Title	Datasets for AVC (H.264) and HEVC (H.265) evaluation of dynamic adaptive streaming over HTTP (DASH)
Author(s)	Quinlan, Jason J.; Zahran, Ahmed H.; Sreenan, Cormac J.
Publication date	2016-05-10
Original citation	Quinlan, J. J., Zahran, A. H. and Sreenan, C. J. 'Datasets for AVC (H.264) and HEVC (H.265) evaluation of dynamic adaptive streaming over HTTP (DASH)', Proceedings of the 7th International Conference on Multimedia Systems, Klagenfurt, Austria. 2910625: ACM, 1-6. doi:10.1145/2910017.2910625
Type of publication	Conference item
Link to publisher's version	http://dx.doi.org/10.1145/2910017.2910625 Access to the full text of the published version may require a subscription.
Rights	**© Owner/Author ACM 2016. This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in Proceedings of the 7th International Conference on Multimedia Systems, http://dx.doi.org/10.1145/2910017.2910625**
Item downloaded from	http://hdl.handle.net/10468/4755

Datasets for AVC (H.264) and HEVC (H.265) Evaluation of Dynamic Adaptive Streaming over HTTP (DASH)

Jason J. Quinlan
Department of Computer Science
University College Cork, Ireland
j.quinlan@cs.ucc.ie

Ahmed H. Zahran*
Department of Computer Science
University College Cork, Ireland
a.zahran@cs.ucc.ie

Cormac J. Sreenan
Department of Computer Science
University College Cork, Ireland
cjs@cs.ucc.ie

ABSTRACT

In this paper we present datasets for both trace-based simulation and real-time testbed evaluation of Dynamic Adaptive Streaming over HTTP (*DASH*). Our trace-based simulation dataset provides a means of evaluation in frameworks such as NS-2 and NS-3, while our testbed evaluation dataset offers a means of analysing the delivery of content over a physical network and associated adaptation mechanisms at the client. Our datasets are available in both H.264 and H.265 with encoding rates comparative to the representations and resolutions of content distribution providers such as Netflix, Hulu and YouTube.

The goal of our dataset is to provide researchers with a sufficiently large dataset, in both number, and duration, of clips which provides a comparison between both encoding schemes. We provide options for evaluating not only different content and genres, but also the underlying encoding metrics, such as transmission cost, segment distribution (the range of the oscillation of the segment sizes) and associated delivery issues such as jitter and re-buffering. Finally, we also offer our datasets in a header-only compressed format, which allows researchers to download the entire dataset and uncompress locally, thus ensuring that our datasets are accessible both online via remote and local servers.

CCS Concepts

•**Networks** → Public Internet; •**Information systems** → **Multimedia streaming;**

Keywords

AVC, HEVC, Dynamic Adaptive Streaming over HTTP, DASH, Dataset

*This author is also affiliated with the Electronics and Electrical Communications Department of Cairo University, Egypt.

ACM ISBN 978-1-4503-0000-0.

DOI: 10.1145/0000

1. INTRODUCTION

There has been a radical change in how we view video content. We are no longer tied to a device at a fixed location on a specific time and date, but we are now able to stream movies and shows when and where we want. This has led to a significant increase in the utilisation of the data networks of both wired and wireless networks. Video traffic now accounting for over 55% of all mobile data traffic, and is expected to take over 75% by 2020 [1].

Due to variations in the delivery rate of the transmission network and the volume of background traffic from other sources such as web browser and social media, the achievable quality of experience at the device can vary quite considerably. Typically due to video stalls, re-buffering and similar delivery issues. To counteract this issue, progressive download or Adaptive HTTP streaming techniques such as Dynamic Adaptive Streaming over HTTP (*DASH*) [2] have been proposed, and are widely adopted by content provider, such as Netflix, YouTube, Amazon, and Hulu. Adaptive video techniques strive to improve the quality of experience by changing the requested video quality to match the current network conditions.

