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 DECompresses in the field

Choosing a Codec

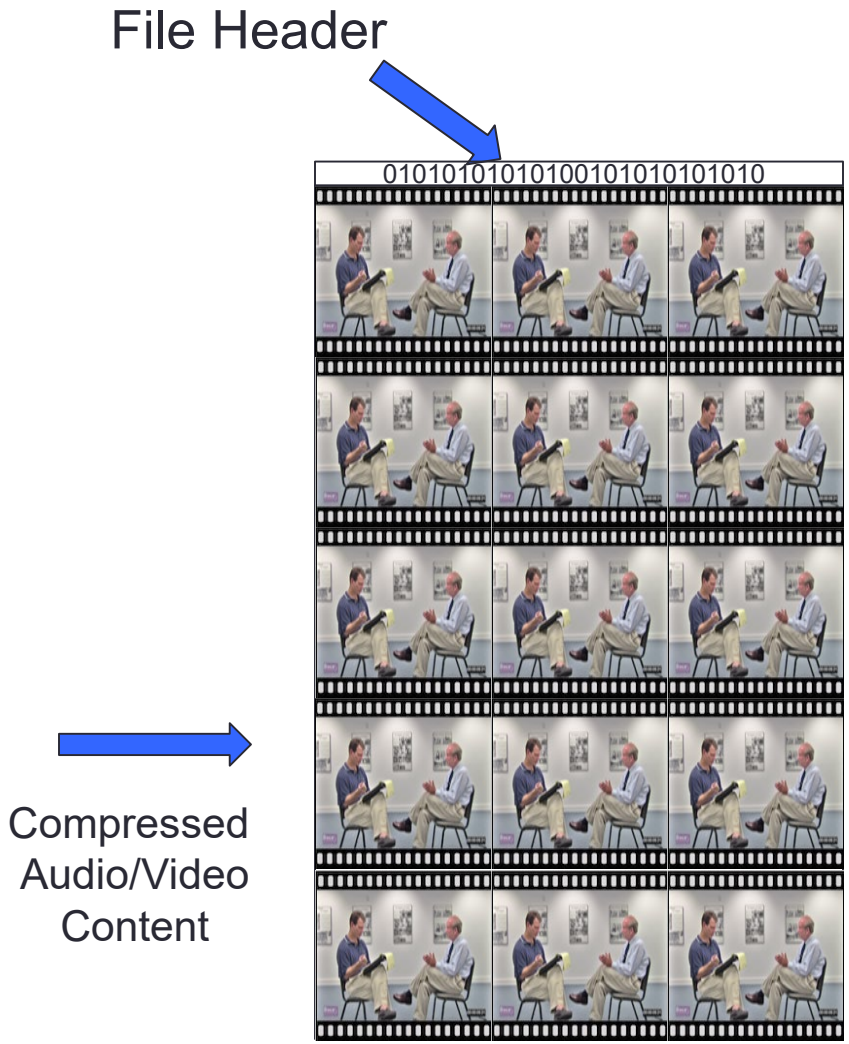
- Choose based upon target device or devices
  - H.264 is close to universal
  - HEVC and VP9 deliver same quality as H.264 at lower bitrates, but not universally supported
  - AV1 is the open-source up and coming codec
  - VVC (Versatile Video Coding) is the standards-based successor to HEVC
  - Much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

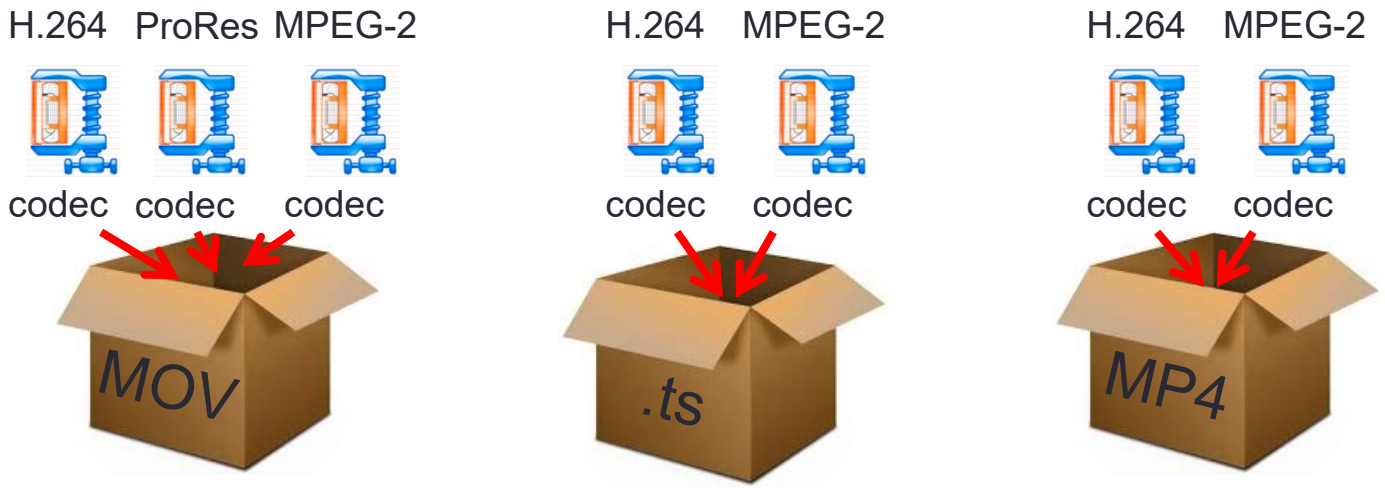
Where is Container Format?

- 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



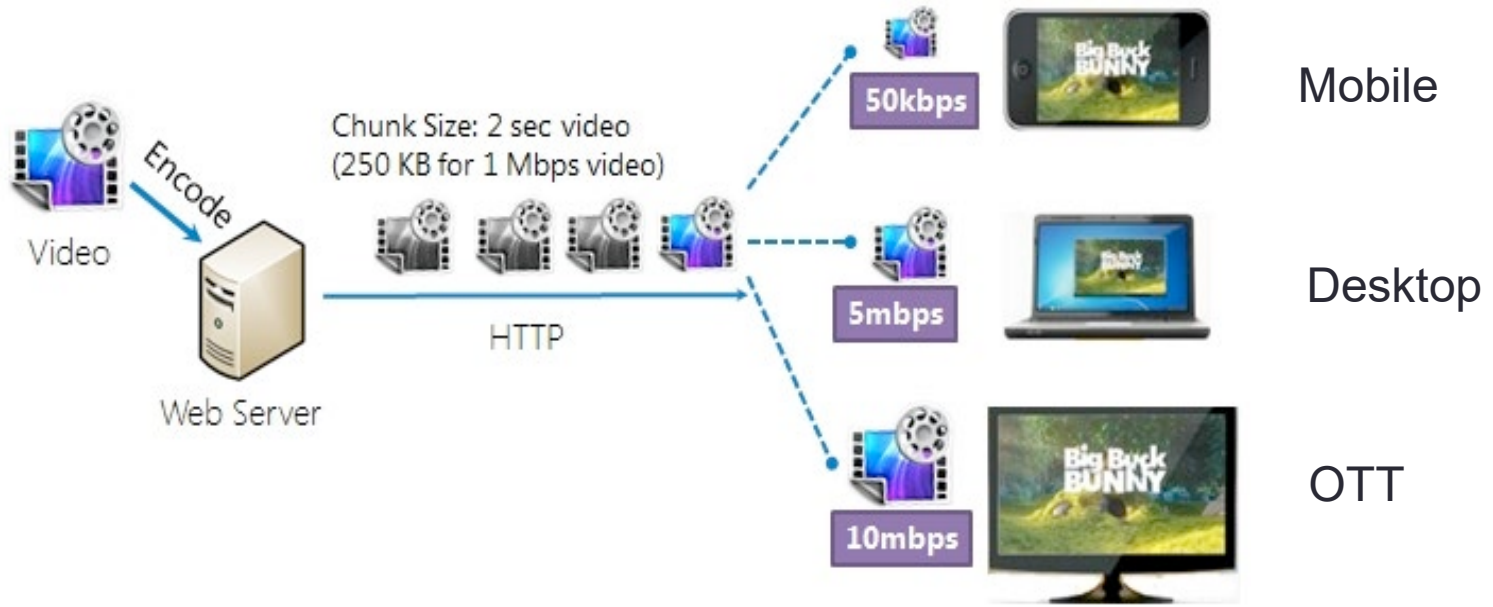
- Whenever you configure encoder for streaming, be aware of selected codec ***and*** container format

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



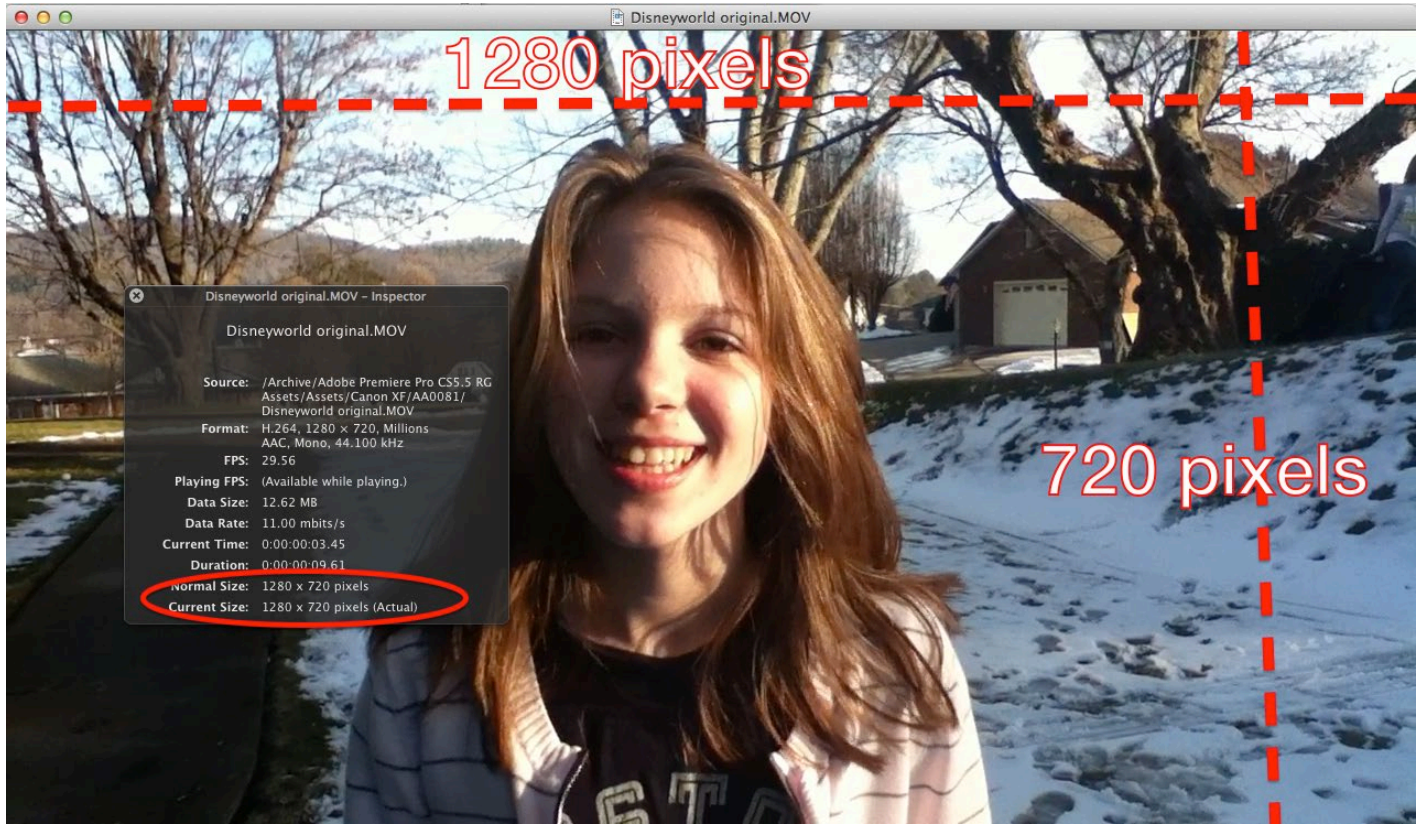
Adaptive Bitrate Encoding Ladder

- 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

HEVC/H.265	H.264/AVC	Resolution	Frame rate
145	145	416 x 234	≤ 30 fps
350	365	480 x 270	≤ 30 fps
660	730	640 x 360	≤ 30 fps
990	1100	768 x 432	≤ 30 fps
1700	2000	960 x 540	same as source
2400	3000	1280 x 720	same as source
3200	4500	same as source	same as source
4500	6000	same as source	same as source
5800	7800	same as source	same as source

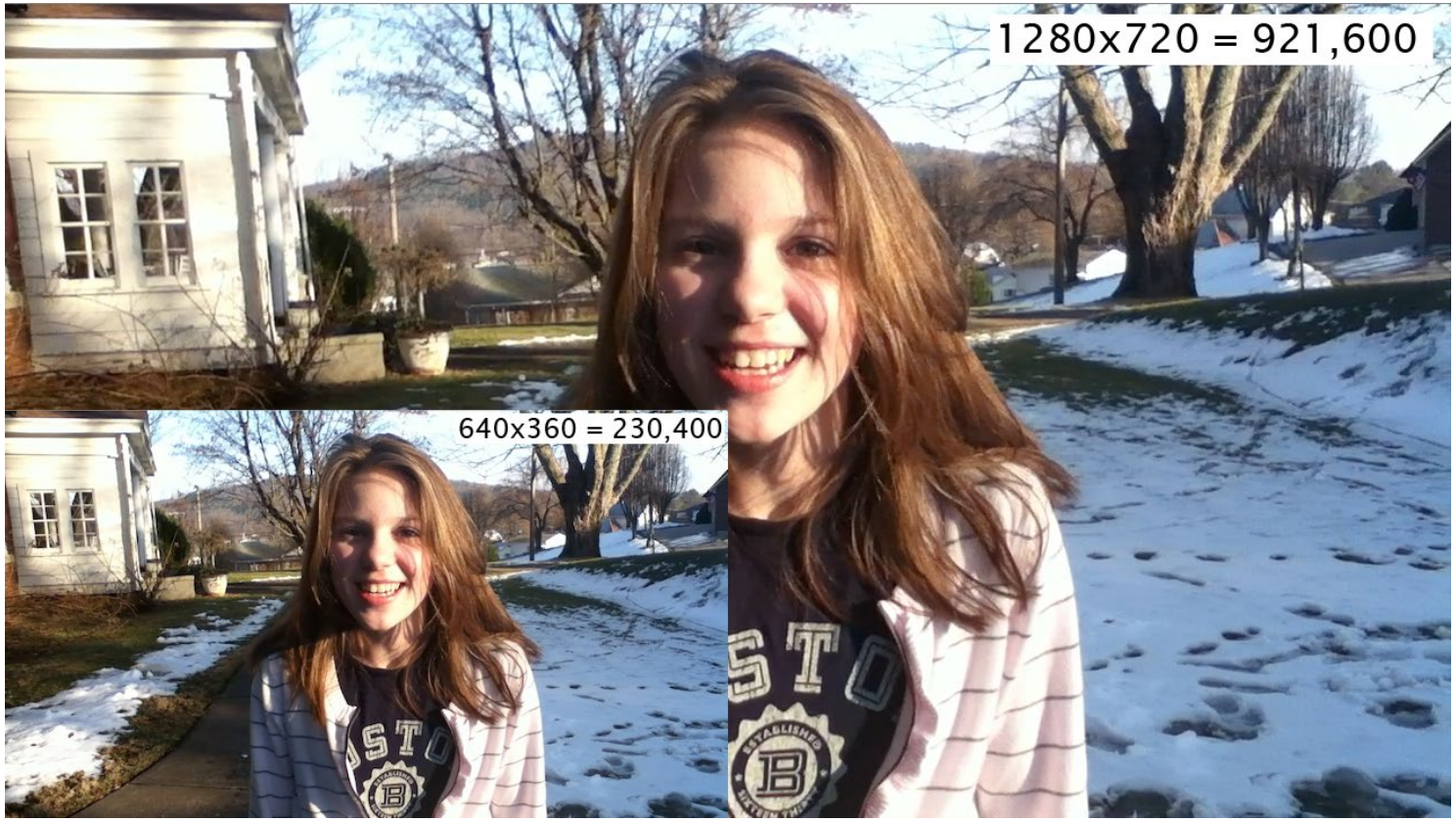
Configuration Basics – Video Resolution

- Width and height of video in a file
- Significant determinant of video quality
  - The more pixels, the harder a files is to compress
  - Fewer pixels, easier to compress



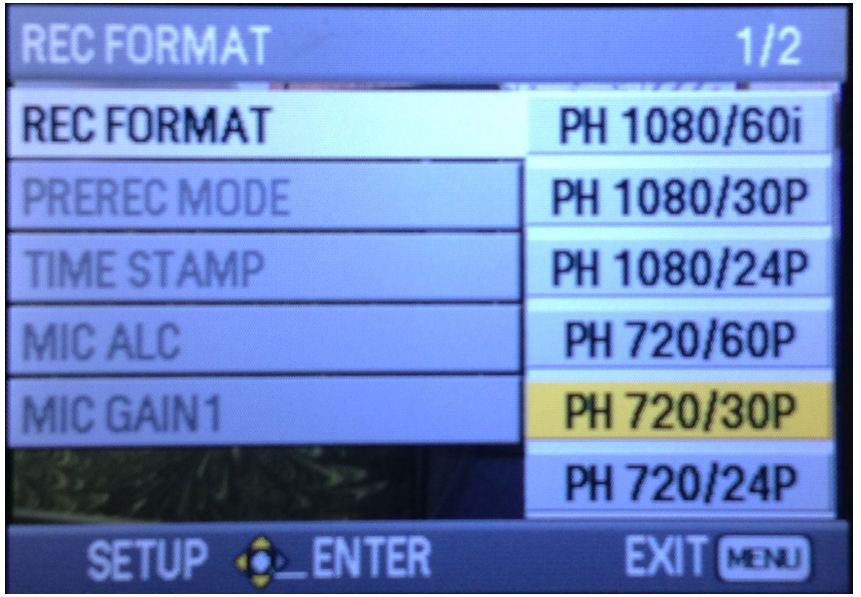
Configuration Basics – Video Resolution

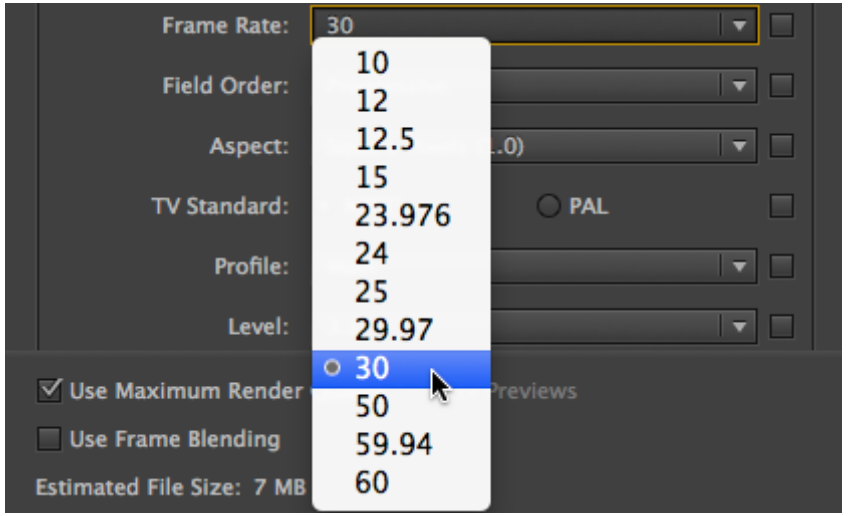
- That's why video files are often scaled down for streaming
- Particularly at the lower end of the encoding ladder



Configuration Basics – Frame Rate

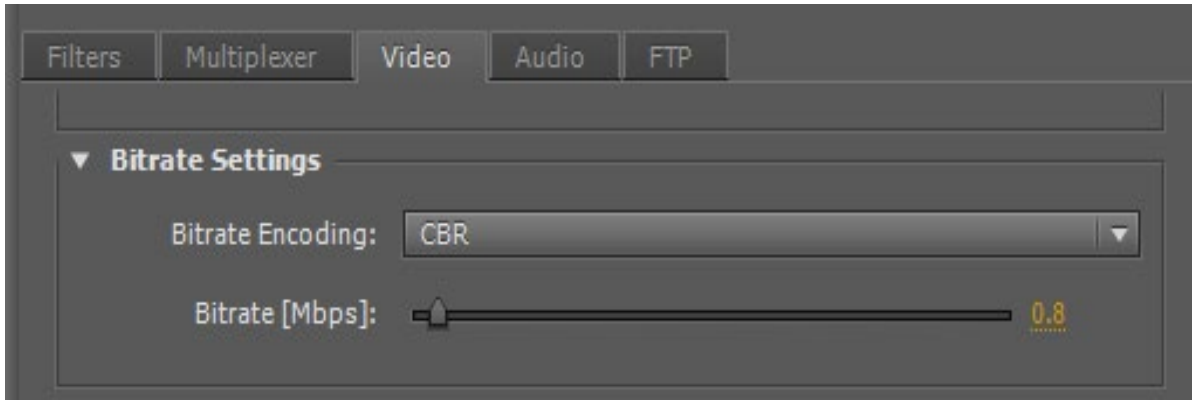
- Frames per second in the file
- Set during recording (top)
- Usually maintained during streaming
  - Sometimes reduced for lowest rungs on encoding ladder
  - Saw three slides ago

