes in the case of the *Cartoon* up to 150 minutes for *Musical*. The movie contents were used for encoding purposes and only the data created during the encoding process were further analyzed and hence the decoded video contents were not used for any screening purposes afterwards.

Content type	Duration	Frame rate [fps]	Resolution
Action	1:56:38	23.97	1920x792
Cartoon	1:30:58	23.97	1920x1036
Drama	1:50:12	24	1920x804
Musical	2:32:35	23.97	1280x720

Tab. 2. Characteristics of used test sequences.

3.1 Encoding

The video content was encoded compliant to Advanced Video Coding and High Efficiency Video Coding standards, as described in Sec. 2.1. Both AVC and HEVC include JM¹ and HM² reference encoder implementations, respectively. In this experiment, however, we used the x264³ and x265⁴ implementations in order to create the encoded video files. Compared to the reference implementations, x264 and x265 bring better encoding speed and wide variety of settings at the cost of slightly lower visual quality, [7], [8]. In proposed experiment, only the video data were considered and hence, no audio track was used. Furthermore, in a typical DASH scenario, the video content is encoded at several target quality levels (bitrate levels). As we want to analyze the properties of the segments only, we use only one quality representation for each content and codec.

As the ratio of the quality and bitrate needed to encode the sequence is highly dependent on the video content, we did not use any fixed bitrate values. Instead of that, Constant Rate Factor (CRF) was used to produce the video content with comparable visual quality and the specific bitrate values to encode the source sequences were decided by the encoders. Based on a short pre-test, the value CRF = 30 was used to offer transparent visual quality. Furthermore, each video content was encoded using several lengths of the Group of Pictures (GOP), which corresponds to the distance between two consecutive intra-coded frames. The value of GOP length is defined in the number of frames and the proposed scenarios used were as described in Tab. 3.

We propose 4 GOP creation methods in total, 3 of which with fixed lengths of 48, 120 and 168 frames. This corresponds to 2 s, 5 s and 7 s for our case of videos with 24 frames per second. These values were chosen based on the typical segment lengths in currently used HAS implementations as depicted in Tab. 1.

	Mode	GOP length	Duration [s]
1	fixed	48	2
2		120	5
3		168	7
4	adaptive	24–168	1–7

Tab. 3. Description of proposed GOP creation scenarios.

¹<http://iphome.hhi.de/suehring/tml/>

²<https://hevc.hhi.fraunhofer.de/>

³<https://www.videolan.org/developers/x264.html>

⁴<http://x265.org/>

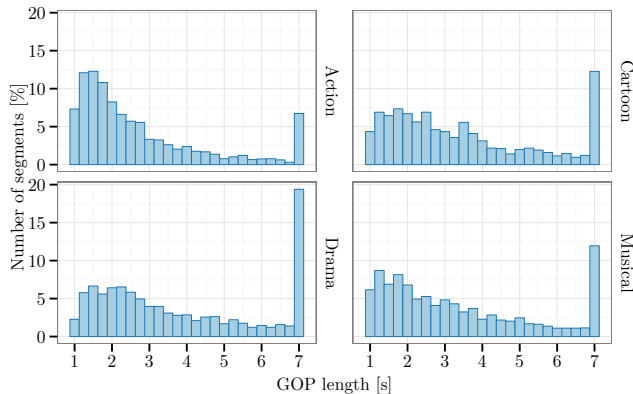


Fig. 1. Distribution of the GOP lengths for used source sequences in the case of adaptive GOP length (scenario 4).

In addition, one special case was an adaptive mode of GOP creation with variable length of GOP based on the scene change in the video. Such an approach may lead to bitrate savings as the prediction used to encode the video frames can be more efficient [19]. In this scenario, the GOP length was allowed to be in range from 24 frames to 168 frames corresponding to 1–7 seconds. The decision if the specific frame belongs to currently encoded scene or is a part of the next scene was based on the computation of Sum of Absolute Transform Differences (SATD) directly by the encoder(s). SATD is a method used by common video encoders to offer decision performance comparable to rate-distortion optimization at much lower computational requirements, [20]. In Fig. 1 the visualization of the GOP lengths based on scene change detection as in scenario 4 can be seen. The horizontal axis depicts the GOP length expressed in seconds while the vertical axis shows the percentage of corresponding segments with given GOP length. It can be seen, that the distribution of the GOP lengths is dependent on the specific content. Where for example in the case of *Action* content the majority of the GOPs had the length lower than 2 seconds, for *Drama* content, the encoders used the longest GOP allowed most often.

