Modeling, Identifying, and Emulating

Dynamic Adaptive Streaming over HTTP (DASH)

MS Thesis Presentation
by
Andrew Reed

March 31, 2014
What is DASH?

- A method to stream video used by...

  ![Logos of Netflix, Amazon Instant Video, ABC, NBC, The CW TV Now]
What is DASH?

- According to the DASH Industry Forum it’s…

      THE FUTURE OF VIDEO

      Addressing the dramatic growth of broadband video by defining a universal delivery format that provides end users with the best possible video experience by dynamically adapting to changing network conditions.

- Sandvine 2013 – Netflix accounts for 31.6% of all traffic to households in North America
Motivation – Researcher’s POV

Why study DASH?

- **Observe** its behavior “in the wild”
- **Characterize** its impact on network performance
- **Discover** methods to improve video quality
- **Study** its effect on user privacy
Motivation – User’s POV

If you have promised your 2 year old that he can watch Transformers…
... you don’t want this.
And when he finally goes to sleep and you get to watch *The Avengers* in **HD** on your high speed cable connection…
Motivation – User’s POV

... you don’t want this.
Key Takeaway

- The Real-time Transport Protocol (RTP/UDP) is **no longer** the go-to standard for video streaming.

Industry is using **HTTP/TCP**.
Outline

- DASH Overview
- Research Problems
- Thesis Statement
- Background
- **Identifying** DASH
- **Emulating** DASH
  - Modeling in both
- Future Work
- Summary
- Questions
DASH Overview
DASH – Technical Overview

1. Request Video
2. Use CDN 3 with Token T
3. Fetch Video with Token T
4. Token Verified – Deliver Video

DASH – Benefits

- Content is “streamed” from standard HTTP servers
  - Videos can be served by Content Distribution Networks (CDNs)
- All decisions are made by the streaming client
  - “Streaming” servers perform minimal processing
- Streaming client buffers a limited amount of video
  - Bandwidth has not been wasted if a viewer cancels playback
Research Problems
Research Problems: *Identification*

- **Problem:** DASH streams look like standard HTTP traffic
- **Issues:**
  - Difficult for researchers to isolate and study DASH streams given an anonymized, header-only trace
  - DASH traffic has the potential to skew studies of normal browsing activity
Research Problems: Experimentation

**Problems:**

- *Copyright law* – a researcher cannot distribute a DASH data set consisting of copyright material
- *Labor- and storage-intensive* – even a short video results in gigabytes of data

**Issues:**

- Popular videos are rarely used as data
- Most experiments use only a handful of videos
  - Even 2 full-length videos (~21 GB) would present a challenge for an environment such as the Global Environment for Network Innovations (GENI)
A note on Constant Bitrate (CBR) vs. Variable Bitrate (VBR)

![Graph showing Video Bitrate vs. Seconds into Video for Netflix: Avengers, Netflix: Hunger Games, and Amazon: Hunger Games]
Thesis
Thesis Statement

- Application data unit (ADU)-level analysis of captured Dynamic Adaptive Streaming over HTTP (DASH) streams will enable us to develop a model of DASH traffic that can be leveraged to identify DASH source IP addresses in anonymized, header-only traces.

- Furthermore, an ADU-centric representation of DASH videos will enable us to design a lightweight, highly-configurable, distributed DASH emulator.
Background
**tmix – a-b-t Connection Vectors**

- Creates synthetic workloads by replaying the sequence of **a-b** exchanges using dummy payloads that are separated by **t** intervals.