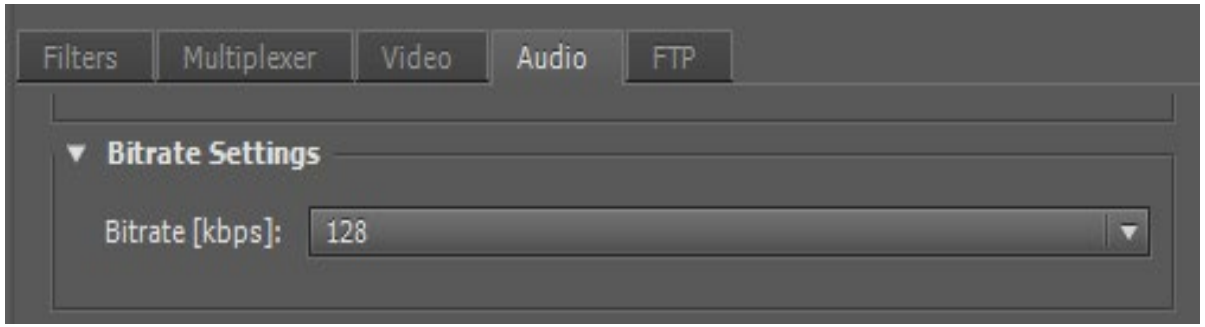




Configuration Basics – Data Rate

- 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

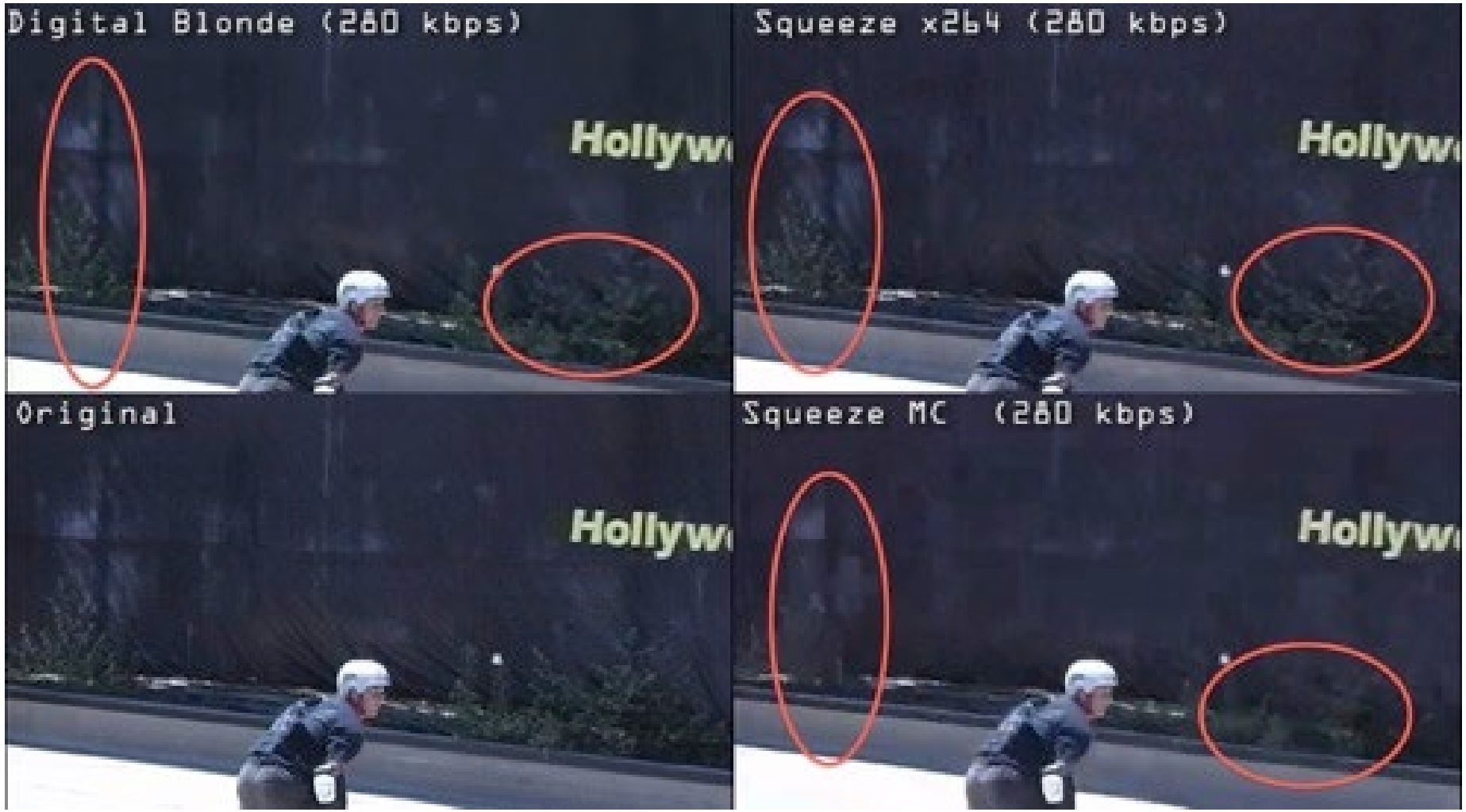
What Are Objective Quality Metrics

- 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

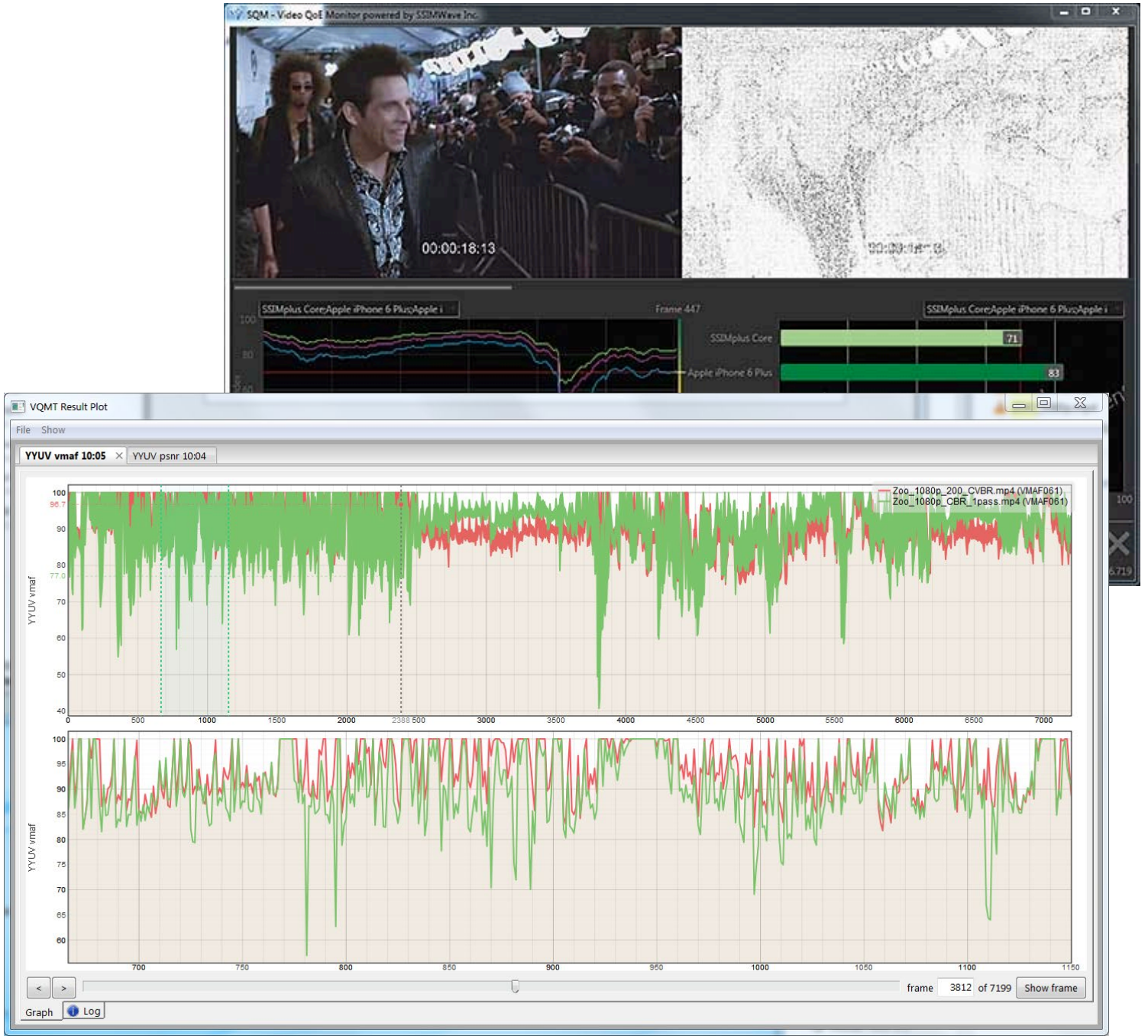


Time consuming and error prone
Subjective comparisons

To Here

VQM (lower is better)					
	Codec A	Codec B	Codec C	High > Low	Codec A > Codec B
Office 1	0.36	0.36	0.37	-3.54%	0.61%
Office 2	0.69	0.61	0.70	-13.51%	12.32%
Office 3	0.28	0.28	0.32	-14.74%	1.32%
Office 4	0.87	0.79	0.87	-9.63%	9.63%
Parking 1	0.68	0.61	0.74	-21.23%	10.90%
Parking 2	0.57	0.55	0.64	-15.47%	3.04%
Parking 3	1.86	1.58	1.76	-17.88%	17.88%
Parking 4	0.47	0.49	0.51	-8.86%	-3.81%
Retail 1	0.56	0.54	0.56	-4.27%	4.27%
Retail 2	0.68	0.66	0.69	-4.45%	3.39%
Retail 3	0.78	0.72	0.76	-8.64%	8.64%
Retail 4	0.73	0.67	0.88	-32.16%	8.52%
Traffic 1	0.55	0.50	0.58	-15.89%	9.14%
Traffic 2	0.34	0.32	0.38	-17.79%	6.39%
Traffic 3	0.52	0.49	0.55	-11.42%	5.29%
Traffic 4	0.68	0.61	0.66	-11.56%	11.56%
Total	**10.61**	**9.78**	**10.96**		
7.84%	Difference between Codec A and Codec B				
-3.34%	Difference between Codec A and Codec C				
-12.13%	Difference between Codec B and Codec C				
	0.61				
	Green equals best in category				
	Orange means worst in category				
	Difference greater than 7.5%				

Statistically meaningful comparisons

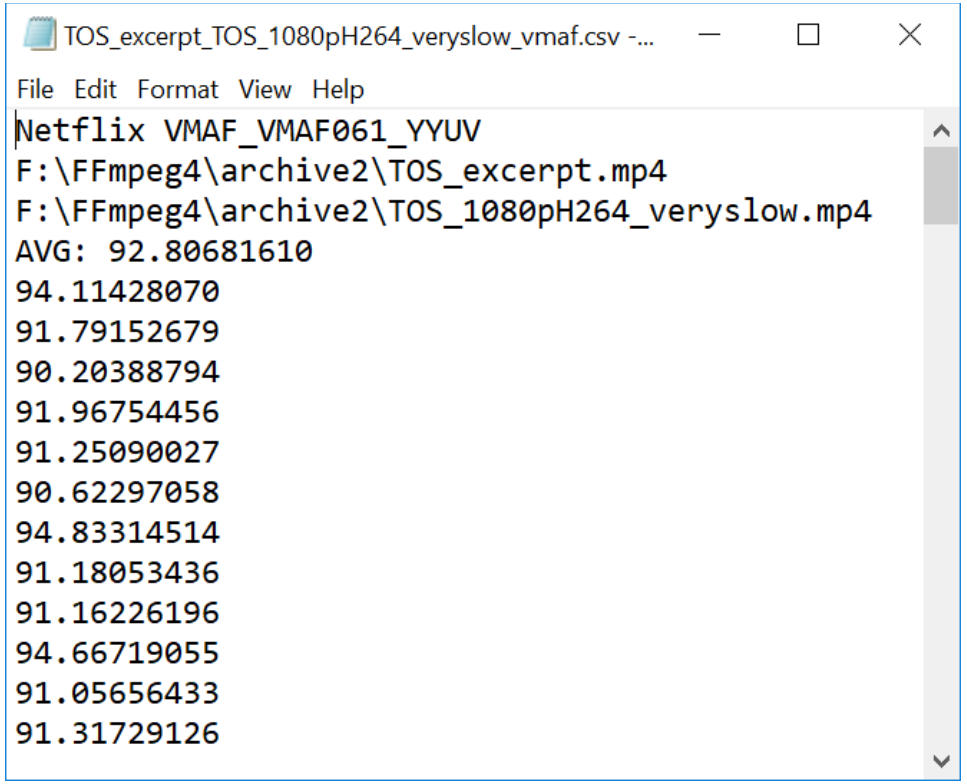


With Objective Quality Metrics You Get

- 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?

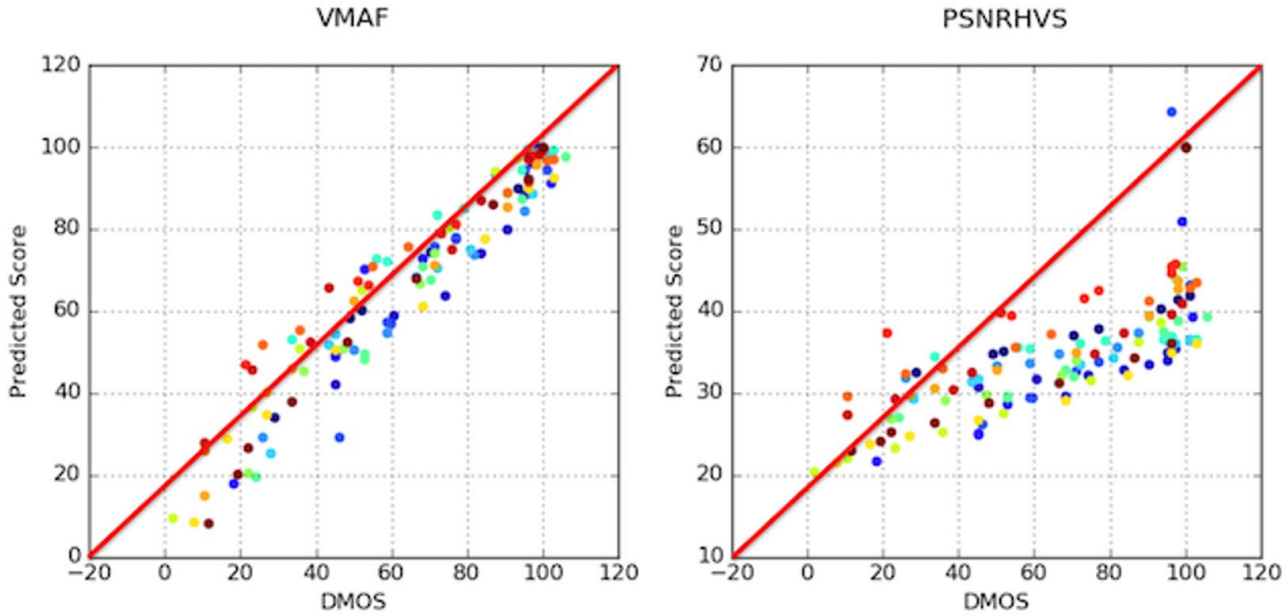
- Four Metrics are fused using a Support Vector Machine (SVM)-based regression to a single output score ranging from 0–100 per video frame
  - 100 being identical to the reference video
  - Frame values are averaged to compute a single score
  - So, a high score can mask many ugly frames (more later)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



- Horizonal 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

4K Source

Encode

4K output

2K output

1080p output

720p output

480p output

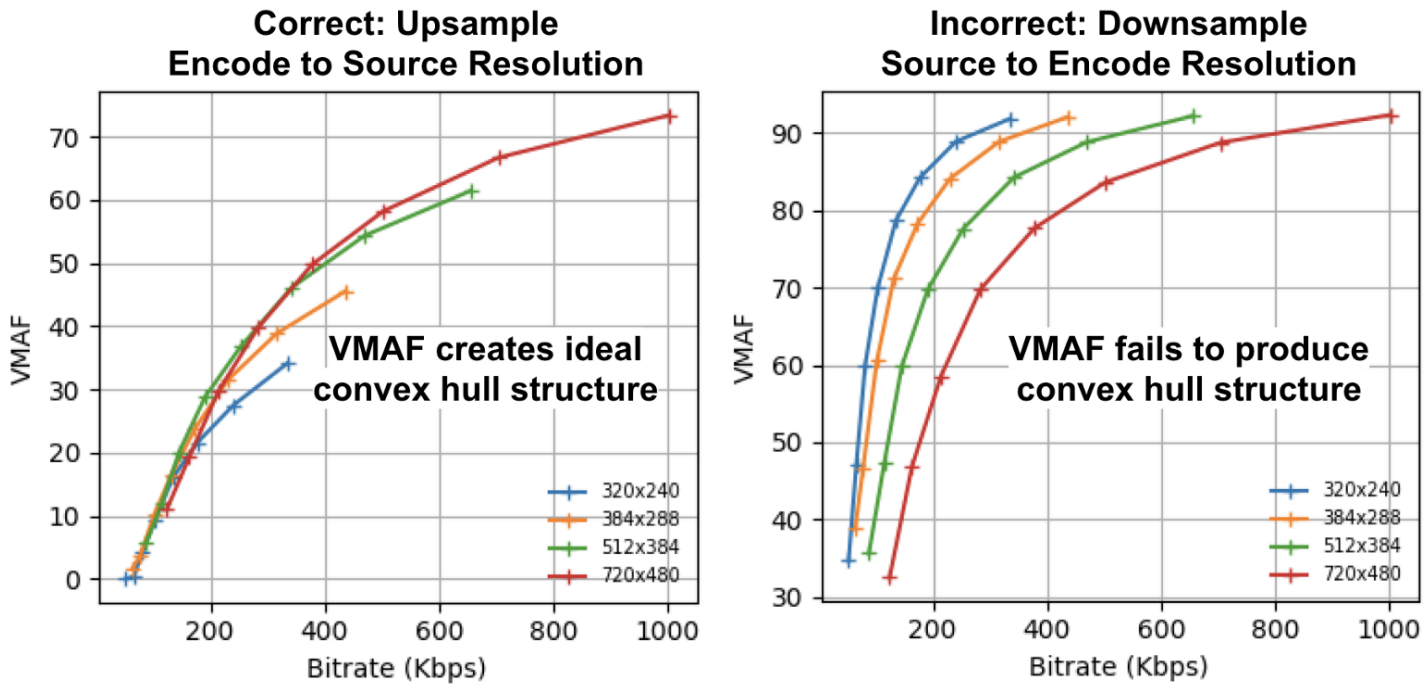
Compare to:

4K Source

http://bit.ly/VMAF_journey

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



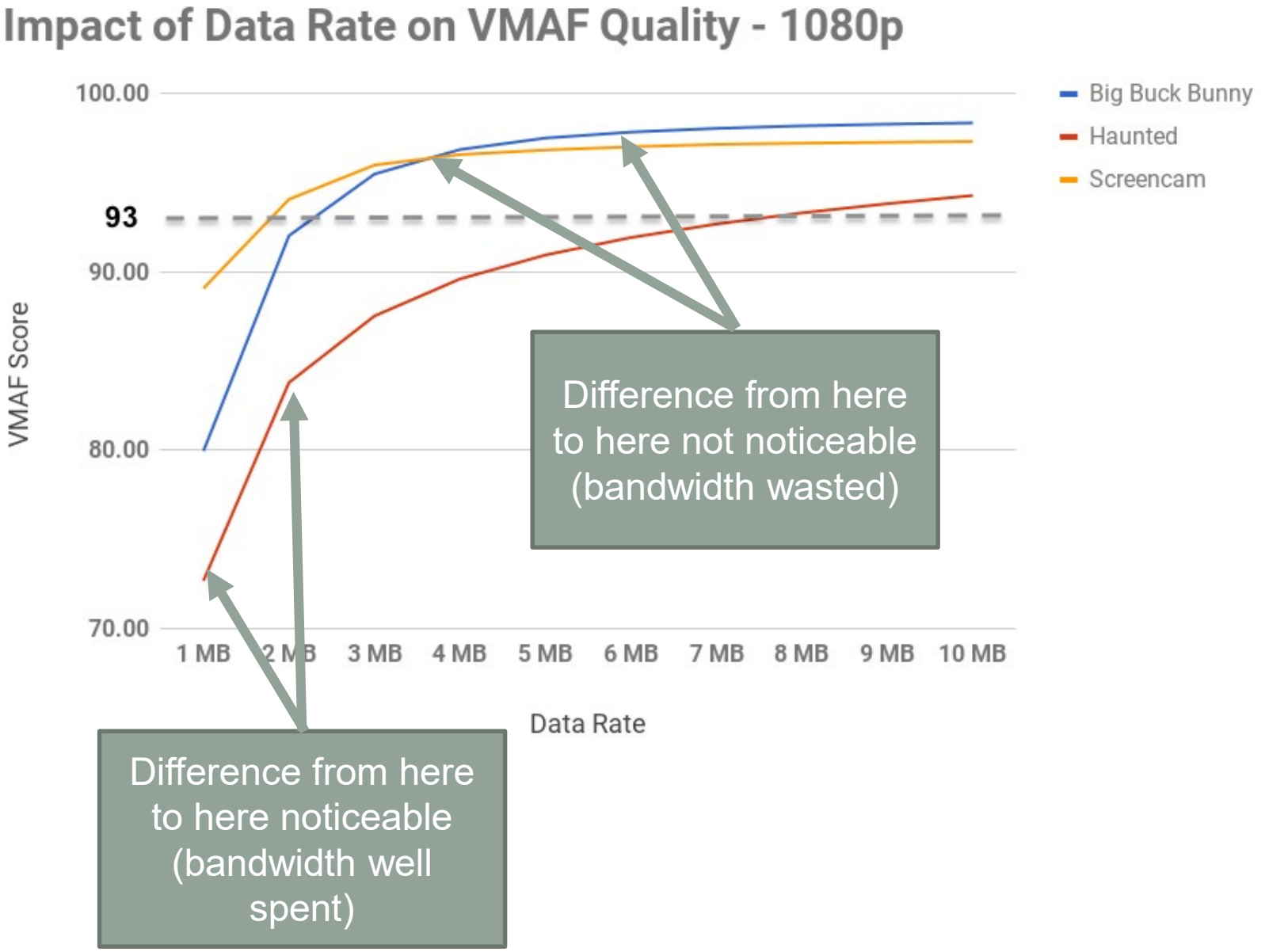
http://bit.ly/VMAF_journey

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.***
  - http://bit.ly/vrqm_5

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



VMAF Models

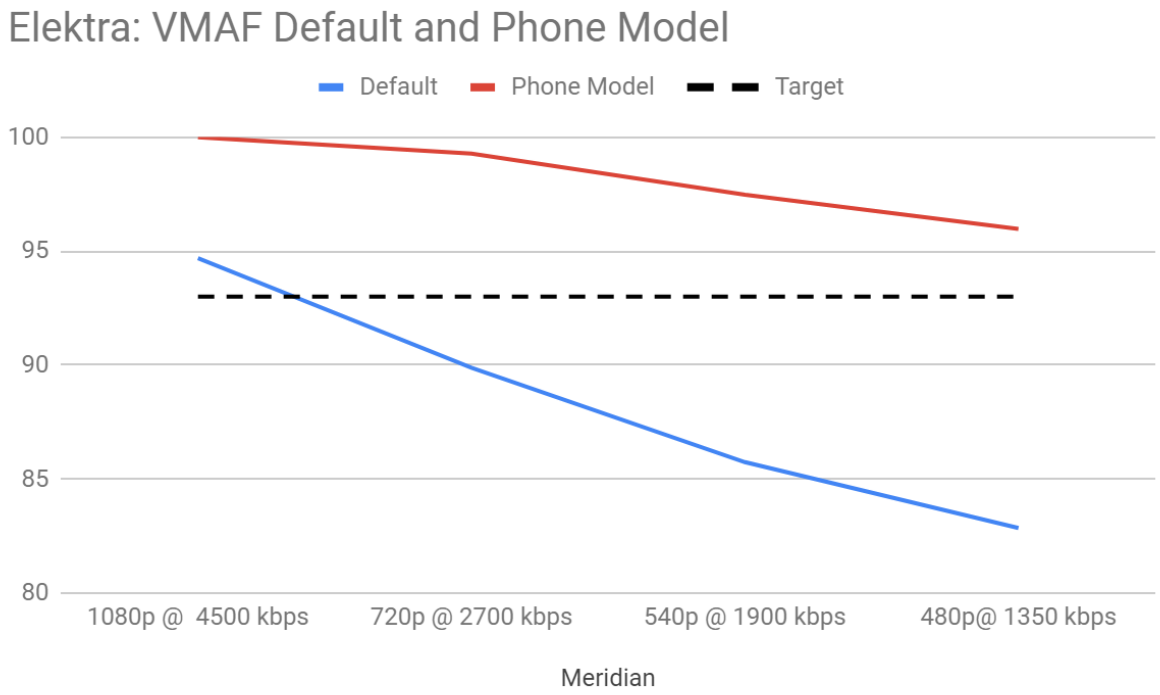
- Original (Default) model
  - Assumes that viewers watch a 1080p display with the viewing distance of 3x the screen height (3H).
- Phone model
  - Assume viewers watch on a mobile phone
- 4K Model
  - Video displayed on a 4K TV and viewed from a distance of 1.5H

