HOW TO CUSTOMIZE ENCODING, PACKAGING & DRMS FOR OTT

YURIY REZNIK | YREZNIK@BRIGHTCOVE.COM

OUTLINE

- About Brightcove
- Brightcove VideoCloud architecture
- Key challenges and means of addressing them
  - Dynamic delivery
  - Quality assessment
  - Encoding
- ABR profile generation
  - Single codec case
  - Multi-codec case
- Practical examples
- Q&A

ABOUT BRIGHTCOVE

FOUNDED

2004
IPO 2012

HEADQUARTERS

BOSTON,
MASSACHUSETTS
500+ EMPLOYEES

REVENUE

$165M
IN 2018

CUSTOMERS

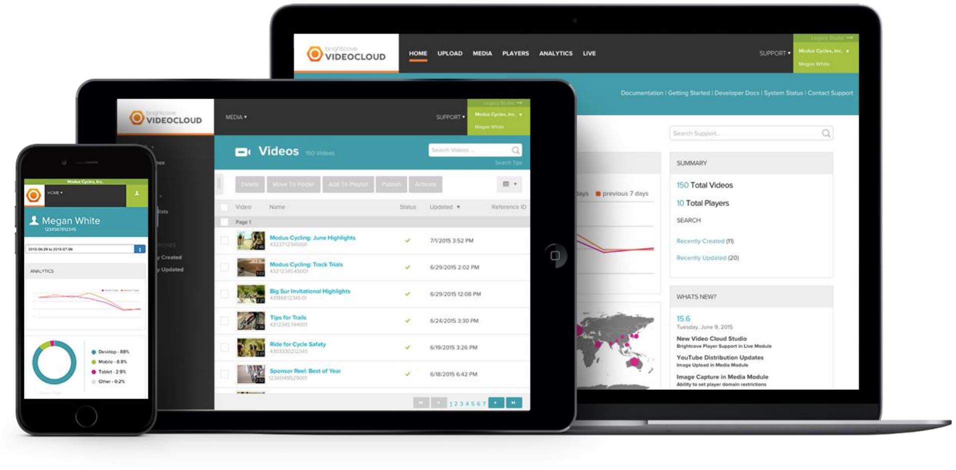
3,700+
CUSTOMERS IN MORE
THAN 70 COUNTRIES

SCALE

31B
VIDEO STREAMS
2B
HRS OF VIDEO VIEWED
125M
VIDEOS ENCODED
IN 2018

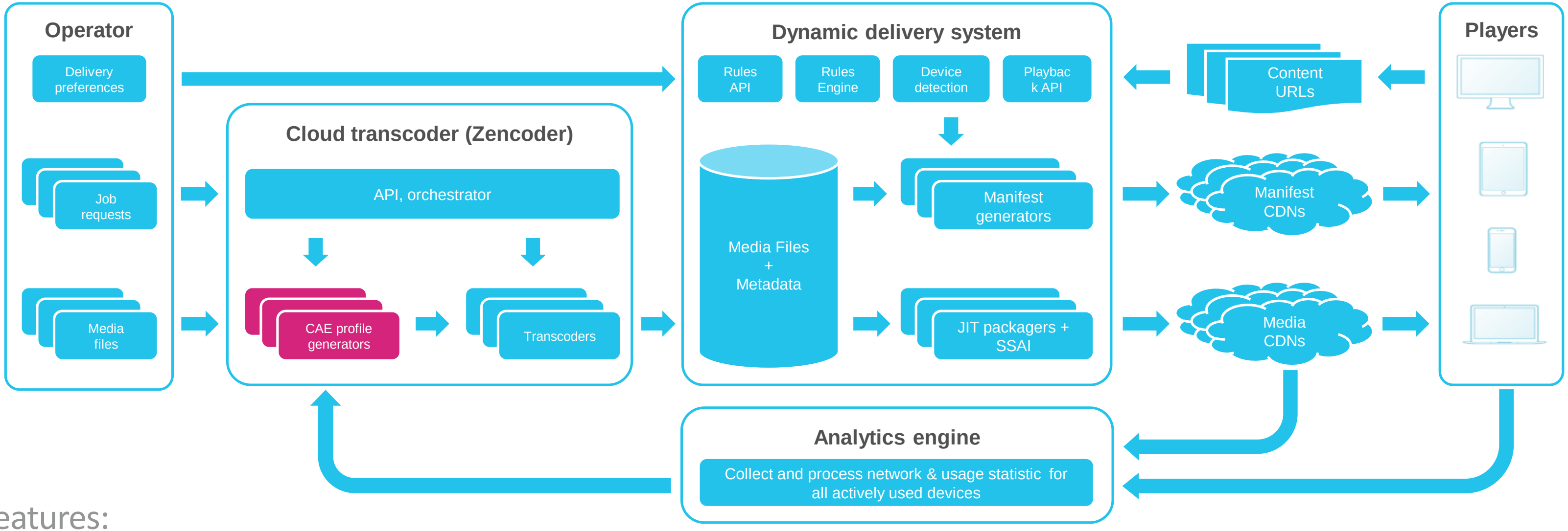
GLOBAL PRESENCE & REACH

brightcove VIDEOCLOUD



VIDEOCLOUD ARCHITECTURE

- VOD delivery chain:



- Key features:
  - **CAE** = context-aware encoding
  - **Internal format** is used for storage of transcoded streams
  - **Just-in-time manifest generation and packaging**
  - **2 layers of CDNs** for media delivery
  - Analytics engine & **closed loop** to CAE

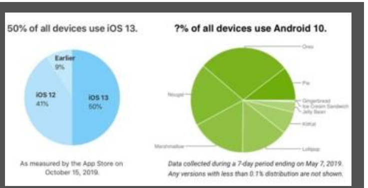
KEY PROBLEMS & MEANS OF ADDRESSING THEM

Challenge	Mean(s)	Examples/Comments
Scale	Use of CDNs, cloud, edge platforms	
Time-varying network capacity	ABR streaming with different rates	E.g. 400, 800, 1200 Kbps streams
Different screens	ABR streaming with different resolutions	E.g. 480p, 720p, 1080p streams
Multiple codecs	Multi-codec profiles, client detection, manifest filtering	E.g. selective use of HEVC
Multiple video formats	Multi-representation profiles, client detection, manifest filtering	E.g. SDR, HDR10, HLG, DV, etc.
Multiple delivery formats	Dynamic packaging & delivery	E.g. HLS for Apple devices, DASH for CTVs
Multiple DRMs	Dynamic packaging & delivery	E.g. FairPlay for Apple, Widevine for Androids
Differences in mezzanine formats	Pre-processing, quality analysis	Deinterlace, inverse telecine, etc.
Differences in content complexity	Content aware encoding	aka Per-title-encoding
Differences in usage patterns	Context aware encoding	Pioneered by Brightcove CAE
Differences in network statistics	Context aware encoding	Pioneered by Brightcove CAE
End-to-end efficiency	Context aware encoding	Pioneered by Brightcove CAE

WHY DYNAMIC PACKAGING & DELIVERY IS NEEDED?

- Basically – we live in multi-DRM world
  - Apple devices require FairPlay
  - Androids need Widevine
  - Game consoles, and some TVs – need PlayReady
- Then come delivery formats
  - HLS (v3+, TS-based) – needed by old Apple devices
  - DASH (IOP v1.0+) – needed by smart TVs
  - HLS (v23+, MP4-based) – can finally support CMAF
- And then there are codecs
  - H.264, HEVC, VP8-9, AV1, etc.
  - Support for which also varies among devices & OSes

Fragmentation of iOS & Android devices:



Source: https://venturebeat.com/2019/10/17/as-ios-13-hits-50-adoption-android-fragmentation-keeps-getting-worse/

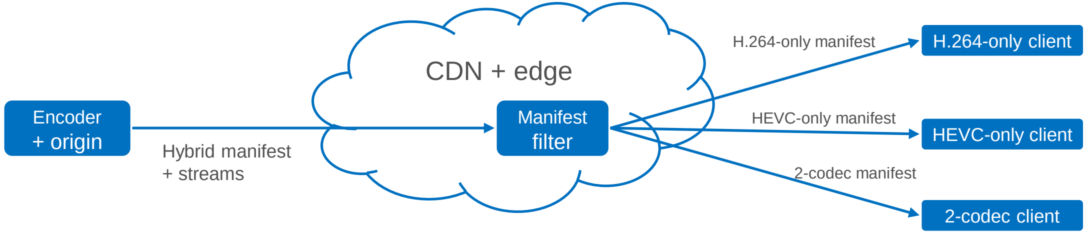
Platform / DRM support matrix:

Media players		PlayReady	Widevine Modular	Widevine Classic	FairPlay	Primetime	Marlin	CMLA-OMA
Browsers	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+	✗	✗	✗	✓	✗	✗	✗
Mobile	Android (6+) [2]	✗	✓	✗	✗	✗	✗	✗
	Android (4.4 – 5.1)	✗	✓	✓	✗	✗	✗	✗
	Android (3.1 – 4.3)	✗	✗	✓	✗	✗	✗	✗
	iOS (6+)	✗	✗	✗	✓	✗	✗	✗
	Windows Phone	✓	✗	✗	✗	✗	✗	✗
Set-top-boxes	Chromecast	✓	✓	✗	✗	✗	✗	✗
	Android TV	✓	✓	✗	✗	✗	✗	✗
	Roku	✓	✓	✗	✗	✓	✗	✗
	Apple TV	✗	✗	✗	✓	✗	✗	✗
	Amazon Fire TV	✓	✗	✗	✗	✗	✗	✗
	Google TV	✓	✗	✓	✗	✗	✗	✗
Smart TVs	Samsung (Tizen) 2017-2018+ MODELS	✓	✓	✗	✗	✗	✗	✗
	Samsung (Tizen) 2015-2017 MODELS	✓	✗	✓	✗	✗	✗	✗
	Samsung (Orsay) [3] 2010-2015 MODELS	✓	✗	✓	✗	✗	✗	✗
	LG (webOS & Netcast)	✓	✗	✓	✗	✗	✗	✗
	Smart TV Alliance [4] LG, PHILIPS, TOSHIBA, PANASONIC	✓	✗	✓	✗	✗	✗	✗
	Android TV	✓	✓	✗	✗	✗	✗	✗
GCs	Xbox One / 360	✓	✗	✗	✗	✗	✗	✗
	PlayStation 3 / 4	✓	✗	✗	✗	✗	✓	✗
Specs	TNT (2.0+)	✓	✗	✗	✗	✗	✓	✗
	HbbTV (1.5+)	✓	✓	✗	✗	✓	✓	✓

Source: https://castlabs.com/resources/drm-comparison/

WHY MANIFEST FILTERING IS NEEDED?

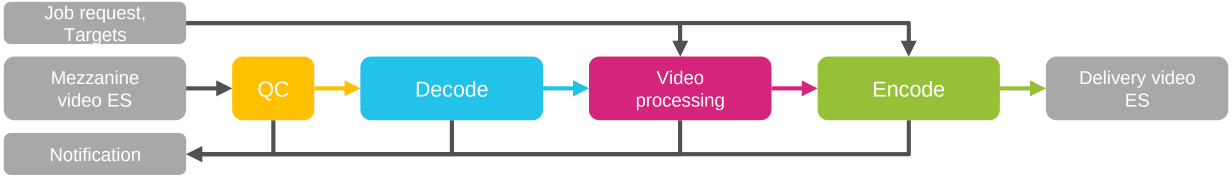
- To ensure we send streams that devices can decode!
- Example: delivery of HEVC streams:



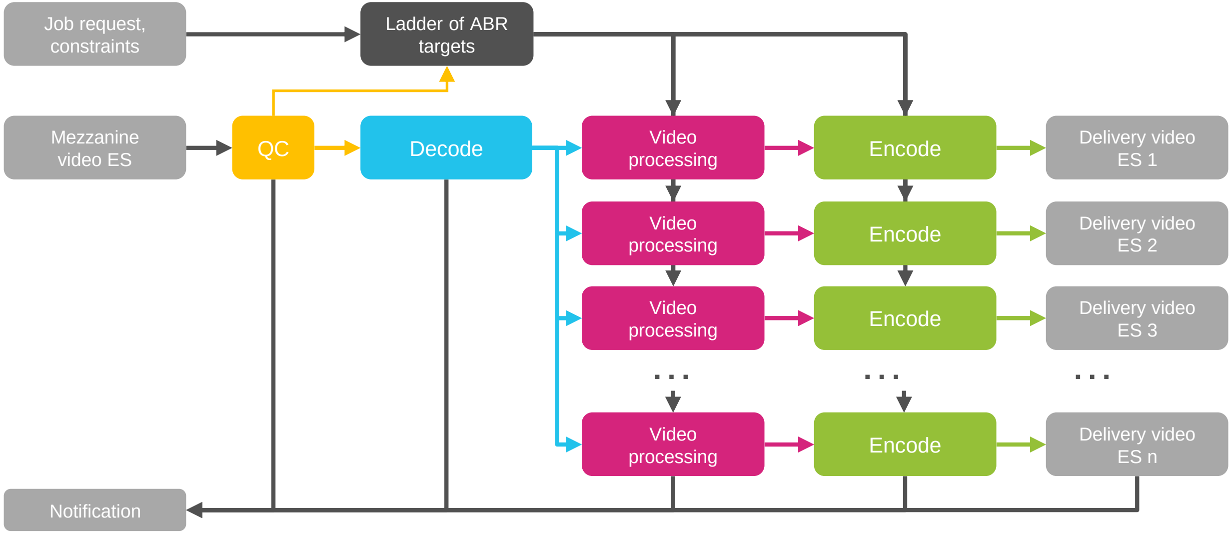
- Device detection:
  - Determines type of client requesting access to manifest
- Manifest filtering logic:
  - removes HEVC streams from manifests if requesting device is a legacy, non-HEVC capable device
  - removes H.264 streams from manifests heading to HEVC capable non-switchable devices
  - leaves both HEVC and H.264 streams if devices are capable of decoding both codecs and switching between them
- 2-codec manifests:
  - HLS: mixed variant streams (ordered by bitrate)
  - DASH: separate adaptation sets for HEVC and H.264 + supplemental properties declaring them as switchable

TRANSCODING

- Single rate transcoding



- ABR transcoding:

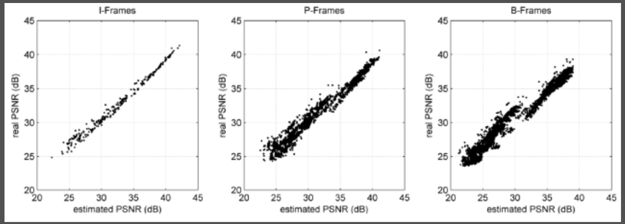


- Key operations:
  - QC – quality checks
  - Ladder design = CAE
  - Pre-processing
  - Efficient encoding
  - Conformance checks

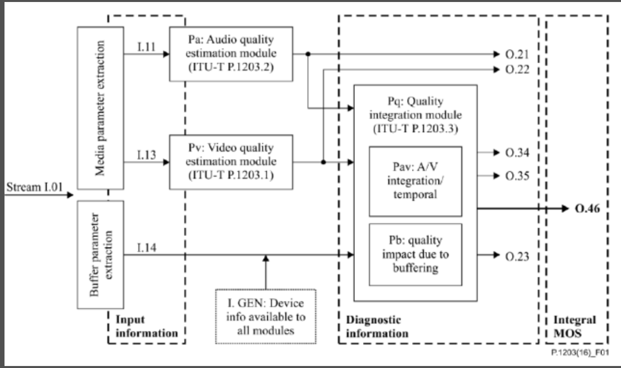
QUALITY CHECKS

- In many cases we just receive mezzanine and have to figure out what quality of content it has
- There are several techniques available:
  - Prediction of PSNR based on QPs and distributions of TCoeffs
    - Eden 2007, Elecard ePSNR, etc.
  - **Parametric metrics:**
    - ITU-T P.1203.1
      - recent standard, accounts for many effects such as upscale factor, temporal quality variations, etc.
  - **Non-reference metrics:**
    - BRISQUE (Mittal el at, 2012)
    - BLIINDS (Saad et al, 2010)
    - STAIND (Chu et al, 2012)
    - etc.
- None of them is perfect, but they certainly can be used to spot obvious problems and improve robustness of the product

Accuracy of PSNR estimation (Eden 2007):

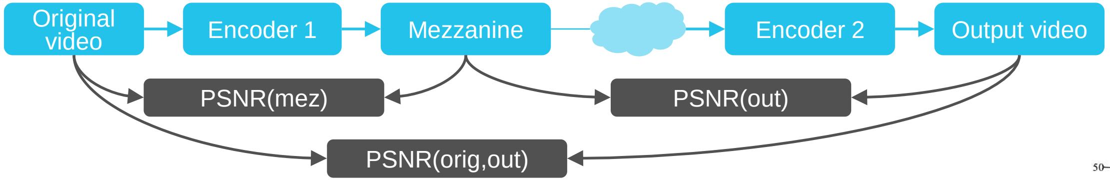


Block diagram of ITU-T P.1203:



RELATIONSHIP BETWEEN QUALITY OF OUTPUT AND MEZZANINE

- Let's take a look at processing chain & 3 measurements:



- By using simple math we can show that:

$$PSNR(orig, out) \leq 10\log_{10}\left(\frac{255^2}{10^{\frac{PSNR(mez)}{10}}} + \frac{255^2}{10^{\frac{PSNR(out)}{10}}}\right)$$

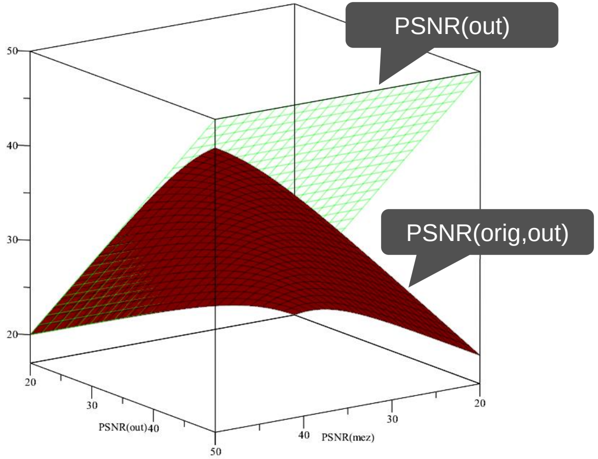
- It we further assume that 2nd encoder is adding at least as much of noise as 1st:

$$PSNR(out) \leq PSNR(mez)$$

then it follows that

$$PSNR(orig, out) \leq PSNR(out) - 10\log_{10}(2)$$

$$\sim PSNR(out) - 3.01\text{dB}$$



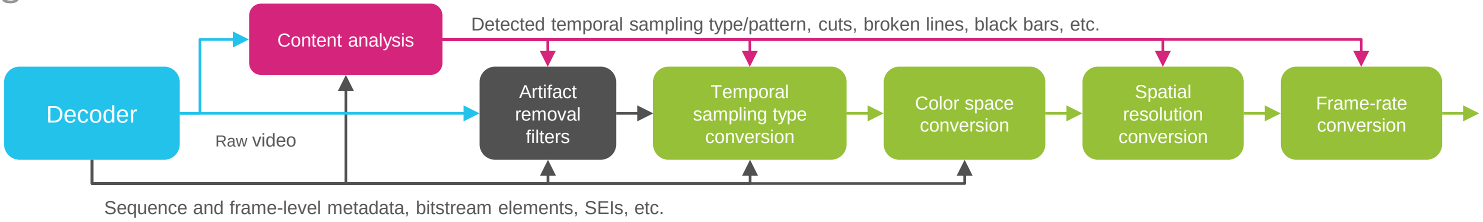
- In other words, if you must to ensure certain end-to-end quality, you **must have at least 3dB extra** at mezzanine encoding level! This has to be checked first.

PRE-PROCESSING

Key functions:

- **understand** what type of content it is (which may be different from the way it was previously encoded)
    - progressive / interlace / telecine, cadence type, field order, etc.
- **minimize artifacts** introduced by prior generation encoder or sampling process
    - blocking, ringing, broken lines, temporal noise, etc.
- **perform conversion** from source format to format needed for delivery. This includes conversions of:
    - spatial resolution
    - chroma sampling type (4:4:4, 4:2:2, 4:2:0, different kinds of 4:2:0)
    - frame rate
    - temporal sampling type (progressive, telecine, interlace, field order)
    - color (gamma, matrix, primaries, EOTF, mastering display color volume, other display-related metadata, etc.).

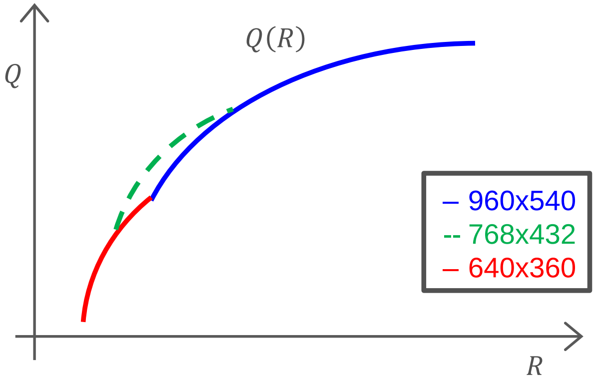
Processing chain:



ON CHOICES OF RESOLUTIONS

In theory, the more resolutions are available the better:

- allows better quality / rate tradeoffs:



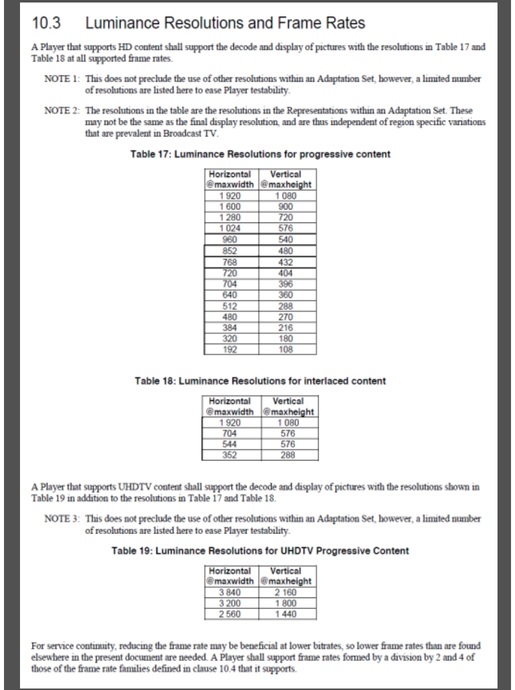
In practice, the choices are also constrained by

- content owners
- industry forum guidelines: DVB, HBBTV, DTV, etc.
- capabilities of playback devices

But equally important is also to look at

- **actual distributions of resolutions of players as the play the content!**
- the closer they can be matched the better

DVB recommended resolutions



10.3 Luminance Resolutions and Frame Rates

A Player that supports HD content shall support the decode and display of pictures with the resolutions in Table 17 and Table 18 at all supported frame rates.

NOTE 1: This does not preclude the use of other resolutions within an Adaptation Set, however, a limited number of resolutions are listed here to ease Player testability.

NOTE 2: The resolutions in the table are the resolutions in the Representations within an Adaptation Set. These may not be the same as the final display resolution, and are thus independent of region specific variations that are prevalent in Broadcast TV.

Table 17: Luminance Resolutions for progressive content

Horizontal @maxwidth	Vertical @maxheight
1 920	1 080
1 600	900
1 280	720
1 024	576
960	540
852	480
768	432
720	404
704	396
640	360
512	288
480	270
384	216
320	180
192	108

Table 18: Luminance Resolutions for interlaced content

Horizontal @maxwidth	Vertical @maxheight
1 920	1 080
704	576
544	576
352	288

A Player that supports UHDTV content shall support the decode and display of pictures with the resolutions shown in Table 19 in addition to the resolutions in Table 17 and Table 18.

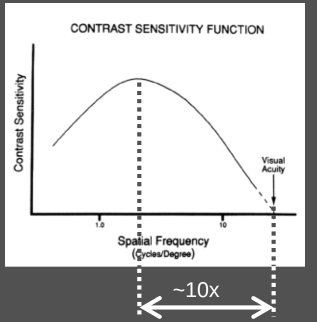
NOTE 3: This does not preclude the use of other resolutions within an Adaptation Set, however, a limited number of resolutions are listed here to ease Player testability.

Table 19: Luminance Resolutions for UHDTV Progressive Content

Horizontal @maxwidth	Vertical @maxheight
3 840	2 160
3 200	1 800
2 560	1 440

For service continuity, reducing the frame rate may be beneficial at lower bitrates, so lower frame rates than are found elsewhere in the present document are needed. A Player shall support frame rates formed by a division by 2 and 4 of those of the frame rate families defined in clause 10.4 that it supports.

Range of resolutions



Note: 10x downscaled videos start losing details that are most prominently visible in normal reproduction setting.

This suggests that the range of resolutions that can be used for encoding should not be larger than 10.

Coincidentally, this is precisely the range supported by the DVB ladder.

