

PER-TITLE, CONTENT, AND CONTEXT-AWARE ENCODING FOR STREAMING

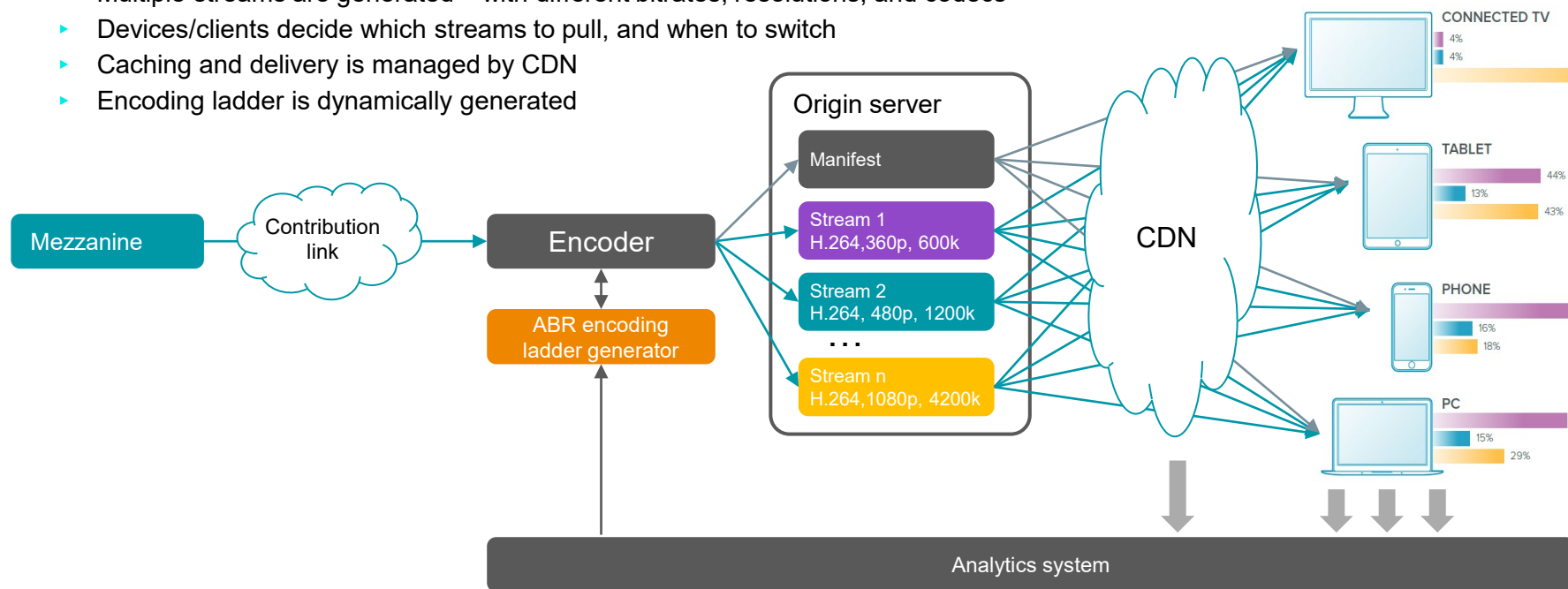
Yuriy Reznik, Brightcove, Inc.

IEEE ICIP Tutorial
Sept. 19, 2021

Adaptive Bitrate (ABR) Streaming

Assumed architecture:

- ▶ DASH and/or HLS streaming systems
- ▶ Multiple streams are generated – with different bitrates, resolutions, and codecs
- ▶ Devices/clients decide which streams to pull, and when to switch
- ▶ Caching and delivery is managed by CDN
- ▶ Encoding ladder is dynamically generated



Encoding Profiles

Example profiles:

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

Source: https://developer.apple.com/documentation/http_live_streaming/hls_authoring_specification_for_apple_devices

Notes on HLS and DASH manifests:

- ▶ HLS: defines single list of all renditions
- ▶ DASH: may include multiple “adaptation sets”
- ▶ DASH: “adaptation-set-switching” properties
- ▶ HLS: may include “SCORE” attributes
- ▶ DASH: may include “qualityRanking” attributes

HLS manifest

```
#EXTM3U
#EXT-X-STREAM-INF:BANDWIDTH=365000,CODECS="avc1.420015",RESOLUTION=480x270,...
Rendition1/Rendition1.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=660000,CODECS="hvc1.1.6.L63.90",RESOLUTION=640x360,...
Rendition2/Rendition2.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=1100000,CODECS="avc1.42001e",RESOLUTION=768x432,...
Rendition3/Rendition3.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=1700000,CODECS="hvc1.1.6.L90.90",RESOLUTION=960x540,...
Rendition4/Rendition4.m3u8
...
```

DASH manifest

```
<MPD xmlns="urn:mpeg:dash:schema:mpd:2011" minBufferTime="PT1.500S" type="static" ... >
<Period duration="PT0H12M14.167S">
<AdaptationSet id="1">
  <SupplementalProperty schemeIdUri="urn:mpeg:dash:adaptation-set-switching:2016" value="2" />
  <Representation id="1" mimeType="video/mp4" codecs="avc1.420015" bandwidth="365000" ... />
  <Representation id="2" mimeType="video/mp4" codecs="hvc1.42001e" bandwidth="1100000" ... />
  ...
</AdaptationSet>
<AdaptationSet id="2">
  <SupplementalProperty schemeIdUri="urn:mpeg:dash:adaptation-set-switching:2016" value="1" />
  <Representation id="3" mimeType="video/mp4" codecs="hvc1.1.6.L63.90" bandwidth="660000" ... />
  <Representation id="4" mimeType="video/mp4" codecs="hvc1.1.6.L90.90" bandwidth="1700000" ... />
  ...
</AdaptationSet>
</Period>
</MPD>
```

Construction of ABR Profiles

Static ABR profiles are suboptimal !

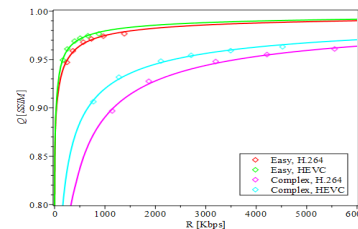
- ▶ Content can be different!
- ▶ Networks can also be different
- ▶ User devices and their usage statistics can be different
- ▶ Codec-capabilities of user devices can be different
- ▶ Many other factors may also be influential...

Most modern systems generate profiles dynamically

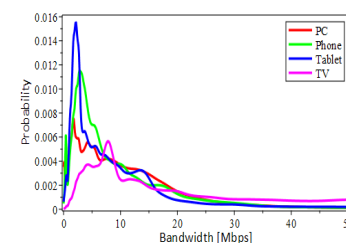
- ▶ “Per-title” techniques – account for differences in content
- ▶ “Network-aware” techniques – account for differences in networks
- ▶ “Context-aware” techniques – account for differences in all aspects

We will review these approaches next...

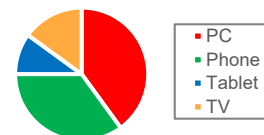
Quality-rate functions



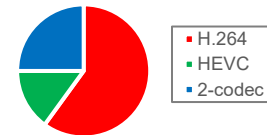
Network bandwidth pdfs



Device usage



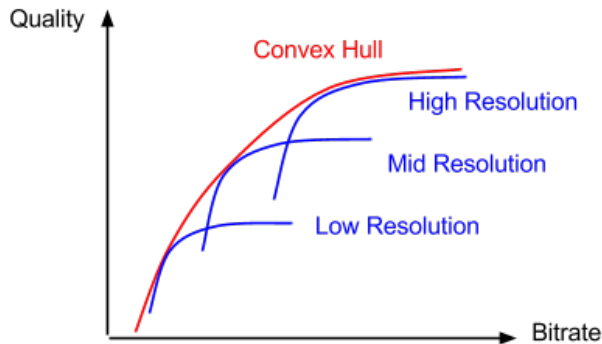
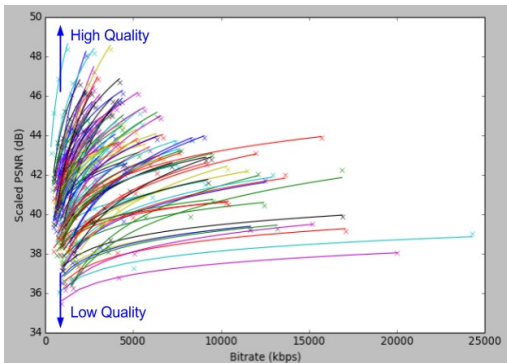
Codec support



Per-Title Encoding

Origin & key ideas:

- ▶ A. Aaron, Z. Li, M. Manohara, J. De Cock, and D. Ronca, “Per-Title Encode Optimization,” 15 Dec. 2015.
<https://medium.com/netflix-techblog/per-title-encodeoptimization-7e99442b62a2>



- ▶ Primary idea: design ABR encoding profiles individually for each video sequence (“title”)
- ▶ Secondary idea: place ladder points such that they belong to the “convex hull”

Notes:

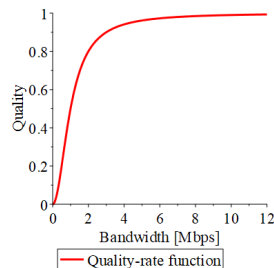
- ▶ The “convex hull” argument suggests a method for finding suitable bitrates for each resolution, but
 - It does not, say how such resolutions should be selected, or how many of them are needed
 - It is not a fully formed optimization problem

Context-Aware Encoding

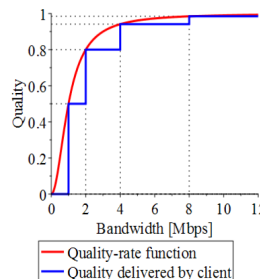
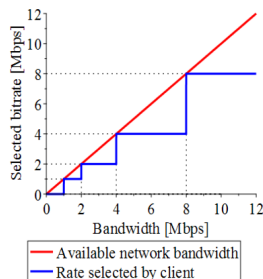
Origin & key ideas:

- ▶ Y. Reznik, et al, "Optimal design of encoding profiles for ABR streaming", SPIE ADIP, August 2017.
- ▶ Input models:

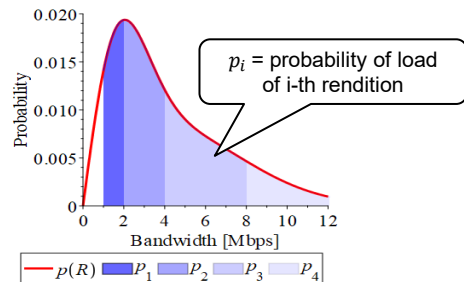
Quality-rate model $Q(R)$



Client model: $R^{selected}(R) \rightarrow Q^{selected}(R)$



Network model $p(R)$



Average bitrate and quality: $\bar{R} = \int_0^{\infty} R^{selected}(R) p(R) dR, \quad \bar{Q} = \int_0^{\infty} Q^{selected}(R) p(R) dR$

Quality-optimal ladder design problem:

- ▶ Find a set of ladder bitrates $\hat{R}_1, \dots, \hat{R}_n$, such that:
- ▶ R_{min}, R_{max} - overall ladder bitrate constraints
- ▶ $R_{1,max}$ - extra constraint on 1st rendition

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

Example Inputs

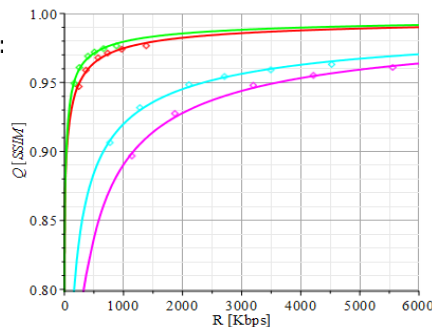
Quality-rate models:

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:

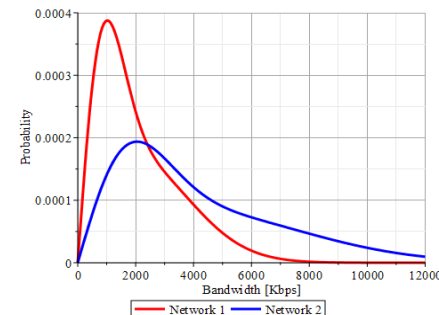
Content	Codec	α	β
Easy	H.264	0.542079	0.483651
Easy	HEVC	0.483928	0.506898
Complex	H.264	20.526129	0.547788
Complex	HEVC	10.666744	0.538406

Network models:

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.

