W1. INTRODUCTION TO ABR PRODUCTION AND DELIVERY
STREAMING MEDIA EAST - 2019

Jan Ozer
www.streaminglearningcenter.com
jozer@mindspring.com/
276-235-8542
@janozer
Agenda

• Introduction
• Lesson 1: Streaming fundamentals
• Lesson 2: Intro to objective quality metrics
• Lesson 3: Bitrate control
• Lesson 4: I, B, and P frames
• Lesson 5: Encoding with H.264
• Lesson 6: Introduction to ABR streaming
• Lesson 7: Distributing to computers, mobile and OTT
• Lesson 8: Introduction to encoding ladders
• Lesson 9: Choosing a codec in 2019
Lesson 1: Streaming Fundamentals

- Compression and codecs
  - Video codecs
  - Audio codecs
  - Choosing a codec
- Container formats
- Distribution alternatives
  - Streaming
  - Adaptive Streaming
- Configuration basics
  - Video resolution
  - Frame rate
  - Data rate
  - About video quality metrics
Compression and Codecs

- Compression
  - Used to shrink the size of video/audio
- Common codecs
  - Video - H.264/AVC, H.265/HEVC, VP9
  - Audio - AAC, Opus, Dolby
- Codecs - all of the above
  - Any technology that **COmpresses** in the studio, then **DEC**ompresses in the field
Choosing a Codec

- Choose based upon target device or devices
  - H.264 is close to universal
  - HEVC and VP9 deliver same quality as H.264 at lower bitrates, but not universally supported
  - AV1 is the open-source up and coming codec
  - VVC (Versatile Video Coding) is the standards-based successor to HEVC
- Much more later
Codecs and Container Formats

• **Codecs**: Compression technologies
  - H.264, VP9, HEVC

• **Container formats**
  - Specs detailing how data/metadata are stored in a file
    - MP4, WEBM, .MPD, .TS, .ISMV, .F4F
  - Also called “wrappers”
    - As in, “encoded the file using the H.264 codec in a QuickTime wrapper”

• **Why important?**
  - File must be in proper container format to play on target platforms
Where is Container Format?

- It’s in the file header
  - Very small percentage of overall content
- Can quickly change the container format without affecting A/V content
  - Called transmuxing
  - Very useful when delivering adaptive bitrate video in different formats (like DASH, HLS)
Key Point on Container Formats

- Separate and distinct from choice of codec
  - Can store MPEG-2 compressed video in MP4 file
  - Can store H.264 video in MPEG-2 transport stream

- Whenever you configure encoder for streaming, be aware of selected codec and container format
Distribution Alternatives

• Single file
  • One file delivered to all viewers

• Adaptive bitrate streaming (ABR)
  • Single input file (live or VOD)
  • Encoded to multiple targets
  • Delivered adaptively based upon playback CPU and connection bandwidth
Adaptive Streaming

- Adaptive streaming
  - Single input file (live or VOD)
  - Encoded to multiple outputs

- Delivered adaptively based upon playback CPU and connection bandwidth
  - Technically complex, but optimizes experience across all platforms and connection types
Adaptive Bitrate Encoding Ladder

- Contains the multiple configurations that each file is encoded into (this ladder is from a later lesson)
- Parameters shown must be configured correctly to ensure compatibility and optimize quality
- You will learn much more about ABR streaming and encoding ladders in later lessons

<table>
<thead>
<tr>
<th>HEVC/H.265</th>
<th>H.264/AVC</th>
<th>Resolution</th>
<th>Frame rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>145</td>
<td>145</td>
<td>416 x 234</td>
<td>≤ 30 fps</td>
</tr>
<tr>
<td>350</td>
<td>365</td>
<td>480 x 270</td>
<td>≤ 30 fps</td>
</tr>
<tr>
<td>660</td>
<td>730</td>
<td>640 x 360</td>
<td>≤ 30 fps</td>
</tr>
<tr>
<td>990</td>
<td>1100</td>
<td>768 x 432</td>
<td>≤ 30 fps</td>
</tr>
<tr>
<td>1700</td>
<td>2000</td>
<td>960 x 540</td>
<td>same as source</td>
</tr>
<tr>
<td>2400</td>
<td>3000</td>
<td>1280 x 720</td>
<td>same as source</td>
</tr>
<tr>
<td>3200</td>
<td>4500</td>
<td>same as source</td>
<td>same as source</td>
</tr>
<tr>
<td>4500</td>
<td>6000</td>
<td>same as source</td>
<td>same as source</td>
</tr>
<tr>
<td>5800</td>
<td>7800</td>
<td>same as source</td>
<td>same as source</td>
</tr>
</tbody>
</table>
Configuration Basics – Video Resolution

• Width and height of video in a file
• Significant determinant of video quality
  • The more pixels, the harder a file is to compress
  • Fewer pixels, easier to compress
Configuration Basics – Video Resolution

- That’s why video files are often scaled down for streaming
- Particularly at the lower end of the encoding ladder
Configuration Basics – Frame Rate

- Frames per second in the file
- Set during recording (top)
- Usually maintained during streaming
  - Sometimes reduced for lowest rungs on encoding ladder
  - Saw three slides ago
Configuration Basics – Data Rate

• You set data rate for both video and audio for every file that you encode

• Video
  • Data rate is the most important factor in overall quality
  • The higher the data rate, the better the quality; but the harder to deliver

• Audio
  • For most audio files, values beyond 128 kbps are a waste
  • Music videos and other high value productions are the exception
Questions

Should be: 9:20
Lesson 2: Introduction to Objective Quality Metrics

• What they are
• Why we need them
• Meet VMAF
What Are Objective Quality Metrics

• Mathematical formulas that (attempt to) predict how human eyes would rate the videos
  • Faster and less expensive
  • Automatable

• Examples
  • Peak Signal to Noise Ratio (PSNR)
  • Structural Similarity Index (SSIM)
  • SSIMPlus
  • VMAF (Video Multimethod Assessment Fusion)
Why Do We Need Them?

- So many encoding decisions
  - Data rate
  - Keyframe interval
  - B-frame interval
  - Bitrate control technique (VBR vs. CBR)
  - Choice of codec
  - Profile
  - Preset
- All have tradeoffs (quality vs. encoding time)

- Objective quality metrics allow us to mathematically measure quality
- Uses
  - Drive many per-title encoding technologies (Netflix)
  - Useful for many critical encoding decisions
Took Me From Here

Time consuming and error prone
Subjective comparisons
To Here

Statistically meaningful comparisons
With Objective Quality Metrics You Get

• More data
  • Can run many more tests in much less time
• Better data
  • Mathematical models can detect smaller changes than your eye can easily discern
What is VMAF?

- Four Metrics are fused using a Support Vector Machine (SVM)-based regression to a single output score ranging from 0–100 per video frame
  - 100 being identical to the reference video
  - Frame values are averaged to compute a single score
  - So, a high score can mask many ugly frames (more later)
- Or, in short, Netflix’s metric
What is VMAF?

- VMAF is “trainable”
  - Compute VMAF
  - Measure human subjective ratings
  - Feed those results back into VMAF to make the algorithm “smarter”

- Uses
  - Train for different types of content (animation, sports)
  - Train for different viewing conditions
VMAF is a Good Predictor of Subjective Ratings

- Horizontal axis is DMOS rating (human scores)
- Vertical is metric (VMAF on left, PSNR on right)
- Red line is perfect score – metric exactly matches subjective evaluation

VMAF is more tightly clumped around red line, which means it’s more accurate
  - Machine learning means it can get more accurate over time
- PSRN is much more scattered, and as a fixed algorithm, will never improve
Computing VMAF

Source

Encode

4K output

Compare to:

4K Source

2K output


1080p output

4K Source

720p output

4K Source

480p output
What’s This Mean

- Lower resolution rungs necessarily lose detail, yet get compared to 4K
  - Appropriate – assuming viewed on 4K TV
- Ensures that scores will drop at lower resolutions
- Scores range from 0 – 100
  - 80-100 – excellent
  - 60 – 80 – good
  - 40 – 60 – fair
  - 20 – 40 – poor
  - Below 20 - bad

VMAF Verification – 93 is a Number

- Real Networks White Paper - VMAF Reproducibility: Validating a Perceptual Practical Video Quality Metric
  - 4K 2D videos
- The results indicate that if a video service operator were to encode video to achieve a **VMAF score of about 93** then they would be confident of optimally serving the vast majority of their audience with content that *is either indistinguishable from original or with noticeable but not annoying distortion.*
Working With VMAF

• Range – 0 – 100
• Top rung target – typically 93 – 95
  • Higher is a waste
• Scores map to subjective
  • 0-20 bad      - 20 – 40 poor
  • 40 – 60 fair  - 60 – 80 good
  • 80 – 100 excellent
• 6 VMAF points = Just noticeable difference

Impact of Data Rate on VMAF Quality - 1080p

- Difference from here to here noticeable (bandwidth well spent)
- Difference from here to here not noticeable (bandwidth wasted)
VMAF Models

• Original (Default) model
  • Assumes that viewers watch a 1080p display with the viewing distance of 3x the screen height (3H).

• Phone model
  • Assume viewers watch on a mobile phone

• 4K Model
  • Video displayed on a 4K TV and viewed from a distance of 1.5H

1080p display

Mobile Phone

4K display
Phone vs. Default Model

- 4 encodes, 1080p, 720p, 540p, 480p
- Phone and default VMAF models; 93 target
- With phone model, 480p is above the 93 target in both videos
  - Any reason to transmit 540p+ rungs to mobile phones?
- Only 1080p file is above 93 using default model
  - Need 1080p video in your encoding ladder to achieve 93 score on 1080p displays
  - Certainly: Should run both models on 1080p footage targeted at mobile phones and larger displays
In this Presentation

• Mostly VMAF (scores to 100)
  • Always default model
• Sometimes PSNR (up to about 45)
• With both, higher scores are better
Computing VMAF

- Moscow State University VQMT - $995
- Hybrik Cloud – at least $1,000/month
- VMAF Master – Free
- Elecard Video Quality Estimator - $850
Questions

Should be: 9:35
Lesson 3: Bitrate Control

- How VBR and CBR work
- Differences in overall frame quality
- How both techniques affect deliverability
How VBR and CBR Work

VBR

CBR

Low Motion

Moderate Motion

Low Motion

Moderate Motion

High Motion
• Faint (sorry) wavy blue line is data rate
• Relatively consistent throughout
VBR File Illustrated

- Faint (sorry) wavy blue line is data rate
- Varies with scene complexity

596 kbps
Average
### How Much Better Quality is VBR over CBR?