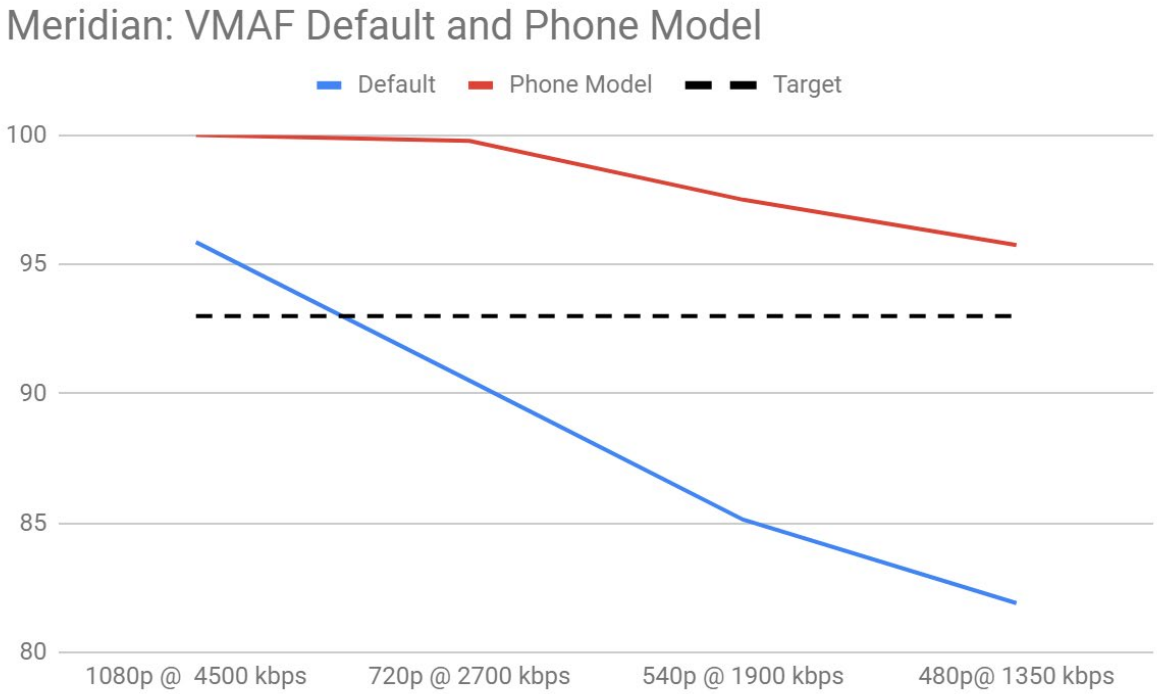
1080p display

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

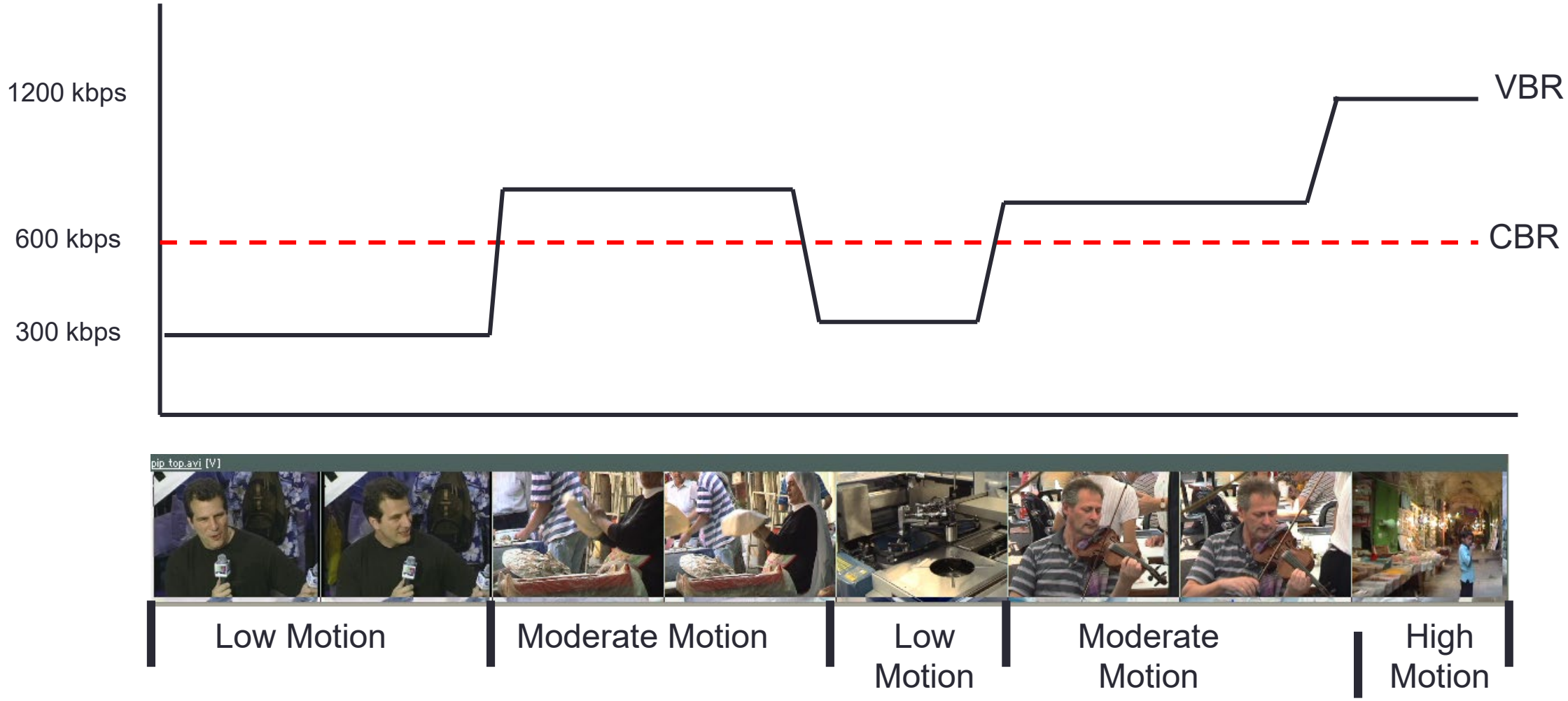
Questions

Should be: 9:35

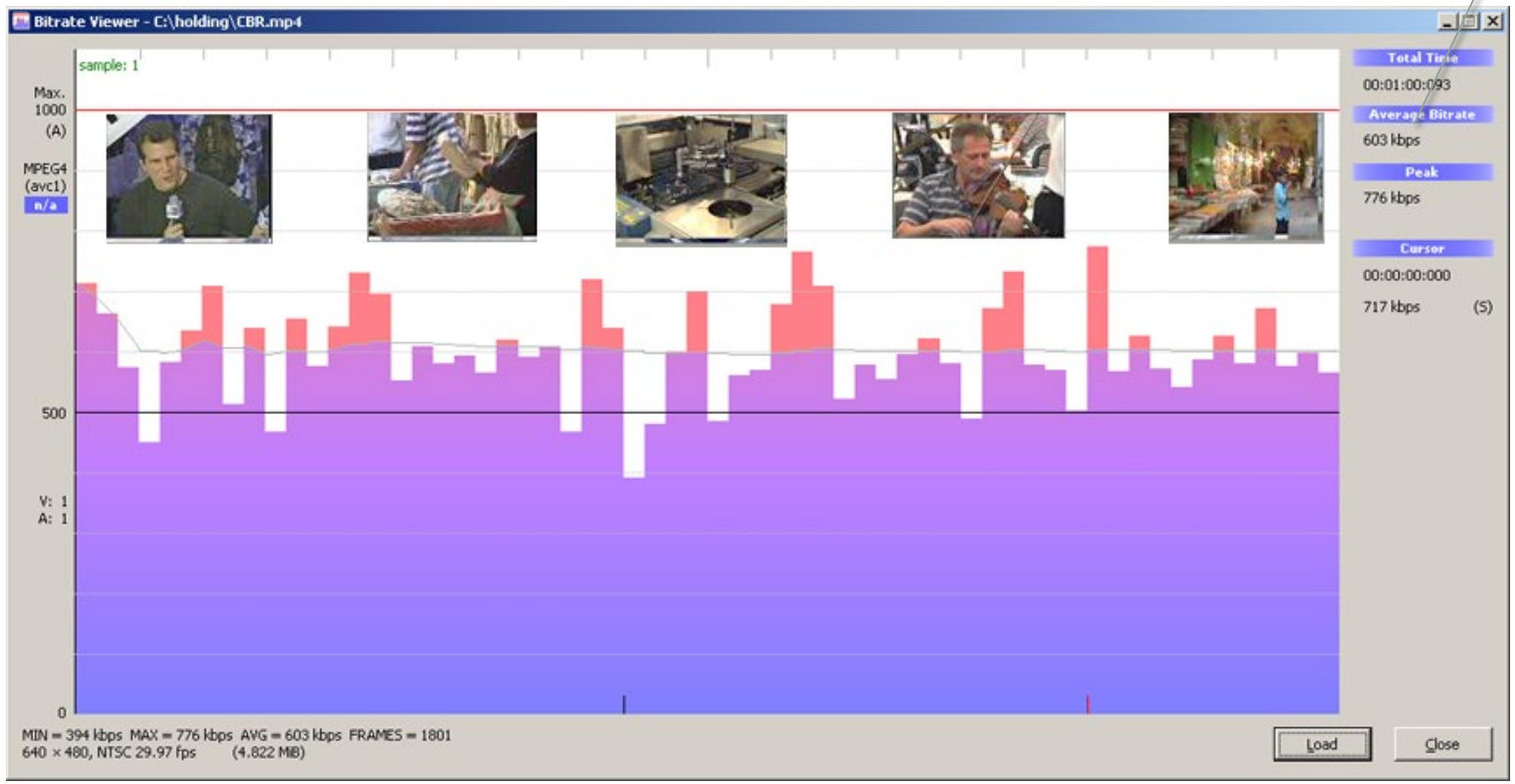
Lesson 3: Bitrate Control

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

How VBR and CBR Work



CBR File Illustrated



- Faint (sorry) wavy blue line is data rate
- Relatively consistent throughout

VBR File Illustrated

596 kbps Average

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

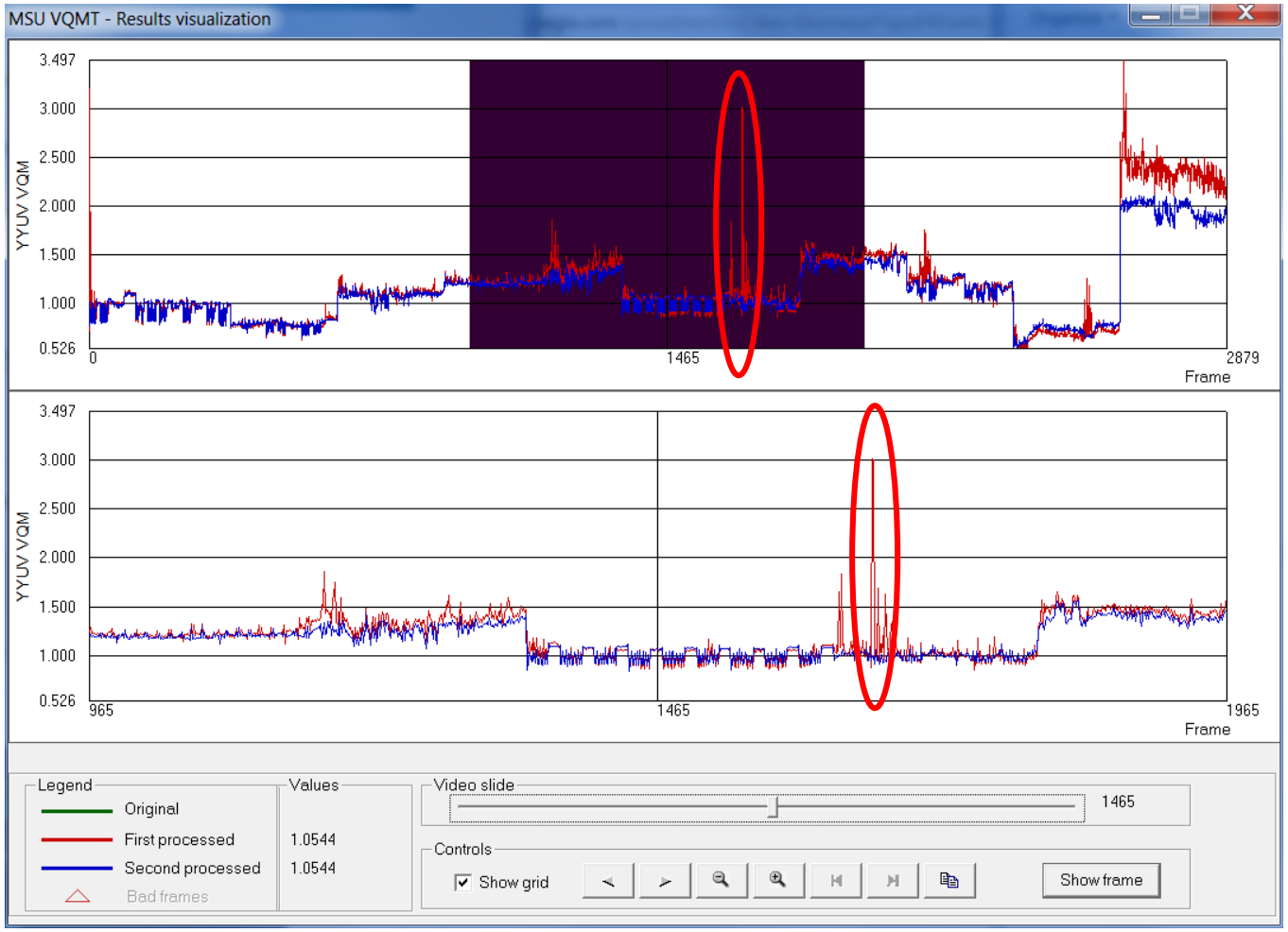
How Much Better Quality is VBR over CBR?

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

- Across the spectrum of content – not that much – average .58 VMAF at 1080p

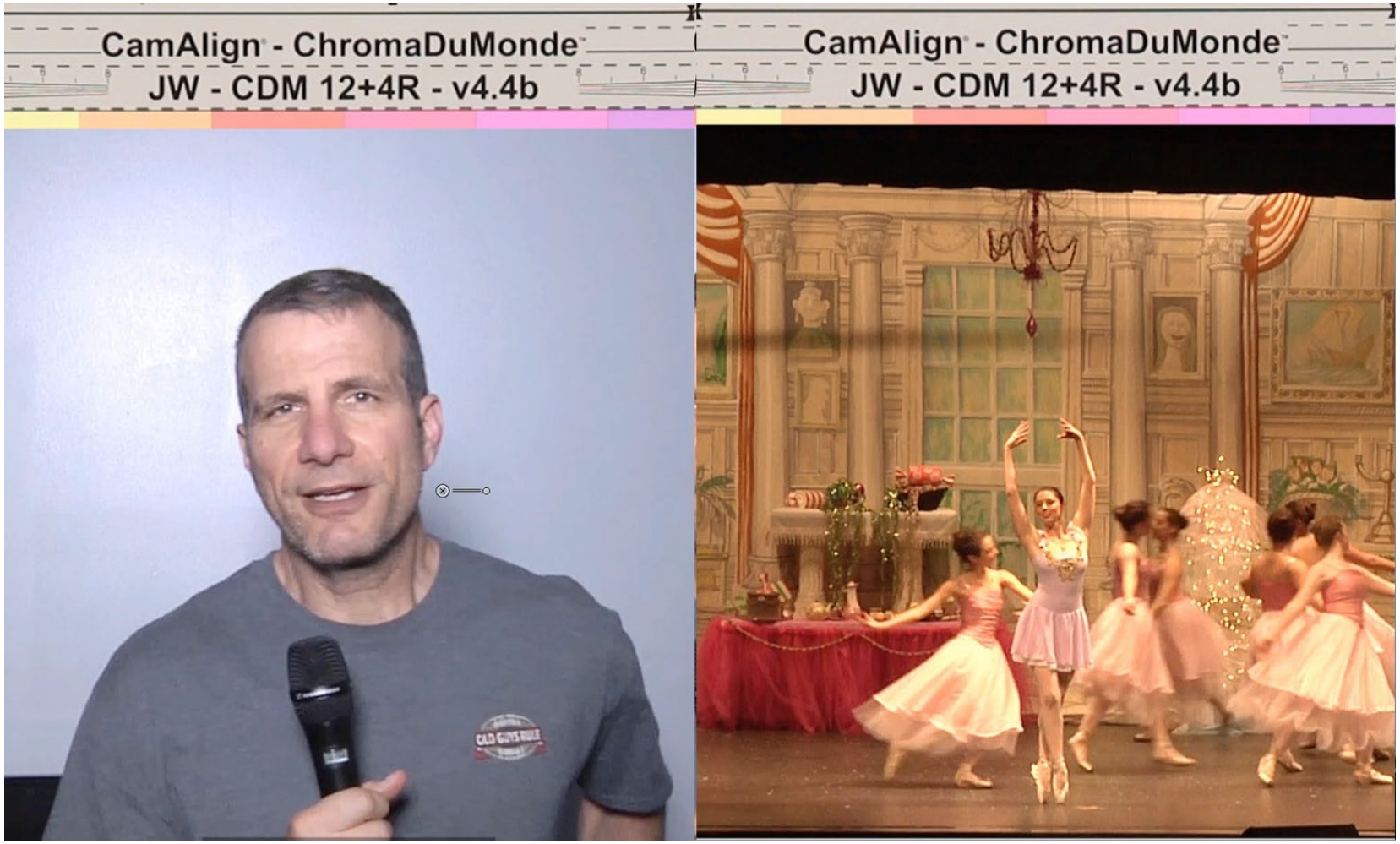
With Some Files, There May Be Spikes Where CBR Gets Ugly

- 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

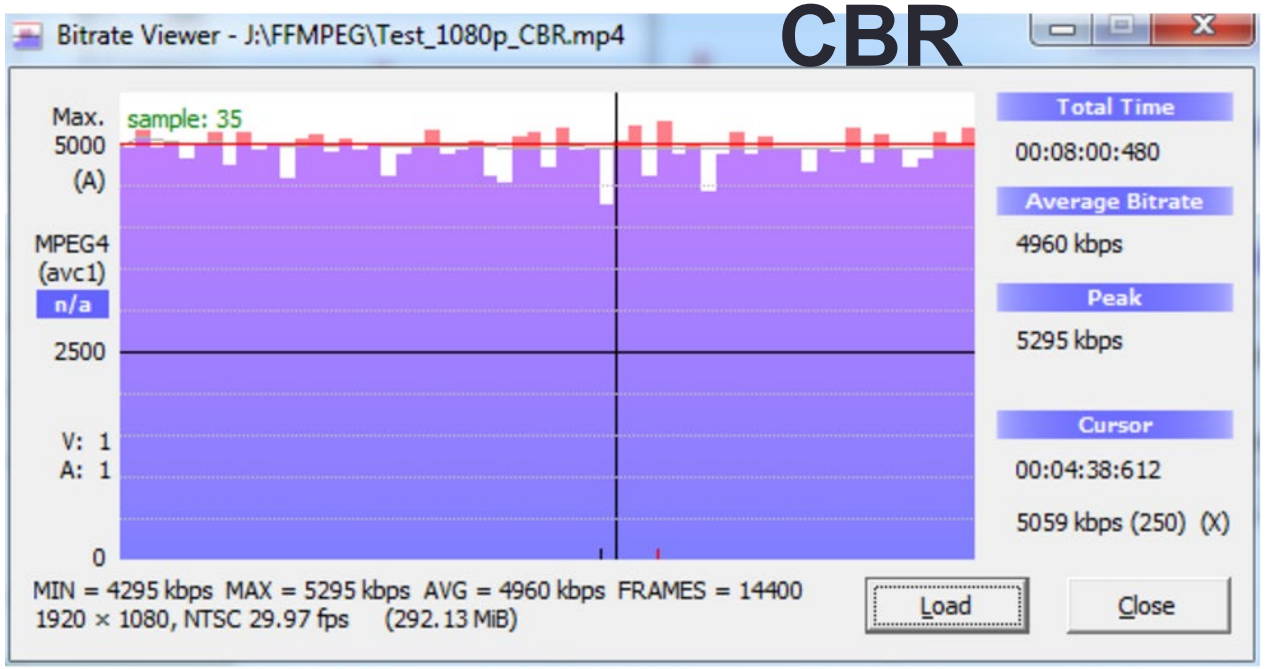
Bitrate Control Test Video

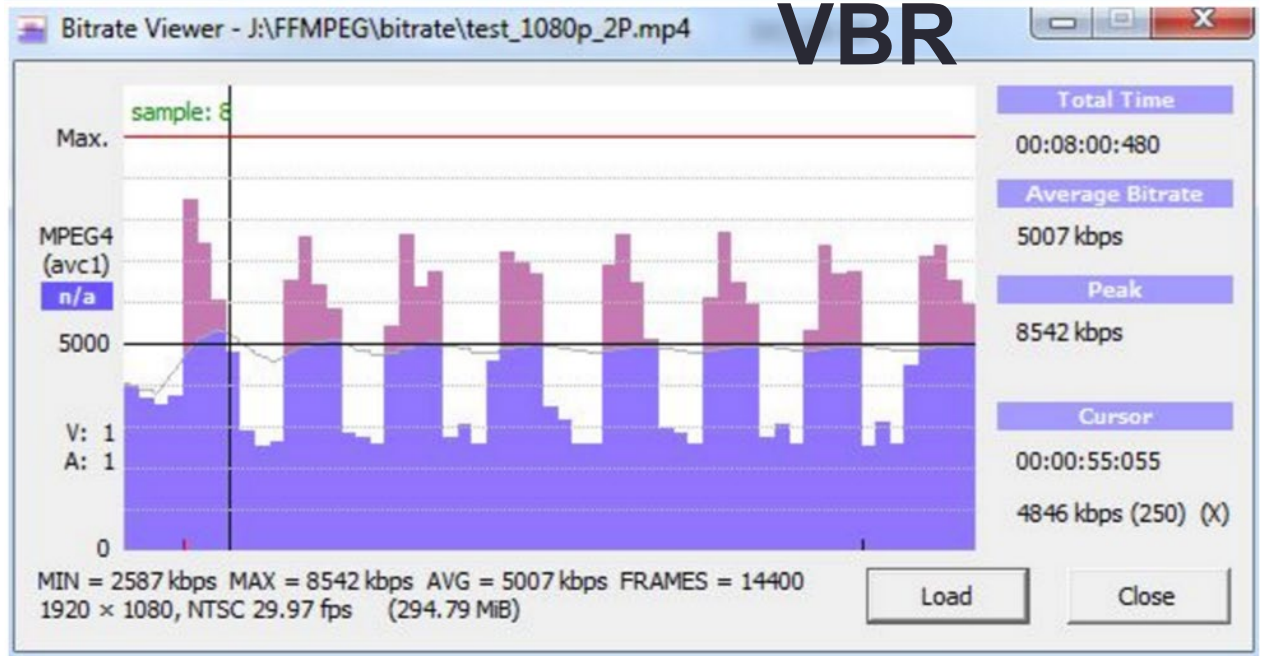


30 seconds talking head/30 seconds ballet

Deliverability

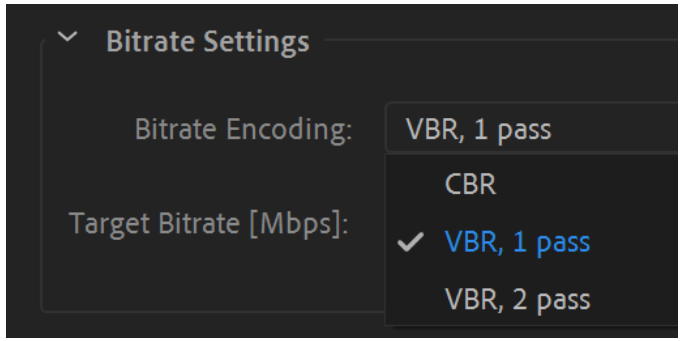
- Which file is easier to deliver over fixed bandwidth connections?
  - Overall bitrate very similar (CBR slightly higher)
  - But, data rate is much more predictable, and therefore easier to deliver
- So, limit variability by implementing *constrained* VBR
  - Limit peaks to % over target

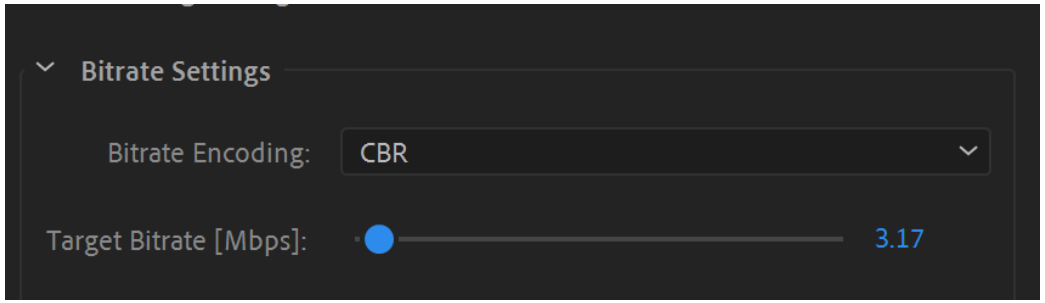




Producing CBR

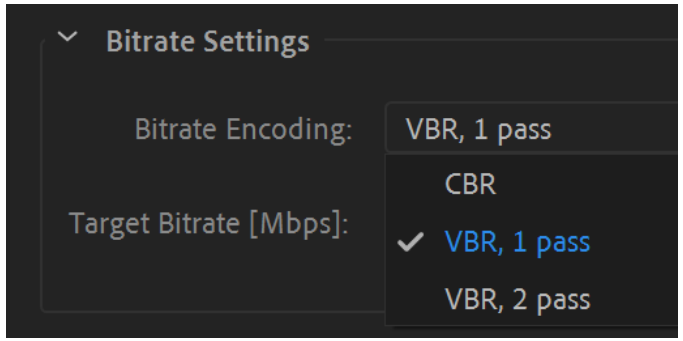
- Typical uses:
  - Live
  - Streaming to constrained lower bitrate connections like 3G
- Typically single-pass, but can be two-pass
  - Adobe Media Encoder – single pass only
  - Choose CBR, then choose target bitrate