Bandwidth PDFs:



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

Optimal Encoding Ladders

The results:

Network 1, “Easy” content, H.264:

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

Network 1, “Complex” content, H.264:

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

Network 1, “Easy” content, HEVC:

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

Network 1, “Complex” content, HEVC:

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

Metrics:

Q_n - quality of last rendition [SSIM], \bar{Q} - average quality [SSIM], \bar{R} - average bitrate [Kbps]

Impact of CAE in Practice

Efficiency for different types 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

Y. Reznik, et al, "Optimizing Mass-Scale Multiscreen Video Delivery," *SMPTE Motion Imaging Journal*, Apr., 2020

Impact of CAE in Practice

Example efficiency gains with different network / video service operators:

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%

Y. Reznik, et al, "Optimizing Mass-Scale Multiscreen Video Delivery," *SMPTE Motion Imaging Journal*, Apr., 2020

Notes:

- ▶ For operator 1, 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

References

- ▶ A. Aaron, et al, "Per-Title Encode Optimization," Dec. 2015. Netflix tech blog.
- ▶ Y. Reznik, et al, "Optimal Design of Encoding Profiles for ABR Streaming," *SPIE ADIP*, Aug. 2017
- ▶ Y. Reznik, et al, "Optimal Design of Encoding Profiles for ABR Streaming," *Packet Video*, June 2018
- ▶ C. Chen, et al, "Optimized transcoding for large scale adaptive streaming using playback statistics," ICIP 2018.
- ▶ Y. Reznik, et al, "Optimizing Mass-Scale Multiscreen Video Delivery," *SMPTE Motion Imaging Journal*, Apr., 2020
- ▶ A. Katsenou, et al, "Efficient Bitrate Ladder Construction for Content-Optimized Adaptive Video Streaming," arXiv, 2021.
- ▶ P. Lebreton et al, "Network and Content-Dependent Bitrate Ladder Estimation for Adaptive Bitrate Video Streaming," ICASSP 2021

OPTIMAL MULTI-CODEC STREAMING

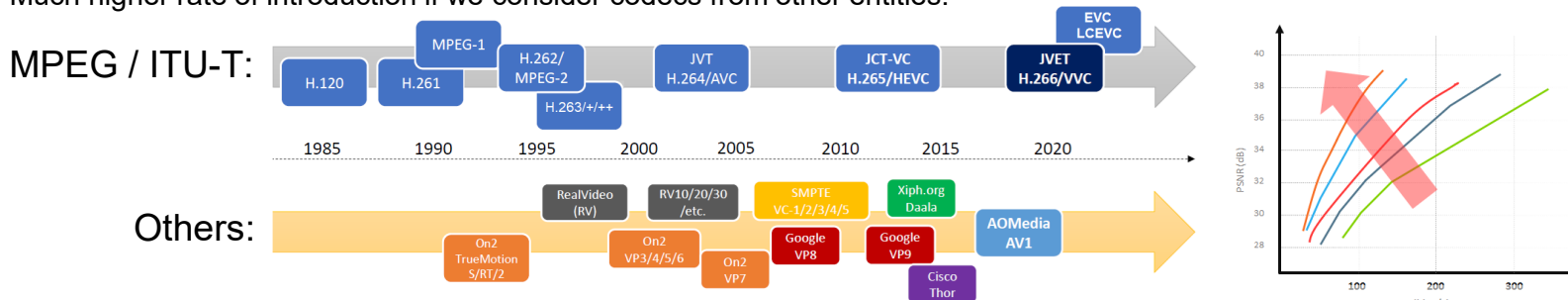
Yuriy Reznik, Brightcove

IEEE ICIP Tutorial
Sept. 19, 2021

Multiple Video Codecs

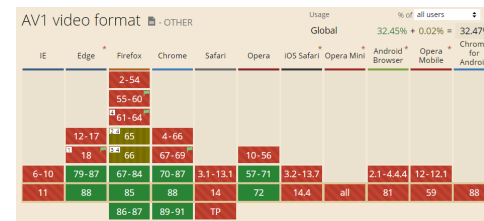
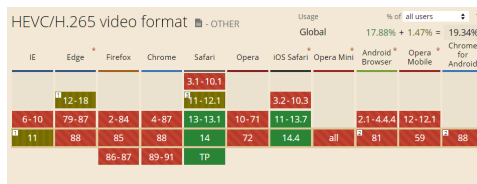
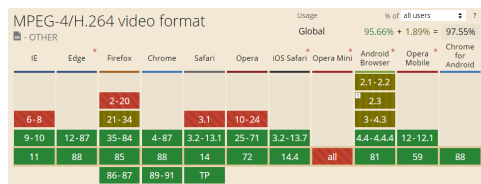
Since 1980s, new MPEG/ITU-T video codecs have been introduced every 3-8 years:

- ▶ Much higher rate of introduction if we consider codecs from other entities:



But only few become ubiquitous:

- ▶ MPEG-2 and H.264/AVC are the only ones that become almost universally adopted
- ▶ Most newer codecs, including HEVC and AV1 have smaller and fragmented support across devices/platforms:



If the goal is to reach all devices today – we have to use multiple codecs!

Multi-Codec Encoding Profiles

Example profiles:

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

Source: https://developer.apple.com/documentation/http_live_streaming/hls_authoring_specification_for_apple_devices

Notes on multi-codec profiles:

- ▶ HLS: mixed-codec renditions are in the same list
- ▶ DASH: separate “adaptation sets” for each codec
- ▶ DASH: “adaptation-set-switching” properties
- ▶ HLS: “SCORE” attributes
- ▶ DASH: “qualityRanking” attributes

HLS manifest

```
#EXTM3U
#EXT-X-STREAM-INF:BANDWIDTH=365000,CODECS="avc1.420015",RESOLUTION=480x270,...
Rendition1/Rendition1.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=660000,CODECS="hvc1.1.6.L63.90",RESOLUTION=640x360,...
Rendition2/Rendition2.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=1100000,CODECS="avc1.42001e",RESOLUTION=768x432,...
Rendition3/Rendition3.m3u8
#EXT-X-STREAM-INF:BANDWIDTH=1700000,CODECS="hvc1.1.6.L90.90",RESOLUTION=960x540,...
Rendition4/Rendition4.m3u8
...
```

DASH manifest

```
<MPD xmlns="urn:mpeg:dash:schema:mpd:2011" minBufferTime="PT1.500S" type="static" ... >
<Period duration="PT0H12M14.167S">
<AdaptationSet id="1">
  <SupplementalProperty schemeIdUri="urn:mpeg:dash:adaptation-set-switching:2016" value="2" />
  <Representation id="1" mimeType="video/mp4" codecs="avc1.420015" bandwidth="365000" ... />
  <Representation id="2" mimeType="video/mp4" codecs="hvc1.42001e" bandwidth="1100000" ... />
  ...
</AdaptationSet>
<AdaptationSet id="2">
  <SupplementalProperty schemeIdUri="urn:mpeg:dash:adaptation-set-switching:2016" value="1" />
  <Representation id="3" mimeType="video/mp4" codecs="hvc1.1.6.L63.90" bandwidth="660000" ... />
  <Representation id="4" mimeType="video/mp4" codecs="hvc1.1.6.L90.90" bandwidth="1700000" ... />
  ...
</AdaptationSet>
</Period>
</MPD>
```

Related Features of MPEG DASH

Key DASH elements:

- ▶ “qualityRanking” – defines relative scores within each adaptation set
- ▶ “qr-equivalence” – expands the scope of applicability of qualityRanking attributes to multiple adaptation sets
- ▶ “adaptation-set-switching” – enables switching between adaptation sets

Example encoding profiles:

- ▶ AVC adaptation set:

#	Codec	Resolution	Frame-rate	Bitrate [Kbps]	Quality [MOS]	Quality rank
1	AVC	384x216	25	261	2.18	10
2	AVC	512x288	25	513	2.72	8
3	AVC	768x432	25	1024	3.41	6
4	AVC	1280x720	25	2075	4.21	4
5	AVC	1920x1080	25	4203	4.77	2

- ▶ HEVC adaptation set:

#	Codec	Resolution	Frame-rate	Bitrate [Kbps]	Quality [MOS]	Quality rank
1	HEVC	512x288	25	300	2.53	9
2	HEVC	768x432	25	607	3.26	7
3	HEVC	1024x576	25	1166	3.79	5
4	HEVC	1600x900	25	2362	4.55	3
5	HEVC	1920x1080	25	4203	4.91	1