ENSURING INTEROPERABILITY

VBR / HRD control

- for ABR delivery all streams must be capped VBR !!!
  - the use of highly-variable VBR encoding may confuse clients and cause buffering
- typically, maximum bitrate cap is set to about 10-35% above average bitrate
  - must be lower than next target bitrate in the ABR encoding ladder
- decoder buffer size must also be limited

GOP length and type

- GOP length must be shorter or equal than GCD of delivery segment lengths
  - E.g. for 4,6, and 10-sec segments, GOP <= gcd(4,6,10) = 2 sec.
- GOP length may also be affected by the need to support SSAI / splicing
- closed GOP, SAP type 1 is a requirement for HLS

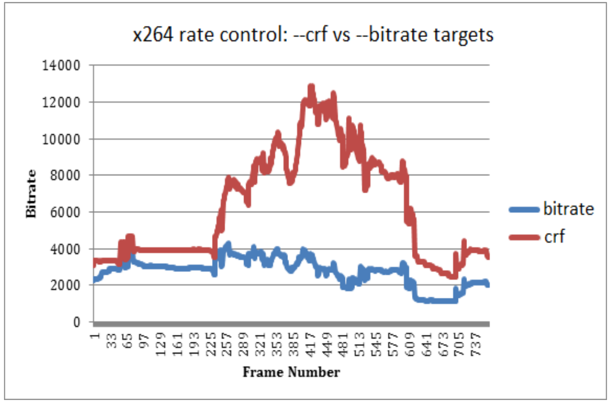
Profiles & levels

- H.264 Baseline profile is needed for legacy devices (e.g. mobiles prior to 2012)
- Main profile is adequate for streams of up to 720p
- High is better for 1080p and beyond
- Level must be sufficient to allow given resolution, framerate, bitrate, CPB size

Reference frames, B frames:

- for legacy devices (e.g. mobiles prior to 2012) – no B frames, 1 reference
- most STBs can support up to 4 reference frames, 3 B-frames

Example of uncapped VBR behavior:

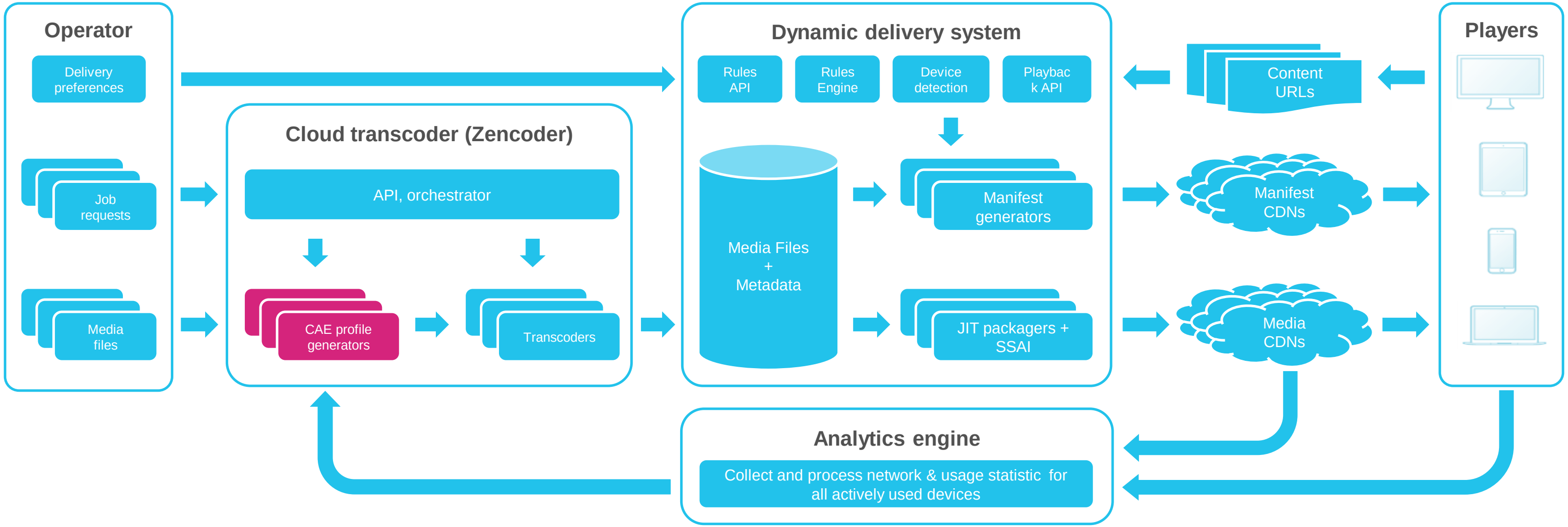


Example level/profile combinations:

Profile	Level	@codec Parameter (avc1 sample entry)	@codec Parameter (avc3 sample entry)
Constrained Baseline	2.1	avc1.42c015	avc3.42c015
Constrained Baseline	3.0	avc1.42c01e	avc3.42c01e
Main	3.0	avc1.4d401e	avc3.4d401e
Main	3.1	avc1.4d401f	avc3.4d401f
High	3.0	avc1.64001e	avc3.64001e
High	3.1	avc1.64001f	avc3.64001f
High	3.2	avc1.640020	avc3.640020
High	4.0	avc1.640028	avc3.640028

CONTEXT AWARE ENCODING

- Overall architecture:



- Context Aware Encoding (CAE) is basically a
  - ABR encoding profile generator that considers:
    - properties of content and
    - properties of networks and devices used to receive content

ABR ENCODING PROFILES

- Define sets of encoding parameters for each rendition
  - Resolutions, Bitrates, Codec constraints, etc.

- Examples of existing ABR profiles:
  - Apple HLS guidelines:

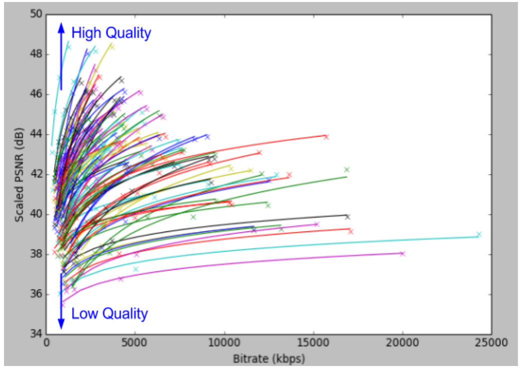
HEVC/H.265	H.264/AVC	Resolution	Frame rate
145	145	416 x 234	≤ 30 fps
350	365	480 x 270	≤ 30 fps
660	730	640 x 360	≤ 30 fps
990	1100	768 x 432	≤ 30 fps
1700	2000	960 x 540	same as source
2400	3000	1280 x 720	same as source
3200	4500	same as source	same as source
4500	6000	same as source	same as source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

Brightcove VideoCloud (legacy static profiles):

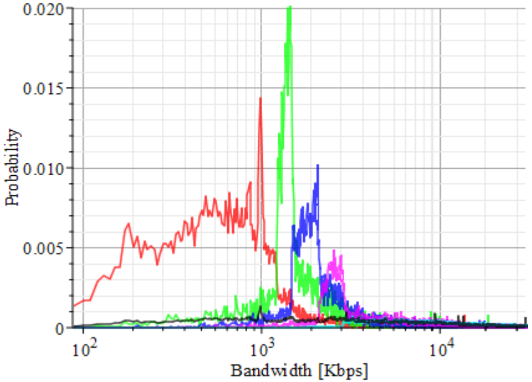
video bitrate	decoder bitrate cap	decoder buffer size	max frame rate	width	height	h264 profile
450	771	1028	30	480	270	baseline
700	1194	1592	30	640	360	baseline
900	1494	1992	30	640	360	main
1200	1944	2592	30	960	540	main
1700	2742	3656	30	960	540	main
2500	3942	5256	30	1280	720	main
3500	5442	7256	30	1920	1080	high
3800	6192	8256	30	1920	1080	high

WHY STATIC ABR PROFILES ARE BAD?

