EVOLUTION OF VIDEO

AND VIDEO DELIVERY TECHNOLOGIES

Yuriy A. Reznik

Streaming Labs, London, UK

Outline

A look at the history of video and streaming

- A look at a longer timeline
- Examples of some early inventions
- What was before the Internet and streaming?

Evolution of streaming

- Early systems
- ABR streaming before HTTP
- ABR streaming with HTTP
- Evolution of ABR systems

What may come next?

- New forms of “video”
- In pursuit of lower delays
- Back to … some earlier ideas?

Evolution of video

Evolution of Video Technologies

THE PAST:

Invention of camera, still image photography, color reproduction, film, moving pictures

THE PRESENT:

New delivery methods: TV, recordable media, digital compressed formats, Internet streaming, mobile.

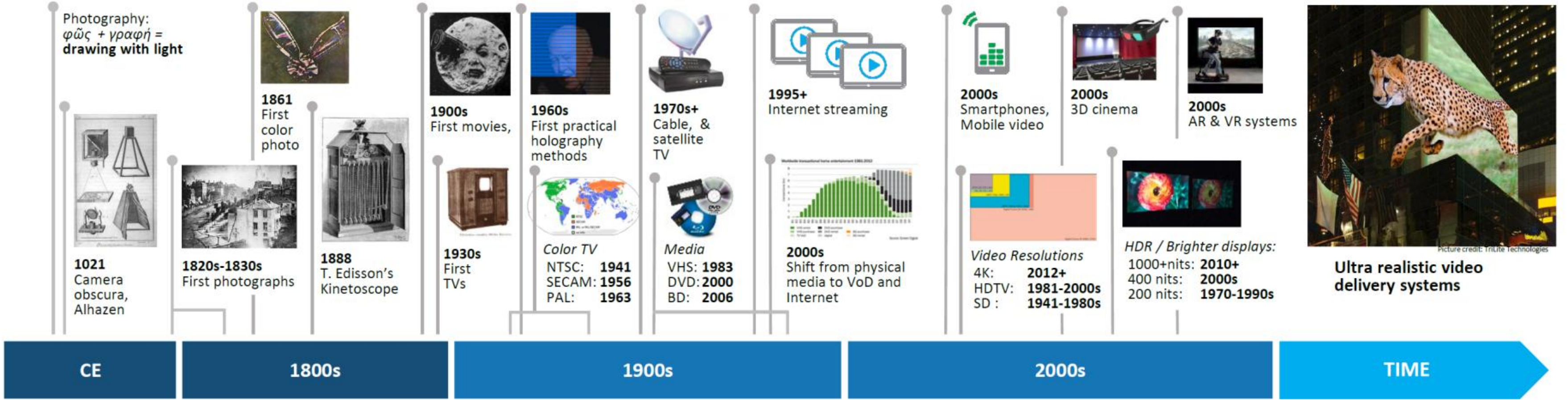
Increasing degree of realism: immersive video, 3D (holography, stereoscopic rendering, etc.)

Increasing presence of artificially-generated content:

GenAI, AI in postproduction, AI-based editing, FX, etc.

THE FUTURE:

Recording & reproduction systems making rendered video undistinguishable from reality.



Everything we know about video are the results of human inventions

- Cameras, photographs, film, CCDs, digital media formats, displays, TVs, compression algorithms, streaming, etc.
- But as time progresses, we often forget what, why, and for which reason was initially invented.

Examples of some early decisions

Frames and framerates

- 10-15fps – zoographiscope animated images (E. Muybridge, 1880s)
- 24fps – first film projectors (T. Eddison & Co., 1930s)
- 25/30fps – first B&W TV receivers, synchronized by 50/60Hz AC (1940s)
- 29.97fps – NTSC (1953), fitting chroma in a fixed band allocated for B&W TVs

Lines and scan orders

- 1880 – Maurice Leblanc's patent
- 1931 – first CRT tubes and CRT-based TV systems (V. Zworykin et al, RCA).
- 1937 – 240 lines TV systems (Baird & Co.)
- 1941 – 441 lines TV systems (Bosch, Telefunken, et al.)
- 1948 – 525- and 625-lines TV systems (all interlaced!)

YUV color spaces

- Designed in 1938(!) for backwards compatibility with B&W TV systems
- Luma = "intensity" in earlier systems, "chroma" = complementary channels
- Variants: YPbPr, YDbDr, YIQ, YCbCr, etc.

XYZ → XYZ→RGB matrix multiplication → YUV gamma → R'G'B' → Y'UV matrix multiplication → Y'UV

24 fps framerates

Framerate adopted in film movie projectors. 1930s. T. Eddison. Note: first film cameras were hand-cranked!

Scan orders

Maurice Leblanc, "Etude sur la transmission électrique des impressions lumineuses", La Lumière Électrique, **Dec 1, 1880.**

YUV color space

Invented in 1938 by Georges Valensi as a mean to make color TV system compatible with B&W TV receivers. Y channel in YUV was meant to be B&W TV signal.

