brightcove

OPTIMAL MULTI-CODEC ADAPTIVE BITRATE STREAMING

XIANGBO LI | BRIGHTCOVE INC.

Agenda

- Introduction of ABR
- ABR evolvement
  - Per-title/Content-aware encoding
  - Context-aware encoding
- Optimal multi-codec ABR streaming
- Q & A

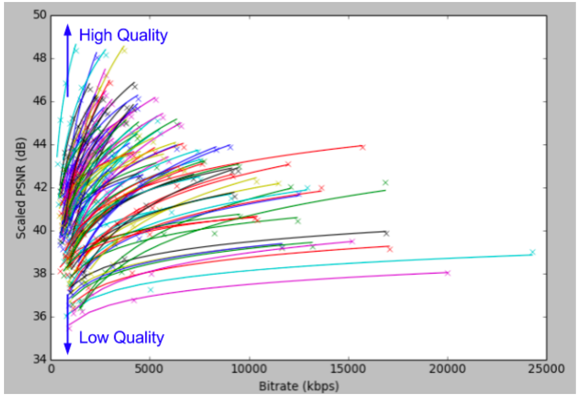
What is ABR streaming?

https://www.encoding.com/http-live-streaming-hls/

Encoding Ladder

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

- This “**one-size-fits-all**” bitrate ladder achieves, for most content, good quality encodes given the bitrate constraint
- However, it ignores the content

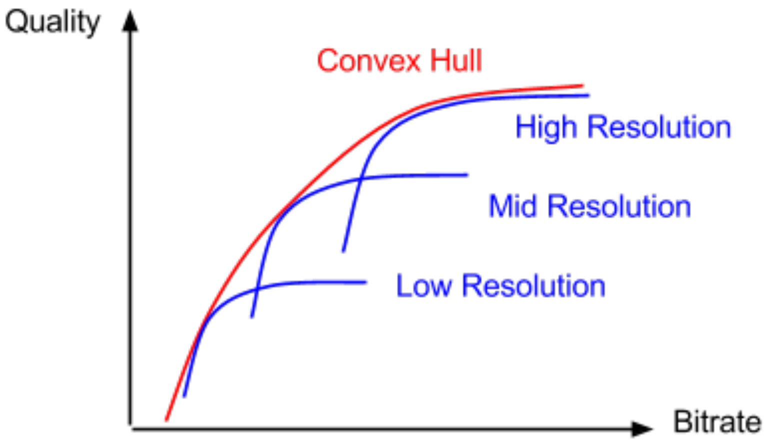


https://medium.com/netflix-techblog/per-title-encode-optimization-7e99442b62a2

Design encoding ladders dynamically

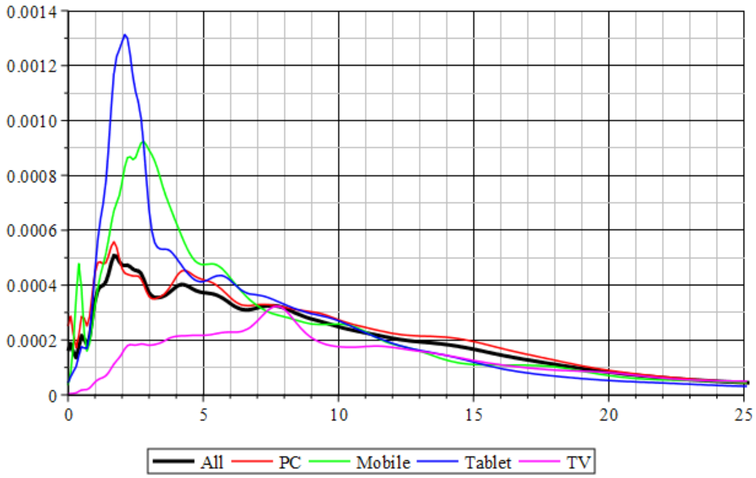
Characteristics of video contents

Per-Title / Content-Aware Encoding (Netflix 15')

