DASH Lab Assignment

Surbhi Sonkiya

October 29, 2018

1 Introduction

This document presents the observations and explanation for the following questions on adaptive multimedia streaming, MPD, representations, segments, buffer and shaper.

2 How long did it take for the video to stall? Explain the factors that have an impact on that time.

Time taken for the video to stall: **20 seconds**.

Factors that might have an impact on the time taken:

1. **Buffering**: In order to make the video appear smooth, the system stores data until it has enough to produce a few minutes of video. To overcome "continuous playout constraint", there is a need for client-side buffering to match the original media and playout requirements.

2. **Bandwidth**: 0.75 Mbps is provided to play the video. Video streams can contain huge amounts of data and if the bandwidth is too small the system can only receive it as fast as the bandwidth allows it to come through.

3. **Quality**: The video quality used is 2Mbps. The higher the quality of the video, higher bandwidth it would require to play the video.

4. **Memory** Sufficient RAM and an external video card (atleast 2GB) would reduce the buffering time. It would help take the load of processing the video off the core processor on the motherboard. Integrated graphics share RAM that the computer uses for other things as well and the CPU decides what gets priority.

5. **Overhead of the browser**: Use a browser with low overhead while streaming a video would also help reduce the buffering time. Since, the case with Firefox is not the same, it adds in extra time to buffer the video.

3 Explain briefly the video representation and segment elements in the MPD, including examples from your videobbb.mpd

1. **Representation**: Representations encapsulate media streams that have same content encoded in different ways. Advantage of representation is that it allows users to request for the highest quality of video that can be played on their machine without compromising on the bandwidth. Representation can also be encoded with different codecs. However, for the purpose of the assignment, bitrate (bps) and resolution (width:height) were modified to create four representations of the same video. This can be seen in the MPD file.

   From the MPD file in **Figure 1**, the "Representation" tag shows the width and height of the video representation. Within this tag, "SegmentTemplate" tag shows the media (name) of the video representation and duration.

   There is an auto-switch mode on the browser through which it selects the representation by its own and plays the video. However, the auto-switch depends upon the available bandwidth. But having multiple representation provides an additional benefit to the users, the fact that they can manually switch the quality of the video depending upon the bandwidth they want to invest.

2. **Segment**: Segments are the actual video files that is being played on the DASH client. They are played in back-to-back fashion, leaving an impression that only one video file is being played all through the time.

   In this assignment, short duration of segment (6000 msec) was used that provided the advantage of high switching granularity. Segment location is described as a URL (http://192.168.56.101//bbb/videobbb.mpd) for a lists of video representations created earlier.

