Course: Networking II

Topic: Video streaming over the Internet

Eurecom 2014-2015

Lucile Sassatelli

sassatelli@i3s.unice.fr

Sources

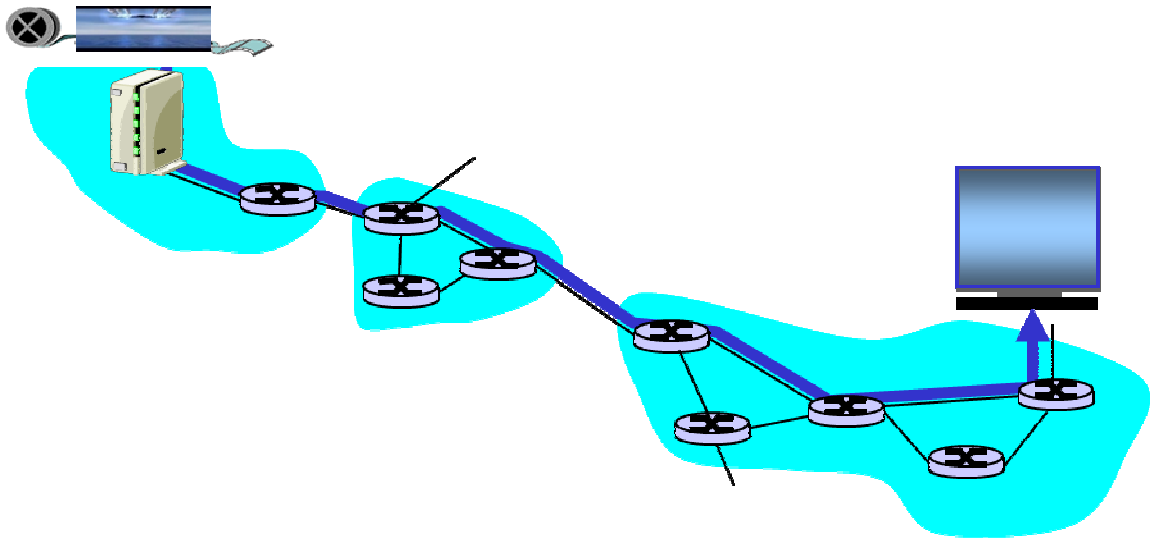
- G. Urvoy-Keller and L. Sassatelli, *Video streaming*, lecture M2 Ubinet, UNS, 2014
- M. Siekkinen, *Video streaming*, lecture Aalto Univ., 2014
- H. Riiser, *Adaptive Bitrate Video Streaming over HTTP in Mobile Wireless Networks*, PhD thesis, Univ. of Oslo, 2013
- Z. Morley Mao, *Multimedia Networking*, lecture, lecture Univ. of Michigan
- *Video Communications and Video Streaming Over Internet: Issues and Solutions*, lecture Sharif Univ. of Technology
- A. C. Begen and T. Stockhammer, *HTTP Adaptive Streaming: Principles, Ongoing Research and Standards*, ICME 2013
- S. Akhshabi, S. Narayanaswamy, A. C. Begen, C. Dovrolis, *An experimental evaluation of rate-adaptive video players over HTTP*, Elsevier SP, Oct. 2011
- And others along the way…

Outline

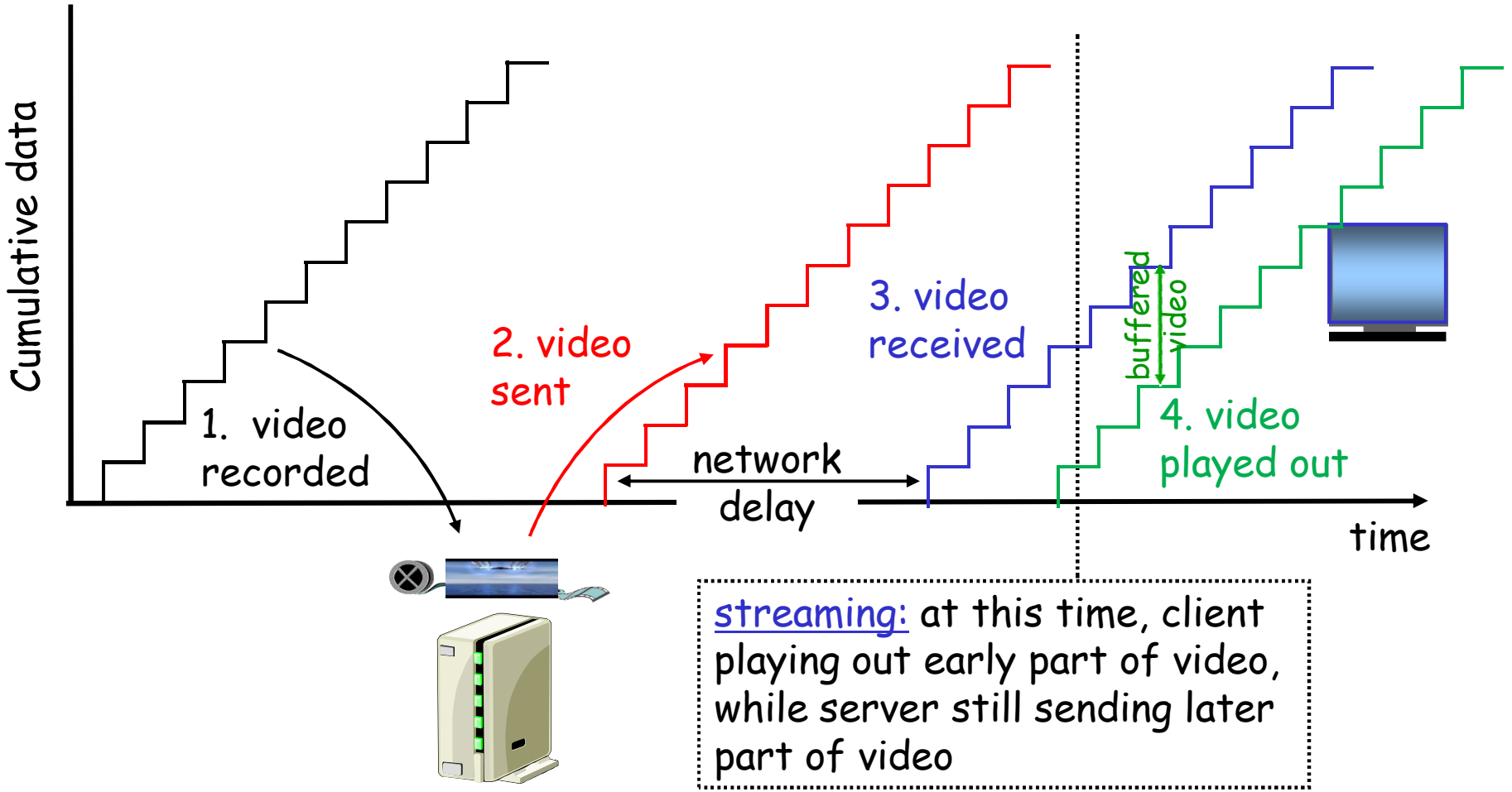
- A general overview of HTTP-based video streaming
  - Principle
  - Current traffic figures
  - Basic architecture
- Client side
  - Quality of Experience: the metrics that matter
  - Various HTTP-based streaming strategies
  - HTTP Adaptive Streaming (HAS)
- Server side
  - Content Distribution Networks (CDNs)
  - Impact of CDN management on QoE

What is streaming?

- **Streaming**: the media stored at a source is transmitted to the client. The client playout starts before all the data has arrived.
- Timing constraint for still-to-be transmitted data: in time for playout
- Notation:
  - "video rate"="bitrate"=the rate at which the played video has been encoded
  - "bandwidth"="downloading rate"="sending rate"= the data rate achieved between the source and destination nodes



Streaming Stored Multimedia: What is it?



Video Streaming Applications

1) Live, interactive video

   Video teleconferencing, video phones, etc.

2) Live, non-interactive video

   Course lectures, news, sporting events, conferences

3) Stored, non-interacive video

   Movies, distance learning, Web videos, etc.

- In this course, we focus on 3): streaming of stored data.

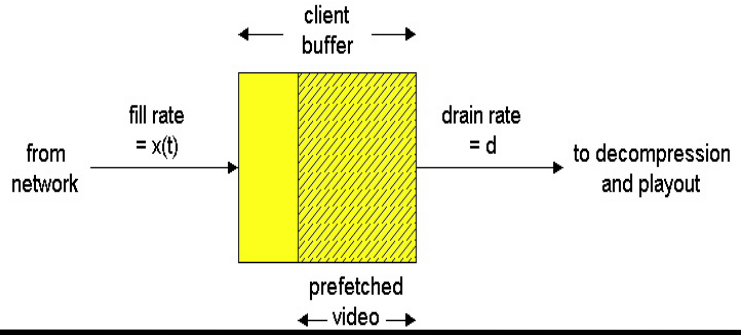
QoS constraints for video streaming

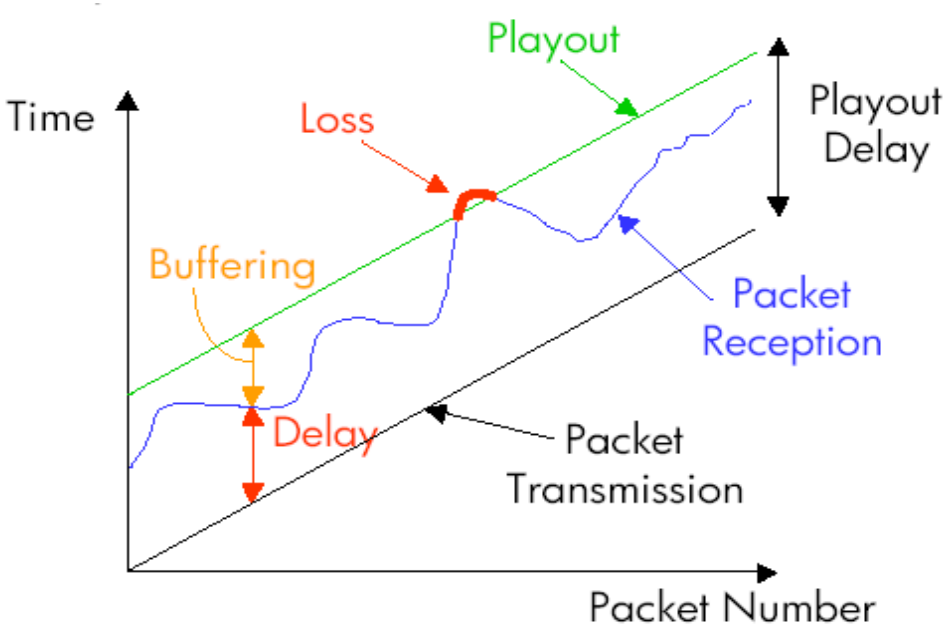
- Problem: Internet only offers best-effort service
- No guarantees on:
  - Bandwidth
    - Can not reserve bandwidth in Internet today
    - Available bandwidth is dynamic
    - If transmit faster than available bandwidth, then congestion occurs, packet loss, and drop in video quality
    - If transmit slower than available bandwidth, then sub-optimal video quality
    - Goal: Match video bit rate with available bandwidth
  - Loss rates
  - Delay jitter
- Specifically, these characteristics are **unknown** and **dynamic**
- Goal: Design a system to reliably deliver high-quality video over the Internet