DASH manifest:

```
<MPD xmlns="urn:mpeg:dash:schema:mpd:2011" minBufferTime="PT1.500S" type="static" ... >
<Period duration="PT0H12M14.167S">
<SupplementalProperty schemeIdUri="urn:mpeg:dash:qr-equivalence:2019" value="1,2" />
<AdaptationSet id="1">
  <SupplementalProperty schemeIdUri="urn:mpeg:dash:adaptation-set-switching:2016" value="2" />
  <Representation id="1" mimeType="video/mp4" codecs=" avc1.420015" bandwidth="261000" qualityRanking="10" .../>
  <Representation id="2" mimeType="video/mp4" codecs=" avc1.42001e" bandwidth="513000" qualityRanking="8" .../>
  <Representation id="3" mimeType="video/mp4" codecs=" avc1.42001e" bandwidth="1024000" qualityRanking="6" .../>
  <Representation id="4" mimeType="video/mp4" codecs=" avc1.42001e" bandwidth="2075000" qualityRanking="4" .../>
  <Representation id="5" mimeType="video/mp4" codecs=" avc1.42001e" bandwidth="4203000" qualityRanking="2" .../>
</AdaptationSet>
<AdaptationSet id="2">
  <SupplementalProperty schemeIdUri="urn:mpeg:dash:adaptation-set-switching:2016" value="1" />
  <Representation id="1" mimeType="video/mp4" codecs="hvc1.1.6.L63.90" bandwidth="300000" qualityRanking="9" .../>
  <Representation id="2" mimeType="video/mp4" codecs="hvc1.1.6.L90.90" bandwidth="607000" qualityRanking="7" .../>
  <Representation id="3" mimeType="video/mp4" codecs="hvc1.1.6.L90.90" bandwidth="1166000" qualityRanking="5" .../>
  <Representation id="4" mimeType="video/mp4" codecs="hvc1.1.6.L90.90" bandwidth="2362000" qualityRanking="3" .../>
  <Representation id="5" mimeType="video/mp4" codecs="hvc1.1.6.L90.90" bandwidth="4203000" qualityRanking="1" .../>
</AdaptationSet>
</Period>
</MPD>
```

System with 2 Codecs and 3 Clients

Codecs:

- ▶ H.264
- ▶ HEVC

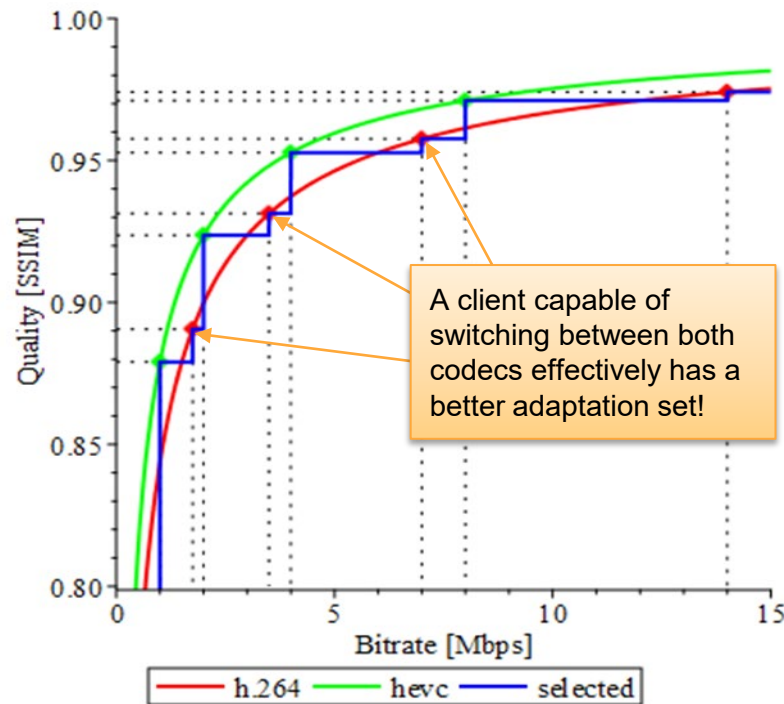
Clients:

- ▶ H.264 client – can only decode H264 streams
- ▶ HEVC client – can only decode HEVC streams
- ▶ 2-codec client – can decode and switch between both

Key observation:

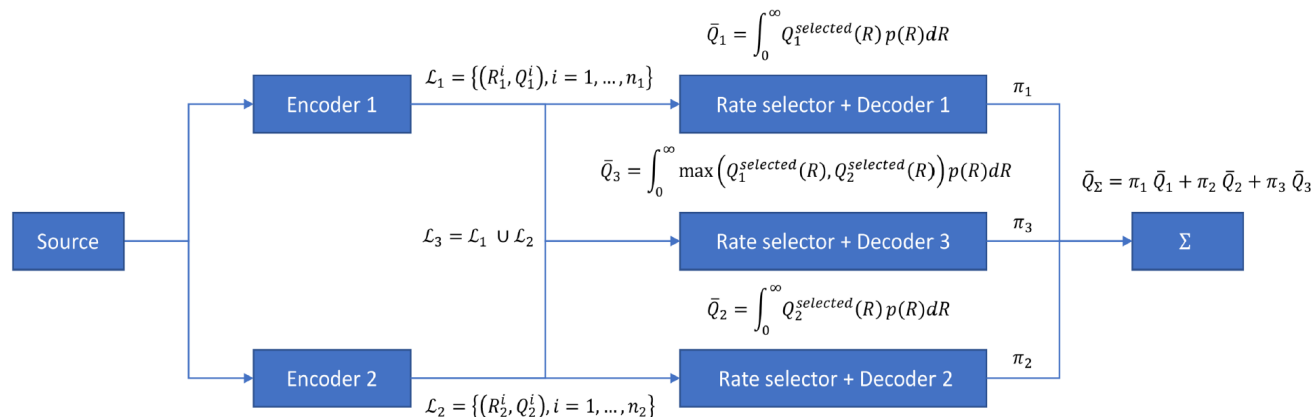
- ▶ 2-codec client can achieve better performance (finer-grain adaptation to network) than single codec-ones!!!
- ▶ This requires special design of a 2-codec ladder:
 - codec bitrates should form interleaved pattern,
 - quality of all steps should increase monotonically,
 - steps should be meaningful in size,
 - etc.

2-codec client switching behavior:



2-Codec Ladder Design Problem

Average quality in system with 2 codecs and 3 types of clients:



- ▶ \mathcal{L}_1 : H.264 ladder with n_1 points
- ▶ \mathcal{L}_2 : HEVC ladder with n_2 points
- ▶ \mathcal{L}_3 : mixed ladder with $n = n_1 + n_2$ points
- ▶ \bar{Q}_1 : average quality of H.264 client
- ▶ \bar{Q}_2 : average quality of HEVC client
- ▶ \bar{Q}_3 : average quality of 2-codec client
- ▶ \bar{Q}_Σ : overall average quality achieved
- ▶ π_1, π_2, π_3 : usage probabilities of clients of each kind (H.264, HEVC, 2-codec)

Quality optimal ladder design 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 the overall quality \bar{Q}_Σ is maximal:

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

A Related Problem

Multiple-description coding:

- ▶ It might be refreshing to note, that from mathematical perspective, the described problem is nothing new
- ▶ It generally belongs to what is known in information theory as “**multiple description coding**” problem, and where specific example of a system with 2 encoders and 3 receivers was studied by L. Ozarow in 1980:

Copyright © 1980 American Telephone and Telegraph Company
THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 59, No. 10, December 1980
Printed in U.S.A.

On a Source-Coding Problem with Two Channels and Three Receivers

By L. OZAROW

(Manuscript received May 15, 1980)

This paper treats the problem of communicating a memoryless unit-variance Gaussian source to three receivers. Two channels are available, each with a separate receiver. A third receiver has the outputs of both channels available. We obtain an expression for the simultaneously achievable distortions (mean-squared error). This

A natural problem for the network of Fig. 1 is to characterize the set

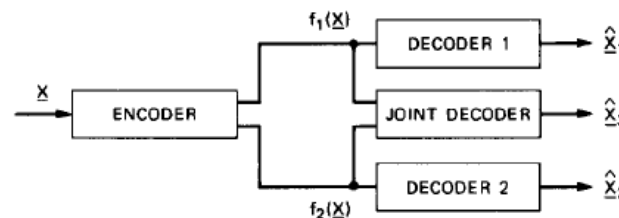


Fig. 1—The channel-splitting problem.

1910 THE BELL SYSTEM TECHNICAL JOURNAL, DECEMBER 1980

- ▶ The practical difference, however, is that information theory is mostly concerned with finding “regions of achievability” of R/D tradeoffs, whereas in our case, we are trying to find maximum of a mix of all outputs.
- ▶ Another difference is that in our case, this coding problem is applied as tool for **encoding of the network bandwidth** (treated as an information source), and not for encoding of media sent over network. Encoding profiles = codebooks for coding of bandwidth.

Optimal 2-Codec Ladders

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}	\bar{R}
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

Vs. optimal single-codec ladders:

H.264 ladder:

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

Metrics:

- ▶ Q_n - quality of last rendition [SSIM], \bar{Q} - average quality [SSIM], \bar{R} - average bitrate [Kbps]

Observations:

- if $n=2..5$ a 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 the same quality as 8-stream H.264 + 3-4-stream HEVC ladders constructed separately: 36-42% savings in the number of streams!

HEVC ladder:

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 Ladders: “Complex” Content

Optimal 2-codec ladders:

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}	\bar{R}
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=2 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: 50% reduction in number of streams!

Vs. optimal single-codec ladders:

H.264 ladder:

n	Ladder Bitrates [kbps]	Q_n	\bar{Q}	\bar{R}
2	210, 946	0.8971	0.8598	773.7
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

HEVC ladder:

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

Metrics:

- ▶ Q_n - quality of last rendition [SSIM], \bar{Q} - average quality [SSIM], \bar{R} - average bitrate [Kbps]

Discussion

Using the proposed Multi-Codec profile generation technique we can:

- ▶ Achieve **up to 50% reduction in the number of streams** relative to the brute force mix of ladders generated for each codec
 - this makes the whole concept of multi-codec streaming much more practical!

Consequences:

- ▶ Fewer streams = lower transcoding costs
- ▶ Fewer streams = lower storage costs
- ▶ Fewer streams = lower CDN cache miss probability (lower chances of buffering)
- ▶ Fewer streams = less traffic on origins (lower compute / auto-scaling costs, cloud egress, etc.)
- ▶ etc.

Caveats:

- ▶ The platforms currently supporting switching between codecs are
 - most new Apple devices – can switch between H.264 and HEVC streams
 - Chrome v70+ and Firefox v63+ support MSE `changeType()` method – enabling codec switching in JS-based web clients
- ▶ But there are many others which will only be able to pull either H.264 or HEVC streams
- ▶ To practice this method efficiently – understanding of analytics and distribution of devices of each kind is critical!

References

- ▶ Y. Reznik, et al, "Optimal Design of Encoding Profiles for ABR Streaming," *Packet Video*, June 2018
- ▶ Y. Reznik, et al, "Optimal Design of Multi-codec Profiles for ABR Streaming," *ICME*, July 2019
- ▶ Y. Reznik, et al, "Optimizing Mass-Scale Multiscreen Video Delivery," *SMPTE Motion Imaging Journal*, Apr., 2020

OPTIMAL MULTI-SCREEN STREAMING

Yuriy Reznik, Brightcove

IEEE ICIP Tutorial
Sept. 19, 2021

Multi-screen

What is it?