First communication systems

First electromagnetic telegraphs

- 1833 – Carl Friedrich Gauss & Wilhelm Weber, U. Göttingen, Germany
- 1837 – Samuel Morse & Alfred Vail, first commercial telegraph, D.C., USA
- 1866 – First transatlantic telegraph line, Anglo-American Telegraph Co.

First wireless communication systems

- 1893 – Nikola Tesla, first demo of wireless telegraph, Chicago World's Fair.
- 1896 – Guglielmo Marconi, demonstration of wireless telegraph, London, UK
- 1896 – Alexander Popov, demonstration of radio transmission, St. Petersburg, RU
- 1902 – Marconi & Co., first transatlantic conmmunication

First telephone calls

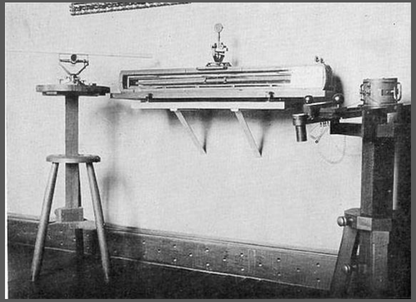
- 1892 – Alexander Graham Bell, call from New York to Chicago, Bell Telephone Co.
- 1973 – John F. Mitchell and Martin Cooper of Motorola, first "mobile" phone call

First video calls

- 1927 – AT&T's first demo of video phone: *ikonophone*

1833 First Telegraph

Carl Friedrich Gauss and Wilhelm Weber,
U. Göttingen, Germany, 1833



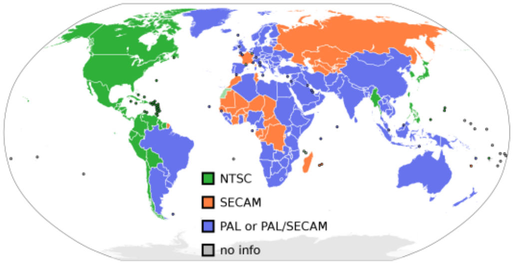
Carl Friedrich Gauss
1777-1855

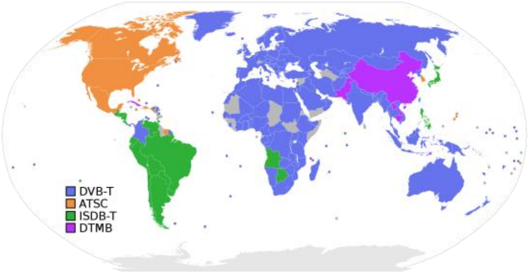
Wilhelm Weber
1804-1891

What was before streaming?

Video broadcast systems

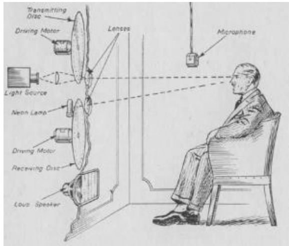
- Terrestrial, DHT satellite, Cable, hybrid.
- Several generations (from analog NTSC/PAL/SECAM in 1950a to digital ATSC/DVB/ISDB/TDMB in 1990s) been deployed
- They all used **purposedly built video distribution networks and receivers** to deliver video to the masses





Video conferencing systems

- 1927 – AT&T’s first demo of video phone
- 1959 – AT&T’s Picturephone (180p, 40kbps)
- 1976 – NTT, Mitsubishi AtariTel (48kbps)
- 1982 – CLI video phone system (first digital!)
- 1986 – PictureTel – first successful system
- 1990s – H.324 & H.323-based systems
- **Low-delay, 2-way comm. systems!**

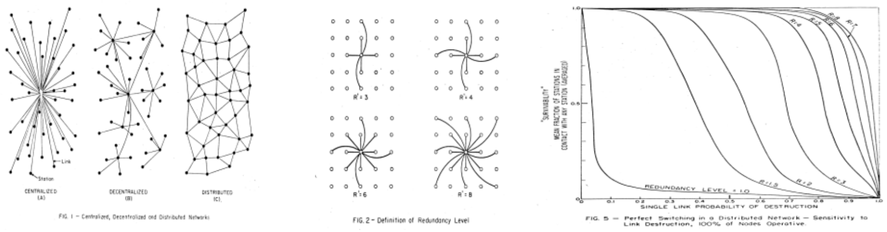


Evolution of Internet Streaming

Packet switched networks

Motivation 1: (Paul Baran, RAND, 1962):

- Packet switching is needed to increase the survivability (resilience) of tactical communication systems



Motivation 2: (Lenny Kleinrock, MIT, 1959)

- Circuit switching is problematic because data communications is bursty, that is, it is typically dominated by short bursts of activity with long periods of inactivity.
- Any static assignment of network resources, as is the case with circuit switching, would be extremely wasteful of those resources, whereas dynamic assignment (on-demand sharing) would be highly efficient.