<table>
<thead>
<tr>
<th>VMAF</th>
<th>200% VBR</th>
<th>150% VBR</th>
<th>110% VBR</th>
<th>2-Pass CBR</th>
<th>1-Pass CBR</th>
<th>Total Delta</th>
<th>Delta 110%-200%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tears of Steel</td>
<td>97.6</td>
<td>97.1</td>
<td>97.2</td>
<td>97.1</td>
<td>97.0</td>
<td>0.53</td>
<td>-0.34</td>
</tr>
<tr>
<td>Sintel</td>
<td>97.6</td>
<td>97.5</td>
<td>97.7</td>
<td>97.6</td>
<td>97.2</td>
<td>0.56</td>
<td>0.11</td>
</tr>
<tr>
<td>Big Buck Bunny</td>
<td>96.8</td>
<td>96.5</td>
<td>95.9</td>
<td>95.6</td>
<td>95.9</td>
<td>1.18</td>
<td>-0.89</td>
</tr>
<tr>
<td>Talking Head</td>
<td>95.7</td>
<td>95.7</td>
<td>95.7</td>
<td>95.7</td>
<td>95.7</td>
<td>0.09</td>
<td>-0.03</td>
</tr>
<tr>
<td>Freedom</td>
<td>97.6</td>
<td>96.7</td>
<td>96.6</td>
<td>96.3</td>
<td>96.4</td>
<td>1.32</td>
<td>-1.05</td>
</tr>
<tr>
<td>Haunted</td>
<td>94.5</td>
<td>94.5</td>
<td>94.4</td>
<td>94.3</td>
<td>94.5</td>
<td>0.24</td>
<td>-0.10</td>
</tr>
<tr>
<td>Screencam</td>
<td>95.5</td>
<td>95.3</td>
<td>94.9</td>
<td>94.3</td>
<td>95.2</td>
<td>1.18</td>
<td>-0.66</td>
</tr>
<tr>
<td>Tutorial</td>
<td>97.3</td>
<td>97.2</td>
<td>97.1</td>
<td>97.1</td>
<td>97.1</td>
<td>0.19</td>
<td>-0.12</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td><strong>96.6</strong></td>
<td><strong>96.3</strong></td>
<td><strong>96.2</strong></td>
<td><strong>96.0</strong></td>
<td><strong>96.1</strong></td>
<td><strong>0.58</strong></td>
<td><strong>-0.38</strong></td>
</tr>
</tbody>
</table>

- Across the spectrum of content – not that much – average .58 VMAF at 1080p
With Some Files, There May Be Spikes Where CBR Gets Ugly

- Red is first file (CBR)
- Blue is second (VBR)
- Graph tracks rating over entire file
- Top graph is entire file
- Bottom graph is expanded view of dark region up top
- Circled area shows very significant quality delta
VBR vs. CBR - Zoolander

1080p@2500kbps CBR  1080p@2500kbps VBR
Bitrate Control Test Video

30 seconds talking head/30 seconds ballet
Deliverability

- Which file is easier to deliver over fixed bandwidth connections?
  - Overall bitrate very similar (CBR slightly higher)
  - But, data rate is much more predictable, and therefore easier to deliver
- So, limit variability by implementing *constrained* VBR
  - Limit peaks to % over target
Producing CBR

• Typical uses:
  • Live
  • Streaming to constrained lower bitrate connections like 3G
• Typically single-pass, but can be two-pass
  • Adobe Media Encoder – single pass only
  • Choose CBR, then choose target bitrate
Producing VBR

- Typical uses
  - Most VOD streaming
  - Most mezz file creation
- Typically two-pass, but can be single or multiple
  - Adobe Media Encoder – 1 and 2 pass (typically choose 2 pass)
  - Choose VBR, then choose:
    - Target
    - Maximum (1.1x – 2x, here 1.5x)
    - Sometimes minimum (typically .5x)
A Word About Video Buffer Verifier

- Highly technical configuration option
- Key – the larger the buffer, the larger the variance in data rate
  - If trying for low variance to improve deliverability, keep VBV short (usually one second)
## CBR/VBR Summaries

### Constant Bitrate

- **Pros:**
  - Easiest stream to deliver
- **Cons**
  - Lowest overall quality
  - Transient quality issues
- **Best application**
  - Live streaming (beyond scope)

### Variable Bitrate

- **Pros:**
  - Best overall quality
  - No transient quality problems
- **Cons**
  - Can cause deliverability issues
- **Best application**
  - VOD
Questions

Should be: 10:00
Lesson 4: Frame Type Overview

• I, B, and P-frames
  • What they are and how to use them
  • Definition of a Group of Pictures (GOP)
Frame Types

- **I-frame** – complete frame
  - Least efficient

- **P-frame** – predictive frame
  - Can look backwards for interframe redundancies

- **B-frame** - bi-directional predictive frame
  - Can look forwards and backwards for redundancies

- **Group of Pictures – GOP –**
  - From I-frame to frame immediately preceding next I-frame
What are B-Frames and P-Frames Searching For?

- Interframe redundancies
  - Macro blocks that don’t change from frame to frame
  - This fuels interframe compression
    - Why talking heads encode more efficiently than soccer matches

- I-frames only use intraframe compression
  - Essentially JPEG
  - Largest, least efficient frames
About I-Frames

- I-frame – complete frame
  - Least efficient frame
  - Want as few as possible
- All playback starts with I-frame
  - For files that will be interactively viewed want regular keyframes
I-Frame Interval and Quality

• Quality
  • Longer the interval, the higher the quality
  • But, playback starts on I-frame
  • 10 seconds is a good target for a single file (not adaptive bitrate)

<table>
<thead>
<tr>
<th></th>
<th>.5 Sec</th>
<th>1 Sec</th>
<th>2 Sec</th>
<th>3 Sec</th>
<th>5 Sec</th>
<th>10 Sec</th>
<th>Max Delta</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tears of Steel</td>
<td>38.22</td>
<td>39.05</td>
<td>39.49</td>
<td>39.64</td>
<td>39.74</td>
<td>39.87</td>
<td>4.32%</td>
</tr>
<tr>
<td>Sintel</td>
<td>37.09</td>
<td>38.06</td>
<td>38.57</td>
<td>38.75</td>
<td>38.97</td>
<td>39.08</td>
<td>5.37%</td>
</tr>
<tr>
<td>Big Buck Bunny</td>
<td>37.03</td>
<td>37.93</td>
<td>38.52</td>
<td>38.68</td>
<td>38.64</td>
<td>39.09</td>
<td>5.57%</td>
</tr>
<tr>
<td>Talking Head</td>
<td>43.63</td>
<td>44.10</td>
<td>44.40</td>
<td>44.51</td>
<td>44.61</td>
<td>44.68</td>
<td>2.42%</td>
</tr>
<tr>
<td>Freedom</td>
<td>40.33</td>
<td>40.67</td>
<td>40.88</td>
<td>40.96</td>
<td>40.99</td>
<td>41.03</td>
<td>1.72%</td>
</tr>
<tr>
<td>Haunted</td>
<td>41.89</td>
<td>42.20</td>
<td>42.35</td>
<td>42.39</td>
<td>42.45</td>
<td>42.49</td>
<td>1.44%</td>
</tr>
<tr>
<td>Average</td>
<td><strong>39.26</strong></td>
<td><strong>39.96</strong></td>
<td><strong>40.37</strong></td>
<td><strong>40.51</strong></td>
<td><strong>40.59</strong></td>
<td><strong>40.75</strong></td>
<td><strong>3.88%</strong></td>
</tr>
<tr>
<td>Screencam</td>
<td>35.35</td>
<td>38.13</td>
<td>37.68</td>
<td>38.86</td>
<td>40.78</td>
<td>41.26</td>
<td>16.71%</td>
</tr>
<tr>
<td>Tutorial</td>
<td>38.26</td>
<td>43.06</td>
<td>43.61</td>
<td>44.65</td>
<td>46.15</td>
<td>47.89</td>
<td>25.17%</td>
</tr>
</tbody>
</table>
Scene Detection

- Scene change detection
  - Inserts I-frame at scene change to improves overall quality
  - For single files, enable I-frames at scene changes
I-Frames and Adaptive Groups

- Need I-frame at start of each segment
  - I-frame interval must divide into segment size
  - 6 second segments, use 1, 2, 3, or 6
    - Apple spec calls for 2-second interval

- Need regular I-frames: Either
  - Disable scene change detection
  - Typically what I do
  - Force keyframes at specified interval
I-Frames – Every 2 seconds, No Scene Change

- Telestream Switch
  - Yellow – I-frame
  - Purple – R-frame
  - Blue – B-frame

- Mission accomplished
  - Regular key frames
  - No scene change detection
2 Second GOP and I-frames at Scene Changes

- Mission accomplished
  - I-frames every 2 seconds
  - I-frames at scene changes

- Quality delta?
  - PSNR – no scene change - 41.222 dB
  - PSNR – scene change - 41.256 dB
  - About .08% difference
I-Frames and Scene Detection

Single File

• Large GOP (I-frame ~ 10 seconds)
• Enable scene change detection
  • Use defaults for minimum I-frame duration and scene change

Multiple File Adaptive Bitrate

• Shorter GOP (2-seconds)
  • Must divide evenly into segment size
• Disable scene change detection
  • For simplicity
  • Can enable, but more complex and no real quality improvement
About B-Frames

- B-frame – looks forward and backwards for redundancies
  - Most efficient frame
  - Want as many B-frames as possible

- B-frame interval set by preset choice (later lesson)
  - Choice is number of B-frames between I and P frames
    - 3 above
Typical B-Frame Encoding Parameters

- Number is number of B frames between I and P-Frames; (IBBBPBBBPBBBPBBBPBBBP)
  - What’s the best value?
  - Many programs don’t let you choose this option

- Reference frames (both P and B-frames)
  - Number of frames searched for redundancies
  - Many programs don’t let you choose
  - What’s the best value?
B-Frames and Quality

For most files, 3-4 delivers the best overall quality.

Max delta in most files is modest (.94% average for real world files).

Not a big deal either way.