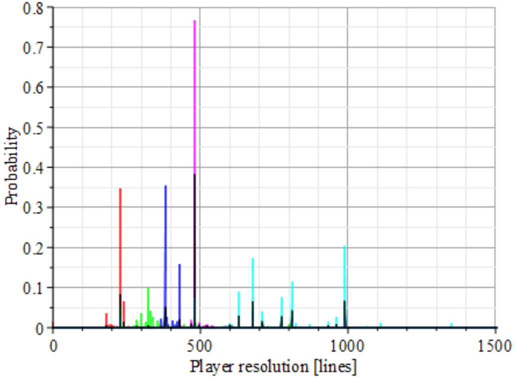
- Static encoding profiles are not accounting for:
  - differences in video complexity: differences in networks: differences in devices & user preferences:



Source: Netflix, 2015



Source: Brightcove VideoCloud analytics, 2019



Source: Brightcove VideoCloud analytics, 2019

- **A better approach is to design encoding profiles dynamically,** accounting for characteristics of
  - content → content-aware encoding (aka per-title encoding)
  - network → network-aware encoding
  - full context (content + network + user statistics) → **context-aware encoding**

DESIGN OF OPTIMAL ABR ENCODING PROFILES

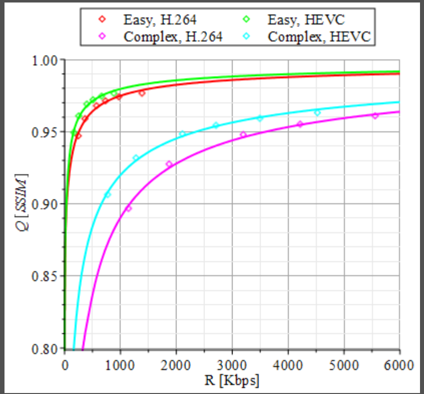
CONTENT, CODECS, QUALITY METRICS

- Content may vary in complexity
  - E.g. cartoons take less bits to encode than high-action sports
- Codecs vary in efficiency
  - E.g. HEVC in most cases is more efficient than H.264
  - AV1 and VVC are even better
  - But differences are content-dependent
  - And in many cases deltas can be small (e.g. 10-20%)
- There are many "quality" metrics
  - MSE and PSNR measure average amount of noise
  - SSIM normalizes it by local energy, making it closer to "contrast sensitivity"
  - VDP and PQR use more sophisticated models of vision
  - VMAF fuses several basic metrics – still work in progress
- However, for each combination of (content, codec, metric) it is always possible to define **quality-rate function Q(R)**

Examples of quality-rate functions

Content: "Easy" = cartoon, "Complex" = soccer game, 720p24
Codecs: H.264, HEVC (main profile, 2sec GOP, CRF rate control)
Metric: SSIM

Quality-rate functions:



Models:

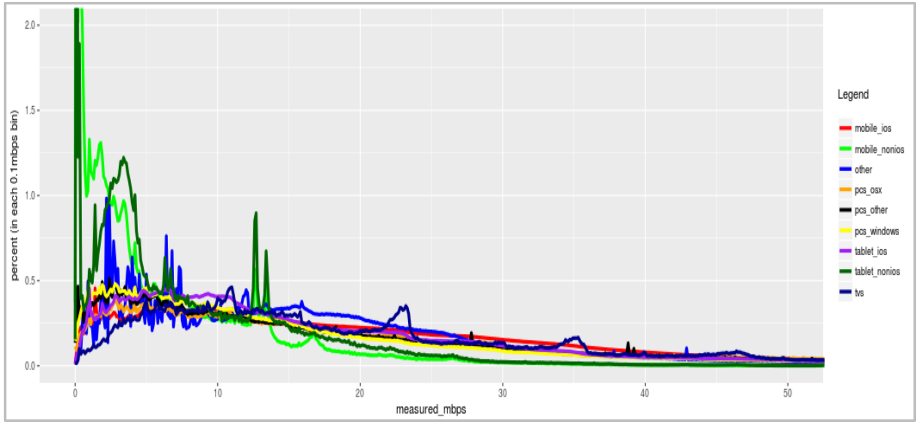
$$Q(R) = \frac{R^\beta}{\alpha^\beta + R^\beta}$$

Model parameters:

Content	Codec	α	β
Easy	H.264	0.542079	0.483651
Easy	HEVC	0.483928	0.506898
Complex	H.264	20.526129	0.547788
Complex	HEVC	10.666744	0.538406

NETWORK MODELS

- This is something that needs be measured by using player or CDN-originated logs
- Such distributions will be different for each category of receiving devices, region, CDN configuration, etc.



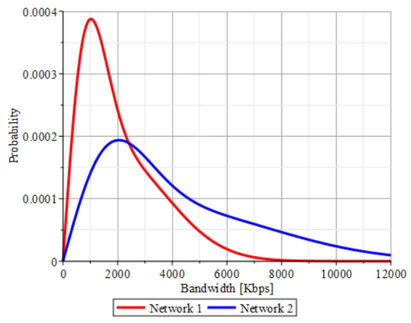
- But regardless of the context, and measurement technique, what matters in the end – is **bandwidth distribution model p(R)**

Examples of network models

Networks: LTE with 10 and 20 users in a cell

Based on: TCP-level throughput measurements reported in:
J. Karlsson, and M. Riback. Initial field performance measurements of LTE, Ericsson review, 3, 2008.

Network models:



Models:

$$p(R) = \alpha\, f(R,\sigma_1) + (1-\alpha) f(R,\sigma_2), \quad f(x,\sigma) = \frac{x}{\sigma^2}\exp\left(-\frac{x^2}{2\sigma^2}\right)$$

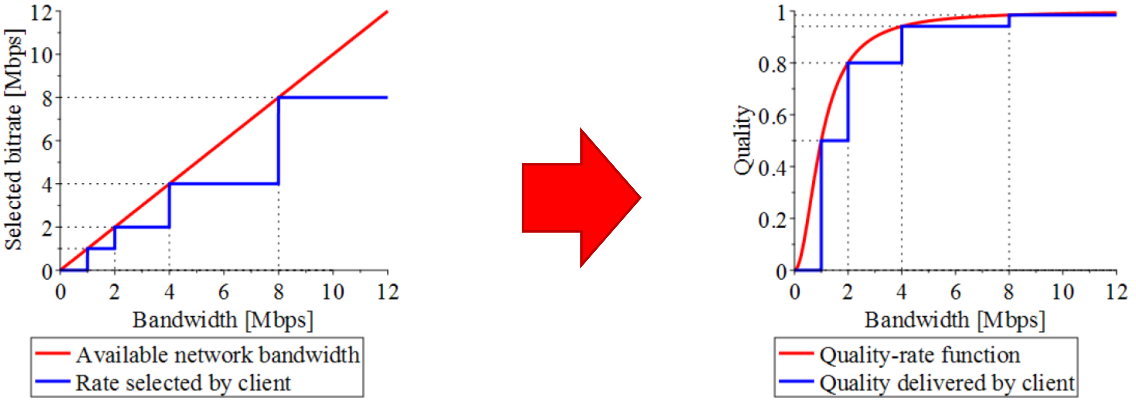
Model parameters:

Network	α	σ_1	σ_2
Network 1	0.4287	901.1	2249.6
Network 2	0.4287	1802.2	4499.2

CLIENT MODELS

- Let's assume that $R_1, \dots, R_n$ denote rates of encoding ladder, and R denotes available network bandwidth
- Then, the simplest conceptual client model is the following:

$$R^{selected}(R) = \max_{i=1,\dots,n} R_i \leq R, \qquad Q^{selected}(R) = Q(R^{selected}(R))$$



- Then, using such model we can easily compute **average rate** and **average quality** delivered by the system:

$$\bar{R}(R_1, \dots, R_n, p) = \int_0^\infty R^{selected}(R)p(R)dR\,, \qquad \bar{Q}(R_1, \dots, R_n, p) = \int_0^\infty Q^{selected}(R)p(R)dR$$

where $p(R)$ is a bandwidth distribution model.

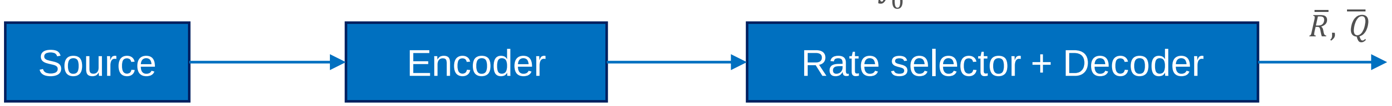
QUALITY-OPTIMAL LADDERS

- Conceptually, ABR streaming system can be understood as a chain:

$$\bar{R} = \int_0^\infty R^{selected}(R)\, p(R) dR$$

$$\bar{Q} = \int_0^\infty Q^{selected}(R)\, p(R) dR$$

$$\mathcal{L} = \{(R_i, Q_i), i = 1, \dots, n\}$$



- One problem that can be immediately posed is following:
  - given the number of renditions n, quality-rate model $Q(R)$, and network model $p(R)$
  - find a set of rates $\hat{R}_1, \dots, \hat{R}_n$, such that:

$$\bar{Q}(\hat{R}_1, \dots, \hat{R}_n, p) = \max_{\substack{R_{\min} < R_1 \le \cdots \le R_n < R_{\max} \\ R_1 \le R_{1,\max}}} \bar{Q}(R_1, \dots, R_n, p)$$

- We will call ABR ladder with rates $\hat{R}_1, \dots, \hat{R}_n$ -- **quality-optimal ladder**