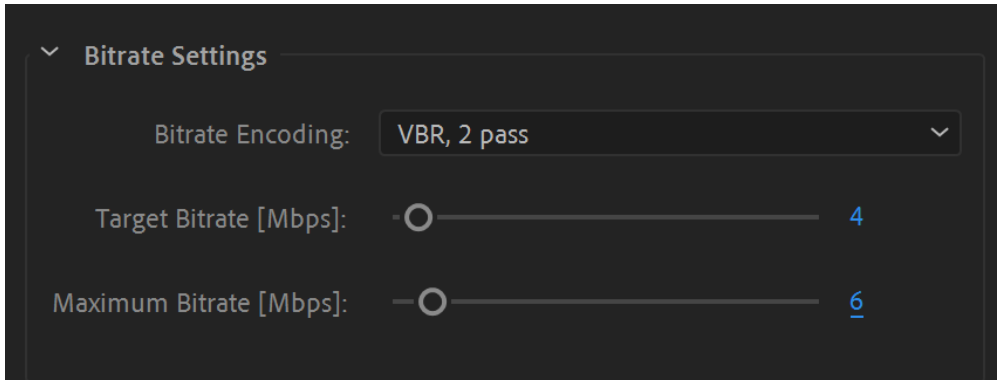




Producing VBR

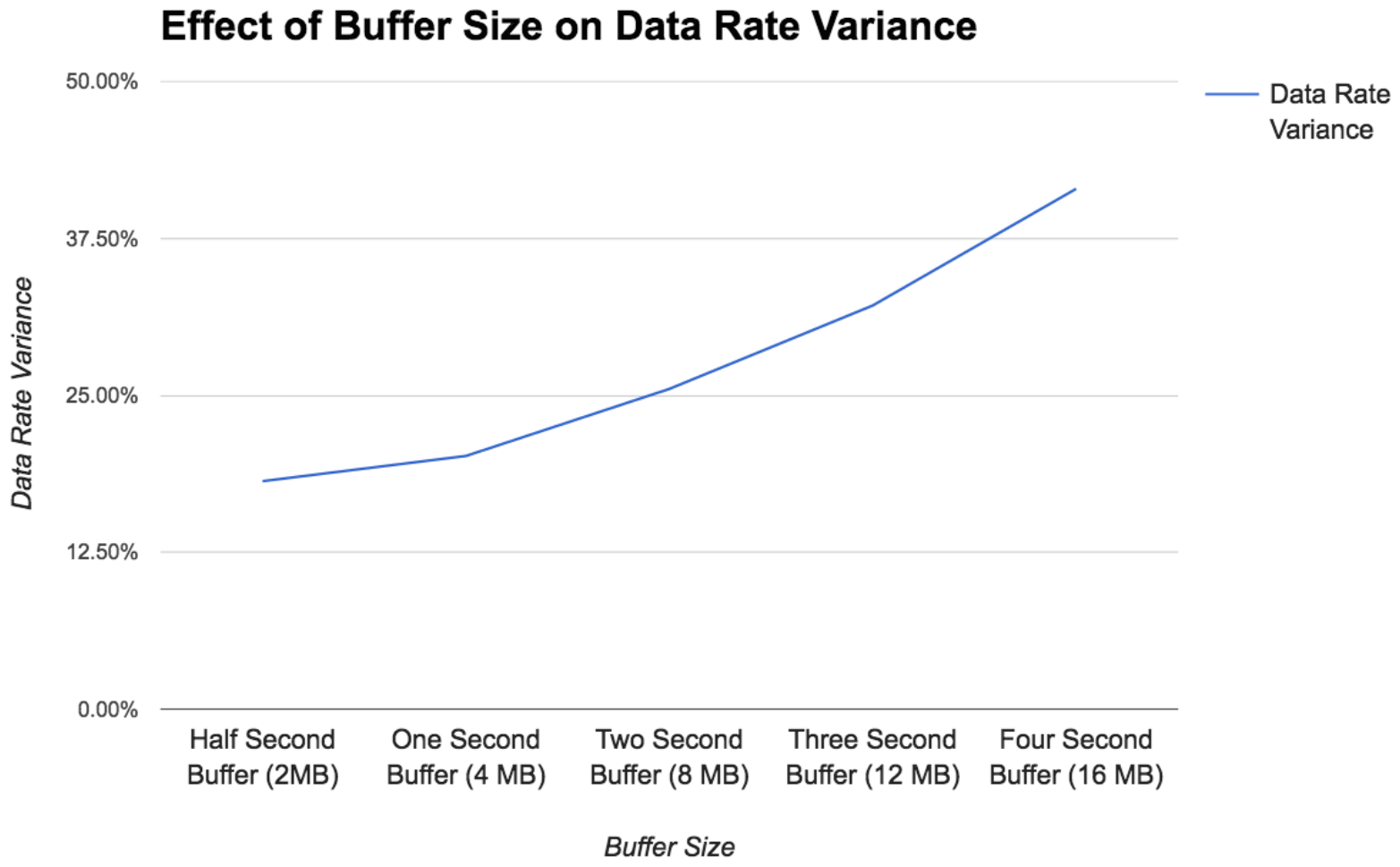
- Typical uses
  - Most VOD streaming
  - Most mezz file creation
- Typically two-pass, but can be single or multiple
  - Adobe Media Encoder – 1 and 2 pass (typically choose 2 pass)
  - Choose VBR, then choose:
    - Target
    - Maximum (1.1x – 2x, here 1.5x)
    - Sometimes minimum (typically .5x)





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

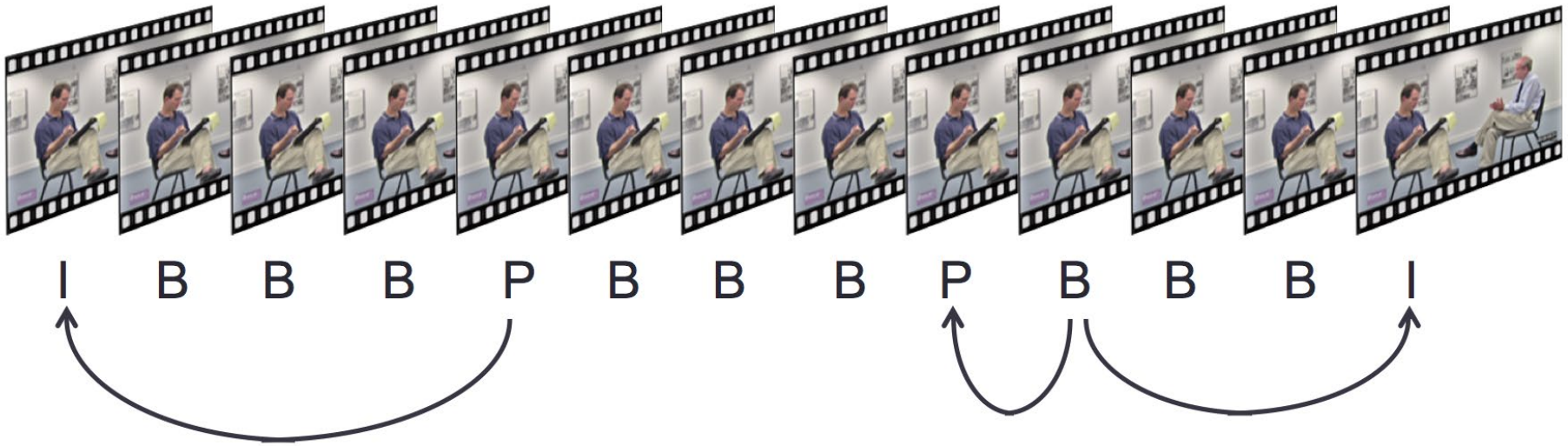
Questions

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

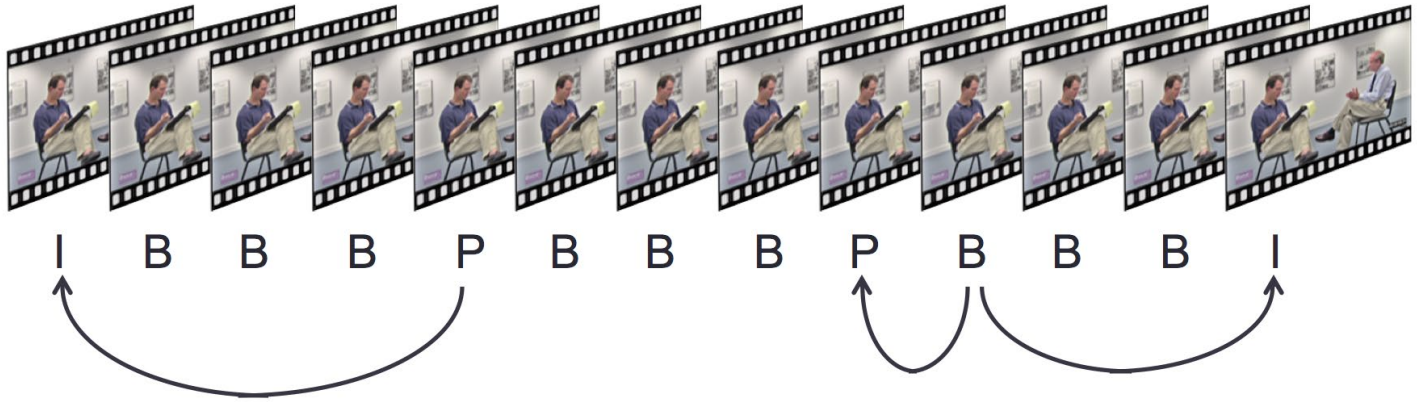


- I- frame – complete frame
  - Least efficient
- P-frame – predictive frame
  - Can look backwards for interframe redundancies
- B-frame - bi-directional predictive frame
  - Can look forwards and backwards for redundancies
- Group of Pictures – GOP –
  - From I-frame to frame immediately preceding 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



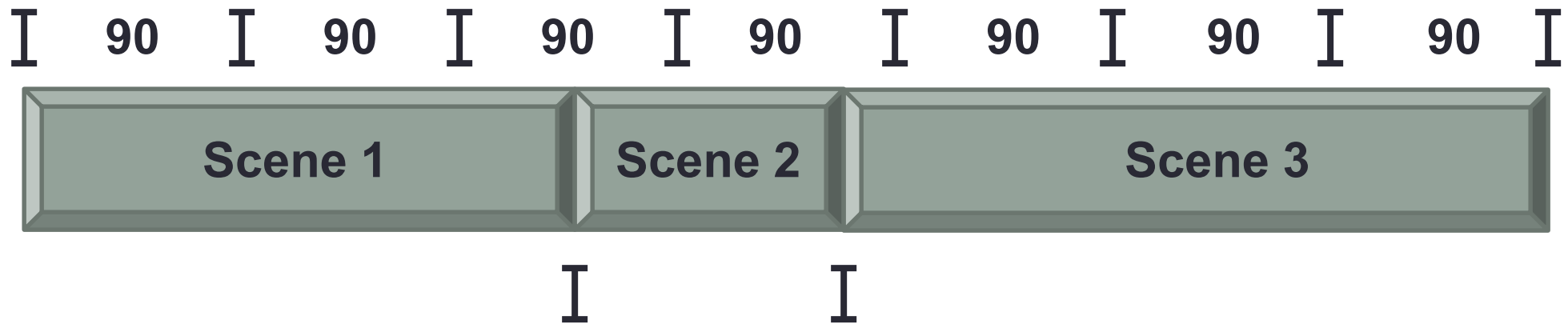
- I- frame – complete frame
  - Least efficient frame
  - Want as few as possible
- All playback starts with I-frame
  - For files that will be interactively viewed want regular keyframes

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)***

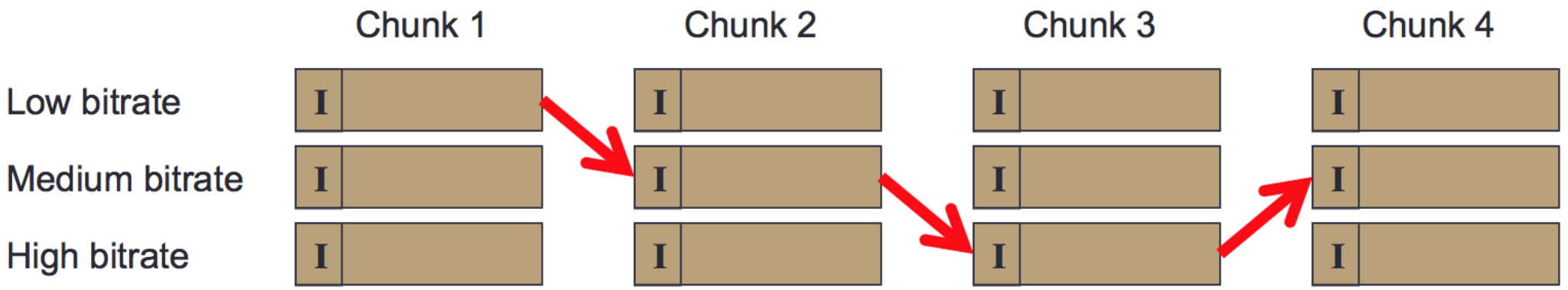
	.5 Sec	1 Sec	2 Sec	3 Sec	5 Sec	10 Sec	Max Delta
Tears of Steel	38.22	39.05	39.49	39.64	39.74	39.87	4.32%
Sintel	37.09	38.06	38.57	38.75	38.97	39.08	5.37%
Big Buck Bunny	37.03	37.93	38.52	38.68	38.64	39.09	5.57%
Talking Head	43.63	44.10	44.40	44.51	44.61	44.68	2.42%
Freedom	40.33	40.67	40.88	40.96	40.99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

Scene Detection



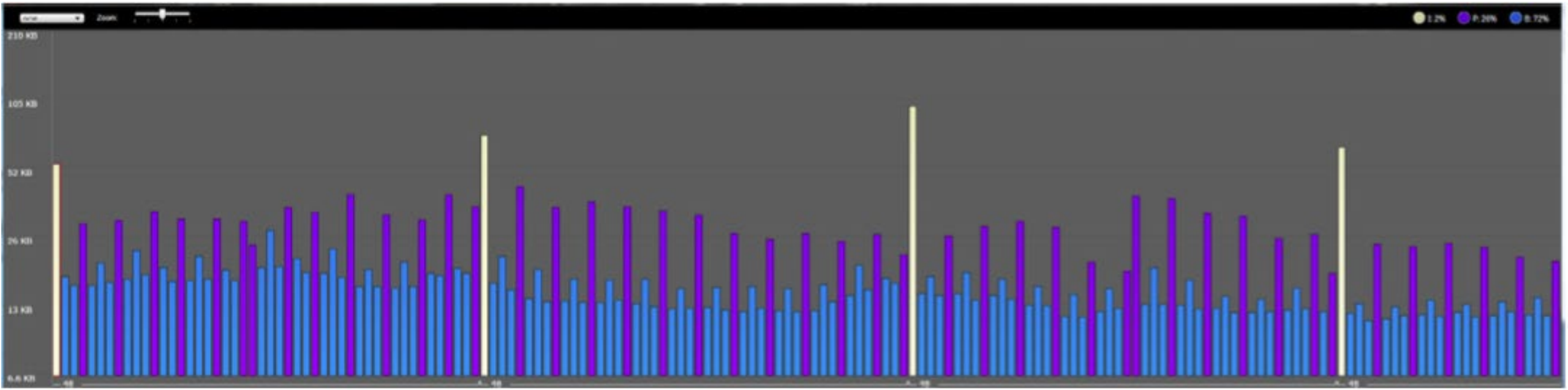
- Scene change detection
  - Inserts I-frame at scene change to improves overall quality
  - For single files, enable I-frames at scene changes

I-Frames and Adaptive Groups



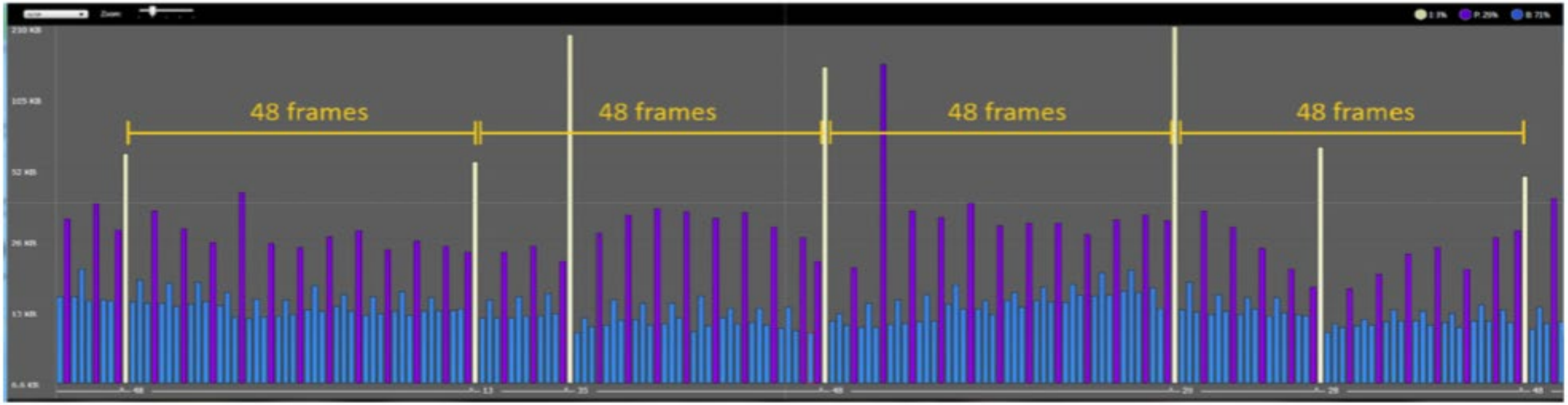
- Need I-frame at start of each segment
  - I-frame interval must divide into segment size
  - 6 second segments, use 1, 2, 3, or 6
    - Apple spec calls for 2-second interval
- Need regular I-frames: Either
  - Disable scene change detection
    - Typically what I do
  - Force keyframes at specified 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

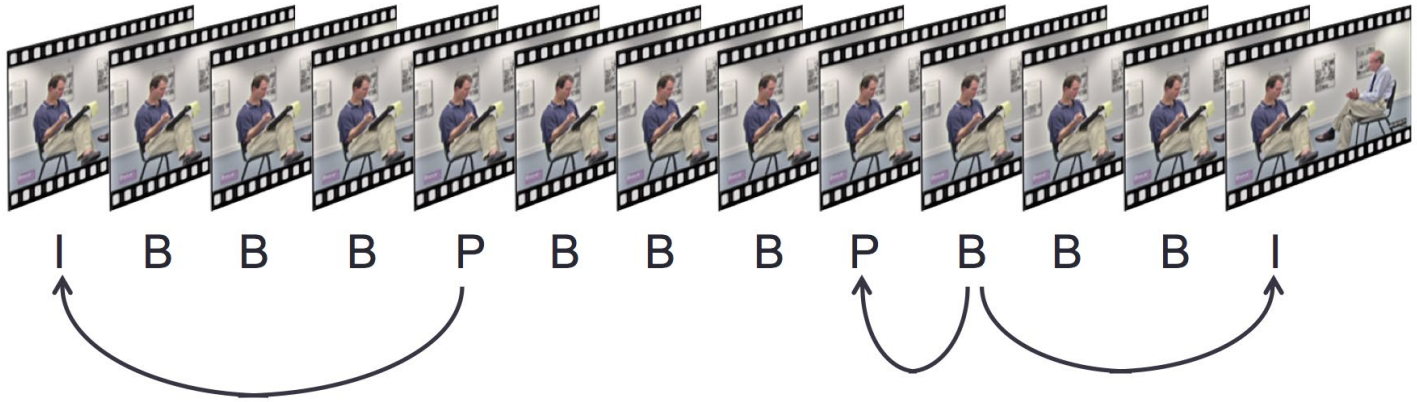
Single File

- 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

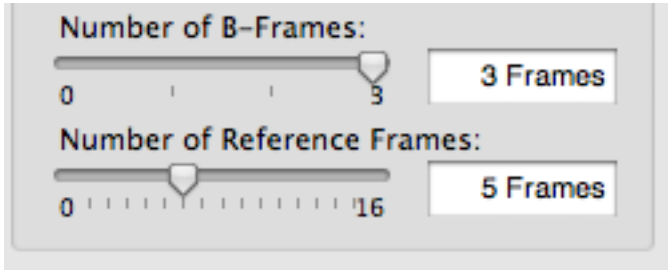
About B-Frames



- 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; (IBBBPBBBPBBBPBBBP)
  - 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

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

- 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

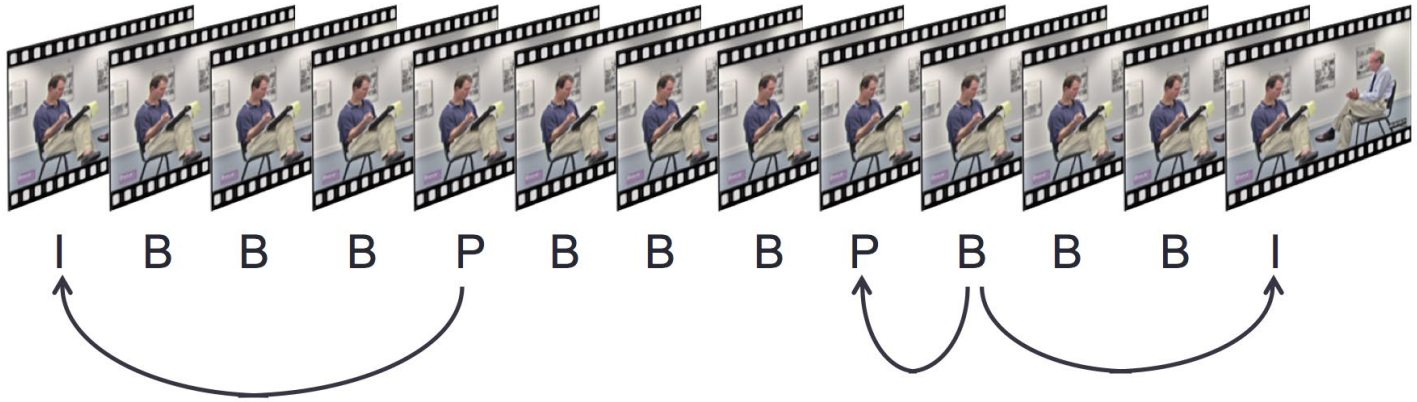
B-Frame Recommendations

- 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

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

- For most files, 16 delivers the most quality
- Max delta is miniscule

Reference Frames and Encoding Time

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

- 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 acces 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

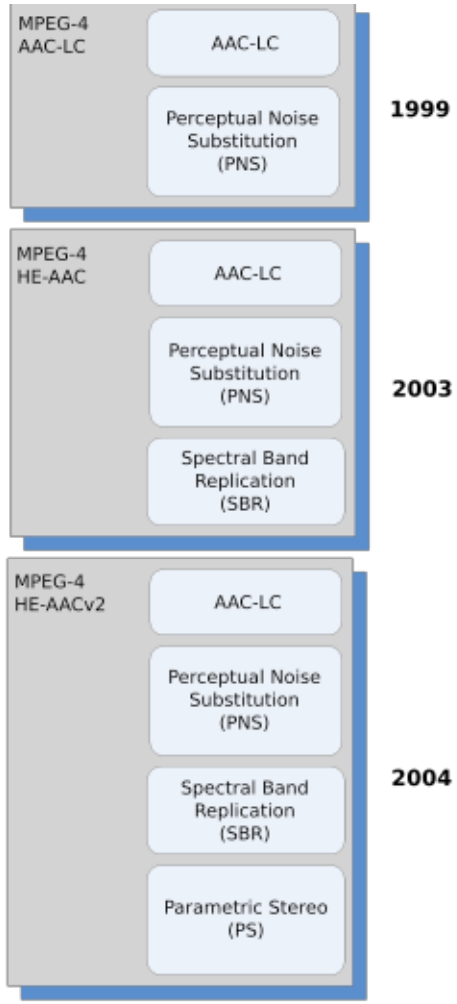
- About H.264
- Encoding with H.264
  - Profiles
  - Levels
  - Entropy coding
  - X264 presets

What is H.264?