Image from [http://groups.geni.net/geni/wiki/TmixDetails](http://groups.geni.net/geni/wiki/TmixDetails)
Tmix assumes that the ADU sizes \((a, b)\) and the inter-exchange times \((t)\) are constant

- Not true for DASH

**\(a-b-t\)** model can be augmented so that each \(b\) is a **set** of sizes that represent the options for a given video segment
adudump

- Analyzes TCP/IP headers and generates *a-b-t* connection vectors for every exchange in a TCP connection…

  …*in one pass!*

<table>
<thead>
<tr>
<th>Direction</th>
<th>Timestamp</th>
<th>Local IP + Port</th>
<th>Remote IP + Port</th>
<th>Size</th>
<th>SEQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADU:</td>
<td>1202706002.688748</td>
<td>1.2.3.4.443</td>
<td>&lt; 5.6.7.8.62015</td>
<td>163</td>
<td>SEQ</td>
</tr>
<tr>
<td>ADU:</td>
<td>1202706002.733813</td>
<td>1.2.3.4.443</td>
<td>&gt; 5.6.7.8.62015</td>
<td>2886</td>
<td>SEQ</td>
</tr>
<tr>
<td>ADU:</td>
<td>1202706002.738254</td>
<td>1.2.3.4.443</td>
<td>&lt; 5.6.7.8.62015</td>
<td>198</td>
<td>SEQ</td>
</tr>
<tr>
<td>ADU:</td>
<td>1202706002.801408</td>
<td>1.2.3.4.443</td>
<td>&gt; 5.6.7.8.62015</td>
<td>59</td>
<td>SEQ</td>
</tr>
</tbody>
</table>

Hadoop

adudump log

Data loading step

Node 1
Slice of input

Node 2
Slice of input

Node 3
Slice of input

Hadoop

Key Takeaway

The combination of *adudump* and Hadoop provides:

- Distributed, replicated storage
- An architecture where *computation scales with added storage*
- A straightforward method to analyze header-only traces *by TCP connection*
Identifying DASH (ongoing work)
Model of a DASH Connection: DASH Traffic Properties

- Outbound ADUs are sent at regular intervals
- The sizes of the outbound ADUs exhibit low variance
- The maximum size for a segment is capped
- The average inbound data rate is roughly equivalent to the bitrate of the video
Model of a DASH Connection: Gathering Baseline Data
### DASH Traffic Model Measurement

<table>
<thead>
<tr>
<th>Measurement</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average ADU Out (bytes)</td>
<td>433</td>
<td>570</td>
</tr>
<tr>
<td>ADU Out Standard Deviation (bytes)</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>Average Interval (seconds)</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>Interval Standard Deviation (seconds)</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Max ADU In (bytes)</td>
<td>481,107</td>
<td>3,275,999</td>
</tr>
<tr>
<td>Average Data Rate (Kbits/s)</td>
<td>469</td>
<td>3,095</td>
</tr>
<tr>
<td>Data Rate Standard Deviation (Kbits/s)</td>
<td>174</td>
<td>2,145</td>
</tr>
</tbody>
</table>
Model of a DASH Video

- Video segment sizes for each bitrate encoding can be interleaved into a single, sequential ordering (i.e. a **fingerprint**)

- For example, a video with 3 Segments across 2 Bitrates would have a fingerprint of the form

  \[ \{ S_1B_1.size, S_1B_2.size, S_2B_1.size, S_2B_2.size, S_3B_1.size, S_3B_2.size \} \]

- Fingerprints are easy to create
DASH Identification Steps

1. Filter HTTP connections based on the model of a DASH connection
   • This will identify potential DASH connections

2. For each potential DASH connection:
   • Compute the Longest Common Subsequence (LCS) between the potential connection and each fingerprint in a “database”
   • The fingerprint that yields the “longest” LCS is a match
DASH Identification: Future Work

- Validation Testing
- Performance Testing
- Case Studies
  - “Binge watching”
Emulating DASH
Emulate DASH services that use:

- **Fixed-duration** video segments
- **Fixed-duration** audio segments
- **Constant-bitrate** audio segments

**Not** within scope:

- Designing a DASH client emulator that replicates the **exact** behavior of a **specific** client
DASH Traffic Observations

Point at which all three playbacks stabilize to the high definition encoding.