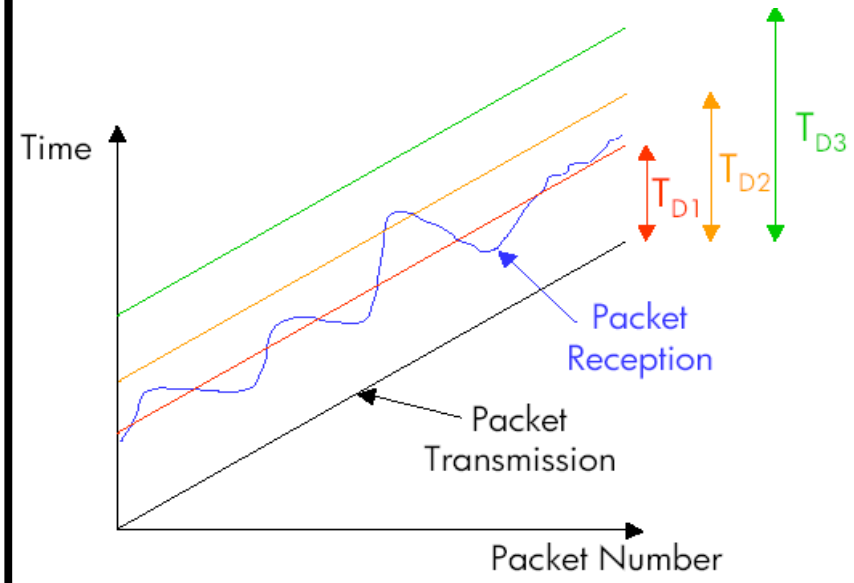
Overcoming Internet best-effort quality: Playout buffer

- Meant to compensate for bandwidth variations, delay jitter and enables retransmission of lost packets.
- Corresponds to adding an offset to the playout time of each packet
  - If (packet delay < offset) then OK
  - Buffer packet until its playout time
  - If (packet delay > offset) then problem
- Playout buffer typically 30 secs to several minutes of playback

Overcoming Internet best-effort quality: Playout buffer







Effect of different buffering delays

Outline

- A general overview of HTTP-based video streaming
  - Principle
  - Current traffic figures
  - Basic architecture
- Client side
  - Quality of Experience: the metrics that matter
  - Various HTTP-based streaming strategies
  - HTTP Adaptive Streaming (HAS)
- Server side
  - Content Distribution Networks (CDNs)
  - Impact of CDN management on QoE

Sandvine traffic reports

- Following info is extracted from Sandvine reports
  - Semiannual reports on Internet evolution in terms of traffic and usage too
- Sandvine is a Canadian company
  - Customers are tier-1 and tier-2
  - Products to monitor traffic, optimize content delivery
- Idea is to get an understanding of the role of video delivery in the Internet

Fixed Access: traffic breakdown

	Upstream		Downstream		Aggregate	
Rank	Application	Share	Application	Share	Application	Share
1	BitTorrent	24.53%	Netflix	34.21%	Netflix	31.09%
2	HTTP	14.27%	YouTube	13.19%	YouTube	12.28%
3	SSL	6.54%	HTTP	11.65%	HTTP	11.84%
4	Netflix	6.44%	iTunes	3.64%	BitTorrent	5.96%
5	YouTube	5.52%	SSL	3.42%	SSL	3.80%
6	Skype	2.23%	BitTorrent	3.40%	iTunes	3.33%
7	Facebook	2.17%	MPEG	2.85%	MPEG	2.62%
8	FaceTime	1.50%	Facebook	1.99%	Facebook	1.83%
9	Dropbox	1.20%	Amazon Video	1.90%	Amazon Video	1.82%
10	iTunes	1.15%	Hulu	1.74%	Hulu	1.58%
		64.40%		76.24%		74.58%

sandvine

Table 2 - Top 10 Peak Period Applications - North America, Fixed Access

	Upstream		Downstream		Aggregate	
Rank	Application	Share	Application	Share	Application	Share
1	BitTorrent	33.20%	YouTube	19.27%	YouTube	17.38%
2	HTTP	10.07%	HTTP	17.46%	HTTP	16.26%
3	YouTube	7.67%	BitTorrent	11.10%	BitTorrent	14.71%
4	SSL	5.63%	SSL	6.19%	SSL	6.10%
5	Skype	4.54%	Facebook	3.88%	Facebook	3.95%
6	Facebook	4.29%	RTMP	3.66%	RTMP	3.27%
7	eDonkey	3.64%	MPEG	3.54%	MPEG	3.21%
8	Dropbox	2.11%	Netflix	3.23%	Netflix	2.98%
9	MPEG	1.51%	Flash Video	2.37%	Flash Video	2.17%
10	iTunes	1.30%	iTunes	2.23%	iTunes	2.08%
		72.66%		70.69%		70.01%

sandvine

Table 6 - Top 10 Peak Period Applications - Europe, Fixed Access

- Video streaming dominates almost everywhere
  - Netflix clearly the biggest (31%) in North America followed by YouTube (12%)
  - YouTube is the largest in Europe (17%) as well as in Latin America (26%)
    - Netflix gaining ground rapidly
  - BitTorrent traffic still largest (27%) in Asia-Pacific
- P2P has declined sharply (except in Asia-Pacific)
  - Now less than 10% of total traffic, was 31% in 2008

Streaming user behavior

- Deeper analysis of fixed access usage in the US
- Classified into three types of streaming service users
  - 15 % of users contribute 54 % of traffic
  - 15 % less active contribute 0.5 % of traffic

Top 15th percentile

15th – 85th percentile

Bottom 15th percentile

	"Cord Cutter"	Typical Subscriber	"Non-Streamer"
Mean Monthly Usage	212 GB	29 GB	4.5 GB
Mean Real-Time Entertainment Usage	153 GB	13 GB	40 MB
Streaming Share	72%	45%	1%
Average Hours of Streaming	100 !!	9	<1
Share of Total Traffic	53.9%	45.7%	0.5%

Mobile streaming traffic

	Upstream		Downstream		Aggregate	
Rank	Application	Share	Application	Share	Application	Share
1	Facebook	26.95%	YouTube	17.61%	YouTube	17.26%
2	SSL	12.49%	Facebook	14.03%	Facebook	14.76%
3	HTTP	11.80%	HTTP	12.70%	HTTP	12.59%
4	YouTube	3.77%	MPEG	8.64%	MPEG	7.77%
5	Instagram	3.47%	SSL	6.52%	SSL	7.25%
6	BitTorrent	2.09%	Google Market	5.27%	Google Market	4.78%
7	MPEG	1.70%	Pandora Radio	5.15%	Pandora Radio	4.72%
8	Pandora Radio	1.61%	Netflix	5.05%	Netflix	4.55%
9	Gmail	1.61%	Instagram	3.49%	Instagram	3.49%
10	iCloud	1.56%	iTunes	3.10%	iTunes	2.84%
		65.50%		78.46%		77.17%

sandvine

Table 4 - Top 10 Peak Period Applications - North America, Mobile Access

- Real-time entertainment (read: streaming) is clearly the most dominant traffic category (over 30%) in North America, Europe, and Asia-Pacific (over 40%)
  - Comes second in South America behind Social Networking
  - Very small share of traffic in Africa

Role of video streaming in the Internet

- Overall, it is dominant and growing fast
- Forecasts: Globally, consumer IP video traffic will be 83% of consumer IP traffic in 2018, up from 73% in 2013 [Cisco visual index 2013-2018]
- Mobile streaming growth will be accelerated by
  - continuing 4G deployments
  - evolution towards 5G (1000x capacity)
- The importance is huge to ISPs (both fixed and mobile)
- Lots of money involved in the business
  - Video and CDN providers

A few words about video compression

Examples:

- MPEG 1 (CD-ROM) 1.5 Mbps
- MPEG2 (DVD) 3-6 Mbps
- MPEG4 (often used in the Internet, < 1 Mbps)

New trends:

- **Ultra High Definition, a.k.a. 4K** (because it packs 4 times as many pixels as 1080p): new identified trend in video distribution at the 2014 CES.
- Main content provider like Netflix and YouTube plan to offer it, and the demand for streaming is expected to increase because such a resolution does not fit onto a Blu-ray anymore.
- Required bandwidth for 4K videos: **15Mbps** -> the residential access is expected to quickly grow -> **4K is foreseen to put a heavy strain on ISPs**

Extensions:

- Layered (scalable) video
  - adapt layers to available bandwidth
  - Multiple Description Codes (MDC)
  - Scalable Video Coding (SVC): Annex G extension of the H.264/MPEG-4 AVC video compression standard

Outline

- A general overview of HTTP-based video streaming
  - Principle
  - Current traffic figures
  - Basic architecture
- Client side
  - Quality of Experience: the metrics that matter
  - Various HTTP-based streaming strategies
  - HTTP Adaptive Streaming (HAS)
- Server side
  - Content Distribution Networks (CDNs)
  - Impact of CDN management on QoE