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

	ITU – International Telecommunications Union Telephone, Radio, TV	ISO – International Standardization Organization Photography, Computer, Consumer Electronics
1984	H.120	
1990	H.261 – Video Conferencing	
1993		MPEG-1 – Video CD
1994	(H.262)	MPEG-2 – Digital Cable and Satellite TV
1995	H.263 – Improved Video Conferencing	
1997		ATSC – U.S. HDTV
1999		MPEG-4
2002	AVC (H.264)	AVC (MPEG-4 Part 10)

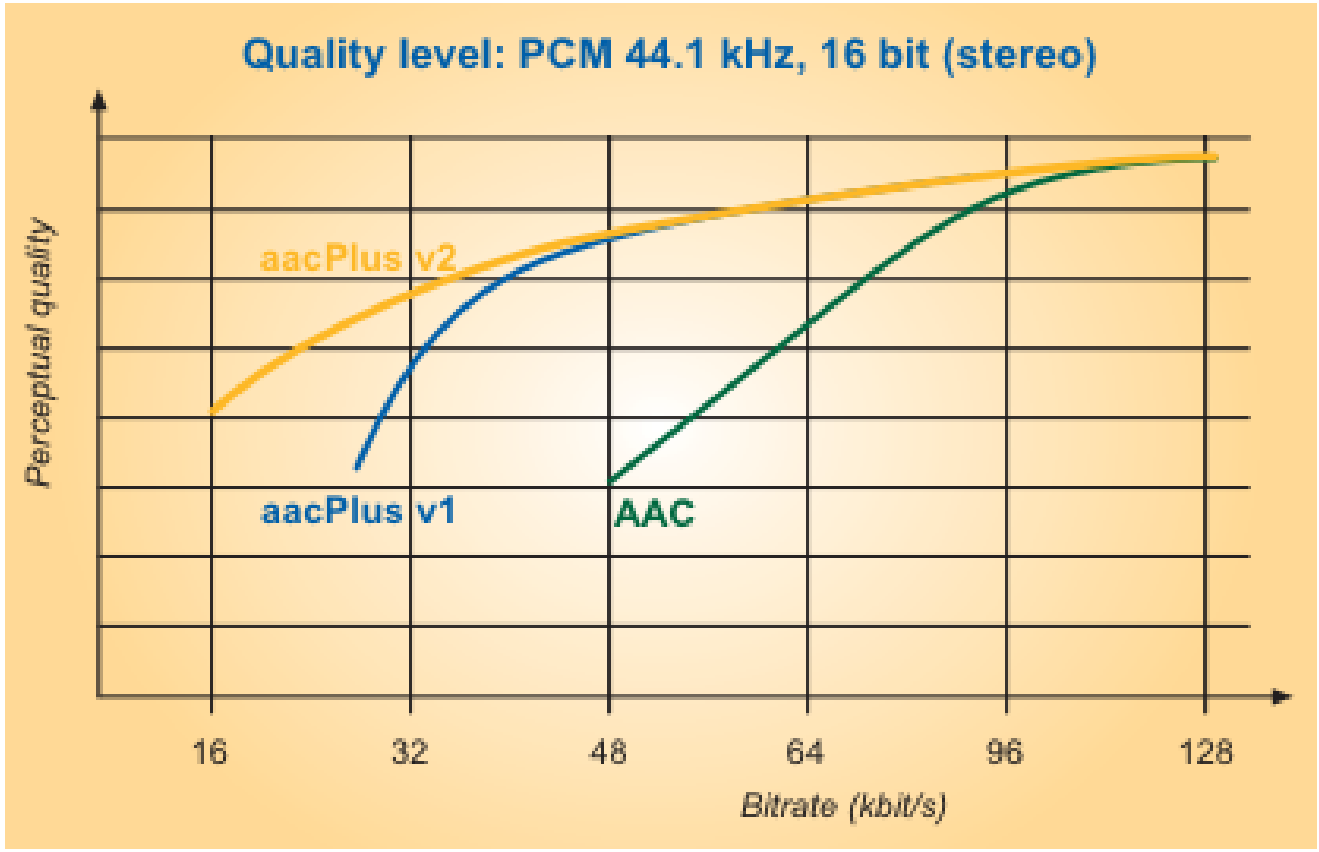
MPEG-4 Audio



- 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

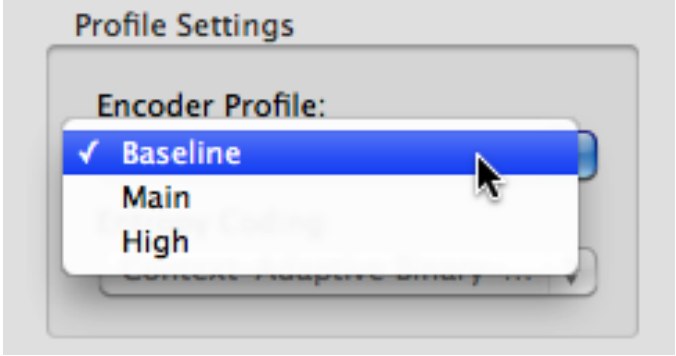
What Profiles are and Why They Exist

- 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

	Baseline	Main	High
I and P Slices	Yes	Yes	Yes
B Slices	No	Yes	Yes
Multiple Reference Frames	Yes	Yes	Yes
In-Loop Deblocking Filter	Yes	Yes	Yes
CAVLC Entropy Coding	Yes	Yes	Yes
CABAC Entropy Coding	No	Yes	Yes
Interlaced Coding (PicAFF, MBAFF)	No	Yes	Yes
8x8 vs. 4x4 Transform Adaptivity	No	No	Yes
Quantization Scaling Matrices	No	No	Yes
Separate Cb and Cr QP control	No	No	Yes
Separate Color Plane Coding	No	No	No
Predictive Lossless Coding	No	No	No
	Baseline	Main	High

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

VMAF-Average	Baseline	Main	High	Delta - Baseline/Main	Delta - Main/High	Total Delta
Tears of Steel	92.83	95.46	96.23	2.83%	0.80%	3.66%
Sintel	93.65	95.78	96.38	2.27%	0.63%	2.91%
Big Buck Bunny	92.14	94.72	95.52	2.80%	0.83%	3.67%
Talking Head	94.35	94.93	95.19	0.61%	0.28%	0.90%
Freedom	92.87	94.65	95.36	1.91%	0.74%	2.67%
Haunted	89.56	91.11	91.99	1.73%	0.95%	2.70%
Screencam	92.80	94.01	94.34	1.30%	0.35%	1.66%
Tutorial	95.85	96.13	96.15	0.29%	0.03%	0.32%
Average	**93.01**	**94.60**	**95.14**	**1.72%**	**0.57%**	**2.31%**

- 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%

iOS History Lesson

Width	Height	Frame Rate	Video Bitrate	Audio Bitrate	I-Frame	Profile	B-frames	Segment Size	iPod Touch 2-4	iPod Touch 5	iPhone 3G, 3GS, 4	iPhone 4S, 5, 5C, 5S	iPad 1,2	iPad 3, 4, 5	Apple TV 2	Apple TV 3
416	234	12	200	64	36	Baseline	NA	9	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
480	270	15	400	64	45	Baseline	NA	9	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
640	360	29.97	600	64	90	Baseline	NA	9	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
640	360	29.97	1200	96	90	Baseline	NA	9		Yes		Yes	Yes	Yes	Yes	Yes
960	540	29.97	3500	96	90	Main		9		Yes		Yes	Yes	Yes	Yes	Yes
1280	720	29.97	5000	128	90	Main	As	9		Yes		Yes	Yes	Yes	Yes	Yes
1280	720	29.97	6500	128	90	Main	needed	9		Yes		Yes	Yes	Yes	Yes	Yes
1920	1080	29.97	8500	128	90	High		9		Yes		Yes		Yes		Yes

- Initial version of TN2224 customized profile for different targets

Current HLS Authoring Specs Abandon Legacy Devices

HDR (HEVC) 30 fps	HEVC/H.265 30 fps	H.264/AVC	Resolution 16:9 aspect ratio	Frame rate
160	145	145	416 x 234	≤ 30 fps
360	300	365	480 x 270	≤ 30 fps
800	660	730	640 x 360	≤ 30 fps
1200	990	1100	768 x 432	≤ 30 fps
2050	1700	2000	960 x 540	same as source
2900	2400	3000	1280 x 720	same as source
3850	3200	4500	1280 x 720	same as source
5400	4500	6000	1920 x 1080	same as source
7000	5800	7800	1920 x 1080	same as source
9700	8100	n/a	2560 x 1440	same as source
13900	11600	n/a	3840 x 2160	same as source
20000	16800	n/a	3840 x 2160	same as source

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

http://bit.ly/A_Devices_Spec

Encoding for Android Devices

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

	SD (Low quality)	SD (High quality)	HD 720p (N/A on all devices)
Video resolution	176 x 144 px	480 x 360 px	1280 x 720 px
Video frame rate	12 fps	30 fps	30 fps
Video bitrate	56 Kbps	500 Kbps	2 Mbps
Audio codec	AAC-LC	AAC-LC	AAC-LC
Audio channels	1 (mono)	2 (stereo)	2 (stereo)
Audio bitrate	24 Kbps	128 Kbps	192 Kbps

- 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
- Class poll?

How Much Quality Difference in Encoding Ladder?

Talking Head Video

Talking Head	Data Rate	Baseline	High	Delta
234p	145,000	33.79	34.20	1.22%
270p	350,000	35.72	35.99	0.75%
360p	600,000	38.16	38.37	0.54%
540p	1,000,000	40.04	40.33	0.71%
720p	1,500,000	40.78	41.32	1.34%
1080p	2,500,000	43.53	44.11	1.34%
Average		**38.67**	**39.05**	**0.98%**

High Motion Video

Haunted	Data Rate	Baseline	High	Delta
234p	145,000	30.46	31.56	3.61%
270p	365,000	33.14	33.73	1.79%
360p	900,000	35.99	36.38	1.10%
540p	1,500,000	38.09	38.62	1.38%
720p	2,500,000	39.28	39.84	1.42%
1080p	6,000,000	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,

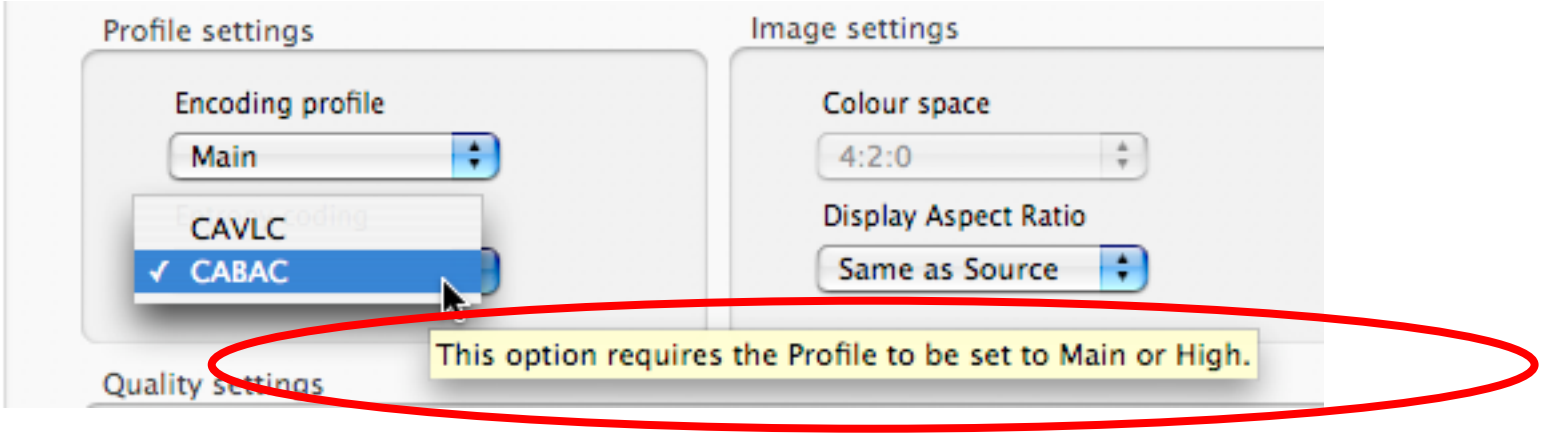
- Levels set further limits on how video can be produced
  - See above specs for the iPhone 7
- In general, solely a concern for mobile

Levels from Wikipedia

Level	Max decoding speed		Max frame size		Max video bit rate for video coding layer (VCL) kbit/s			Examples for high resolution @ highest frame rate (max stored frames)
	Luma samples/s	Macroblocks/s	Luma samples	Macroblocks	Baseline, Extended and Main Profiles	High Profile	High 10 Profile	Toggle additional details
4	62,914,560	245,760	2,097,152	8,192	20,000	25,000	60,000	2,048×1,024@30.0 (4
4.1	62,914,560	245,760	2,097,152	8,192	50,000	62,500	150,000	2,048×1,024@30.0 (4
4.2	133,693,440	522,240	2,228,224	8,704	50,000	62,500	150,000	2,048×1,080@60.0 (4

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

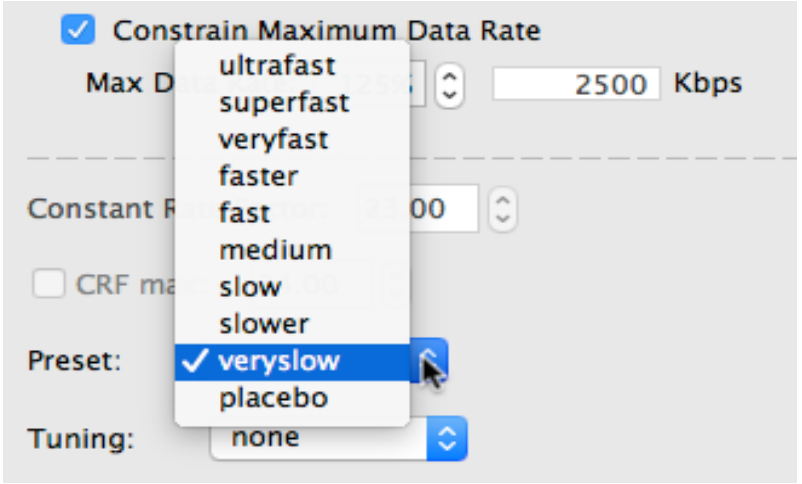
Entropy Encoding



- 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

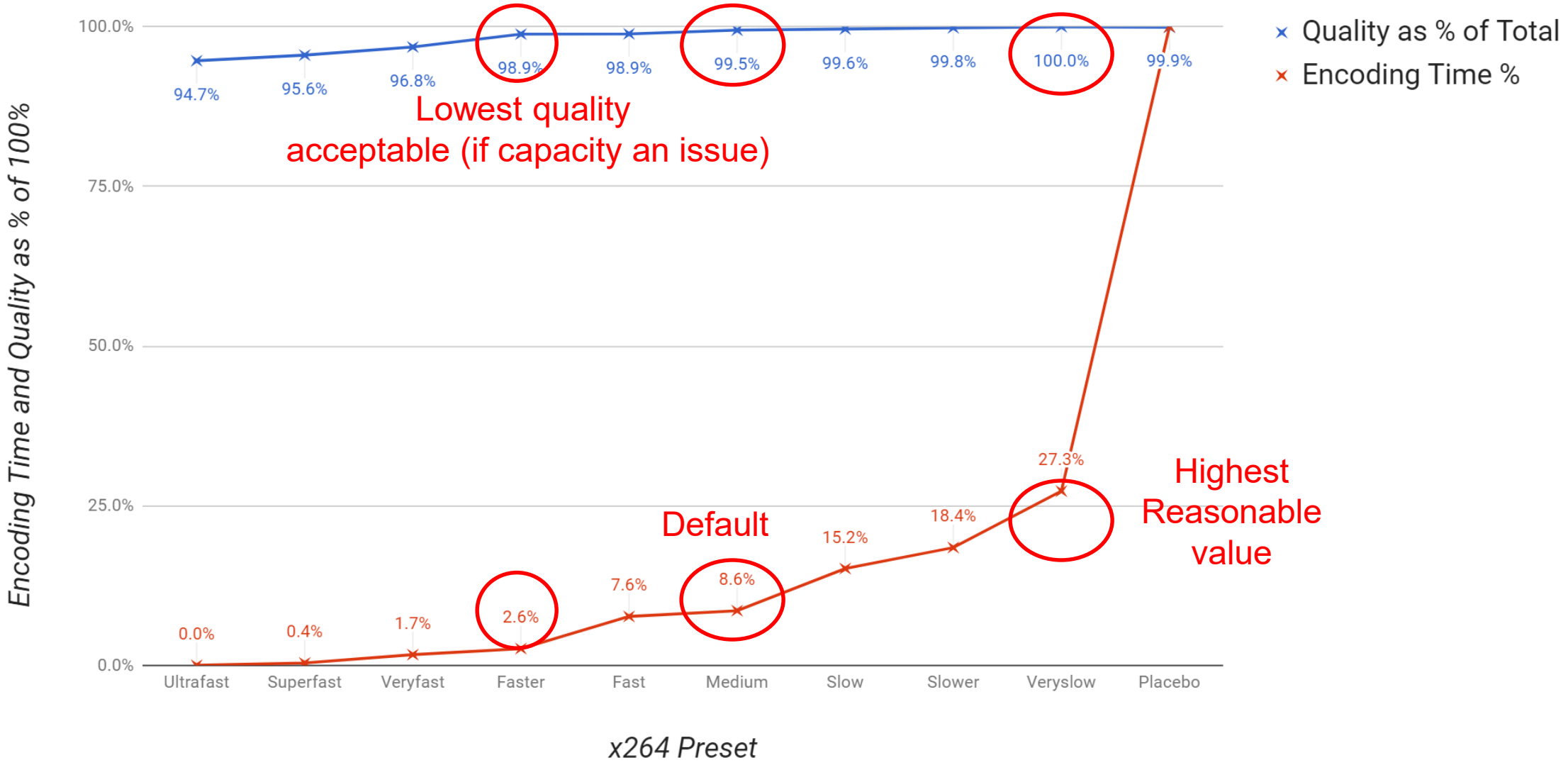
- Eight files
  - 1 movie (Tears of Steal)
  - 2 animations (Sintel, BBB)
  - Two general purpose (concert, advertisement)
  - One talking head
  - Screencam
  - Tutorial (PPT/Video)
- Encode to all presets
- Measure encoding time
- Measure VMAF

Results Please

Average Quality	Ultrafast	Superfast	Veryfast	Faster	Fast	Medium	Slow	Slower	Veryslow	Placebo	Total Delta
Tears of Steel	89.20	92.00	93.29	95.45	95.59	96.22	96.43	96.56	96.67	96.65	8.38%
Sintel	88.29	92.66	93.85	95.84	95.99	96.38	96.56	96.68	96.83	96.75	9.68%
Big Buck Bunny	87.26	91.26	92.68	95.03	95.29	95.53	95.75	95.87	96.05	96.01	10.08%
Talking Head	95.19	92.55	93.66	94.90	94.86	95.18	95.29	95.43	95.51	95.39	3.20%
Freedom	91.95	91.15	92.63	94.58	94.51	95.37	95.59	95.84	96.15	96.04	5.48%
Haunted	91.30	88.61	89.43	91.30	91.08	91.98	92.08	92.35	92.49	92.45	4.38%
Screencam	90.92	92.56	93.52	94.75	94.75	94.70	94.77	94.86	94.92	94.91	4.41%
Tutorial	93.42	94.66	95.55	96.16	96.17	96.17	96.26	96.28	96.29	96.10	3.07%
Average	**90.53**	**91.37**	**92.59**	**94.52**	**94.56**	**95.11**	**95.28**	**95.46**	**95.62**	**95.55**	**6.08%**

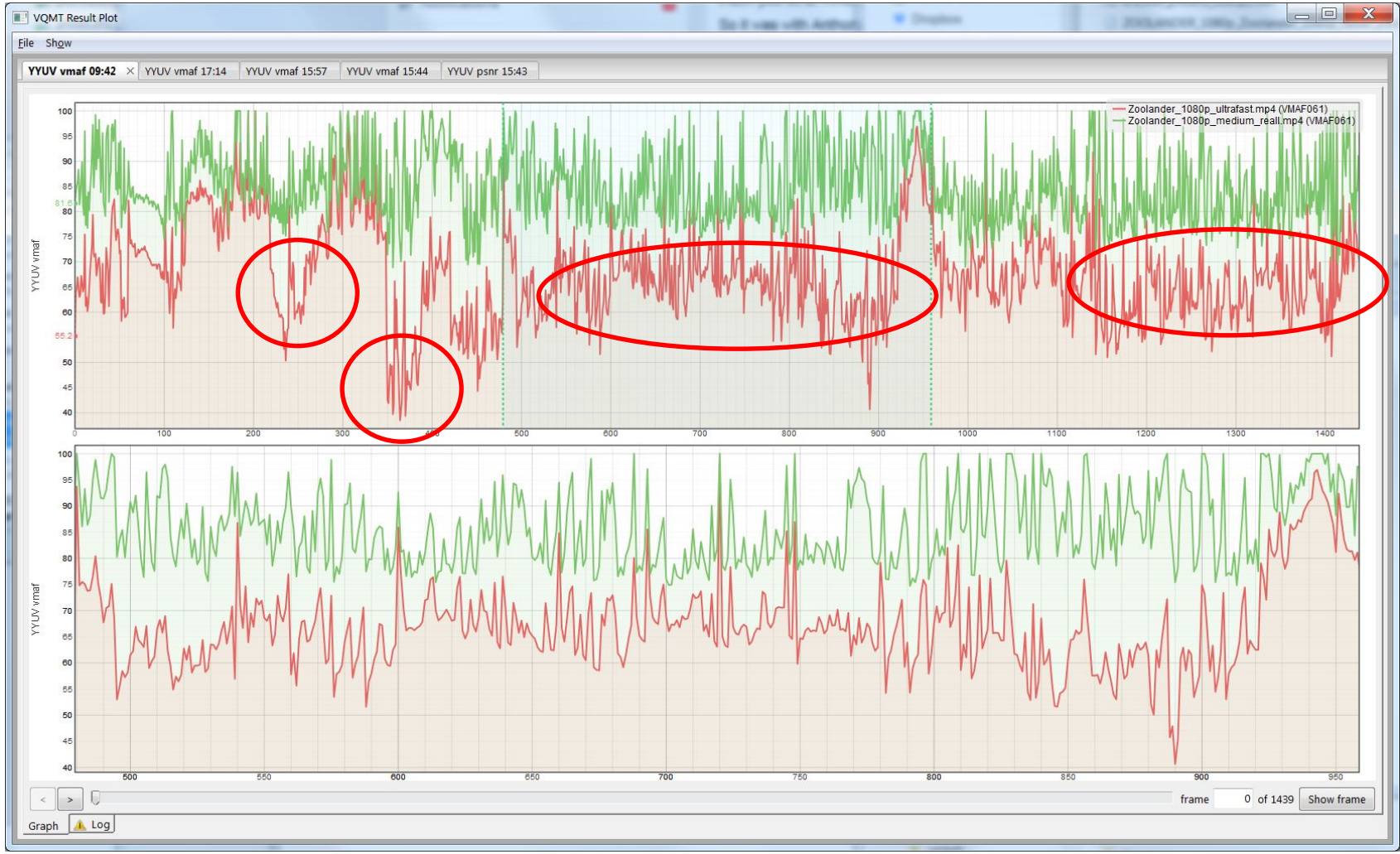
- Red is lowest quality
- Green highest quality
- Note top values – average 95.62 (not Placebo)
- Very slow averages best quality
  - But only 8% spread between best and worst

Quality and Encoding Time of x264 Presets



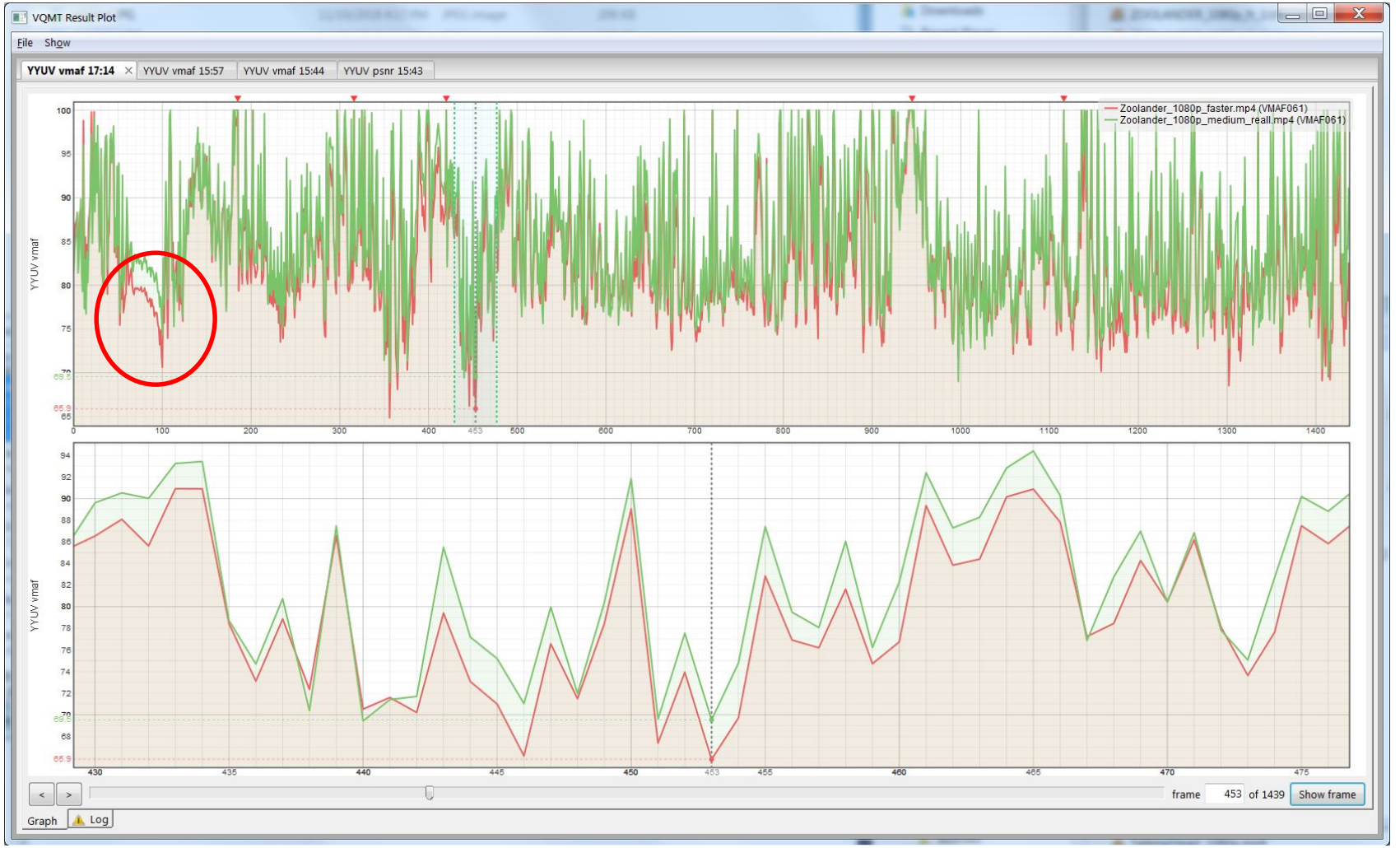
Check Results Plot – Ultrafast (red) vs Medium

