Computer Networks: Internet Video

Ankit Singla

Photo: ETH Zürich / Gian Marco Castelberg
Where we are in the course ...

Part 1: Overview & Principles

Part 2: Applications

Part 3: Transport

Part 4: Algorithms
Computer networks

Part 2: Applications

#1 DNS: How do we name and discover services?

#2 The Web: How do you see weather.com?

#3 Video: How does video streaming work?
We want the highest video quality
Without seeing this ...
Why do we care?

Mark Cuban: Only a 'Moron' Would Buy YouTube

Data: Cisco VNI. 2016 to 2020 forecasted.
Naive approach

Progressive Video file

1280 x 720 pixels

Same file size for every device & screen size

[bitmovin.com]
OK, suppose you’re Netflix

How would you deliver “Mr. Robot” to your subscribers?
End-end workflow

[Adapted from: Adaptive Streaming of Traditional and Omnidirectional Media, Begen & Timmerer, ACM SIGCOMM Tutorial, 2017]
Broad solution approach

- Encode video in multiple bitrates
- Replicate using a content delivery network
- Video player picks bitrate adaptively
  - Estimate connection's available bandwidth
  - Pick a bitrate $\leq$ available bandwidth
Encoding

Replication

Adaptation
1920 x 1080 px

Fast Internet

Screen size: 1920 x 1080 px
With fast internet.

Video plays at high quality
1920 x 1080 px with no buffering

1280 x 720 px

Slow Internet

Screen size: 1920 x 1080 px
With slower internet.

Video plays at medium quality
1280x720 px with no buffering
Normal connection: The Player downloads the best quality video

Poor connection: The Player changes to downloading a smaller, faster video file

Normal connection: The Player returns to the maximum quality video file
Encoding: “bitrate ladders”

<table>
<thead>
<tr>
<th>Bitrate (kbps)</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>235</td>
<td>320x240</td>
</tr>
<tr>
<td>375</td>
<td>384x288</td>
</tr>
<tr>
<td>560</td>
<td>512x384</td>
</tr>
<tr>
<td>750</td>
<td>512x384</td>
</tr>
<tr>
<td>1050</td>
<td>640x480</td>
</tr>
<tr>
<td>1750</td>
<td>720x480</td>
</tr>
<tr>
<td>2350</td>
<td>1280x720</td>
</tr>
<tr>
<td>3000</td>
<td>1280x720</td>
</tr>
<tr>
<td>4300</td>
<td>1920x1080</td>
</tr>
<tr>
<td>5800</td>
<td>1920x1080</td>
</tr>
</tbody>
</table>
## Encoding: “bitrate ladders”

<table>
<thead>
<tr>
<th>Bitrate (kbps)</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>235</td>
<td>320x240</td>
</tr>
<tr>
<td>375</td>
<td>384x288</td>
</tr>
<tr>
<td>560</td>
<td>512x384</td>
</tr>
<tr>
<td>750</td>
<td>512x384</td>
</tr>
<tr>
<td>1050</td>
<td>640x480</td>
</tr>
<tr>
<td>1750</td>
<td>720x480</td>
</tr>
<tr>
<td>2350</td>
<td>1280x720</td>
</tr>
<tr>
<td>3000</td>
<td>1280x720</td>
</tr>
<tr>
<td>4300</td>
<td>1920x1080</td>
</tr>
<tr>
<td>5800</td>
<td>1920x1080</td>
</tr>
</tbody>
</table>

[Netflix.com] [Bitmovin.com]
Encoding

Video Asset

Complexity analysis
Every asset is encoded with no fixed CRF to measure complexity

Adjusted Encoding Profile
A new configuration encoding file optimizes the encoding ladder with settings specific to the asset

Encoding
The asset is encoded with the adjusted bitrate ladder

ABREncoded Content
The encoded content is delivered to storage as per the normal encoding workflow