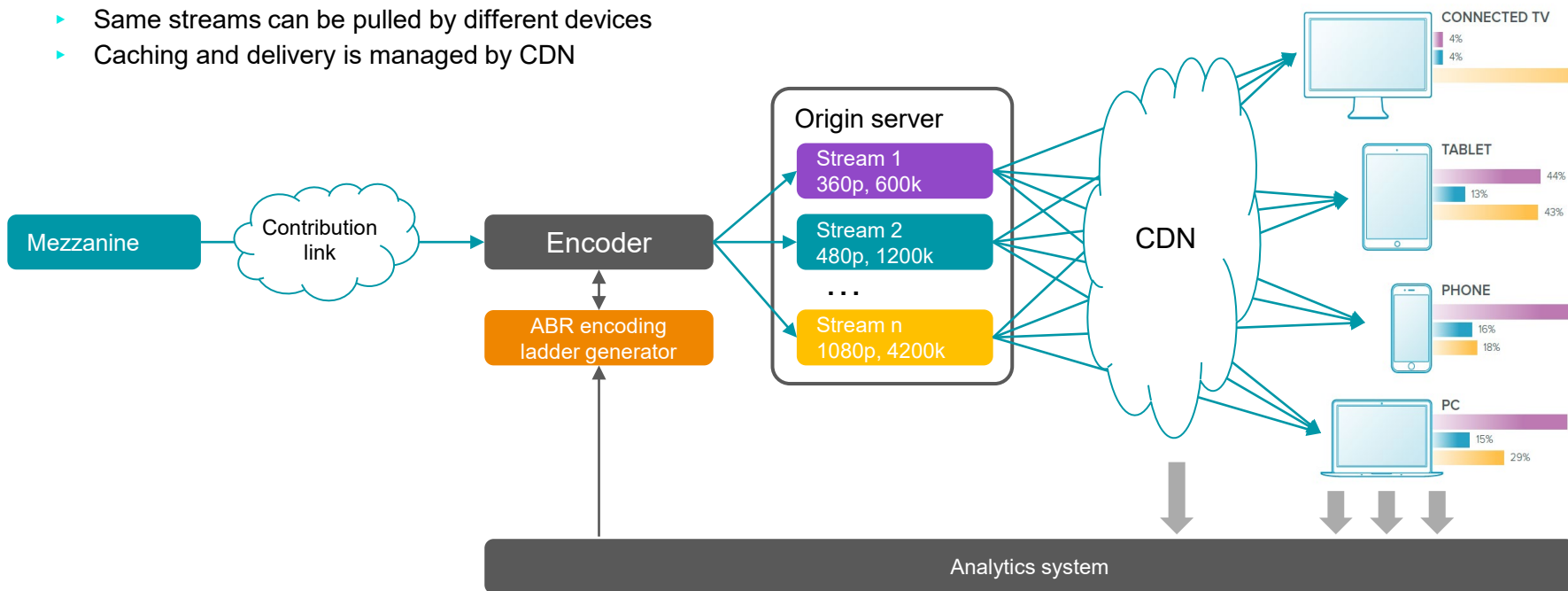
- ▶ Simultaneous delivery of video to multiple devices:
 - PCs / Laptops
 - Mobiles
 - Tablets
 - Connected TVs, etc.
- ▶ More narrowly, it can be also understood as OTT delivery to all devices, utilizing ABR streaming protocols such as DASH or HLS
- ▶ Streaming of videos embedded in web-pages may also be considered a part of (or form of) multi-screen



Multi-screen Delivery System

Assumed architecture

- ▶ Video encoded in HLS and/or DASH streams
- ▶ Encoding profiles are custom generated for each input video & delivery context
- ▶ Same streams can be pulled by different devices
- ▶ Caching and delivery is managed by CDN



Challenges

Diversity of device capabilities and parameters

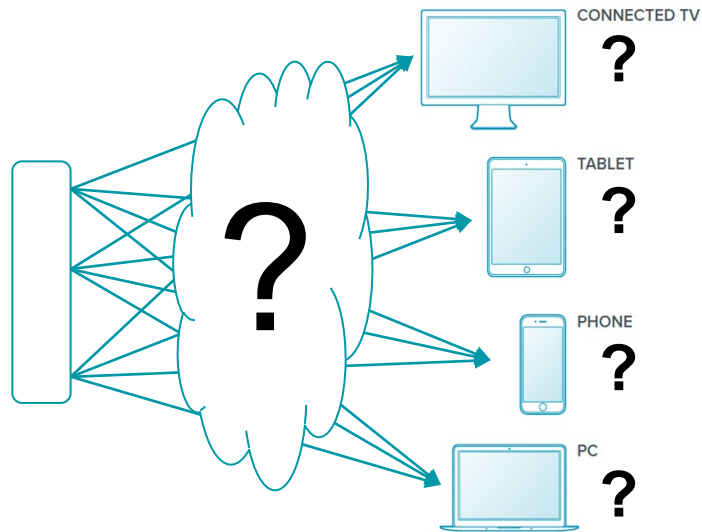
- ▶ Different device capabilities: codecs, DRM and formats support
- ▶ Different device form factors, user positions & viewing preferences
- ▶ Different networks connecting devices

Additional effects

- ▶ Perception of video on different devices is different
- ▶ Different devices may pull different subsets of streams
- ▶ Distribution of content usage across different devices may be different

Key questions:

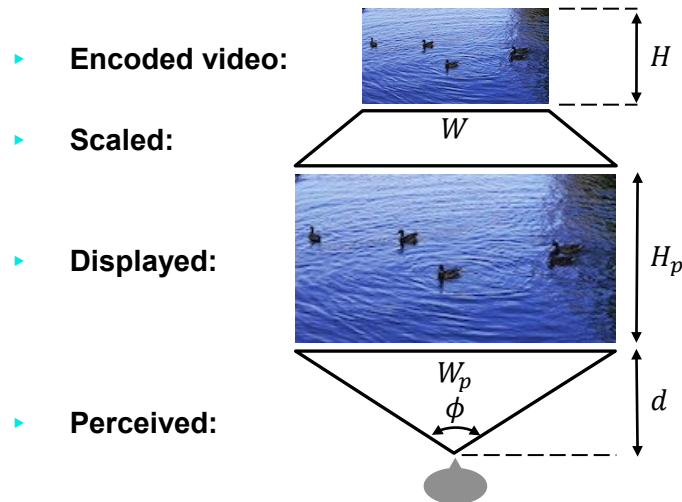
- ▶ How to model differences in perceived quality on different devices?
- ▶ How to model components of this system (networks, clients, etc.)?
- ▶ How to characterize performance of the entire system?
- ▶ How to optimize it?



Perceived Video Quality on Different Devices

Angular Metrics

Video reproduction chain



Main parameters involved

Parameters	Meaning	Unit
W, H	encoded video width, height	pixels
W_p, H_p	display/player width, height	pixels
d	viewing distance	inches
ρ	display pixel density	dots per inch
$\phi = 2 \arctan\left(\frac{W_p}{2d\rho}\right)$	viewing angle	degrees
$\phi_c = 2 \arctan\left(\frac{W_p/W}{d\rho}\right)$	angle to 2 pixels (1 cycle)	degrees
$u = \frac{1}{\phi_c}$	angular resolution of video	cycles per degree (cpd)

Relevant for human perception

- viewing angle ϕ** → angular span of video frame, as visible on screen
- angular resolution u** → inverse of angular span of 2 pixels (length of smallest "cycle") in encoded video

Note: Another way to describe angular resolution is to say that it is a Nyquist frequency of video, expressed in angular units, reflecting projection the screen.

Differences in Viewing Setups

Viewing setup parameters

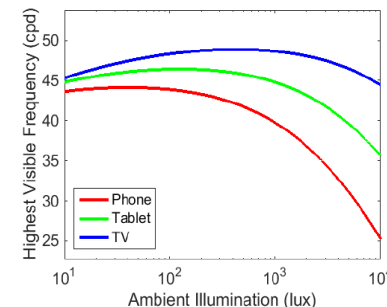
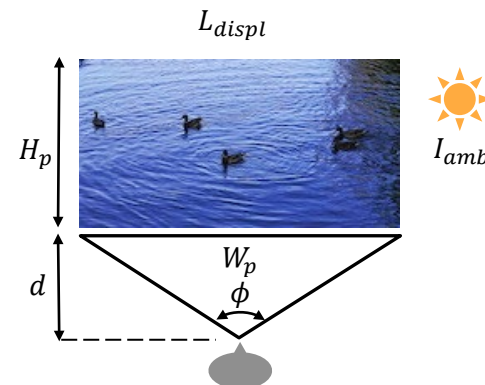
Device	Viewing distance	Display size	Brightness	Ambient	Background
TV	2-6H, med=3H	32-80", med=50"	400 nits	50-200 lux	10-30% reflective
PC / laptop	12-30", med=24"	13-36", med=20"	200 nits	100-500 lux	Varies
Tablet	10-22", med=18"	7-12", med=9"	200-400 nits	50-500 lux	Varies
Phone	7.5-22", med=14"	4-6", med=5.5"	100-300 nits	10-10000 lux	Varies

Angular metrics

Device	Viewing angle	Angular resolutions when rendering video full screen					Max. visible resolution*
		360p	540p	720p	1080p	4k	
TV	33.0 degrees	9.4 cpd	14.3 cpd	18.9 cpd	28.3 cpd	56.5 cpd	48.3 cpd
PC/Laptop	40+ degrees	7.7 cpd	11.5 cpd	15.4 cpd	23.1 cpd	46.1 cpd	46.2 cpd
Tablet	24.6 degrees	12.8 cpd	19.2 cpd	25.6 cpd	38.4 cpd	76.9 cpd	46.2 cpd
Phone	18.2 degrees	17.9 cpd	26.9 cpd	35.9 cpd	53.8 cpd	107.6 cpd	44.1 cpd

Observations

- ▶ Viewing angles and angular resolutions are very different on different devices!
- ▶ With high-resolution content and small form-factor devices, angular resolutions may exceed maximum resolutions visible by human eye!



(*) L. Kerofsky, R. Vanam and Y. Reznik, "Adapting Objective Video Quality Metrics to Ambient Lighting," QOMEX 2015.

Perceived Quality

Westerink & Roufs experiments (1989)*

- ▶ Controlled environment, 20 subjects, 5 images, 0-10 categorical scale
- ▶ Varied: viewing distance, resolution, and picture size

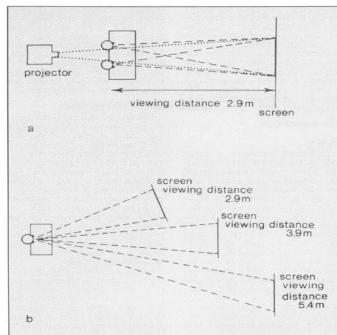


Figure 1. Experimental configurations: (a) Experiment 1; (b) Experiment 2.

