An Experimental Evaluation of Rate Adaptation Algorithms in Adaptive Streaming over HTTP

Saamer Akhshabi, Constantine Dovrolis
Georgia Institute of Technology

Ali C. Begen
Cisco Systems

February 24, 2011
ACM Multimedia Systems Conference 2011
Cisco Systems, San Jose, United States
Objectives

• Examine the performance of adaptive streaming over HTTP

• Three important operating conditions
  • How adaptive players react to available bandwidth variations
    • Persistent variations
    • Short-term variations (spikes)
  • How adaptive players compete for available bandwidth
  • How adaptive streaming performs with live content
    • What are the differences with on-demand content?
Outline

• Overview of adaptive streaming over HTTP
• Experimental methodology
• Rate adaptation under available bandwidth variations
  – Microsoft Smooth Streaming player
  – Netflix player
  – Adobe OSMF player
• Competition between two players
• Live streaming
• Conclusions
Outline

• Overview of adaptive streaming over HTTP
  • Experimental methodology
  • Rate adaptation under available bandwidth variations
    – Microsoft Smooth Streaming player
    – Netflix player
    – Adobe OSMF player
• Competition between two players
• Live streaming
• Conclusions
Adaptive Streaming over HTTP

IIS Smooth Streaming

Smooth Streaming Presentation

<table>
<thead>
<tr>
<th>Media Type</th>
<th>File Name</th>
<th>Bit Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audio</td>
<td>Big Buck Bunny_2436000.ismv</td>
<td>64000</td>
</tr>
<tr>
<td>Video</td>
<td>Big Buck Bunny_2436000.ismv</td>
<td>243600</td>
</tr>
<tr>
<td>Video</td>
<td>Big Buck Bunny_1636000.ismv</td>
<td>153600</td>
</tr>
<tr>
<td>Video</td>
<td>Big Buck Bunny_1233000.ismv</td>
<td>123300</td>
</tr>
<tr>
<td>Video</td>
<td>Big Buck Bunny_866000.ismv</td>
<td>86600</td>
</tr>
<tr>
<td>Video</td>
<td>Big Buck Bunny_608000.ismv</td>
<td>60800</td>
</tr>
<tr>
<td>Video</td>
<td>Big Buck Bunny_427000.ismv</td>
<td>42700</td>
</tr>
<tr>
<td>Video</td>
<td>Big Buck Bunny_300000.ismv</td>
<td>30000</td>
</tr>
</tbody>
</table>

300 Kbps – 2 Mbps
Variable Bandwidth

300 Kbps
Low Bandwidth

2 Mbps
High Bandwidth

From IIS Smooth Streaming Website
Adaptive Streaming over HTTP: Manifest File and Fragments

<SmoothStreamingMedia MajorVersion="1" Duration="150483666" …>

<StreamIndex Type="video" Chunks="52"
Url="QualityLevels({bitrate})/Fragments(video={start time})" …>

  <QualityLevel Bitrate="3450000" Width="1280" Height="720" …/>
  <QualityLevel Bitrate="1950000" Width="848" Height="480" …/>
  <QualityLevel Bitrate="1250000" Width="640" Height="360" …/>
  …….

  <c n="0" d="9342667" />
  <c n="1" d="5338666" />
  <c n="2" d="11678334" />
  …….
Outline

- Overview of adaptive streaming over HTTP
- Experimental methodology
- Rate adaptation under available bandwidth variations
  - Microsoft Smooth Streaming player
  - Netflix player
  - Adobe OSMF player
- Competition between two players
- Live streaming
- Conclusions
Experimental Methodology
Outline

- Overview of adaptive streaming over HTTP
- Experimental methodology
- Rate adaptation under available bandwidth variations
  - Microsoft Smooth Streaming player
  - Netflix player
  - Adobe OSMF player
- Competition between two players
- Live streaming
- Conclusions
Smooth Streaming Player
Smooth Streaming Player

• Sample HTTP Request:
  - GET
    /mediadl/iisnet/smoothmedia/Experience/BigBuckBunny720p.ism/QualityLevels(2040000)/Fragments(video=400000000)
  HTTP/1.1
Smooth Streaming Player
Buffering and Steady State

- One fragment per HTTP request
- No HTTP pipelining

Two states:
1. Buffering state
   - Request fragments as fast as possible
2. Steady-state
   - Request new fragment every T seconds
Smooth Streaming Player
Behavior under Unrestricted Available Bandwidth

