

Trick Mode in dashdemux/adaptivedemux

- How to handle fast rates with limited bandwidth/cpu ?
 - Downloading/decoding everything faster would exceed limits
 - Just download keyframes ! And advance by more than 1 step if needed
- DASH fragment header (moof) tells us where keyframes are located
 - Download the fragment header and parse it (isobmff parsing)
 - Decided what to download next (byte range)
 - Like just the keyframes :D

Limitations and WIP

- While just downloading keyframes reduces the bandwidth/cpu usage, it has issues
 - Request latency (you are bounded by how fast you can download)
 - Cpu usage (decoding limitations)
 - Need to behave as a “live” source
 - i.e. no (re)buffering allowed
- Idea is after each keyframe is downloaded we decide what to download next based on:
 - Running download time
 - Current downstream position

Improve adaptivedemux

- Investigation led to figuring out a better decision system
- During hackfest, Stephan led me to latest paper from Akamai/Amazon, also based on experience from Netflix:
 - BOLA: Near-Optimal Bitrate Adaptation for Online Videos
- Close to what I had figured out :)
- Would require queue2 after adaptivedemux, knowing the size and fill level, and then decide what to download next based on heuristics
- Would also help with solving optimal keyframe-only trick-mode