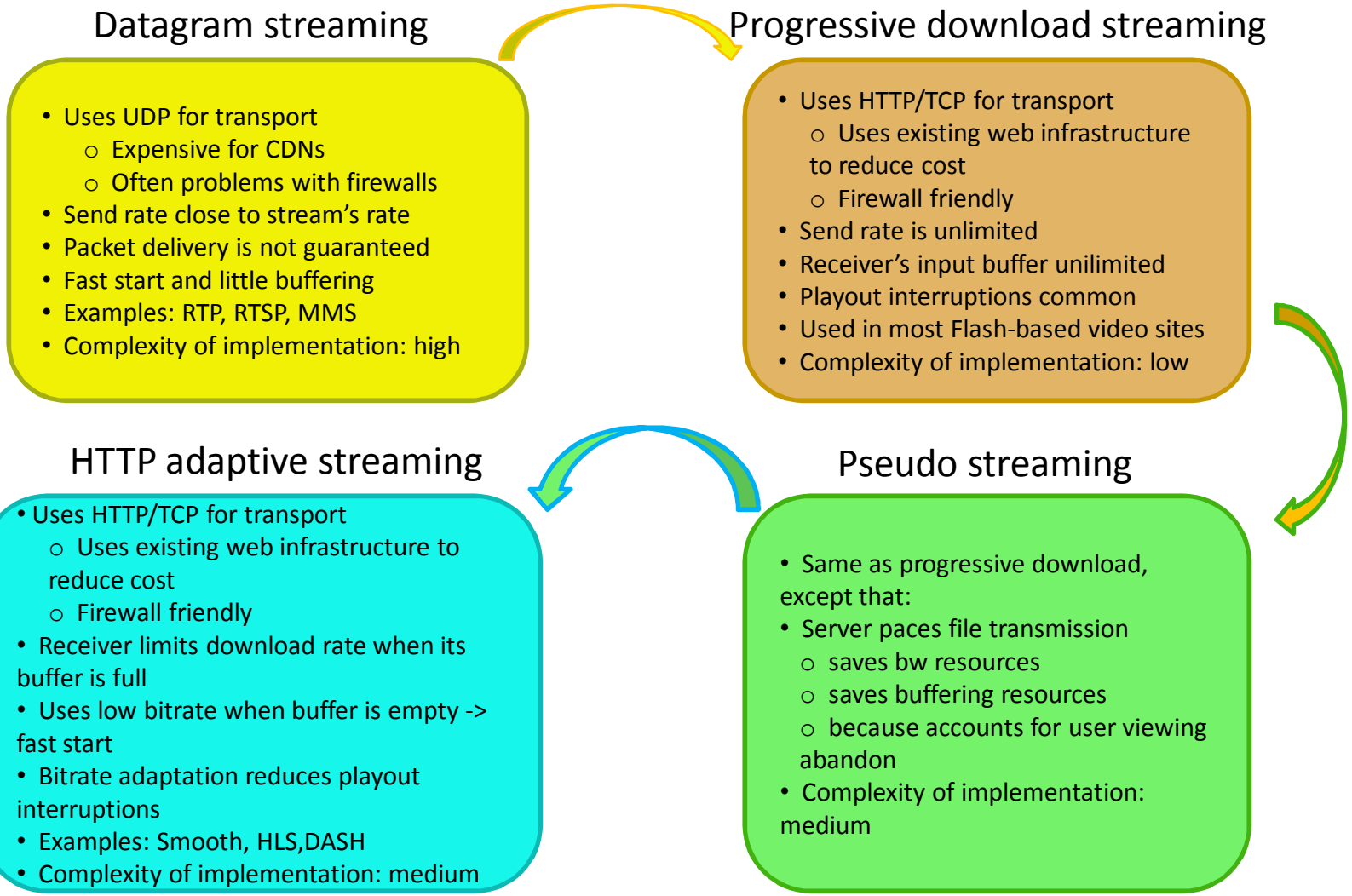
A brief history of video streaming

- Used to be admitted that real-time video over a best-effort network like the Internet would have to be streamed using a datagram protocol: giving the streaming application packet-level control
- -> this meant in practice that it should be carried by UDP, not TCP:
  - RTP streaming systems can have full control over packet retransmission -> can trade packet loss for delay (e.g., audio and video conferencing). Problem with RTP: new media codecs cannot easily be supported.
  - RTP requires out-of-band signaling (RTSP-Real Time Streaming Protocol , and SIP)
- In addition to this, protocols based on datagram streaming suffer from:
  - Packet-level control -> very complicated implementation as needs to deal with flow and congestion control, packet loss, out-of-order delivery, etc.
  - Firewalls and NAT routers frequently cause problems with UDP
  - Higher cost of the infrastructure as Content Delivery Networks (CDNs) require specialized solutions for caching and load balancing (almost all deployed infrastructure optimizations target HTTP because of its massive popularity)
- => most of the industry adopted progressive download streaming using HTTP
- Its simplicity has made it the streaming protocol most commonly used today (e.g., by YouTube).

Whereby HTTP-based streaming

- With this approach, the client simply downloads a media stream as a file in a normal media container format such as MP4, and plays back the video while it is downloading.
- Benefits:
  - firewall traversing
  - all CDNs support it (can automatically take advantage of transparent web caching)
- Downsides:
  - playout interruptions are more likely to occur
  - larger buffer is required (limiting progressive streaming's suitability for real-time communication)
  - multicast is not an option (no longer considered a big loss, since there is no widely available multicast infrastructure)
- -> The biggest arguments for datagram protocols for non-interactive streaming is now mostly irrelevant

Evolution from datagram streaming to adaptive HTTP streaming



Outline

- A general overview of HTTP-based video streaming
  - Principle
  - Current traffic figures
  - Basic architecture
- Client side
  - Quality of Experience: the metrics that matter
  - Various HTTP-based streaming strategies
  - HTTP Adaptive Streaming (HAS)
- Server side
  - Content Distribution Networks (CDNs)
  - Impact of CDN management on QoE

Quality of Experience in video streaming

- QoE: this is the quality the client **perceives**
- -> subjective, and maybe given by complex functions of the QoS metrics
  - Video quality, does it play smooth, how long does it take to start, …
- Different stakeholders
  - ISP: The one providing client's access to Internet
    - often the one customers blame for poor QoE
  - Video service provider: YouTube, Netflix, …
  - CDN operator: YouTube, Akamai, Limelight networks, Level 3, …
  - Client: you, the one who matters
  - Device manufacturer: hardware performance
  - Client software provider: OS and player software
- What determines QoE, i.e. who is in control?
  - Jointly the behavior of all the above stakeholders
  - Outcome of a rather complex process

QoE metrics for video streaming

- Buffering events
  - If playback buffer runs dry, playback pauses -> annoys the user
  - Happens if download rate is smaller than encoding rate for too long
  - Initial buffering attempts to avoid such events
  - Studies suggest this is the most important reason for abandoning viewing
- Initial startup delay (a.k.a. joining time)
  - Time it takes for playback to begin after clicking play
  - Not as fatal as buffering events for audience retention

QoE metrics for video streaming

- More “traditional” quality metrics
  - Video resolution, frame rate, target encoding rate (quality)
  - PSNR
  - Subjective metrics (e.g. MOS)
  - ...
- Adaptive streaming metrics
  - Rate and amplitude of quality switches
  - Avg quality

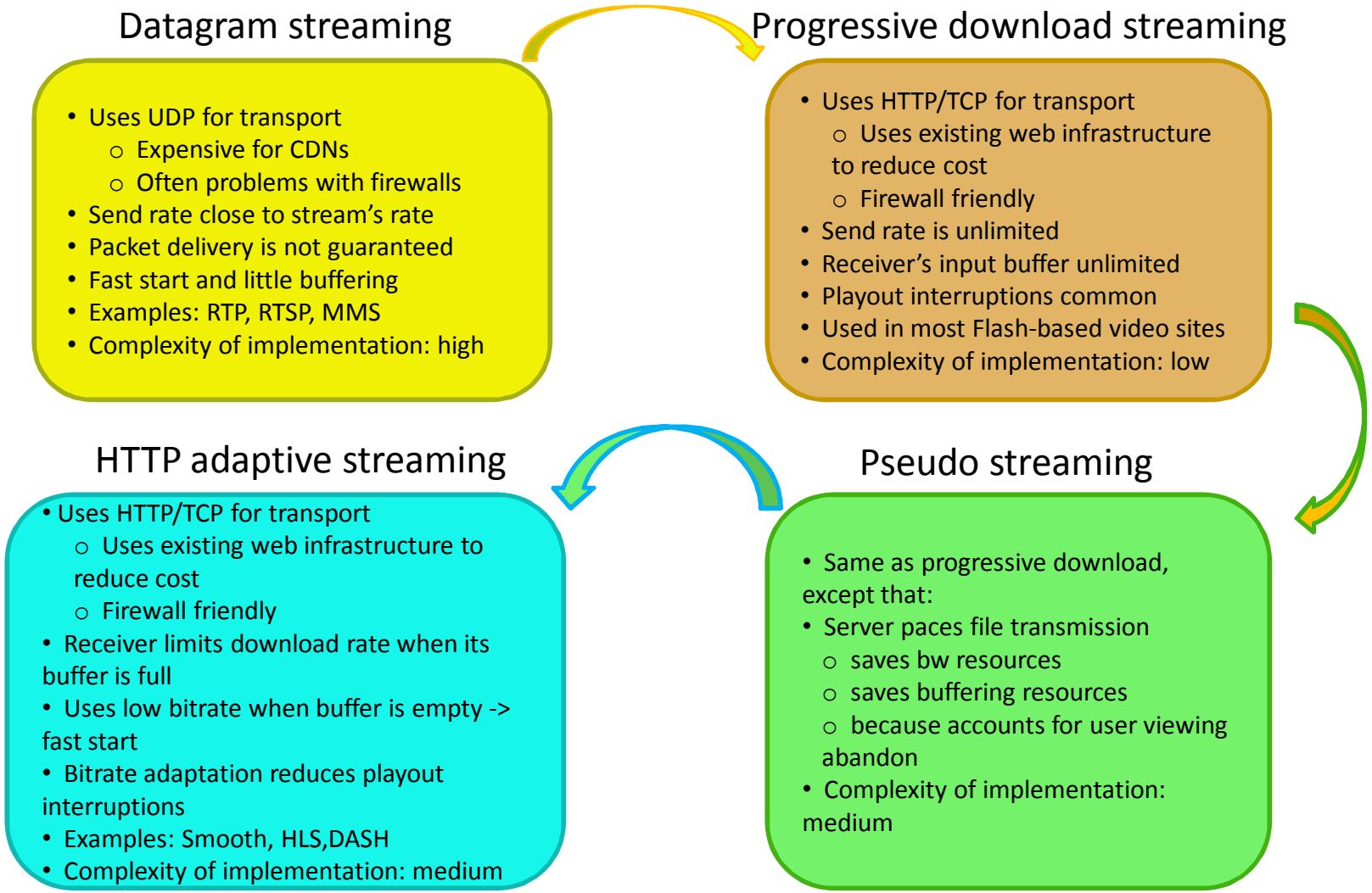
Delivery related QoE metrics

- Imbalance between download rate vs. encoding rate
- Several factors influence the probability of experiencing buffering event
  - Amount of initially buffered content
  - Instantaneous TCP bulk transfer capacity from server to client
  - Content download strategy, i.e. streaming strategy

Outline

- A general overview of HTTP-based video streaming
  - Principle
  - Current traffic figures
  - Basic architecture
- Client side
  - Quality of Experience: the metrics that matter
  - Various HTTP-based streaming strategies
  - HTTP Adaptive Streaming (HAS)
- Server side
  - Content Distribution Networks (CDNs)
  - Impact of CDN management on QoE