Generally the content utilised by these delivery providers is copyrighted, while their respective encoding framework is proprietary in nature. Originally content utilised for experimentation would need to be generated which typically would be very time consuming and unique to the specific research. Recently a number of datasets have become available which increases the number of clips available for evaluation. These datasets, which are explained in further detail in section 2, provide a means of permitting researchers access the same content with different profiles and encoding schemes.

To this end, we present DASH datasets for both H.264 (*AVC*) and H.265 (*HEVC*), with encoding rates comparative to the representations and resolutions of well-known content distribution providers. Our datasets also provide different segment sizes including (2-second, 4-second, 6-second, 8-second, and 10-second) for ten- or sixteen-minute videos. Additionally, our H.264 and H.265 datasets include the same twenty three clips from varying genres (action, comedy, documentary, animation, thriller, sci-fi). Hence, our datasets address the lack of variety of genres in existing datasets. Additionally, our HEVC dataset provides low data-rate video suitable for networks with limited bandwidth.

The remainder of the paper is organised as follows. Section 2 presents relevant background on DASH and other datasets. Section 3 presents an overview of our datasets and the rationale for their creation and intended use. Section 4 presents details on the content generation and output dynamics of the datasets, while Section 5 is dedicated to evaluation results. Section 6 describes future plans and the paper concludes in Section 7.

2. BACKGROUND & RELATED DATASETS

Adaptive HTTP streaming techniques such as DASH, provide a means of adapting the quality of the requested video based on the underlying delivery metrics of the transmission medium, typically determined from the delivery rate and the client buffer levels. With DASH, the video clip is split into small pieces, generally known as segments.

Each segment is encoded into a number of different quality representations that are characterised by their resolutions and average encoding rates. This average rate is estimated over all clip segments. It is not uncommon that individual segment rates feature a high variability over time. Typically, the client adapts by selecting the next segment representation based on a pre-specified policy.

Most modern video content is now encoded with either H.264 or H.265. H.264 [3], MPEG-4 Part 10 Advanced Video Coding (MPEG-4 *AVC*), is the default standard for most modern recordings. It is used in all Blu-ray discs, and is widely used in multimedia broadcasts in both terrestrial and satellite cable system. Newer content is now being encoded with H.265 [4], also known as High Efficiency Video Coding (*HEVC*). H.265 provides a means of increasing the overall resolution/quality of the content, typically up to 4K, while improving the compression rate, and ultimately the transmission cost, in comparison to H.264.

The Dynamic Adaptive Streaming over HTTP Dataset [5] is the first of the publicly available DASH datasets. This dataset provides content encoded with H.264. Six clips were encoded with up to twenty representation rates (from 50 kbps to 8Mbps) over five resolutions (from 320x240 to 1920x-1080). The clips are available in six different segment durations, namely 1, 2, 4, 6, 10 and 15 seconds. The content is licensed under the Creative Common By-2 and is free to share and modify.

This dataset was followed by the Ultra High Definition HEVC DASH Data Set [6]. This dataset comprised of a single clip which is encoded in H.265 with three resolutions (from 1280x720 to 3840x2160), thirteen different representation rates (from 1.8Mbps to 18Mbps), two frame rates (30 and 60 fps), five segment durations (2, 4, 6, 10 and 20 seconds) and four DASH profiles. The DASH profiles (main, on demand, live and live with bitstream switching) provide a means of testing different use cases, typically with respect to latency and initial buffering delay. The profiles vary the group of picture (*GOP*) allocation per segment, the hierarchy of the file reference to the initialisation segment and the availability of byte-range indexing.

The availability of DASH content was further expanded by the Scalable Video Coding (*SVC*) Dataset [7]. This dataset of four large and eight small duration clips encoded with H.264 with three resolutions (640x3600 to 1920x1080), twelve representation rates over the base and enhancement layers (from 600kbps to 10.4Mbps), two frame rates (30 and 60 fps), and four different encoded variants which modify the spatial and temporal scalability of the encoded clip.