Storage CDN
Encoding

Video quality (PSNR in dB)
“Chunks” of video at each bitrate
Client gets metadata about chunks via “Manifest”

```xml
<?xml version="1.0" encoding="UTF-8"?>
<MPD xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xmlns="urn:mpeg:DASH:schema:MPD:2011"
     xsi:schemaLocation="urn:mpeg:DASH:schema:MPD:2011"
     profiles="urn:mpeg:dash:profile:isoff-main:2011"
     type="static"
     mediaPresentationDuration="PT0H9M56.46S"
     minBufferTime="PT15.0S">
  <BaseURL>http://witestlab.poly.edu/~ffund/video/2s_480p_only/</BaseURL>
  <Period start="PT0S">
    <AdaptationSet bitstreamSwitching="true">
      <Representation id="0" codecs="avc1" mimeType="video/mp4"
                      width="480" height="360" startWithSAP="1" bandwidth="101492">
        <SegmentBase>
          <Initialization sourceURL="bunny_2s_100kbit/bunny_100kbit.mp4"/>
        </SegmentBase>
        <SegmentList duration="2">
          <SegmentURL media="bunny_2s_100kbit/bunny_2s1.m4s"/>
          <SegmentURL media="bunny_2s_100kbit/bunny_2s2.m4s"/>
          <SegmentURL media="bunny_2s_100kbit/bunny_2s3.m4s"/>
          <SegmentURL media="bunny_2s_100kbit/bunny_2s4.m4s"/>
          <SegmentURL media="bunny_2s_100kbit/bunny_2s5.m4s"/>
          <SegmentURL media="bunny_2s_100kbit/bunny_2s6.m4s"/>
        </SegmentList>
      </Representation>
    </AdaptationSet>
  </Period>
</MPD>
```
A client can fetch chunks of different qualities
Open Connect: Starting from a Greenfield (a mostly Layer 0 talk)

Dave Temkin
06/01/2015

Storage Appliance
- Still 4U high
- ~550 watts
- 288 TB of storage
- 2x 10G ports
- 20Gbit/s delivery

Flash Appliance
- 1U
- ~175 watts
- 24 TB of flash
- 2x 40G ports
- 40Gbit/s delivery
Network
Capacity (Mbps)

Time

Network

Downloading

1s chunks at different bit-rates

Playing out

Capacity < current rate ⇒ decrease rate
Common solution approach

• Encode video in multiple bitrates
• Replicate using a content delivery network
• Video player picks bitrate adaptively
  • Estimate connection’s available bandwidth
  • Pick a bitrate ≤ available bandwidth
Estimating available capacity

[A Buffer-Based Approach to Rate Adaptation: Evidence from a Large Video Streaming Service, Huang et al., ACM SIGCOMM 2014]
“A random sample of 300,000 Netflix sessions shows that roughly 10% of sessions experience a median throughput less than half of the 95th percentile throughput.”

“20–30% of rebuffers are unnecessary”

[A Buffer-Based Approach to Rate Adaptation: Evidence from a Large Video Streaming Service, Huang et al., ACM SIGCOMM 2014]
Capacity estimation

Decide based on the buffer alone?
Buffer-based adaptation

Nearly full buffer $\Rightarrow$ large rate
Buffer-based adaptation

Nearly empty buffer $\Rightarrow$ small rate
Buffer-based adaptation

Low buffer: use $R_{\text{min}}$

High buffer: use $R_{\text{max}}$

Safe from Unnecessary rebuffering

[A Buffer-Based Approach to Rate Adaptation: Evidence from a Large Video Streaming Service, Huang et al., ACM SIGCOMM 2014]
Problem: startup phase?

Pick a rate based on immediate past throughput
Buffer-based adaptation

Normalized number of rebuffers per hour (%)

[A Buffer-Based Approach to Rate Adaptation: Evidence from a Large Video Streaming Service, Huang et al., ACM SIGCOMM 2014]
Buffer-based adaptation

[A Buffer-Based Approach to Rate Adaptation: Evidence from a Large Video Streaming Service, Huang et al., ACM SIGCOMM 2014]
Summary

- Encode video in multiple bitrates
- Replicate using a content delivery network
- Video player picks bitrate adaptively
- Problem of active research interest, many competing algorithms and objectives
Let’s see this in action!