EXAMPLES OF QUALITY-OPTIMAL LADDERS

"Easy" content, H.264, Network 1:

n	Ladder Bitrates [kbps]	Q_n	$\bar{Q}$	$\bar{R}$
2	91, 719	0.9700	0.9607	627.5
3	59, 403, 1222	0.9767	0.9676	929.4
4	50, 293, 773, 1736	0.9802	0.9706	1160
5	50, 242, 585, 1123, 2214	0.9824	0.9723	1331
6	50, 209, 473, 850, 1421, 2568	0.9836	**0.9733**	**1445**
7	50, 187, 401, 692, 1087, 1687, 2843	0.9844	0.9739	1527
8	50, 170, 351, 589, 893, 1302, 1933, 3076	0.9849	0.9744	1590

"Easy" content, HEVC, Network 1:

n	Ladder Bitrates[kbps]	Q_n	$\bar{Q}$	$\bar{R}$
2	85, 695	0.9755	0.9674	611.3
3	54, 384, 1188	0.9812	**0.9735**	**913**
4	50, 286, 758, 1706	0.9843	0.9761	1151
5	50, 237, 573, 1104, 2182	0.9861	0.9775	1323
6	50, 205, 463, 835, 1399, 2537	0.9871	0.9784	1438
7	50, 183, 393, 679, 1068, 1662, 2812	0.9878	0.979	1520
8	50, 166, 343, 577, 876, 1280, 1904, 3045	0.9883	0.9794	1584

"Complex" content, H.264, Network 1:

n	Ladder Bitrates [kbps]	Q_n	$\bar{Q}$	$\bar{R}$
2	210, 946	0.8971	0.8598	773.8
3	147, 576, 1456	0.9182	0.8796	1043
4	114, 418, 928, 1942	0.9301	0.8893	1239
5	93, 327, 686, 1233, 2339	0.9369	0.8951	1375
6	79, 267, 544, 925, 1499, 2640	0.9409	0.8988	1470
7	69, 226, 451, 744, 1137, 1735, 2868	0.9436	0.9013	1540
8	61, 197, 387, 627, 930, 1338, 1967, 3099	0.9460	**0.9032**	**1599**

"Complex" content, HEVC, Network 1:

n	Ladder Bitrates[kbps]	Q_n	$\bar{Q}$	$\bar{R}$
2	163, 860	0.9292	**0.9044**	**721.2**
3	111, 509, 1363	0.9442	0.9191	1000
4	85, 364, 859, 1847	0.9524	0.9261	1205
5	69, 281, 630, 1169, 2261	0.9573	0.9302	1350
6	58, 228, 494, 870, 1437, 2576	0.9601	0.9328	1450
7	51, 192, 408, 697, 1087, 1682, 2830	0.9621	0.9346	1526
8	50, 174, 356, 592, 893, 1298, 1922, 3059	0.9636	0.9359	1589

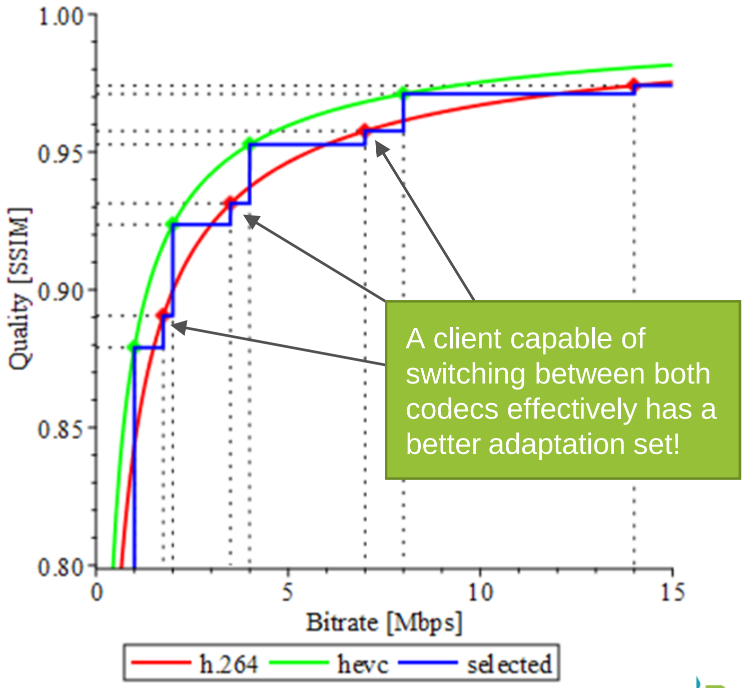
Notations: Q_n is quality at top rendition [SSIM], $\bar{Q}$ is an average quality [SSIM], and $\bar{R}$ is an average bitrate [Kbps].

MULTI-CODEC ABR PROFILES

- **Consider now a system with**
  - 2 codecs, e.g. H.264 and HEVC
  - 3 types of client devices:
    - 1st can decode only first codec (H.264)
    - 2nd can decode only second codec (HEVC), and
    - 3rd can decode streams encoded using **both codecs**, and it that can also switch between such streams

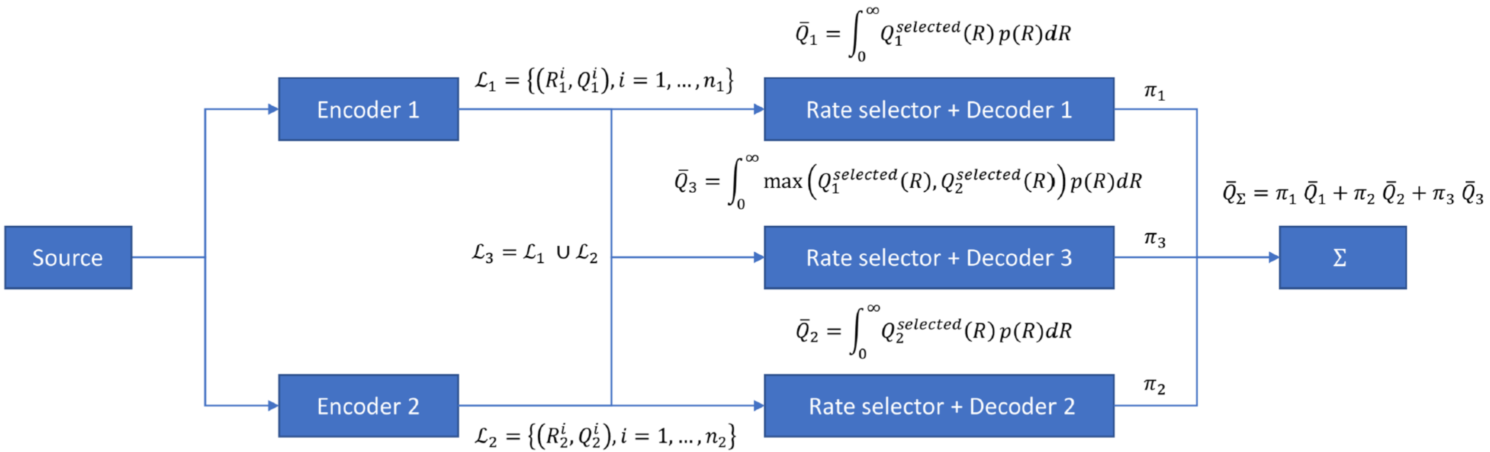
- The existence of the **3rd type** of client is important, as it could, in principle, achieve better quality than the other two clients:

- This of course requires special design of a mixed ladder: interleaving of rates allocated to each codec, making sure that quality-wise they form monotonically increasing sequence, that steps between renditions offer meaningful increments, etc. But this is all doable!