It is important to note that DASH is not the only video configuration available, with [8] illustrating a study on identifying different codec configurations employed in a large H.264 video database and studying their objective evaluations, and http://www.cdvl.org/ and http://dbq.multimediatech.cz/ are digital video libraries that offer researchers access to repositories of various free video content in numerous configurations.

Table 1: Quality Level Distribution

Resolution	Encoding Rate	Quality Level
320x240	235 kbps	very low quality
384x288	375 kbps	low quality
512x384	560 kbps	VHS-ish quality
512x384	750 kbps	better VHS-ish quality
640x480	1050 kbps	analog TV quality
720x480	1750 kbps	DVD-ish quality
1280x720	2350 kbps	720p low quality
1280x720	3000 kbps	720p high quality
1920x1080	3850 kbps	1080p low quality
1920x1080	4300 kbps	1080p medium quality

3. DATASET OVERVIEW

In this paper, we present datasets for real-time testbed evaluation and trace-based simulation of Dynamic Adaptive Streaming over HTTP (*DASH*) content.

All of our datasets are encoded with both H.264 or H.265. We believe that our dataset is the first such set to offer a comparison between both encoders for the same content. All of our content is available with encoding rates comparative to the representations and resolutions of content distribution providers, such as Netflix [9], Hulu [10] and YouTube. Table 1, modified from Netflix [9], but similar to Hulu and YouTube, illustrates the distribution of the ten representation rates over the seven resolutions, and their relative quality level.

Finally, our datasets are available with five segment durations (2, 4, 6, 8, and 10 seconds) and one DASH profile (full). All two hundred and thirty generated MPD files have been validated with the online DASH validator [1] of "dashif.org" for conformance with ISO/IEC 23009-1 MPEG-DASH.

In this paper we offer three datasets by which streaming can be evaluated using simulation and a real testbed.

1. Our testbed evaluation dataset offers a means of analysing the delivery of content over a physical network and associated adaptation mechanisms at the client. The evaluation of this dataset was undertaken with "MP4Client", a multimedia player from GPAC [11], an open-source multimedia framework used for research and academic purpose. For this dataset content we utilise three well-known animated videos, Big Buck Bunny (*BBB*) - 9 minutes 46 seconds (9:46) in length, Elephant Dreams (*ED*) - (10:54) and Sita Sings the Blues (*SSTB*) - (16:00), which were obtained as 1920x-1080 YUV files from [12].

2. To supplement our testbed evaluation dataset, and to provide clips across a range of genres, not only animated, we created an additional 20 video clips, extracted from High Definition (*HD* - 1920x1080) Blu-Ray content. Thus offering clips with a mixture of fast

[1] http://dashif.org/conformance.html

Table 2: H.264 - Segment Distribution Output for clip 5

Clip_5	seg_Dur	4267	3818	2976	2328	1734	1038	740	552	370	232
x264		1920	1920	1280	1280	720	640	512	512	384	320
		1080	1080	720	720	480	480	384	384	288	240
0		820	820	820	820	820	820	820	820	820	820
1	4	1878426	1645796	1264115	950941	695637	403209	279730	205976	137267	83929
2	4	2445453	2218132	1775801	1424383	1088997	662920	483338	359490	242483	150508

and slow action, and with static and dynamic scenes. All content is extracted ten minutes into the video, so as not to have similar content, i.e. opening credits, at the start of every clip. All clips are sixteen minutes in length.

Due to copyright on the original Blu-ray content, we cannot provide the underlying DASH content, but we provide the encoded segment specifications for each clip in the format shown in Table 2, thus offering a means of evaluating this dataset using a trace-based framework, such as the popular network simulators NS-2 and NS-3 [13]. The evaluation of this dataset was with NS-3. The top row of Table 2 details the clip name and the encoded representation rate per column. Row 2 and 3 present the encoder, H.264 in this instance, and the associated resolution (WxH). Row 4, index 0 (defined by the index in column one), denotes the transmission cost in bytes for the mp4 header information, also known as the init_segment. The remainder of the rows show the transmission cost in bytes for each segment index per representation rate (column). The provided information enables evaluating streaming performance using trace-based simulations and are suitable for use in a wide variety of simulators

This dataset offers a means of comparing the streaming performance of a wide variety of different clips as well as the segment distribution, encoded rate, and overall transmission cost between different encoders for the same clip.