Origins of packet switching:

L. Kleinrock, "Message Delay in Communication Nets with Storage," Ph.D. dissertation, MIT, Cambridge, MA, 1962.

P. Baran, "On Distributed Communication Networks," Rand Paper P-2626, Sept. 1962.

D. W. Davies, "Proposal for a Digital Communication Network," unpublished memo, June 1966,

D. W. Davies et al., "A Digital Communication Network For Computers Giving Rapid Response at Remote Terminals," ACM Symp. Op. Sys. Principles, Gatlinburg, TN, Oct. 1967.

D. W. Davies, "The Principles of a Data Communication Network for Computers and Remote Peripherals," Proc. IFIP Hardware, Edinburgh, 1968.

L. Kleinrock, "An early history of the internet [History of Communications]." IEEE Communications Magazine 48, no. 8 (2010): 26-36.

Leonard Kleinrock
CUNY, MIT, UCLA
1934 -

Paul Baran
UCLA, RAND Corp.
1926 - 2011

Donald Davies
Imperial College, NPL
1926 - 2011

First protocols for streaming

1973-77: NVP: Network Voice protocol

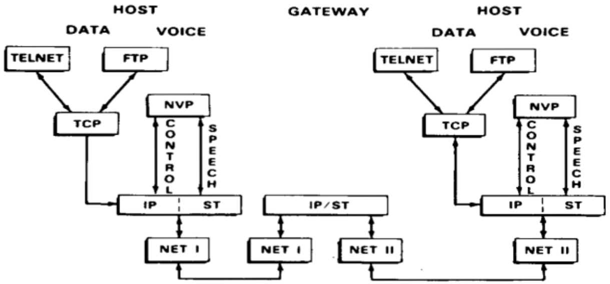
- Danny Cohen et al, USC, MIT Lincoln lab; RFC 741 (1977)
- Defines session control, capability negotiation, data transfer protocol
- Allows multiple codecs (vocoders) and data rates !!!

1976-79: TCP/IP split, addition of UDP

- Bob Kahn, Dave Reed, Dave Clark, Vince Cerf, Danny Cohen
- Initial TCP (Cerf & Kahn 1974) was split in 2 layers IP+TCP
- UDP added to support real-time traffic; RFC 768 (1980)

1979: ST: Internet Stream Protocol

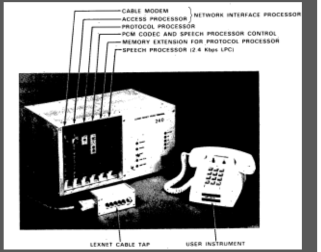
- Jim Forgie, MIT Lincoln lab; published as IEN119 (1979)
- Introduces an alternative layer to IP (IPv5)
- Introduces network-supported sessions and resource provisioning



First packet-based voice systems (1973-77)

Early voice terminal device built using NVP + ST.
MIT Lincoln Lab 1979.

C. Weinstein and J. Forgie, "Experience with Speech Communication in Packet Networks," JSAC 1/6,1983



Danny Cohen
Harvard, Caltech, USC, Sun
1937-2019

Jim Forgie
MIT Lincoln Lab
1929-2011

Progress in media coding

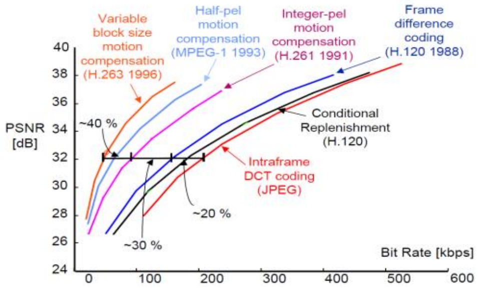
PCM, DPCM, LPC vocoders

- PCM: A. Reeves, 1939
- DPCM: C. Cutler, 1950;   ADPCM, N. Jayant, et al.1973
- LPC coding of speech, B.S. Atal & M.R. Schroeder, 1969

Transform-based codecs

- DFT & DHT-based image coding, Andrews & Pratt, 1968
- DCT-II and DCT-based coding, Ahmed, Natarajan, Rao, 1974
- DPCM+DCT-based coding, M. Schroeder 1972, A. Jain et al 1979+

1980s+: H.120, H.261, JPEG, MPEG codecs



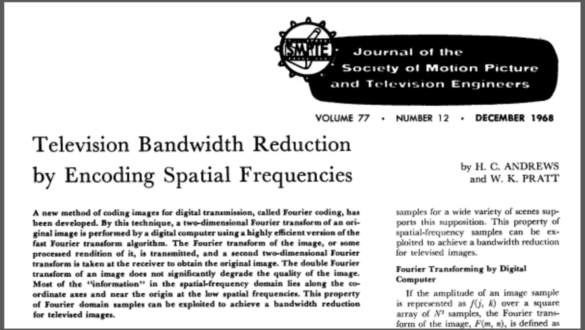
B. Girod, EE398B Image Communication II, Video Coding Standards, 2005.