<table>
<thead>
<tr>
<th></th>
<th>0B</th>
<th>1B</th>
<th>2B</th>
<th>3B</th>
<th>4B</th>
<th>5B</th>
<th>10B</th>
<th>15B</th>
<th>Max Delta</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tears of Steel</td>
<td>38.95</td>
<td>39.41</td>
<td>39.56</td>
<td>39.65</td>
<td>39.62</td>
<td>39.61</td>
<td>39.60</td>
<td>39.63</td>
<td>1.75%</td>
</tr>
<tr>
<td>Sintel</td>
<td>38.34</td>
<td>38.71</td>
<td>38.74</td>
<td>38.76</td>
<td>38.76</td>
<td>38.75</td>
<td>38.75</td>
<td>38.75</td>
<td>1.07%</td>
</tr>
<tr>
<td>Big Buck Bunny</td>
<td>39.96</td>
<td>40.34</td>
<td>40.41</td>
<td>40.40</td>
<td>40.38</td>
<td>40.41</td>
<td>40.40</td>
<td>40.39</td>
<td>1.13%</td>
</tr>
<tr>
<td>Talking Head</td>
<td>44.21</td>
<td>44.44</td>
<td>44.50</td>
<td>44.52</td>
<td>44.51</td>
<td>44.51</td>
<td>44.50</td>
<td>44.50</td>
<td>0.68%</td>
</tr>
<tr>
<td>Freedom</td>
<td>40.76</td>
<td>40.93</td>
<td>40.93</td>
<td>40.96</td>
<td>40.93</td>
<td>40.93</td>
<td>40.91</td>
<td>40.91</td>
<td>0.49%</td>
</tr>
<tr>
<td>Haunted</td>
<td>42.19</td>
<td>42.33</td>
<td>42.39</td>
<td>42.41</td>
<td>42.36</td>
<td>42.38</td>
<td>42.36</td>
<td>42.36</td>
<td>0.50%</td>
</tr>
<tr>
<td>Average</td>
<td>40.74</td>
<td>41.03</td>
<td>41.09</td>
<td><strong>41.11</strong></td>
<td>41.09</td>
<td>41.10</td>
<td><strong>41.09</strong></td>
<td><strong>41.09</strong></td>
<td><strong>0.94%</strong></td>
</tr>
<tr>
<td>Screencam</td>
<td>44.46</td>
<td>44.20</td>
<td>43.85</td>
<td>43.73</td>
<td>43.60</td>
<td>43.57</td>
<td><strong>43.26</strong></td>
<td>43.35</td>
<td>2.69%</td>
</tr>
<tr>
<td>Tutorial</td>
<td>48.35</td>
<td>48.57</td>
<td>48.71</td>
<td>48.72</td>
<td><strong>48.74</strong></td>
<td>48.72</td>
<td>48.72</td>
<td>48.72</td>
<td>0.81%</td>
</tr>
</tbody>
</table>
B-Frame Recommendations

• Most encoders don’t provide access to B-frame settings
• Many control B-frames with a “preset” like medium, slow, fast, or placebo
  • If you don’t change manually, preset controls, which is fine
  • If setting is ridiculous, like 0-1, or 5-15, change to ~3-4
  • Otherwise, leave it alone
Choosing the Number of Reference Frames

• About reference frames
• Reference frames and quality
• Reference frames and encoding time
• Choosing the number of reference frames
About Reference Frames

- How many frames P and B frames search for redundancies
  - If 1, search 1 frame; if 16, search 16

- Obviously impacts:
  - Quality
  - Encoding time
Reference Frames and Quality

<table>
<thead>
<tr>
<th>Average Quality</th>
<th>1 Ref</th>
<th>5 Ref</th>
<th>10 Ref</th>
<th>16 Ref</th>
<th>Max Delta</th>
<th>10 - 16 Delta</th>
<th>16 - 5 Delta</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tears of Steel</td>
<td>39.34</td>
<td>38.99</td>
<td>39.47</td>
<td>39.49</td>
<td>1.28%</td>
<td>-0.04%</td>
<td>-1.26%</td>
</tr>
<tr>
<td>Sintel</td>
<td>38.45</td>
<td>38.54</td>
<td>38.58</td>
<td>38.59</td>
<td>0.35%</td>
<td>-0.02%</td>
<td>-0.12%</td>
</tr>
<tr>
<td>Big Buck Bunny</td>
<td>39.99</td>
<td>40.09</td>
<td>40.11</td>
<td>40.11</td>
<td>0.31%</td>
<td>0.00%</td>
<td>-0.05%</td>
</tr>
<tr>
<td>Talking Head</td>
<td>44.27</td>
<td>44.36</td>
<td>44.39</td>
<td>44.40</td>
<td>0.29%</td>
<td>-0.03%</td>
<td>-0.10%</td>
</tr>
<tr>
<td>Freedom</td>
<td>40.68</td>
<td>40.80</td>
<td>40.85</td>
<td>40.87</td>
<td>0.47%</td>
<td>-0.06%</td>
<td>-0.19%</td>
</tr>
<tr>
<td>Haunted</td>
<td>42.24</td>
<td>42.32</td>
<td>42.35</td>
<td>42.36</td>
<td>0.26%</td>
<td>-0.02%</td>
<td>-0.08%</td>
</tr>
<tr>
<td>Average - 720p</td>
<td>40.83</td>
<td>40.85</td>
<td>40.96</td>
<td>40.97</td>
<td>0.34%</td>
<td>-0.03%</td>
<td>-0.29%</td>
</tr>
<tr>
<td>Screencam</td>
<td>43.59</td>
<td>43.73</td>
<td>43.76</td>
<td>43.70</td>
<td>0.38%</td>
<td>0.14%</td>
<td>0.07%</td>
</tr>
<tr>
<td>Tutorial</td>
<td>48.58</td>
<td>48.65</td>
<td>48.68</td>
<td>48.68</td>
<td>0.22%</td>
<td>-0.01%</td>
<td>-0.07%</td>
</tr>
</tbody>
</table>

- For most files, 16 delivers the most quality
- Max delta is miniscule
### Reference Frames and Encoding Time

<table>
<thead>
<tr>
<th>Encoding Time</th>
<th>1 Ref</th>
<th>5 Ref</th>
<th>10 Ref</th>
<th>16 Ref</th>
<th>Max Delta</th>
<th>10 - 16 Delta</th>
<th>16 - 5 Delta</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tears of Steel</td>
<td>39</td>
<td>49</td>
<td>72</td>
<td>91</td>
<td>133%</td>
<td>-21%</td>
<td>-46%</td>
</tr>
<tr>
<td>Sintel</td>
<td>40</td>
<td>53</td>
<td>71</td>
<td>76</td>
<td>90%</td>
<td>-7%</td>
<td>-30%</td>
</tr>
<tr>
<td>Big Buck Bunny</td>
<td>41</td>
<td>53</td>
<td>68</td>
<td>85</td>
<td>107%</td>
<td>-20%</td>
<td>-38%</td>
</tr>
<tr>
<td>Talking Head</td>
<td>37</td>
<td>47</td>
<td>61</td>
<td>77</td>
<td>108%</td>
<td>-21%</td>
<td>-39%</td>
</tr>
<tr>
<td>Freedom</td>
<td>99</td>
<td>142</td>
<td>200</td>
<td>263</td>
<td>166%</td>
<td>-24%</td>
<td>-46%</td>
</tr>
<tr>
<td>Haunted</td>
<td>47</td>
<td>65</td>
<td>93</td>
<td>123</td>
<td>162%</td>
<td>-24%</td>
<td>-47%</td>
</tr>
<tr>
<td>Average - 720p</td>
<td>51</td>
<td>68</td>
<td>94</td>
<td>119</td>
<td>136%</td>
<td>-21%</td>
<td>-43%</td>
</tr>
</tbody>
</table>

- 16 is more than twice as long as 1, and just under twice as long as 5
  - Negligible quality difference
- Opportunity to increase throughput (or cut cloud encoding costs)
Reference Frame Recommendations

• Many encoders don’t provide access to reference frame settings
• Many control reference frames with a “preset” like medium, slow, fast, or placebo
  • If you do nothing reference frames will value specified by the preset
• If encoding time or cost isn’t a consideration, go with preset
• Cut to 5 or 1 to save time with minimal impact on quality
Questions

Should be: 10:10
Lesson 4 – Encoding with H.264

- About H.264
- Encoding with H.264
  - Profiles
  - Levels
  - Entropy coding
  - X264 presets
What is H.264?

• Part 10 of the MPEG-4 specification
• Adapted by ISO and ITU
  • Telephony/cellular
  • TV - consumer electronics
  • Computer electronics
MPEG-4 Audio

- **AAC-Low Complexity (AAC-LC)**
  - The most basic and most broadly compatible
  - In my tests, indistinguishable from HE AAC/HE AACv2
- **High Efficiency AAC (2003)**
  - Also called AAC+ and aacPlus
- **High Efficiency AACv2 (2006)**
  - Also called enhanced AAC+, aacPlus v2 and eAAC+
MPEG-4 Audio Summary

• Recommendations
  • aacPlus and aacPlus v2 are really low bitrate codecs
  • If 128 kbps stereo (or 64 kbps mono), stay with AAC LC
What's MPEG-4/H.264 Cost?

- For free Internet video (e.g. no subscription or pay per view), free in perpetuity
  - Still technically an obligation to sign a license, but there are no teeth and no motivation to enforce
- For subscription or PPV, there may be a royalty obligation
- Check www.mpeg-la.com

Where End User pays for AVC Video
- Subscription (not limited by title) – 100,000 or fewer subscribers/yr = no royalty; >100,000 to 250,000 subscribers/yr = $25,000; >250,000 to 500,000 subscribers/yr = $50,000; >500,000 to 1M subscribers/yr = $75,000; >1M subscribers/yr = $100,000
- Title-by-Title - 12 minutes or less = no royalty; >12 minutes in length = lower of (a) 2% or (b) $0.02 per title

Where remuneration is from other sources
- Free Television - (a) one-time $2,500 per transmission encoder or (b) annual fee starting at $2,500 for >100,000 HH rising to maximum $10,000 for >1,000,000 HH
- Internet Broadcast AVC Video (not title-by-title, not subscription) – no royalty for life of the AVC Patent Portfolio License
- Enterprise cap: $3.5M per year 2006-07, $4.25M per year 2008-09, $5M per year 2010, $6.5M per year 2011-2015; $8.125M in 2016 and $9.75M per year in 2017 through 2020
- Royalties begin January 1, 2006
H.264 Profiles

- What profiles are and why they exist
- Compatibility aspects
- Quality-related aspects
What Profiles are and Why They Exist

- Profiles enable different encoding techniques to balance decoding complexity
  - Baseline uses the fewest, so is easiest to decode
    - Early video-capable iPods only supported the Baseline codec
  - High uses the most, so is the hardest to decode
    - All computers, mobile devices, TVs, STBs manufactured in the last 6+ years can play the High profile

<table>
<thead>
<tr>
<th>Feature</th>
<th>Baseline</th>
<th>Main</th>
<th>High</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 and P Slices</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>B Slices</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Multiple Reference Frames</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>In-Loop Deblocking Filter</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>CAVLC Entropy Coding</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>CABAC Entropy Coding</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Interlaced Coding (PicAFF, MBAFF)</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>8x8 vs. 4x4 Transform Adaptivity</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Quantization Scaling Matrices</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Separate Cb and Cr QP control</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Separate Color Plane Coding</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Predictive Lossless Coding</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Feature</th>
<th>Baseline</th>
<th>Main</th>
<th>High</th>
</tr>
</thead>
</table>
Encoding