Note: as the Blu-Ray content was not ours to share, a decision was made not to encode the audio for this dataset. This was extended to all clips to provided consistency across all content, thus mandating all datasets would be composed of video content only.

3. One issue with publicly available datasets is that they are only accessible online and the delivery of the content can succumb to congestion on the delivery network, which can lead to issues such as delay, jitter, and ultimately re-buffering. These issues can ultimately affect the repeatability of the experiments. Thus we feel that the availability of a local, as needed, DASH dataset is very beneficial. We do appreciate that an entire dataset could be downloaded using a web crawler or similar mechanism, but as we will show later in the paper, our entire dataset of 464GB is reduced to 518MB which is more convenient in terms of download time and bandwidth usage.

   To that end we offer our third dataset, composed of the header information from each of our two hundred and thirty evaluated clips (twenty three clips * two encoders * five segment durations). As we are not making the content available and only the decoding parameters in the header files, we can release the header information for both our testbed evaluation and trace-based datasets. Per clip, we offer the respective MPD file, the header information for the MP4 file as well as the header information per segment.

   As this dataset does not contain any decodable content, the dataset was evaluated using GPAC with a modified configuration file. Specifically we changed the "Video Driver Name" to *RAW*, by which the video is not displayed (the UI is not shown), but the underlying request/receive mechanisms of the DASH adaptation algorithm in MP4Client works as normal. In depth details on the creation and usage of this dataset is given in the next Section.

Details of content, usage and build configurations for all of our datasets are available at [14].

4. CONTENT GENERATION

The underlying encoding steps for all content is the same, but due to space limitations we illustrate the steps for the twenty Blu-ray clips only. For encoding with H.264 we use version .146 of x264 from "VideoLan" [2], while for H.265 encoding we use version1.5 of x265 also from "VideoLan". As stated, we extract 16 minutes of video content from the Blu-Ray disc and create a Matroska (*mkv*) 1920x1080 file, we convert this to Raw YUV at the resolutions presented in Table 1. Both the mkv and the YUV files are generated using FFMPEG [15], a well-known open-source cross-platform solution for converting and streaming audio and video. Over 3TB of raw YUV is created in this step.

We then convert the YUV files to H.264/5 based on the set range of input encoding representation rates as shown in Table 1 and the underlying segment duration mandated per clip, thus minimising the number of I-frames per generated DASH segment. As seen in Table 2, the output encoded representation rates per clip are different to the input encoding rates, but this is typically based on the content and range of motion within each clip.

At this point, we use "MP4Box', a multimedia packaging tool from GPAC, to wrap the encoded content within an MP4 header file. Then we again use MP4Box to create our segmented content. The DASH content is generated based on the segment duration as outlined in the encoding step, thus mandating that each segment is a stand alone moment in time, which can be decoded and used independent of all other segments over all representation rates. An as example, for the 4-second segment duration, this will mandate 150 segments for BBB, 164 segments for ED and 240 segments for all other clips. For our trace-based dataset we create a segment distribution file from each of the two hundred and

[2]http://www.videolan.org/index.html

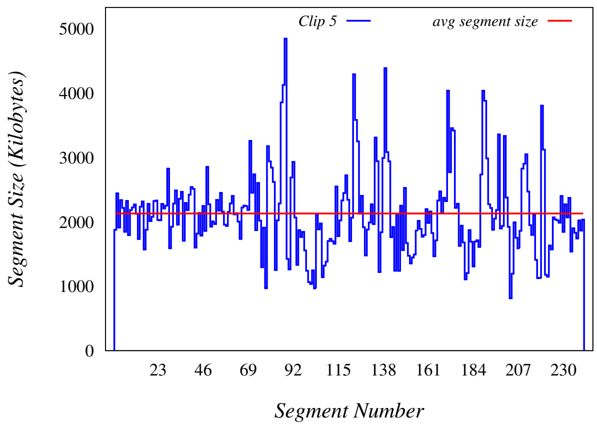

Figure 1: H.264 segment distribution for the highest representation rate for clip 5 with a 4-second segment duration