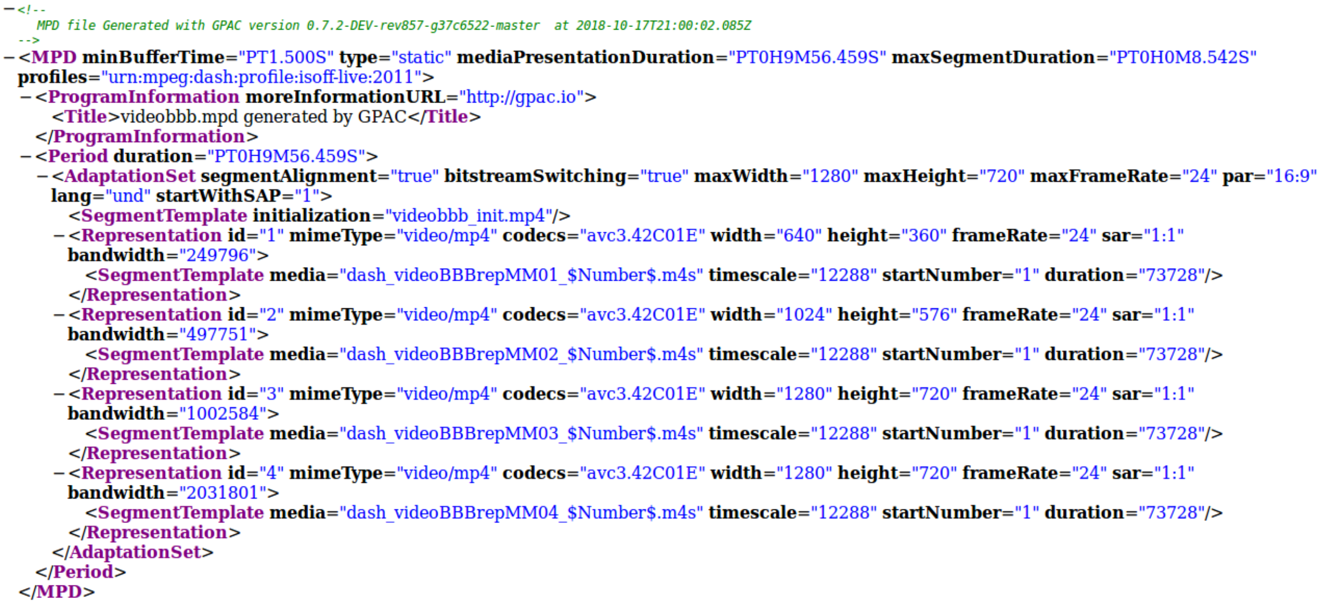
4 Explain what representation in being downloaded, if the behaviour is consistent with what you expected and why.

As seen from the **Figure 2**, the video representation with bitrate 500 kbps was being downloaded and played. The resolution was 1024:576 (width:height). So, it was "videoBBBrepMM02.mp4" video which was being downloaded and played.

In the "metric-750.txt" file, the tcp bandwidth is 0.75 Mbps for the duration of 240 seconds. Therefore, a straight line was observed for video bitrate as expected. Since, there was only one bandwidth provided, only one video representation within that bandwidth was being downloaded and played.

Output on the terminal:

At last 240 seconds your TCP BW is 0.75mbit - waiting time switch is 240 seconds.. Delay of packets: 10ms , packet loss: 0%



```
-<!--
   MPD file Generated with GPAC version 0.7.2-DEV-rev857-g37c6522-master  at 2018-10-17T21:00:02.085Z
 -->
-<MPD minBufferTime="PT1.500S" type="static" mediaPresentationDuration="PT0H9M56.459S" maxSegmentDuration="PT0H0M8.542S"
 profiles="urn:mpeg:dash:profile:isoff-live:2011">
  -<ProgramInformation moreInformationURL="http://gpac.io">
     <Title>videobbb.mpd generated by GPAC</Title>
   </ProgramInformation>
  -<Period duration="PT0H9M56.459S">
    -<AdaptationSet segmentAlignment="true" bitstreamSwitching="true" maxWidth="1280" maxHeight="720" maxFrameRate="24" par="16:9"
     lang="und" startWithSAP="1">
       <SegmentTemplate initialization="videobbb_init.mp4"/>
      -<Representation id="1" mimeType="video/mp4" codecs="avc3.42C01E" width="640" height="360" frameRate="24" sar="1:1"
       bandwidth="249796">
         <SegmentTemplate media="dash_videoBBBrepMM01_$Number$.m4s" timescale="12288" startNumber="1" duration="73728"/>
       </Representation>
      -<Representation id="2" mimeType="video/mp4" codecs="avc3.42C01E" width="1024" height="576" frameRate="24" sar="1:1"
       bandwidth="497751">
         <SegmentTemplate media="dash_videoBBBrepMM02_$Number$.m4s" timescale="12288" startNumber="1" duration="73728"/>
       </Representation>
      -<Representation id="3" mimeType="video/mp4" codecs="avc3.42C01E" width="1280" height="720" frameRate="24" sar="1:1"
       bandwidth="1002584">
         <SegmentTemplate media="dash_videoBBBrepMM03_$Number$.m4s" timescale="12288" startNumber="1" duration="73728"/>
       </Representation>
      -<Representation id="4" mimeType="video/mp4" codecs="avc3.42C01E" width="1280" height="720" frameRate="24" sar="1:1"
       bandwidth="2031801">
         <SegmentTemplate media="dash_videoBBBrepMM04_$Number$.m4s" timescale="12288" startNumber="1" duration="73728"/>
       </Representation>
     </AdaptationSet>
   </Period>
 </MPD>
```