• Profiles/Levels
  • Most critical *compatibility-related* setting
    • Encode using wrong profile, file won’t play on target device
  • Profile is available on all encoding tools

• Don’t exceed profile of target device
  • Exclusively a concern with older mobile
  • Computers and OTT devices can play High profile (any level)
Profiles and Quality

- High is always the best; Baseline always the worst
  - Jump from Baseline > Main more significant than Main > High
- Difference is greater in hard to encode files
  - TOS – 3.66%
  - Talking Head – .9%

<table>
<thead>
<tr>
<th>VMAF-Average</th>
<th>Baseline</th>
<th>Main</th>
<th>High</th>
<th>Delta - Baseline/Main</th>
<th>Delta - Main/High</th>
<th>Total Delta</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tears of Steel</td>
<td>92.83</td>
<td>95.46</td>
<td>96.23</td>
<td>2.83%</td>
<td>0.80%</td>
<td>3.66%</td>
</tr>
<tr>
<td>Sintel</td>
<td>93.65</td>
<td>95.78</td>
<td>96.38</td>
<td>2.27%</td>
<td>0.63%</td>
<td>2.91%</td>
</tr>
<tr>
<td>Big Buck Bunny</td>
<td>92.14</td>
<td>94.72</td>
<td>95.52</td>
<td>2.80%</td>
<td>0.83%</td>
<td>3.67%</td>
</tr>
<tr>
<td>Talking Head</td>
<td>94.35</td>
<td>94.93</td>
<td>95.19</td>
<td>0.61%</td>
<td>0.28%</td>
<td>0.90%</td>
</tr>
<tr>
<td>Freedom</td>
<td>92.87</td>
<td>94.65</td>
<td>95.36</td>
<td>1.91%</td>
<td>0.74%</td>
<td>2.67%</td>
</tr>
<tr>
<td>Haunted</td>
<td>89.56</td>
<td>91.11</td>
<td>91.99</td>
<td>1.73%</td>
<td>0.95%</td>
<td>2.70%</td>
</tr>
<tr>
<td>Screencam</td>
<td>92.80</td>
<td>94.01</td>
<td>94.34</td>
<td>1.30%</td>
<td>0.35%</td>
<td>1.66%</td>
</tr>
<tr>
<td>Tutorial</td>
<td>95.85</td>
<td>96.13</td>
<td>96.15</td>
<td>0.29%</td>
<td>0.03%</td>
<td>0.32%</td>
</tr>
<tr>
<td>Average</td>
<td>93.01</td>
<td>94.60</td>
<td>95.14</td>
<td>1.72%</td>
<td>0.57%</td>
<td>2.31%</td>
</tr>
<tr>
<td>Width</td>
<td>Height</td>
<td>Frame Rate</td>
<td>Video Bitrate</td>
<td>Audio Bitrate</td>
<td>I-Frame</td>
<td>Profile</td>
</tr>
<tr>
<td>-------</td>
<td>--------</td>
<td>------------</td>
<td>----------------</td>
<td>---------------</td>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>416</td>
<td>234</td>
<td>12</td>
<td>200</td>
<td>64</td>
<td>36</td>
<td>Baseline</td>
</tr>
<tr>
<td>480</td>
<td>270</td>
<td>15</td>
<td>400</td>
<td>64</td>
<td>45</td>
<td>Baseline</td>
</tr>
<tr>
<td>640</td>
<td>360</td>
<td>29.97</td>
<td>600</td>
<td>64</td>
<td>90</td>
<td>Baseline</td>
</tr>
<tr>
<td>640</td>
<td>360</td>
<td>29.97</td>
<td>1200</td>
<td>96</td>
<td>90</td>
<td>Baseline</td>
</tr>
<tr>
<td>960</td>
<td>540</td>
<td>29.97</td>
<td>3500</td>
<td>96</td>
<td>90</td>
<td>Main</td>
</tr>
<tr>
<td>1280</td>
<td>720</td>
<td>29.97</td>
<td>5000</td>
<td>128</td>
<td>90</td>
<td>Main</td>
</tr>
<tr>
<td>1280</td>
<td>720</td>
<td>29.97</td>
<td>6500</td>
<td>128</td>
<td>90</td>
<td>Main</td>
</tr>
</tbody>
</table>

• Initial version of TN2224 customized profile for different targets
Current HLS Authoring Specs Abandon Legacy Devices

- Significant change:
  - Expect all to play High profile
  - Keyframe – 2 seconds
  - Segment size – 6 seconds
  - Still 200% constrained VBR
  - Class poll

Encoding for Android Devices

- Android support is bifurcated
  - In OS software – Baseline profile only
  - In hardware/device supplied software, up to High

- Google recommends using Baseline
  (bit.ly/androidvideospecs)
  - Ignored by many

---

Table 2. Examples of supported video encoding parameters for the H.264 Baseline Profile codec.

<table>
<thead>
<tr>
<th></th>
<th>SD (Low quality)</th>
<th>SD (High quality)</th>
<th>HD 720p (N/A on all devices)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video resolution</td>
<td>176 x 144 px</td>
<td>480 x 360 px</td>
<td>1280 x 720 px</td>
</tr>
<tr>
<td>Video frame rate</td>
<td>12 fps</td>
<td>30 fps</td>
<td>30 fps</td>
</tr>
<tr>
<td>Video bitrate</td>
<td>55 Kbps</td>
<td>500 Kbps</td>
<td>2 Mbps</td>
</tr>
<tr>
<td>Audio codec</td>
<td>AAC-LC</td>
<td>AAC-LC</td>
<td>AAC-LC</td>
</tr>
<tr>
<td>Audio channels</td>
<td>1 (mono)</td>
<td>2 (stereo)</td>
<td>2 (stereo)</td>
</tr>
<tr>
<td>Audio bitrate</td>
<td>24 Kbps</td>
<td>128 Kbps</td>
<td>192 Kbps</td>
</tr>
</tbody>
</table>
How Much Quality Difference in Encoding Ladder?

- **Talking Head Video**
  - 234p: 33.79, 34.20, 0.42%
  - 270p: 35.72, 35.99, 0.75%
  - 360p: 38.16, 38.37, 0.54%
  - 540p: 40.04, 40.33, 0.71%
  - 720p: 40.78, 41.32, 1.34%
  - 1080p: 43.53, 44.11, 1.34%
  - **Average**: 38.67, 39.05, 0.98%

- **High Motion Video**
  - 234p: 30.46, 31.56, 3.61%
  - 270p: 33.14, 33.73, 1.79%
  - 360p: 35.99, 36.38, 1.10%
  - 540p: 38.09, 38.62, 1.38%
  - 720p: 39.28, 39.84, 1.42%
  - 1080p: 41.31, 41.86, 1.32%
  - **Average**: 36.38, 37.00, 1.77%

- FFmpeg/x264/New TN2224/PSNR
- Very minor difference at all configurations
Encoding for Mobile - Choices

- Ignore older devices – all high profile
- Or, one set of files – mixed baseline, main, high, for all targets
  - Cheapest, easiest
  - May be leaving some quality on the table
- Or, separate ABR groups customized for devices:
  - Baseline – old iOS and Android
  - Main – old iOS and Android
  - High – new iOS, computers and OTT
  - Optimal quality, but more encoding, storage and administrative costs
Conclusions

• More and more, it seems as if publishers DON'T customize streams for different targets; either:
  • Go High profile and abandon legacy (really iPhone 4 and previous)
  • Use one set of streams with mixed profiles

• Justification
  • Quality isn’t that different
What Levels are and Why They Exist

Video formats supported: H.264 video up to 4K, 30 frames per second, High Profile level 4.2 with AAC-LC audio up to 160 Kbps, 48kHz, stereo audio or Dolby Audio up to 1008 Kbps,

- Levels set further limits on how video can be produced
  - See above specs for the iPhone 7
- In general, solely a concern for mobile
Levels from Wikipedia

<table>
<thead>
<tr>
<th>Level</th>
<th>Max decoding speed</th>
<th>Max frame size</th>
<th>Max video bit rate for video coding layer (VCL) kbit/s</th>
<th>Examples for high resolution @ highest frame rate (max stored frames)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Luma samples/s</td>
<td>Macroblocks/s</td>
<td>Luma samples</td>
<td>Macroblocks</td>
</tr>
<tr>
<td>4</td>
<td>62,914,560</td>
<td>245,760</td>
<td>2,097,152</td>
<td>8,192</td>
</tr>
<tr>
<td>4.1</td>
<td>62,914,560</td>
<td>245,760</td>
<td>2,097,152</td>
<td>8,192</td>
</tr>
<tr>
<td>4.2</td>
<td>133,693,440</td>
<td>522,240</td>
<td>2,228,224</td>
<td>8,704</td>
</tr>
</tbody>
</table>

- You have to make sure to keep your encodes within these constraints
Entropy Encoding

- Always use CABAC (highest quality option)
Choosing an X264 Preset

• What are presets?
  • X264-only
    • Simple way to adjust multiple parameters to balance quality and encoding time
    • Most other H.264 codecs have something similar
  • Medium is generally the default preset
    • Is this the best for you?
Test Presets

- Eight files
  - 1 movie (Tears of Steal)
  - 2 animations (Sintel, BBB)
  - Two general purpose (concert, advertisement)
  - One talking head
  - Screencam
  - Tutorial (PPT/Video)

- Encode to all presets
- Measure encoding time
- Measure VMAF
Results Please

- Red is lowest quality
- Green is highest quality
- Note top values – average 95.62 (not Placebo)
- Very slow averages best quality
  - But only 8% spread between best and worst
Quality and Encoding Time of x264 Presets

Lowest quality acceptable (if capacity an issue)

Highest Reasonable value

- Quality as % of Total
- Encoding Time %
Check Results Plot – Ultrafast (red) vs Medium

- Plot of VMAF values over duration of clip
  - Red is ultrafast
  - Green is Medium
- Multiple deep drops that would be noticeable
  Never use ultrafast (even in live)
Check Results Plot – Faster (red) vs Medium

- One problem area, but no major quality differences
- Faster should be acceptable starting point for VOD and live
  - Cut encoding time by over 66% with no quality hit
  - Said another way, triple capacity
Bottom Line

- Medium may not be the best preset if you’re reaching encoding capacity
Questions

Should be: 10:30
Lesson 6: Introduction to ABR Streaming

• Adaptive streaming
  • Single input file (live or VOD)
  • Encoded to multiple outputs