Figure 2 shows the bitrate values achieved by the encoders. On the horizontal axis, the different lengths of GOP are shown. The adaptive setting of GOP creation is denoted as *adapt.*. The values are grouped by the content and the colors of the bars represent AVC and HEVC, respectively. Several findings can be seen. Firstly, HEVC needs approximately 50% of the bitrate needed by AVC. This is in accordance with [6]. However, for this study, a more interesting result can be seen. The bitrate is dependent on the GOP length and is usually lower for longer GOPs. This is more clearly illustrated in Fig. 3. This plot shows the bitrate savings compared to GOP length of 48 frames (2 s)⁵. It can be seen, that based on the content, the bitrate savings for longer GOP lengths can reach up to 11%.

Based on this, video sequences encoded using longer GOP lengths might be more appropriate for use in an HAS service in case higher encoding rate - distortion efficiency is demanded.

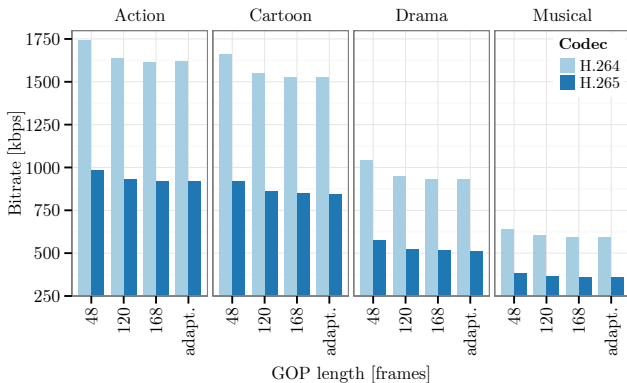


Fig. 2. Bitrate values used by the encoders for encoding the tested video contents, fixed Constant Rate Factor CRF=30.

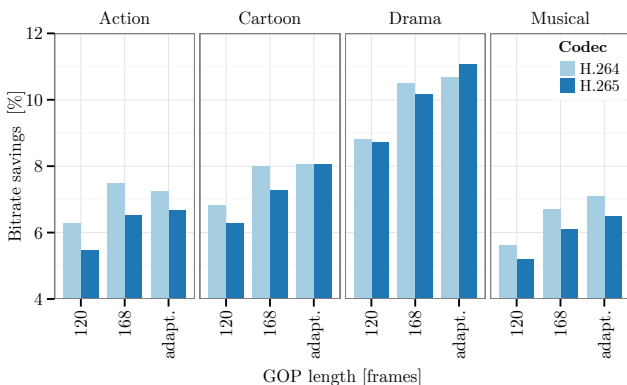


Fig. 3. Bitrate savings for tested video contents compared to case with GOP length of 48 frames.

In a real situation, the visual quality of the video content plays also an important role. Therefore, directly during the encoding process, we also computed the Peak Signal to Noise Ratio (PSNR) of the processed video sequences. PSNR is only a simple metric computed based on per pixel differences between the original and processed image but it offers a basic insight in the visual quality of the video sequence, [21]. As we used the PSNR only to compare the quality among video sequences with the same content and the same video codec, we did not use any other more sophisticated video quality metric, as the PSNR has been proven to offer reliable performance in such scenarios, [22].

