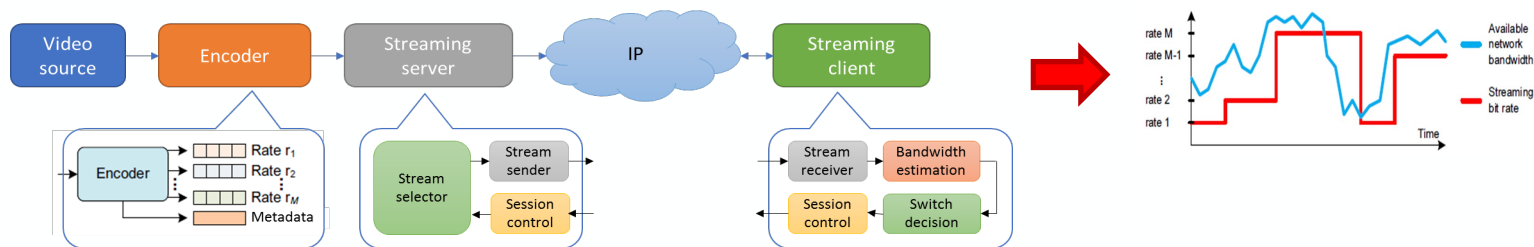


# HTTP Streaming and CDN Performance



# The original (pre-HTTP-era) ABR architecture

RealSystem G2 (1998):



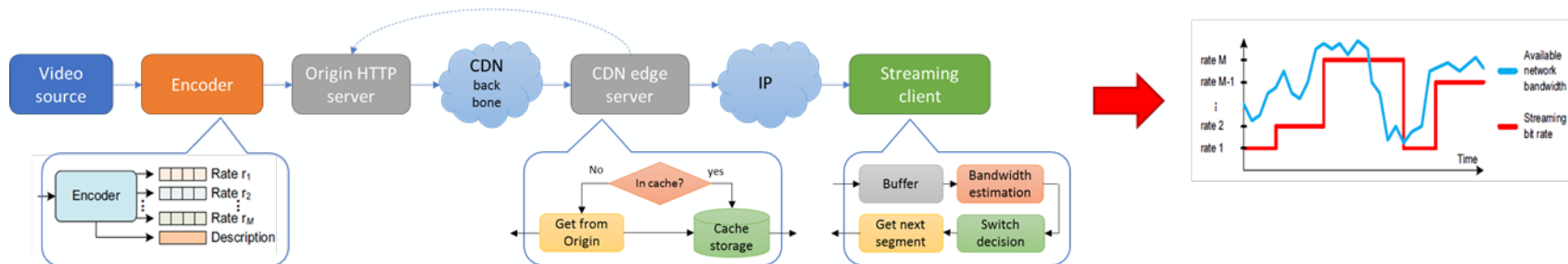
- ▶ RTSP session control, RDT, RTP, or TCP for stream transmissions. Public IP is used for delivery.
- ▶ Stream adaptation was done by server, but it was client-driven: client was sending requests to switch
- ▶ Server was also responsible for retransmissions, mixing in FEC packets, etc.
- ▶ Everything was sent in “packets”

Most important feature:

- ▶ There was **only one version of stream** sent by the server over IP network to the receiving device!
- ▶ Multi-rate encodings have only existed in the interface between encoder and streaming server, but distribution was fundamentally done by single streams.

# HTTP-based ABR streaming

HTTP-based ABR system:



Key differences from the original design:

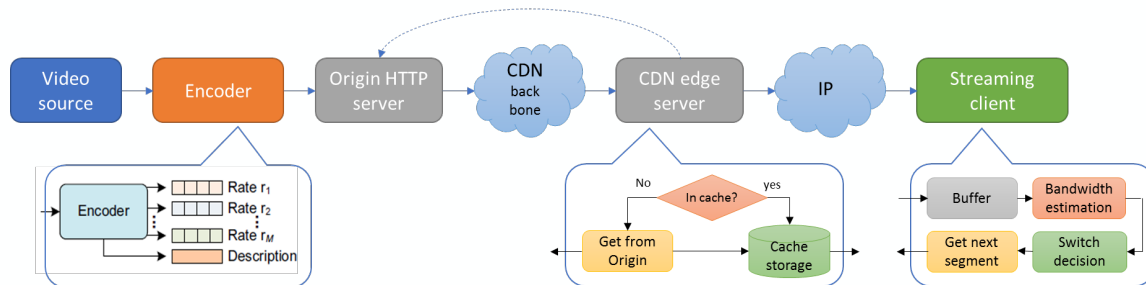
- ▶ Instead of custom “streaming server”, a regular HTTP server is used as origin
- ▶ Stream switching is trivialized to HTTP GET operations originating from streaming client
- ▶ The scaling and delivery is delegated to CDN, which caches content on its edges, and thus reduces the load on the origin

