# A Geo-Adaptive JavaScript DASH Player

Garson Zhong, Ayub Bokani
School of Computer Science and Engineering
The University of New South Wales, Sydney, NSW, Australia
{gzho871, abokani}@cse.unsw.edu.au

## ABSTRACT

In Dynamic Adaptive Streaming over HTTP (DASH), the video bitrate is adapted to the network's condition based on client's feedback. Typically, a high bitrate is selected for next video chunk when the average of previously observed bandwidth samples is high and vice versa. An unexpected drop in the network throughput causes video freezing as the adaptation is done based on network's status in a previous moments. In vehicular environment, this problem is more likely to occur due to rapid movements of client to new locations where the network throughput can be very different. In this paper, we present a modified *JavaScript* DASH player that enables the streaming client to optimize the streaming performance by taking advantage of pre-collected bandwidth statistics in different locations. In this video player, Markov Decision Process (MDP) has been considered as the underlying optimization framework.

## Keywords

Adaptive Video Streaming; Bandwidth Map

## 1. INTRODUCTION

Increasingly, video is being watched on mobile devices. This is due to content publishers making the switch from traditional methods of multimedia to online delivery and more specifically making content available through mobile devices. Typically, received mobile signal strength is extra volatile for users that are on the move, severely affecting the quality of experience of streaming multimedia via wireless devices. This paper proposes our solution to seamlessly stream multimedia and compares its performance against conventional video quality adaption solutions.

## 2. BANDWIDTH MAP

In this section, we describe our bandwidth collection approach to create bandwidth statistics. Bandwidth measurements are conducted on an Android application that continuously measures the available *downlink* mobile bandwidth at intervals. The algorithm used by the application resembles that used by the openly available Speedtest.net with a few differences. The size of the download test file was set to be 1MB rather than dynamically determined, which is approximately the average size of a single video chunk. Another difference is that bottom 20% of samples caused by TCP's slow ramp up are kept rather than discarded as it will inevitable be experienced as video is requested over HTTP. Each test is tagged with time, client GPS coordinates, and service provider name and code. The mean and standard deviation of different sections along a route can be extracted from the generated bandwidth map. These bandwidth statistics are then used in our MDP framework to compute optimal streaming policies for different segments of our route.

MDP as a *well-known* optimization framework for making decision on uncertain environments has been recently considered by researchers in adaptive video streaming. In this study, we also modeled the streaming adaptation as an MDP problem which is mainly based on [1].

## 3. DASH PLAYER AND MODIFICATIONS

In this section we briefly describe the DASH Reference Player 1.0.0 [2] and our MDP optimized player.

The algorithm used by the DASH Player calculates an initial download ratio by dividing the video chunk length by the download duration. This initial ratio is multiplied by a constant 'safety factor', which is a specified value between 0 and 1. If the final download ratio is greater than 1, then the algorithm will switch up, and if less than 1 it will switch down.

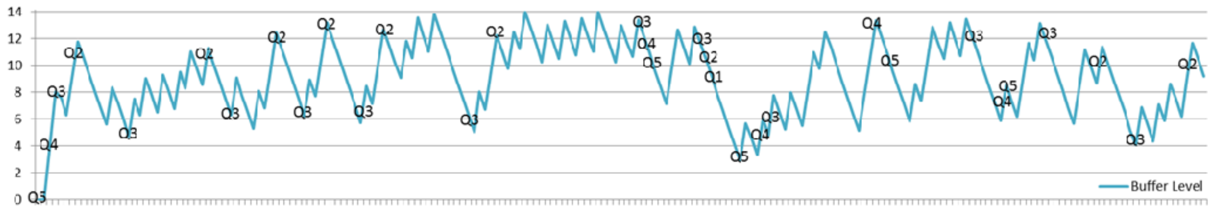
The player used to test our MDP optimization framework was created by extending the DASH Player. The MDP strategy is loaded when the player is initialized and once buffer controller determines the current location, a lookup index is computed to select the best quality transition.

## 4. RESULTS

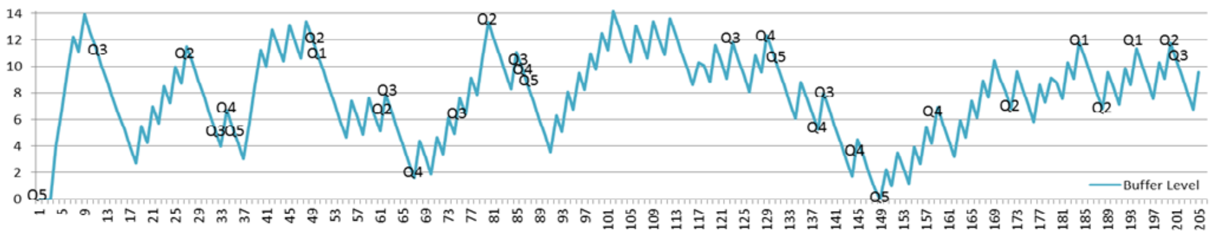
We propose the following evaluation framework for comparing the quality of service of the two players during streaming of the Envivio Red Bull Racing clip. This clip is encoded in following bit rates: 350kbps (Q5), 600kbps (Q4), 1000kbps (Q3), 2000kbps (Q2) and 3000kbps (Q1).