Figure 4 shows the mean value of the per-frame computed PSNR of the luminance component for the used contents and respective GOP lengths. From the first look it can be seen, that the quality varies based on the video content but is in range {40, 43} dB. These values correspond to the video quality rated as good by the consumers and a non-expert viewer is usually not able to tell the difference between the original and the processed image. The situation can be seen more clearly in Fig. 5. Here, the bars represent the difference of the mean PSNR compared to case with GOP length of 48 frames. From the first view the mean PSNR is usually lower for all the other GOP settings. However, the absolute value of the difference is 0.2 dB on average, which is practically

⁵We use the case of 2 s long segments as a ground truth as this segment length is supported by majority of the implementations.

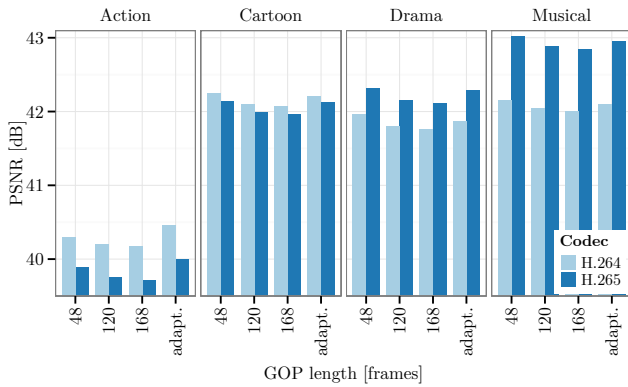


Fig. 4. Objectively measured PSNR for tested video contents and used GOP lengths. Bars represent the mean PSNR computed from the per frame PSNR values of the whole video sequences.

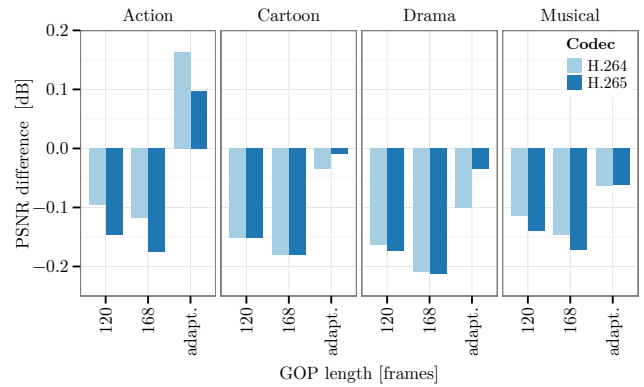


Fig. 5. The difference of the mean PSNR of the tested video contents compared to case with GOP length of 48 frames.

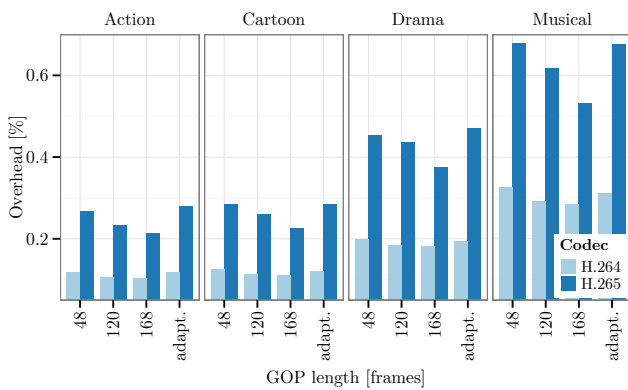


Fig. 6. Relative size overhead of the DASH segmented content, compared to the size of corresponding raw encoded video data without media container.

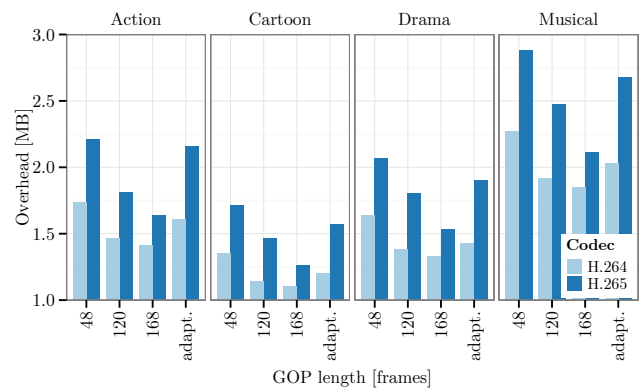


Fig. 7. Absolute size overhead of the DASH segmented content, compared to the size of corresponding raw encoded video data without media container.

imperceptible. Although there can be both highly distorted frames and frames without any degradation of the quality in the whole sequence, we use the mean as the pooling method, as it offers good correlation with subjective scores, [23]. Moreover, when we analyzed the per-frame difference of the PSNR, in approximately 98% of the frames the difference has not exceeded 1 dB. Based on that, there is not any perceivable degradation in the visual quality.