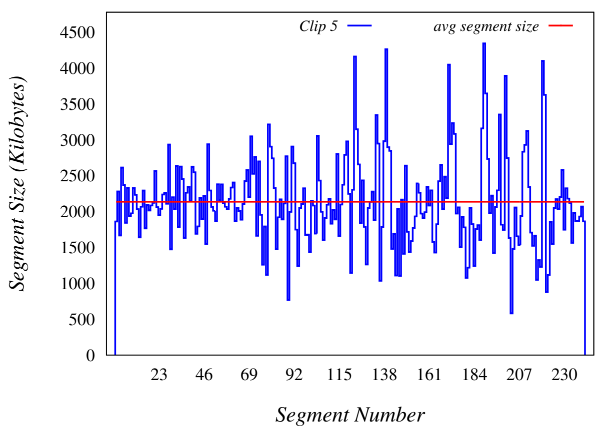

Figure 2: H.265 segment distribution for the highest representation rate for clip 5 with a 4-second segment duration

thirty encoded clips (utilising the twenty Blu-ray, and the original three animated, clips), an example of the segment distribution file is shown in Table 2. Over 1.7TB of data is created in this step with the DASH component of the testbed dataset composed of approximately 47GB.

As an example of the range of segment distribution for a given clip, Figure 1 plots the H.264 segment distribution for the highest representation rate for clip 5. While Figure 2 illustrates the same clip but encoded with H.265. From both figures, the changes of the per segment transmission cost can vary quite considerably. Note that the highest representation rate is relatively the same in both figures, denoting that H.265 is of a higher quality, while the amplitude is of a lower range in H.265. The segment distribution is relatively close, as segment content is based primarily on motion content.

While we do appreciate that multiple passes of the encoder on the content can improve the distribution of the encoded information over the entire length of the video stream, thus reducing the impact of the amplitude of the segment transmission cost, it was felt that our choice on the number of passes would introduce a bias on our behalf based on a level of distribution we feel would be adequate. Consequently we choose the default encoding settings, 2 pass only, as recommended by the instruction provided by the "bitmovin" team at "dash-player.com" [3].

4.1 Header-Based Dataset Generation

4.1.1 Overview of Segment Structure

As defined in the "DASH-IF interoperability points and extensions" documentation [4], and based on MPEG file formats [16], the DASH segment content is typically composed of the Movie Fragment Box *moof*, which is the segment meta data, and the Media Data Box *mdat*, which contains the video and audio content. Thus for our header dataset, per segment, we will need to extract the moof and a subset of the mdat data. We use MP4Box to determine the header structure of the segment files, using "MP4Box -diso <filename.m4s>", this provides specific information, in an xml format, on the byte ranges of the aforementioned data structure boxes. For the following we will use a segment duration of 4-seconds as an example, but similar values and output are seen for all other segment durations.

For H.264, we observe that all the MP4Box generation segments contain a *moof* of approximately 928 bytes, but this value varied per representation rate and underlying clip, and an *mdat* of the remaining data per segment. For H.265, we note that the moof is larger than in H.264.

As stated we use GPAC to evaluate the header dataset and GPAC like most players has two principal states: start up and steady state. The start up phase is when the client is buffering data to a certain threshold, in this phase the client requests the next segment immediately upon receipt of the current segment. The steady state is when the buffered content has reached a certain level and the client moved to an on/off phase, where the next segment is requested typically based on a lower level threshold in the buffer. If GPAC determines that the content of the mdat is un-decodable, GPAC will request the next segment immediately, thus never leaving the start up phase. So it is important that the mdat contains some decodable content.