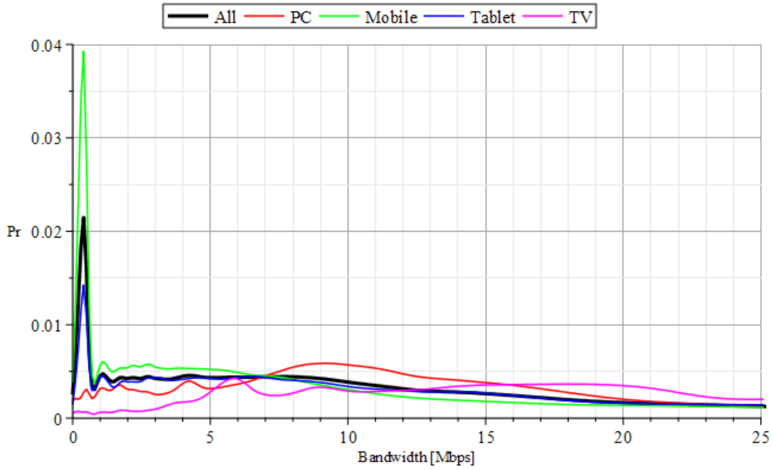


- The upper boundary of all such functions forms a **convex hull**.
- Hence, the key idea of per-title encoding is to **pick ladder points such that they belong to the convex hull.**
- It does not say how such bitrates should be placed, or how many of them are needed.

Network and usage statistics



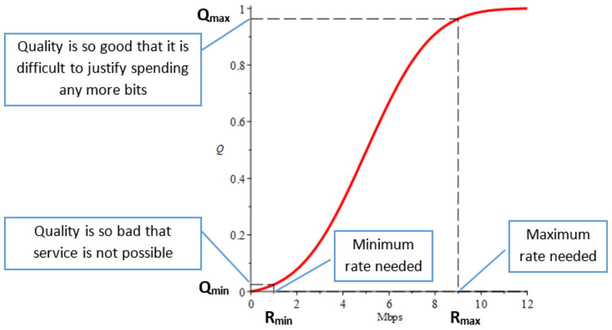
Australia



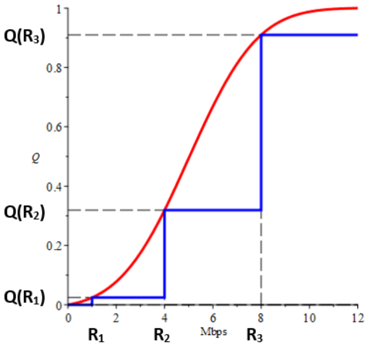
Thailand

Context-Aware Encoding (Brightcove 17')

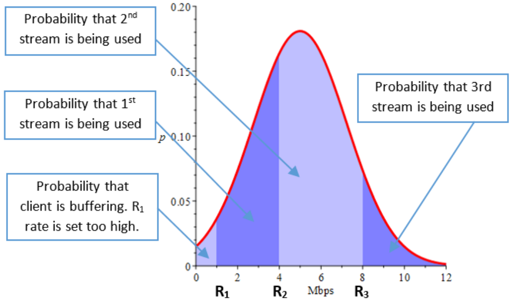
Quality-rate function $Q(R)$:



Quality at each encoding point:



Probabilities of loading each stream:

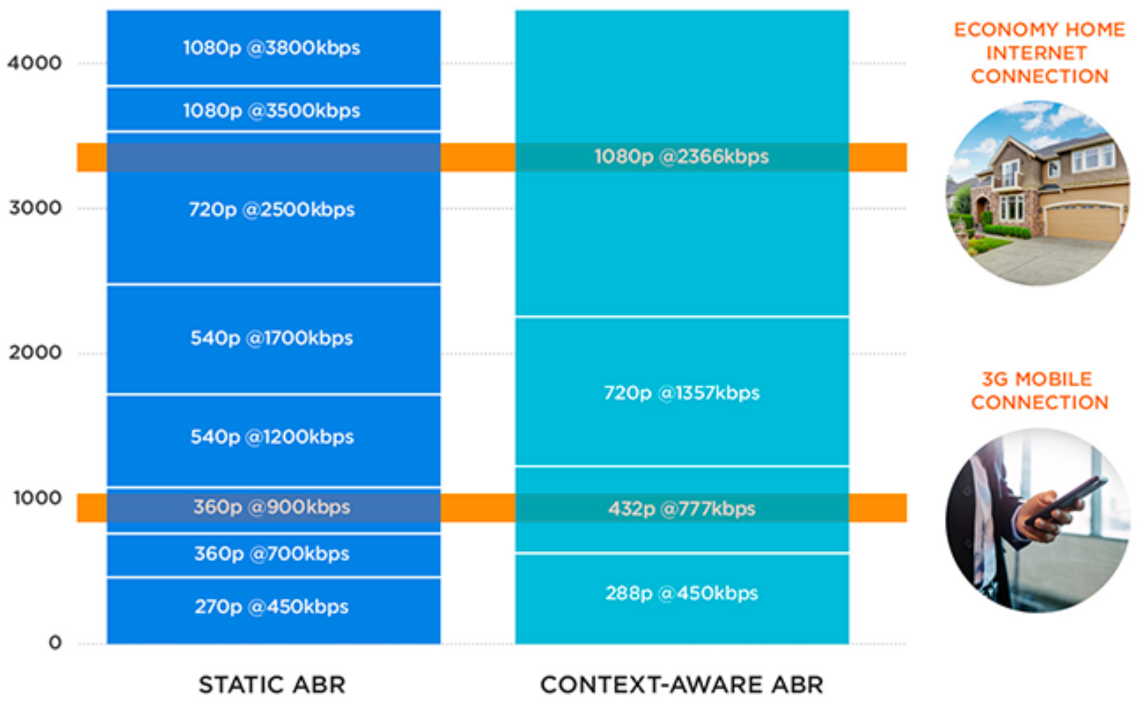


$$p(R_i \leq R < R_{i+1}) = \int_{R_i}^{R_{i+1}} p(R)dR$$

$$\overline{Q}(R_1, \ldots, R_n, p) = Q(R_1)\int_{R_1}^{R_2} p(R)dR \; + \; Q(R_1)\int_{R_2}^{R_3} p(R)dR + \; \ldots \; + Q(R_n)\int_{R_n}^{R_{\max}} p(R)dR$$

A **quality-optimal profile** is set of rates $R_1^*, \ldots, R_n^*$, such that:

$$\overline{Q}(R_1^*, \ldots, R_n^*, p) = \max_{\substack{R_{\min} < R_1 \leq \cdots \leq R_n < R_{\max} \\ R_1 \leq R_{1,\max}}} \overline{Q}(R_1, \ldots, R_n, p)$$


4000
3000
2000
1000
0
1080p @3800kbps
1080p @3500kbps
720p @2500kbps
540p @1700kbps
540p @1200kbps
360p @900kbps
360p @700kbps
270p @450kbps
1080p @2366kbps
720p @1357kbps
432p @777kbps
288p @450kbps
STATIC ABR
CONTEXT-AWARE ABR
ECONOMY HOME INTERNET CONNECTION
3G MOBILE CONNECTION

The era of multi-codec streaming

WWDC17

The following devices can capture media in HEIF or HEVC, if using **iOS 11 or later** or **macOS High Sierra or later**.

- iPhone 7 or iPhone 7 Plus or later
- iPad (6th generation)
- iPad Air (3rd generation)
- iPad mini (5th generation)
- iPad Pro (10.5 inch)
- iPad Pro (11 inch)
- iPad Pro 12.9-inch (2nd generation) or later
- Macbook