1980s, and then 2020+:

- Learning-based codecs. End-to-end, hybrid, INR, GenAI-based, etc.

First transform-based codecs

H. C. Andrews and W. K. Pratt "Television Bandwidth Reduction by Encoding Spatial Frequencies", J. SMPTE, Vol. 77 (December 1968), pp. 1279-1281.



Journal of the Society of Motion Picture and Television Engineers

VOLUME 77 • NUMBER 12 • DECEMBER 1968

Television Bandwidth Reduction by Encoding Spatial Frequencies

by H. C. ANDREWS and W. K. PRATT

A new method of coding images for digital transmission, called Fourier coding, has been developed. By this technique, a two-dimensional Fourier transform of an original image is performed by a digital computer using a highly efficient version of the fast Fourier transform algorithm. The Fourier transform of the image, or some processed rendition of it, is transmitted, and a second two-dimensional Fourier transform is taken at the receiver to obtain the original image. The double Fourier transform of an image does not significantly degrade the quality of the image. Most of the "information" in the spatial-frequency domain lies along the coordinate axes and near the origin at the low spatial frequencies. This property of Fourier domain samples can be exploited to achieve a bandwidth reduction for televised images.

samples for a wide variety of scenes supports this supposition. This property of spatial-frequency samples can be exploited to achieve a bandwidth reduction for televised images.

Fourier Transforming by Digital Computer

If the amplitude of an image sample is represented as $f(j, k)$ over a square array of N^2 samples, the Fourier transform of the image, $F(m, n)$, is defined as

First neural-networks-based codecs

K. Sonehara, M. Kawato, S. Miyake, and K. Nakane, "Image data compression using a neural network model," *Int. Joint Conference on Neural Networks (IJCNN'89)*, Washington, DC, USA, 1989, pp. 35-41 vol.2,

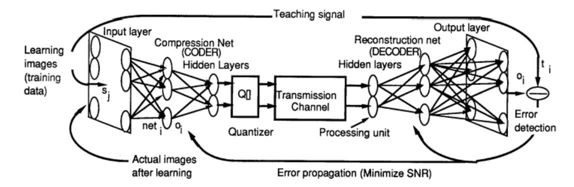
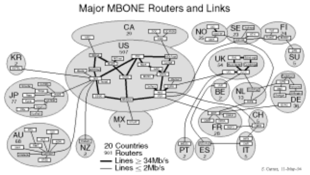


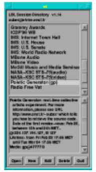
Fig. 1 A Neuro-CODEC Model

Early streaming systems

1993: MBONE

- Virtual multicast network connecting several universities & ISPs
- RTP-based video conferencing tool (vic) is used to send videos
- 1994 Rolling Stones concert – first major event streamed online





1995: RealAudio, 1997: RealVideo

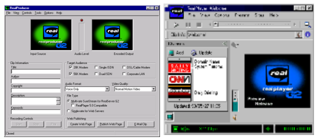
- First commercially successful mass-scale streaming system
- Proprietary protocols, codecs: PNA, RealAudio, RealVideo
- Worked over UDP, TCP, and HTTP ("cloaking" mode)
- First major broadcast: 1995 Seattle Mariners vs New York Yankees

1995+: VDOnet, Vivo, NetShow, VXtream, ...

- Many vendors have tried to compete in streaming space initially
- Vivo & Xing got acquired by Real, VXtreme by Microsoft
- By 1998, 3 main vendors remained: Real, Microsoft and Apple

1998: RealSystem G2

- First ABR streaming system

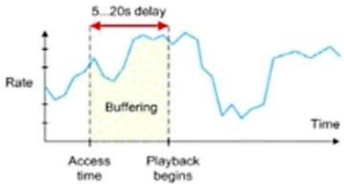


1990s: some key innovations

Introduction of long pre-roll delay

- Many early systems (Vivo, VDOnet, etc.) have tried to use H.324 / H.323- video conferencing stacks for streaming. But they worked very poorly!
- The first important discovery and deviation in the design of streaming systems from video conferencing was the *introduction of a much longer initial delay!*

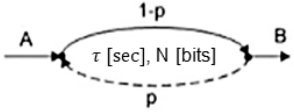
Initial delay:



Original uses of pre-roll delay / buffer

- Leaky bucket: reducing probability of stalls with network bandwidth fluctuations
- Reordering of out-of-order received UDP packets
- Limited retransmissions (ARQ) – unlimited ARQ or TCP was simply non-practical !
- Interleaving / multiple-description coding of audio

Expected delay & throughput in a system with unlimited retransmissions:

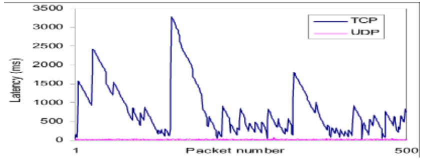


$$\bar{T}(A,B) = \sum_{i\geq 0}(1-p)p^i\,(1+i)\tau = \frac{\tau}{1-p}$$

$$\bar{R}(A,B) = \sum_{i\geq 0}(1-p)p^i\,\frac{N}{(1+i)\tau} = \frac{N}{\tau}\frac{(1-p)}{p}\log\left(\frac{1}{1-p}\right)$$