• Delivered adaptively based upon playback CPU and connection bandwidth
  • Technically complex, but optimizes experience across all platforms

Illustration courtesy of www.bitmovin.net
ABR Technology Overview

- Two types of systems
  - Server-based (Flash, RTMP)
    - Legacy; on the way out
  - HTTP (most new installations) has various flavors
    - HTTP Live Streaming (HLS)
    - Dynamic Adaptive Streaming over HTTP (DASH)
    - Smooth Streaming (MS game platforms)

encoding.com – Global Format Report
Perspective

- All HTTP Technologies work similarly
  - Encoding ladder comprised of multiple rungs

<table>
<thead>
<tr>
<th>16:9 aspect ratio</th>
<th>H.264/AVC</th>
<th>Frame rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>416 x 234</td>
<td>145</td>
<td>≤ 30 fps</td>
</tr>
<tr>
<td>640 x 360</td>
<td>365</td>
<td>≤ 30 fps</td>
</tr>
<tr>
<td>768 x 432</td>
<td>730</td>
<td>≤ 30 fps</td>
</tr>
<tr>
<td>768 x 432</td>
<td>1100</td>
<td>≤ 30 fps</td>
</tr>
<tr>
<td>960 x 540</td>
<td>2000</td>
<td>same as source</td>
</tr>
<tr>
<td>1280 x 720</td>
<td>3000</td>
<td>same as source</td>
</tr>
<tr>
<td>1280 x 720</td>
<td>4500</td>
<td>same as source</td>
</tr>
<tr>
<td>1920 x 1080</td>
<td>6000</td>
<td>same as source</td>
</tr>
<tr>
<td>1920 x 1080</td>
<td>7800</td>
<td>same as source</td>
</tr>
</tbody>
</table>

Apple HLS Authoring Specification
Encoding and Packaging

• Encoder creates:
  • Chunked video files
  • Index files (M3U8) with file descriptions (rez/data rate/profile) and chunk URLs

• Uploads to HTTP web server
FILES AND BIT RANGE REQUEST

• When HTTP-based ABR started, all content files were split into multiple discrete files
  • Created administrative nightmare
  • Hundreds of thousands of files for even short videos
  • Most producers still use files for HLS

• Now all can use “byte range requests” from a single file
  • Upload a single file per layer with data in the header that identifies the relevant segments
    • MPEG-2 ts for HLS
    • fMP4 for DASH, Smooth Streaming, HDS, HLS
  • Talk about segments, mean both approaches
Player Side

- Player side
  - Loads the master manifest file
  - Starts playing first file listed in the master manifest file
  - Monitors playback buffer and (sometimes) CPU use
  - Changes streams as necessary
  - Uses index files to find the right files
DASH

Main manifest file (.mpd)

stream (variant) manifest files (.mpd)

Content files (.mp4)
Captions and DRM

- Caption formats are specific to each ABR format and are listed in the manifest files
- DRM is handled as part of the final file packaging (more later)
HTTP Adaptive Summary (review)

• All technologies work similarly
  • Chunked or segmented video files
  • Manifest data files
  • HTTP server
  • Player driven operation
• The big differentiating issues are:
  • Where they play
  • Whether they are a standard or proprietary
  • How much they cost (DASH=CA$H)
From Plug-ins to HTML: A Retrospective

• HTML5’s key benefit
• Where we are today?
Working in the HTML5 Environment

- HTML5’s key benefit
  - Video playback without plug-ins

- How it works
  - Instead of obtaining decoders for H.264 and other codecs from plug-ins like Flash/Silverlight
  - Browsers supply players and decoders
    - Decoders can be in the browser (Chrome, Safari, IE)
    - Decoders can be in the OS (Firefox, Opera)
HTML5 – Where We Are Today

No DRM/Advertising

Mostly gone by 2016 or so

DRM/Advertising

Still in use for some applications on some sites
Pieces of the Puzzle

- Media Source Extensions - MSE
- Dynamic Adaptive Streaming over HTTP - DASH
- Encrypted Media Extensions - EME
- ISO-Base Media File Format - BMFF
Media Source Extensions (MSE)

- JavaScript interface to play back media data provided by the JavaScript layer
- A W3C HTML Working Group spec
- More flexible than video tag
  - Media chunks (adaptive) and (closer to) true streaming than progressive
  - Live
  - Better support for captions and DRM (via Encrypted Media Extensions)
What is Dynamic Adaptive Streaming over HTTP (DASH)

- Standardized file format
  - HLS, Smooth, HDS all proprietary
- Like all HTTP-based technologies, it has
  - Fragmented video chunks (or single file with segments)
  - Manifest files
- Now may be subject to a royalty (MPEG-LA)
What is DASH? CA$H!
IP History

• MPEG DASH finalized in 2011-2012
• In November 2016, MPEG LA announces license (http://bit.ly/DASH_license)
Analysis and Implications

- This is the first royalty on free internet video
- CNN distributes free video in H264 or HEVC using HLS
  - No royalty
- CNN distributes free video with DASH
  - Royalty on apps and ultimately perhaps browser-based playback
- No exclusions for churches, charities, governments or otherwise
- Really is remarkable in scope
Analysis

• Industry bigs seem to daring other DASH IP owners to sue
  • Certainly Microsoft and Google are using it
  • Probably many other
• Perception that IP is weak (and won’t survive challenge) or non-essential to DASH operation
  • So, they’re not signing up for a license
  • Ball is in MPEG LA pool’s court
    • MPEG LA can’t sue; it has to be an actual IP owner
• Meanwhile, there’s a real monetary risk for DASH adapters and a real disincentive towards doing so
DASH and MSE

• DASH is **one of** the file formats MSE expects
• Can write JavaScript code enabling MSE to play HLS and other ABR formats
  • Very common among off the shelf players
Encrypted Media Extensions (EME)

- JavaScript API
  - Enables HTML5-based digital rights management (DRM)
  - Extends MSE by providing APIs to control playback of protected content.
- License/key exchange is controlled by the browser
  - Not a plug-in
## The Problem Is – No Universal DRM

### HTML5 Browsers

<table>
<thead>
<tr>
<th>Browser</th>
<th>PlayReady</th>
<th>Widevine MODULAR</th>
<th>Widevine CLASSIC</th>
<th>FairPlay Streaming</th>
<th>Primetime (ACCESS)</th>
<th>Marlin</th>
<th>CMLA-OMA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chrome (35+)</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Firefox (47+) ^1</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Internet Explorer (11)</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Microsoft Edge</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Opera (31+)</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Safari</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
</tr>
</tbody>
</table>

https://drmtoday.com/platforms/

- MS browser and mobile – PlayReady
- Google browser, Android and devices – Widevine
- Apple browser/devices – FairPlay
- Firefox – Primetime/Widevine

- So, you need multiple DRMs to distribute to multiple platforms
It’s OK from a File Creation Standpoint

• Using MPEG DASH (a media format) plus CENC (Common Encryption Scheme),
• Single adaptive group of files can contain multiple DRM key technologies
But You’ll Need a Multi-DRM Service Provider

• Azure
• BuyDRM
• Cisco VideoGuard Everywhere
• DRM Today
• EZDRM
• ExpressPlay
• Verimatrix
• Vualto DRM

• One or more DRMs added during encoding/packaging
• More on this throughout the presentation
Questions

Should be: 11:15
Lesson 7: Choosing an ABR Format

- Computers
- Mobile
- OTT
- Smart TVs
Choosing an ABR Format for Computers

• Can be DASH or HLS
• Factors
  • Off-the-shelf player vendor (JW Player, Bitmovin, THEOPlayer, etc.)
  • Encoding/transcoding vendor
Choosing an ABR Format for iOS

- Native support (playback in the browser)
  - HTTP Live Streaming
- Playback via an app
  - Any, including DASH, Smooth, HDS or RTMP Dynamic Streaming
## iOS Media Support

<table>
<thead>
<tr>
<th></th>
<th>Native</th>
<th>App</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Codecs</strong></td>
<td>H.264 (High, Level 4.2), HEVC (Main10, Level 5 high), VP8, VP9</td>
<td>Any</td>
</tr>
<tr>
<td><strong>ABR formats</strong></td>
<td>HLS</td>
<td>Any</td>
</tr>
<tr>
<td><strong>DRM</strong></td>
<td>FairPlay</td>
<td>Any</td>
</tr>
<tr>
<td><strong>Captions</strong></td>
<td>CEA-608/708, Web VTT, IMSC1</td>
<td>Any</td>
</tr>
<tr>
<td><strong>HDR</strong></td>
<td>HDR10, Dolby Vision</td>
<td>?</td>
</tr>
</tbody>
</table>

HEVC Hardware Support - iOS

No Support, 22%
Hardware-Accelerated High Efficiency Coding (HEVC), 78%

3%
H.265
H.264 97%

H.264 VS H.265
The 2017 WWDC announcement support for H.265/HLS within the iOS and tvOS ecosystem paved the way for the codec to target iOS 11 and later as well as the 4K Apple TV but we have yet to see widespread adoption.

Android: Codec and ABR Format Support

### Codecs

<table>
<thead>
<tr>
<th>Version</th>
<th>Codename</th>
<th>API</th>
<th>Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.3.3 - 2.3.7</td>
<td>Gingerbread</td>
<td>10</td>
<td>0.2%</td>
</tr>
<tr>
<td>4.0.3 - 4.0.4</td>
<td>Ice Cream Sandwich</td>
<td>15</td>
<td>0.3%</td>
</tr>
<tr>
<td>4.1.x</td>
<td>Jelly Bean</td>
<td>16</td>
<td>1.1%</td>
</tr>
<tr>
<td>4.2.x</td>
<td></td>
<td>17</td>
<td>1.5%</td>
</tr>
<tr>
<td>4.3</td>
<td></td>
<td>18</td>
<td>0.4%</td>
</tr>
<tr>
<td>4.4</td>
<td>KitKat</td>
<td>19</td>
<td>7.6%</td>
</tr>
<tr>
<td>5.0</td>
<td>Lollipop</td>
<td>21</td>
<td>3.5%</td>
</tr>
<tr>
<td>5.1</td>
<td></td>
<td>22</td>
<td>14.4%</td>
</tr>
<tr>
<td>6.0</td>
<td>Marshmallow</td>
<td>23</td>
<td>21.3%</td>
</tr>
<tr>
<td>7.0</td>
<td>Nougat</td>
<td>24</td>
<td>18.1%</td>
</tr>
<tr>
<td>7.1</td>
<td></td>
<td>25</td>
<td>10.1%</td>
</tr>
<tr>
<td>8.0</td>
<td>Oreo</td>
<td>26</td>
<td>14.0%</td>
</tr>
<tr>
<td>8.1</td>
<td></td>
<td>27</td>
<td>7.5%</td>
</tr>
</tbody>
</table>