THREE DEVICE CATIGARIES

- **Only H.264 supported**
  Legacy Android and iOS devices

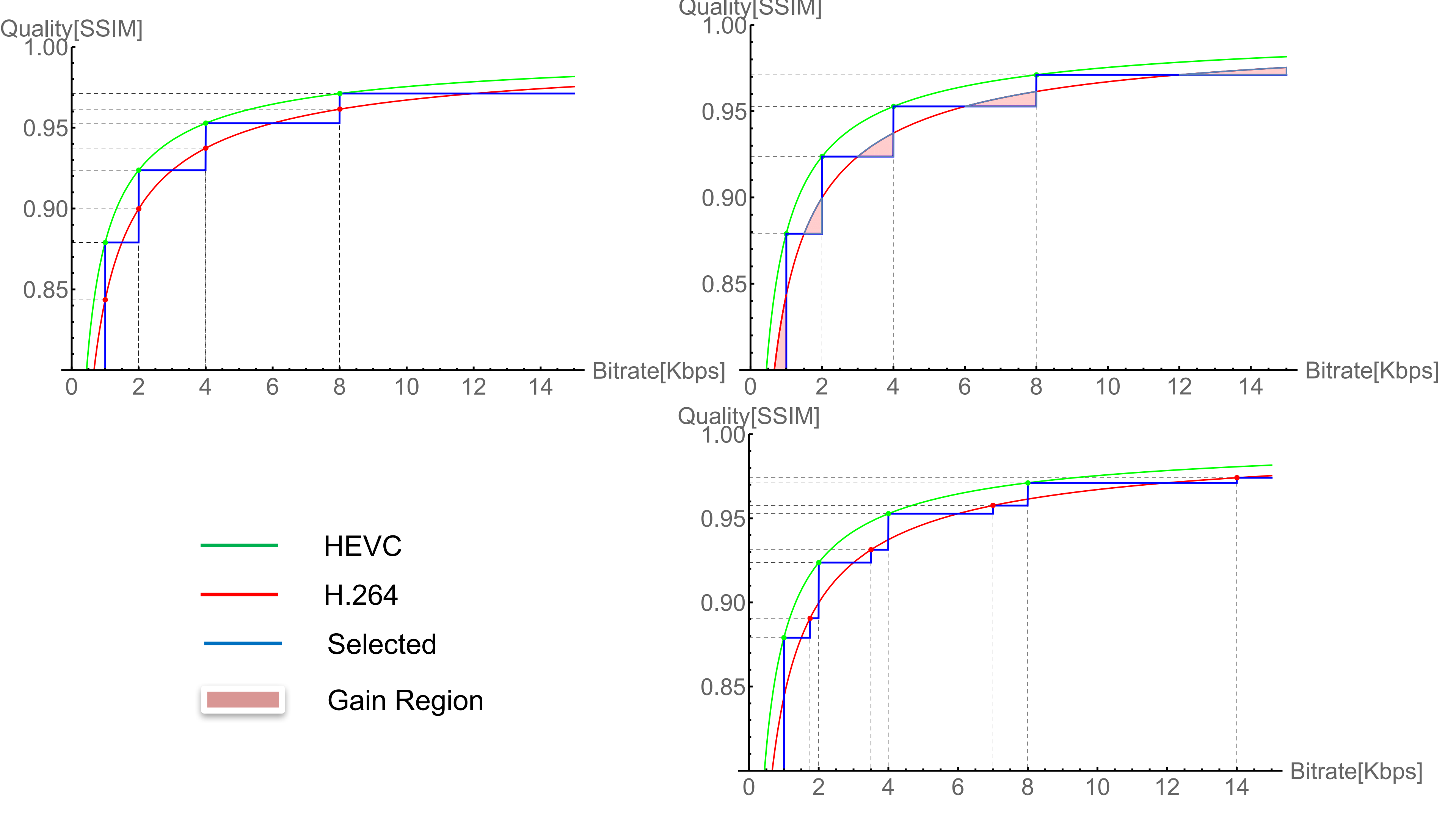
- **H.264 and HEVC switchable**
  Recent Apple devices with iOS 11++ and High Sierra++