3.2 Segmentation

Tested video sequences were further processed to create files usable in DASH. In order to do so, the video data had to be split into segments and an MPD file had to be created. For this purpose, we used the mp4box tool available in the GPAC package⁶. This tool offers manipulation with the MP4 multimedia container as well as creation of content complying to DASH standard. The segments were created in order to follow the GOP lengths used for encoding the video content, hence the lengths of the segments corresponded to 2 s, 5 s and 7 s. Again, the special case of variable GOP length was used. In this scenario, the length of the segments was adjusted to follow the position of I frames in the video sequence. Hence,

the length of the segments varied over the duration of the sequence as well, as described in Sec. 3.1.

Dividing the video content into segments brings additional overhead which is necessary for correct operation of DASH system. This overhead increases with number of segments. Figure 6 displays the relative size overhead of a DASH ready video content compared to the raw encoded video content (.264 and .hevc files without any multimedia container). The color of the bars represents different codecs and the vertical axis depicts the relative size overhead in percent. We can see, that in our experiment, the value of relative overhead is in the range of {0.1,0.6}%. The overhead is slightly lower for longer segment durations. The highest value can be seen for the *Musical* content, which is caused by its highest total number of segments (total duration 152 minutes). Data used to plot this figure come from an analysis of the segment sizes. The only exception is the case of adaptive GOP length and HEVC codec. For this combination, the mp4box tool is not able to correctly create the segments to start with an I frame and the data used to plot corresponding bars are an estimation based on numbers of I frames and average overhead per one segment.

⁶<https://gpac.wp.imt.fr/>

Similarly, in Fig. 7, the absolute overhead expressed in megabytes is depicted. Again, the value for adaptive GOP length with HEVC codec is only an estimation. It can be seen, that average overhead for a DASH ready video content is approximately 2 MB⁷. This is slightly lower in the case of all AVC encoded video segments. However, the total overhead plays only a minor role in the final performance of an HAS service. For a smooth playback, the question of absolute size of the segments is more crucial. In both Fig. 6 and Fig. 7, the size of the MPD file is not considered.

The plot in Fig. 8 shows the variation of the size of the segments over time. The vertical axis is common for all the subplots and depicts the segment size in megabytes while the horizontal axis is separate for each subplot and represents the duration of the sequence in minutes. Furthermore,

the light green line and dark green line show the mean value of the segment size for AVC and HEVC encoded sequences, respectively. This plot is merely intended to show the variation of the segment size during the play back time and not the absolute value for specific segment duration, hence, only the data for the 2 s long segments were used (scenario 1), as the shape of the respective plots would be similar in case of the rest of the scenarios.

Several outcomes can be seen. Although the duration of each of the segments in time is set as constant for all the cases, the size of the segments is highly dependent on the exact position of the segment in the video sequence, and, of course, on the content. Based on that, we can also see, that the contents used have different characteristics and hence represent contents with various temporal properties.

