Performance Characterization of a Commercial Video Streaming Service

Mojgan Ghasemi, Princeton University

P. Kanuparthi, A. Mansy, T. Benson, J. Rexford

1Yahoo, 2Duke University, 3Princeton University
• First study to measure **both** sides

• Video makes up **70%** of the traffic!
Yahoo’s Video Streaming System
Yahoo’s Video Streaming System

- Client receives the manifest
Yahoo’s Video Streaming System

- HTTP requests for chunks share a TCP connection
- Each chunk is 6 seconds
Yahoo’s Video Streaming System

- CDN servers use Apache Traffic Server (ATS), LRU policy
Yahoo’s Video Streaming System

- Chunks pass client’s “download” and “rendering” stack
Our Dataset: Yahoo Videos
Yahoo Videos

Yahoo! News

Yahoo! Sports

Yahoo! Finance
Our Dataset

- **VoD Dataset:**
  - Over 18 days, Sept 2015
  - 85 CDN servers across the US
  - 65 million VoD sessions, 523m chunks

- **Users:**
  - Non-mobile users, no proxy
  - Predominantly in North America (over 93%)

- **Video Streams:**
  - Popularity: 66% of requests for 10% of titles
  - Duration: most videos less than 100 sec
Our Goal

Identify performance problems that impact video

Player ← Network → CDN
Our Goal

Identify performance problems that impact video
Our Goal

Identify performance problems that impact video

A content provider (e.g., Yahoo) controls “both sides”
Our Approach: e2e Per-chunk Measurement

- **End-to-end**
  - Instrumenting both sides (player, CDN servers)

- **Per-chunk**
  - Unit of decision making (e.g., bitrate, cache hit/miss)
  - Sub-chunk is too expensive

- **TCP statistics**
  - Sampled from CDN host’s kernel
  - Operational at scale
Our Approach: e2e Per-chunk Measurement

<table>
<thead>
<tr>
<th>Player</th>
<th>OS</th>
<th>WAN</th>
<th>CDN</th>
<th>Backend</th>
</tr>
</thead>
</table>


Our Approach: e2e Per-chunk Measurement
Our Approach: e2e Per-chunk Measurement
Our Approach: e2e Per-chunk Measurement

\[ D_{FB} \quad \text{HTTP Get} \quad D_{CDN} + D_{BE} \quad \text{Cache miss} \quad D_{DS} \]
Our Approach: e2e Per-chunk Measurement

![Diagram showing the path of data from Player to Backend via OS, WAN, CDN.](#)
Studying QoE Factors Individually

Factors:

- Video startup time
- Rebuffering rate
- Video quality (bitrate, framerate)

We look at individual metrics, because:

- Type of content
- Length of video
Outline

- Introduction
- Measurement Dataset
- Server-side Problems
- Network Performance Problems
- Client’s Performance Problems
- Take-aways and Conclusions
Server-side Performance Problems
Monitoring CDN Performance

Direct measurement

Player
• Session ID
• Chunk ID
• Startup time
• Re-buffering
• Video quality

CDN
• Session ID
• Chunk ID
• Server latency ($D_{CDN}$)
• Backend latency ($D_{BE}$)
• Cache hit/miss
Monitoring CDN Performance

Direct measurement

Player
- Session ID
- Chunk ID
- **Startup time**
- Re-buffering
- Video quality

CDN
- Session ID
- Chunk ID
- **Server latency** ($D_{CDN}$)
- Backend latency ($D_{BE}$)
- Cache hit/miss
Impact of CDN on Startup Time

- Only possible via data from "both ends"
- Startup time vs. server latency in first chunk
1. Cache Misses

- Cache misses increase server latency
  - 40X median, 10X average
- Server latency can be worse than network
  - Caused by cache misses (40% miss rate)
2. Persistent Problems in Unpopular Videos

- Cache misses are **persistent**:  
  - Average: 2%  
  - After one miss: 60%

- **Unpopular** titles have significantly higher cache misses
Network Performance Problems
Network Measurement

CDN’s host kernel

Polling tcp_info
Every 500ms per-chunk

tcp_info maintained by OS:
- weighted average of RTT (SRTT)
- congestion window
- packet retransmissions

Challenges:
- Smoothed average of RTT: SRTT
- Infrequent network snapshots
- Packet traces cannot be collected
1. Network Latency Problems

- **Persistent high latency:**
  - /24 IP prefixes, recurring in 90\textsuperscript{th} percentile
  - 25\% of prefixes are located in the US, with the majority close to CDN nodes

- **High latency variation:**
  - Enterprise networks have higher latency variation
2. Earlier Packet Losses Cause More Rebuffering

- Packet loss is more common in the first chunk (4.5X)
- Packet loss in the first chunk causes more rebuffering
3. Throughput is a Bigger Problem than Latency

\[ \text{perf}_{\text{score}} = \frac{\text{chunk duration}}{D_{\text{FB}} + D_{\text{LB}}} \]

- \( D_{\text{FB}} \): measure of latency, \( D_{\text{LB}} \): measure of throughput

![Diagram showing flow of network components from Player to Backend with delays labeled as \( D_{\text{FB}}, D_{\text{CDN}} + D_{\text{BE}}, D_{\text{DS}} \).]
3. Throughput is a Bigger Problem than Latency

\[
\text{perf}_{\text{score}} = \frac{\text{chunk duration}}{D_{FB} + D_{LB}}
\]

