MPEG-DASH Live Streaming in an Unstable Environment

Avraham Ayaso   Lavy Raz
Itzik Ashkenazi
Technion CS Department
savia, lavyraz, iashken@campus.technion.ac.il

Aviel Glam
Rafael Advanced Defence Systems Ltd.
avielg@rafael.co.il

Keywords MPEG-DASH, HTTP, AIMD, Unstable Communication, Link Capacity Estimation, Live-stream

Abstract

Dynamic Adaptive Streaming over HTTP (DASH[2]) is a protocol that designed to ac hive adaptive streaming in aspect of quality in coherence with the client bandwidth (BW). However, MPEG-DASH was not initially designed to achieve live streaming.

Our research examines the protocol live streaming behaviour, especially in an unstable environment. Several algorithms were implemented and analyzed on different environments, with the goal of optimizing live streaming via adaptive methods based on key factors of the communication.

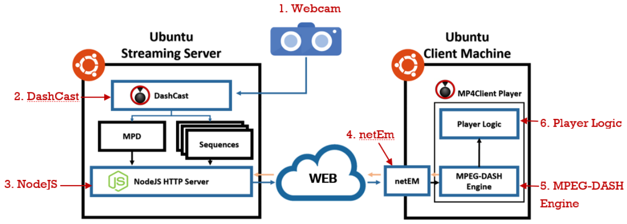

Figure 1. Implemented architecture with single client

The architecture depicted in figure 1 and includes the following components:

1. Webcam: Generates the live video stream.
2. DashCast[1]: Breaks the input stream into sequences of small segments and stores them in several different qualities. It then creates Media Presentation Description (MPD) file that mainly defines the available stream qualities.
3. NodeJS: Server platform that get the MPD file and stream sequences via standard Web interface.
4. netEM: Simulates an unstable network behavior on the client's Interface.
5. MPEG-DASH engine[1]: Generates HTTP GET commands from client to server, asking for the estimated quality while maintaining a tolerable delay.
6. Player Logic: add delay for reordering the video stream.

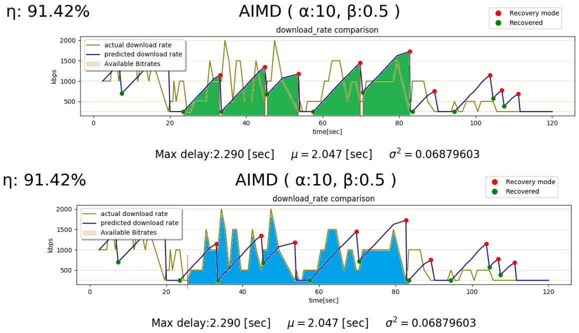

Figure 2. Comparison of the ideal and the predicted result

The research focuses on three main streaming methods: **throughput-based** - relies on the client's BW, **buffer-based** - relies on the client's buffer and **hybrid** which take both methods into account. Specific algorithms for each method were implemented and analyzed. We compared the actual traffic volume received by the client, using the prediction algorithm and the ideal traffic possible volume (fig. 2).

We managed to demo adaptive streaming with tolerable delay of up to 3 seconds in various unstable environments.

Our work leads to the following conclusions: Throughput-based algorithms ignore the current status of the live stream which results with an unstable stream in a noisy environment. Buffer-based algorithms do not learn the network's actual BW and leads to a stable live stream while failing to reach the best possible quality. Hybrid algorithms outperforms both previous methods as it keeps a stable stream with higher quality than buffer-based. This approach adapts to the client's network by learning the download rate history *and* taking into account the streams current state.

References

[1] J. Le Feuvre, C. Concolato, and J.-C. Moissinac. Gpac: open source multimedia framework. In *Proceedings of the 15th ACM international conference on Multimedia*, pages 1009–1012, 2007.

[2] I. Sodagar. The mpeg-dash standard for multimedia streaming over the internet. *IEEE multimedia*, 18(4):62–67, 2011.