

Backward-Shifted Coding (BSC) for HTTP Adaptive Streaming

Zakaria Ye*, Rachid EL Azouzi*, Tania Jimenez*,
Stefan Valentin⁺

University of Avignon, France*, Huawei
Technologies, France⁺
{ zakaria.ye, rachid.elazouzi,
tania.jimenez}@univ-avignon.fr,
stefan.valentin@huawei.com

1. Backward-Shifted Coding

Before introducing our Backward-Shifted Coding, we describe the SVC [1], which composed of a H264/AVC-compatible base layer and one or more enhancement layers. These layers increase the SNR fidelity, spatial and/or temporal quality of the video when added to the base layer. In fact, the description of our scheme will be based on the H.264/SVC codec.

The BSC scheme is entirely client driven. The main idea of this scheme is to decompose the segment or the GoP in base layer frames and enhancement layer frames and shift them. The lag between the base layer and the enhancement layer frames is ϕ . The base layer frames are less quality and they are sent before the enhancement layer frames. Each frame n has its enhancement layer in subsequent frame $n + \phi - 1$ (see fig. 1). Thus, when the enhancement layer is missed, the player can still play out the base layer frame.

In our BSC scheme, each segment or block is encoded with different rates based on the network conditions and the scheduling technique. Typically, that means the encoding rate of the base layer and the encoding rate of the combined base and enhancement layers. The encoding rate of the base layer may differ from one segment to another. That is an interesting property of the BSC system to be explain later. At the user side, incoming bits are

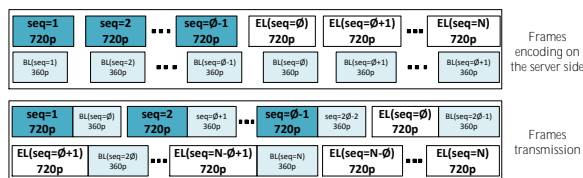


FIGURE 1 – Using SVC in Backward-Shifted Coding

reassembled into video frames by the decoder. Under DASH, each video client is divided into multiple small segments at the media server. Various representations of a segment or block are proposed by using different encoding rates. Hence, the BSC scheme can be used with evident client using the DASH framework. The base layer is re-encoded for each segment, possibly with a different encoding rate, depending on the network throughput measurements. It is a runtime encoding process like in live video streaming systems. Therefore, with BSC under DASH, the adaptation engine requests the two appropriate segments with different bitrates (base layer and enhancement layers).

2. Bitrate Adaptation with Backward-Shifted Coding

2.1. System Description

We consider a video streaming system using the Backward-Shifted Coding described. The server holds the media segments and a HTTP server. The client holds an adaptation engine and a playout buffer where the video frames are decoded and stored to be display on the screen. The main goal of the adaptation engine is to estimate the throughput and select the bitrates of the next segment to be downloaded.

We assume N to be the size of the video file in number of frames. Let d be the buffer playout frequency (e.g., 30fps). The video duration Δ (in seconds) is N/d . A video file is a set of consecutive video segments or chunks, $v = \{1, 2, \dots, K\}$, each of which contains L seconds of video and encoded with different bitrates. The number of segments K in the video is Δ/L . We assume $\mathcal{R} = \{R_1, R_2, \dots, R_r\}$ to be the set of available bitrates. In BSC system, the player downloads the video segment k with bitrate $(R_{ki}, R_{kj}) \in \mathcal{R}$ where R_{ki} is the bitrate of the basic layer and R_{kj} is the bitrate of the combined basic layer and enhancement layer.

2.2. Adaptation methods in BSC

The goal of the bitrate adaptation is to maximize the quality of experience of the user. We consider the throughput based method (TBA) (current segment throughput) and the buffer based method (BBA) (buffer level) algorithms. For TBA, the algorithm selects the adequate bitrates after the download of the current segment. The pseudo-code is provided in algorithm 1 where $R_{<}(t_i)$, $R_{>}(t_i)$, R_{\min} and R_{\max} are such that $R_{\min} < \dots < R_{<}(t_i) < \hat{A}(t_i) < R_{>}(t_i) < \dots < R_{\max}$.

