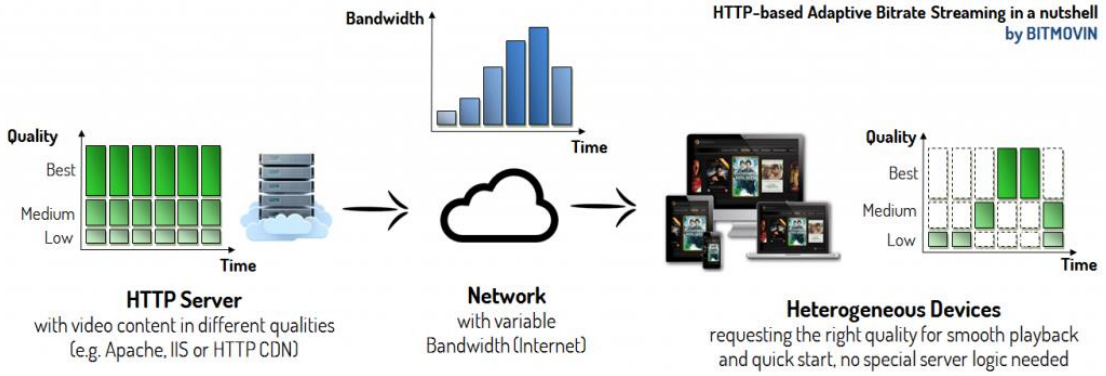
Multiple Media streams synchronization using MPEG-DASH

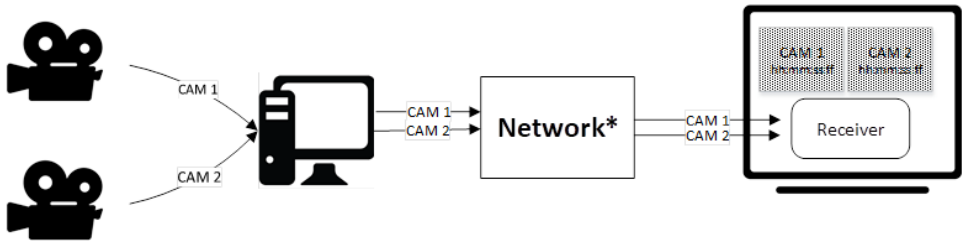
Abstract:

MPEG-DASH *(Moving Picture Experts Group - Dynamic Adaptive Streaming over HTTP)* is a vendor independent, international standard ratified in 2012. One of the main benefits of MPEG-DASH is reduction of startup delays and buffering/stalls during the video and continued adaptation to the bandwidth situation of the client.
Today, MPEG-DASH is gaining more and more deployments, accelerated by services such as Netflix or Google, which recently switched to this new standard. With these two major sources of internet traffic, 50% of total internet traffic is already MPEG-DASH.

The basic idea of MPEG-DASH is as follows: chop the media file into different bitrates or spatial resolutions encoded segments. The segments are provided on a Web server and can be downloaded through HTTP standard compliant GET requests where the HTTP Server serves different qualities, chopped into segments of equal length. Since the client knows its capabilities, received throughput and the context of the user best - the adaptation to the best bitrate or resolution is done on the client side for each segment.



One of the challenges today in the 3D real-time videos and Virtual Reality (VR) is for example to be able to create and manipulate virtual replicas of physical objects in the local environment (Remote collaboration in VR using virtual replicas) VR presents a way for people to truly interact with places far away like surgeon perform surgery from home or an engineer participating in a physical meeting abroad.



Goals:

1. Raise the above setup. For video streams – use live streams or pre-recorded.
2. MPEG-DASH client – use https://github.com/pari685/AStream
3. Video Streams – Tests should include changing the parameters:
    - Frame Rate
    - Bit Rate
    - Bandwidth (stream)
    - Resolution
4. Network – Tests should include changing the parameters (using Mininet):
    - Network Bandwidth
    - Error rate
    - Latency
5. I should be possible to change above parameters on the fly
6. Stretch goal – use QUIC as the transport layer, instead of TCP. Refer to https://bitmovin.com/advanced-transport-options-dash-quic-http2-part-ii

Requirements:

Introduction to Networking (Must), Internet Networking (Optional)

Programming Language:
Python

Guided by:
Boaz Sternfeld, Yaron Honen, Itzik Ashkenazi