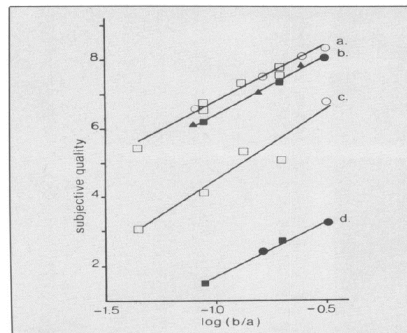


Figure 4. Subjective image quality as a function of the picture angle. The subjective quality values on the vertical axis are plotted as a function of $\log(b/a)$, which differs from the picture angle ϕ only by a constant. The different sets correspond to different resolutions: (a) greater than $35 \sim/\phi$, (b) between 28 and $35 \sim/\phi$, (c) between 8.6 and $6.7 \sim/\phi$, and (d) between 2.6 and $2.7 \sim/\phi$. Different symbols represent different viewing distances: $\square = 2.9$ m, $\Delta = 3.9$ m, $\square = 5.4$ m. Every point is the result of 60 judgments.

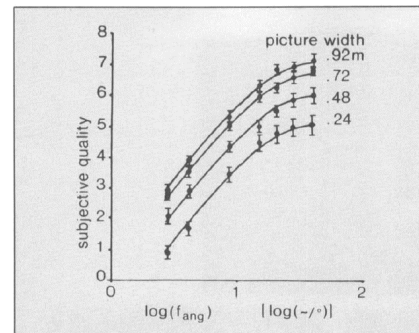


Figure 3. Subjective quality as a function of resolution. Every point is the result of 100 judgments, and the error indicated is plus or minus the standard error of the mean.

(*) J. Westerink and J. Roufs, "Subjective image quality as a function of viewing distance resolution and picture size," SMPTE Journal, vol. 98, 1989, pp. 113-19.

(**) P. G. J. Barten, "Effect of picture size and definition on perceived image quality," IEEE Trans. Electron. Devices, vol. 36, no. 9, pp. 1865-1869, Sept. 1989.

Observed phenomena:

- ▶ Perceived quality grows approximately as logarithm of viewing angle (ϕ)
- ▶ Perceived quality also grows with angular resolution (u), but saturates at around 25-40 cycles/degree

Model describing these effects*

$$Q_{WR}(\phi, u) = 3.6 \log(\phi) + 2.9 + 4.6 \log(u) + 2.7 \log(u)^2 - 1.7 \log(u)^3$$

Scaling and Perceived Quality

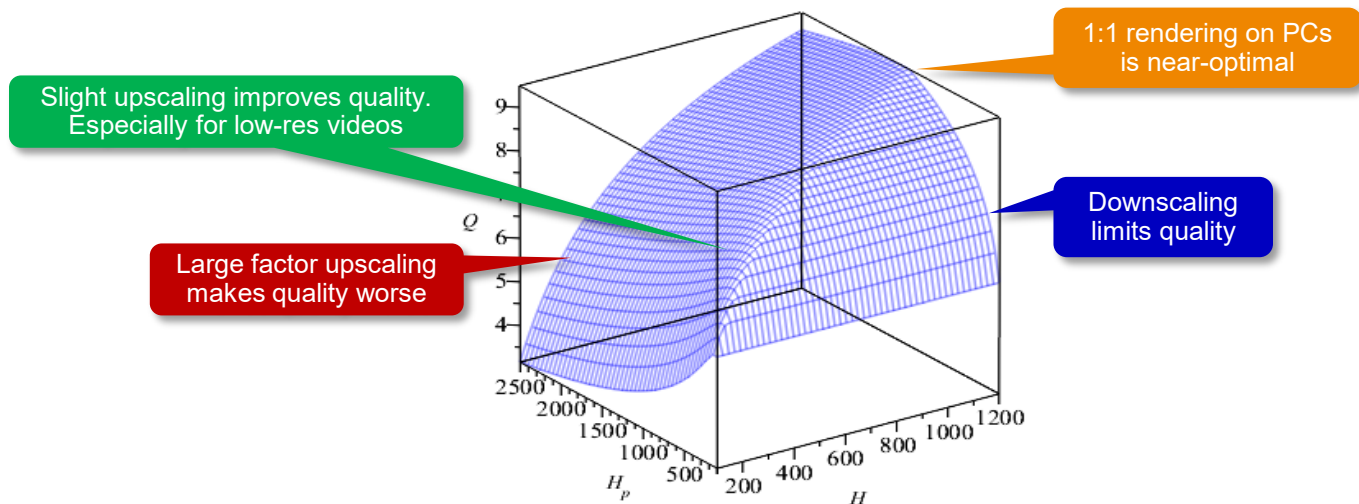
Westerink-Roufs model

$$Q_{WR}(H, H_p) = Q_{WR} \left(2 \arctan \left(\frac{H_p \cdot DAR}{2d\rho} \right), \left(2 \arctan \left(\frac{H_p}{\min(H, H_p) d\rho} \right) \right)^{-1} \right)$$

- ▶ H – video height, H_p – player height, $DAR = W_p/H_p$
- ▶ d – viewing distance, ρ – pixel density, $Q_{WR}(\phi, u)$ – original WR model

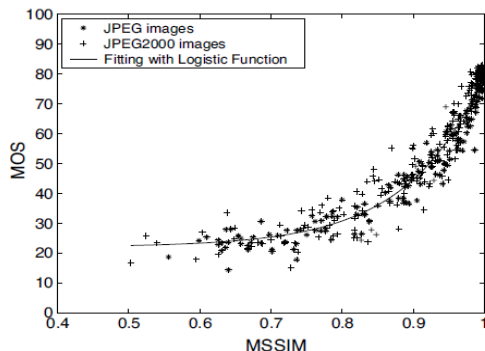
Effects of scaling:

WR model for PC viewing ($d=24, p=96$)

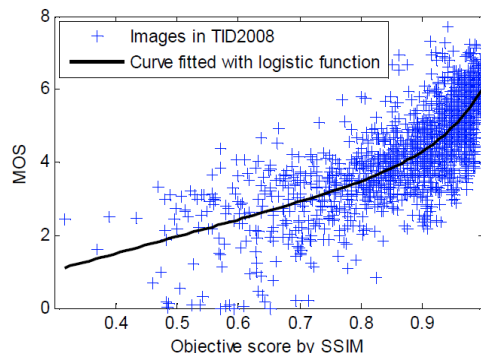


Codec Noise and Perceived Quality

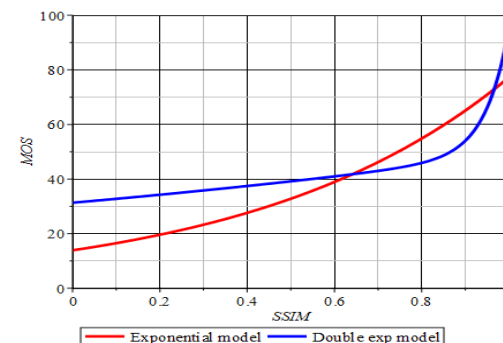
For SSIM* metric this relationship is well studied



(*) Z. Wang, A. Bovik, H. Sheikh, E. Simoncelli, "Image quality assessment: from error visibility to structural similarity". IEEE Transactions on Image Processing 13 (4) (2004).



L. Zhang, L. Zhang, X. Mou, D. Zhang, "FSIM: a feature similarity index for image quality assessment", IEEE Trans Image Process. 20 (8) (2011)



U. Engelke, M. Kusuma, H-J. Zepernick, M. Caldera, "Reduced-reference metric design for objective perceptual quality assessment in wireless imaging". Signal Processing - Image Communication, 24 (7) (2009).

Common SSIM to MOS mapping functions

Polynomial	Exponential	Logistic
$p_1x + p_0$	$a_1e^{b_1x}$	$100/[1 + e^{-l_1(x-l_2)}]$
$p_2x^2 + p_1x + p_0$	$a_1e^{b_1x} + a_2e^{b_2x}$	

Parametric Quality Model

Simple combined model

$$Q_{\Sigma}(H, H_p, D) = \alpha \left(\beta + Q_{WR}(H, H_p) \right) e^{\gamma D}$$

- ▶ H – encoded video height, H_p – player height [pixels]
- ▶ D – codec noise [SSIM], measured at encoded video resolution
- ▶ $Q_{WR}(H, H_p)$ – Westerink-Roufs model, assuming d, ρ as specific to the device
- ▶ α, β, γ – calibration parameters

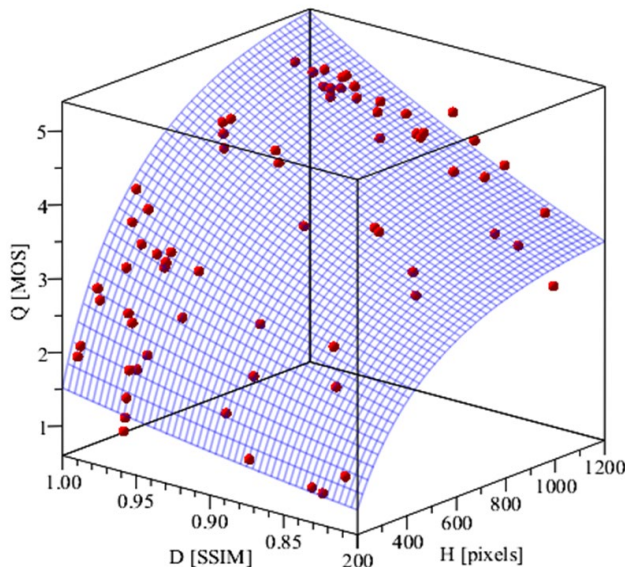
Accuracy w.r.t. Netflix dataset*

- ▶ Using parameters: $\alpha = 0.1075$, $\beta = -4.859$, $\gamma = 2.424467$
- ▶ Also assuming $\phi = 33$ [°] to match MOS test conditions used in this set
- ▶ The resulting **RMSE ~ 0.329** on 1-5 MOS scale.

Why such model is needed?

- ▶ To predict quality when video is played by different devices or players with different window sizes
- ▶ Same video on different devices / players sizes will look different!
- ▶ There is no such thing as single "universal" quality value!