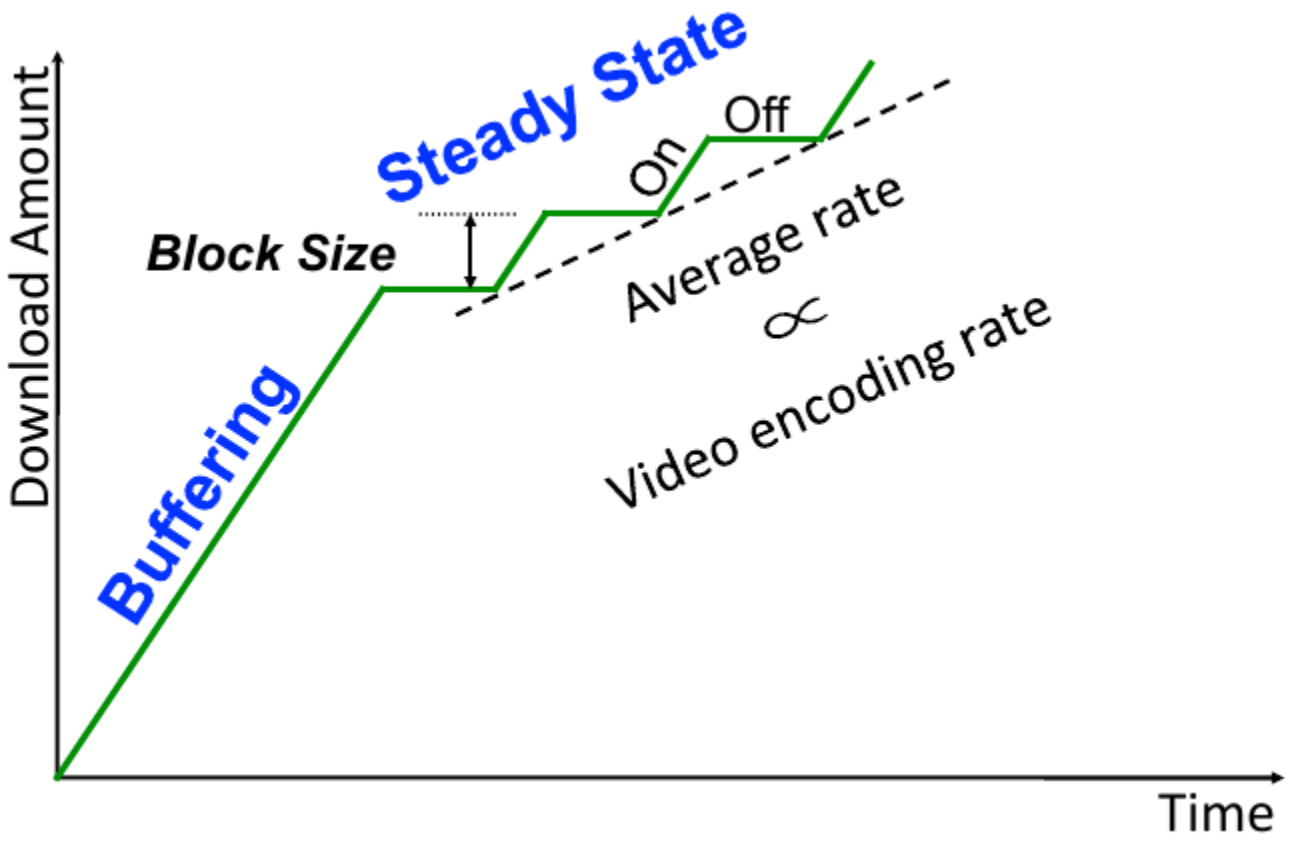
Evolution from datagram streaming to adaptive HTTP streaming



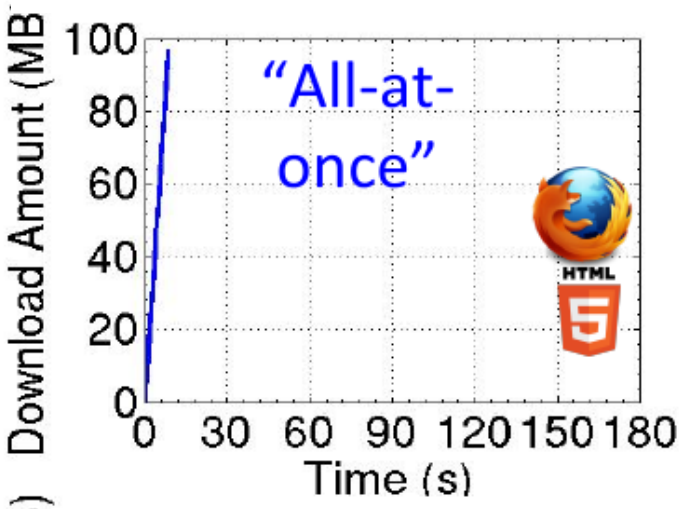
Streaming strategy

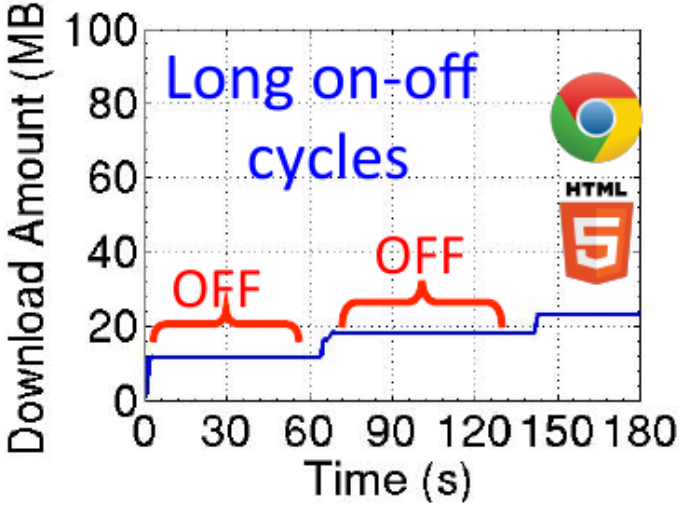
- Turns out that a number of different strategies are used
  - Interplay between server and client application
- We'll borrow some results from earlier studies
  - *A. Rao et al: Network Characteristics of Video Streaming Traffic. CoNEXT 2011.*
  - *M. Hoque et al: Mobile Multimedia Streaming Techniques: QoE and Energy Saving Perspective. Pervasive and Mobile Computing 2014.*

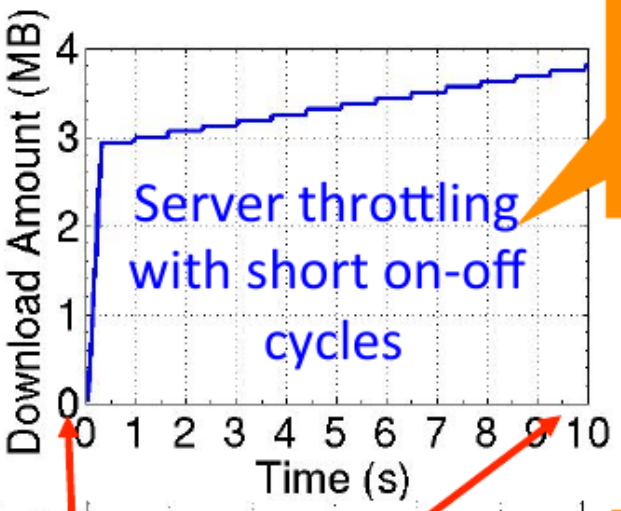
Generic behavior: « pseudo streaming »



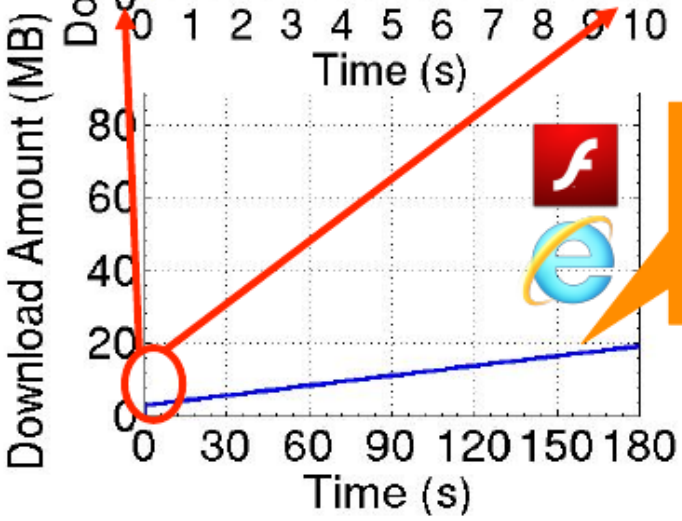
Identified streaming strategies







Server enforces a download rate control (e.g: 1.5x or 2x encoding rate)

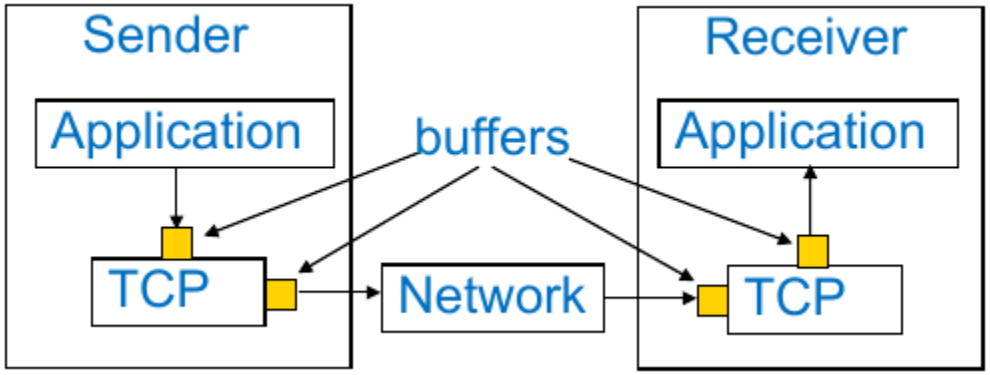


Also *encoding rate* streaming possible: dl at encoding rate in steady state phase

Streaming strategies vastly different

Who chooses the strategy?

- Let's quickly review the interplay between applications and TCP



Who chooses the strategy?

- All-at-once and server throttling are **enforced by server**
  - But all-at-once may be explicitly requested by client application (HTTP request parameter)
- Long ON-OFF cycles are (usually) **caused by client application**
  - Client application periodically stops reading from TCP socket
  - TCP receive buffer fills up -> TCP flow control activates and forbids server side TCP to send more packets
  - Transfer paused until application reads again from socket
- Encoding rate streaming seems to be **unintentional**
  - Client's application buffer fills up -> receive TCP buffer fills up -> TCP flow control activates
  - Client application reads data from socket at encoding rate -> new data transferred at this rate

Streaming strategies in use

Service	YouTube			Netflix
Container	Flash	HD (Flash)	HTML5	Silverlight
IE 9	Short	No	Short	Short
Firefox	Short	No	No	Short
Chrome	Short	No	Long	Short
iOS (native)	-	-	Based on encoding rate	Short
Android (native)	-	-	Long	Long

Streaming strategy differs with application type and container
Clients do not throttle the rate of transfer in wired and wifi!