- **H.264 and HEVC non-switchable**
  Recent Android/DASH devices

CHALLENGES

1. How many renditions in total we should generate?

2. How many rendition should be HEVC and H.264, respectively?

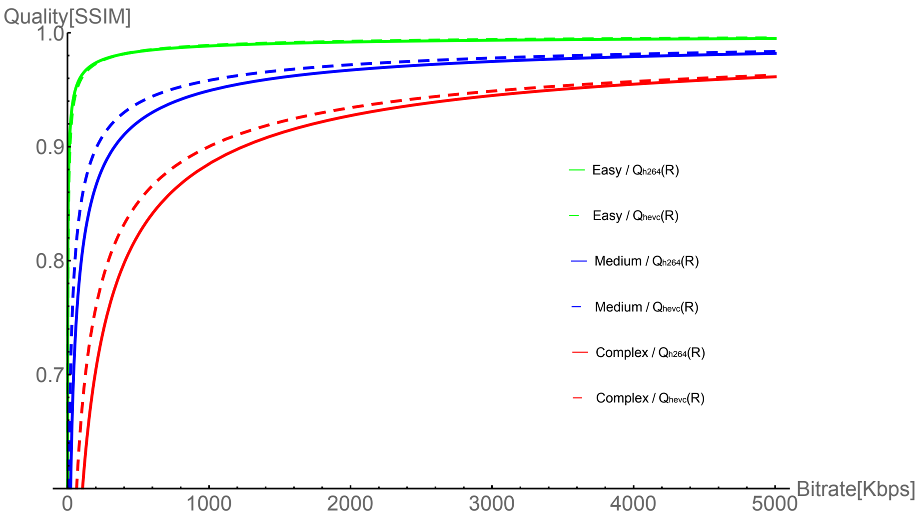


Quality[SSIM]
1.00
0.95
0.90
0.85
Bitrate[Kbps]
0
2
4
6
8
10
12
14
HEVC
H.264
Selected
Gain Region

Optimal Multi-Codec ABR

Experiment Setup

Test sequence	Resolution	Component sequences
Easy	720p	Johnny KristenAndSara FourPeople
Medium	720p	ParkScene FourPeople BasketballDrive Traffic
Complex	720p	BQTerrace BasketballDrive Cactus PeopleOnStreet NebutaFestival



Data Model: $$Q_{\alpha,\beta}(R) = \frac{R^\beta}{\alpha^\beta + R^\beta}$$

Test sequence	**H.264**		**HEVC**	
	α	β	α	β
Easy	0.1935	0.5600	0.3645	0.5674
Medium	12.0449	0.6623	5.1552	0.5947
Complex	60.9995	0.7295	34.7613	0.6548

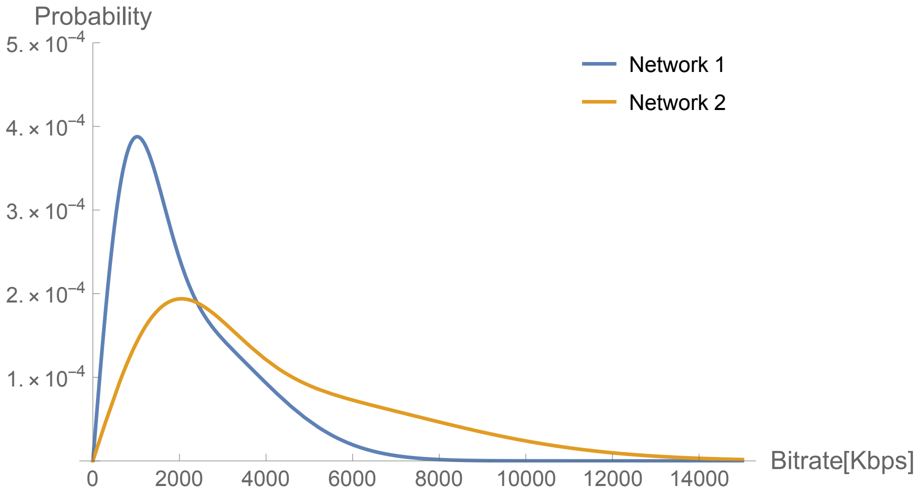
Experiment Setup (cont.)

Rayleigh Distribution:

$$f\left(R|\sigma\right) = \frac{x}{\sigma}\, e^{-\frac{x^2}{2\sigma^2}}$$

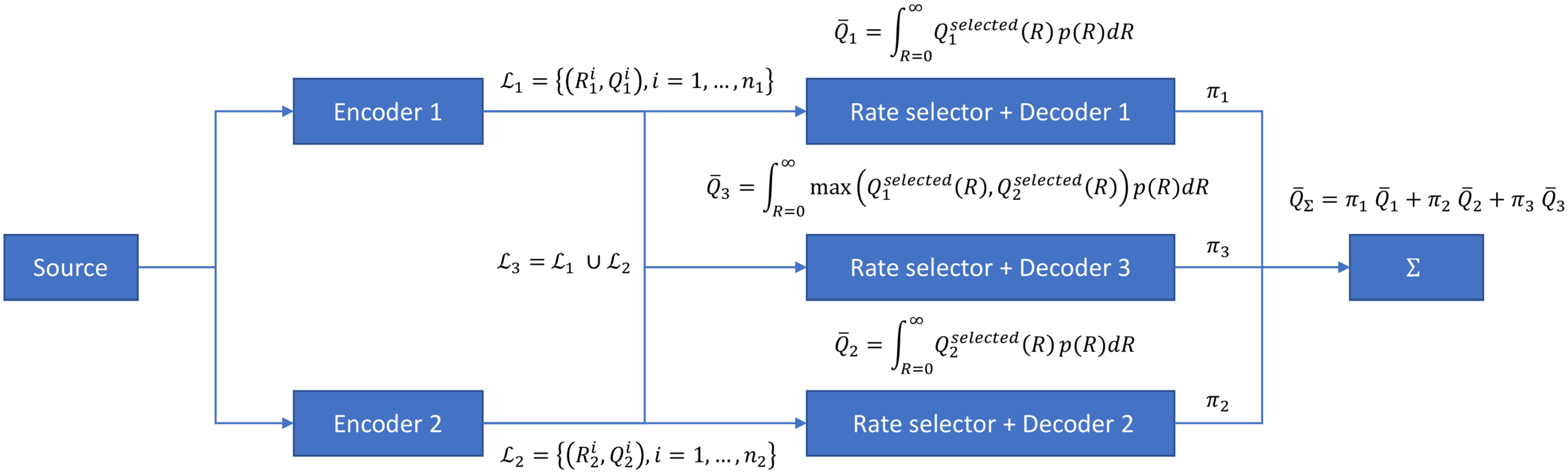
Network Model:

$$p_{\alpha,\sigma_1,\sigma_2}(R) = \alpha\, f\left(R|\sigma_1\right) + (1-\alpha) f\left(R|\sigma_2\right)$$



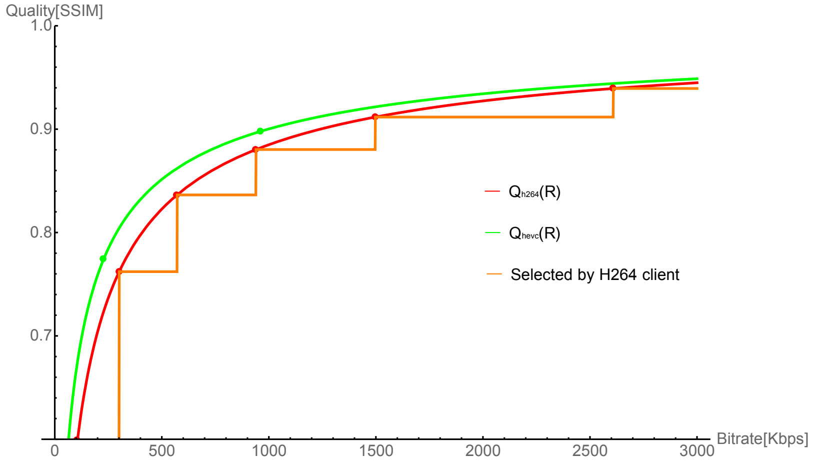
Network	Model Parameters		
	a	σ_1	σ_2
Network 1	0.4287	901.10	2249.64
Network 2	0.4287	1802.20	4499.27