- \(D_{FB}\): measure of latency, \(D_{LB}\): measure of throughput
- \(\text{perf}_{\text{score}} > 1\): More than 1 sec of video delivered per sec
- \(\text{perf}_{\text{score}} < 1\): Less than 1 sec of video per sec
3. Throughput is a Bigger Problem than Latency

\[ \text{perf}_{\text{score}} = \frac{\text{chunk duration}}{D_{FB} + D_{LB}} \]

- \(D_{FB}\): measure of latency, \(D_{LB}\): measure of throughput
- \(\text{perf}_{\text{score}} > 1\): More than 1 sec of video delivered per sec
- \(\text{perf}_{\text{score}} < 1\): Less than 1 sec of video per sec

\(D_{LB}\) has a major contribution (orders of magnitude)
Client’s Download Stack
Performance Problems
Download Stack Latency

- Cannot observe download stack latency ($D_{DS}$) directly
- Detecting “outliers”

$$D_{FB_i} > \mu_{DFB} + 2 \cdot \sigma_{DFB}$$
Download Stack Latency

- Cannot observe download stack latency ($D_{DS}$) directly
- Detecting “outliers”

\[ D_{FB_i} > \mu_{DFB} + 2 \cdot \sigma_{DFB} \]
\[ TP_{inst_i} > \mu_{TP_{inst}} + 2 \cdot \sigma_{TP_{inst}} \]
Download Stack Latency

- Cannot observe download stack latency \((D_{DS})\) directly
- Detecting “outliers”

\[
D_{FB_i} > \mu_{D_{FB}} + 2 \cdot \sigma_{D_{FB}}
\]

\[
TP_{inst_i} > \mu_{TP_{inst}} + 2 \cdot \sigma_{TP_{inst}}
\]

*Similar network and server performance*
Download Stack Latency: Case Study

![Graph showing latency over chunk ID](image1)

- srtt
- server
- first-byte delay

![Graph showing TP over chunk ID](image2)

- connection TP
- download TP
Client’s Download Stack Problems

- **Transient:**
  - Outlier: 1.7M chunks (0.32%)
  - *First* chunks have higher $D_{DS}$

- **Persistent:**
  - In most cases, $D_{DS}$ is higher than network and server latency
Client’s Rendering Stack
Performance Problems
Rendering Stack

- If CPU is busy, rendering quality drops (high frame drops)
- If video tab is not visible, browser drops frames
- Per-chunk data: \( \text{vis} \) (is player visible?), dropped frames
- Per-session data: OS, browser
1. Good Rendering Requires \(1.5 \text{ sec/ssec}\) Download Rate

- De-multiplexing, decoding, and rendering takes time.
2. Higher Bitrates Show Better Rendering

Paradox:

- Higher bitrates put more load on the CPU
- Showed better rendering framerate

Higher bitrates are often requested in connections:

- Lower RTT variation
- Lower retransmission rate
3. Unpopular Browsers Have Worse Rendering

- Chunks with good performance ($rate > 1.5\frac{sec}{sec}$)
- Player is visible (i.e., $vis = true$)
3. Unpopular Browsers Have Worse Rendering

- Chunks with good performance \((rate > 1.5 \frac{sec}{sec})\)
- Player is visible (i.e., \(vis = true\))
Take-aways
### Take-aways: CDN

<table>
<thead>
<tr>
<th>Problem</th>
<th>Take-away</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache miss impact</td>
<td>Cache-eviction policy</td>
</tr>
<tr>
<td>Cache miss persistence</td>
<td>Pre-fetch subsequent chunks</td>
</tr>
</tbody>
</table>
## Take-aways: Network

<table>
<thead>
<tr>
<th>Problem</th>
<th>Take-away</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nearby clients with high latency</td>
<td>Avoid over provisioning servers for nearby clients</td>
</tr>
<tr>
<td>Prefixes with persistent high latency or variation</td>
<td>Adjust ABR algorithm accordingly (more conservative bitrate, increase buffer size)</td>
</tr>
<tr>
<td>Throughput the major bottleneck</td>
<td>Good news for ISPs (e.g., establish more peering points)</td>
</tr>
</tbody>
</table>
## Take-aways: Client

<table>
<thead>
<tr>
<th>Problem</th>
<th>Take-away</th>
</tr>
</thead>
<tbody>
<tr>
<td>Download stack latency</td>
<td>Can cause over-shooting or under-shooting by ABR, incorporate server-side TCP metrics</td>
</tr>
<tr>
<td>Rendering is resource-heavy</td>
<td>Use $1.5 \frac{sec}{sec}$ video arrival rate as a rule-of-thumb</td>
</tr>
</tbody>
</table>
Conclusion

- Instrumenting **both sides**
  - Uncover range of problems for the first time
- **Per-chunk** and per-session data
  - Uncover “persistent” vs. “transient” problems
- Our findings have been used to enhance performance in Yahoo
Thank You!
Network Problems Impact QoE

- Data from “both sides” show the impact
- Startup time vs. SRTT of first chunk
- Network latency significantly impacts video startup time