IEEE International Conference on Multimedia and Expo
June 29 – July 3, 2015
Torino, Italy

I don't mind ads,
I don't mind buffer,
but when ads buffer,
I suffer.
--Internet quote
(unknown source)

Over the Top Content Delivery: State of the Art and Challenges Ahead

IEEE ICME 2015

June 2015

Christian Timmerer, AAU/bitmovin

Ali C. Begen, CISCO

Note Well

The information and slides in this tutorial are public.

However, Alpen-Adria-Universität Klagenfurt, Cisco and their affiliates hold the copyrights.

Please use proper citation when using content from this tutorial.

Slides will be available at
http://www.slideshare.net/christian.timmerer

Presenters Today

Christian Timmerer

- Associate Professor at the Institute of Information Technology (ITEC), Multimedia Communication Group (MMC), Alpen-Adria-Universität Klagenfurt, Austria
- Co-founder CIO of bitmovin (www.bitmovin.com)
- Research Interests
  - Immersive multimedia communication
  - Streaming, adaptation, and
  - Quality of Experience (QoE)
- General Chair of QoMEX’13, WIAMIS’08, TPC-Co Chair of ACM TVX’15, MMSys’16
- AE for Computing Now, IEEE Transactions on Multimedia; EB for IEEE Computer; Area Editor for Signal Processing: Image Communication; Editor for SIGMM Records
- EU projects: DANAE, ENTHRONE, P2P-Next, ALICANTE, SocialSensor, QUALINET, and ICoSOLE
- Active member of ISO/IEC MPEG and DASH-IF
- Blog: http://blog.timmerer.com; @timse7

Ali C. Begen

- Have a Ph.D. degree from Georgia Tech, joined Cisco in 2007
- Works in the area of architectures for next-generation video transport and distribution over IP networks
- Areas of Expertise
  - Networked entertainment
  - Internet multimedia
  - Transport protocols
  - Content distribution
- Senior member of the IEEE and ACM
- Visit http://ali.begen.net for publications and presentations

MMSYS 2016

May 10-13, 2016
Klagenfurt am Wörthersee
AUSTRIA

Submission deadline: November 27, 2015
http://www.mmsys.org/ | http://mmsys2016.itec.aau.at/ | @mmsys2015

What to Expect from This Tutorial

- Upon attending this tutorial, the participants will have an understanding of the following:
  - ✓ Fundamental differences between IPTV and IP (over-the-top) video
  - ✓ Features of various types of streaming protocols
  - ✓ Principles of HTTP adaptive streaming
  - ✓ Content generation, distribution and consumption workflows
  - ✓ Current and future research on unmanaged video delivery
  - ✓ The MPEG DASH standard

Agenda

- Part I: Over-the-Top (OTT) Video and HTTP Adaptive Streaming
  - OTT Delivery and Example Services
  - Media Delivery over the Internet
  - HTTP Adaptive Streaming Building Blocks
  - Workflows for Content Generation, Distribution and Consumption
  - Overview of the MPEG DASH Standard
- Part II: Common Problems in HTTP Adaptive Streaming
  - Multi-Client Competition Problem
  - Consistent-Quality Streaming
  - QoE Optimization and Measurement
  - Inter-Destination Media Synchronization
- Part III: Open Issues and Future Research Directions

First Things First

IPTV vs. IP (Over-the-Top) Video

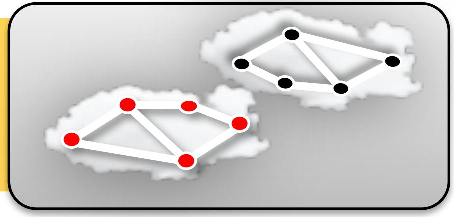
IPTV	IP Video
Managed delivery	Best-effort delivery
Emphasis on quality	Quality not guaranteed
Linear TV plus VoD	Mostly on demand
Paid service	Paid or free service

Three Dimensions of the Problem

Content, Transport and Devices

Managed and Unmanaged Content

Managed and Unmanaged Transport



Managed and Unmanaged Devices

From Totally Best-Effort to Fully-Managed Offerings

Challenge is to Provide a Solution that Covers All

Design to the most general case
Optimize where appropriate

Part I: Over-the-Top (OTT) Video and HTTP Adaptive Streaming

- OTT Delivery and Example Services
- Media Delivery over the Internet
- HTTP Adaptive Streaming Building Blocks
- Workflows for Content Generation, Distribution and Consumption
- Overview of the MPEG DASH Standard

Internet Video Essentials

Reach	▪ Reach all connected devices
Scale	▪ Enable live and on-demand delivery to the mass market
Quality of Experience	▪ Provide TV-like consistent rich viewer experience
Business	▪ Enable revenue generation thru paid content, subscriptions, targeted advertising, etc.
Regulatory	▪ Satisfy regulations such as captioning, ratings and parental control

Creating Revenue – Attracting Eye Balls

- High-End Content
  - Hollywood movies, TV shows
  - Sports
- Excellent Quality
  - HD/3D/UHD audiovisual presentation w/o artifacts such as pixelization and rebuffering
  - Fast startup, fast zapping and low glass-to-glass delay
- Usability
  - Navigation, content discovery, battery consumption, trick modes
- Service Flexibility
  - Linear TV
  - Time-shifted and on-demand services
- Reach
  - Any device, any time
- Auxiliary Services
  - Targeted advertising, social network integration

Internet TV vs. Traditional TV in 2010

- Areas most important to overall TV experience are:
  - Content
  - Timing control
  - Quality
  - Ease of use
- While traditional TV surpasses Internet TV **only in quality**, it delivers better "overall experience"

When comparing traditional and Internet TV, which option is better?