Model vs MOS scores in Netflix dataset*



(*) Z. Li, A. Aaron, et al., Toward A Practical Perceptual Video Quality Metric, June 2016.
<https://netflixtechblog.com/toward-a-practical-perceptual-video-quality-metric-653f208b9652>

Compression Performance

Distortion-Rate Model

"Empirical" SSIM-rate model

$$D(H, R) = \left(1 + \left(\frac{R}{\alpha H^\beta} \right)^{-\gamma} \right)^{-\frac{1}{\gamma}}$$

- ▶ H – resolution at which video is encoded
- ▶ R – bitrate at which video is encoded
- ▶ α, β, γ – model parameters

Fitting process

- ▶ Run several "probe" encodings, covering typical / expected operating range:
 $(H_i, R_i) \in [H_{\min}, H_{\max}] \times [R_{\min}, R_{\max}]$,
- ▶ Compute SSIM values at all points
- ▶ Find parameters α, β, γ providing best fit

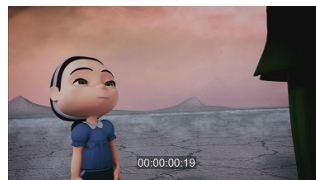
Example applications

- ▶ x264 encoder used with CRF=16,...,36
- ▶ Main profile, Level 4, 2sec GOPs
- ▶ Resolutions: $H=[270, 360, 432, \dots, 1080]$
- ▶ 3 sequences



Models for 3 video sequences

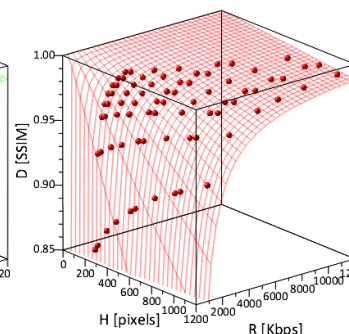
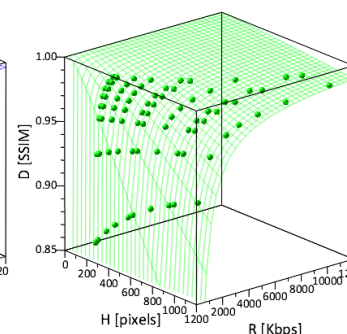
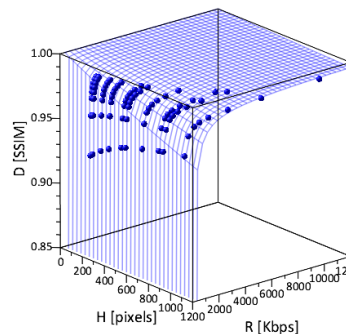
"Easy"



"Medium"



"Complex"



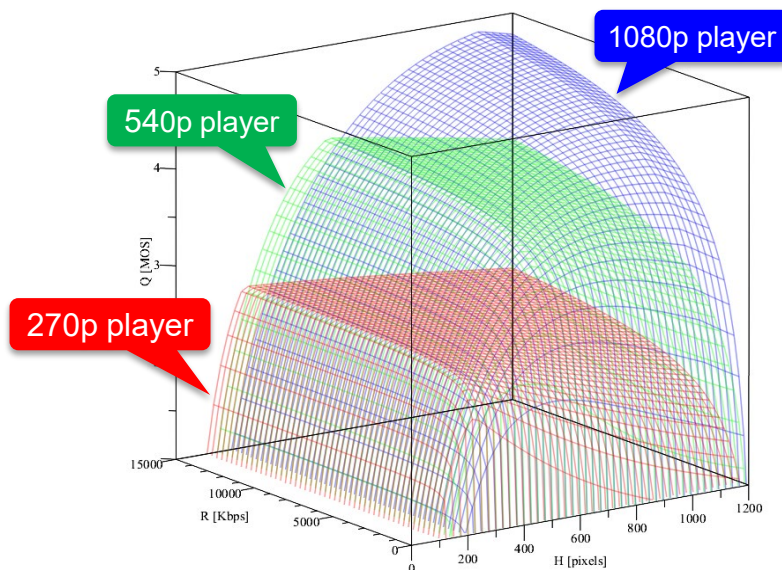
Sequence	α	β	γ	RMSE
"Easy"	0.7844e-3	1.2281	0.7463	0.3404e-2
"Medium"	0.8278e-2	1.3217	0.9593	0.2792e-2
"Complex"	0.07316	1.0957	1.0336	0.1153e-2

Quality-Rate Model

Translation in quality domain

$$Q(H, H_p, R) = Q_{\Sigma}(H, H_p, D(H, R))$$

Example: "Complex" sequence, x264, PC screen:



Dimensions

- ▶ Q – predicted quality [MOS]
- ▶ R – bitrate at which video is encoded
- ▶ H – resolution at which video is encoded
- ▶ H_p – resolution at which video is displayed
- ▶ d, ρ – device-specific parameters

Notes

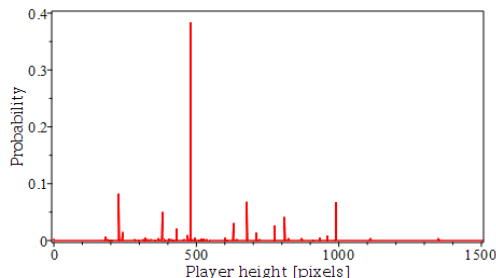
- ▶ This model is specific for a given:
 - content, codec, and
 - parameters of user device
- ▶ With different player sizes we see that shapes of $Q(H, R)$ surfaces are changing significantly
 - there is no "tight nesting" of them
- ▶ All parameters in this model are important!

Behavior of Streaming Clients

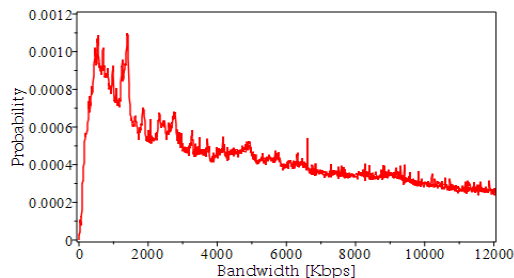
Example Playback Statistics

US Open event, June 13, 2019

Distribution of video player sizes



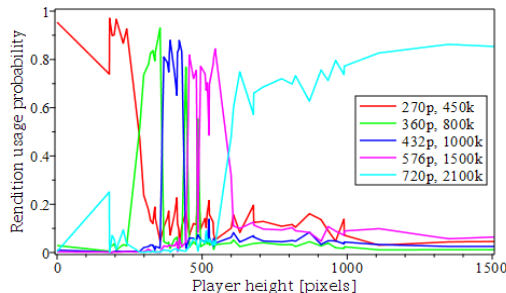
Network bandwidth distribution



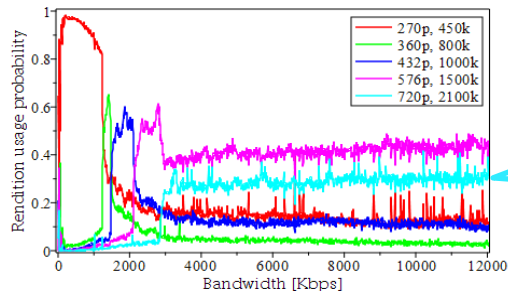
Encoding ladder used

Stream	Profile	Resolution	Framerate	Bitrate
1	Baseline	480x270	23.976	450k
2	Baseline	640x360	23.976	800k
3	Main	768x432	23.976	1000k
4	Main	1024x576	23.976	1500k
5	Main	1280x720	23.976	2100k

Adaptation to player sizes



Adaptation to network bandwidth



Usage of top bitrate rendition does not increase even when bandwidth is no longer a limit !!!

Observations

- ▶ Player sizes apparently impact selection of the streams! This happens not only with web players, but also with mobile apps.
- ▶ Bandwidth-adaptation is no longer the only dimension of the problem!

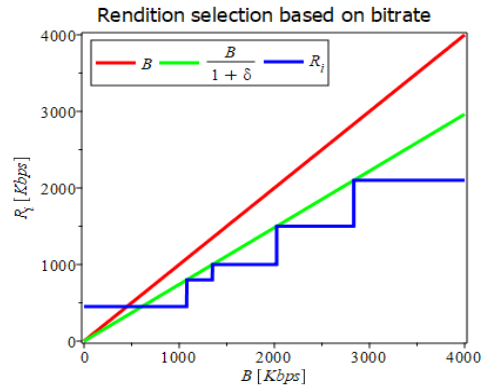
Building a Client Model

Adaptation to bandwidth

$$i_B(B) = \begin{cases} 1 & \text{if } B < T_1^B \\ i & \text{if } T_i^B \leq B < T_{i+1}^B, i=2, \dots, n-2, \\ n & \text{if } B \geq T_{n-1}^B \end{cases}$$

where:

- ▶ B – network bandwidth
- ▶ i_B – index of selected rendition
- ▶ $R_1 < \dots < R_n$ – ladder bitrates
- ▶ $T_i^B = (1 + \delta)R_{i+1}$ – thresholds
- ▶ $\delta \geq 0$ – client-specific constant

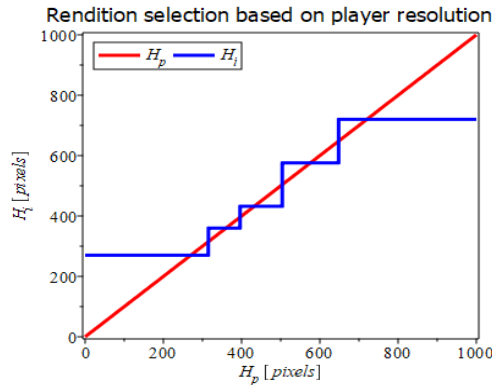


Adaptation to player size

$$i_H(H_p) = \begin{cases} 1 & \text{if } H_p < T_1^H \\ i & \text{if } T_i^H \leq H_p < T_{i+1}^H, i=2, \dots, n-2, \\ n & \text{if } H_p \geq T_{n-1}^H \end{cases}$$

where:

- ▶ H_p – player height
- ▶ i_H – index of selected rendition
- ▶ $H_1 < \dots < H_n$ – ladder resolutions
- ▶ $T_i^H = \alpha H_i + (1 - \alpha)H_{i+1}$ – thresholds
- ▶ $\alpha \in [0,1]$ – client-specific constant

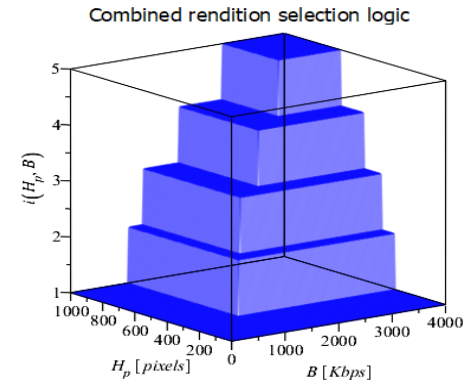


Combined model

$$i(B, H_p) = \min\{i_B(B), i_H(H_p)\}$$

where:

- ▶ i_B – selection based on bitrate
- ▶ i_H – selection based on player size
- ▶ assumed order:
 $R_1 < \dots < R_n$ and $H_1 < \dots < H_n$



Client Model vs Real World

Usage probability of i-th rendition

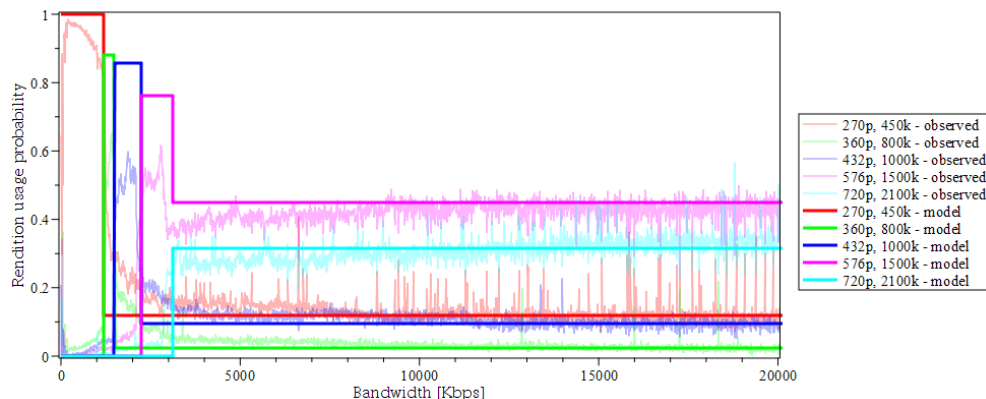
$$p(i | B) = \sum_{H_p: i(B, H_p) = i} p(H_p)$$

- ▶ B – network bandwidth
- ▶ H_p – player size
- ▶ $i(B, H_p)$ – client model
- ▶ $p(H_p)$ – distribution of player sizes

Model-computed vs reported data

Best fit achieved with:

- ▶ $\alpha \approx 0.88$
- ▶ $\delta \approx 0.49$
- ▶ $RMSE \approx 0.05$



Notes

- ▶ The model works reasonably well. Explains behavior in high-bandwidth regime.
- ▶ With different devices/clients/web-pages, client model parameters α , δ , as well as distributions $p(H_p)$ maybe different
- ▶ Some clients may only adapt to bandwidth. They can be modeled as $i(B, H_p) = i_B(B)$.