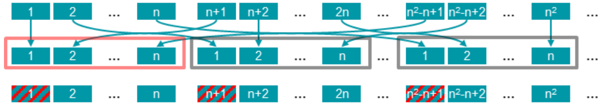
$$\Pr\left(R = \frac{N}{(1+i)\tau}\right) = (1-p)p^i$$

Observed delays: 56K modem, 10% pk. loss:



Interleaved packetization (RealAudio, 1995):

- 20-ms audio frames after encoder:
- UDP packets:
- Effects of loss of a packet:



- Missing audio frames were by-directionally predicted/synthesized during decoding.
- This worked remarkably well even with heavy (5-10%) packet loss rates!.

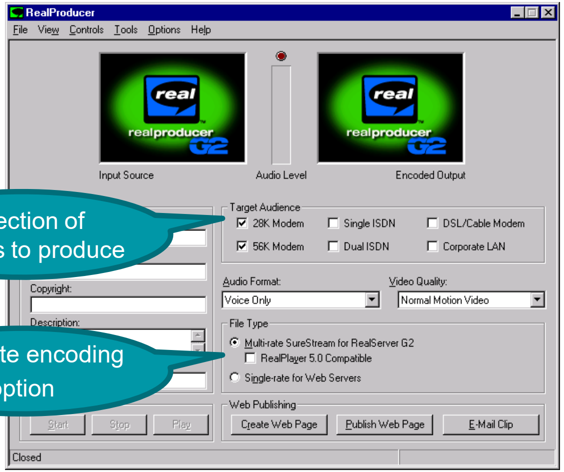
First ABR streaming system

1998: RealSystem G2: “SureStream”

- First commercially successful ABR streaming system
- Encoder:

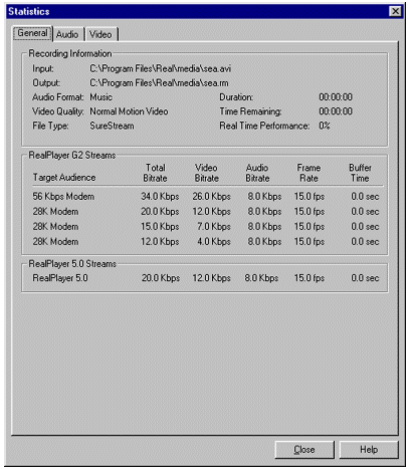
Encoded streams

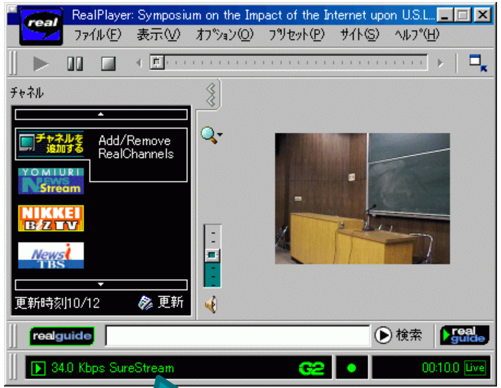
Player



Selection of streams to produce

Multi-rate encoding option





Panel showing which stream is selected

Related publications & patents

- B. Girod, et al, “Scalable codec architectures for Internet video-on-demand,” ACSSC, pp. 357 – 361, 1997.
- Y. Reznik, et al, “Video Coding for Streaming Media Delivery on the Internet," TCSVT, 11 (3), pp. 20-34, 2001.
- US Patents: 6314466, 6480541, 7075986, 7885340

RTP/RTSP streaming standards

1998: RTSP – Real-Time Streaming Protocol

- Session protocol for packet-bases streaming
- Main contributors: RealNetworks, Netscape, Columbia University
- Uses as foundation for most streaming systems of 1998-2008 era

2000: ISMA – Internet Streaming Media Alliance

- Forum created by Apple, Cisco, Kasenna, Philips, and Sun
- ISMA 2.0: RTSP+RTP+RTCP + H.264 and HE-AAC codecs
- ISBMFF with hint tracks is employed for storage of encoded streams
- ISMA 2.0 was supported by many servers and clients of that era

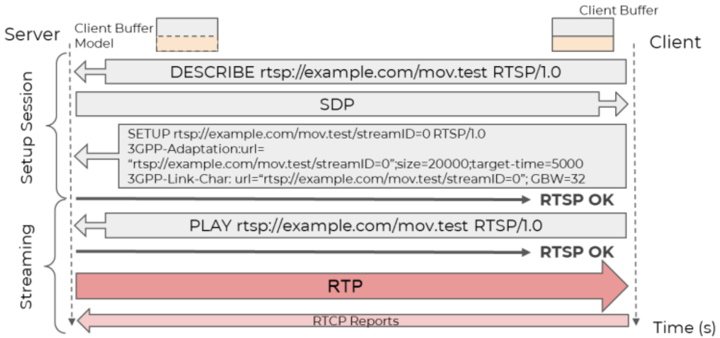
2006: 3GPP PSS – Packet Switched Streaming

