Characterizing Netflix Bandwidth Consumption

Dr. Jim Martin
Associate Professor
School of Computing
Clemson University
jim.martin@cs.clemson.edu
http://www.cs.clemson.edu/~jmarty

Students:
Yunhui Fu (yfu@clemson.edu)
Nicholas Wourms
(nwourms@clemson.edu)

Terry Shaw
Director, Network Systems
CableLabs
t.shaw@cablelabs.com

Presented at IEEE CCNC 2013 on January 13, 2013
Agenda

- Background
- Objectives
- Methodology
- Results and Analysis
- Conclusions
- Future Work
Let’s rewind a bit:

- Video streaming over the Internet has evolved from UDP-based streaming, to TCP-based progressive downloading, to adaptive HTTP-based streaming
  - Warning: Different from IPTV!!

Several well established methods for adaptive HTTP streaming:

- Adobe’s HTTP Dynamic Streaming (HDS)
- Apple’s HTTP Live Streaming (HLS)
- Microsoft’s Smooth Streaming (SmoothHD)

MPEG organization issued a Call for Proposal for an HTTP Streaming Standard (2009).

- Resulting standard is MPEG-DASH over HTTP and has been published as standard ISO/IEC 23009-1.
- The 3GPP community has been involved since 2009. 3GPP-DASH is in 3GPP standards TS 26.234 and 3GPP TS 26.244.
A segment is an independent, viewable period of video/audio/timing data.
- Segment size of 2 seconds or 10 seconds is reasonable.
- Segments are uniquely identified by an HTTP URL.
- A client requests the segment, the bit rate, and optionally a specific byte range in the segment.
  - Clients can issue requests and receive segments over any number of concurrent TCP connections.
- The video segment is sent back by the HTTP server in a ‘burst’.
- The implementation of the client determines how frequently segments are requested, when bit rate adaptation occurs, and the overall ‘sensitivity’ of the application to network congestion.
Background: DASH Behavior

- Looking at the first 80 seconds of a Netflix session on a Windows Desktop device.
- There are multiple ‘startup’ TCP/HTTP connections that establish the session (Netflix servers).
- Once the session is established, a CDN server is selected, and this client initiates a single TCP connection to transfer ALL content (video and audio).
- Trace captured at the client network (all TCP flows captured).
- We see a high initial throughput as the client fills the playback buffer.
- This is followed by a steady state throughput of 4.39 Mbps.
- Upstream throughput during the 80 seconds was about 1 Kbps.
- The ratio of downstream traffic to upstream traffic was 3336/1.
- The ratio of packets sent downstream to upstream was 2.68.

TCP Cx 1

- Buffering state (67.02 Mbps between 10-30 seconds)
- Steady state (4.39 Mbps)

1.0 second samples
10 second samples

Buffering state

10 20 30 40 50 60 70 80
0
10
20
30
40
50
60
70
Time (seconds)
Throughput (Mbps)

TCP Throughput of single Cx used for content (Windows device, high speed Internet access (>100Mbps), very good network conditions)
Agenda

- Background
- Objectives
- Methodology
- Results and Analysis
- Conclusions
- Future Work
Objectives

- Characterize the bandwidth consumption and behaviors of an early DASH implementation: Netflix
  - bandwidth consumed, use of multiple TCP connections
- Explore design/implementation choices:
  - differences across a range of client devices
  - size of playback buffer
Agenda

• Background
• Objectives
• Methodology
• Results and Analysis
• Conclusions
• Future Work
Methodology

- Conduct controlled experiments using live Netflix sessions.
  - Analyzed tcpdump packet captures
  - Used four different client devices

Client Devices
1. Xbox Wired
2. Windows Wired
3. Xbox WiFi
4. Roku WiFi
5. Android WiFi
**Methodology: Measurement Scenarios**

<table>
<thead>
<tr>
<th>ID</th>
<th>Scenario Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Ideal with 3% artificial loss (5 min no impairment, 5 min 3% loss, 5 min no impairment).</td>
</tr>
<tr>
<td>2</td>
<td>Ideal with 40% artificial loss (5 min no impairment, 5 min 40 % loss, 5 min no impairment).</td>
</tr>
<tr>
<td>3</td>
<td>Stepped loss (200 seconds at 0% loss, 100 seconds at 5% loss, 100 seconds at 10% loss, 100 seconds at 0% loss).</td>
</tr>
<tr>
<td>4</td>
<td>Chaotic loss (300 seconds at 0% loss, 300 seconds at variable loss, 300 seconds at 0% loss).</td>
</tr>
</tbody>
</table>
Agenda

• Background
• Objectives
• Methodology
• **Results and Analysis**
• Conclusions
• Future Work
Results: Scenario 1

(a) Trace 1-1 (Xbox Wired)  
(b) Trace 1-5 (Android Wifi)
Results : Scenario 2

a. Trace 2-1 (Xbox wired)

b. Trace 2-2 (Windows wired)
Agenda

- Background
- Objectives
- Methodology
- Results and Analysis
- Conclusions and Future Work
Conclusions

• The basic response to network congestion is similar across the devices that were studied. However, some differences:
  • The steady state bandwidth consumption achieved by different devices during periods of sustained congestion varied.
  • Implementations vary in how aggressive they are to utilize available TCP bandwidth after congestion (and adaptation) has occurred.
• Use of multiple TCP Connections varies widely across devices.
• The playback buffer sizes ranged from 30 seconds (on the Android WiFi device) to 4 minutes (on the Windows Wired device).
Future Work

• It is difficult to address the issue of ‘is the adaptation doing the right thing’ without taking into account perceived video quality.
• Prior subjective video quality studies suggest that the quality perceived by the end user is assumed ‘better’ if the rendering uses lower bitrate representations so as to avoid re-buffering stalls and frequent adaptation.
  • Our next step is to validate this conjecture
Appendix - All Results
Measurement Scenario 1 Results

(a) Trace 1-1 (Xbox Wired)

(b) Trace 1-2 (Windows Wired)

(c) Trace 1-3 (Xbox WiFi)

(d) Trace 1-4 (Roku WiFi)

(e) Trace 1-5 (Android WiFi)
Measurement Scenario 2 Results

(a) Trace 2-1 (Xbox Wired)
(b) Trace 2-2 (Windows Wired)
(c) Trace 2-3 (Xbox WiFi)
(d) Trace 2-4 (Roku WiFi)
(e) Trace 2-5 (Android WiFi)
Measurement Scenario 3 Results

(a) Trace 3-1 (Xbox Wired)
(b) Trace 3-2 (Windows Wired)
(c) Trace 3-3 (Xbox WiFi)
(d) Trace 3-4 (Roku WiFi)
Measurement Scenario 4 Results

(a) Trace 4-1 (Xbox Wired)
(b) Trace 4-2 (Windows Wired)

(c) Trace 4-3 (Xbox WiFi)
(d) Trace 4-4 (Roku WiFi)