Factors affecting scalability:

- ▶ CDN cache efficiency is a key! If it can't cache content on the edges, all load will come back to the origin server.
- ▶ But, notably, **CDN has to cache not a single stream, but all streams** used for encoding and distribution of the content!

# Disconnect Between ABR Streaming and CDN Delivery Model

Let's review this chain again:



Key issues:

- ▶ ABR encoder produces **several encoded versions of the content**, which then “compete” for the CDN edge cache space.
- ▶ In multi-screen systems it is also common to use several different delivery formats: HLSv3, HLSv7, DASH, Smooth, etc.
- ▶ Even more streams are needed to support different DRMs: PlayReady, FairPlay, Widevine, etc.
- ▶ And we also have new codecs, such as HEVC, AV1, and VVC, resulting in even more streams that should be generated.
- ▶ All such additional streams reduce the effectiveness of the CDN. More traffic becomes routed to the origin!

The disconnect:

- ▶ Efficient/scalable ABR streaming requires “more” streams to be generated, while CDN needs “less” to stay effective!

# Modeling CDN Cache Miss Probability

- Consider the following:
  - a set of items  $S = \{s_x, x \geq 1\}$ ,
  - occurring with following probabilities:

$$p(x) = \frac{x^{-\alpha}}{\zeta(\alpha)}$$

where  $\alpha$  is a shape parameter, and  $\zeta(\cdot)$  is a Riemann Zeta function

- If we also assume that cache can only hold  $C$  most probable items, then **cache miss probability** becomes:

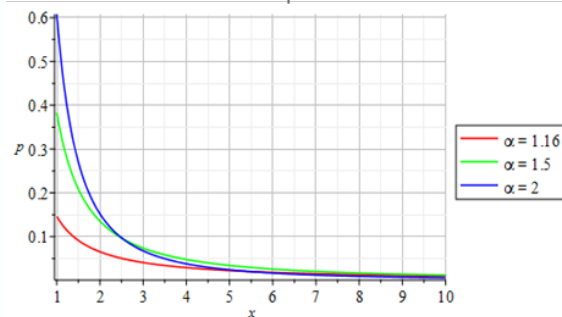
$$p_{miss}(C, \alpha) = 1 - \sum_{x=1}^C p(x) = 1 - \frac{H_{C, \alpha}}{\zeta(\alpha)},$$

where  $H_{C, \alpha} = \sum_{x=1}^C x^{-\alpha}$  is a generalized Harmonic number

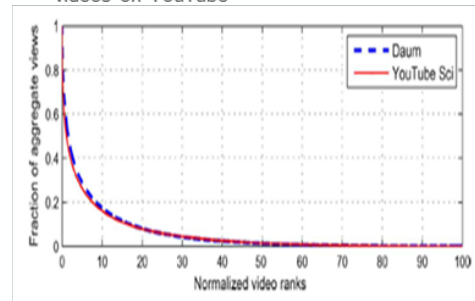
- Asymptotically, with large cache capacity  $C$ :

$$p_{miss}(C, \alpha) \sim \frac{C^{1-\alpha}}{(\alpha-1)\zeta(\alpha)} \left(1 + o\left(\frac{1}{C}\right)\right)$$

Shape of popularity distribution model for different values of parameter  $\alpha$



Empirically measured popularity of videos on YouTube




M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, S. Moon, Sue. (2009). Analyzing the Video Popularity Characteristics of Large-Scale User Generated Content Systems. IEEE/ACM Trans. Netw.. 17. 1357-1370.