### ABR

- VP8 (2.3+)
- H.264 (3+)
- HLS (3+)
- VP9 (4.4+)
- HEVC (5+)
- DASH 4.4+ Via MSE in Chrome

### Multiple codecs and ABR technologies

- Serious cautions about HLS
- DASH now close to 97%

### HEVC

- Main Profile Level 3 – mobile
  - 960×540@30.0
  - Hardware support probably exceeds this
- Main Profile – Level 4.1 – Android TV
  - 2,048×1,080@60.0


### Android Media Support

<table>
<thead>
<tr>
<th></th>
<th>Native</th>
<th>App</th>
</tr>
</thead>
<tbody>
<tr>
<td>Codecs</td>
<td>H.264, VP8, VP9, HEVC</td>
<td>Any</td>
</tr>
<tr>
<td>ABR formats</td>
<td>DASH, HLS</td>
<td>Any</td>
</tr>
<tr>
<td>DRM</td>
<td>Widevine</td>
<td>Any</td>
</tr>
<tr>
<td>Captions</td>
<td>Embedded 608/607 SRT</td>
<td>Any</td>
</tr>
<tr>
<td>HDR</td>
<td>Dolby-Vision, HDR10, VP9-HLG, VP9-PQ</td>
<td>?</td>
</tr>
</tbody>
</table>

[https://developer.android.com/guide/topics/media/media-formats](https://developer.android.com/guide/topics/media/media-formats)
HEVC Hardware Support - Android

- iOS playback more extensive but little penetration
- Hard to imagine
Adaptive Streaming to OTT

- Format support – general
- Roku
- Apple TV
- Chromecast
- Amazon Fire TV
- PS3/PS4
- Xbox 360/Xbox One
Who Matters?

STREAMING MEDIA DEVICE US MARKET SHARE

Roku | 37%
Amazon Fire TV | 28%
Chromecast | 14%
Apple TV | 15%
Other | 6%

Source: 2018 PARKS ASSOCIATES

## OTT Platform-Format Support

<table>
<thead>
<tr>
<th>OTT Platforms</th>
<th>Smooth Streaming</th>
<th>HLS</th>
<th>DASH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roku (bit.ly/roku_vid)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Amazon Fire TV (<a href="https://amzn.to/2L8dCdp">https://amzn.to/2L8dCdp</a>)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes (?)</td>
</tr>
<tr>
<td>ChromeCast (<a href="http://bit.ly/GCast_Media">http://bit.ly/GCast_Media</a>)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Apple TV (bit.ly/AppleTV_recs)</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

**Notes:**
- Roku 4 and Roku4 TVs supports HEVC and VP9
- Fire TV Gen 2 supports HEVC
- Fire TV Supports VP9
- Most recent Apple TV specs do support CMAF
# OTT Platform Codec Support

<table>
<thead>
<tr>
<th>OTT Platforms</th>
<th>H264</th>
<th>HEVC</th>
<th>VP9</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roku (bit.ly/roku_vid)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>None</td>
</tr>
<tr>
<td>Amazon Fire TVInsignia HD (<a href="https://amzn.to/2L8dCdp">https://amzn.to/2L8dCdp</a>)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>VP8, H.263, MPEG-2/4</td>
</tr>
<tr>
<td>ChromeCast Ultra (<a href="http://bit.ly/GCast_Media">http://bit.ly/GCast_Media</a>)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>VP8, HDR10, DolbyVision</td>
</tr>
<tr>
<td>Apple TV (bit.ly/AppleTV_recs)</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>None</td>
</tr>
</tbody>
</table>
# OTT Platform DRM Support

<table>
<thead>
<tr>
<th>OTT Platforms</th>
<th>PlayReady</th>
<th>Widevine</th>
<th>FairPlay</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roku (bit.ly/roku_vid)</td>
<td>Smooth/DASH</td>
<td>DASH (Beta)</td>
<td>No</td>
<td>Adobe, Verimatrix, AES-128</td>
</tr>
<tr>
<td>Amazon Fire TV Insignia HD (<a href="https://amzn.to/2L8dCdp">https://amzn.to/2L8dCdp</a>)</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>HDCP 2.2</td>
</tr>
<tr>
<td>Apple TV (bit.ly/AppleTV_recs)</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>SAMPLE-AES</td>
</tr>
</tbody>
</table>
# OTT Platform HDR Support

<table>
<thead>
<tr>
<th>OTT Platforms</th>
<th>Dolby Vision</th>
<th>HDR 10/10+</th>
<th>HLG</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roku (bit.ly/roku_vid)</td>
<td>No?</td>
<td>Yes/No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Amazon Fire TV Stick 4K</td>
<td>Yes</td>
<td>Yes/Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>(<a href="https://amzn.to/2L8dCdp">https://amzn.to/2L8dCdp</a>)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ChromeCast (<a href="http://bit.ly/GCast_Media">http://bit.ly/GCast_Media</a>)</td>
<td>Yes</td>
<td>Yes/No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Apple TV (bit.ly/AppleTV_recs)</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>
Adaptive Streaming to Smart TVs

- Format support – general
- Samsung
- Vizio
- Sharp
- Panasonic
- LG
- Smart TV Alliance
- HbbTV
Who Matters – Smart TVs?

Share (%) of Smart TVs by OEM in U.S. Wi-Fi Households

- Panasonic: 3%
- Sharp: 8%
- alcatel: 4%
- Sony: 5%
- LG: 7%
- Vizio: 30%
- Samsung: 33%

Source: comScore Connected Home, U.S. April 2017

Who Matters – Smart TV OS Market Share?

SMART TV OS MARKET SHARE

| Android TV | 40% |
| Tizen (Samsung) | 23% |
| LG webOS | 13% |
| Roku | 6% |
| Other | 18% |

Source: 2018 IHS Market

## Android TV – Same as Android

<table>
<thead>
<tr>
<th></th>
<th>Native</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Codecs</strong></td>
<td>H.264, VP8, VP9, HEVC</td>
</tr>
<tr>
<td><strong>ABR formats</strong></td>
<td>DASH, HLS</td>
</tr>
<tr>
<td><strong>DRM</strong></td>
<td>Widevine</td>
</tr>
<tr>
<td><strong>Captions</strong></td>
<td>Embedded 608/607 SRT</td>
</tr>
<tr>
<td></td>
<td>Dolby-Vision, HDR10, VP9-HLG, VP9-PQ</td>
</tr>
</tbody>
</table>

[https://developer.android.com/guide/topics/media/media-formats](https://developer.android.com/guide/topics/media/media-formats)
## Samsung Format Support (Tizen)


<table>
<thead>
<tr>
<th></th>
<th>TV 2019</th>
<th>TV 2018</th>
</tr>
</thead>
<tbody>
<tr>
<td>codecs</td>
<td>H.264, HEVC, WMV, VP9</td>
<td>H.264, HEVC, WMV, VP9</td>
</tr>
<tr>
<td>ABR formats</td>
<td>DASH, HLS, Smooth</td>
<td>DASH, HLS, Smooth</td>
</tr>
<tr>
<td>DRM</td>
<td>Widevine, AES-128, Verimatrix WebClient</td>
<td>Widevine, AES-128, Verimatrix WebClient</td>
</tr>
<tr>
<td>Captions</td>
<td>SMI, SRT, SMPTE-TT, WebVTT, 608/708</td>
<td>SMI, SRT, SMPTE-TT, WebVTT, 608/708</td>
</tr>
<tr>
<td>HDR</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Vizio Format Support - ?

- Data not publicly available
Sharp Format Support -?

• Data not publicly available
Smart TV Alliance

- **Members**
  - Panasonic, LG, Toshiba
- **Spec – 5.0 (9/2015)**
- **Codecs**
  - H.264, HEVC
- **ABR formats (M=mandatory)**
  - MPEG DASH, Smooth Streaming, HLS
- **DRM**
  - PlayReady, Widevine
- **Captions**
  - W3C TTML

<table>
<thead>
<tr>
<th>Function</th>
<th>Detail</th>
<th>A/V content</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td>HTTP 1.1 with Range request</td>
<td>M</td>
</tr>
<tr>
<td></td>
<td>HTTPS streaming over SSL</td>
<td>M</td>
</tr>
<tr>
<td>Adaptive</td>
<td>HTTP Live Streaming</td>
<td>M</td>
</tr>
<tr>
<td></td>
<td>Microsoft Smooth Streaming</td>
<td>M</td>
</tr>
<tr>
<td></td>
<td>MPEG-DASH (ISOBMFF &amp; CENC) according to HbbTV version 1.2.1 profile [26]</td>
<td>M</td>
</tr>
</tbody>
</table>

http://www.smarttv-alliance.org/specification.html
HbbTV 2.01 – 4/16/2016

- Codecs
  - H.264, HEVC
- ABR formats
  - DASH
- DRM
  - CENC
- Captions
  - W3C TTML

HTTP adaptive streaming shall be supported using MPEG DASH as defined in annex E.
Questions

Should be: 11:30
Lesson 8: Introduction to Encoding Ladders

• What they are and do
• A brief history of encoding ladder
• Creating a simple ladder – HD/H.264
• Creating a simple ladder 4K/HEVC
What Encoding Ladders Are and What They Do

- What they are
  - Collection of files encoded at different resolutions and data rates
  - Ensures that all viewers on all devices and connection speeds have a stream to view
  - Allows ABR technologies to adapt to changing bandwidth conditions
    - When bandwidth drops, player retrieves lower quality stream
    - When bandwidth increases, player retrieves higher quality stream
A Brief History ofEncoding Ladders

- Apple and TN2224
  - First really well developed specification
  - Very specific as to configurations
  - Some aspects tied to App store approval
  - Ensured playback on a range of old and new Apple devices
  - Given great credence by producers; some followed exactly
  - Later superceded by HLS Authoring Specification