Algorithm 1: Adaptation Algorithm

Input:

$\hat{A}(t_i)$: the estimated throughput of the previous segment i

$R_i(BL)$: the bitrate of the base layer of the previous segment i

Output:

$R_{i+1}(BL)$: the bitrate of the base layer of the next segment $i + 1$

$R_{i+1}(BL + EL)$: the bitrate of the combined layers of the next segment $i + 1$

```

1  $R_{<}(t_i) \leftarrow \hat{A}(t_i)^\downarrow$ ;
2  $R_{>}(t_i) \leftarrow \hat{A}(t_i)^\uparrow$ ;
3 if  $\hat{A}(t_i) \leq R_{min}$  then
4    $R_{i+1}(BL) := R_{i+1}(BL + EL) := R_{min}$ ;
5 else
6   if  $\hat{A}(t_i) \geq R_{max}$  then
7      $R_{i+1}(BL) := R_{i+1}(BL + EL) := R_{max}$ ;
8   else
9     if  $\hat{A}(t_i) < R_i(BL)$  then
10       $R_{i+1}(BL) := R_{<}(t_i)$ ;
11       $R_{i+1}(BL + EL) := R_{>}(t_i)$ ;
12     else
13       $R_{i+1}(BL) := R_i(BL)^\uparrow$ ;
14       $R_{i+1}(BL + EL) := R_{i+1}(BL)^\uparrow$ ;

```

For BBA we define three buffer thresholds B_{min} , B_{low} and B_{high} and make the bitrate selection decision on the buffer filling level. Algorithm 1 is used at the beginning of this method to increase quickly the bitrate at the beginning of the video playback in order to maximize the video quality.

3. Simulations and Numerical results

3.1. Simulation setup

We use MATLAB to simulate the event-driven bitrate switching system. Our traffic model (see fig.

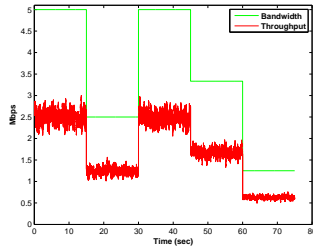


FIGURE 2 – Background traffic with Level shift

2) simulates a realistic network bandwidth variations with congestion level shift where they inject background TCP traffic between the server and the client. The link capacity between the server and the client is set to be 5Mbps. The traffic rate jumps between four different levels, oscillations within the same level are due to TCP congestion control mechanisms. The link capacity is higher than the TCP throughput.

3.2. Numerical Results

This set of experiments compares the requested bitrate with BSC and DASH under the network conditions of fig. 2. The file size in the experiments is up to 100 seconds of video while the playback frequency is 25 fps. We consider the following set of available bitrates $\{140, 250, 420, 760, 1000, 1500, 2100, 2900\}$ (Kbps).

The video segment duration is set to 2 seconds.

Figure 3 shows the requested bitrate for BSC and

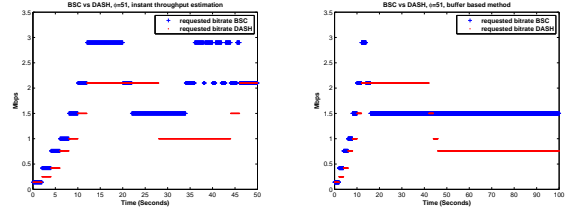


FIGURE 3 – BSC vs DASH instant throughput estimation
 FIGURE 4 – BSC vs DASH buffer based method

DASH for TBA where the throughput is estimated over 1 segment. The average video bitrate is much higher with BSC system for a few additional bitrate switchings. These bitrate switchings are tolerable since they are between two consecutive bitrates. We can still cope these switchings by forcing the system to stay at the base layer frames if the time spent at the base level does not exceed a certain threshold.

Figure 4 shows the requested bitrate for BSC and DASH for BBA $B_{min} = 5sec$, $B_{low} = 7sec$, $B_{high} = 50sec$. BSC system achieves a better video quality and decreases the bitrate switchings compared to TBA.

4. Conclusion

We proposed a novel coding scheme to improve the performance of the HTTP adaptive video streaming. BSC is inspired from the FEC method. The media server transmits two shifted segments with different bitrates, i.e., a base layer with enhancement layers, in order to enhance the video quality. We proposed bitrate adaptation algorithms in BSC. We further performed simulations to show the efficiency of BSC system compared to existing DASH solutions.

Bibliographie

1. ISO/IEC – Coding of Audio-visual Objects - Part 10 : Advanced Video Coding 14496-10 :2012 Information Technology, 2012.