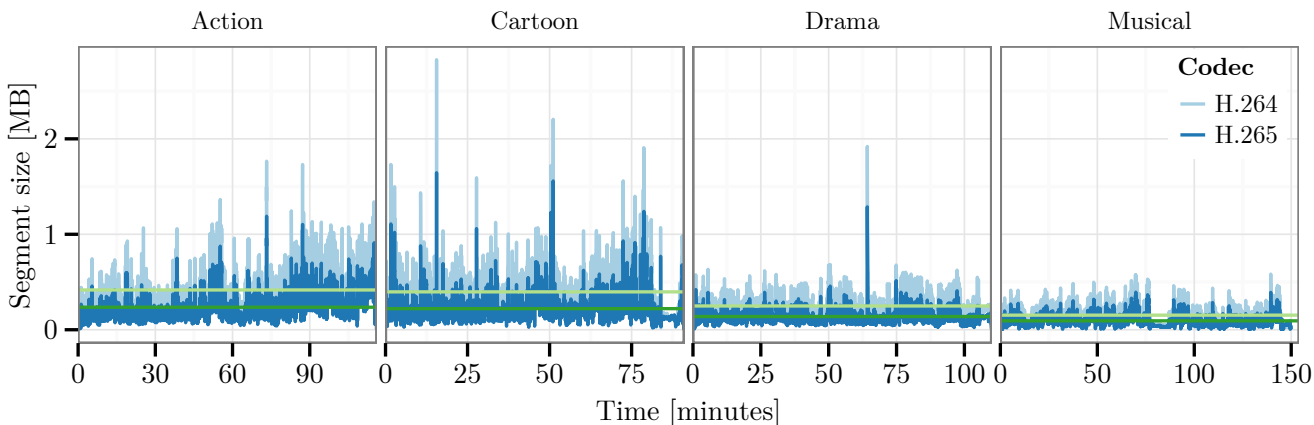


Fig. 8. Time variation of the size of the segments. The light green and dark green lines represent the mean value of the segment size for AVC and HEVC encoded content, respectively. Data used to create the plot correspond to segments with duration of 2 s (scenario 1).

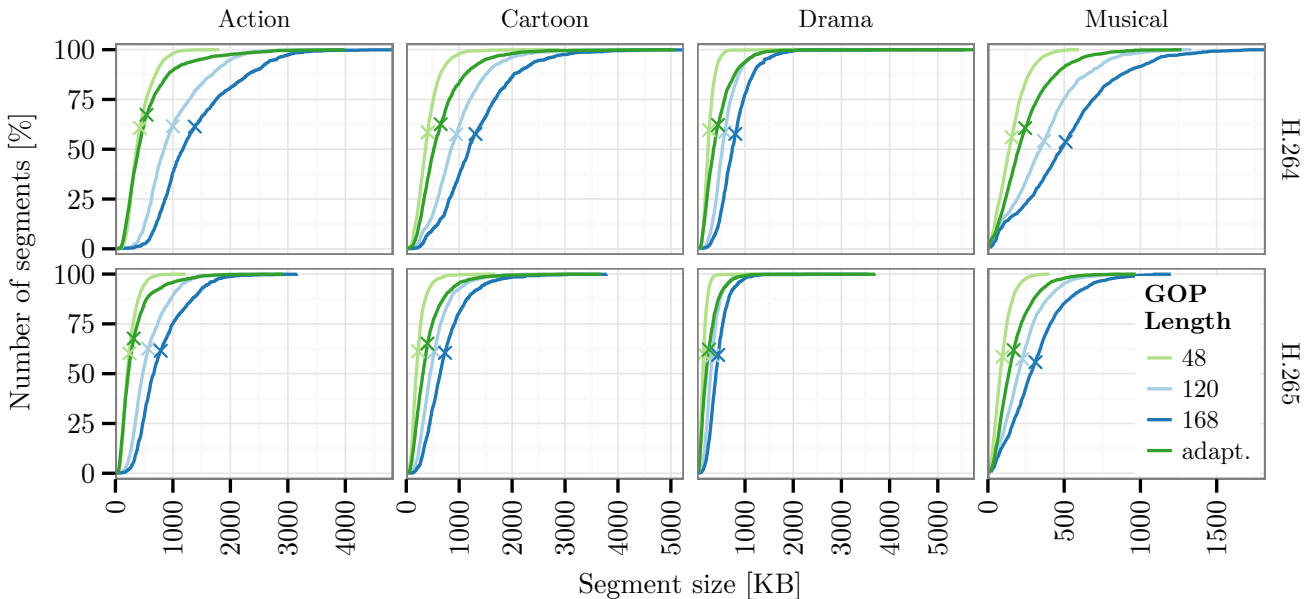


Fig. 9. The CDF of segment sizes, here represented in a percentage of the number of the segments. Different colors represent given GOP modes, which also correspond to DASH segment duration. The cross at the lines represents the mean value of the segment size for the specific combination of mode, codec and content. The width of the bin to compute the CDF was set to 5 KB.