Strategies used by mobile clients

Table 3
Streaming techniques for popular video streaming services to mobile phones of three major platforms. The selection of a streaming technique does not depend on the wireless interface being used for, rather depends on the player, video quality, device and the video service provider.

	iPhone4S (iOS 5.0)	iPhone5 (iOS 7.0)	Galaxy S3/Galaxy S3 LTE (Android-4.0.4)		Lumia825 (WP8)
YouTube					
Streaming	*(App)* Throttling Factor = 2.0	*(App)* Throttling Factor = 1.25	*(Flash)* Encoding rate(HD), Throttling(<HD) Factor = 1.25	*(App& HTML5)* ON–OFF-M	(App) Fast Caching
Quality	LD(240p), SD(360p), HD(720p)	LD(240p), SD(360p), HD(720p)	LD(240p), SD(360, 480p), HD(720,1080p)	LD(240p), SD(360, 480p), HD(720p)	SD(270p), HD(720p)
Container	MP4(360,720p)	MP4(360,720p) 3GPP(240p)	XFLV	MP4(>240p) WebM(>240p) 3GPP(270p)	MP4(720p) 3GPP(270p)
Vimeo					
Streaming	*(App)* HLS Chunk Size = 10s	*(App)* ON–OFF-M	*(App)* ON–OFF-S		*(App)* Fast Caching
Quality	240–720p	SD(270, 480p), HD(720p)	SD(270p), HD(720p)		HD(720p)
Container	MP4	MP4	MP4		MP4
Dailymotion					
Streaming	*(App)* Throttling Factor = 1.25	*(App)* HSL Chunk Size = 10s	*(App)* Fast Caching(288p), ON–OFF-S(>288p)		*(App)* Throttling Factor = 1.25
Quality	LD(240)	240–720p	SD(288, 480p), HD(720p)		SD(288p)
Container	MP4	MP4	MP4		MP4
Netflix					
Streaming	*(App)* HLS Chunk Size = 10s	*(App)* HLS Chunk Size = 10s	*(App)* ON–OFF-S		*(App)* MSS Chunk Size = 4s
Quality	240–720p	240–720p	HD(720p)		240–720p
Container	isma, ismv	isma, ismv	MP4		isma, ismv

Impact of streaming strategy on QoE

- Amount of buffered data directly relates to the probability of a buffering event
  - Level of protection against transient network issues
- But service providers need to consider other issues too
  - Unnecessarily served content upon abandoning: **half videos are abandoned before half viewing**
  - **->** Bandwidth waste
  - Capped data plans
  - Energy efficiency

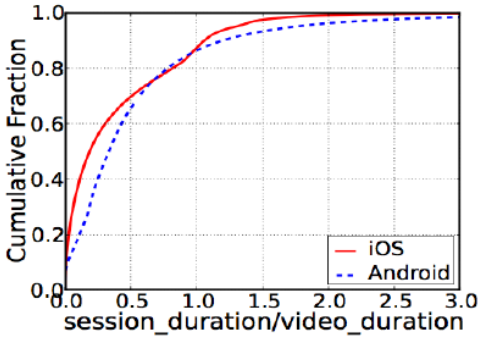
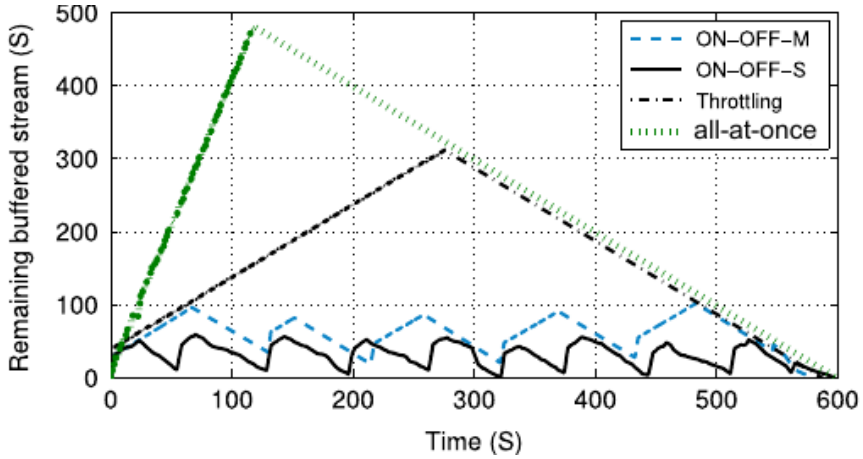


Fig. 1: Ratio Between Session Duration and Video Duration (CDF)



Strategy	No ON-OFF	Long ON-OFF	Short ON-OFF
Engineering Complexity	Not required	Explicit support at Application Layer	
Receive buffer occupancy	Large	Moderate	Small
Unused bytes on user interruption	Large amount	Moderate amount	Small amount

Y. Liu, F. Li, L. Guo, B. Shen and S. Chen, "A Comparative Study of Android and iOS for Accessing Internet Streaming Services," in PAM, Hong Kong, 2013.

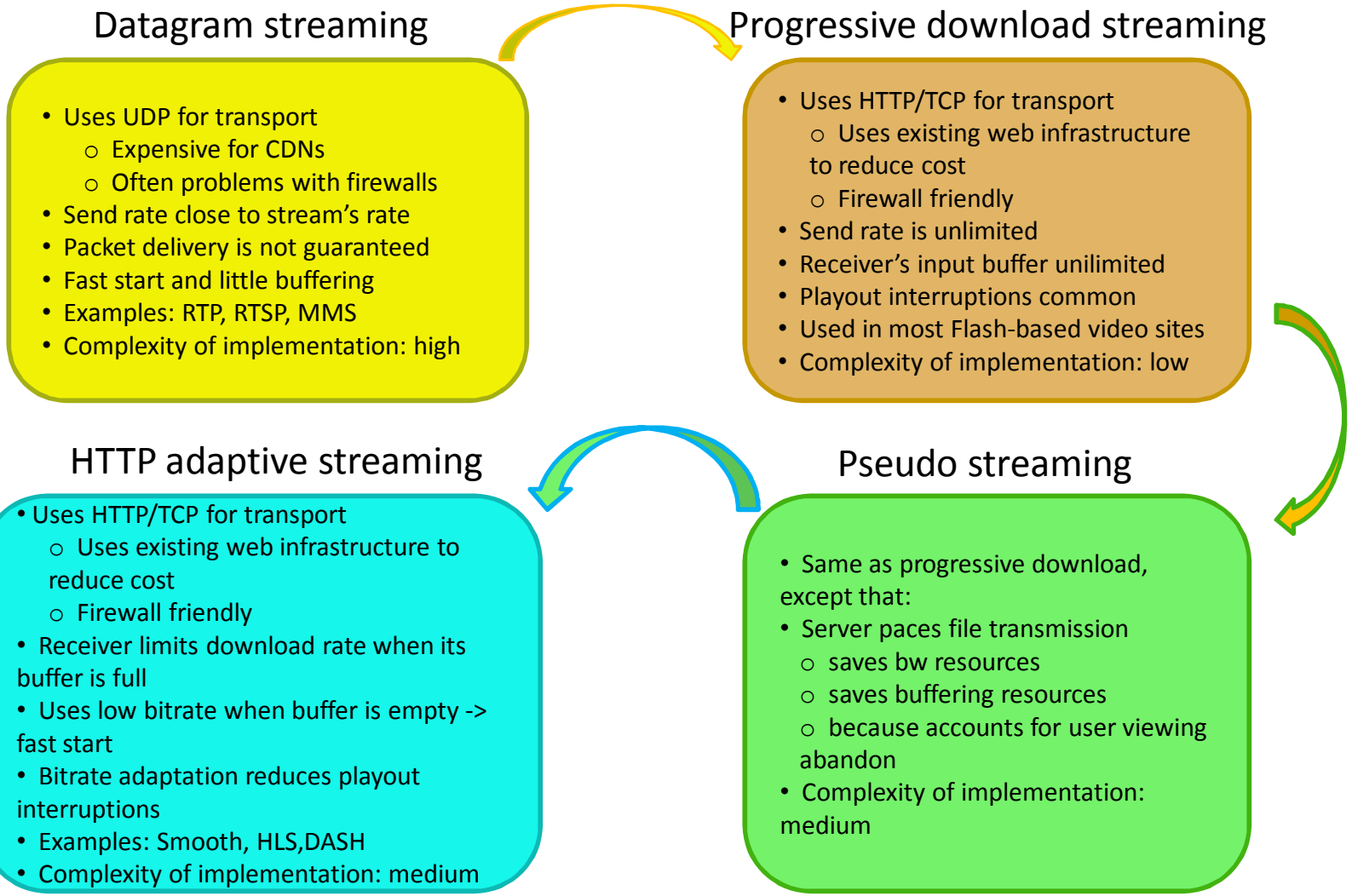
QoE summary

- Many metrics to capture quality
  - "Traditional" – QoS - metrics being replaced by video delivery based metrics - QoE
- QoE is crucially important for video providers and telcos:
  - Direct impact on user engagement -> revenue (through ads etc.)
  - Currently lots of efforts to model and improve it
- Many factors influence QoE
  - Different stakeholders
  - Video CDN operations are a crucial one
  - Streaming strategy is another important one
- There is still room for improvement

Outline

- A general overview of HTTP-based video streaming
  - Principle
  - Current traffic figures
  - Basic architecture
- Client side
  - Quality of Experience: the metrics that matter
  - Various HTTP-based streaming strategies
  - HTTP Adaptive Streaming (HAS)
- Server side
  - Content Distribution Networks (CDNs)
  - Impact of CDN management on QoE

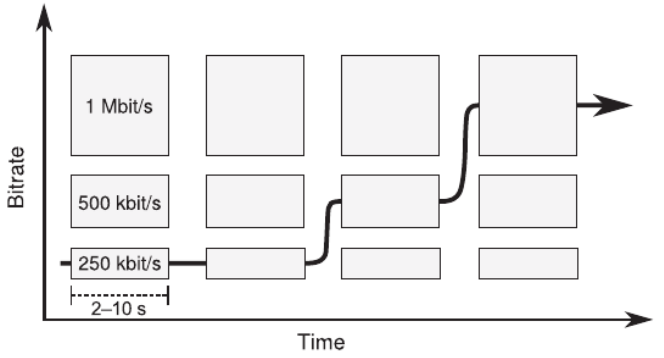
Evolution from datagram streaming to adaptive HTTP streaming



HTTP Adaptive Streaming: Motivation

- Idea: make the requested bitrate of the video fit the (possibly varying) network ressources
- For strained networks: wireless (4G/5G), and wired networks (4K exacerbates) – shared networks like HFC or DSL backhaul are likely to face bandwidth issues

- “available resources”: usually network bandwidth, but also possibly CPU load, battery capacity, and screen size
- client-side adaptation by far the most popular in recent systems: all the information that is relevant when choosing which quality to use is available to the client, not the server
- HAS offers a solution for the most significant problem with streaming over HTTP: fluctuating bandwidth
- -> Being able to switch seamlessly to a stream with a lower bitrate whenever the buffer state is too low makes HTTP much more usable for video streaming, especially on mobile devices.
- -> HAS bound to generalize



HAS: principle

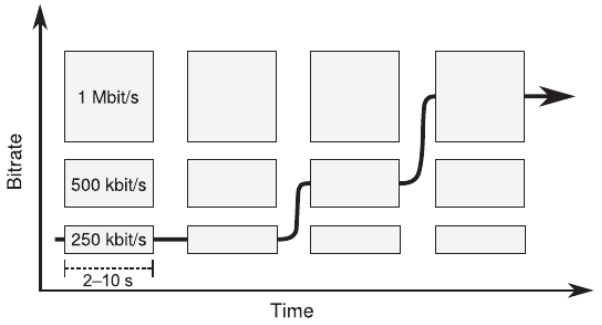
- A stream is split into a sequence of segments downloaded individually, instead of performing one large file download per stream.
- Each segment is typically 2–10 seconds of the stream
- The segment data can be either a multiplexing container format that mixes data from several tracks (audio, video, subtitles, etc.), or it can contain data from just a single track, requiring the receiver to download and process segments from several tracks in parallel.
- The video track is available in multiple different bitrates, each representing a different quality level. The quality can only change on segment boundaries.