- Describes RTSP+RTP+RTCP ABR adaptive streaming system with several standard video, audio and speech codecs
- 3GPP version of RTSP/RTP-based stack

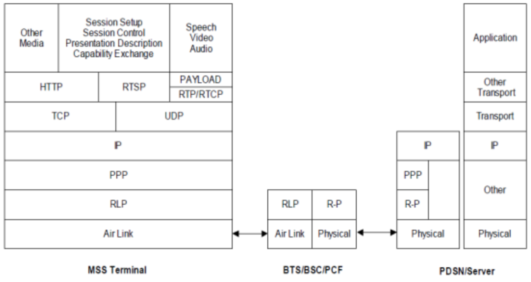
2006: 3GPP2 MSS – Multimedia Streaming Services

- Similar to 3GPP PSS, but differs in speech codecs & network stack

Session setup and streaming phases:



Full protocol stack in 3GPP2 MSS:



2000s: Transition to HTTP

Networks have improved!!

- When streaming started, 28k and 56k modems were the common connections available
- But by mid-2000s consumers moved to Cable, DSL, or other high-speed connections
- Bitrates went up 5-100x, latencies went down 4-10x, packet losses dropped to under 1-2%
- This relaxed requirements dramatically!
- Progressive downloads become feasible alternatives to streaming!

CDNs become ubiquitous

- By mid-2000s Akamai, Limelight and few other CDNs were well deployed
- CDNs provided better density and reach than RTSP-based delivery networks (RBN, etc.)

Other practical & business reasons

- The space was fragmented: Real, Microsoft, Apple, and then Adobe used significantly different implementations of their stacks. Even codecs and file formats were different! RTSP and ISMA offered only some basic level of interoperability!
- RTSP systems were complex: servers and clients were extremely complex, error concealment was a major pain, etc.

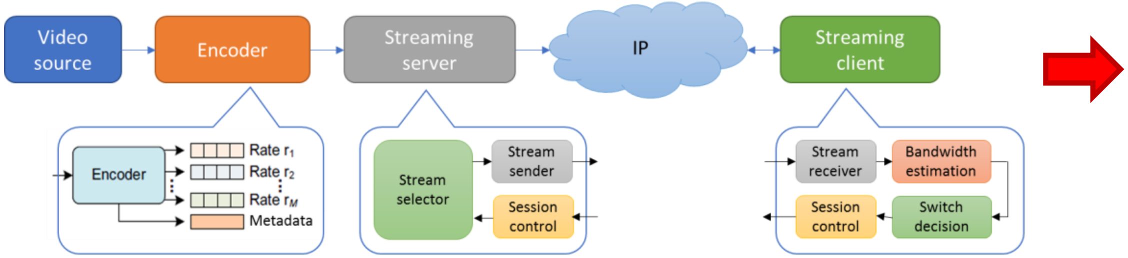
And… one day a much simpler solution was found

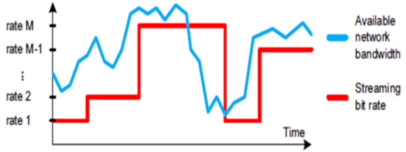
- Store encoded media streams in 5-10sec chunks on a web server… pull them using HTTP GET, catenate, and play
- About same delays, no packet losses or retransmissions, and with good enough networks – it may just work.

ABR systems & their evolutions

How first ABR system worked?

RTP/RTSP-based ABR streaming architecture:





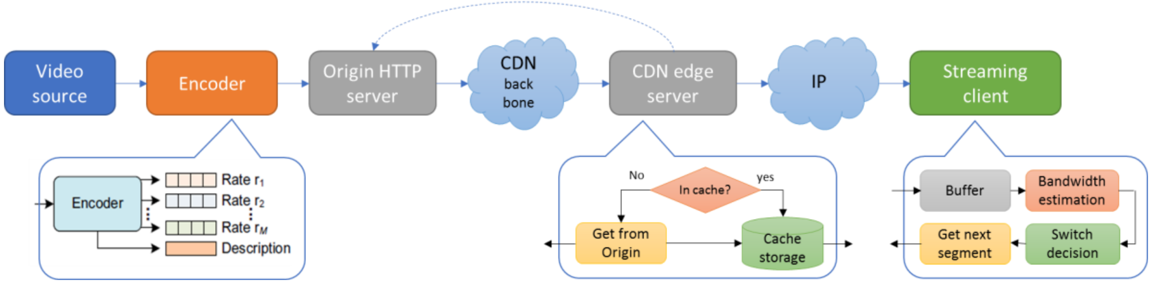
- Public internet is used for delivery
- RTSP was used for session control, and UDP (plus RTP or proprietary transport) for sending the data
- The server was driving stream adaptation. Client-driven switching was tried in some applications, but it was less common.
- Server was also responsible for retransmissions, injecting extra FEC packets, etc.
- Everything was sent in “packets”

