Challenges with Large-Scale Adaptive Streaming Deployments

Ali C. Begen with Co-Authors:
Tim Donahue, Josh Gahm and Zhi Li
HTTP Adaptive Streaming (HAS)
Streaming over HTTP – The Promise

- Leverage tried-and-true Web infrastructure for scaling!
  - Video is just ordinary Web content
- Leverage tried-and-true TCP!
  - Congestion avoidance
  - Reliability
  - No special QoS for video

*It should all “just work” 😊*
Does Streaming over HTTP Scale?

- When HAS clients compete with non-HAS traffic, mostly yes
- But when HAS clients compete with each other for bandwidth, we begin to see problems:
  - The clients’ adaptation behaviors interact with each other:
    - One client upshifts → Other clients get less bandwidth and may downshift
    - One client downshifts → Other clients get more bandwidth and may upshift
  - The competing clients form an “accidental” distributed control-feedback system
    - Such systems often exhibit unanticipated behaviors
    - A variety of such behaviors can be seen with widely deployed HAS clients
- In this talk, we will see examples of these problems, explain the root cause, and describe some solutions
Topics

- Examples: Test-Bed Experiments
- Understanding the Root Cause
- Solutions
  - A Client-Based Solution: Probe AND Adapt (PANDA)
  - Network-Based Approaches
  - Control Plane Approaches
- Conclusions
Examples: Test-Bed Experiments
36 Smooth Clients Sharing 100 Mbps Link
(50 ms RTT, Single RED Queue)
Result: Excessive Shifting, Degraded QoE

Requested Bit Rate vs. Time for Four Individual Smooth Clients in A Population

- Client 0
- Client 1
- Client 2
- Client 3
30 Apple Clients (Lion) Sharing 100 Mbps Link
(50 ms RTT, Single RED Queue)
22 Apple Clients (Mavericks) Sharing 100 Mbps Link (50 ms RTT, Single RED Queue)

Clients seem to “lock in” after a while; persistent unfairness?
Notes on Scope of the Problem

- Similar issues occur across
  - All HAS formats (e.g., Smooth, HLS, HDS, DASH)
  - All HAS clients we have tried so far
  - Just a few or ~50 clients (probably more clients as well)

- Severity of issues is quite variable, depending on many factors
  - Subjective QoE can vary from mildly degraded (Microsoft) to unwatchable (Apple OSX/Lion, Adobe)
  - In general, adding more clients seems to make things worse

- Does not occur when network is overprovisioned

- Cross-traffic mix does affect problem, but problem emerges even with significant cross-traffic
  - Smooth streaming oscillations still intact with 30% Web-like cross traffic
Understanding the Root Cause
Download Rates Experienced by Apple Clients

Observed Download Rates vs. Time for Representative Clients

Fair-share bandwidth
Overestimation of Available Bandwidth due to Link Underutilization

Requested Rate, Download Rate, and Link Utilization for 36 Smooth Clients on a 100Mbps Link

- Avg. Req. Video Rate (2 Sec. Intvl.)
- Avg. Frag. Download Rate (2 Sec. Intvl.)
- Link BW Usage (right scale)
Bandwidth Estimation Cliff

Clients cannot Determine Bandwidth Use until They Use Too Much

- 100 “client” emulator instances
- All clients in “steady state”
- Rate shifting disabled
- 2 second fetch interval
- 100Mbps link $\rightarrow$ fair share $\sim$1Mbps
- Vary fragment size to achieve desired link subscription %

“cliff” almost exactly at full link subscription
A Client-Based Solution: Probe AND Adapt (PANDA)
PANDA – Design Philosophy

- Avoid the root cause that triggers bitrate oscillation
  - Use the TCP throughput measurement only when the link is over-subscribed
- How to tell when the link is under/over-subscribed?
  - Apply “probing” (i.e., small increments of data rate)
  - Additive-Increase, Multiplicative-Decrease (AIMD) for probing (similar to TCP)
- How to continuously vary the data rate (the video bitrate is discrete)?
  - Fine-tune the inter-request time
36 PANDA vs. Smooth Clients Sharing 100 Mbps

PANDA players can effectively stop oscillations!
Network-Based Approaches
Could Network QoS in the CMTS Help?

- Idea: Apply QoS to downstream HAS packet streams to stabilize client rate selections

- Questions:
  - What QoS policy will help?
  - How to recognize which service flows carry adaptive streaming traffic?
  - Can the solution fit within existing platform QoS mechanisms?
  - Can solution work with existing clients?

- We are actively investigating these questions
Control Plane Approaches
I want to make sure that I provide the best possible DASH quality

I want to control the general quality-of-experience of all my subscribers, potentially differentiate and avoid overload and congestion situations

I want to make sure that my cheaper distribution is used when it is available

I want to make sure that my content is protected and does not leak

I want to make sure that my ad is viewed and I know that it is viewed

I want to make sure that the servers in the network are properly used
Everybody says...

I'm not a control freak
I just happen to know what's best for everybody else
How to Control the Streaming Clients?

- (Blind) Bandwidth throttling ×
- MPD offerings and updates ×
- DASH events ×
- HTTP operation (Redirects) ×
- Control plane and session management ✓
Control Plane and Session Management
Server(s) and Network Providing Assistance to Clients

- Control plane that enables to exchange messages between the client and the server
  - Control plane typically has 1:1 correspondence and is bi-directional between client and server
  - Control plane carries operational data in both directions
  - Control plane is independent from the media/manifest distribution
Architecture for SAND in MPEG DASH

DANE: DASH-assisting network element
PER: Parameters for enhancing reception
PED: Parameters for enhancing delivery

DANE (Media Origin)  DANE (Third-Party Server)  Regular Network Element  DASH Client

Media  PER Messages  Metrics and Status Messages  PED Messages
Conclusions

- The promise that rate adaptation should “just work” is problematic once adaptive streaming traffic is predominant
  - Bandwidth cliff effect makes design of adaptation algorithms difficult
- To address the competing client problems, we can use any combination of the following:
  - Client-based approaches
  - Network-based approaches
  - Control plane approaches