# Cache Behavior with 2 Versions of the Same Content

- Let's consider 2 sets:  $S_1 = \{s_{1,x}, x \geq 1\}$ , and  $S_2 = \{s_{2,x}, x \geq 1\}$
- Let also  $\pi = \{\pi_1, \pi_2\}$  denote **usage probabilities** of both versions
- The **full probabilities** of such items:

$$p(s_{1,x}) = \pi_1 \cdot p(s_x) \quad \text{and} \quad p(s_{2,x}) = \pi_2 \cdot p(s_x)$$

- Then, **at the top of the CDN cache** we will observe the following structure (case when  $\pi_1 > \pi_2$ ):



Item	Probability	Comments
$s_{1,1}$	$\pi_1 p(1)$	First go items of more frequently used format
...	...	...
$s_{1,x}$	$\pi_1 p(x)$	$x = \left\lceil \left( \frac{\pi_1}{\pi_2} \right)^{\frac{1}{\alpha}} \right\rceil$ , solution of $\pi_1 p(x) = \pi_2 p(1)$
$s_{2,1}$	$\pi_2 p(1)$	Now comes first item in less frequently used format
$s_{1,x+1}$	$\pi_1 p(x+1)$	Then follow items in more frequently used content
...	...	...
$s_{1,x_2}$	$\pi_1 p(x_2)$	$x_2 = \left\lceil 2 \left( \frac{\pi_1}{\pi_2} \right)^{\frac{1}{\alpha}} \right\rceil$ , solution of $\pi_1 p(x_2) = \pi_2 p(2)$
$s_{2,2}$	$\pi_2 p(2)$	Now comes second item in less frequently used format
$s_{1,x_2+1}$	$\pi_1 p(x_2+1)$	Then follow items in more frequently used content
...	...	...

- In other words, **items in less frequently used format become interleaved** with step size  $x \sim (\pi_1/\pi_2)^{1/\alpha}$  !

# Cache Miss Probability with 2 Versions of the Same Content

- Using simple math, we can show that for 2 versions of the content:

$$p_{miss,2}(C, \alpha, \pi) \sim \left( \pi_1^{\frac{1}{\alpha}} + \pi_2^{\frac{1}{\alpha}} \right)^{\alpha} \frac{C^{1-\alpha}}{(\alpha-1)\zeta(\alpha)} \left( 1 + O\left(\frac{1}{C}\right) \right)$$

- Comparing it with cache miss probability for single content:

$$p_{miss}(C, \alpha) \sim \frac{C^{1-\alpha}}{(\alpha-1)\zeta(\alpha)} \left( 1 + O\left(\frac{1}{C}\right) \right)$$

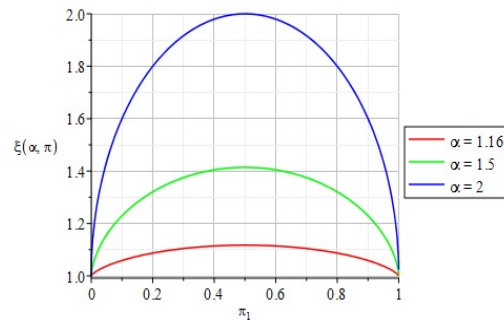
we observe that the ratio:

$$\xi(\alpha, \pi) = \frac{p_{miss,2}(C, \alpha, \pi)}{p_{miss}(C, \alpha)} \sim \left( \pi_1^{\frac{1}{\alpha}} + \pi_2^{\frac{1}{\alpha}} \right)^{\alpha}$$

becomes asymptotically independent on C!

- Hence, **considering a CDN with a reasonably large cache**, we can predict that **the use of 2 formats will increase its cache miss probability by  $\left( \pi_1^{1/\alpha} + \pi_2^{1/\alpha} \right)^{\alpha}$**
- The worst impact happens when both versions are equally probable ( $\pi_1 = \pi_2$ ).

Relative increase in cache miss probability in case of using 2 formats.