<table>
<thead>
<tr>
<th>Clients</th>
<th>Dimensions for 16:9 aspect ratio</th>
<th>Dimensions for 4:3 aspect ratio</th>
<th>Frame rate</th>
<th>Video bit rate (average)</th>
<th>Video bit rate (peak)</th>
<th>Audio bit rate</th>
<th>Total bit rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>CELL</td>
<td>416 x 234</td>
<td>400 x 300</td>
<td>12</td>
<td>145</td>
<td>200</td>
<td>64</td>
<td>264</td>
</tr>
<tr>
<td>CELL</td>
<td>480 x 270</td>
<td>480 x 360</td>
<td>15</td>
<td>365</td>
<td>400</td>
<td>64</td>
<td>464</td>
</tr>
<tr>
<td>ATV</td>
<td>640 x 360</td>
<td>640 x 480</td>
<td>29.97</td>
<td>730</td>
<td>800</td>
<td>64</td>
<td>864</td>
</tr>
<tr>
<td>WiFi</td>
<td>768 x 432</td>
<td>640 x 480</td>
<td>29.97</td>
<td>1100</td>
<td>1200</td>
<td>96</td>
<td>1296</td>
</tr>
<tr>
<td>WiFi</td>
<td>ATV</td>
<td>960 x 540</td>
<td>29.97</td>
<td>2000</td>
<td>2200</td>
<td>96</td>
<td>2296</td>
</tr>
<tr>
<td>WiFi</td>
<td>ATV</td>
<td>1280 x 720</td>
<td>29.97</td>
<td>3000</td>
<td>3300</td>
<td>96</td>
<td>3396</td>
</tr>
<tr>
<td>WiFi</td>
<td>ATV</td>
<td>1280 x 960 or source</td>
<td>29.97</td>
<td>4500</td>
<td>5000</td>
<td>128</td>
<td>5128</td>
</tr>
<tr>
<td>WiFi</td>
<td>ATV</td>
<td>1280 x 720 or source</td>
<td>29.97</td>
<td>6000</td>
<td>6500</td>
<td>128</td>
<td>6628</td>
</tr>
<tr>
<td>WiFi</td>
<td>ATV</td>
<td>1920 x 1080</td>
<td>29.97</td>
<td>7800</td>
<td>8600</td>
<td>128</td>
<td>8728</td>
</tr>
</tbody>
</table>

Ladder from Authoring Specification

• Superceded by Authoring spec
  • Codec specific ladders (this for H.264)
  • Many producers simply start with this ladder and adapt

Table 2-1 Video average bit rate (kb/s) table 1

<table>
<thead>
<tr>
<th>Aspect Ratio</th>
<th>H.264/AVC</th>
<th>Frame Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>416 x 234</td>
<td>145</td>
<td>≤ 30 fps</td>
</tr>
<tr>
<td>640 x 360</td>
<td>365</td>
<td>≤ 30 fps</td>
</tr>
<tr>
<td>768 x 432</td>
<td>730</td>
<td>≤ 30 fps</td>
</tr>
<tr>
<td>768 x 432</td>
<td>1100</td>
<td>≤ 30 fps</td>
</tr>
<tr>
<td>960 x 540</td>
<td>2000</td>
<td>same as source</td>
</tr>
<tr>
<td>1280 x 720</td>
<td>3000</td>
<td>same as source</td>
</tr>
<tr>
<td>1280 x 720</td>
<td>4500</td>
<td>same as source</td>
</tr>
<tr>
<td>1920 x 1080</td>
<td>6000</td>
<td>same as source</td>
</tr>
<tr>
<td>1920 x 1080</td>
<td>7800</td>
<td>same as source</td>
</tr>
</tbody>
</table>

Apple Authoring Specification
Adopting the Apple Spec: High End First

- Full screen viewing on all devices
- Highest quality streams that you can afford

<table>
<thead>
<tr>
<th>16:9 aspect ratio</th>
<th>H.264/AVC</th>
<th>Frame rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>416 x 234</td>
<td>145</td>
<td>≤ 30 fps</td>
</tr>
<tr>
<td>640 x 360</td>
<td>365</td>
<td>≤ 30 fps</td>
</tr>
<tr>
<td>768 x 432</td>
<td>730</td>
<td>≤ 30 fps</td>
</tr>
<tr>
<td>768 x 432</td>
<td>1100</td>
<td>≤ 30 fps</td>
</tr>
<tr>
<td>960 x 540</td>
<td>2000</td>
<td>same as source</td>
</tr>
<tr>
<td>1280 x 720</td>
<td>3000</td>
<td>same as source</td>
</tr>
<tr>
<td>1280 x 720</td>
<td>4500</td>
<td>same as source</td>
</tr>
<tr>
<td>1920 x 1080</td>
<td>6000</td>
<td>same as source</td>
</tr>
<tr>
<td>1920 x 1080</td>
<td>7800</td>
<td>same as source</td>
</tr>
</tbody>
</table>
Desktop (browser-based) Next

- At least one stream for each window size in web site (MTV)
- Try to use same configurations as mobile to match Window size

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Format</th>
<th>Frame Size</th>
<th>Total Bitrate</th>
<th>Audio Bitrate</th>
<th>bits/px @ 30 fps</th>
<th>bits/px @ 24 fps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mobile &amp; constrained (low)</td>
<td>baseline, mono, 10 fps</td>
<td>448x252</td>
<td>150</td>
<td>48</td>
<td>0.09</td>
<td>0.09</td>
</tr>
<tr>
<td>Mobile &amp; constrained (high)</td>
<td>baseline, mono</td>
<td>448x252</td>
<td>450</td>
<td>48</td>
<td>0.12</td>
<td>0.15</td>
</tr>
<tr>
<td>Sidebar placements</td>
<td>main profile, stereo</td>
<td>384x216</td>
<td>400</td>
<td>96</td>
<td>0.12</td>
<td>0.15</td>
</tr>
<tr>
<td>Small in-page</td>
<td>main profile, stereo</td>
<td>512x288</td>
<td>750</td>
<td>96</td>
<td>0.15</td>
<td>0.18</td>
</tr>
<tr>
<td>Medium in-page</td>
<td>main profile, stereo</td>
<td>640x360</td>
<td>1200</td>
<td>96</td>
<td>0.16</td>
<td>0.20</td>
</tr>
<tr>
<td>Large in-page</td>
<td>main profile, stereo</td>
<td>768x432</td>
<td>1700</td>
<td>96</td>
<td>0.16</td>
<td>0.20</td>
</tr>
<tr>
<td>Full size in-page</td>
<td>main profile, stereo</td>
<td>960x540</td>
<td>2200</td>
<td>96</td>
<td>0.14</td>
<td>0.17</td>
</tr>
<tr>
<td>HD 720p (full screen)</td>
<td>high profile, stereo</td>
<td>1280x720</td>
<td>3500</td>
<td>96</td>
<td>0.12</td>
<td>0.15</td>
</tr>
</tbody>
</table>
Configuring Your Streams: Mobile Last

• How low will you go?
  • Slowest connection, lowest quality
    • Many drop data rate to preserve frame quality
  • Many producers don’t deploy 145 kbps stream
  • Some deploy audio-only stream
  • Try to configure at same resolutions as low end computer targets

<table>
<thead>
<tr>
<th>Aspect Ratio</th>
<th>Bit Rate</th>
<th>Frame Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>16:9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>416 x 234</td>
<td>145</td>
<td>≤ 30 fps</td>
</tr>
<tr>
<td>640 x 360</td>
<td>365</td>
<td>≤ 30 fps</td>
</tr>
<tr>
<td>768 x 432</td>
<td>730</td>
<td>≤ 30 fps</td>
</tr>
<tr>
<td>768 x 432</td>
<td>1100</td>
<td>≤ 30 fps</td>
</tr>
<tr>
<td>960 x 540</td>
<td>2000</td>
<td>same as source</td>
</tr>
<tr>
<td>1280 x 720</td>
<td>3000</td>
<td>same as source</td>
</tr>
<tr>
<td>1280 x 720</td>
<td>4500</td>
<td>same as source</td>
</tr>
<tr>
<td>1920 x 1080</td>
<td>6000</td>
<td>same as source</td>
</tr>
<tr>
<td>1920 x 1080</td>
<td>7800</td>
<td>same as source</td>
</tr>
</tbody>
</table>
What Data Rates?

• Apple TN2224: Keep adjacent bit rates a factor of 1.5 to 2 apart
  • If too close together, you waste band-width because quality difference is minimal (150 kbps and 180 kbps streams)
  • If too far apart, could strand some clients to lower quality stream unnecessarily
Minding the Jump

- Google sheet
  - Compute percentage jump from rung to rung
  - Red is outside 100% - 200%
  - Orange is close

<table>
<thead>
<tr>
<th>Width</th>
<th>Height</th>
<th>Data Rate</th>
<th>% Jump</th>
<th>FPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>234p</td>
<td>416</td>
<td>234</td>
<td>145</td>
<td>15</td>
</tr>
<tr>
<td>270p</td>
<td>480</td>
<td>270</td>
<td>365</td>
<td>15</td>
</tr>
<tr>
<td>360p</td>
<td>640</td>
<td>360</td>
<td>730</td>
<td>30</td>
</tr>
<tr>
<td>432p</td>
<td>768</td>
<td>432</td>
<td>1100</td>
<td>1.51</td>
</tr>
<tr>
<td>540p</td>
<td>960</td>
<td>540</td>
<td>2000</td>
<td>1.82</td>
</tr>
<tr>
<td>720p</td>
<td>1280</td>
<td>720</td>
<td>3000</td>
<td>1.50</td>
</tr>
<tr>
<td>1080p_l</td>
<td>1920</td>
<td>1080</td>
<td>4500</td>
<td>1.50</td>
</tr>
<tr>
<td>1080p_m</td>
<td>1920</td>
<td>1080</td>
<td>6000</td>
<td>1.33</td>
</tr>
<tr>
<td>1080p_h</td>
<td>1920</td>
<td>1080</td>
<td>7800</td>
<td>1.30</td>
</tr>
<tr>
<td>1440p</td>
<td>2560</td>
<td>1440</td>
<td>8100</td>
<td>1.04</td>
</tr>
<tr>
<td>2160p_low</td>
<td>3840</td>
<td>2160</td>
<td>11600</td>
<td>1.43</td>
</tr>
<tr>
<td>2160p_high</td>
<td>3840</td>
<td>2160</td>
<td>16800</td>
<td>1.45</td>
</tr>
</tbody>
</table>
HEVC/VP9/AV1

• Apple has a separate ladder for HEVC
  • Increases resolution for lowest data rates

<table>
<thead>
<tr>
<th>16:9 aspect ratio</th>
<th>HEVC/H.265 30 fps</th>
<th>HDR (HEVC) 30 fps</th>
<th>Frame rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>640 x 360</td>
<td>145</td>
<td>160</td>
<td>≤ 30 fps</td>
</tr>
<tr>
<td>768 x 432</td>
<td>300</td>
<td>360</td>
<td>≤ 30 fps</td>
</tr>
<tr>
<td>960 x 540</td>
<td>600</td>
<td>730</td>
<td>≤ 30 fps</td>
</tr>
<tr>
<td>960 x 540</td>
<td>900</td>
<td>1090</td>
<td>≤ 30 fps</td>
</tr>
<tr>
<td>960 x 540</td>
<td>1600</td>
<td>1930</td>
<td>same as source</td>
</tr>
<tr>
<td>1280 x 720</td>
<td>2400</td>
<td>2900</td>
<td>same as source</td>
</tr>
<tr>
<td>1280 x 720</td>
<td>3400</td>
<td>3850</td>
<td>same as source</td>
</tr>
<tr>
<td>1920 x 1080</td>
<td>4500</td>
<td>5400</td>
<td>same as source</td>
</tr>
<tr>
<td>1920 x 1080</td>
<td>5800</td>
<td>7000</td>
<td>same as source</td>
</tr>
<tr>
<td>2560 x 1440</td>
<td>8100</td>
<td>9700</td>
<td>same as source</td>
</tr>
<tr>
<td>3840 x 2160</td>
<td>11600</td>
<td>13900</td>
<td>same as source</td>
</tr>
<tr>
<td>3840 x 2160</td>
<td>16800</td>
<td>20000</td>
<td>same as source</td>
</tr>
</tbody>
</table>

Apple Authoring Specification
What’s the Problem With a Single Encoding Ladder?

