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
Naive approach

[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
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* 1280 x 720 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>

[netflix.com]
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

ABR Encoded 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>
      </Representation>
    </AdaptationSet>
  </Period>
</MPD>
```
A client can fetch chunks of different qualities

[netflix.com]
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
Encoding

Replication

Adaptation
1s chunks at different bit-rates

Capacity (Mbps)

Time

Network

Downloading

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 $\leq$ available bandwidth
### A Buffer-Based Approach to Rate Adaptation: Evidence from a Large Video Streaming Service

Te-Yuan Huang, Ramesh Johari, Nick McKeown, Matthew Trunnell*, Mark Watson*
Stanford University, Netflix*
{huangty, rjohari, nickm}@stanford.edu, {mtrunnell, watsonm}@netflix.com

---

**Abstract**

During the evening peak hours (8pm–1am EDT), well over millions of video sessions start at Netflix. We provide an overview of how Netflix's Edgecast CDN handles the initial startup phase of video sessions. Although peak Video On Demand (VoD) traffic is a large fraction of overall CDN traffic, we show that capacity estimation is unnecessary in steady state; predicting future capacity is challenging.

Existing Adaptive Bit Rate (ABR) algorithms face a challenge in estimating available network capacity. We empirically observe that roughly 10% of video sessions experience a median throughput less than half of the 95th percentile throughput.

Video sessions experience highly variable throughput. Of course, this would risk extensive rebuffering of video. One approach is to pick a video rate by estimating future network capacity. However, estimating events and estimating the available network capacity is challenging. Figure 1 is a sample trace reported by a Netflix video player, showing how the video starts playing within seconds. Each video is encoded with constant throughput, yet this extreme would lead to low video quality. The design goal of an ABR algorithm is to obtain high performance on both metrics in order to give users a good viewing experience [7].

<table>
<thead>
<tr>
<th>Avg. throughput over chunk download (kbps)</th>
<th>Max</th>
<th>Avg.</th>
<th>Min</th>
<th>Avg. throughput over half of the 95th percentile throughput (kbps)</th>
<th>Max</th>
<th>Avg.</th>
<th>Min</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>6000</td>
<td>4000</td>
<td>500</td>
<td>1000</td>
<td>2500</td>
<td>1600</td>
<td>1000</td>
</tr>
</tbody>
</table>

“20–30% of rebuffers are unnecessary”

---

[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.”](http://dx.doi.org/10.1145/2619239.2626296)
Capacity estimation

Decide based on the buffer alone?
Buffer-based adaptation

Network

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

Network

Nearly empty buffer $\Rightarrow$ small rate
A Buffer-Based Approach to Rate Adaptation:
Evidence from a Large Video Streaming Service

Te-Yuan Huang, Ramesh Johari, Nick McKeown, Matthew Trunnell∗, Mark Watson∗
Stanford University, Netflix∗
{huangty,rjohari,nickm}@stanford.edu, {mtrunnell,watsonm}@netflix.com

Buffer occupancy

Next chunk’s rate

Low buffer: use R_{min}

High buffer: use R_{max}

R_{max}

R_{min}

{\text{Safe from Unnecessary rebuffering}}

{\text{Risky Area}}

{\text{Playout & Buffer & Occupancy}}

B_{max}
Problem: startup phase?

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

A Buffer-Based Approach to Rate Adaptation: Evidence from a Large Video Streaming Service

Te-Yuan Huang, Ramesh Johari, Nick McKeown, Matthew Trunnell*, Mark Watson*
Stanford University, Netflix*
{huangty, rjohari, nickm}@stanford.edu, {mtrunnell, watsonm}@netflix.com

During the evening peak hours (8pm–1am EDT), well over a third of the video requests are for standard-definition video, which is the most bandwidth-hungry. On the other hand, very few requests are for high-definition video, and the data shows a small portion is for TV-quality video (720p). The graph shows the normalized number of rebuffers per hour for peak hours and control, as well as the lower bound. The rebuffers are shown for BBA and the control group. The y-axis represents the number of rebuffers per hour, and the x-axis represents the hours in GMT. The graph demonstrates that BBA reduces the number of rebuffers compared to the control group.
Buffer-based adaptation

A Buffer-Based Approach to Rate Adaptation: Evidence from a Large Video Streaming Service

Te-Yuan Huang, Ramesh Johari, Nick McKeown, Matthew Trunnell\*, Mark Watson\*
Stanford University, Netflix\*
{huangty,rjohari,nick}@stanford.edu, {mtrunnell,watsonm}@netflix.com

ACM SIGCOMM 2014

Video rate difference (kbps)

Control algorithm

BBA

Hours in GMT

Peak Hours