*Stall frequency (Deadline miss):* This metric measures how frequent the video player buffer runs dry.


*VideoNext'14,* December 2, 2014, Sydney, Australia.
ACM 978-1-4503-3281-1/14/12.
http://dx.doi.org/10.1145/2676652.2683463.




**(a)**




**(b)**

**Figure 1:** Buffer level over time during live mobility testing - (MDP) Day Test 4 (a) , Night Test 2 (b)

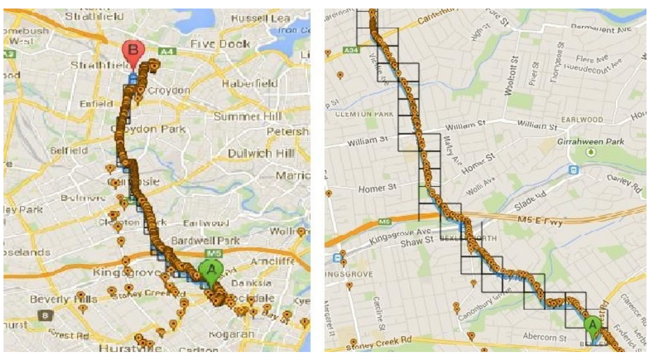


**Figure 2:** Trajectory of chosen route (left), Subroute used under live mobility testing (right)

*Quality level (Bit rate):* The average and median of quality level per stream will allow us to determine the overall quality level of the stream.

*Average buffer level:* The average buffer level allows us to determine if the algorithm for selecting quality transitions is performing as expected.

Figure 2 (left) illustrates the route we considered to generate the MDP optimal streaming strategy for the player. The quality of service provided by our video player will be optimized when streaming video along the route marked from A to B. Figure 2 (right) displays a subsection of the entire route. This subroute was selected for our tests due to levels of traffic experienced along this section which affects the bandwidth fluctuation level.

Figure 1a illustrates the buffer level experienced during one of our live daytime (2:28PM) tests. The playback of the video did not stall after the initial buffer time of 3 seconds and the average buffer level achieved during this test was 9.07 seconds with an average quality of Q3. Our pre-computed MDP optimal strategy allowed the player to immediately switch down to the lowest quality Q5 from Q2 at 105 seconds into playback upon entering a section of the route that has been measured to have a lower level of downstream throughput.

Figure 1b shows a night test (8:51PM) along the same route. The initial buffering time was 3 seconds and the average buffer level achieved was 7.76 seconds. The average quality level was Q4 and playback experienced two stalls, only momentarily in seconds 149 and 152. It can be observed that we were in a section pre-determined to have lower levels of downstream throughput around 85 seconds into playback. Although the buffer level was quite high at 11 seconds, which is more than 3 seconds higher than the average buffer level for this trip, the MDP optimal strategy dropped the quality level of the stream to the lowest available, Q5. The buffer level drops significantly right after but due to the early switch prevented a stall.

The two live tests with our *Geo-adaptive* player performed quite differently, indicating that period of day and other factors severely affect the volatility of the underlying bandwidth. The DASH Player performed quite poorly in all real live testing under vehicular mobility. It was observed that DASH performed quite well when experiencing conditions of low speed mobility, allowing the *download ratio* algorithm to adapt the quality appropriately to the available bandwidth. However when under high speed mobility, the player was observed to stall roughly every few seconds of playback.

# 5. CONCLUSIONS

We have evaluated the performance of MDP as an optimization framework for HTTP-based adaptive video streaming. Initial testing has shown that pre-computed MDP optimizations using past bandwidth observations along a route have shown significant improvements in overall quality of service provided compared with the default implementation of the DASH player.

# 6. REFERENCES


[1] A. Bokani, M. Hassan, and S. Kanhere. HTTP-based Adaptive Streaming for Mobile Clients using Markov Decision Process. In *Packet Video Workshop (PV)*, San Jose, USA. 12 December 2013.

[2] DASH-IF. DASH Reference Player 1.0.0. [Online accessed 01-October-2014], URL: http://dashif.org/reference/players/javascript/1.0.0/.