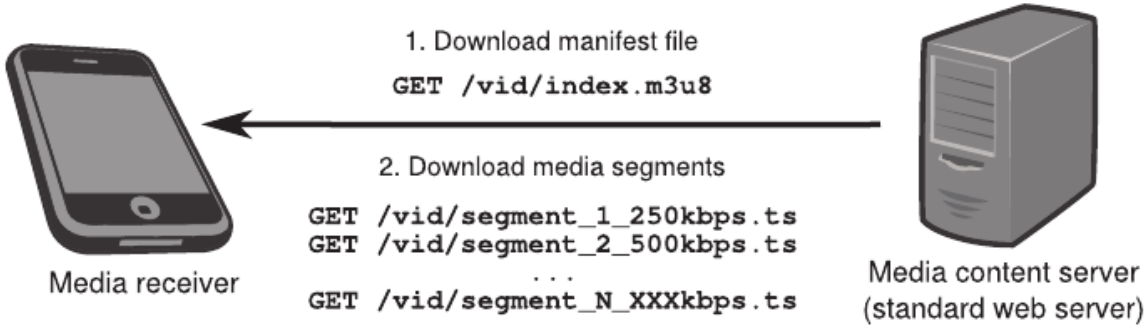
- HAS flavors: DASH, Microsoft Smooth Streaming, HLS,…
- How the client chooses the bitrate to request: up to its software, no need to make it uniform.

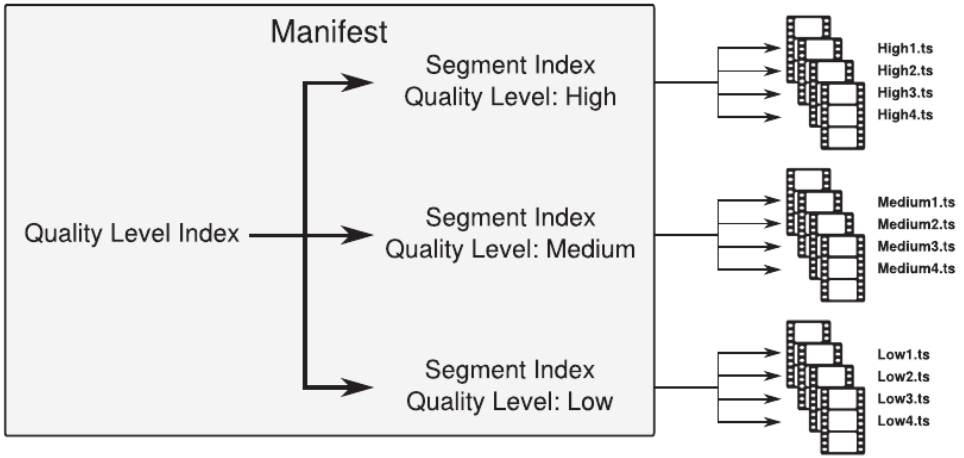
HAS: manifest file

- Each segment is separately downloadable using its own URL.
- Standard HTTP GET requests are sent for every segment, and the URLs for these requests contain what is needed to uniquely identify the segment to be downloaded (timestamp of the first video frame in the segment, bitrate values, …).
- To reduce the entire stream to a single URL, most adaptive formats use a manifest file to describe the stream's structure. The manifest file includes:
  - General info (total stream duration, encryption info, if it is VOD or Live,…)
  - Which types of streams are available (e.g., audio, video, subtitles, etc.)
  - URLs for each media segment, and info about the segments' durations and start times
  - Quality levels are available for each stream
- -> a media player needs only the URL to the manifest file to start playing video, because all the segment URLs will be known after it has downloaded and parsed the manifest

Workflow of HAS and manifest file







Performance of HAS

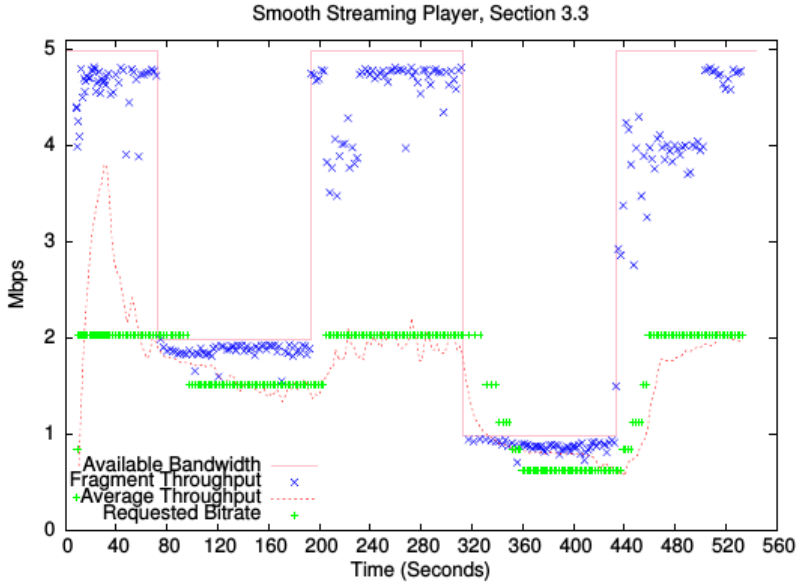


Figure 3: Per-fragment throughput, average TCP throughput and the requested bitrate for the video traffic under persistent avail-bw variations. Playback starts at around t=10 s, almost 3 s after the user clicked PLAY.

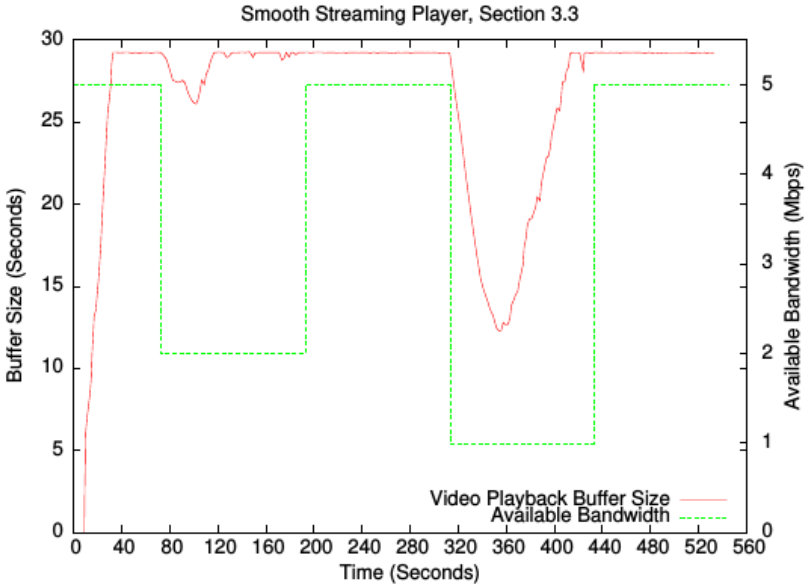


Figure 4: Video playback buffer size in seconds under persistent avail-bw variations.

Performance of HAS

- 1) 1st first player uses the highest profile (P2.75)
- 2) The 2 players could have shared the 4 Mbps bottleneck by switching to P1.52, however, they do not: player 2 oscillates between lower profiles.
- 3) The oscillations continue for both players.
- 4) Stable because lowest rate
- 5) The two players start oscillating in a synchronized manner.

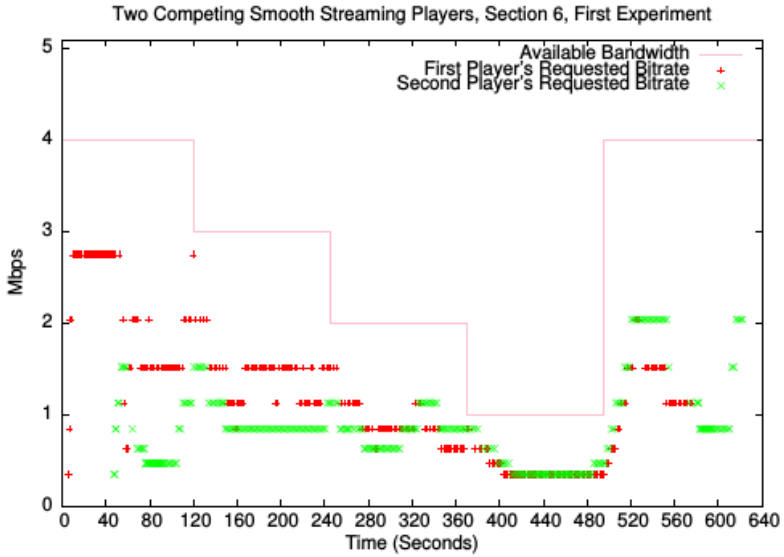
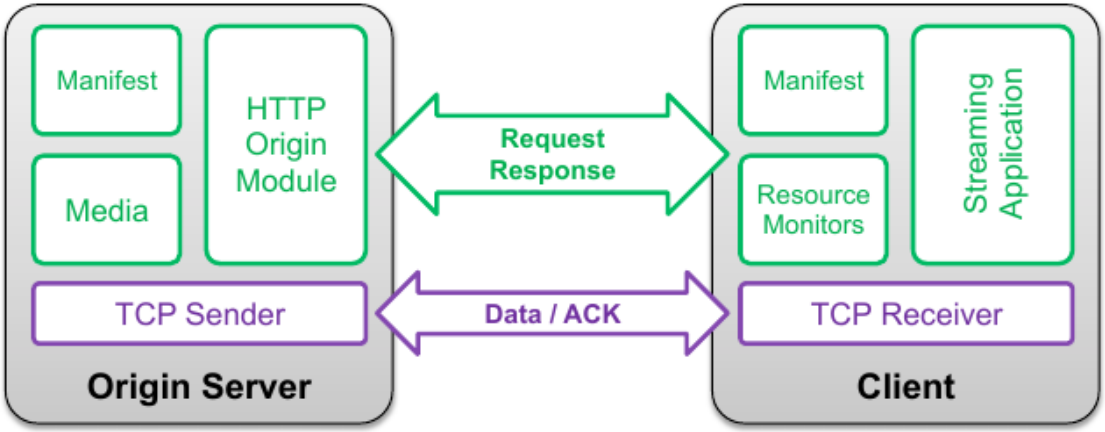


Figure 16: Two Smooth Streaming players compete for avail-bw. The players start the playback at around t=7 s and t=57 s, respectively.

Performance of HAS



- The interplay between TCP loop control and HAS loop control is intricate, specifically when the RDA bases its decision solely on the measured dl rate.
- -> Can generate inefficiency and unfairness
- New players (Netflix) are increasingly using the buffer state info.