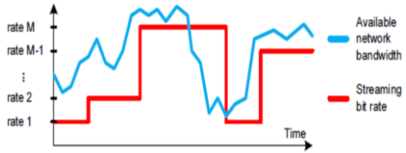
Important design elements:

- Only one stream was sent over IP for delivery to each client!
- Multiple renditions were stored only on the (origin) streaming server, and transmissions of such “stacks of streams” to other servers were not envisioned.
- With early RTP/RTSP distribution networks, the relays carried only single-rate streams.

HTTP-based ABR Streaming

Modern-era HLS/DASH streaming architecture:





Key differences from RTSP/UDP streaming:

- instead of 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 the edge servers, reducing the load on the origin...

Important new factors:

- This works well when the content is *"popular"* and it becomes *cached* in the edge cache
- If content is not popular, and not stored at the edge cache – it becomes pulled from the origin server (in which case CDN only adds latency and increases cost of delivery)
- In other words – *CDN helps a lot on average*, but in the worst case – it does not.

Disconnect between ABR and CDN models

Key issues:

- ABR systems fundamentally need **several encoded versions of the content:**
  - Multiple streams are needed to achieve better network adaptation and minimize the visibility of stream switches.
  - Multiple streams are also needed to support different delivery formats (HLS, DASH, MSS, etc.) and DRM systems.
  - Support for multiple video codecs (H.264, HEVC, AV1, and VVC) also results in a creation of multiple streams

- However, once multiple streams are created, and different client start pulling different versions of then – such streams start **"competing" for the CDN edge cache disk space.** This results in mode CDN cache misses, and higher load on origin server. This also increases delivery costs and makes whole system less reliable, less scalable, etc.

- In other words, while **ABR streaming concept promotes the creation of "more" streams, what CDNs need to be the most effective is "less"!**

Effects of multiple streams

Effects on cache miss probability:

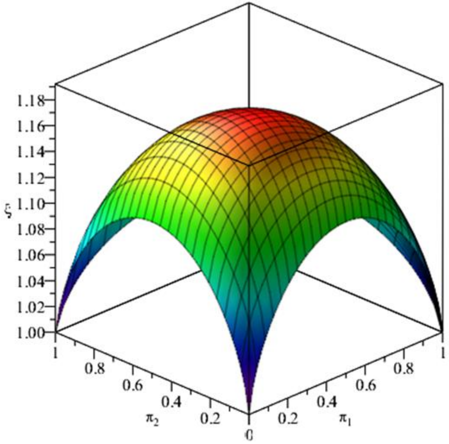
- Sending k variant streams increases CDN cache miss probability by a factor [1]

$$\xi(\alpha,\pi)=\frac{p_{miss,k}(C,\alpha,\pi)}{p_{miss}(C,\alpha)}\sim\left(\sum_{i=1}^{k}\pi_i^{\frac{1}{\alpha}}\right)^{\alpha}=\|\pi\|_{\frac{1}{\alpha}}$$

Here: α is a parameter of content popularity model, and $\pi=\{\pi_1,\ldots,\pi_k\}$ are the usage probabilities of each stream

[1] *Y. Reznik et al, "On multiple media representations and CDN performance", MHV 2022*

Relative increase in cache miss probability in case of using 3 formats.



Observations:

- The worst impact happens when all formats are equally probable: $\pi_1=\cdots=\pi_k$
- The higher is the asymmetry in usage of different formats (or renditions), the better it is from CDN efficiency standpoint: $\pi_i\to 1\Rightarrow\ \xi(\alpha,\pi)\to 1$

Possible solutions / workarounds:

- Reduce the number of streams;
- Pick one "preferred" representation, and direct as many possible clients/devices use it
- Consider alternatives to "simulcast ABR": scalable coding, multiple description, etc.

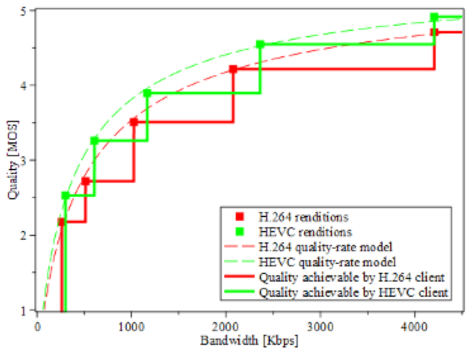
New Codecs. Multi-codec systems

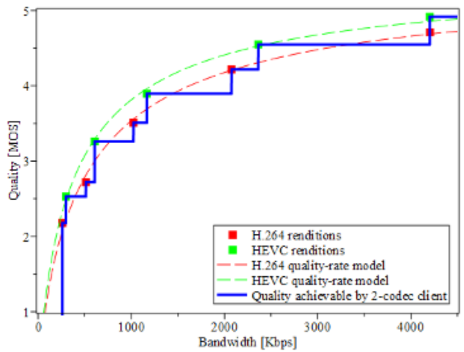
Multiple codecs bring more problems to CDNs:

- Even as newer codecs are getting better, adding new streams to CDNs may increase delivery costs instead of reducing them.
- Old streams must be retained for compatibility with older systems.