# Applications

Next, we will show how above derived model can be used to:

- Predict when it makes sense to deploy 3<sup>rd</sup> streaming format (e.g. CMAF, considering that HLS and DASH are already deployed and used by some devices)
- Predict when it makes sense to use 2 codecs (e.g. HEVC in addition to H.264)



# Application 1: Deployment of CMAF in addition to HLS and DASH

- Consider deployment of CMAF, where it has to co-exist with older flavors of HLS and DASH
  - In other words, we must compare 3-format system (HLS+DASH+CMAF) vs 2-format system (HLS+DASH)
  - The cache miss probabilities in such systems:

$$p_{miss,2}(C, \alpha, \pi) \sim \left(\pi_1^{\frac{1}{\alpha}} + \pi_2^{\frac{1}{\alpha}}\right)^\alpha \frac{C^{1-\alpha}}{(1-\alpha)\zeta(\alpha)}; \quad p_{miss,3}(C, \alpha, \rho) \sim \left(\rho_1^{\frac{1}{\alpha}} + \rho_2^{\frac{1}{\alpha}} + \rho_3^{\frac{1}{\alpha}}\right)^\alpha \frac{C^{1-\alpha}}{(1-\alpha)\zeta(\alpha)}$$

where  $\rho = \{\rho_1, \rho_2, \rho_3\}$  are usage probabilities in 3 format system.

- Considering that CMAF may be supported by a fraction  $\kappa$  of both HLS and DASH players, we can write:

$$\rho_1 = \kappa(\pi_1 + \pi_2), \quad \rho_2 = (1 - \kappa)\pi_1, \quad \rho_3 = (1 - \kappa)\pi_2$$

- Then, by using inequality:

$$p_{miss,3}(C, \alpha, \rho) < p_{miss,2}(C, \alpha, \pi)$$

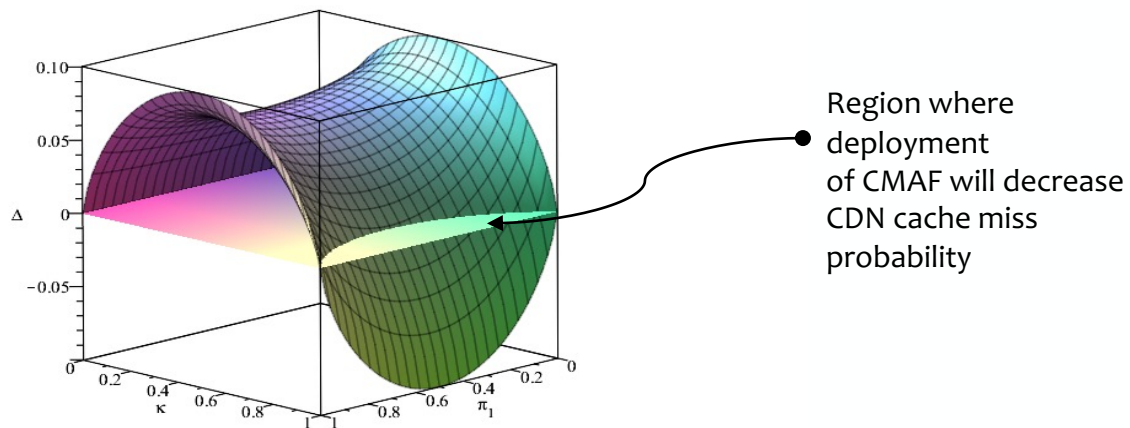
or equivalently:

$$\Delta(\kappa, \alpha, \pi) = \kappa^{\frac{1}{\alpha}} + (1 - \kappa)^{\frac{1}{\alpha}} \left(\pi_1^{\frac{1}{\alpha}} + \pi_2^{\frac{1}{\alpha}}\right) - \left(\pi_1^{\frac{1}{\alpha}} + \pi_2^{\frac{1}{\alpha}}\right) < 0$$

we can arrive at **region where deployment of CMAF will reduce CDN cache miss probability.**

# Application 1: Deployment of CMAF – Minimum Reach Needed

- Let's plot  $\Delta(\kappa, \alpha, \pi)$  with respect to  $\kappa$  (percentage of CMAF capable players), and  $\pi_1$  (percentage of currently deployed HLS content vs DASH):



- It can be observed that there exists region, where deployment of CMAF will lead to negative  $\Delta(\kappa, \alpha, \pi)$  and hence lower CDN cache miss probabilities.
- Naturally, this will only be possible when percentage of CMAF-capable receiving devices will be high.
- For example, for current ballpark numbers ( $\alpha = 1.16$  and  $\pi_1 = 0.75$ ), the **minimum CMAF deployment factor that is needed to break even in CDN costs is  $\kappa \sim 0.84$** .