QUALITY OPTIMAL 2-CODEC LADDERS

- Let's now formalize the problem of optimal design of dual-codec ladder:



- The problem: find numbers $\hat{n}_1 + \hat{n}_2 = n$, and ladder rates $\hat{R}_1^1, \dots, \hat{R}_1^{\hat{n}_1}$ and $\hat{R}_2^1, \dots, \hat{R}_2^{\hat{n}_2}$, such that overall quality $\bar{Q}_\Sigma$ is maximal:

$$\bar{Q}_\Sigma\left(p, \pi, n, \hat{R}_1^1, \dots, \hat{R}_1^{\hat{n}_1}, \hat{R}_2^1, \dots, \hat{R}_2^{\hat{n}_2}\right) = \max_{\substack{n_1 + n_2 = n \\ R_{\min} \le R_1^1 \le \dots \le R_1^{n_1} \le R_{\max} \\ R_{\min} \le R_2^1 \le \dots \le R_2^{n_2} \le R_{\max} \\ R_1^1, R_2^1 \le R_{\max}^1}} \bar{Q}_\Sigma\left(p, \pi, n, R_1^1, \dots, R_1^{n_1}, R_2^1, \dots, R_2^{n_2}\right)$$

OPTIMAL 2-CODEC LADDERS

- Network 1, "Easy" content, 2-codec ladder, $\pi_{hevc} = 0.4, \pi_{h264} = 0.2$:

n	H.264 bitrates [kbps]	HEVC bitrates [kbps]	Q_n	$\bar{Q}_\Sigma$	$\bar{R}_\Sigma$
2	91, 719		0.97	0.9607	627.4
3	59, 403, 1222		0.9767	0.9676	929.3
4	50, 293, 773, 1736		0.9802	0.9706	1160
5	50, 242, 585, 1123, 2214		0.9824	0.9723	1331
6	91, 719	50, 286, 758, 1706	0.9843	**0.9733**	1050
7	59, 403, 1222	50, 286, 758, 1706	0.9843	**0.9744**	1152
8	59, 403, 1222	50, 237, 573, 1104, 2182	0.9861	0.9756	1249

Observations:

- if n<6 single codec (H.264) is used
- at n=6 dual-codec ladder attains same average quality as 6-point H.264 ladder, yet reducing bitrate by almost 40%
- at n=7 dual codec ladder attains same quality as 8-stream H.264 ladder + 4-stream HEVC ladder constructed separately

- Optimal H.264-only ladders:

n	Ladder Bitrates [kbps]	Q_n	$\bar{Q}$	$\bar{R}$
2	91, 719	0.97	0.9607	627.5
3	59, 403, 1222	0.9767	0.9676	929.4
4	50, 293, 773, 1736	0.9802	0.9706	1160
5	50, 242, 585, 1123, 2214	0.9824	0.9723	1331
6	50, 209, 473, 850, 1421, 2568	0.9836	**0.9733**	1445
7	50, 187, 401, 692, 1087, 1687, 2843	0.9844	0.9739	1527
8	50, 170, 351, 589, 893, 1302, 1933, 3076	0.9849	**0.9744**	1590

Optimal HEVC-only ladders:

n	Ladder Bitrates[kbps]	Q_n	$\bar{Q}$	$\bar{R}$
2	85, 695	0.9755	0.9674	611.3
3	54, 384, 1188	0.9812	**0.9735**	913
4	50, 286, 758, 1706	0.9843	**0.9761**	1151
5	50, 237, 573, 1104, 2182	0.9861	0.9775	1323
6	50, 205, 463, 835, 1399, 2537	0.9871	0.9784	1438
7	50, 183, 393, 679, 1068, 1662, 2812	0.9878	0.979	1520
8	50, 166, 343, 577, 876, 1280, 1904, 3045	0.9883	0.9794	1584

OPTIMAL 2-CODEC LADDERS, COMPLEX CONTENT

- Network 1, "Complex" content, 2-codec ladder, $\pi_{hevc} = 0.4, \pi_{h264} = 0.2$:

n	H.264 bitrates [kbps]	HEVC bitrates [kbps]	Q_n	$\bar{Q}_\Sigma$	$\bar{R}_\Sigma$
2	210, 946		0.8971	0.8598	773.7
3	391	163, 860	0.9292	**0.8833**	651.6
4	391	111, 509, 1363	0.9442	0.8956	879.9
5	210, 946	111, 509, 1363	0.9442	**0.9072**	954.9
6	210, 946	85, 364, 859, 1847	0.9524	0.9129	1118
7	147, 576, 1456	85, 364, 859, 1847	0.9524	0.9168	1172
8	147, 576, 1456	69, 281, 630, 1169, 2261	0.9573	0.9201	1288

Observations:

- if n<3 single codec (H.264) is used
- at n=3 dual-codec ladder attains higher average quality as 3-point H.264 ladder, yet reducing bitrate by almost 40%
- at n=5 dual-codec ladder attains quality comparable to one of 8-stream H.264 + 2-stream HEVC ladders constructed separately!

- Optimal H.264-only ladders:

n	Ladder Bitrates [kbps]	Q_n	$\bar{Q}$	$\bar{R}$
2	210, 946	0.8971	0.8598	773.8
3	147, 576, 1456	0.9182	0.8796	1043
4	114, 418, 928, 1942	0.9301	**0.8893**	1239
5	93, 327, 686, 1233, 2339	0.9369	0.8951	1375
6	79, 267, 544, 925, 1499, 2640	0.9409	0.8988	1470
7	69, 226, 451, 744, 1137, 1735, 2868	0.9436	0.9013	1540
8	61, 197, 387, 627, 930, 1338, 1967, 3099	0.946	**0.9032**	1599

Optimal HEVC-only ladders:

n	Ladder Bitrates[kbps]	Q_n	$\bar{Q}$	$\bar{R}$
2	163, 860	0.9292	**0.9044**	721.2
3	111, 509, 1363	0.9442	0.9191	1000
4	85, 364, 859, 1847	0.9524	0.9261	1205
5	69, 281, 630, 1169, 2261	0.9573	0.9302	1350
6	58, 228, 494, 870, 1437, 2576	0.9601	0.9328	1450
7	51, 192, 408, 697, 1087, 1682, 2830	0.9621	0.9346	1526
8	50, 174, 356, 592, 893, 1298, 1922, 3059	0.9636	0.9359	1589

SOME REFERENCES

Original paper on quality-optimal streaming:

Y. Reznik, K. Lillevold, A. Jagannath, J. Greer, and J. Corley, "Optimal design of encoding profiles for ABR streaming," *Proc. Packet Video Workshop*, Amsterdam, The Netherlands, June 12, 2018.

Multi-codec ladder design optimizations:

Y. Reznik, X. Li, K. Lillevold, A. Jagannath, and J. Greer, "Optimal Multi-Codec Adaptive Bitrate Streaming," *Proc. IEEE Int. Conf. Multimedia and Expo (ICME)*, Shanghai, China, July 8-12, 2019.

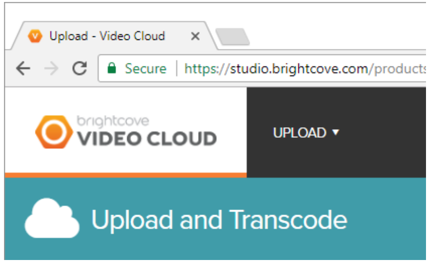
Additional details and case studies:

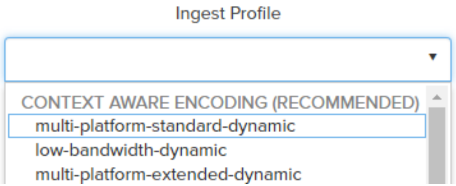
Y. Reznik, X. Li, K. Lillevold, R. Peck, T. Shutt, R. Marinov, "Optimizing Mass-Scale Multi-Screen Video Delivery," *Proc. 2019 NAB Broadcast Engineering and Information Technology Conference*, Las Vegas, NV, April 6-11, 2019.

IN PRACTICE...

CAE PROFILE GENERATOR

"CAE" ingest profiles in VideoCloud and Zencoder:





- In VideoCloud there are several standard CAE profiles, as shown above.
- Advanced users can also define custom CAE profiles using JSON descriptor.
- This way users may specify limits for:
  - number of renditions
  - range of bitrates to be used
  - list of allowed video resolutions, framerates, codecs, codec profiles & levels
  - allowed granularity of rate steps in the profile
  - network and usage statistics, etc.

Example custom CAE profile:


```
{
"id": "1234567890",
"version": 1,
"name": "custom-cae-profile",
"dynamic_origin": {
  "dynamic_profile_options": {
    "min_renditions": 2,
    "max_renditions": 6,
    "max_resolution": {
      "width": 1920,
      "height": 1080
    },
    "max_bitrate": 4200,
    "max_first_rendition_bitrate": 400,
    "max_frame_rate": 30,
    "keyframe_rate": 0.5,
    "codecs": ["h264","hevc"],
    "max_granularity": 100,
    "video_configurations": [
        {"width": 1280, "height": 720},
        {"width": 960,  "height": 540},
        {"width": 640,  "height": 360}
    ]
}}}
```

RESULTS – ADAPTATIONS TO DIFFERENT TYPES OF CONTENT

Study conducted using:

- 500 assets
- 120 hours of view time
- 34 different content categories
- Including movies, cartons, sports, etc.

