Streaming Video over HTTP with Consistent Quality

ACM MMSys - March 2014

Zhi Li, Ali C. Begen, Joshua Gahm, Yufeng Shan, Bruce Osler and David Oran
Adaptive Video Streaming over HTTP
What is Wrong with Existing HAS Solutions?

- Each segment is more or less constant-bitrate (CBR) encoded
- Client fetches segments based on bitrate information only
- Viewer quality of experience varies because of
  - Low-motion/complexity vs. high-motion/complexity scenes
  - Upshifts and downshifts dictated by the adaptation logic
Thought Process: Shift Bits between Scenes

- If we can steal some bits from the simple scene and stuff them into the complex scene, the overall viewing experience would have been better.
- This boils down to an optimization problem that temporally allocates bits among video segments to yield an optimal overall quality.
Tradeoffs in Adaptive Streaming

Improve
- Proximity to Live Point
- Quality Stability
- Overall Quality

Reduce
- Zapping Time
- Stalls
Dimensions – In-Stream vs. Between-Streams

- Same principle applies to both:
  - In-stream Case: Temporal bit shifting between segments
  - Between-streams Case: Bit shifting between streams sharing a bottleneck link
Scope of Optimization

- Modeling the QoE
- Client-buffer constraints
- Fairness
- Whose responsibility, server, client or network?
In-Stream Bitrate Allocation: Challenges

- Challenge 1: How to measure video quality?
  - How to measure the quality of each segment?
  - Temporal pooling – How a viewer forms an overall impression over a sequence of segments?

- Challenge 2: We must meet client-buffer constraints
  - We must not drain the buffer
  - We must maintain buffer below an upper bound, too

- Challenge 3: Optimization is myopic
  - Client does not know available bandwidth in the future
  - Only a finite horizon of video information might be available
Video Quality – A Generic Framework

- Quality score for a segment: PSNR, -MSE, SSIM, JND, ...
- Temporal Pooling: Possible objective functions
  - Max-Sum: Maximize the sum of (or average) quality over segments
  - Max-Min: Maximize the worst-case quality over segments
- Temporal pooling using $\alpha$-fairness utility function [Srikant’04]

$$\max \sum_{n} U_{\alpha}(Q(n)),$$
where
$$U_{\alpha}(q) := \frac{q^{1-\alpha}}{1-\alpha}$$

- Special cases
  - Max-Sum ($\alpha=0$)
  - Max-Min ($\alpha=\infty$)
  - Proportional fairness ($\alpha=1$)
Bandwidth and Video Bitrate Variability

- Quality optimization poses higher risk of buffer underrun/overshoot than conventional streaming
- We need to
  - Impose lower and upper bounds on buffer evolution
  - Have a fast algorithm to detect bandwidth drops
  - Have proper balance between these two
- Proposed Solution
  - Use a fast algorithm (e.g., PANDA) to quickly detect bandwidth changes
  - Apply an online algorithm to adapt to network bandwidth step by step
  - Use dynamic programming (DP) to program buffer evolution within a sliding window

Reading: “Probe and adapt: rate adaptation for HTTP video streaming at scale,” IEEE JSAC, Apr. 2014
A Toy Example

![Graph showing the relationship between buffer (in seconds) and step with points labeled Q = 1, Q = 2, Q = 2, and Q = 4.](image-url)
Dynamic Programming Solution

- Brute-force search has exponential complexity
  → Dynamic programming reduces processing time to polynomial time

\[ Q^* (B[0] \rightarrow B[N]) = \max_{B[n]} \left\{ Q^* (B[0] \rightarrow B[n]) + Q^* (B[n] \rightarrow B[N]) \right\} \]
Simulation Results – Elysium
Quality-Unaware vs. Mean Quality Optimized vs. Minimum Quality Optimized

[Graphs showing PSNR (dB) and Rate (Mbps) over Time (Sec)]
Simulation Results – Avatar
Quality-Unaware vs. Mean Quality Optimized vs. Minimum Quality Optimized
Demo

• Sample 1: CBR encoded, quality-unaware streaming at 800 Kbps
• Sample 2: VBR encoded, quality-unaware streaming at 800 Kbps
• Sample 3: VBR encoded, consistent-quality streaming at 800 kbps
• Also available at https://sites.google.com/site/cqhttpstreaming
Simulation Results – Avatar
Three PANDA or PANDA/CQ Clients

(a1) Quality (PANDA)

(a2) Rate (PANDA)

(b1) Quality (PANDA/CQ)

(b2) Rate (PANDA/CQ)
Contributions and Future Work

- Contributions
  - We presented the general concept of consistent-quality streaming and outlined a few directions in this space
  - We showed how consistent-quality streaming could be designed and implemented for the single-client case

- Future Work
  - We are looking into low-complexity quality metric(s)
  - We will work on multi-client scenarios
Thank you.