- Plot of VMAF values over duration of clip
  - Red is ultrafast
  - Green is Medium
- Multiple deep drops that would be noticeableNever 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

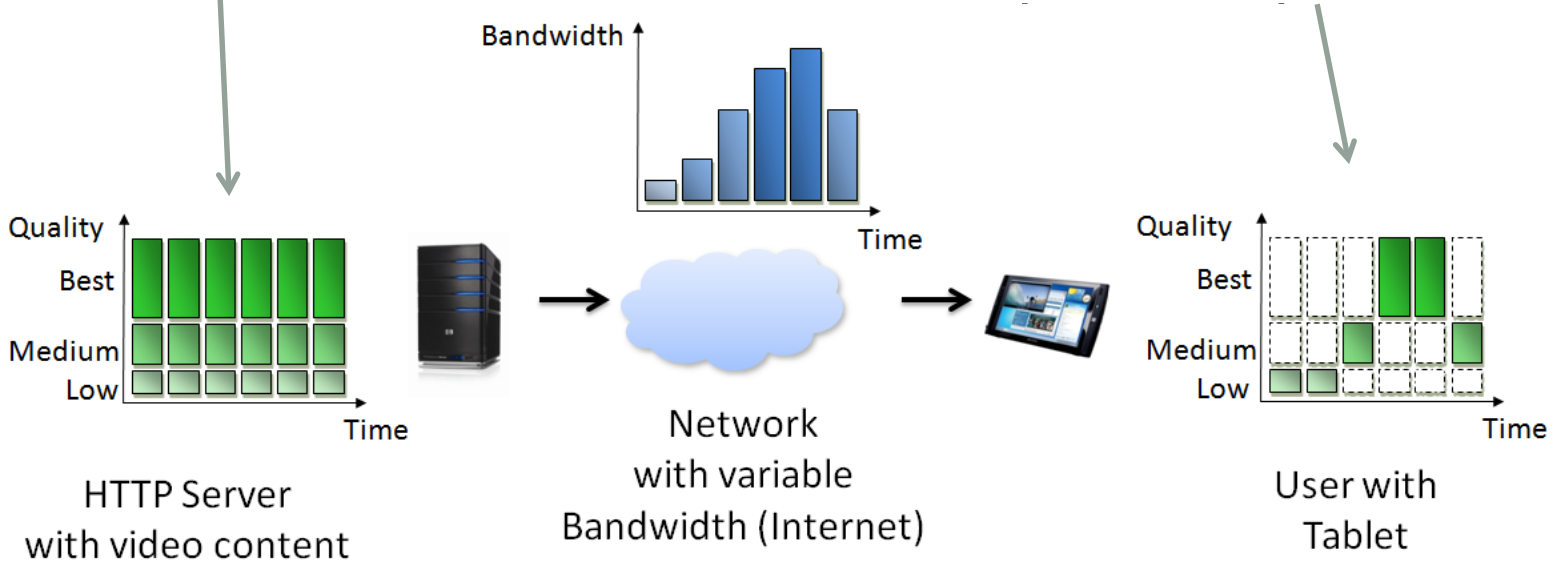
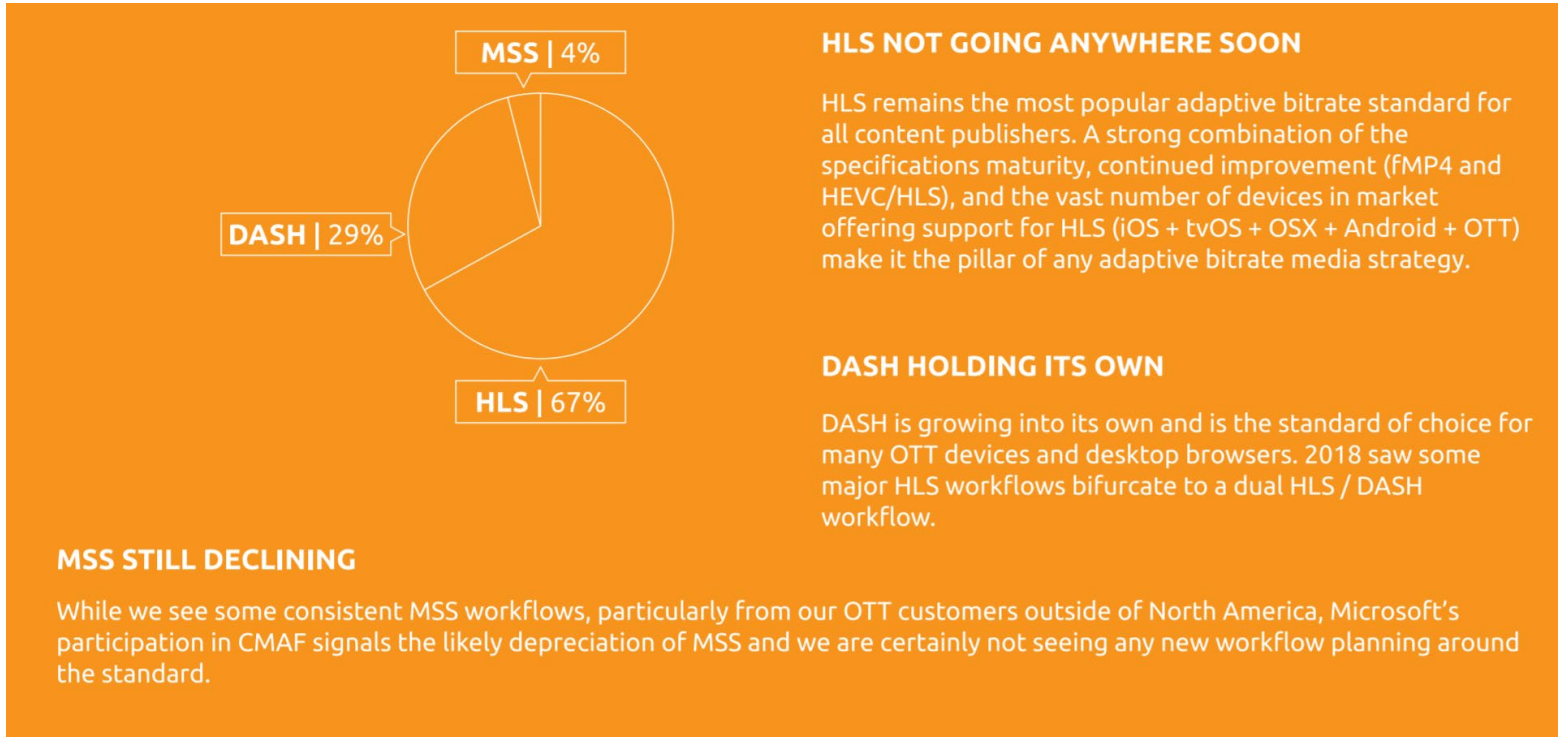


Illustration courtesy of www.bitmovin.net

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
http://bit.ly/glob_med_2019

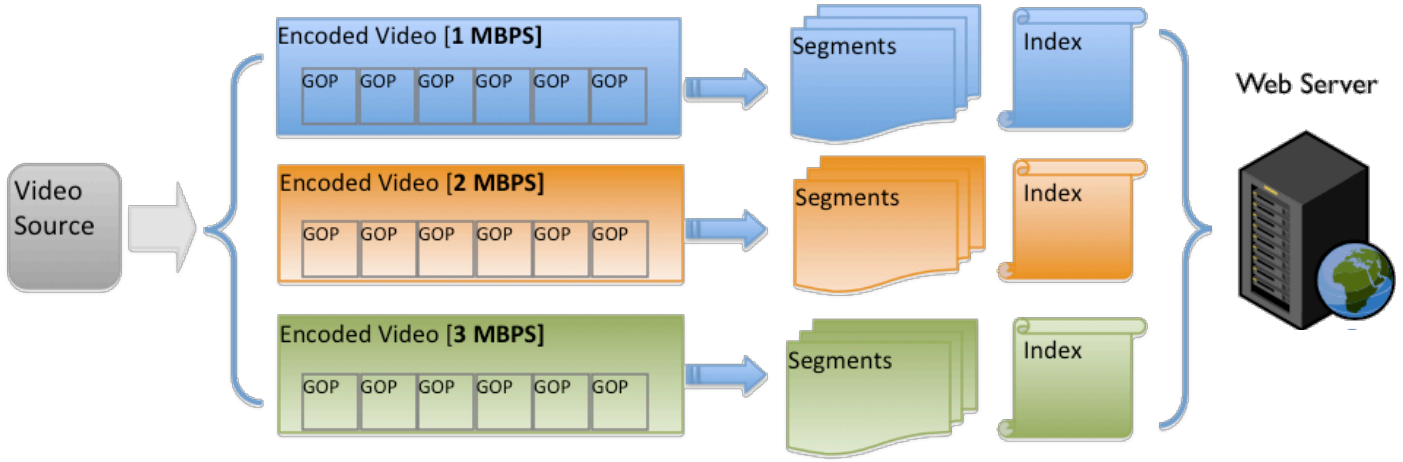
Perspective

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

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

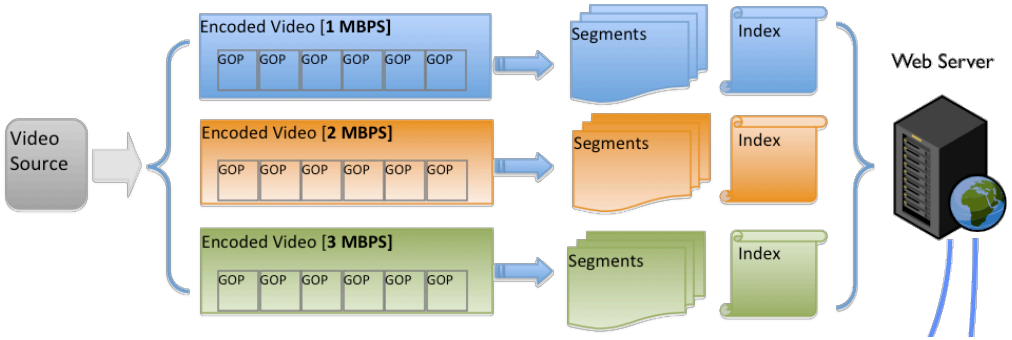
Apple HLS Authoring Specification
http://bit.ly/hls_spec_2017

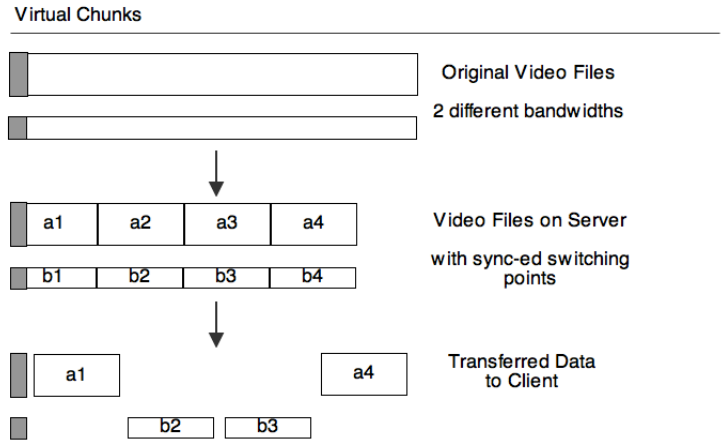
Encoding and Packaging



- 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



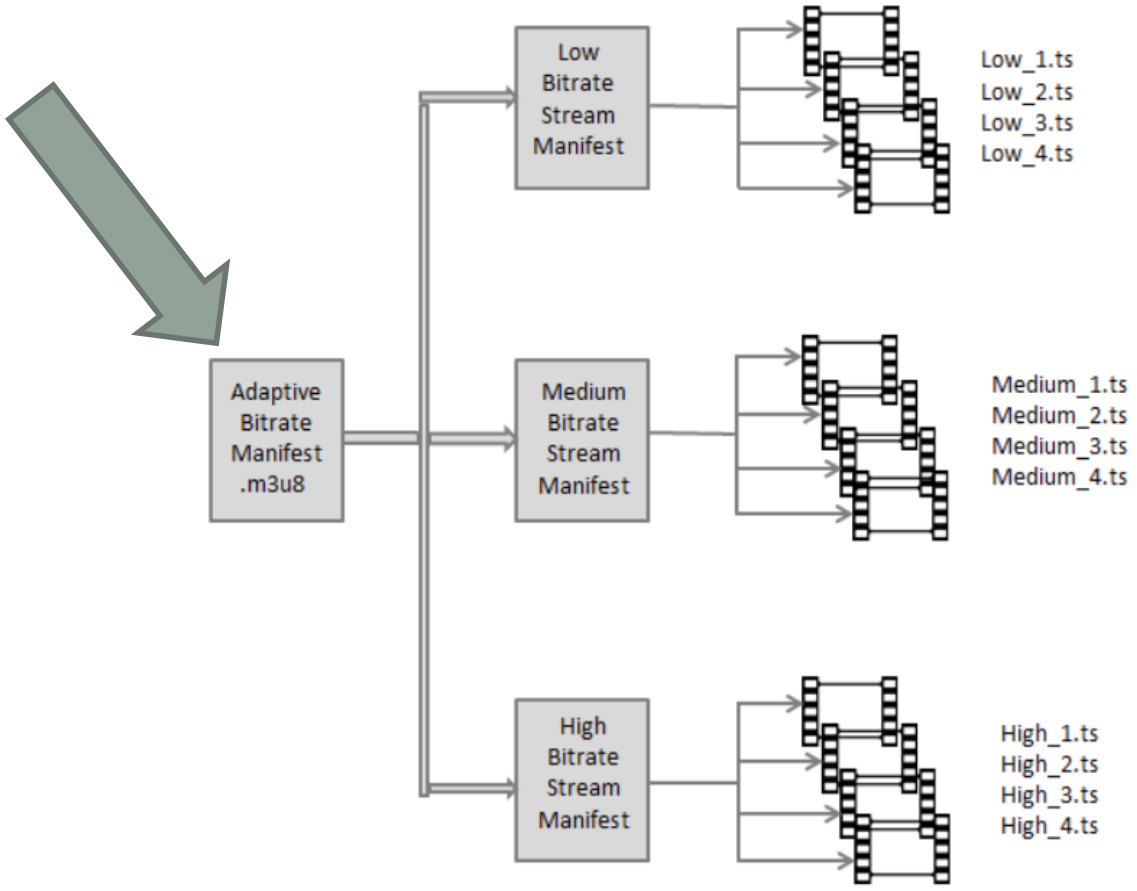


- When HTTP-based ABR started, all content files were split into multiple discrete files
  - Created administrative nightmare
  - Hundreds of thousands of files for even short videos
  - Most producers still use files for HLS

- 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

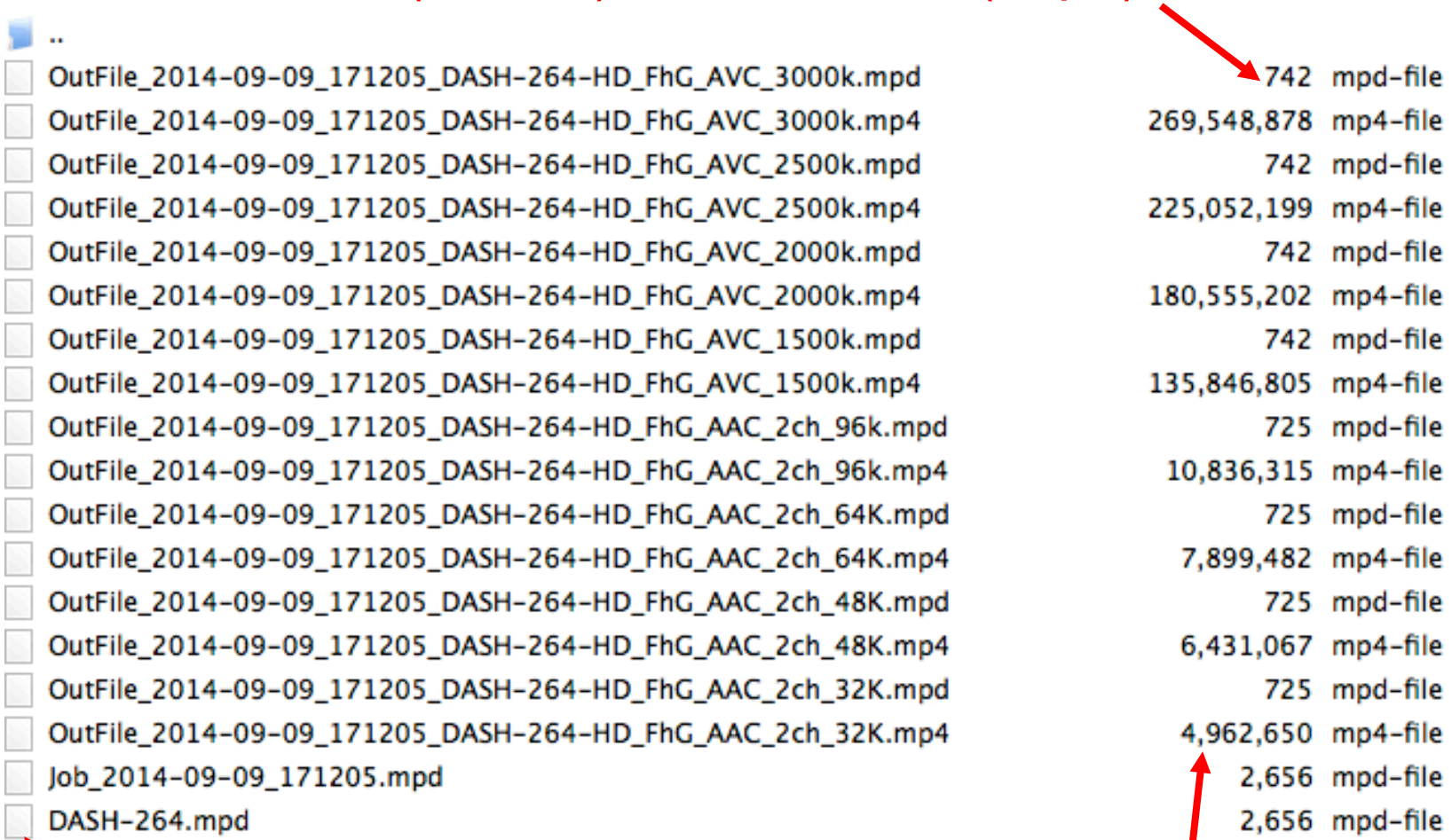
Player Side

- 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

stream (variant) manifest files (.mpd)



Content files (.mp4)

Main manifest file (.mpd)

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

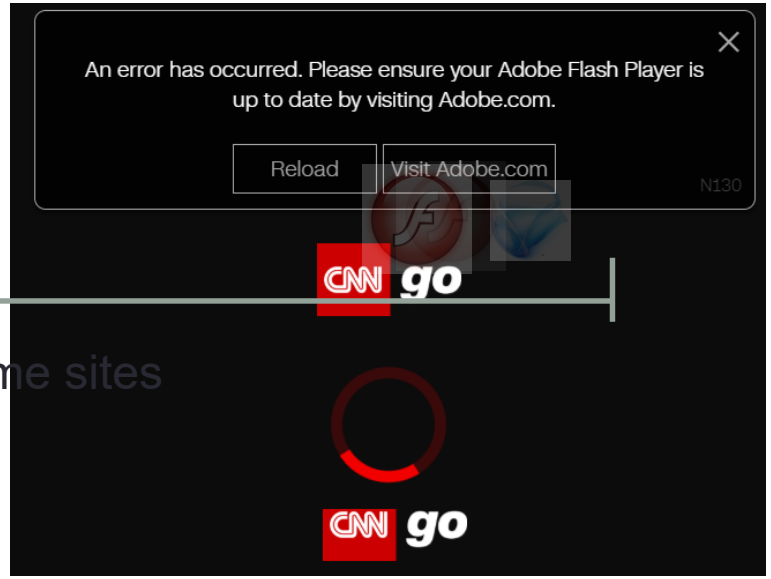
2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021

No DRM/Advertising

Mostly gone by 2016 or so

DRM/Advertising

Still in use for some applications on some sites

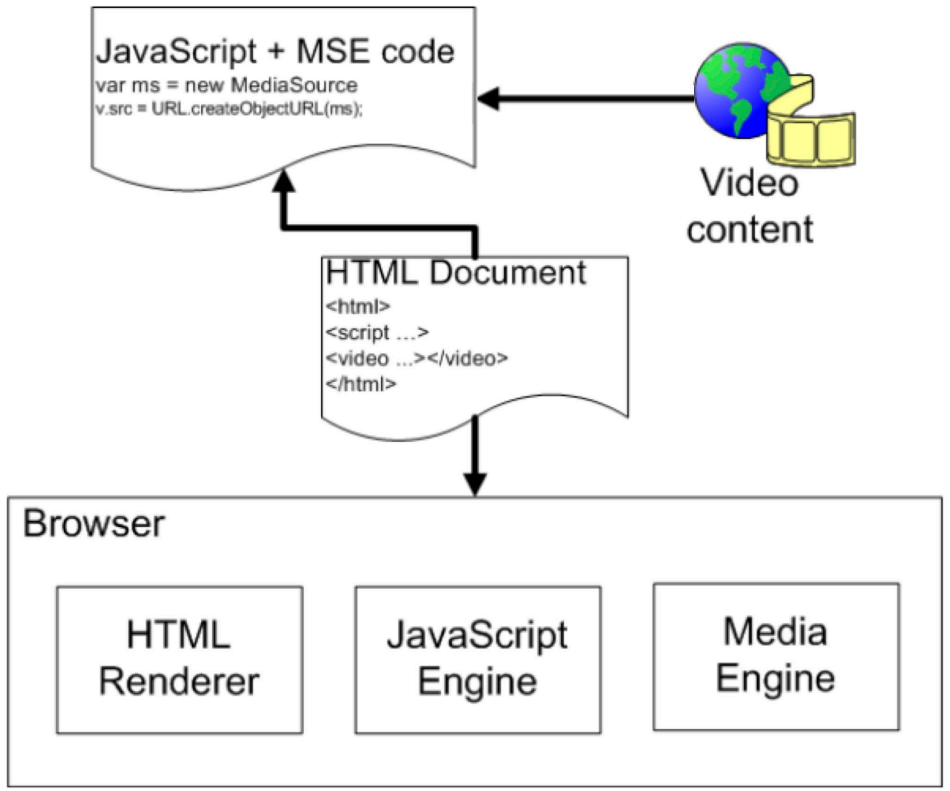


Pieces of the Puzzle

- 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
- July 2015, MPEG LA announces pool (http://bit.ly/DASH_pool_formed)
- In November 2016, MPEG LA announces license (http://bit.ly/DASH_license)

Analysis and Implications

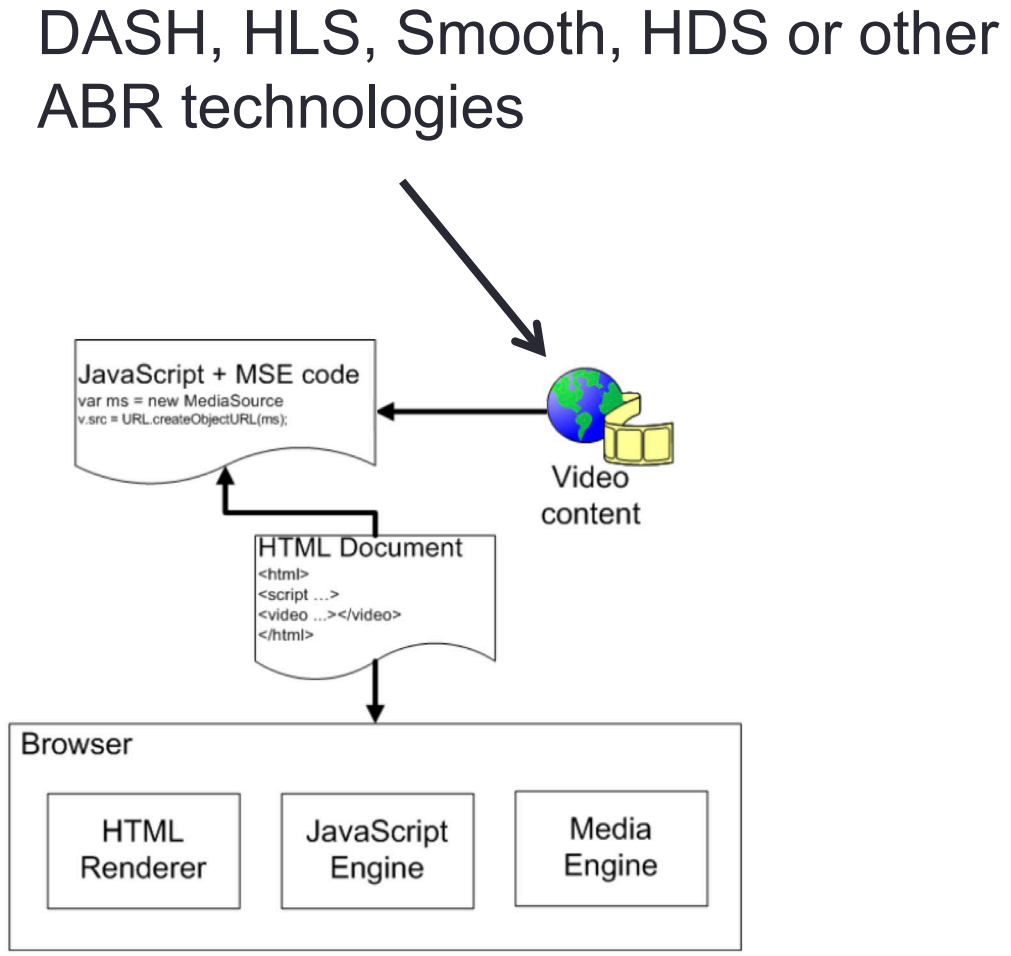
- This is the first royalty on free internet video
- CNN distributes free video in H264 or HEVC using HLS
  - No royalty
- CNN distributes free video with DASH
  - Royalty on apps and ultimately perhaps browser-based playback
- No exclusions for churches, charities, governments or otherwise
- Really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

