Evaluation of the bitmovin MPEG-DASH client in mobile environments

Abstract: MPEG's Dynamic Adaptive Streaming over HTTP (MPEG-DASH) is an emerging standard designed for media delivery over the top of existing infrastructures and able to handle varying bandwidth conditions during a streaming session. This requirement is very important, specifically within mobile environments and, thus, DASH could potentially become a major driver for mobile multimedia streaming. Hence, this report provides a detailed evaluation of bitmovin's implementation of MPEG-DASH compared to the most popular proprietary systems, i.e., Microsoft Smooth Streaming, Adobe HTTP Dynamic Streaming, and Apple HTTP Live Streaming. In particular, these systems will be evaluated under restricted conditions which are due to vehicular mobility. In anticipation of the results, our first prototype implementation of MPEG-DASH can very well compete with state-of-the-art solutions and further developments have shown that our client can even surpass existing clients. Thus, MPEG-DASH and our bitmovin client can be regarded as mature and ready for broad industry adaption.
1 Introduction

The Hypertext Transfer Protocol (HTTP) is currently one of most used protocols on the application layer. In particular, real-time entertainment (i.e., multimedia communication) is one of the major drivers and currently accounting for more than 50% of the Internet traffic both in the wired and wireless domain. See Figure 1 for current and projected mobile network traffic.

![Mobile Network Traffic - United States](image)

*Figure 1. Mobile Network Traffic (Source: Sandvine Global Internet Phenomena Report 1H 2012).*

Most of the real-time entertainment services are deployed over-the-top (OTT) utilizing existing Internet infrastructures. This approach works surprisingly well without any particular support from the underlying network due to the use of efficient audio/video compression, content delivery networks (CDNs), and adaptive audio/video players.

MPEGs' Dynamic Adaptive Streaming over HTTP (MPEG-DASH) is an emerging standard designed for OTT media delivery and able to handle varying bandwidth conditions during a streaming session. Hence, this report provides a detailed evaluation of bitmovins' implementation of MPEG-DASH compared to the most popular propriety systems, i.e., Microsoft Smooth Steaming, Adobe HTTP Dynamic Streaming, and Apple HTTP Live Streaming.
2 Methodology

For the experiment we have encoded a video sequence at 14 different bitrates (100, 200, 350, 500, 700, 900, 1100, 1300, 1600, 1900, 2300, 2800, 3400, and 4500 kbit/s) with a 2 second segment length. All systems have been evaluated under three different network emulation settings that have been recorded during separate freeway car drives with a HUAWEI E169 HSPDA USB Stick. Therefore, we used the A2 freeway in Carinthia/Austria shown in Figure 2 which has speed limits between 100 and 130 km/h:

- Experiment 1 / Track 1 (601 seconds): Drive on the freeway A2, passing by the city of Villach in the direction to Klagenfurt.
- Experiment 2 / Track 2 (575 seconds): From the Alpen- Adria-Universität Klagenfurt on the freeway A2 until the service area around Techelsberg.
- Experiment 3 / Track 3 (599 seconds): From the service area around Techelsberg on the freeway A2 to the exit of Klagenfurt.

Wireshark has been used to capture the behavior of each system in a consistent way which simply records the HTTP GET requests and marks them with timestamps.

We have used four metrics in order to make each system comparable: average bitrate, number of quality switches, buffer level, and stalls. The architecture of the experimental setup is depicted in Figure 3 and consists of four devices, i.e., evaluation client, bandwidth shaping, network emulation, and HTTP server.
3 Evaluation Results

Figure 4 shows the behavior of the Microsoft Smooth Streaming client for experiment 3 / track 3. Figure 4(a) shows the captured bandwidth compared to the throughput, i.e., utilized bandwidth at the client (adaptation) and Figure 4(b) shows the buffer fill state.

Figure 5 shows the behavior of Adobe HTTP Dynamic Adaptive Streaming for experiment 3 / track 3. The adaptation process is very unpredictable and in comparison to Microsofts’ system Adobe is very aggressive and does not act in a stepwise manner. Furthermore, very often the buffer is empty resulting into stalls which corresponds to very bad Quality of Experience (QoE).
Figure 6 shows the behavior of Apple HTTP Live Streaming for experiment 3 / track 3. Interestingly, it also uses the stepwise approach like Microsoft but it seems that the step size is larger compared to Microsoft. Furthermore, also the buffer size seems to be very large (approx. 200-250 seconds) compared to others.

Finally, Figure 7 shows the behavior of the bitmovin MPEG-DASH client for experiment 3 / track 3. In particular, our client is able to react immediately on bandwidth fluctuations while maintaining a stable buffer resulting in good Quality of Experience for the end user, even in mobile environments.
A comparison of the different approaches with respect to the metrics average bitrate, average switches, and stalls is given in Table 1. The average bitrate shows how much of the available bandwidth is utilized, i.e., the higher the number, the better the available bandwidth is utilized. The average switches indicates how often the client reacts on bandwidth fluctuations. For this metric, a trade-off with respect to the average bitrate needs to be taken into account as a low number of average switches implies a low average bitrate. Finally, the stalls represent the time in which the display is frozen due to buffer underruns. That is, zero means a smooth playback of the video without stalls.

Table 1. Comparison of HTTP Streaming Solutions.

<table>
<thead>
<tr>
<th>Name</th>
<th>Average Bitrate [kpbs]</th>
<th>Average Switches [# of Switches]</th>
<th>Stalls [Seconds]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft</td>
<td>1522</td>
<td>51</td>
<td>0</td>
</tr>
<tr>
<td>Adobe</td>
<td>1239</td>
<td>97</td>
<td>64</td>
</tr>
<tr>
<td>Apple</td>
<td>1162</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>bitmovin Open Source¹)</td>
<td>1464</td>
<td>166</td>
<td>0</td>
</tr>
<tr>
<td>bitmovin DASH264²)</td>
<td>2341</td>
<td>81</td>
<td>0</td>
</tr>
<tr>
<td>bitmovin DASHSVC³)</td>
<td>2738</td>
<td>101</td>
<td>0</td>
</tr>
</tbody>
</table>

¹) Our open source implementation which is also fully integrated into VLC (deployed) and Firefox (under development)
²) Our commercially available MPEG-DASH client
³) Our MPEG-DASH client utilizing scalable video coding, not yet commercially available.

Our open source implementation is comparable with existing state-of-the-art solutions. However, our commercially available bitmovin DASH264 outperforms all other players in terms of the average bitrate except the bitmovin DASHSVC which is a research prototype not yet commercially available.

4 Publications