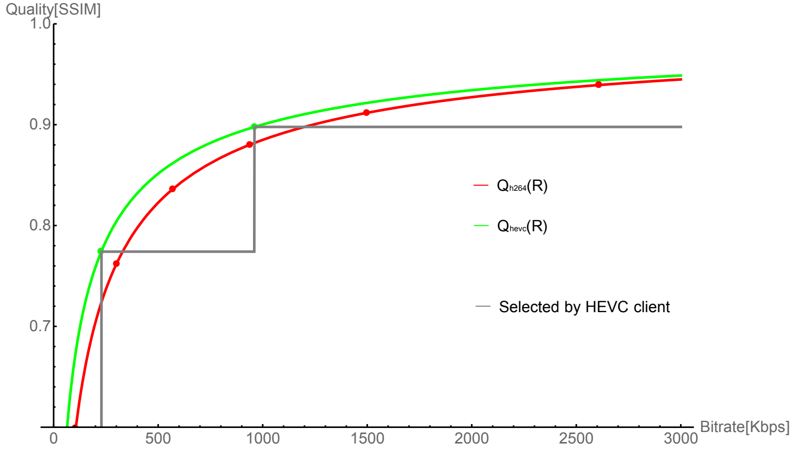
Architecture

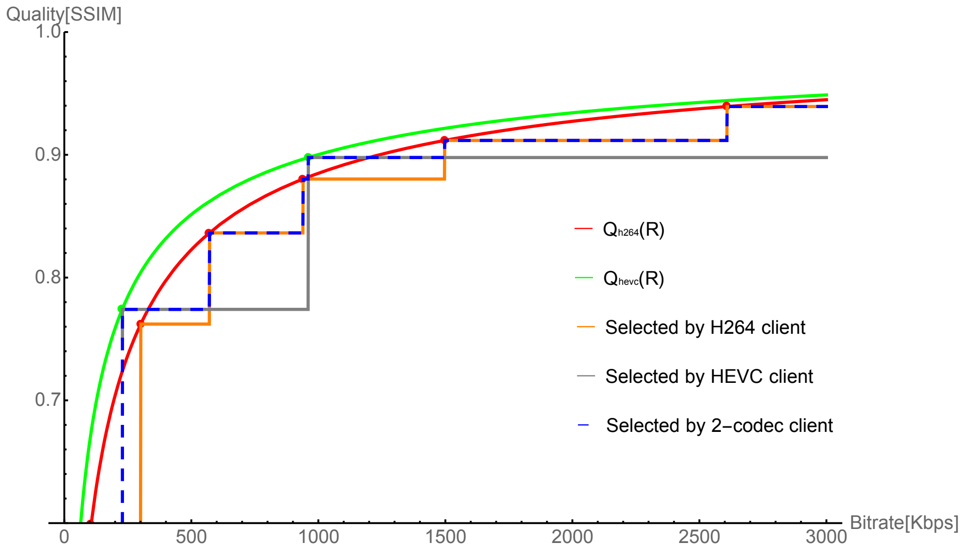


$\pi_1 = 60\%, \quad \pi_2 = 30\%, \quad \pi_3 = 10\%,$

Results


Quality[SSIM]
1.0
0.9
0.8
0.7
0
500
1000
1500
2000
2500
3000
Bitrate[Kbps]
Qh264(R)
Qhevc(R)
Selected by H264 client


Quality[SSIM]
1.0
0.9
0.8
0.7
0
500
1000
1500
2000
2500
3000
Bitrate[Kbps]
Qh264(R)
Qhevc(R)
Selected by HEVC client


Quality[SSIM]
1.0
0.9
0.8
0.7
0
500
1000
1500
2000
2500
3000
Bitrate[Kbps]
Qh264(R)
Qhevc(R)
Selected by H264 client
Selected by HEVC client
Selected by 2−codec client

Results (cont.)