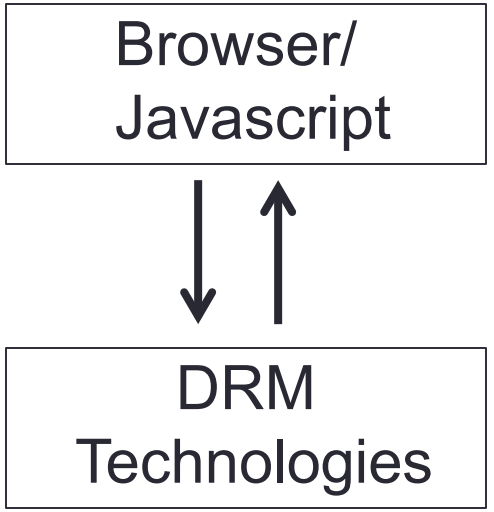
DASH and MSE

- 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	PlayReady	Widevine MODULAR	Widevine CLASSIC	FairPlay Streaming	Primetime (ACCESS)	Marlin	CMLA-OMA
Chrome (35+)	✗	✓	✗	✗	✗	✗	✗
Firefox (47+) [1] ON WINDOWS VISTA+, MAC OS X 10.9+, LINUX	✗	✓	✗	✗	✗	✗	✗
Internet Explorer (11) ON WINDOWS 8.1+	✓	✗	✗	✗	✗	✗	✗
Microsoft Edge	✓	✗	✗	✗	✗	✗	✗
Opera (31+)	✗	✓	✗	✗	✗	✗	✗
Safari SAFARI 8+ ON MACOS & SAFARI ON IOS 11.2+	✗	✗	✗	✓	✗	✗	✗

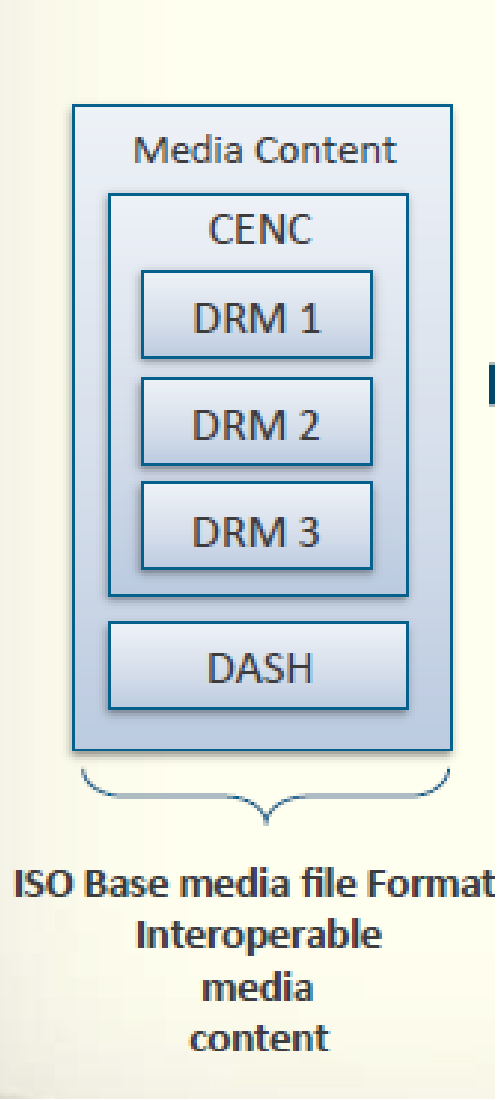
https://drmtoday.com/platforms/

- 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

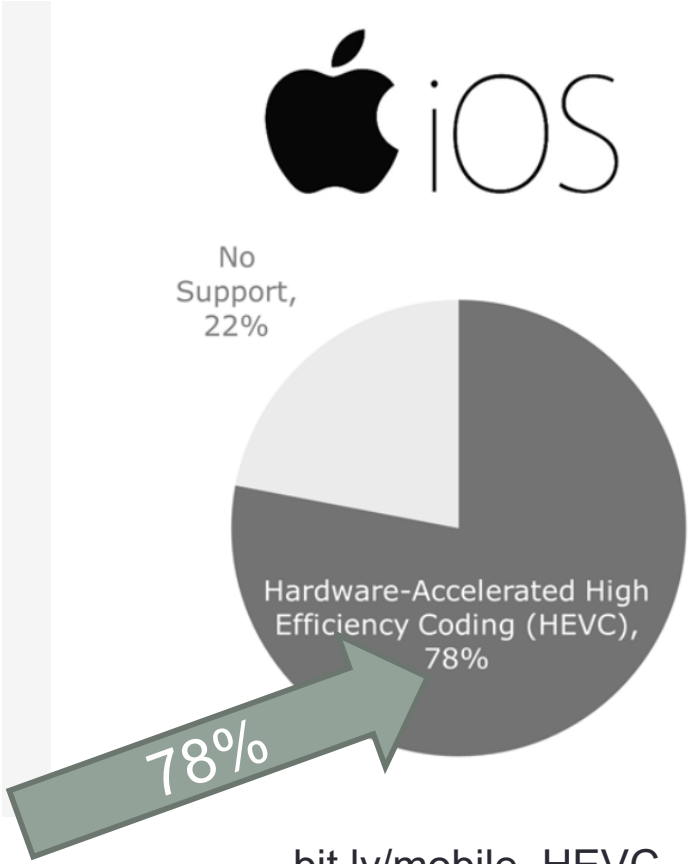
- Native support (playback in the browser)
  - HTTP Live Streaming
- Playback via an app
  - Any, including DASH, Smooth, HDS or RTMP Dynamic Streaming

iOS Media Support

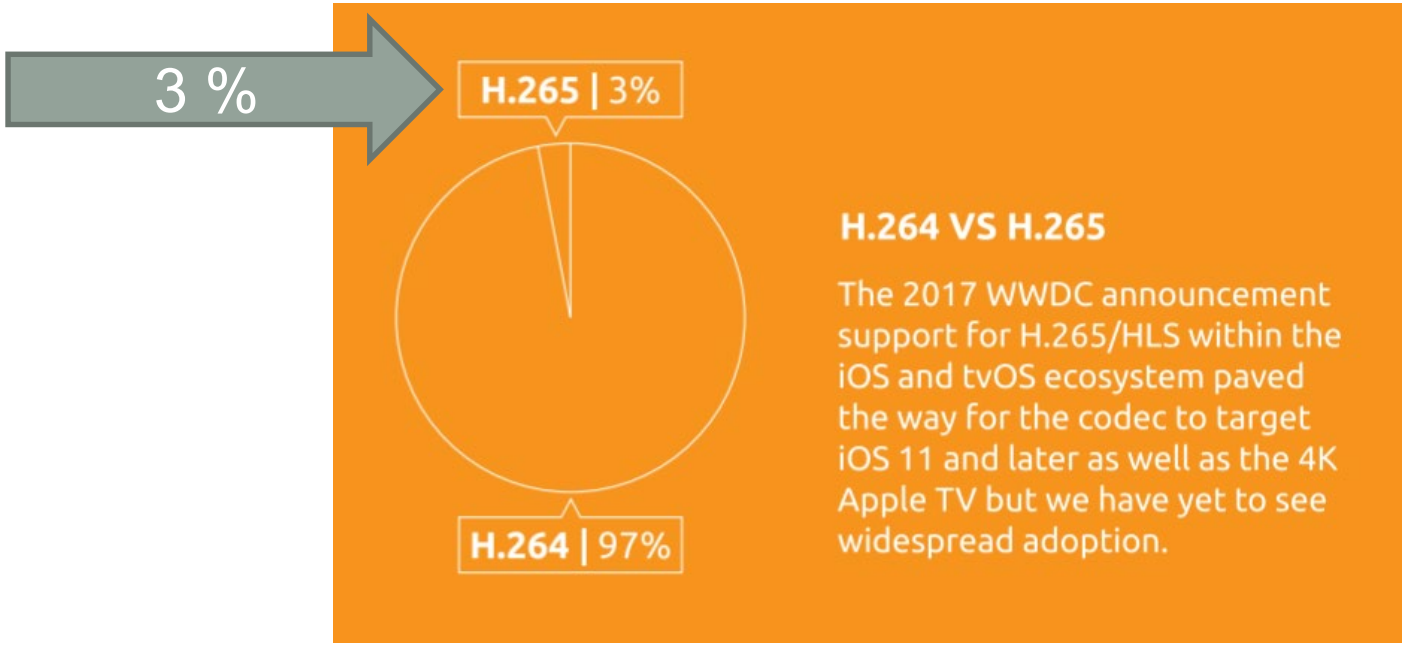
	Native	App
Codecs	H.264 (High, Level 4.2), HEVC (Main10, Level 5 high), VP8, VP9	Any
ABR formats	HLS	Any
DRM	FairPlay	Any
Captions	CEA-608/708, Web VTT, IMSC1	Any
HDR	HDR10, Dolby Vision	?

http://bit.ly/hls_spec_2017

HEVC Hardware Support - iOS



bit.ly/mobile_HEVC



http://bit.ly/glob_med_2019

Android: Codec and ABR Format Support

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

http://bit.ly/And_ver
(from 10/26/2018 – not updated)

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

- 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

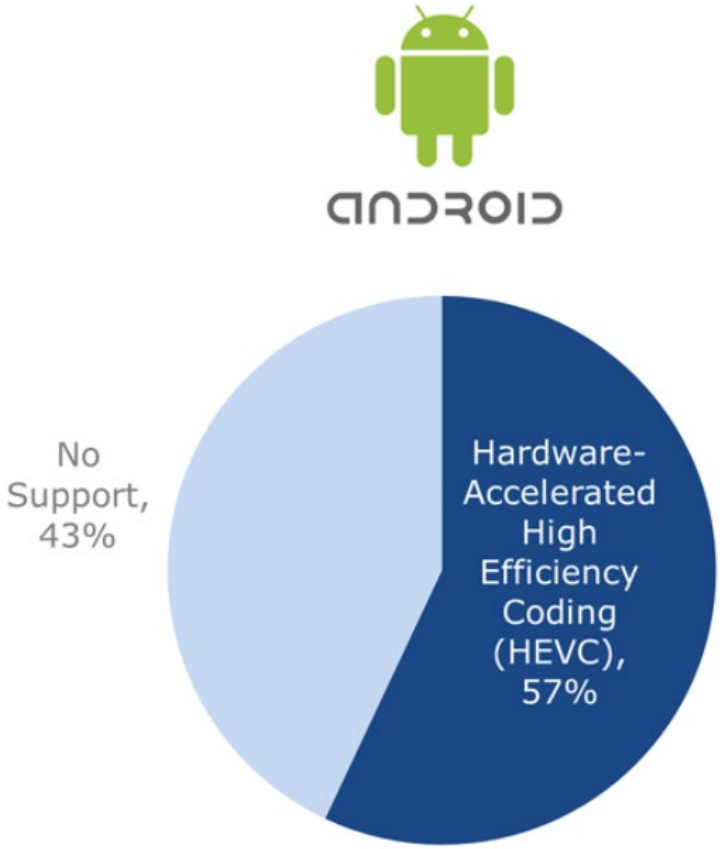
http://bit.ly/androidvideospecs

Android Media Support

	Native	App
Codecs	H.264, VP8, VP9, HEVC	Any
ABR formats	DASH, HLS	Any
DRM	Widevine	Any
Captions	Embedded 608/607 SRT	Any
HDR	Dolby-Vision, HDR10, VP9-HLG, VP9-PQ	?

https://developer.android.com/guide/topics/media/media-formats

HEVC Hardware Support - Android



- iOS playback more extensive but little penetration
- Hard to imagine I

bit.ly/mobile_HEVC

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

Source : 2018 PARKS ASSOCIATES

http://bit.ly/glob_med_2019

OTT Platform-Format Support

OTT Platforms	Smooth Streaming	HLS	DASH
Roku (bit.ly/roku_vid)	Yes	Yes	Yes
Amazon Fire TV (https://amzn.to/2L8dCdp)	Yes	Yes	Yes (?)
ChromeCast (http://bit.ly/GCast_Media)	Yes	Yes	Yes
Apple TV (bit.ly/AppleTV_recs)	No	Yes	No

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

OTT Platforms	H264	HEVC	VP9	Other
Roku (bit.ly/roku_vid)	Yes	Yes	Yes	None
Amazon Fire TVInsignia HD (https://amzn.to/2L8dCdp)	Yes	Yes	Yes	VP8, H.263, MPEG-2/4
ChromeCast Ultra (http://bit.ly/GCast_Media)	Yes	Yes	Yes	VP8, HDR10, DolbyVision
Apple TV (bit.ly/AppleTV_recs)	Yes	Yes	No	None

OTT Platform DRM Support

OTT Platforms	PlayReady	Widevine	FairPlay	Other
Roku (bit.ly/roku_vid)	Smooth/ DASH	DASH (Beta)	No	Adobe, Verimatrix, AES-128
Amazon Fire TV Insignia HD (https://amzn.to/2L8dCdp)	Yes	Yes	No	HDCP 2.2
ChromeCast (http://bit.ly/GCast_Media)	(DASH/ Smooth)	DASH/HLS	No	AES128, SAMPLE AES
Apple TV (bit.ly/AppleTV_recs)	No	No	Yes	SAMPLE-AES

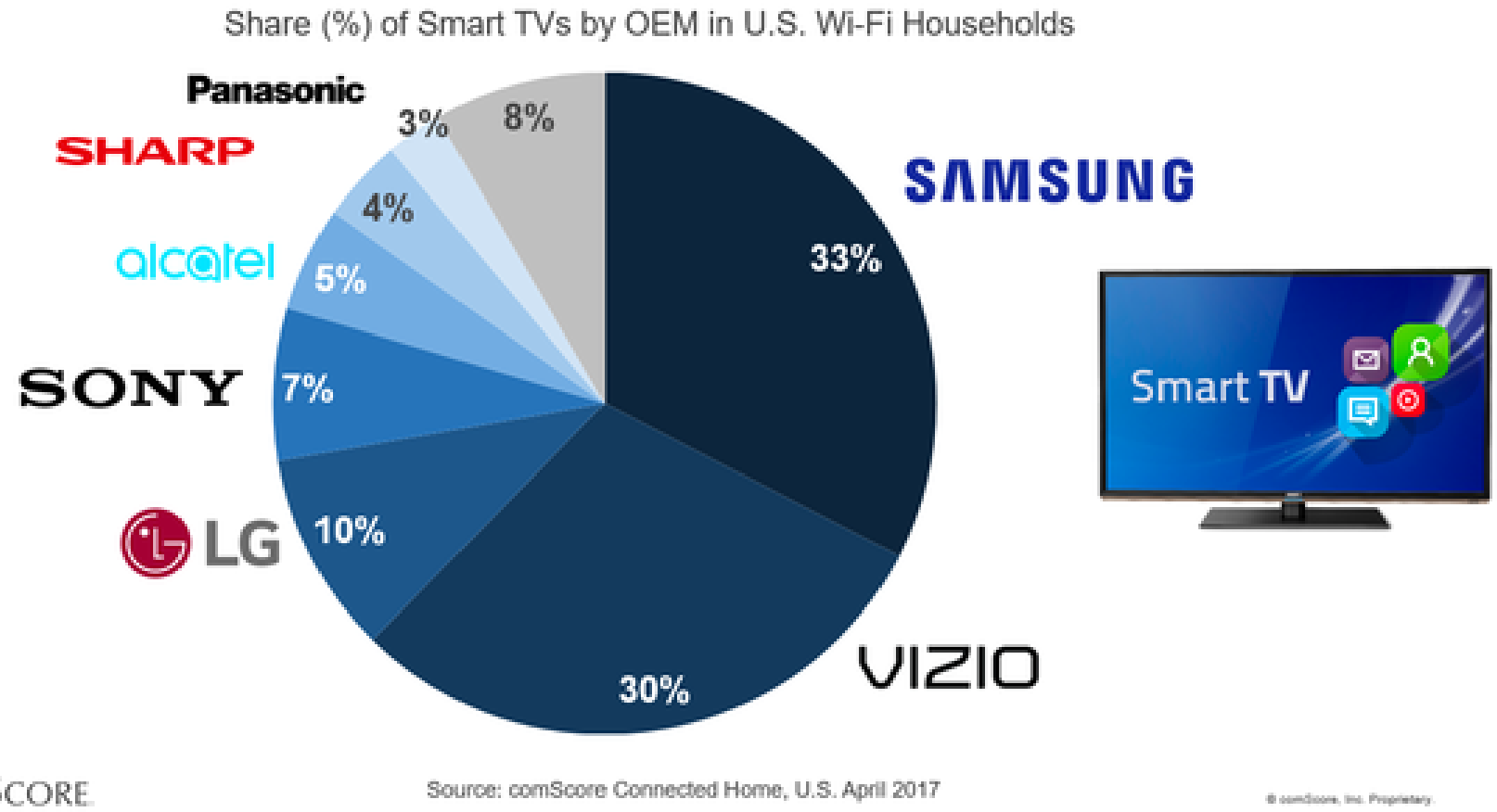
OTT Platform HDR Support

OTT Platforms	Dolby Vision	HDR 10/10+	HLG	Other
Roku (bit.ly/roku_vid)	No?	Yes/No	No	No
Amazon Fire TV Stick 4K (https://amzn.to/2L8dCdp)	Yes	Yes/Yes	Yes	No
ChromeCast (http://bit.ly/GCast_Media)	Yes	Yes/No	No	No
Apple TV (bit.ly/AppleTV_recs)	Yes	Yes	No	No

Adaptive Streaming to Smart TVs

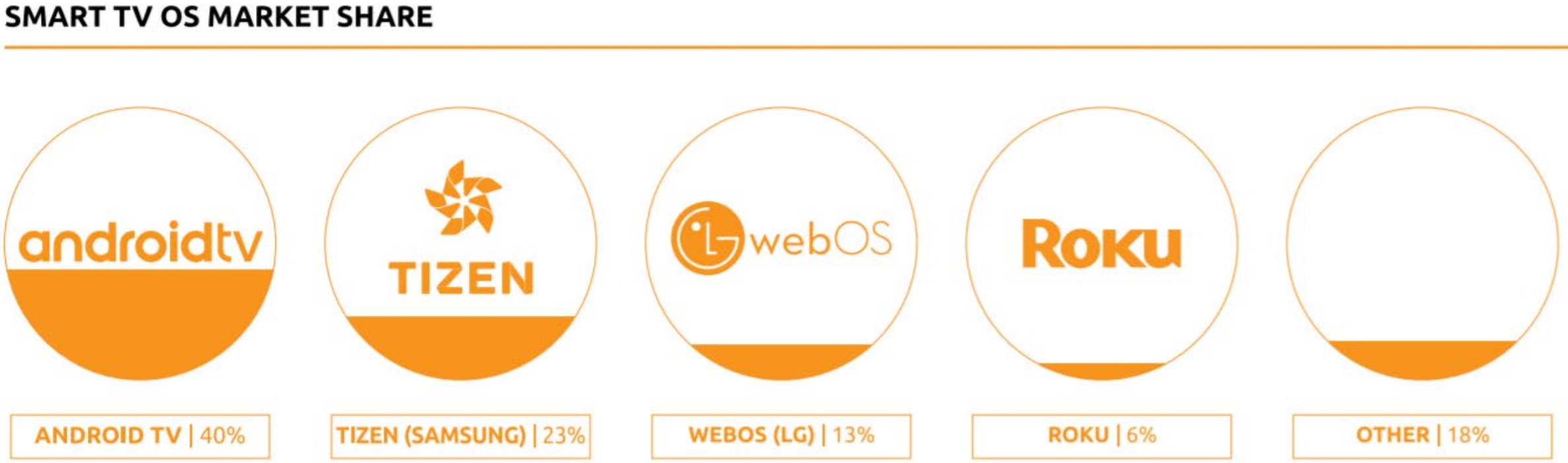
- Format support – general
- Samsung
- Vizio
- Sharp
- Panasonic
- LG
- Smart TV Alliance
- HbbTV

Who Matters – Smart TVs?



http://bit.ly/comscore_SMTV

Who Matters – Smart TV OS Market Share?



http://bit.ly/glob_med_2019

Android TV – Same as Android

	Native
Codecs	H.264, VP8, VP9, HEVC
ABR formats	DASH, HLS
DRM	Widevine
Captions	Embedded 608/607 SRT
	Dolby-Vision, HDR10, VP9-HLG, VP9-PQ

https://developer.android.com/guide/topics/media/media-formats

Samsung Format Support (Tizen)

- Very well defined - bit.ly/tizen_media

	TV 2019	TV 2018
codecs	H.264, HEVC, WMV, VP9	H.264, HEVC, WMV, VP9
ABR formats	DASH, HLS, Smooth	DASH, HLS, Smooth
DRM	Widevine, AES-128, Verimatrix WebClient	Widevine, AES-128, Verimatrix WebClient
Captions	SMI, SRT, SMPTE-TT, WebVTT, 608/708	SMI, SRT, SMPTE-TT, WebVTT, 608/708
HDR		

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

Function	Detail	A/V content
General	HTTP 1.1 with Range request	M
	HTTPS streaming over SSL	M
Adaptive	HTTP Live Streaming	M
	Microsoft Smooth Streaming	M
	MPEG-DASH (ISOBMFF & CENC) according to HbbTV version 1.2.1 profile [26]	M

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.

bit.ly/HbbTV_201

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

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

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

A Brief History of Encoding 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

Clients			Dimensions for 16:9 aspect ratio	Dimensions for 4:3 aspect ratio	Frame rate	Video bit rate (average)	Video bit rate (peak)	Audio bit rate	Total bit rate
	CELL		416 x 234	400 x 300	12	145	200	64	264
	CELL	ATV	480 x 270	480 x 360	15	365	400	64	464
WiFi	CELL	ATV	640 x 360	640 x 480	29.97	730	800	64	864
WiFi	CELL	ATV	768 x 432	640 x 480	29.97	1100	1200	96	1296
WiFi		ATV	960 x 540	960 x 720	29.97 or source	2000	2200	96	2296
WiFi		ATV	1280 x 720	960 x 720	29.97 or source	3000	3300	96	3396
WiFi		ATV	1280 x 720 or source	1280 x 960 or source	29.97 or source	4500	5000	128	5128
WiFi		ATV	1280 x 720 or source	1280 x 960 or source	29.97 or source	6000	6500	128	6628
WiFi		ATV	1920 x 1080	1920 x 1440	29.97 or source	7800	8600	128	8728

http://bit.ly/appletn2224

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

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

Apple Authoring Specification
http://bit.ly/hls_spec_2017

Adopting the Apple Spec: High End First

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

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

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

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

Scenario	Format	Frame Size	Total Bitrate	Audio Bitrate	bits/pixel *frame @ 30 fps	bits/pixel *frame @ 24 fps
Mobile & constrained (low)	baseline, mono, 10 fps	448x252	150	48	0.09	0.09
Mobile & constrained (high)	baseline, mono	448x252	450	48	0.12	0.15
Sidebar placements	main profile, stereo	384x216	400	96	0.12	0.15
Small in-page	main profile, stereo	512x288	750	96	0.15	0.18
Medium in-page	main profile, stereo	640x360	1200	96	0.16	0.20
Large in-page	main profile, stereo	768x432	1700	96	0.16	0.20
Full size in-page	main profile, stereo	960x540	2200	96	0.14	0.17
HD 720p (full screen)	high profile, stereo	1280x720	3500	96	0.12	0.15

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

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

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

	Width	Height	Data Rate	% Jump	FPS
234p	416	234	145		15
270p	480	270	365	2.52	15
360p	640	360	730	2.00	30
432p	768	432	1100	1.51	30
540p	960	540	2000	1.82	30
720p	1280	720	3000	1.50	30
1080p_l	1920	1080	4500	1.50	30
1080p_m	1920	1080	6000	1.33	30
1080p_h	1920	1080	7800	1.30	30
1440p	2560	1440	8100	1.04	30
2160p_low	3840	2160	11600	1.43	30
2160p_high	3840	2160	16800	1.45	30

