Content Aware Video Encoding for Adaptive HTTP Streaming

Velibor ADZIC, Hari KALVA, and Borko FURHT

***Abstract*—We evaluate the impact of content encoding and present an improved encoding scheme for preparation of video content for adaptive streaming over HTTP. Current solutions are content agnostic and do not take advantage of frame level information such as scene changes. Experiments show that by using our model we can achieve savings in bitrate of up to 15 percent for the same quality level. This reduction in bandwidth is significant for services such as such as Internet streaming which takes up most of the bandwidth in peak hours.**

I. INTRODUCTION

Adaptive streaming over HTTP is gaining popularity and is becoming main platform for delivery of online video. The use of adaptive HTTP streaming began about two years ago and commonly used solutions are Smooth Streaming [1], HTTP Dynamic Streaming [2], and HTTP Live Streaming [3]. Since all of the implementations were proprietary and strong industry interest motivated MPEG to start efforts on standardizing Dynamic Adaptive Streaming over HTTP (DASH) [4]. This only confirms that adaptive streaming will become de-facto standard for online video. One of the most popular destinations for online streaming, Netflix is using adaptive streaming for delivery of video content. There are already reports [5] which state that "In North America, Netflix is now 29.7% of peak downstream traffic and has become the largest source of Internet traffic overall. Currently, Real-Time Entertainment applications consume 49.2% of peak aggregate traffic, up from 29.5% in 2009 – a 60% increase". It is essential to investigate possible optimizations that would introduce savings in bandwidth spent on video content. In order to address this issue we created simple optimization model that introduces gains that are significant when put into context of aforementioned statistics.

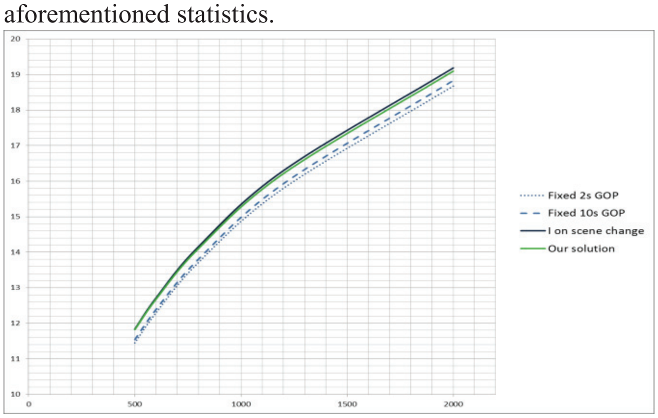

Fig. 1. RD characteristics for all sequences encoded with fixed GOP 2s and 10s segments and sequences prepared using our algorithm. X-axis shows average bitrate of a sequence (in kbps), Y-axis shows SSIM levels (in dB).

II. ADAPTIVE STREAMING

All adaptive streaming platforms have same basic model – a video stream is encoded in multiple constant bitrate (CBR) representations, which allows end user to switch between lower or higher bitrates as his available bandwidth changes. Content is divided either in segments (separate files) or fragments (virtual partitions of one file) with specified duration. Every segment has to start with Random Access Point (IDR frame in MPEG AVC/H.264) in order to allow independent segment playout and better trick mode options handling. Current DASH specification does not mandate any specifics for encoding of content as long as valid bit stream is produced. Also, the segment duration is not standardized. It is left to content creators to decide on specific parameters. Smooth Streaming recommends 2 seconds duration for segments, HTTP Dynamic Streaming recommends 4 seconds and Live Streaming uses 10 seconds as default segment duration. Most of the content is encoded with H.264/AVC codec as one fixed closed group of pictures (GOP) per segment, having I (IDR) frame as first frame and only P frames in the rest of segment (for baseline profiles commonly used for mobile video delivery) or combination of P and B frames for higher levels.

III. CONTENT AWARE MODEL

Our model introduces optimized GOP structure that takes into account scene changes. We use additional logic during encoding that detects significant scene changes (changes that result in residual that is over specified threshold) and place I frames on these boundaries. Such a placement produces better rate distortion (RD) performance for the coded video. As we can see in Figure 1, the best performance is obtained with placing I frames only on scene changes. However if we use I-frames only at scene changes, we would have to align segments with them in order to enable segment level bitrate switching and that would result in variable segment lengths that are content dependent. Results of such variability can be unpredictable for network delivery platform which means that in-time delivery is not guaranteed. We tested sequences with only scene change I frames. Results of network simulation for one of the sequences from movie "Matrix" is shown in Figure 2. Playback started after 3 segments are downloaded and segments are downloaded at regular intervals without pause; three segments is the longest buffering time allowed in current implementations. As we can see 18 out of 72 segments arrived too late, which resulted in 18 pauses during video playback.

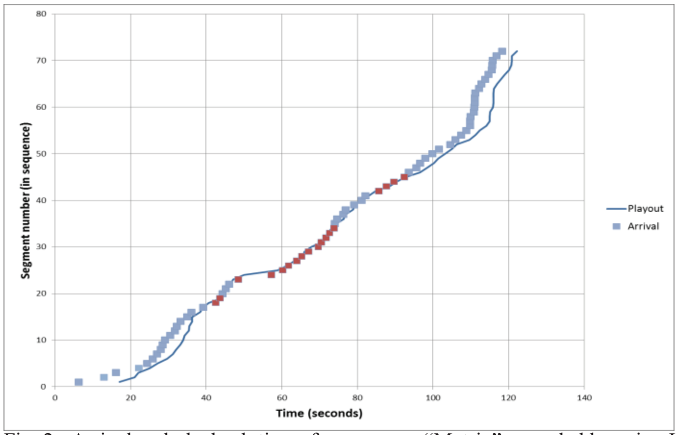

Fig. 2. Arrival and playback times for sequence "Matrix", encoded by using I frame insertion only at scene changes. Segments marked with red arrived late.