- Average throughput: running average of two-second TCP throughput measurements.
- Fragment throughput: per-fragment throughput measurement
• Two successive, say video, requests sent at times $t_1$ and $t_2$ ($t_1 < t_2$) with timestamps $t_1'$ and $t_2'$ ($t_1' < t_2'$) respectively

• The playback buffer size (in seconds) for video at time $t_2$ is estimated as:

$$B(t_2) = B(t_1) - (t_2 - t_1) + (t_2' - t_1')$$
Smooth Streaming Player Behavior Under Persistent Changes in Available Bandwidth

- Rate adaptation occurs after long delays
- The player estimates available bw using a running average of the per-fragment TCP throughput measurements
Smooth Streaming Player

Playback Buffer Size under Persistent Changes in the Available Bandwidth

- Playback buffer size decreases when available bandwidth is less than the requested bitrate.
- Playback buffer size increases when player goes into “buffering state” requesting fragments as fast as possible.
  - Together with switching to bitrate < available bw.
The client reacts to the spikes by switching to a lower bitrate too late.

Stays at that bitrate for long after the spike has passed.
Outline

• Overview of adaptive streaming over HTTP
• Experimental methodology
• Rate adaptation under available bandwidth variations
  • Microsoft Smooth Streaming player
  • Netflix player
  • Adobe OSMF player
• Competition between two players
• Live streaming
• Conclusions
Netflix Player
Behavior under Unrestricted Available Bandwidth

- Player accumulates 5-min playback buffer!
Occasionally, the player requests a higher bitrate than available bw!

Utilize large playback buffer size to optimize video quality
Outline

• Overview of adaptive streaming over HTTP
• Experimental methodology
• Rate adaptation under available bandwidth variations
  • Microsoft Smooth Streaming player
  • Netflix player
  • Adobe OSMF player
• Competition between two players
• Live streaming
• Conclusions
Adobe OSMF Player

Akamai HD for Adobe Flash Platform 2.0
with Adobe HTTP Dynamic Streaming

Key Statistics
- BUFFER LENGTH (s): 5.91
- MEMORY (MB): 40.86
- DOWNLOAD RATIO: 56.8
- DROPPED FRAMES: 7
- INDEX: 7 of 7
- PLAYBACK (kbps): 2657
- CURRENT FPS: 24.34
- DIMENSIONS: 1280x720

Bitrate playing: 3000 kbps

Select Video
- Robin Hood
- Amour Imaginaire
- Freeway
- Beach scene
- Matthew Sweet
- Train
- Life is
The client often fails to select the highest possible bitrate for the given available bandwidth.

Also, player often oscillates between bitrates, mostly the lowest and the highest bitrates.
Outline

• Overview of adaptive streaming over HTTP
• Experimental methodology
• Rate adaptation under available bandwidth variations
  • Microsoft Smooth Streaming player
  • Netflix player
  • Adobe OSMF player
• Competition between two players
• Live streaming
• Conclusions
Two Smooth Streaming Players Compete

- Fairness issue: one stream may get much lower bitrate than the other
- Players can get into oscillation between bitrates even when available bw is constant
- Synchronization can cause simultaneous bitrate drops
Outline

- Overview of adaptive streaming over HTTP
- Experimental methodology
- Rate adaptation under available bandwidth variations
  - Microsoft Smooth Streaming player
  - Netflix player
  - Adobe OSMF player
- Competition between two players
- Live streaming
- Conclusions
Smooth Live Streaming
Playback Buffer Size

- Player starts streaming with 8-seconds delay
- Playback delay increases over time whenever playback buffer gets empty
  - Player does not skip fragments
Outline

• Overview of adaptive streaming over HTTP
• Experimental methodology
• Rate adaptation under available bandwidth variations
  • Microsoft Smooth Streaming player
  • Netflix player
  • Adobe OSMF player
• Competition between two players
• Live streaming
• Conclusions
Summary of the Key Differences Between Players

• Smooth Streaming player
  – Playback buffer size of 10s of seconds
  – Conservative in selecting bitrate
    (bitrate < available bw)

• Netflix player
  – Playback buffer size of few minutes
  – More aggressive than Smooth player
    (sometimes bitrate > available bw)

• OSMF player
  – Erratic bitrate selection
  – Is open source and requires customization
Research Challenges for Adaptive Streaming over HTTP

- Reducing the large delay in responding to persistent available bw variations
- Correcting erratic rate adaptations under short-term variations
- Avoiding oscillations and unfairness when multiple players compete
- Improving the performance of live streaming
Ongoing Work

• Continue the analysis of commercial players to understand how they work
  – And identify weaknesses
• Expand study of multiple player competition
• Design and implement an adaptive steaming adaptation logic that can address all previous issues
Questions