HEVC/VP9/AV1

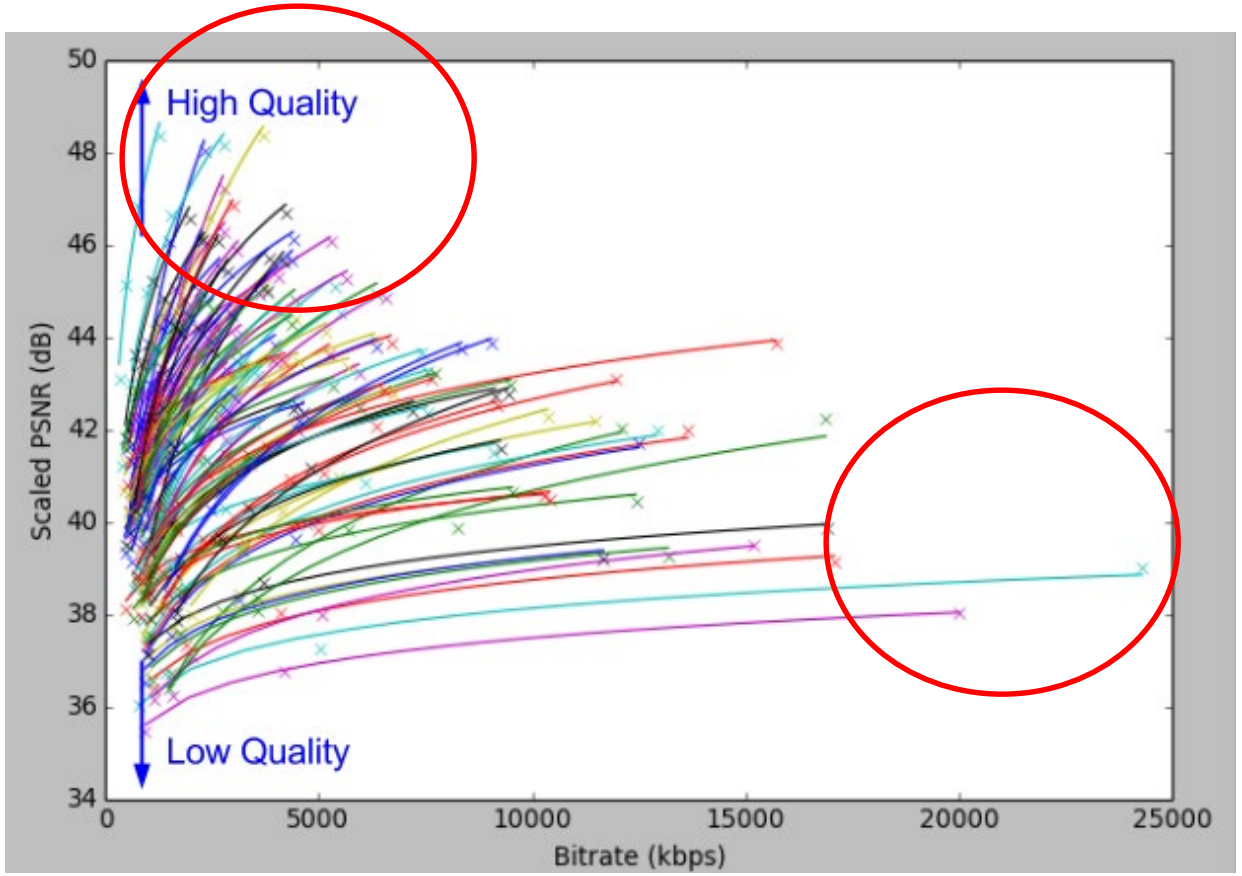
- Apple has a separate ladder for HEVC
  - Increases resolution for lowest data rates

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

Apple Authoring Specification
http://bit.ly/hls_spec_2017

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

Before | After

Resolutions	Default bitrate ladder	Per-title bitrate ladder
320x240	235	150
384x288	375	200
512x384	560	290
512x384	750	
640x480	1050	
720x480	1750	440
720x480		590
1280x720	2350	830
1280x720	3000	1150
1920x1080	4300	1470
1920x1080	5800	2150
1920x1080		3840

Pros and Cons of Per-Title

Pros

- Reduced bandwidth and storage for easy to encode clips
- Improved QoE
  - Instead of 720p stream, get 1080p stream
- Improved quality (for hard to encode clips)

Cons

- Cost
- Encoding time
- Complexity
- But
  - Easier and cheaper than deploying a new codec (uses same player)
  - Delivers many of the same benefits

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

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

*Includes only two of three known royalty groups

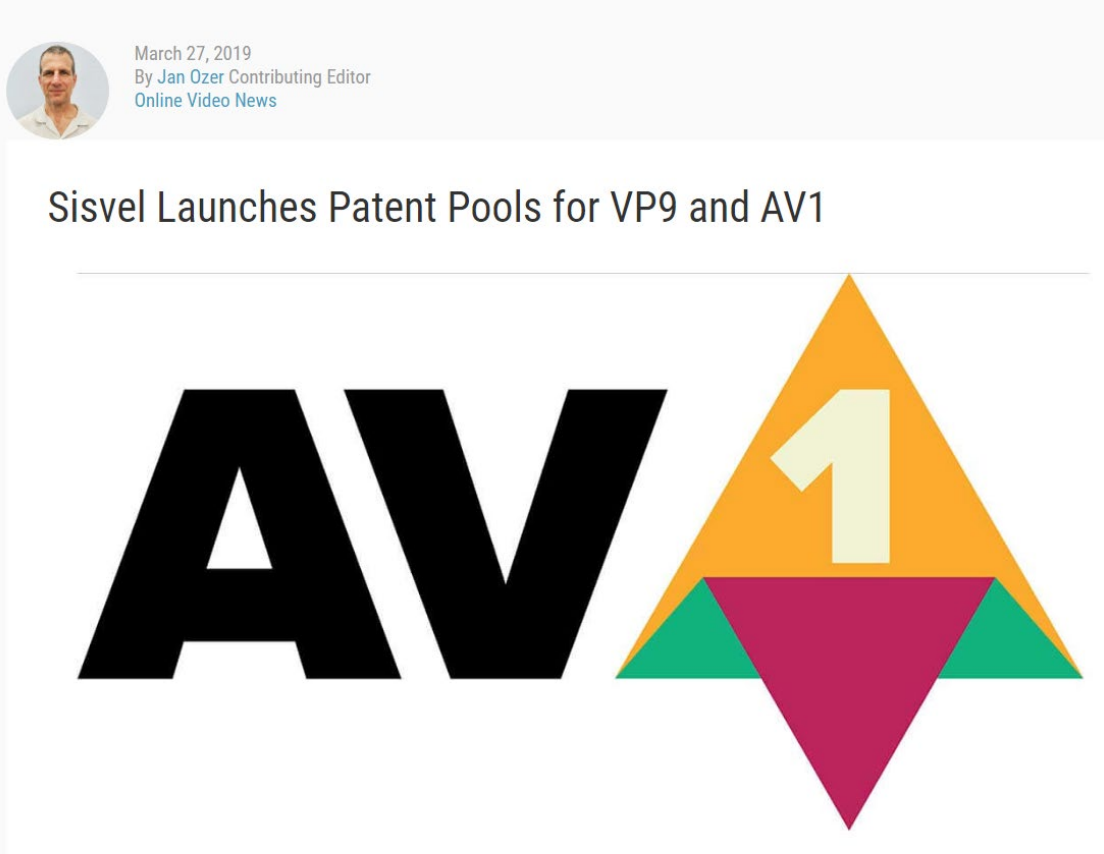
Choosing a Codec – First it Must Play

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

	Computer/ Notebook	iOS	Android	Retail OTT (Roku, Apple TV)	Smart TV
H.264	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**
HEVC	**MacOS/Windows 10 with h/w and Edge**	**Current to level 5**	**Version 5+ to 540p**	**Most**	**All 4K**
VP9	**Chrome, Firefox, Opera, Edge**	**No**	**Version 4**	**Most (not Apple TV)**	**Most Newer**
AV1	**Will have soon**	**2020**	**2020**	**2020**	**2020**

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

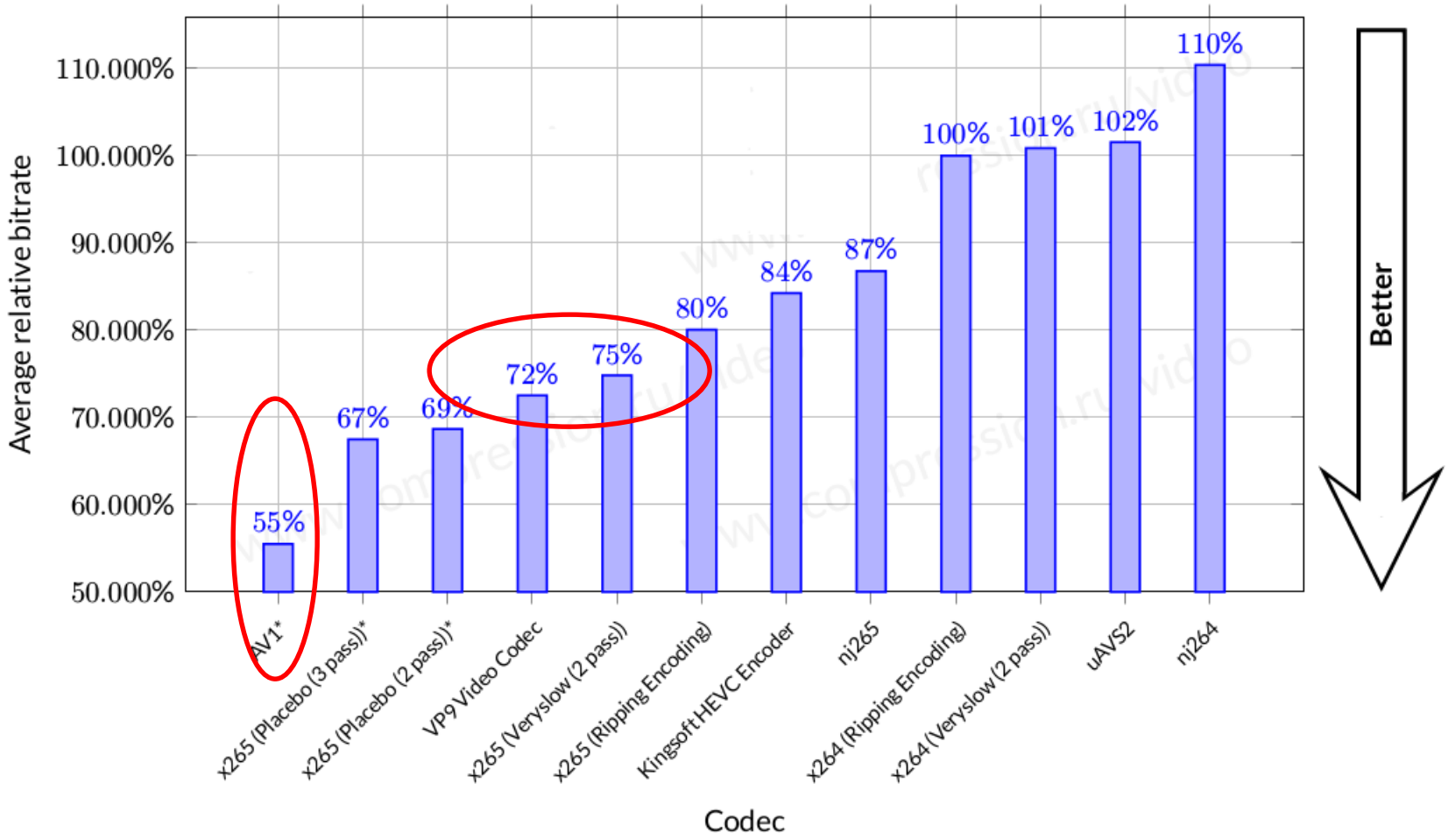


Figure 13: Average bitrate ratio for a fixed quality—use case "Ripping Encoding," all sequences, YUV-SSIM metric.

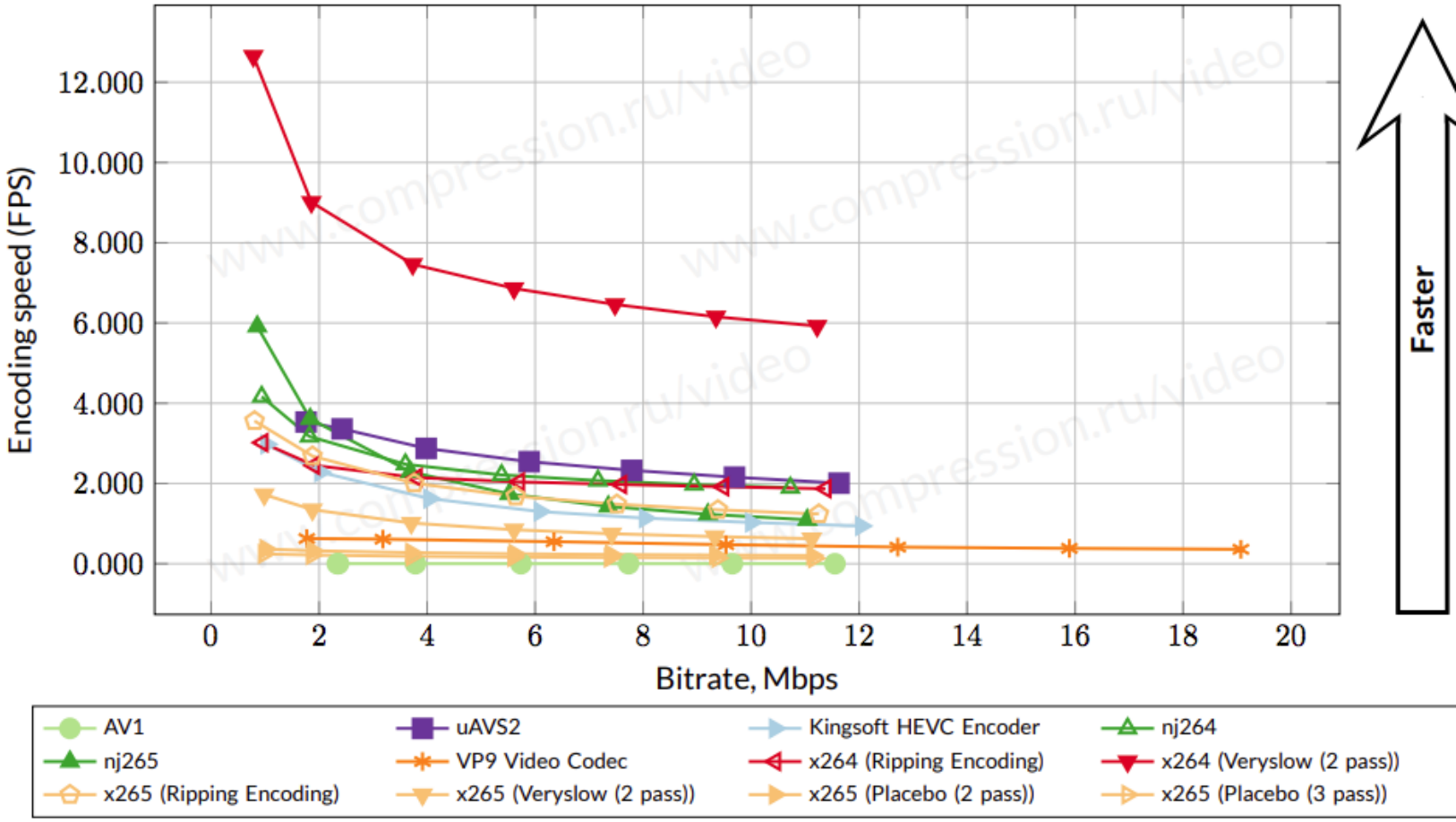
http://bit.ly/msu_hevc_2017

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 e
coding speed comparing to other encoders. AV1 encoder has extremely low speed – 2500-3000 times lower th
competitors. X265 Placebo presets (2 and 3 passes) have 10-15 times lower speed than the competitors.



http://bit.ly/msu_hevc_2017

My Tests

- My Tests: August 2018

Then	Encoding Time (seconds)	Times Real Time
AV1	226,080	45,216
x265	289	58
LibVPx	226	45
x264	18	4

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

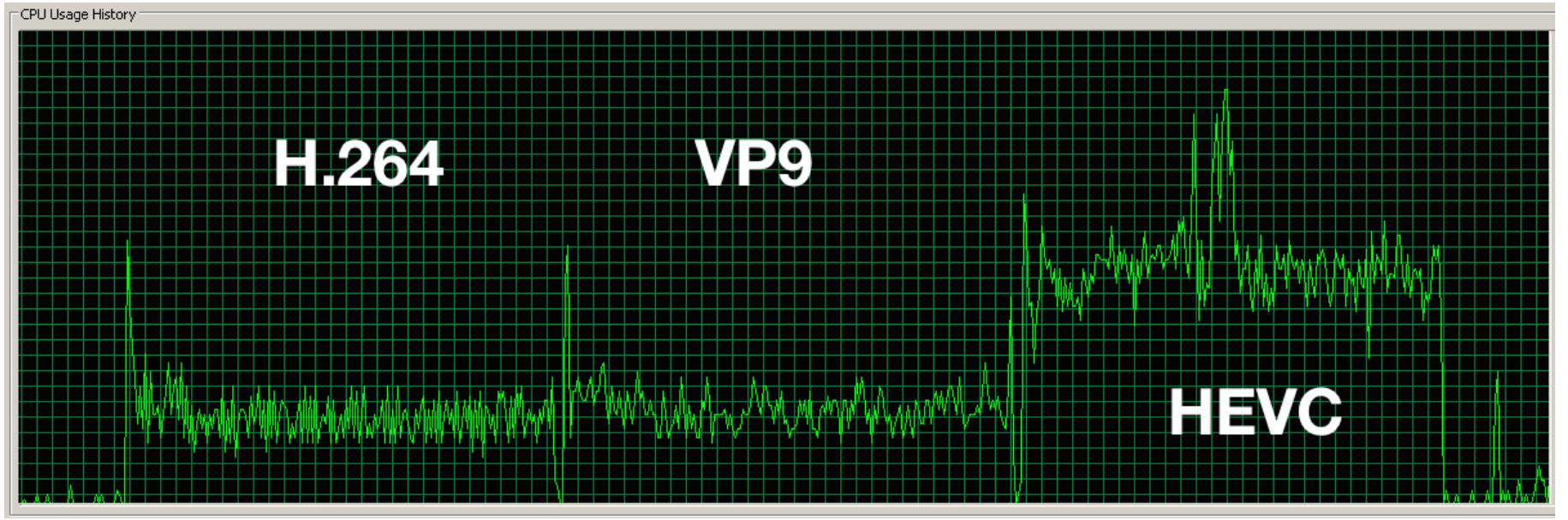
- My Tests: February 2019

	Encoding Time (Seconds)	Times Real Time	VMAF
AV1 - cpu-used 5	736	147.20	95.55
x265 - slow	38	7.60	94.83
LibVPx - speed 2	35	7.00	93.07
x264 - slow	7	1.40	92.27

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

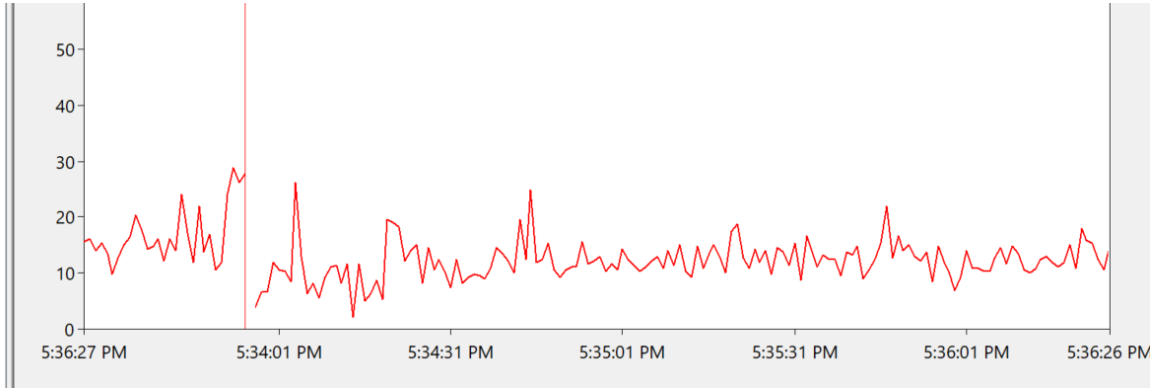
http://bit.ly/AV1_encode

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 signifcant advantage over AV1 until devices with hardware decode arrive (2020)

Decode CPU – AV1 Appears Reasonable



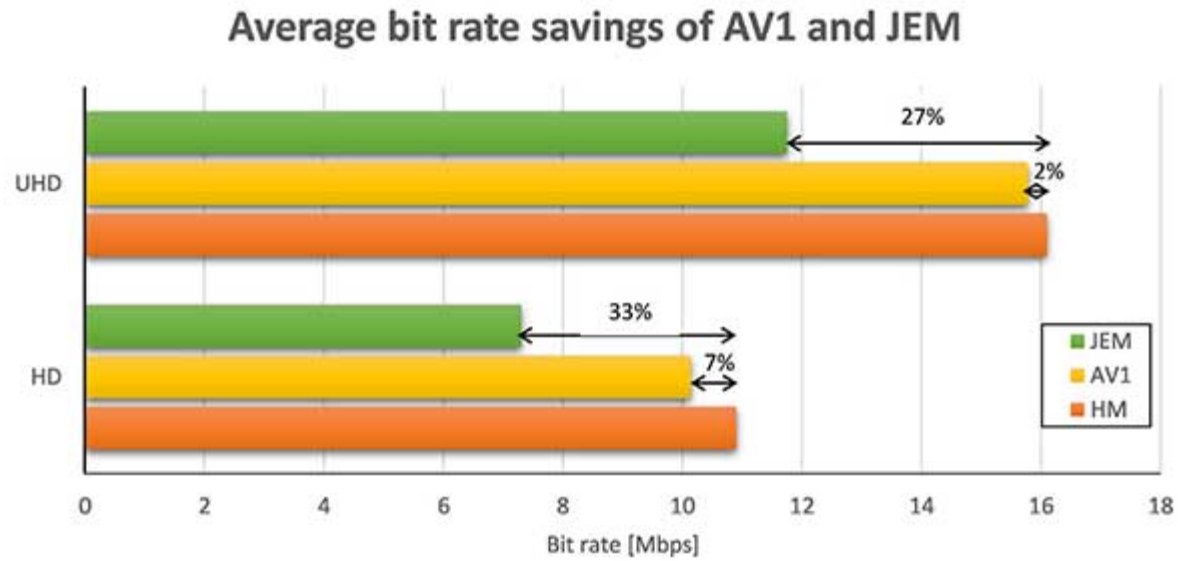
- Playback on an HP ZBook notebook (Xeon processor
- 1080p video from YouTube played back in Firefox

- 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

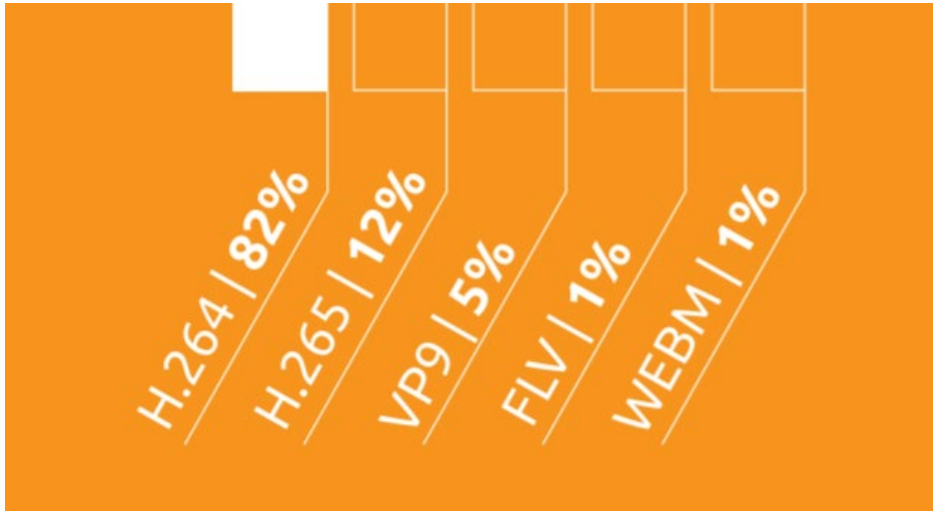


- HM = HEVC
- AV1 = AV1
- JEM = VVC (don't ask
- Chart shows data rate needed for equivalent quality
  - Shorter is better

- 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



http://bit.ly/glob_med_2019

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