4.1.2 Header Dataset Structure

To create the header dataset we first replicate the folder structure of the original DASH content, so as not to break any links in the MPD file. For every clip we copy its MPD file and its MP4 header file to their respective folder location. We also copy the size data of the segment distribution file for the respective clip to the new folder structure. This file, a sample of which is shown in Table 2, contains the byte transmission cost for every segment and shall be used when we rebuild the dataset so as to mandate the correct byte size of each reconstructed segment.

For every segment per representation rate we determine the location of the mdat content within the segment byte file and based on the structure of the underlying TrackRun Box with reference to the individual entries, we can determine and reduce the size of the segment to header only content. This reduces our original DASH content dataset from approximately 464GB to 727MB, which we further compress to 518MB.

[3]http://www.dash-player.com/blog/2014/11/mpeg-dash-content-generation-using-mp4box-and-x264/
[4]http://dashif.org/guidelines/

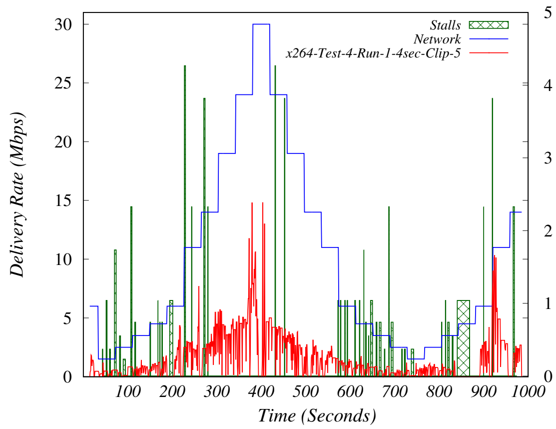

Figure 3: x264 segment distribution for the highest representation rate for clip 5 with a 4-second segment duration

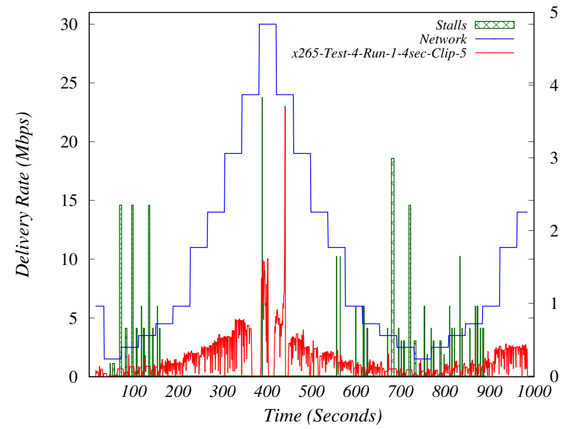

Figure 4: x265 segment distribution for the highest representation rate for clip 5 with a 4-second segment duration

4.1.3 Dataset Creation

Begin by downloading "header_dataset_and_seg_dist.zip" from [14]. Unzip and cd into the newly extracted folder. This folder contains a folder structure for each clip based on the encoder - segment duration pair (<encoder_segduration>) and three files, namely: folder_structure.sh, dataset_creation.sh and build_segments.awk, which generates the dummy video folder structure called "iVid_dataset", determines the file structure variables and builds the segments within the dummy dataset respectively. Further information on the processes of each of these files is found in a "READ_ME.txt" file on the dataset website. This entire operation assumes the presence of awk, dd, and cat linux utilities in the system.

All code is commented and is easy to understand. These scripts have been tested on a Dell Optiplex 960 Intel Core-2 Duo 3GHz with Ubuntu 14.04 and 8GB of ram. Depending on your system, you may need to make the shell script executable, using "chmod +x <file.sh>". Extracting and building the entire dataset of 464GB, takes approximately ten hours to create, taking in the order of thirty minutes per encoder_segment duration pair.