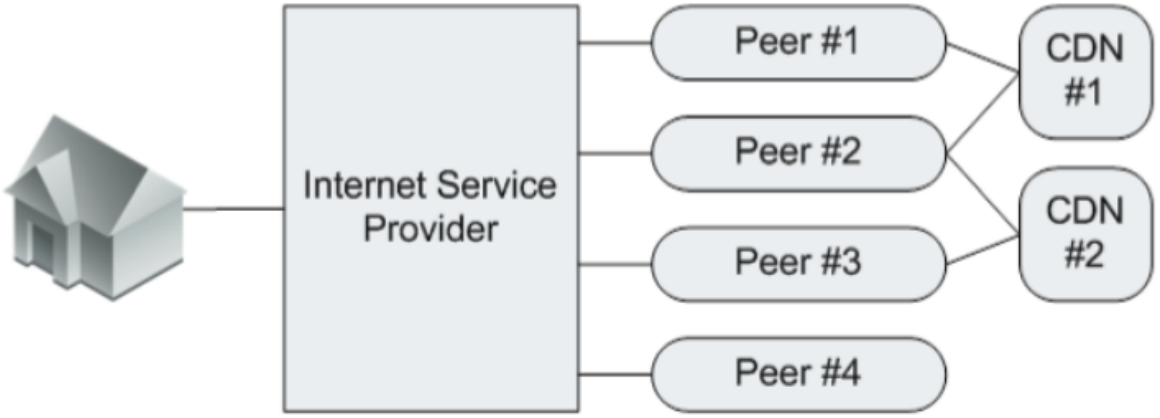
Outline

- A general overview of HTTP-based video streaming
  - Principle
  - Current traffic figures
  - Basic architecture
- Client side
  - Quality of Experience: the metrics that matter
  - Various HTTP-based streaming strategies
  - HTTP Adaptive Streaming (HAS)
- Server side
  - Content Distribution Networks (CDNs)
  - Impact of CDN management on QoE

How is Internet video delivered?

- Typically it is not
  - A small set of centralized servers
  - Clients connecting to those servers
- Why not?
  - Inefficient: long distance between server and client -> delay and bandwidth issues
  - Expensive: requires high capacity servers
- Use of a (large) set of servers distributed near clients
  - Under the control of a company...
  - … that makes agreement with local ISP
  - CDN: Content Distribution Networks
  - E.g.: Akamai, Limelight but also Google and Netflix

Modern video delivery architecture

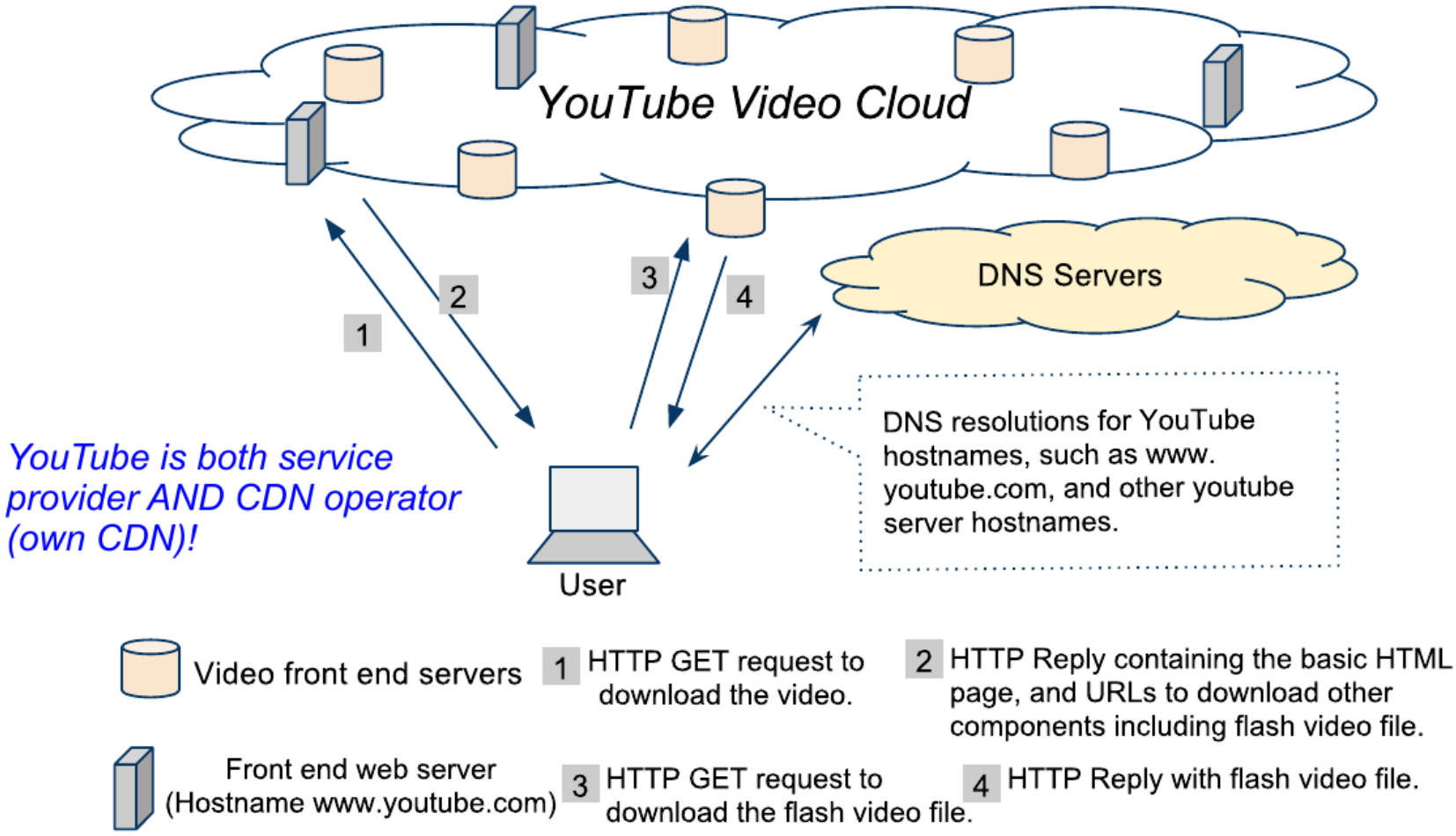


- Your ISP peers with other ISPs which connect to CDNs
- Several choices available
  - Several CDNs serving same content may exist
  - CDN possibly multihomed (servers connected by several of the peering ISPs)

How to choose from alternatives

- Q: Who and based on which criteria…
  - ...chooses the CDN?
  - ...chooses the ISP-level path?
- Partly you by using specific hardware
  - Apple TV, iPad, Android tablet, etc.
  - Visible to service provider through the user-agent string in HTTP
- Partly content provider
  - Can be a function of load, time, type of device,…
  - Example: Netflix serving the same episode of House of Cards at the same time through the same Internet connection
    - Apple TV -> Open Connect (Netflix's own CDN)
    - iPad -> Limelight CDN at off peak times, Open Connect at peak hours
- Partly CDN provider
  - Through which (peering) ISPs content is delivered
- Partly even your ISP
  - Makes an agreement with CDN provider

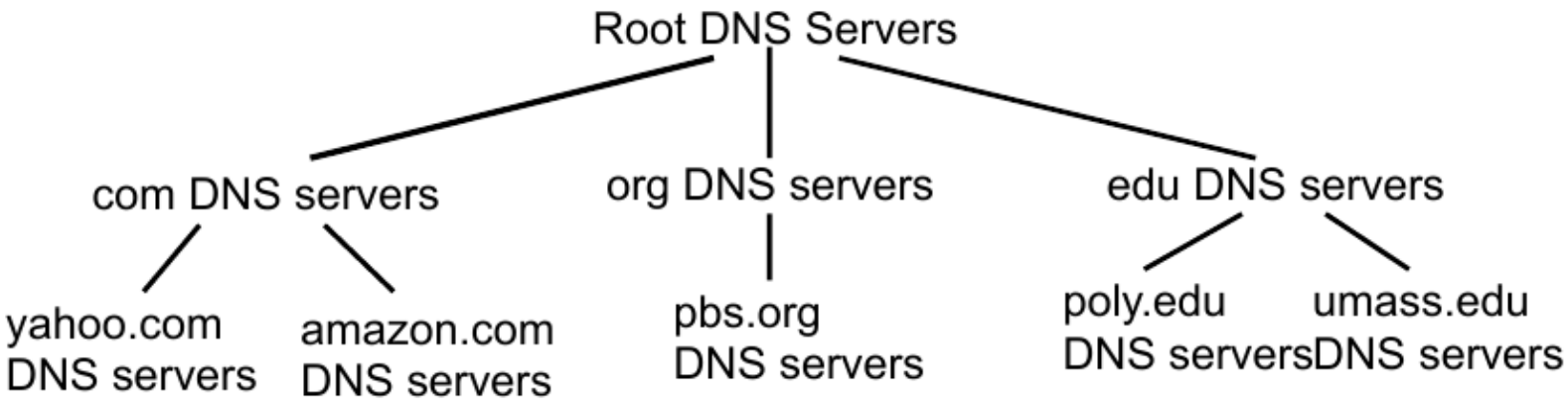
Example: YouTube video delivery basics



Mechanisms for client to server allocation

- Service provider can dynamically control which server name is provided by video web page
  - YouTube: web page is www.youtube.com/...”videoid”...
  - YouTube: server name (e.g. v23.lscache5.c.youtube.com) embedded in web page
  - Gives way to choose the CDN (and server in CDN)
- CDN provider has a couple of different ways to control server allocation
  - Dynamic DNS mappings
  - HTTP redirect
  - Can choose the ISP through which a given client is served
  - Can choose the individual server within CDN

Refresher: DNS – Distributed, Hierarchical Database



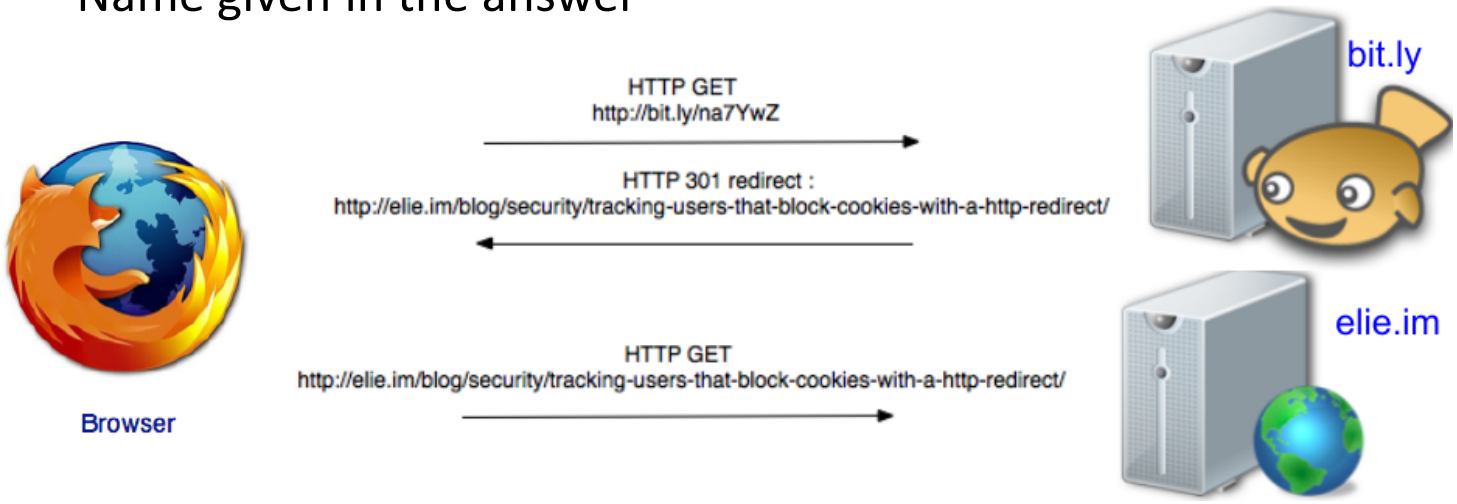
- Client wants IP for www.amazon.com:
  1. client queries a root server to find com DNS server
  2. client queries com DNS server to get amazon.com DNS server
  3. client queries amazon.com DNS server to get IP address for www.amazon.com
- Local DNS servers may do this on behalf of client (recursively)
  - They also cache results which get stale over time -> ask again from authoritative server