Average Performance of Multi-Screen Streaming Systems

Input Distributions

Devices

- ▶ We will assume that delivery is done to a *set of devices* Ω , and that relative consumption of content by these devices can be modeled by a discrete random variable $\omega \in \Omega$, with probability mass function $p(\omega)$
- ▶ Display and client model parameters $(d, \rho, \alpha, \delta)$ will be specific to each device. We will use sub-index ω to denote this. E.g. d_ω, ρ_ω , etc.

Networks

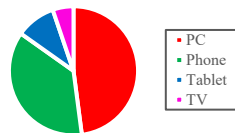
- ▶ *Network bandwidth* B will be treated as a continuous random variable, defined over $[0, \infty)$, and with probability density function $q(B)$
- ▶ Network density specific to a particular device will be denoted by $q_\omega(B)$

Player sizes

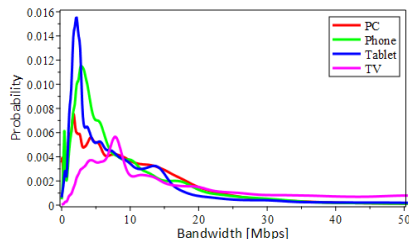
- ▶ Video player size H_p will also be treated as a discrete random variable, producing values from a set \mathcal{H}_p , with probability mass function $r(H_p)$
- ▶ Player size distribution specific to a particular device will be denoted by $r_\omega(H_p)$

Examples

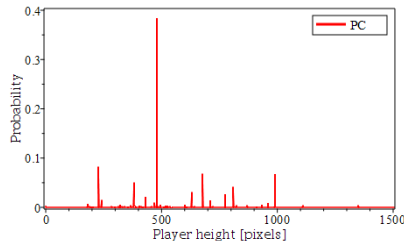
- ▶ Device distribution



- ▶ Network distributions



- ▶ Player size distribution



Average Performance Parameters

Average bitrate

$$\bar{R}(H_1, \dots, H_n, R_1, \dots, R_n) = \int_0^\infty q(B) \sum_{H_p} r(H_p) R_{i(B, H_p)} dB$$

Average resolution

$$\bar{H}(H_1, \dots, H_n, R_1, \dots, R_n) = \int_0^\infty q(B) \sum_{H_p} r(H_p) H_{i(B, H_p)} dB$$

Average distortion

$$\bar{D}(H_1, \dots, H_n, R_1, \dots, R_n) = \int_0^\infty q(B) \sum_{H_p} r(H_p) D(H_{i(B, H_p)}, R_{i(B, H_p)}) dB$$

Average quality

$$\bar{Q}(H_1, \dots, H_n, R_1, \dots, R_n) = \int_0^\infty q(B) \sum_{H_p} r(H_p) Q(H_{i(B, H_p)}, H_p, R_{i(B, H_p)}) dB$$

Averaging over
bandwidth B

Averaging over
player sizes H_p

Client model selecting
rendition based on B, H_p

Parameter that
is averaged

Notations

- ▶ H_1, \dots, H_n – ladder resolutions
- ▶ R_1, \dots, R_n – ladder bitrates
- ▶ B – network bandwidth
- ▶ H_p – player size
- ▶ $i(B, H_p)$ – index of rendition selected
- ▶ $q(B)$ – network bandwidth distribution
- ▶ $r(H_p)$ – distribution of player sizes
- ▶ $D(H, R)$ – distortion-rate model
- ▶ $Q(H, H_p, R)$ – quality-rate model, specific to a given codec, content, and user device

Average Performance for Multiscreen

Average bitrate

$$\bar{R}(H_1, \dots, H_n, R_1, \dots, R_n) = \sum_{\omega} p(\omega) \int_0^{\infty} q_{\omega}(B) \sum_{H_p} r_{\omega}(H_p) R_{i_{\omega}(B, H_p)} dB$$

Average resolution

$$\bar{H}(H_1, \dots, H_n, R_1, \dots, R_n) = \sum_{\omega \in \Omega} p(\omega) \int_0^{\infty} q_{\omega}(B) \sum_{H_p} r_{\omega}(H_p) H_{i_{\omega}(B, H_p)} dB$$

Average distortion

$$\bar{D}(H_1, \dots, H_n, R_1, \dots, R_n) = \sum_{\omega \in \Omega} p(\omega) \int_0^{\infty} q_{\omega}(B) \sum_{H_p} r_{\omega}(H_p) D(H_{i_{\omega}(B, H_p)}, R_{i_{\omega}(B, H_p)}) dB$$

Average quality

$$\bar{Q}(H_1, \dots, H_n, R_1, \dots, R_n) = \sum_{\omega} p(\omega) \int_0^{\infty} q_{\omega}(B) \sum_{H_p} r_{\omega}(H_p) Q_{\omega}(H_{i_{\omega}(B, H_p)}, H_p, R_{i_{\omega}(B, H_p)}) dB$$

Averaging over
devices ω

Averaging over
bandwidth B

Averaging over
player sizes H_p

Client model selecting
rendition based on ω, B, H_p

Parameter that
is averaged

Marked red:

- ▶ summation over devices ω
- ▶ usage probability of each device
 - $p(\omega)$
- ▶ elements depended on a device:
 - $i_{\omega}(B, H_p)$ – client model
 - $q_{\omega}(B)$ – network pdf
 - $r_{\omega}(H_p)$ – player size pmf
 - $Q_{\omega}(H, H_p, R)$ – quality rate model

Computation of Performance Metrics

System configurations

- ABR ladder:

#	Codec	Profile	Resolution	Framerate	Bitrate
1	H.264	Baseline	480x270	23.976	450k
2	H.264	Baseline	640x360	23.976	800k
3	H.264	Main	768x432	23.976	1000k
4	H.264	Main	1024x576	23.976	1500k
5	H.264	Main	1280x720	23.976	2100k

- Content:

Sequence	α	β	γ
"Easy"	0.7844e-3	1.2281	0.7463
"Medium"	0.8278e-2	1.3217	0.9593
"Complex"	0.07316	1.0957	1.0336

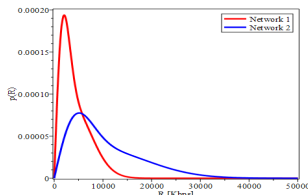
$$D(H, R) = \left(1 + \left(\frac{R}{\alpha H^\beta}\right)^{-\gamma}\right)^{-\frac{1}{\gamma}}$$

- Networks:

Network	α	σ_1	σ_2
"Network 1"	0.4287	1802.2	4499.28
"Network 2"	0.4287	4505.5	11248.2

$$q(B) = \alpha f(B, \sigma_1) + (1 - \alpha) f(B, \sigma_2),$$

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



- Devices / Players:

Player	Player sizes \mathcal{H}_p	Player size probabilities $q(\mathcal{H}_p)$
"1080p player"	{1080}	{1.0}
"Web player"	{228, 240, 380, 430, 480, 630, 678, 710, 774, 810, 990}	{0.103188906, 0.017734224, 0.062664264, 0.026945508, 0.480776451, 0.038259368, 0.083865235, 0.018247353, 0.033203174, 0.051450527, 0.08366499}

Other model parameters: $d=24$ [in], $\rho=96$ [dpi], $\alpha=0.88$, $\delta=0.49$

Average performance

Network	Player	Content	\bar{H}	\bar{D}	\bar{Q}	\bar{R}
Network 1	1080p player	Easy	647.8	0.9910	4.075	1846.5
		Medium	647.8	0.9718	3.891	1846.5
		Complex	647.8	0.9585	3.774	1846.5
	Web player	Easy	465.2	0.9903	3.563	1151.8
		Medium	465.2	0.9701	3.395	1151.8
		Complex	465.2	0.9522	3.258	1151.8
Network 2	1080p player	Easy	706.3	0.9912	4.343	2076.5
		Medium	706.3	0.9724	4.150	2076.5
		Complex	706.3	0.9606	4.034	2076.5
	Web player	Easy	486.5	0.9904	3.653	1232.6
		Medium	486.5	0.9704	3.482	1232.6
		Complex	486.5	0.9532	3.347	1232.6

\bar{Q} = average quality [MOS]

\bar{R} = average bitrate [Kbps]

\bar{H} = average resolution [lines]

\bar{D} = average distortion [SSIM]

Observations

- Complex content gets encoded & delivered at lower quality
- Network 2 allows more bits to be pulled and quality to increase
- Web players deliver much lower average resolution and consume fewer bits compared to full-screen/1080p players

Related Optimization Problems

Optimal Encoding Ladders

Maximum-quality ladder

- Given parameters needed for computation of $\bar{Q}(H_1, \dots, H_n, R_1, \dots, R_n)$, find $\hat{R}_1, \dots, \hat{R}_n$ and $\hat{H}_1, \dots, \hat{H}_n$ such that:

$$\bar{Q}(\hat{H}_1, \dots, \hat{H}_n, \hat{R}_1, \dots, \hat{R}_n) = \max_{\substack{R_{\min} \leq R_1 < \dots < R_n \leq R_{\max} \\ H_{\min} \leq H_1 < \dots < H_n \leq H_{\max} \\ R_1 \leq R_{1,\max}, H_1 \leq H_{1,\max}}} \bar{Q}(H_1, \dots, H_n, R_1, \dots, R_n) \quad (*)$$

- The simplest problem that can be posed.