## Application 2: Deployment of HEVC in addition to H.264

- Consider the following:
  - $R$  = encoding rate used by H.264
  - $\delta$  = rate savings of HEVC vs H.264 (in practice,  $\delta \approx 0.25..0.5$ )
  - $\pi$  = percentage of devices that can play HEVC (in practice,  $\pi \approx 25..100\%$  depending on customer)
  - $\alpha$  = shape parameter of content popularity distribution, which is modeled as  $p(x) = \frac{x^{-\alpha}}{\zeta(\alpha)}$

- Then:

- The amount of **storage** that will be used by mix of H.264 + HEVC streams:

$$S = R + R (1 - \delta) = R (2 - \delta) \rightarrow \text{always larger than } R$$

- Average **bandwidth** that will be used by a mix of H.264 and HEVC streams:

$$\bar{R} = R (1 - \pi) + R \pi (1 - \delta) = R (1 - \pi \delta) \rightarrow \text{always less than } R$$

- **CDN cache miss probability** will be affected by a factor:

$$\xi(\delta, \pi, \alpha) = \left( \pi^{\frac{1}{\alpha}} + (1 - \pi)^{1/\alpha} \right)^{\alpha} \left( 1 - \delta \frac{\pi^{\frac{1}{\alpha}}}{\pi^{\frac{1}{\alpha}} + (1 - \pi)^{1/\alpha}} \right)^{\alpha-1}$$

Factor  $\xi(\delta, \pi, \alpha)$  can be greater or less than 1 based on values  $\pi$  and  $\delta$ .

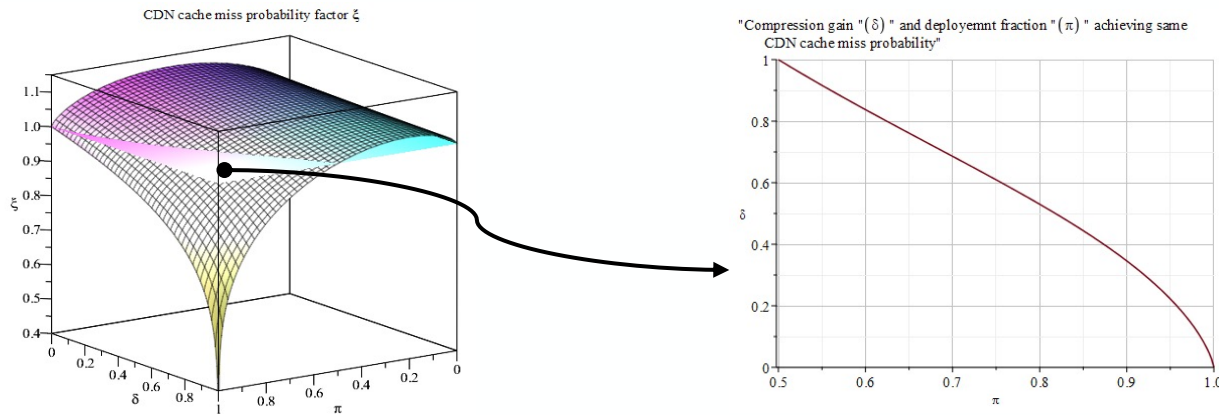
## Application 2: Deployment of HEVC – Cache Miss Probability

- Concurrent deployment of HEVC changes CDN cache miss probability by a factor:

$$\xi(\delta, \pi, \alpha) = \left( \pi^{\frac{1}{\alpha}} + (1 - \pi)^{\frac{1}{\alpha}} \right)^{\alpha} \left( 1 - \delta \frac{\pi^{\frac{1}{\alpha}}}{\pi^{\frac{1}{\alpha}} + (1 - \pi)^{\frac{1}{\alpha}}} \right)^{\alpha - 1}$$

where  $\delta$  = rate savings,  $\pi$  = device support.

- Visualizations:



- NB: with HEVC rate savings is  $\delta = 0.5$ , we must have  $\pi \approx 0.82$  (about 82% HEVC support across devices) to achieve the same CDN cache miss probability.