It is clear that additional I frames are needed to stabilize segment duration and bitrate. We implemented algorithm that inserts minimal number of additional frames needed for stable delivery. We are setting two decision thresholds:

B – bits spent on current GOP
F – number of frames in current GOP

The algorithm pseudo code is as follows:

while current GOP (since last I frame)
  if ($B > B_{threshold}$ or $F > F_{threshold}$)
    insert I frame
  endif
next frame
end

Thresholds for bitrate and number of frames are calculated using statistics from sequence with I frames inserted only on scene changes.

IV. EXPERIMENTS AND RESULTS

We implemented our algorithm using H.264 encoder (2 pass CBR coding) on a dataset of movie clips that had following parameters: duration – 120 seconds, framerate – 25, bitrates 500, 500, 800, 1200 and 2000 kbps and resolution 1280x720p. Results for a sequence "Good Will" that has few scene changes (only 8) and low motion across frames are shown in Figure 3.

By using this algorithm we managed to stabilize segment duration and bitrate which significantly improved network performance. We also compared rate distortion (RD) characteristics of content prepared with our model and contents prepared by recommendations for 2 and 10 seconds. Results are shown in Figure 1 (green curve). Savings that are achieved across all sequences in our test set are up to 10%. For some sequences, depending on the distribution of scene changes this savings can go up to 15%.

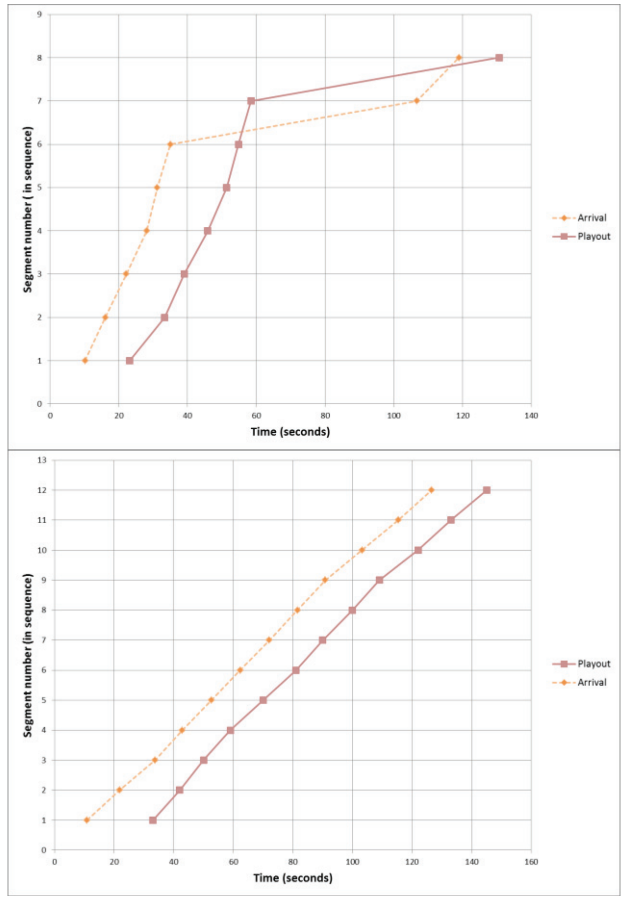

Fig. 3. Arrival and playback times for sequence "Good Will" – top: sequence with only scene change I frames, bottom: network performance after content is optimized with our algorithm.

V. CONCLUSIONS

We have introduced content aware model for preparation of video content for adaptive streaming over HTTP. We show that using I frames only on scene changes leads to variable size segments and result in buffer underflow and stalled playback. In the proposed approach, segments are coded in order to minimize the number of I frames per segment while using I-frames at scene changes. Through experiments we demonstrated advantages of using our algorithm for stabilization of segment characteristics and savings that can be achieved by using our model compared to recommended implementations that are currently used.

REFERENCES

[1] A. Zambelli, "Smooth Streaming Technical Overview". http://learn.iis.net/page.aspx/626/smooth-streaming-technical-overview , 2010.
[2] "Using Adobe HTTP Dynamic Streaming". http://help.adobe.com/en_US/HTTPStreaming/1.0/Using/index.html , 2010
[3] R. Pantos, (Ed). "HTTP Live Streaming draft". http://tools.ietf.org/html/draft-pantos-http-live-streaming-06, 2011).
[4] T. Stockhammer, P. Fröjdh, I. Sodagar, S. Rhyu,(ed.) "Information technology — MPEG systems technologies — Part 6: Dynamic adaptive streaming over HTTP (DASH)",ISO/IEC, MPEG Draft International Standard ,2011.
[5] Sandvine Intelligent Broadband Networks, "Internet Phenomena report", Spring 2011, http://www.sandvine.com ,2011