⁷Throughout this paper, 1 MB is understood as 1024 KB and similarly, 1 KB represents 1024 B.

Where as for the content denoted as *Drama* the segment size varies only moderately (with one exception), for the *Action* and *Cartoon* contents, the changes of the absolute segment size are more frequent. As the variations of the segment size can highly influence the buffer occupancy, and hence, the overall experience of a user with HAS based service, further analysis was performed.

In Fig. 9, the Cumulative Distribution Function (CDF) of the segment sizes is depicted. The data are grouped by content and codec, the different colors represent then the specific GOP length (~ segment duration). The horizontal axis shows the segment size and finally, the vertical axis presents the CDF, here in a form of percentage of the number of the segments. The data to plot the lines corresponding to case of adaptive segment length and HEVC codec is, again, an estimate based on the sizes of frames in the GOPs as direct segmentation using the mp4box tool was not feasible.

From the plots, we can see several results. As expected, in the case of the 2 s long segments, the segments have the lowest size. Surprisingly, the case of the adaptive segment duration shows similar behavior until the mean value of the segment size is reached. It can be also seen, that in the adaptive mode, the segment sizes are lower for the majority of the segments compared to 5 s and 7 s modes. Furthermore, in adaptive mode, the percentage of segments with size lower than or equal to the mean value is always the highest among the tested modes as can be seen in Tab. 4.

For a smooth playback, however, the absolute size of the segments is not the most crucial aspect. The most important is the ratio of segment size to duration of the segment. We call this ratio "Segment bitrate" and it can be defined as in (1).

$$B_{\text{seg}} = \frac{\text{Segment size [Kb]}}{\text{Duration [s]}} \quad (1)$$

The segment bitrate represents the minimal throughput of the connection of the user to ensure smooth playback with constant buffer size. In Fig. 10, the maximal segment bitrate is shown. The horizontal axis represents the specific GOP lengths (corresponding to segment lengths) while on the vertical axis, the the maximal bitrate as a ratio to corresponding encoding bitrate value is depicted.

It can be seen, that the maximal value can reach up to 9times the encoding bitrate. However, segments with such high segment bitrate represent only a small percentage of all segments. The results of further analysis showed, that in majority of the cases (from 87% up to 98% segments), the segment bitrate reaches no more than two times the encoding bitrate. However, using longer GOP lengths resulted in slightly better performance.

In a HAS based service, the buffering algorithm plays also an important role. Correct buffering assures a smooth and undisturbed playback and prevents stalling events. There are many buffering algorithms, however, this a very complex issue which is beyond the scope of this paper.

	Content	GOP mode			
		48	120	168	adapt.
AVC	Action	62	61	61	67
	Cartoon	57	58	58	63
	Drama	59	58	60	62
	Musical	54	54	56	61
HEVC	Action	62	61	60	68
	Cartoon	61	60	61	65
	Drama	59	59	60	62
	Musical	57	56	59	62

Tab. 4. The percentage of segments with size lower than or equal to the average size of the segment. The highest value is typed in bold face.

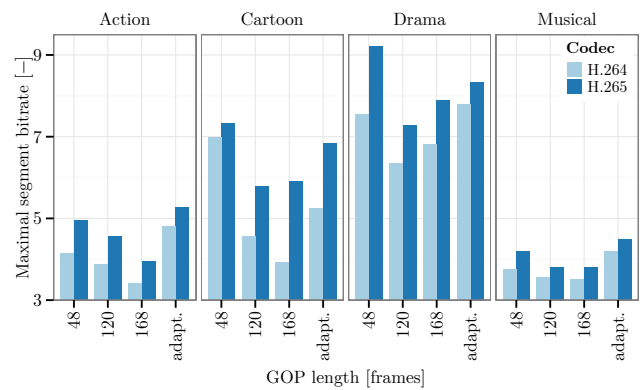


Fig. 10. Maximal segment bitrate, here represented by a ratio of maximal value of segment bitrate to the average encoding bitrate as in Fig. 2.