5. EVALUATION RESULTS

As previously stated we evaluate our datasets with GPAC for our real-time testbed and our header dataset, and NS-3 for our segment distribution trace-based simulation. Due to space limitation we will only provide results from the evaluation of our header dataset. Once you have downloaded "GPAC_test_code.zip" from [14] , unzip and view the "READ_ME.txt" file. This folder contains four scripts. The pertinent two scripts to call are 1) source_and_build_GPAC_ubuntu.sh - a script to install GPAC and its dependencies on a fresh install of Ubuntu 14.04 and 2) DASH_MP4Client_script.sh - a script to call GPAC and generate the trace log file. This script can be used to call GPAC with "X11" for our real-time testbed and with "RAW" for our header dataset. Note: we have also tested out header dataset with "X11" and it works fine, but does not display content, just colours on the screen.

For our evaluation, we utilise the same generated log files but extend these scripts to provide a means of determining per segment, the arrival time (milliseconds - ms), delivery time (ms), stall duration (ms), representation rate and buffer level (in seconds) at the client. A sample of this content is shown in Table 4.

In our evaluation of the header dataset, we mandate that a number of clients, 6 in our experiments, share the total bandwidth on a constrained link thus forcing the client to vary the representation rate over time. Figure 3 illustrates an example of one of these clients viewing the H.264 clip shown in Figure 1. The blue line denotes the variation of the total bandwidth over time, the red line the delivery rate of the client and the green box denotes stalls, where the height of the box denotes the representation rate the stall occurred at while the width of the box denotes the duration of the stall. Figure 4 illustrates the same experimentation but for the H.265 clip shown in Figure 2.

As can be seen H.264 suffers from more stalls and for longer durations. Due to the level of stalls neither encoder downloads all segments within the 16 minutes playout time but H.264 typically receives more segments within the time frame, in this example 217 segments versus 198 segments.

Benchmarking the Encoding Efficiency of H.265/HEVC and H.264/AVC [17] and Compression performance and video quality comparison of HEVC and AVC [18] are not the focus of this work, but illustrate the comparison between achievable quality and transmission cost of H.264 and H.265.

5.0.1 PSNR Results

To supplement the evaluated testbed results, we also generated quantitive evaluation results, i.e. PSNR results, for each of the encoded mp4 files, per segment, thus providing a comparison of the encoding efficiency of H.264 and H.265, as well as a means of quantifying the achievable quality at the device by summation of the quantitive values of each received segment at the appropriate representation rate. Table 3 illustrate an example of the PSNR values per segment for the four second segment duration H.264 encoding of Clip 5. Table 3 is comparative to the structure and content in Table 2, with respect to representation rates, resolution values and the underlying PSNR value per segment. While PSNR values are only shown for the first two segments of Clip 5 it can be seen that the higher the representation rate the higher the PSNR value. Some variation in PSNR values can occur between representation rate over different segments, e.g. segment 1 - PSNR for rep 4267 is 45.17, while segment 2 - PSNR for rep 3818 is higher at 45.28. PSNR values for all content is available at [14].

Table 3: H.264 - PSNR Output for clip 5

Clip_5	seg_Dur	4267	3818	2976	2328	1734	1038	740	552	370	232
x264		1920	1920	1280	1280	720	640	512	512	384	320
		1080	1080	720	720	480	480	384	384	288	240
0											
1	4	45.17	44.84	41.38	41.02	37.17	36.36	34.99	34.59	32.77	31.29
2	4	45.47	45.28	41.75	41.53	37.52	36.86	35.52	35.24	33.37	31.92

Table 4: GPAC trace log file determined Metrics

Seg#	Arr_time	Del_Time	Stall_Dur	Rep_Rate	Buff_Level
21	130223	3635	0	740	2.136
22	135437	4106	0	740	0.923
23	142688	4667	2327	740	0.000
24	144674	1221	0	552	2.014
25	149525	3993	0	1038	1.163

6. FUTURE WORK

We are currently extending our quantitive evaluation results to include VQM. We also propose to supplementing the existing datasets with a Scalable Video Coded version of both H264 and H265, possibly using the toolchain provided in [7]. We intend to investigate a Server side segment generator, which will create segments in real-time based on the clip requested, the header data in "iVid_dataset" and the transmission byte cost in the relative segment distribution file, thus reducing the need to build the entire "iVid_dataset".