Figure 1: MPD file

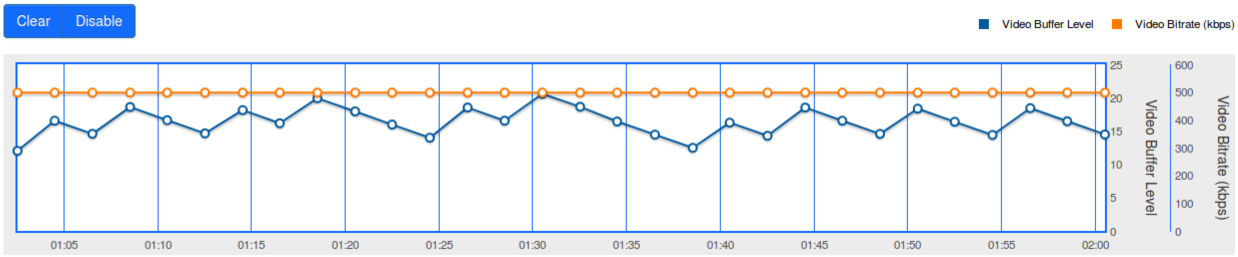


Figure 2: DASH client canvas with shaper metric-750

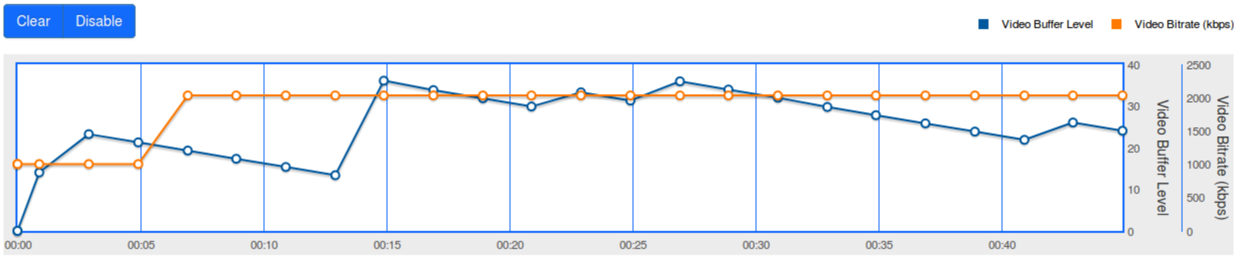


Figure 3: DASH client canvas with shaper metric-var

5 Explain the evolution of the buffer and of the downloaded representation, taking into account the bandwidth constraints applied with the shaper.

In the “metric-var.txt” file, the tcp bandwidth is 4, 1.2, 0.4, and 4.0 Mbps for the duration of 30, 60, 90, and 120 seconds respectively. Therefore, a variation in the line was observed for video bitrate and video buffer as expected. Since, there were different bandwidths provided in the shaper (metric-var.txt file), different video representations were being downloaded and played depending upon the bandwidth for that duration of the time. After every 30 second, bandwidth would switch and another video representation would be downloaded and played. At the initial start, there is a little buffer time needed, but once the video starts playing, even though it switches between the various bandwidths and video representation, user doesn’t have to wait for anymore buffering in between.

Figure 3, 4, and 5 show the different representations of the video being played.