4. Conclusion

In this work, we investigated the influence of GOP length and corresponding DASH ready segment length on the overall performance of a DASH based video service. We used real life sequences with duration up to 2.5 hours to simulate a real situation. We found out, that although shorter segments (2 s) bring better performance regarding the low segment size and the variability of the segment size, longer segments offer up to 11% saving in the bitrate needed to encode the video content. A specific case of variable GOP length was also studied. This mode offers quality comparable to other modes while preserving the benefits of significant bitrate savings compared to shorter segments. Furthermore, in this case, the variability of the segment size was lower compared to longer segments. Further research can be carried in influence of the peak segment size on the buffering of the segments and corresponding smoothness of the playback. The analysis of performance of video codecs confirmed, that according to [6], HEVC can bring bitrate savings up to 50%, however, with slightly increased overhead when used in DASH.

Acknowledgments

Research described in this paper was financed by Czech Ministry of Education in frame of National Sustainability Program under grant LO1401. For research, infrastructure of the SIX Center was used.

References

- [1] Cisco. *The Zettabyte Era: Trends and Analysis*. 32 pages. [Online] Cited 2017-09-12. Available at <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.html>
- [2] POLAK, L., KALLER, O., KLOZAR, L., et al. Coexistence between DVB-T/ T2 and LTE standards in common frequency bands. *Wireless Personal Communications*, 2016, vol. 88, no. 3, p. 669–684. DOI:10.1007/s11277-016-3191-2
- [3] HOSSFELD, T., SEUFERT, M., HIRTH, M., et al. Quantification of YouTube QoE via crowdsourcing. In *Proceedings of IEEE International Symposium on Multimedia*. Dana Point (USA), 2011, p. 494–499. DOI: 10.1109/ISM.2011.87
- [4] ITU-T Rec. H.264. *Advanced Video Coding for Generic Audiovisual Services*. ITU, Geneva (Switzerland), 2014.
- [5] ITU-T Rec. H.265. *High Efficiency Video Coding*. ITU, Geneva (Switzerland), 2016.
- [6] SULLIVAN, G., OHM, J., HAN, W. J., WIEGAND, T. Overview of the high efficiency video coding (HEVC) Standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 2013, vol. 22, no. 12, p. 1649–1668. DOI: 10.1109/TCSVT.2012.2221191
- [7] RAO, K. R., KIM, D. N., HWAN, J. J. *Video Coding Standards*. London (UK): Springer, 2013. ISBN 978-94-007-6741-6
- [8] ZACH, O., SLANINA, M. A comparison of H.265/ HEVC Implementations. In *Proceedings of ELMAR*. Zagreb (Croatia), 2014, p. 143–146. DOI: 10.1109/ELMAR.2014.6923337
- [9] Apple Inc. *HTTP Live Streaming*. [Online] Cited 2017-09-14. Available at <https://developer.apple.com/streaming/>
- [10] Adobe Systems Inc. *Adobe HTTP Dynamic Streaming Specification (Version 3.0)*. [Online] 2017-09-14. Available at <https://www.wimages2.adobe.com/content/dam/Adobe/en/devnet/hds/pdfs/adobe-hds-specification.pdf>
- [11] ZAMBELLI, A. *IIS Smooth Streaming Technical Overview*. [Online] 2017-10-29. Available at http://www.bogotobogo.com/VideoStreaming/Files/iis8/IIS_Smooth_Streaming_Technical_Overview.pdf
- [12] International Standards Organization/International Electrotechnical Commission (ISO/IEC). *23009-1:2012 Information Technology: Dynamic Adaptive Streaming over HTTP (DASH) Part 1: Media Presentation Description and Segment Formats*. 2012.