• The Apple specs were the Rosetta Stone for most early producers
• Then Netflix recognized that all videos encode differently
  • Scale on chart (quality/data rate)
  • These high quality at a low bitrate
  • These don’t achieve same quality even at a much higher bitrate
Netflix Invented Per-Title Encoding

- All videos encode differently
- Fixed bitrate latter (animated file)
  - Either data rate too high (wasted bandwidth), or
  - Data rate too low (quality not optimized)
- Per-title – analyzed file
  - Created ladder with unique:
    - Number of rungs
    - Resolutions
    - Data rates

<table>
<thead>
<tr>
<th>Resolutions</th>
<th>Default bitrate ladder</th>
<th>Per-title bitrate ladder</th>
</tr>
</thead>
<tbody>
<tr>
<td>320x240</td>
<td>235</td>
<td>150</td>
</tr>
<tr>
<td>384x288</td>
<td>375</td>
<td>200</td>
</tr>
<tr>
<td>512x384</td>
<td>560</td>
<td>290</td>
</tr>
<tr>
<td>512x384</td>
<td>750</td>
<td></td>
</tr>
<tr>
<td>640x480</td>
<td>1050</td>
<td></td>
</tr>
<tr>
<td>720x480</td>
<td>1750</td>
<td>440</td>
</tr>
<tr>
<td>720x480</td>
<td></td>
<td>590</td>
</tr>
<tr>
<td>1280x720</td>
<td>2350</td>
<td>830</td>
</tr>
<tr>
<td>1280x720</td>
<td>3000</td>
<td>1150</td>
</tr>
<tr>
<td>1920x1080</td>
<td>4300</td>
<td>1470</td>
</tr>
<tr>
<td>1920x1080</td>
<td>5800</td>
<td>2150</td>
</tr>
<tr>
<td>1920x1080</td>
<td></td>
<td>3840</td>
</tr>
</tbody>
</table>
### Pros and Cons of Per-Title

<table>
<thead>
<tr>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Reduced bandwidth and storage for easy to encode clips</td>
<td>• Cost</td>
</tr>
<tr>
<td>• Improved QoE</td>
<td>• Encoding time</td>
</tr>
<tr>
<td>• Instead of 720p stream, get 1080p stream</td>
<td>• Complexity</td>
</tr>
<tr>
<td>• Improved quality (for hard to encode clips)</td>
<td>• But</td>
</tr>
<tr>
<td></td>
<td>• Easier and cheaper than deploying a new codec (uses same player)</td>
</tr>
<tr>
<td></td>
<td>• Delivers many of the same benefits</td>
</tr>
</tbody>
</table>
Bottom Line

• Per-title is key technology for all producers distributing mission critical video
• Either
  • Higher QoE
  • Lower bandwidth/storage
  • or, both
• Session on per-title later in the week
Questions

Should be: 11:45
Lesson 9: Choosing a Codec 2019

• Choosing a codec
  • Heritage/cost
  • Playback
  • Quality
  • Encoding time
  • Playback performance
## Heritage/Cost

<table>
<thead>
<tr>
<th></th>
<th>H.264</th>
<th>HEVC</th>
<th>VP9</th>
<th>AV1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heritage</td>
<td>Standards-based</td>
<td>Standards-based</td>
<td>Google</td>
<td>Alliance for Open Media</td>
</tr>
<tr>
<td>Cost – free streaming</td>
<td>None</td>
<td>None</td>
<td>Royalties</td>
<td>Royalties</td>
</tr>
<tr>
<td>Cost – PPV/Subscription</td>
<td>Royalty</td>
<td>Uncertain</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>Cost - hardware</td>
<td>Up to $9.75 million cap</td>
<td>$60 million+ annual cap*</td>
<td>.24 Euro proposed</td>
<td>.32 Euro proposed</td>
</tr>
<tr>
<td>Cost – software player</td>
<td>Up to $9.75 million cap (total/year)</td>
<td>Same</td>
<td>None</td>
<td>None</td>
</tr>
</tbody>
</table>

*Includes only two of three known royalty groups*
Choosing a Codec – First it Must Play

- Codec – stands for enCOde/DEcode
  - Need the decode side to play the video
- Which platforms have decoders?

<table>
<thead>
<tr>
<th>Codec</th>
<th>Computer/Notebook</th>
<th>iOS</th>
<th>Android</th>
<th>Retail OTT (Roku, Apple TV)</th>
<th>Smart TV</th>
</tr>
</thead>
<tbody>
<tr>
<td>H.264</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>HEVC</td>
<td>MacOS/Windows 10 with h/w and Edge</td>
<td>Current to level 5</td>
<td>Version 5+ to 540p</td>
<td>Most</td>
<td>All 4K</td>
</tr>
<tr>
<td>VP9</td>
<td>Chrome, Firefox, Opera, Edge</td>
<td>No</td>
<td>Version 4</td>
<td>Most (not Apple TV)</td>
<td>Most Newer</td>
</tr>
<tr>
<td>AV1</td>
<td>Will have soon</td>
<td>2020</td>
<td>2020</td>
<td>2020</td>
<td>2020</td>
</tr>
</tbody>
</table>
VP9/AV1: What’s it Cost You?

- Royalty free, but no indemnifications from Google
- Sisvel patent pool for AV1/VP9 and threats from Velos
  - Consumer device only
  - No content
  - No cap
  - Software tbd

bit.ly/sisvel_av1pool
Codec Quality

- HEVC and VP9 are roughly the equivalent
  - Close enough so that it’s not a relevant decision factor
- AV1 is up to 30% more efficient than HEVC/VP9

Figure 13: Average bitrate ratio for a fixed quality—use case “Ripping Encoding,” all sequences, YUV-SSIM metric.

Encoding Speed

- HEVC is slower than VP9, but it’s system and settings dependent.
- Both are much slower than H.264.
- AV1 is glacial “2500 – 3000 times slower than competitors” ~ December 2017.

5. Encoding Speed

Figures below show difference in encoding speed among participating codecs. AVS2 encoder shows better encoding speed comparing to other encoders. AV1 encoder has extremely low speed ~ 2500-3000 times lower than competitors. X265 Placebo presets (2 and 3 passes) have 10-15 times lower speed than the competitors.

My Tests

• My Tests: August 2018

<table>
<thead>
<tr>
<th>Then</th>
<th>Encoding Time (seconds)</th>
<th>Times Real Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>AV1</td>
<td>226,080</td>
<td>45,216</td>
</tr>
<tr>
<td>x265</td>
<td>289</td>
<td>58</td>
</tr>
<tr>
<td>LibVPx</td>
<td>226</td>
<td>45</td>
</tr>
<tr>
<td>x264</td>
<td>18</td>
<td>4</td>
</tr>
</tbody>
</table>

• Highest quality settings
• AV1 about 753 times slower than x265

• My Tests: February 2019

<table>
<thead>
<tr>
<th>Encoding Time (Seconds)</th>
<th>Times Real Time</th>
<th>VMAF</th>
</tr>
</thead>
<tbody>
<tr>
<td>AV1 - cpu-used 5</td>
<td>736</td>
<td>147.20</td>
</tr>
<tr>
<td>x265 - slow</td>
<td>38</td>
<td>7.60</td>
</tr>
<tr>
<td>LibVPx - speed 2</td>
<td>35</td>
<td>7.00</td>
</tr>
<tr>
<td>x264 - slow</td>
<td>7</td>
<td>1.40</td>
</tr>
</tbody>
</table>

• Typical producer settings
• AV1 about 19 times slower than x265
• Still significant, but rumors of real time encoding at NAB, 2019

Decode CPU

- Software-only playback on 2006 era Dell workstation
  - Much less on more modern computer, especially with hardware acceleration

- Most battery-powered devices (where higher CPU load decreases battery life) have hardware HEVC/VP9/H.264 decode
  - So, all three have a very significant advantage over AV1 until devices with hardware decode arrive (2020)
Decode CPU – AV1 Appears Reasonable

- Playback on an HP ZBook notebook (Xeon processor)
- 1080p video from YouTube played back in Firefox
- AV1 decode appears reasonable
- Facebook reportedly already distributing streams to iOS and Android devices
  - Decoder in their app
AV1 Summary

• Quality is alluring, but
  • Encoding cost will be expensive for the foreseeable future
    • Still makes sense if your videos are watched by millions (Netflix, YouTube, Hulu, etc)
    • Not for dozens or even hundreds of thousands of views
  • Quality starting to come into question, particularly respecting MPEG-next, or VVC (Versatile Video Coding)
VVC in a Nutshell from BBC Report

- HM = HEVC
- AV1 = AV1
- JEM = VVC (don’t ask)
- Chart shows data rate needed for equivalent quality
  - Shorter is better
- VVC appears to have a significant advantage over AV1 and HEVC
  - But it’s two years from being final, about 1.5 years behind AV1, maybe more
- HEVC and AV1 appear about equal
- BBC is in the HEVC patent pool
2018 Numbers from encoding.com

• Files produced by their customers
  • Big media companies, but not Netflix, YouTube, Hulu, etc.
• H.264 still king (*increased* by 2%)
• HEVC up but still in trial phase
  • Mostly encoded for Smart TVs and OTT, not computers/mobile
• VP9 down from 11% in 2016

Changing Codecs is a Big Deal

- While bandwidth savings are alluring:
  - Still need to encode to H.264 for legacy targets, so encoding and storage costs are additive
  - New codecs reduce caching benefits in distribution infrastructure
- The most attractive option is adding HEVC to HLS, but that’s been slow to develop
  - 2019 could be the year
- Per-title encoding delivers many of the same benefits without need to change infrastructure
Questions

Should be: 12:00