Reference profile – Apple TV:

Resolution	Bitrate	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

Relative changes [in %] for each category of content:

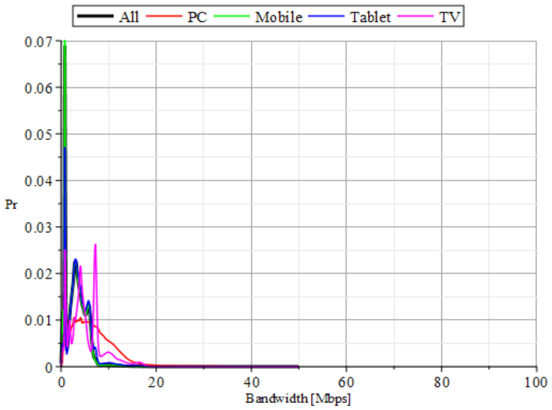
Category	Streams	Storage	Bandwidth	Resolution
Action	-35.05	-77.28	-59.16	+3.57
Adventure	-29.63	-70.17	-51.33	+3.32
Comedy	-25.12	-62.16	-41.28	+2.33
Drama	-32.36	-73.29	-55.83	+3.55
Scifi	-31.38	-71.89	-53.17	+3.27
Cartoon	-30.15	-68.82	-47.71	+2.93
Video game	-29.2	-67.76	-46.17	+3.17
Baseball	-21.57	-61.09	-50.89	+0.76
Basketball	-22.1	-57.82	-34.15	+1.72
Boxing	-23.71	-65.33	-43.03	+3.1
Cricket	-14.29	-58.12	-50.13	+0.97
Cycling	-23.11	-58.92	-36.55	+2.35
Field hockey	-22.22	-51.57	-22.66	+1.1
Football	-28.57	-79.12	-52.25	+1.69
Golf	-28.57	-79.38	-74.2	+1.69
Gymnastics	-26.1	-65.45	-44.01	+2.79
Hockey	-22.22	-51.26	-20.39	+0.08

Category	Streams	Storage	Bandwidth	Resolution
Mixed sports	-23.63	-55.47	-29.22	+1.35
Racing	-28.57	-74.68	-66.96	+1.5
Running	-23.3	-56.66	-31.99	+2.52
Squash	-27.56	-67.18	-47.11	+3.22
Swimming	-22.22	-50.04	-19.67	+0.17
Tennis	-18.72	-61.04	-51.44	+1.07
Weightlifting	-31.44	-72.6	-51.66	+3.78
Documentary	-25.72	-59.85	-34.19	+2.19
Game show	-28.16	-65.18	-40.95	+3.02
Interview	-37.33	-81.17	-74.2	+1.6
Kids channel	-24.75	-59.52	-34.04	+1.69
Talk show	-36.07	-77.76	-59.02	+3.99
News	-25.97	-62.36	-39.64	+2.24
Reality TV	-24.94	-58.51	-33.52	+2.46
Sitcom	-31.49	-71.93	-54.04	+3.23
Soap opera	-34.92	-76.61	-58.83	+3.8
Overall	**-28.42**	**-65.64**	**-43.76**	**+2.65**

RESULTS – ADAPTATIONS TO OPERATORS/NETWORKS

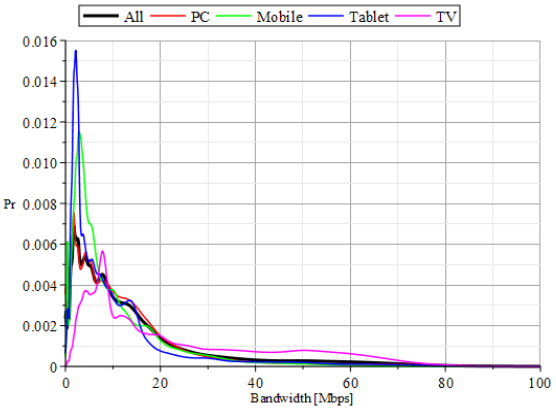
Study considering 3 operators, same content, same reference profiles:

Operator 1:



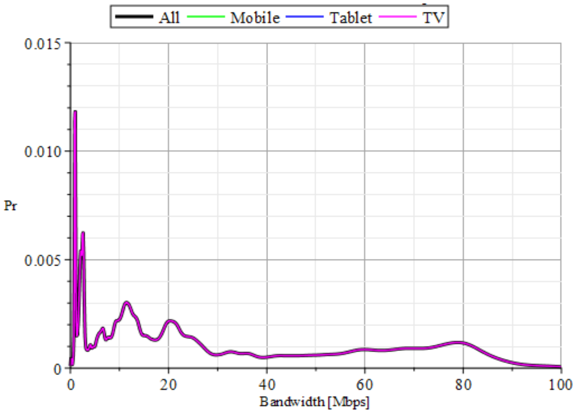
Device type	Usage [%]	Average bandwidth [Mbps]
PC	0.004	7.5654
Mobile	94.321	3.2916
Tablet	5.514	3.8922
TV	0.161	5.4374
All devices	**100**	**3.3283**

Operator 2:



Device type	Usage [%]	Average bandwidth [Mbps]
PC	63.49	14.720
Mobile	6.186	10.609
Tablet	9.165	12.055
TV	21.15	24.986
All devices	**100**	**16.393**

Operator 3:



Device type	Usage [%]	Average bandwidth [Mbps]
PC	0.0	N/A
Mobile	0.0	N/A
Tablet	0.0	N/A
TV	100	35.7736
All devices	100	35.7736

ADAPTATIONS TO DIFFERENT NETWORKS/DEVICES

Content: movie trailer sequence

CAE encoding profiles generated for different operators:

Operator 1

Stream	Profile	Resolution	Framerate	Bitrate	SSIM
1	Baseline	320x180	30	125	0.9336
2	Baseline	480x270	30	223.08	0.9379
3	Main	640x360	30	398.11	0.9463
4	Main	960x540	30	774.78	0.9495
5	Main	1280x720	30	1549.5	0.9563
6	High	1600x900	30	2765.3	0.9610
7	High	1920x1080	30	4935.1	0.9657
Storage				10771	

Higher density at lower rates

Operator 2:

Stream	Profile	Resolution	Framerate	Bitrate	SSIM
1	Baseline	320x180	30	125	0.9333
2	Baseline	480x270	30	239.71	0.9412
3	Main	640x360	30	469.54	0.9520
4	Main	1024x576	30	939.08	0.9522
5	Main	1280x720	30	1568.8	0.9565
6	High	1600x900	30	2765.3	0.9610
7	High	1920x1080	30	4935.1	0.9657
Storage				11026	

Higher density mid range

Operator 3:

Stream	Profile	Resolution	Framerate	Bitrate	SSIM
1	Baseline	320x180	30	125	0.9344
2	Baseline	512x288	30	307.42	0.9485
3	Main	960x540	30	803.59	0.9505
4	Main	1280x720	30	1727.8	0.9586
5	High	1920x1080	30	5050.7	0.9659
Storage				8014.6	

Fewer renditions! Only last matters!

RESULTS – ADAPTATIONS TO OPERATORS/NETWORKS

Relative performance changes for each operator:

Metric	Operator 1	Operator 2	Operator 3
Renditions	-22.2%	-22.2%	-44.4%
Storage	-57.9%	-56.9%	-68.7%
Bandwidth	-8.4%	-31.3%	-33.8%
Resolution	+27.3%	+6.59%	+2.03%
SSIM	-0.9%	-0.74%	-0.68%
Buffering	-1.74%	-1.04%	-1.56%
Start time	-5.7%	-1.0%	-1.6%

Notes:

- For operator 1, having worst networks, the savings in bandwidth are smaller, but the average delivered resolution increases by 83%
- For operators 2 and 3, the savings in bandwidth increase to 31.3 and 33.89% respectively
- For operator 3, the number of streams is further reduced, leading to significant savings in transcoding and storage costs
- In all cases optimization have also improved start up time and % of time buffering
- All savings are achieved with negligible changes in codec noise as indicated by relative SSIM change values

CONCLUSIONS

- Modern-era OTT video delivery poses a number of challenges, and requires some dedicated tools:
  - **Dynamic packaging & delivery**
    - addresses the need to support multiple devices, codecs, DRMs, and delivery formats
    - minimizes storage, bandwidth and CDN costs
  - **High-scale, high-quality & high-reliability transcoder**
    - supporting all sorts of input formats
    - performing pre- and post-quality checks
    - having advanced set of pre-processing tools
    - conformant to existing ecosystem standards and deployment guidelines
  - **Context-aware encoding**
    - end-to-end optimization tool taking into account specifics of content, usage- and networks statistics
- It is indeed to a lot of fun to build such a system, and
  - even more so – to test it and make sure it works at scale!

brightcove

THANK YOU!

YURIY REZNIK | YREZNIK@BRIGHTCOVE.COM