Role of DNS in content distribution

- DNS is typically used to perform initial assignments of clients to servers
  - Client resolves video server's DNS name to IP address
- DNS name to address mapping under control of the content provider
  - Provider's DNS servers provide authoritative answers to DNS requests
- DNS mapping often anycast -> one name maps to one of several addresses
  - Suitable server can be chosen for a given client at a given time instance

Role of HTTP redirection

- Another standard way to dynamically change client to server allocation
- Server replies with “HTTP redirect” to guide client to another server
  - Name given in the answer



Picture source : http://www.elie.net/

Outline

- A general overview of HTTP-based video streaming
  - Principle
  - Current traffic figures
  - Basic architecture
- Client side
  - Quality of Experience: the metrics that matter
  - Various HTTP-based streaming strategies
  - HTTP Adaptive Streaming (HAS)
- Server side
  - Content Distribution Networks (CDNs)
  - Impact of CDN management on QoE

Why should we care?

- The choices may influence your streaming experience
  - Quality of Experience (QoE)
- Poor choice can lead to
  - stalled video
  - long initial startup delay
  - lower quality and/or more quality switches in case of adaptive streaming
- Underlying reasons
  - Video server / CDN overloaded
  - ISPs' peering poorly dimensioned
  - Server located far away

A glimpse at a study on YouTube architecture

- Based on 2012 article
  - Adhikari et al: Vivisecting youtube: An active measurement study. Infocom 2012.
- YouTube internals and operations are not all public
  - Need to reverse engineer
- A kind of blackbox study to understand the behavior
  - Use the system in different ways (stimuli) and observe the response

YouTube video ID space & namespace mapping

- YouTube video objects are identified with 11 literals
  - {a-Z, 0-9, , -} for the first 11 positions
  - 16 symbols only for the last position
    - -> ID space is 6311x16
    - E.g. MQCNuv2QxQY
  - The investigated 434k videos are uniformly distributed over the ID space
- Videos are mapped on several DNS namespaces
  - Only DNS names belonging to lscache namespace is visible in web pages (landing page of a video)
    - E.g.: v23.lscache1.c.youtube.com
  - Others appear during redirection process

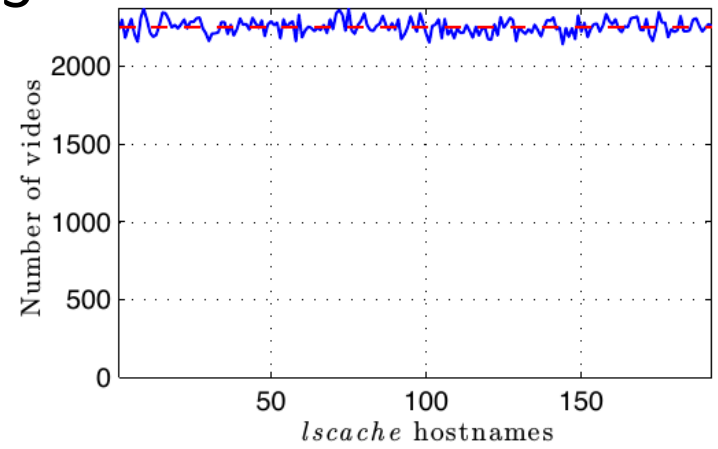
YouTube namespaces

TABLE I
YOUTUBE *Anycast* (FIRST FIVE) AND *Unicast* (LAST TWO) NAMESPACES.

DNS namespace	format	# hostnames	# IPs	# prefixes	# locations	any/uni-cast
lscache	v[1-24].lscache[1-8].c.youtube.com	192	4,999	97	38	anycast
nonxt	v[1-24].nonxt[1-8].c.youtube.com	192	4,315	68	30	anycast
tccache	tc.v[1-24].cache[1-8].c.youtube.com	192	636	15	8	anycast
cache	v[1-8].cache[1-8].c.youtube.com	64	320	5	5	anycast
altcache	alt1.v[1-24].cache[1-8].c.youtube.com	64	320	5	5	anycast
rhost	r[1-24].city[01-16][s,g,t][0-16].c.youtube.com	5,044	5,044	79	37	unicast
rhostisp	r[1-24].isp-city[1-3].c.youtube.com	402	402	19	13	unicast

Mapping of video IDs to DNS namespaces

- A given video ID always maps to a specific DNS name
  - Location of the client does not influence
- Name is always one of the 192 DNS names in lscache namespace
  - E.g.: MQCNuv2QxQY -> v23.lscache1.c.youtube.com
- Names in lscache are anycast
  - E.g. v23.lscache1.c.youtube.com resolves to different IP address for different clients
- Mapping of videos to lscache appears uniform



Cache server namespaces & addresses

- YouTube utilizes 5 anycast namespaces and 2 unicast namespaces (name to IP is constant)
- A total number of 6000 observed IPs
  - "Physical" video cache servers distributed around the world
- 80 % of addresses come from Google/YouTube and 20 % from ISPs
  - YouTube deploys servers in ISPs point of presence (PoP)
    - Google Global Cache (GGC)

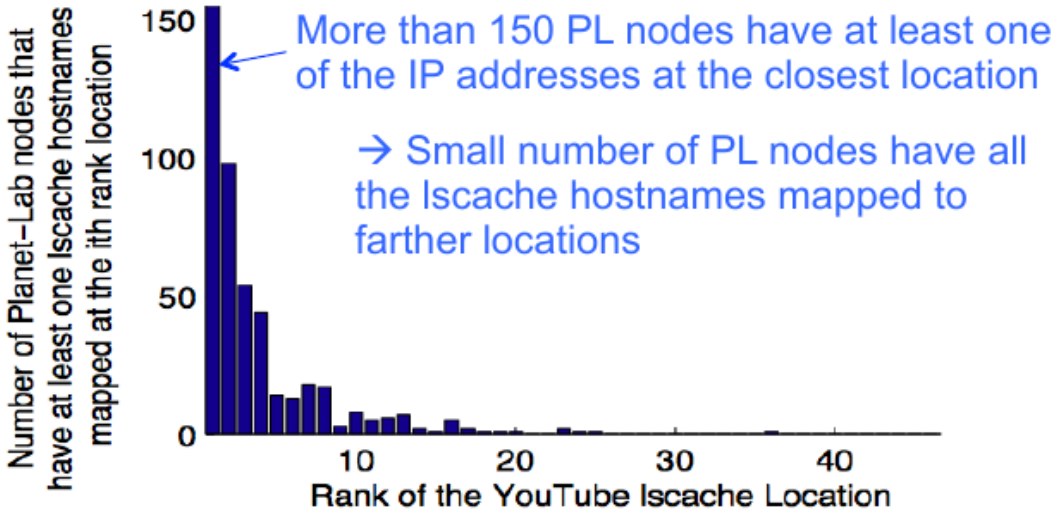
Video delivery dynamics: locality aware DNS resolution

- Q: Does YouTube make efforts to deliver video from nearby server?
- Known facts:
  - Each hostname in the primary lscache namespace is mapped to >75 unique IP addresses
    - Distributed across the 38 primary cache locations
  - Each PL node site generally sees only one IP address per lscache name at a time
- Investigate how far the IP seen by given PL node is
  - Rank primary cache locations based on RTT (ping) from a given PL node
  - Repeat the above for each PL node -> dataset of locality of DNS resolution by YouTube names

Video delivery dynamics: locality aware DNS resolution (cont.)

- A: Yes, high degree of locality (surprise...)

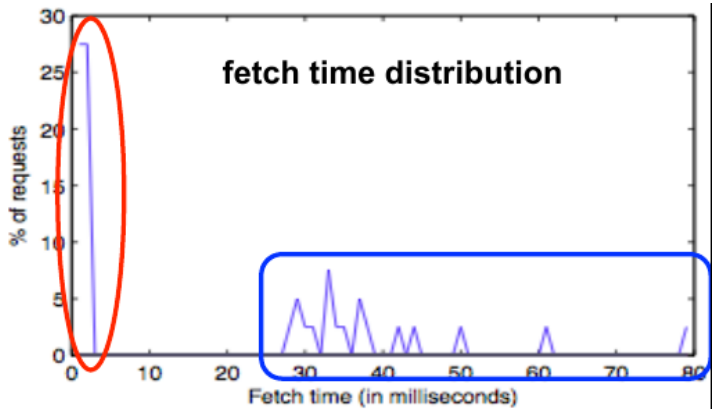
Plot interpretation: Add one if a PL node has at least one *Iscache* hostname mapped to i^{th} rank YouTube location



- YouTube DNS servers generally map each anycast hostname to a nearby YouTube primary cache location

Video delivery dynamics: cache misses

- Two groups of fetch times observed
- Group 1: few milliseconds fetch time
  - When cache server redirects client (no content served) OR
  - When server serves *hot* videos (no redirections)
    - Has the content readily available
- Group 2: tens of milliseconds fetch time
  - When server serves *cold* videos
  - Cache server probably needs to fetch content from backend datacenter



Concluding YouTube case study

- YouTube architecture is complex
- Fine grained control client to server mapping
  - Locality-aware
  - Time of the day dependent
  - Video popularity dependent
  - Load dependent
  - …
- Note: this study is already dated
  - Probably/apparently YouTube undergone changes
  - Take all numbers with a grain of salt
- Main takeaway is the variety of possibilities to dynamically control video content delivery

Wrapping up...

- Video streaming traffic dominates the Internet
  - Will grow to be even more important
  - High stakes at play for many companies (ISPs, video service providers, device manufacturers, etc.)
- Video delivery architectures have grown large and complex
  - Cache servers organized into multiple layers
  - Clever use of DNS and HTTP redirect to dynamically address load balancing, traffic locality, cache misses
- QoE matters a lot
  - Manifestation of many underlying factors
  - Large efforts to optimize it, as this is what clients see