7. CONCLUSION

In this paper, we present datasets for both trace-based simulation and real-time testbed evaluation of Dynamic Adaptive Streaming over HTTP (*DASH*). Our datasets are available in both H.264 and H.265 with encoding rates comparative to the representations and resolutions of content distribution providers such as Netflix, Hulu and YouTube. We offer twenty three clips across a range of genres and over a number of different segment durations. We evaluated our datasets using NS-3 and GPAC, and illustrate the results.

8. ACKNOWLEDGEMENT

This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under Grant Number 13/IA/1892. Special thanks to Darijo Raca and Aleksandr Reviakin for their assistance in the creation and testing of this content.

9. REFERENCES

[1] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast 2015-2020 availble at: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html.

[2] T. Stockhammer. Dynamic Adaptive Streaming over HTTP – Standards and Design Principles. In *MMSys '11 Proceedings of the second annual ACM conference on Multimedia systems*, page 133, New York.

[3] T. Wiegand, G.J. Sullivan, G. Bjontegaard, and A. Luthra. Overview of the H.264/AVC Video Coding Standard. *Circuits and Systems for Video Technology, IEEE Transactions on*, 13(7):560–576, July 2003.

[4] G.J. Sullivan, J. Ohm, Woo-Jin Han, and T. Wiegand. Overview of the High Efficiency Video Coding (HEVC) Standard. *Circuits and Systems for Video Technology, IEEE Trans. on*, 22(12), Dec 2012.

[5] Stefan Lederer, Christopher Müller, and Christian Timmerer. Dynamic Adaptive Streaming over HTTP Dataset. In *Proceedings of the 3rd Multimedia Systems Conference*, MMSys '12, pages 89–94, New York.

[6] J. Le Feuvre, J-M. Thiesse, M. Parmentier, M. Raulet, and C. Daguet. Ultra High Definition HEVC DASH Data Set. In *Proceedings of the 5th ACM Multimedia Systems Conference*, MMSys '14.

[7] Christian Kreuzberger, Daniel Posch, and Hermann Hellwagner. A scalable video coding dataset and toolchain for dynamic adaptive streaming over http. In *Proceedings of the 6th ACM Multimedia Systems Conference*, MMSys '15, pages 213–218.

[8] M. Leszczuk, L. Janowski, and M. Barkowsky. Freely available large-scale video quality assessment database in full-hd resolution with h.264 coding. In *Globecom Workshops (GC Wkshps), 2013 IEEE*, pages 1162–1167, Dec 2013.

[9] Netflix representation rates available at: https://www.pretzellogix.net/2014/02/16/testing-my-netflix-network-speeds/.

[10] Te-Yuan Huang, Nikhil Handigol, Brandon Heller, Nick McKeown, and Ramesh Johari. Confused, Timid, and Unstable: Picking a Video Streaming Rate is Hard. In *Proceedings of the 2012 ACM Conference on Internet Measurement Conference*, IMC '12, 2012.

[11] GPAC availble at: https://gpac.wp.mines-telecom.fr/home/about/.

[12] Video Traces for Network Performance Evaluation. Available at: https://media.xiph.org/video/derf/.

[13] NS3 available at: https://www.nsnam.org.

[14] Datasets available at: http://www.cs.ucc.ie/misl/research/current/ivid_dataset.

[15] FFMPEG availble at: https://www.ffmpeg.org.

[16] MPEG File Formats Available at: http://perso.telecom-paristech.fr/~concolat/MPEGFileFormats.pdf.

[17] H. Koumaras, M. Kourtis, and Drakoulis Martakos. Benchmarking the encoding efficiency of H.265/HEVC and H.264/AVC. In *Future Network Mobile Summit (FutureNetw), 2012*, pages 1–7, July 2012.

[18] H. Koumaras, M.-A. Kourtis, and C. Skianis. Compression performance and video quality comparison of HEVC and AVC. In *Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2014 IEEE 19th International Workshop on*, pages 75–79, Dec 2014.