- [13] SEUFERT, M., EGGER, S., SLANINA, M., et al. A survey on quality of experience of HTTP adaptive streaming. *IEEE Communications Surveys & Tutorials*, 2012, vol. 17, no. 1, p. 469–492. DOI:10.1109/COMST.2014.2360940
- [14] OYMAN, O., SINGH, S. Quality of experience for HTTP adaptive streaming services. *IEEE Communications Magazine*, 2012, vol. 50, no. 4, p. 20–27. DOI: 10.1109/MCOM.2012.6178830
- [15] MOK, P., CHAN, E. W. W., LUO, X., CHANG, R. K. C. Inferring the QoE of HTTP video streaming from user-viewing activities. In *Proceedings of the first ACM SIGCOMM workshop on Measurements up the stack (W-MUST)*. New York (USA), 2011, p. 31–36. DOI: 10.1145/2018602.2018611
- [16] HASSAN, Y. M., HELMY, A., REHAN, M. M. Effect of varying segment size on DASH streaming quality for mobile user. In *Proceedings of the International Conference on Engineering and Technology (ICET)*. Cairo (Egypt), 2014, p. 1–4. DOI: 10.1109/ICEngTech-nol.2014.7016825
- [17] NGUYEN, D. M., TRAN, L. B., LE, H. T., et al. An evaluation of segment duration effects in HTTP adaptive streaming over mobile networks. In *Proceedings of the 2nd National Foundation for Science and Technology Development Conference on Information and Computer Science (NICS)*. Ho Chi Minh (Vietnam), 2015, p. 248–253. DOI: 10.1109/NICS.2015.7302201
- [18] MYONGWOO, K., JIWOON, P., CHUNG, K. Content-aware rate adaptation scheme to improve stability in HTTP Adaptive Streaming. In *Proceedings of the International Conference on Information Networking (ICOIN)*. Da Nang (Vietnam), 2017, p. 401–405. DOI: 10.1109/ICOIN.2017.7899467
- [19] ADZIC, V., KALVA, H., FURHT, B. Optimizing video encoding for adaptive streaming over HTTP. *IEEE Transactions on Consumer Electronics*, 2012, vol. 58, no. 2, p. 397–403. DOI: 10.1109/TCE.2012.6227439
- [20] LEE, Y.-M., SUN, Y.-T., LIN, Y. SATD-based intra mode for H.264/AVC video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 2010, vol. 20, no. 3, p. 463–450. DOI:10.1109/TCSVT.2009.2035853
- [21] EDEN, A. No-reference estimation of the coding PSNR for H.264-coded sequences. *IEEE Transactions on Consumer Electronics*, 2007, vol. 53, no. 2, p. 667–674. DOI:10.1109/TCE.2007.381744
- [22] HUYN-THU, Q., GHANBARI, M. Scope of validity of PSNR in image/video quality assessment. *Electronics Letters*, 2008, vol. 44, no. 13, p. 800–801. DOI: 10.1049/el:20080522
- [23] SEUFERT, M., SLANINA, M., EGGER, S., KOTTKAMP, M. "To pool or not to pool": A comparison of temporal pooling methods for HTTP adaptive video streaming. In *Proceedings of the Fifth International Workshop on Quality of Multimedia Experience (QoMEX)*, Singapore, 2013, p. 52–57. DOI: 10.1109/QoMEX.2013.6603210

About the Authors . . .

Ondrej ZACH was born in Brno, Czech Republic in 1988. He received his Master's degree in Electrical Engineering from Brno University of Technology in 2013. At present he is a Ph.D. student at the Department of Radio Electronics, Brno University of Technology. His field of interest is video technology and video coding.

Martin SLANINA was born in 1982. He received the Master's degree in 2005, the Ph.D. degree in 2009 and Assoc. Prof. in 2016, all in Electronics and Communication from the Brno University of Technology (BUT), Czech Republic. Currently he is an assistant professor at the Department of Radio Electronics, BUT and a researcher at the Center of Sensor, Information and Communication Systems (SIX), BUT. His research interests are in video quality measurement and assessment methodologies, Quality of Experience (QoE) for multimedia in varying transmission conditions and mobile communication systems.