## Application 2: Deployment of HEVC – CDN Traffic

- Generally, when using CDNs we have 2 types of traffic:
  - **CDN edge traffic** – traffic that delivers data to end user, billed at  $C_{edge}$  per GB
  - **CDN origin traffic** – traffic that pulls data back from the origin served, billed at  $C_{origin}$  per GB
- The **total cost of using CDN and origin servers**, therefore becomes
  - $C_{total} = R(C_{edge} + p_{miss}C_{origin})$ , where  $p_{miss}$  is a probability of CDN cache miss

- Now, considering HEVC+H.264 vs H.264-only deployments we will have the following change:

$$\frac{C_{total,H.264+HEVC}}{C_{total,H.264}} = (1 - \pi \delta) \frac{C_{edge} + p_{miss}\xi(\delta, \pi, \alpha)C_{origin}}{C_{edge} + p_{miss}C_{midgresrigins}},$$

where  $(1 - \pi \delta)$  reflects change in average rate, and  $\xi(\delta, \pi, \alpha)$  change in cache miss probability.

- In a case when origin (or midgress) traffic is much more expensive  $C_{origin} \gg C_{edge}$ , we obtain:

$$\frac{C_{total,H.264+HEVC}}{C_{total,H.264}} \sim (1 - \pi \delta) \xi(\delta, \pi, \alpha)$$

which effectively is a change in the traffic to origin.

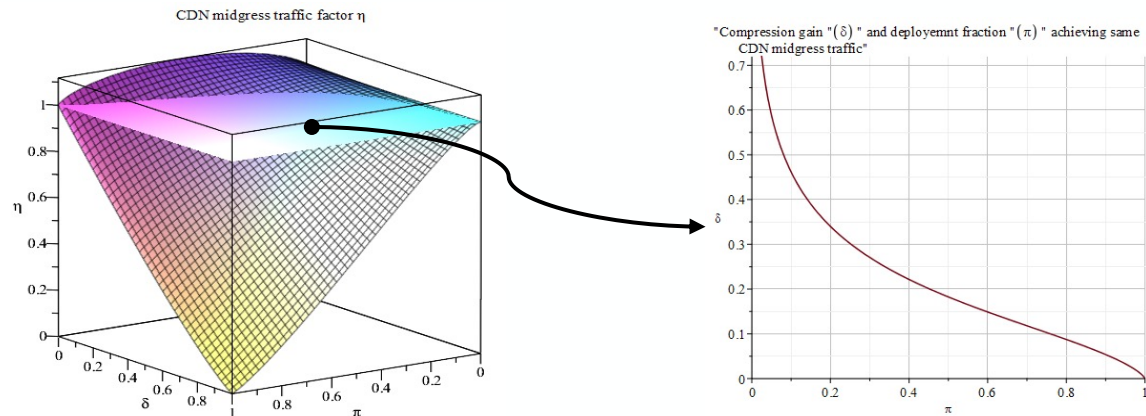
## Application 2: Deployment of HEVC – minimum reach needed

- As explained earlier, deployment of HEVC will affect CDN-origin traffic by a factor:

$$\eta(\delta, \pi, \alpha) = (1 - \pi\delta) \left( \pi^{\frac{1}{\alpha}} + (1 - \pi)^{\frac{1}{\alpha}} \right)^{\alpha} \left( 1 - \delta \frac{\pi^{\frac{1}{\alpha}}}{\pi^{\frac{1}{\alpha}} + (1 - \pi)^{\frac{1}{\alpha}}} \right)^{\alpha - 1}$$

where  $\delta$  = rate savings,  $\pi$  = device support.

- Visualizations:



- NB: with HEVC rate savings of  $\delta = 0.25$  we need at least  $\pi \approx \mathbf{0.33}$  (about 1/3 of all devices supporting HEVC) to break even in the CDN-origin traffic costs. With  $\delta = 0.5$  we need at least  $\pi \approx \mathbf{0.17}$ .

## Conclusions (for HTTP Streaming and CDN Performance)

- While there is an apparent disconnect between ABR streaming model and CDN efficiency, many of the related effects can be modeled, understood, and used in practice to improve efficiency of systems
- Some immediate consequences suggested by this study include:
  - Reducing the total number of streams is always good for increasing CDN efficiency
  - If number of streams cannot be reduced – **increasing the asymmetry in their usage distribution** is another effective technique for improving CDN efficiency
    - E.g. pick a dominant format (e.g. HLS or DASH) and force as many players possible to use it
  - Adding new codecs or formats may lead to reduction of traffic / costs – but only if such codecs or formats are sufficiently well supported by the devices
    - If they are not well supported, delaying deployment of such new technologies makes more practical sense.