Maximum-quality ladder with capped average bitrate

- Given all parameters needed for computation of $\bar{Q}(H_1, \dots, H_n, R_1, \dots, R_n)$, find $\hat{R}_1, \dots, \hat{R}_n$ and $\hat{H}_1, \dots, \hat{H}_n$ such that:

$$\bar{Q}(\hat{H}_1, \dots, \hat{H}_n, \hat{R}_1, \dots, \hat{R}_n) = \max_{\substack{R_{\min} \leq R_1 < \dots < R_n \leq R_{\max} \\ H_{\min} \leq H_1 < \dots < H_n \leq H_{\max} \\ R_1 \leq R_{1,\max}, H_1 \leq H_{1,\max} \\ \bar{R}(\hat{H}_1, \dots, \hat{H}_n, \hat{R}_1, \dots, \hat{R}_n) \leq \bar{R}_{\max}}} \bar{Q}(H_1, \dots, H_n, R_1, \dots, R_n)$$

Minimum bitrate ladder with guaranteed minimum average quality

- Given all parameters needed for computation of $\bar{Q}(H_1, \dots, H_n, R_1, \dots, R_n)$, find $\hat{R}_1, \dots, \hat{R}_n$ and $\hat{H}_1, \dots, \hat{H}_n$ such that:

$$\bar{R}(\hat{H}_1, \dots, \hat{H}_n, \hat{R}_1, \dots, \hat{R}_n) = \min_{\substack{R_{\min} \leq R_1 < \dots < R_n \leq R_{\max} \\ H_{\min} \leq H_1 < \dots < H_n \leq H_{\max} \\ R_1 \leq R_{1,\max}, H_1 \leq H_{1,\max} \\ \bar{Q}(\hat{H}_1, \dots, \hat{H}_n, \hat{R}_1, \dots, \hat{R}_n) \geq \bar{Q}_{\min}}} \bar{R}(H_1, \dots, H_n, R_1, \dots, R_n)$$

- More complete problem. Solution of (*) may be needed first, to set a meaningful limit for \bar{Q}_{\min} .

Examples of Optimal Ladders

Maximum quality ladders

Network	Player	Content	Rendition 1	Rendition 2	Rendition 3	Rendition 4	Rendition 5	\bar{H}	\bar{D}	\bar{Q}	\bar{R}
Network 1	1080p player	Easy	854x480 167k	1024x576 173k	1280x720 277k	1600x900 607k	1920x1080 1557k	1043.7	0.9819	4.955	1388.4
		Medium	854x480 180k	1024x576 410k	1280x720 769k	1600x900 1384k	1920x1080 2804k	975.4	0.9534	4.512	2130.5
		Complex	854x480 180k	1024x576 480k	1280x720 899k	1600x900 1619k	1920x1080 3155k	954.4	0.9392	4.337	2288.0
	Web player	Easy	512x288 180k	768x432 365k	854x480 935k	1280x720 973k	1600x900 1557k	537.7	0.9850	3.719	1094.5
		Medium	480x270 180k	768x432 632k	854x480 1497k	1280x720 1619k	1600x900 2697k	515.9	0.9617	3.473	1262.3
		Complex	480x270 180k	768x432 739k	854x480 1684k	1280x720 1970k	1600x900 3155k	506.1	0.9420	3.316	1407.7
Network 2	1080p player	Easy	854x480 180k	1024x576 203k	1280x720 410k	1600x900 1052k	1920x1080 3033k	1057.8	0.9892	5.000	2811.0
		Medium	854x480 180k	1024x576 540k	1280x720 1138k	1600x900 2305k	1920x1080 5050k	1019.8	0.9747	4.830	4264.4
		Complex	854x480 180k	1024x576 607k	1280x720 1280k	1600x900 2493k	1920x1080 5050k	1017.3	0.9670	4.741	4292.1
	Web player	Easy	480x270 180k	768x432 657k	854x480 1895k	1280x720 1970k	1920x1080 2697k	545.9	0.9898	3.781	1608.6
		Medium	480x270 180k	768x432 1052k	854x480 2804k	1280x720 2917k	1600x900 4856k	530.0	0.9741	3.630	2421.0
		Complex	384x216 180k	768x432 1183k	854x480 3155k	1280x720 3281k	1600x900 5050k	519.1	0.9638	3.531	2635.8

Generated under constraints: $R_{\min}=100, R_{\max}=5050, H_{\min}=180, H_{\max}=1080, R_{1,\max}, R_{1,\max}=180, H_{1,\max}=480, n = 5$

\bar{Q} = average quality [MOS]

\bar{R} = average bitrate [Kbps]

\bar{H} = average resolution [lines]

\bar{D} = average distortion [SSIM]

Observations

- ▶ Compared to a fixed ladder considered earlier, **quality values are now up by 0.1..0.88 MOS!**
- ▶ Optimal profiles are very different for different content: complex content receives more bits
- ▶ Optimal profiles are also very different for different networks: more bits are used with better networks
- ▶ Optimal profiles are also very different for different devices/players: for web-players lower resolutions and higher SSIMs are used as opposed to profiles generated for 1080p/full-screen players

Importance of Right Targeting

Optimal ladders for full-screen players, but played by web players

Network	Player	Content	Rendition 1	Rendition 2	Rendition 3	Rendition 4	Rendition 5	\bar{H}	\bar{D}	\bar{Q}	\bar{R}
Network 1	1080p player	Easy	854x480 167k	1024x576 173k	1280x720 277k	1600x900 607k	1920x1080 1557k	584.4	0.9630	3.569	306.3
		Medium	854x480 180k	1024x576 410k	1280x720 769k	1600x900 1384k	1920x1080 2804k	576.4	0.8737	2.915	485.6
		Complex	854x480 180k	1024x576 480k	1280x720 899k	1600x900 1619k	1920x1080 3155k	574.1	0.8028	2.513	523.4
Network 2	1080p player	Easy	854x480 180k	1024x576 203k	1280x720 410k	1600x900 1052k	1920x1080 3033k	585.9	0.9670	3.609	476.4
		Medium	854x480 180k	1024x576 540k	1280x720 1138k	1600x900 2305k	1920x1080 5050k	581.8	0.8804	2.989	768.5
		Complex	854x480 180k	1024x576 607k	1280x720 1280k	1600x900 2493k	1920x1080 5050k	581.4	0.8115	2.603	800.5

Optimal ladders for web-players, played by web players

Network	Player	Content	Rendition 1	Rendition 2	Rendition 3	Rendition 4	Rendition 5	\bar{H}	\bar{D}	\bar{Q}	\bar{R}
Network1	Web player	Easy	512x288 180k	768x432 365k	854x480 935k	1280x720 973k	1600x900 1557k	537.7	0.9850	3.719	1094.5
		Medium	480x270 180k	768x432 632k	854x480 1497k	1280x720 1619k	1600x900 2697k	515.9	0.9617	3.473	1262.3
		Complex	480x270 180k	768x432 739k	854x480 1684k	1280x720 1970k	1600x900 3155k	506.1	0.9420	3.316	1407.7
Network2	Web player	Easy	480x270 180k	768x432 657k	854x480 1895k	1280x720 1970k	1920x1080 2697k	545.9	0.9898	3.781	1608.6
		Medium	480x270 180k	768x432 1052k	854x480 2804k	1280x720 2917k	1600x900 4856k	530.0	0.9741	3.630	2421.0
		Complex	384x216 180k	768x432 1183k	854x480 3155k	1280x720 3281k	1600x900 5050k	519.1	0.9638	3.531	2635.8

Observations

- ▶ Mistargeting has severe consequences: **we see a drop of 0.15..0.93 MOS when we switch players!!**
- ▶ Web players need high-quality low-res renditions, but in profiles for full screen players they are either missing or encoded at poor quality, as preference is given to resolution.

\bar{Q} = average quality [MOS]
 \bar{R} = average bitrate [Kbps]
 \bar{H} = average resolution [lines]
 \bar{D} = average distortion [SSIM]

How Many Streams are Needed?

Optimal ladders for n=2..5

Network	Player	Content	n	Rendition 1	Rendition 2	Rendition 3	Rendition 4	Rendition 5	\bar{H}	\bar{D}	\bar{Q}	\bar{R}
Network 1	1080p player	Easy	2	854x480 180k	1920x1080 899k				1043.1	0.9754	4.843	854.8
			3	854x480 180k	1600x900 427k	1920x1080 1440k			1047.7	0.9805	4.942	1288.9
			4	854x480 180k	1280x720 228k	1600x900 584k	1920x1080 1557k		1044.2	0.9817	4.954	1385.1
			5	854x480 167k	1024x576 173k	1280x720 277k	1600x900 607k	1920x1080 1557k	1043.7	0.9819	4.955	1388.4
		Medium	2	854x480 180k	1920x1080 1440k				992.6	0.9227	4.186	1256.4
			3	854x480 180k	1600x900 865k	1920x1080 2305k			1000.3	0.9416	4.431	1819.8
			4	854x480 180k	1280x720 584k	1600x900 1280k	1920x1080 2697k		983.4	0.9501	4.496	2061.6
			5	854x480 180k	1024x576 410k	1280x720 769k	1600x900 1384k	1920x1080 2804k	975.4	0.9534	4.512	2130.5
	Complex		2	854x480 180k	1920x1080 1557k				980.0	0.8911	3.911	1327.4
			3	854x480 180k	1600x900 973k	1920x1080 2493k			987.9	0.9199	4.217	1911.5
			4	854x480 180k	1280x720 657k	1600x900 1440k	1920x1080 2917k		969.9	0.9327	4.310	2164.6
			5	854x480 180k	1024x576 480k	1280x720 899k	1600x900 1619k	1920x1080 3155k	954.4	0.9392	4.337	2288.0

Notes

- ▶ Number of streams needed clearly depends on content. E.g. "Easy" content needs only 4 or even 3 streams in this case. Quality saturates with more. "Medium" and "Complex" content need more.
- ▶ Generally, more streams allow better network utilization, and better quality. But return is diminishing.
- ▶ In practice, a threshold on relative quality increase can be used to limit the ladder.
- ▶ Limits on "granularity" of switches can also be used to decide how many streams are sufficient.

\bar{Q} = average quality [MOS]
 \bar{R} = average bitrate [Kbps]
 \bar{H} = average resolution [lines]
 \bar{D} = average distortion [SSIM]

Other Optimization Problems

Many possible extensions and other problems can be proposed

- ▶ Finding optimal parameters (α, δ) for client adaptation model, and then improving client logic
- ▶ Using worst-case instead of average case, or combinations of worst-case and average case metrics in definition of the problem
- ▶ Taking into account effects of multiple streams on CDN cache miss probability and factoring it in the optimization problem
- ▶ Adding time domain to all models, and capturing buffering and delay as additional dimensions of quality
- ▶ etc., etc....

Plenty of opportunities for subsequent research...

References

- ▶ Y. Reznik, et al, "Optimal Design of Encoding Profiles for ABR Streaming," *Packet Video*, June 2018
- ▶ Y. Reznik, et al, "Towards Understanding of the Behavior of Web Streaming", *PCS*, June 2021.
- ▶ Y. Reznik, et al, "Perceptually optimized ABR ladder generation for Web streaming," *Electronic Imaging*, Jan. 2021
- ▶ Y. Reznik, et al, "Average Performance of Adaptive Streaming," *DCC*, Mar. 2021
- ▶ Y. Reznik, et al, "Optimizing Mass-Scale Multiscreen Video Delivery," *SMPTE Motion Imaging Journal*, Apr., 2020

**THANK
YOU**