Smart multi-codec ABR ladders:

- ABR ladder generation with 2+codecs and interleaved bit-allocation ➔ saves the total number of streams needed[1]





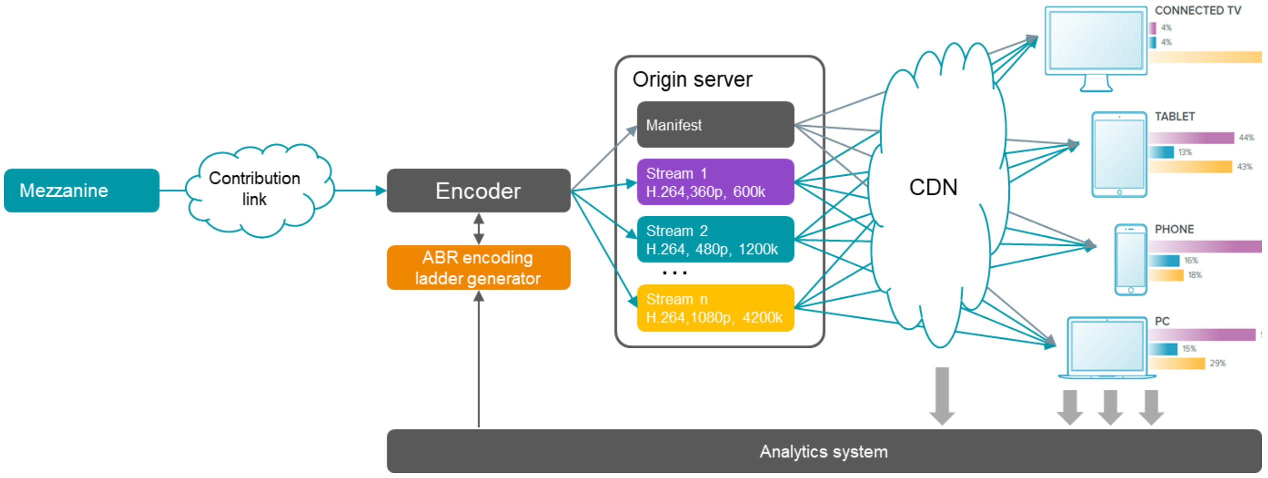
[1] *Y. Reznik, et al, "Towards Efficient Multi-codec streaming", NAB 2022:*

Is this the ultimate solution?

- Indeed no! Codecs fragmentation is a human-created problem!
- Better technical solution: force convergence to the same codec!

Optimizations by ABR ladder construction

With ABR systems, the ladder design emerges as key for end-to-end optimization:



ABR ladder design techniques:

- Per-title or "content-aware" → takes into account only properties of content
  - A. Aaron, et al, "Per-title encoding optimization", Netflix Tech. Blog, Dec. 2015
- Playback statistics or "networks-aware" → take into account playback statistics as a basis for optimization
- "Context-aware" → takes into account both properties of content, as well as its popularity and CDN- and network-related statistics
  - Y. Reznik, et al, "Optimal design of encoding profiles for ABR streaming", Packet Video, 2018
  - Y. Reznik, et al, "Optimizing Mass-Scale Multiscreen Video Delivery," SMPTE Motion Imaging Journal, vol. 129, no. 3, 2020

What may come next?

Future evolutions

New forms of video

- SD->HD->UltraHD, SDR->HDR, 30 degrees -> 360 degrees
- 2D/single view->stereoscopic->multi-view->light field representations
- Higher framerate videos: 30fps->60fps->120fps…1000fps?
- Real world -> metaverse, “GenAI-universe”
- Dependencies: displays, cameras, graphics stacks, and only then delivery systems

Towards lower delays

- HLS/DASH: 10-30sec
- Low-latency HLS/DASH: 3-6 sec
- Back to UDP: WebRTC, QUIC, MOQ: 200-500ms
- Cross-layer Phy->App stacks: 30-100ms (subject to distance, topology, etc.)
- Extreme low-delay case:
  - If ultra-ultra-low delay (~30ms) becomes achievable, then we don’t need much bandwidth!
  - All we need to send is about 1-2 degrees spot at each moment! [foveated video, eye-tracking-based systems]
  - Perceptually perfect transmission can be accomplished at about 700kbps or less

Back to video-centric design of the network?

- Internet streaming have evolved as a technology for sending video over networks initially built for sending data
- But nowadays video is already consuming over 80% of Internet bandwidth!
- Internet is becoming the “video-first” network.. or maybe “GenAI-first” !

THANK YOU