1. A single bitrate encoding can be represented by the sequence of video segment sizes
2. Lower bitrates are well-approximated as a constant percentage less than the highest bitrate*

* Observed for **Netflix** and **Amazon**, which both use **Microsoft Silverlight**
Service-Wide Profiles

For Netflix:

<table>
<thead>
<tr>
<th>Row 1: Video Profile Bitrate</th>
</tr>
</thead>
<tbody>
<tr>
<td>3000</td>
</tr>
<tr>
<td>100 78.333 58.333 35 25 18.666 12.5 7.8333</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>135100</td>
</tr>
</tbody>
</table>

For Amazon:

<table>
<thead>
<tr>
<th>Row 2: Encoding Levels</th>
</tr>
</thead>
<tbody>
<tr>
<td>6000</td>
</tr>
<tr>
<td>100 66.666 41.666 33.333 22.5 15 10 7.5 5 2</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>33090</td>
</tr>
</tbody>
</table>

- **Row 1:** Video Profile Bitrate
- **Row 2:** Encoding Levels
- **Row 3:** Video Segment Duration
- **Row 4:** Video-to-Audio Ratio
- **Row 5:** Audio Segment Size
Video Profiles

- Sequence of video segment sizes for the highest bitrate
- Like a fingerprint, these are easy to create

*The Vampire Diaries*, Season 1, Ep. 1 (Netflix)

*Lost*, Season 1, Ep. 1 (Netflix)
dashem

- Open source
- Non-copyright (CC0 1.0 Universal)
- Written in Java
- Runs from the command line
- Inspired by “real” DASH clients, but does not replicate a specific client
**dashem:**
Example Experimental Design

**Central Server**
- Service-wide profile
- CDN lists
- Video profiles

**CDNs**
- Host a single 14MB file for dummy traffic

*Just HTTP servers serving static files!*

---

Diagram:
- North Region: dashem Clients
  - Traffic Controller
    - North Region: CDN A
    - North Region: CDN B
  - Traffic Controller
  - Central Server
- South Region: dashem Clients
  - Traffic Controller
    - South Region: CDN C
    - South Region: CDN D
dashem: Command Line Parameters

- **Central server address.** Either the IP address or domain name of the central server.

- **Service.** The DASH service to use for the given instance. This allows a single central server to host profiles and CDN lists for any number of services.

- **Region.** The notional geographic region for the given instance.

- **Video Title.** The name of the video profile to stream.

- **Account Name.** Account names are used in logs and can be used by an experimenter to create unique identifiers for each instance of dashem.
Sample Experiment

- Investigate the “Downward Spiral Effect” described by Huang et al.

- Scenario: North American household streaming two videos simultaneously
  - DSL bandwidths: 1 Mbps up / 6 Mbps down
  - RTT: 80 ms

- *Hunger Games* followed 4 minutes later by an episode of *Curious George*

- Ran test twice: once using Netflix data and once using Amazon data
To ensure that *dashem* could select from the same bitrates in both tests:

- Modified row 2 to match Netflix’s video bitrates
- Modified row 5 to match Netflix’s audio bitrate
Sample Experiment - Results

Netflix

Amazon
Sample Experiment - Results

- Poor bandwidth estimation during steady state playback after *Curious George* has ended
- Each video segment request is restarting from TCP slow start
dashem Uses

■ Experiments
  • *Multiple* videos
  • *Multiple* service configurations
  • *Multiple* viewers
  • *Multiple* network conditions
  • *Multiple* networking strategies

■ Bulk DASH traffic generation
Suggested Future Work
Identification

- Study the behavior of DASH clients “in the wild”
- Assess effect on the campus network
- Characterize viewing habits
  - Targeted Ads
- Online Identification
  - OpenFlow
  - Firewalls
Experimentation / Emulation

- Android Port with GUI
- GENI
  - RSpecs for various experimental designs
  - Setup scripts
  - Wiki Tutorial
  - GENI Engineering Conference Presentation
- Classroom Instruction
Summary
Summary

- It appears as if DASH videos are quite easy to identify
- “Real world” videos can be used as data for DASH experiments

Opportunities for DASH research await you!
Questions?