Output on the terminal:

```
At last 30 seconds your TCP BW is 4mbit - waiting time switch is 30 seconds.. Delay of packets: 10ms , packet loss: 0%
eth1
At last 30 seconds your TCP BW is 1.2mbit - waiting time switch is 30 seconds.. Delay of packets: 10ms , packet loss: 0%
eth1
At last 30 seconds your TCP BW is 0.4mbit - waiting time switch is 30 seconds.. Delay of packets: 10ms , packet loss: 0%
eth1
At last 30 seconds your TCP BW is 4.0mbit - waiting time switch is 30 seconds.. Delay of packets: 10ms , packet loss: 0%
```

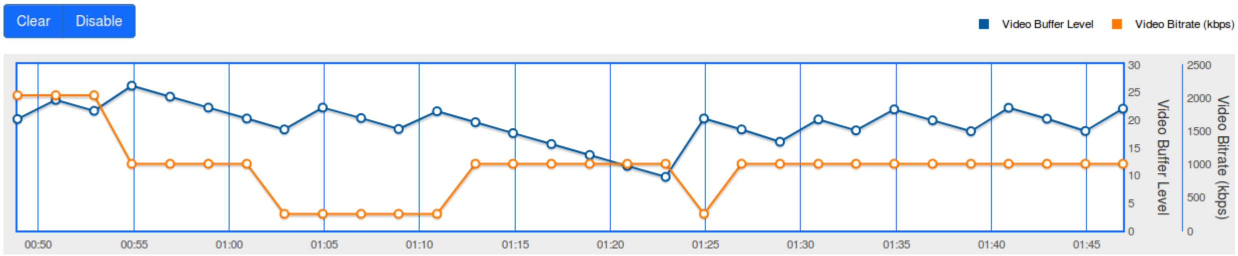


Figure 4: DASH client canvas with shaper metric-var

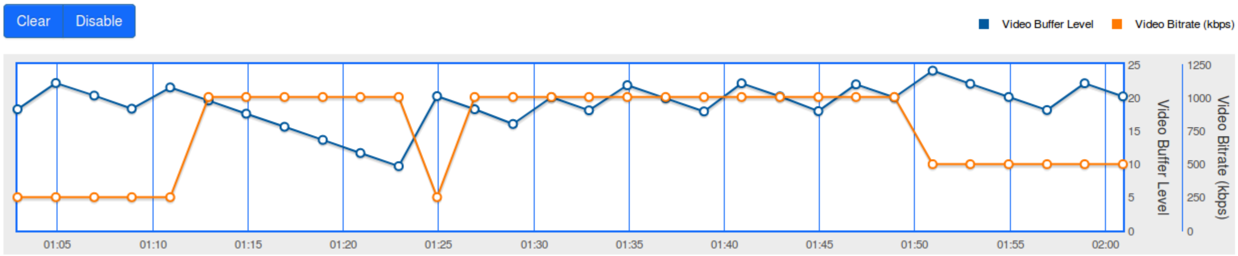


Figure 5: DASH client canvas with shaper metric-var

6 Conclusion

Adaptive bitrate streaming adjusts the quality of the media stream according to the bandwidth and the CPU capacity available at the client end. It uses MPD file that encodes the details of different representations (different in terms of bitrate, resolution, etc.) and URLs of all the segments available for the same media content. Therefore, depending upon the resources at the client end, the player at the client end (essentially the browser or DASH client) does the task of switching between the streaming of various available representations. In this way, it manages to achieve very less buffering time to start the video and a good trade off between wait time and quality of video. Hence, providing satisfactory user experience for both high-end and low-end connections.

7 References

1. https://www.researchgate.net/publication/275801770_MPEG-DASH_Enhanced_Multimedia_Streaming

2. https://en.wikipedia.org/wiki/Adaptive_bitrate_streaming

3. http://www.giraffic.com/cracking-mpeg-dash-optimize-performance-evolving-standard/

4. https://www.brendanlong.com/the-structure-of-an-mpeg-dash-mpd.html

5. https://www.encoding.com/mpeg-dash/