Content	N	Ladder bitrates & codecs: H.264 and HEVC	Results for H.264-only clients				Results for HEVC-only clients				Results for H264 and HEVC capable clients				Weighted average across all clients	
			n	Q_n	$\overline{Q}$	ξ	n	Q_n	$\overline{Q}$	ξ	n	Q_n	$\overline{Q}$	ξ	$\overline{Q}$	ξ
Easy	2	156(h264), 164(hevc)	1	0.9701	**0.9626**	2.78	1	0.9697	**0.9614**	2.94	1	0.9701	**0.9626**	2.78	**0.9625**	2.79
	3	66(h264), 636(h264), 164(hevc)	2	0.9854	**0.9804**	0.98	1	0.9697	**0.9614**	2.94	3	0.9854	**0.9821**	0.81	**0.9790**	1.12
	4	50(h264), 366(h264), 1155(h264), 164(hevc)	3	0.9892	**0.9844**	0.57	1	0.9697	**0.9614**	2.94	4	0.9892	**0.9851**	0.50	**0.9823**	0.79
	5	50(h264), 366(h264), 1155(h264), 70(hevc), 633(hevc)	3	0.9892	**0.9844**	0.57	2	0.9857	**0.9803**	1.04	5	0.9892	**0.9856**	0.46	**0.9843**	0.58
	6	50(h264), 280(h264), 744(h264), 1680(h264), 70(hevc), 633(hevc)	4	0.9911	**0.9860**	0.41	2	0.9857	**0.9803**	1.04	6	0.9911	**0.9864**	0.37	**0.9855**	0.46
	7	50(h264), 232(h264), 562(h264), 1087(h264), 2153(h264), 70(hevc), 633(hevc)	5	0.9922	**0.9868**	0.33	2	0.9857	**0.9803**	1.04	7	0.9922	**0.9871**	0.30	**0.9863**	0.39
	8	50(h264), 232(h264), 562(h264), 1087(h264), 2153(h264), 50(hevc), 355(hevc), 1126(hevc)	5	0.9922	**0.9868**	0.33	3	0.9896	**0.9846**	0.60	8	0.9922	**0.9873**	0.28	**0.9867**	0.34
Medium	2	327(h264), 283(hevc)	1	0.8990	**0.8691**	9.21	1	0.9154	**0.8924**	7.49	1	0.9154	**0.8924**	7.49	**0.8784**	8.52
	3	167(h264), 836(h264), 283(hevc)	2	0.9431	**0.9182**	4.08	1	0.9154	**0.8924**	7.49	3	0.9431	**0.9287**	3.73	**0.9188**	4.31
	4	114(h264), 489(h264), 1304(h264), 283(hevc)	3	0.9570	**0.9328**	2.56	1	0.9154	**0.8924**	7.49	4	0.9570	**0.9375**	2.82	**0.9301**	3.13
	5	88(h264), 348(h264), 815(h264), 1750(h264), 283(hevc)	4	0.9643	**0.9396**	1.84	1	0.9154	**0.8924**	7.49	4	0.9643	**0.9430**	2.24	**0.9359**	2.53
	6	88(h264), 348(h264), 815(h264), 1750(h264), 139(hevc), 795(hevc)	4	0.9643	**0.9396**	1.84	2	0.9524	**0.9339**	3.19	5	0.9643	**0.9461**	1.92	**0.9410**	2.00
	7	71(h264), 268(h264), 595(h264), 1108(h264), 2149(h264), 139(hevc), 795(hevc)	5	0.9687	**0.9436**	1.43	2	0.9524	**0.9339**	3.19	6	0.9687	**0.9482**	1.71	**0.9440**	1.69
	8	71(h264), 268(h264), 595(h264), 1108(h264), 2149(h264), 93(hevc), 459(hevc), 1275(hevc)	5	0.9687	**0.9436**	1.43	3	0.9636	**0.9456**	1.97	7	0.9687	**0.9511**	1.40	**0.9460**	1.47
Complex	2	469(h264), 417(hevc)	1	0.8157	**0.7614**	15.8	1	0.8358	**0.7912**	13.7	1	0.8358	**0.7912**	13.7	**0.7734**	15.0
	3	265(h264), 1009(h264), 417(hevc)	2	0.8856	**0.8334**	7.89	1	0.8358	**0.7912**	13.7	3	0.8856	**0.8517**	7.08	**0.8346**	8.23
	4	190(h264), 625(h264), 1496(h264), 417(hevc)	3	0.9117	**0.8579**	5.18	1	0.8358	**0.7912**	13.7	4	0.9117	**0.8664**	5.48	**0.8538**	6.13
	5	150(h264), 460(h264), 959(h264), 1950(h264), 417(hevc)	4	0.9260	**0.8703**	3.81	1	0.8358	**0.7912**	13.7	4	0.9260	**0.8760**	4.43	**0.8641**	4.99
	6	150(h264), 460(h264), 959(h264), 1950(h264), 228(hevc), 960(hevc)	4	0.9260	**0.8703**	3.81	2	0.8978	**0.8559**	6.62	6	0.9260	**0.8811**	3.87	**0.8721**	4.11
	7	124(h264), 364(h264), 715(h264), 1246(h264), 2322(h264), 228(hevc), 960(hevc)	5	0.9343	**0.8776**	3.00	2	0.8978	**0.8559**	6.62	7	0.9343	**0.8856**	3.38	**0.8779**	3.48
	8	106(h264), 301(h264), 571(h264), 940(h264), 1497(h264), 2609(h264), 228(hevc), 960(hevc)	6	0.9393	**0.8824**	2.48	2	0.8978	**0.8559**	6.62	7	0.9393	**0.8888**	3.04	**0.8817**	3.06