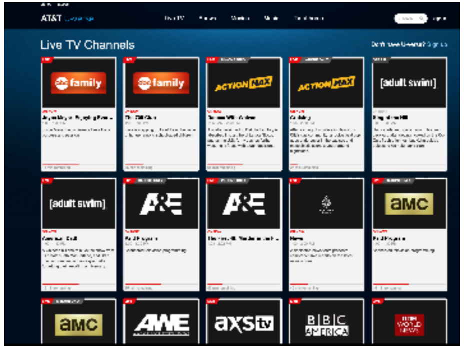
	Traditional	Internet
Content	7%	➢ 79%
Timing / Control	7%	➢ 83%
Quality	➢ **80%**	16%
Ease of Use	23%	➢ 52%
Control (FF, etc.)	9%	➢ 77%
Portability	4%	➢ 92%
Interactivity	31%	➢ 52%
Sharing	33%	➢ 56%
Overall Experience	➢ 53%	33%

Source: Cisco IBSG Youth Survey, Cisco IBSG Youth Focus Group Sessions, 2010

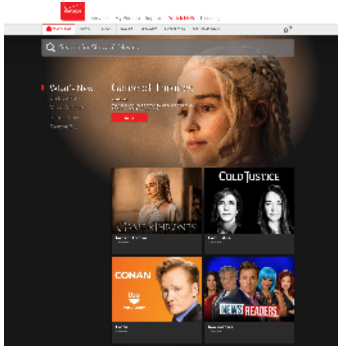
Part I: Over-the-Top (OTT) Video and HTTP Adaptive Streaming

- OTT Delivery and Example Services
- Media Delivery over the Internet
- HTTP Adaptive Streaming Building Blocks
- Workflows for Content Generation, Distribution and Consumption
- Overview of the MPEG DASH Standard

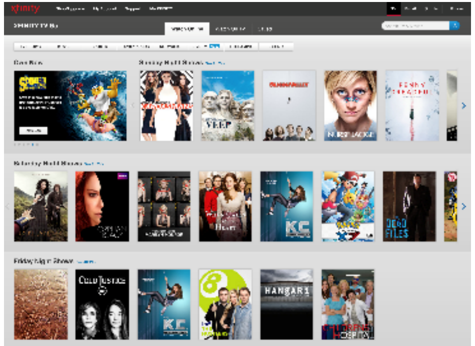
The Lines are *Blurring* between TV and the Web



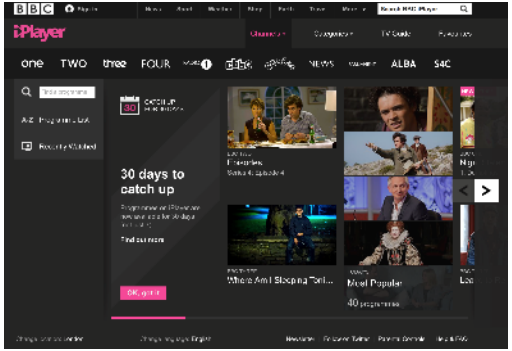
AT&T U-verse



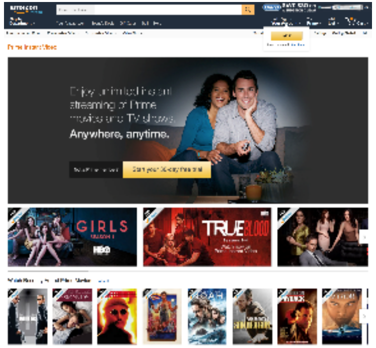
Verizon FiOS



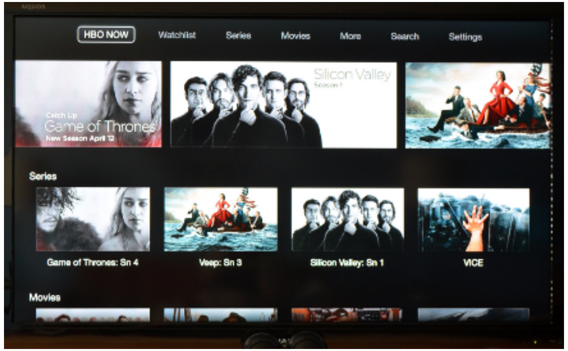
Comcast XFINITY



BBC iPlayer



Amazon



HBO NOW

Netflix

Content

Over 100K titles (DVD)

Shipped 1 billionth DVD in 02/07

Shipped 2 billionth DVD in 04/09

Today: HD, 3D and UHD

Revenue

Most recent Quarter: $1.5B in Q1/2015

FY (2014-10): $5.5B, $4.3B, $3.6B, $3.2B, $2.1B

Streaming Subscribers

41.4M (US) by Q1 2015 (20.8M in 40 countries)

[6M DVD subscribers in the US by Oct. 2014]

Competitors

Hulu+, Amazon Prime, HBO GO, HBO NOW

Difficulties

ISP data caps

ISP/CDN throughput limitations

Big Data at Netflix

Library: 3PB

Ratings: 4M/day, searches: 3M/day, plays: 30M/day

5B hours streamed in Q3 2013 (2B in Q4 2011, 3B in Q3 2012)

Plans

Unlimited streaming 1-stream plan for $7.99

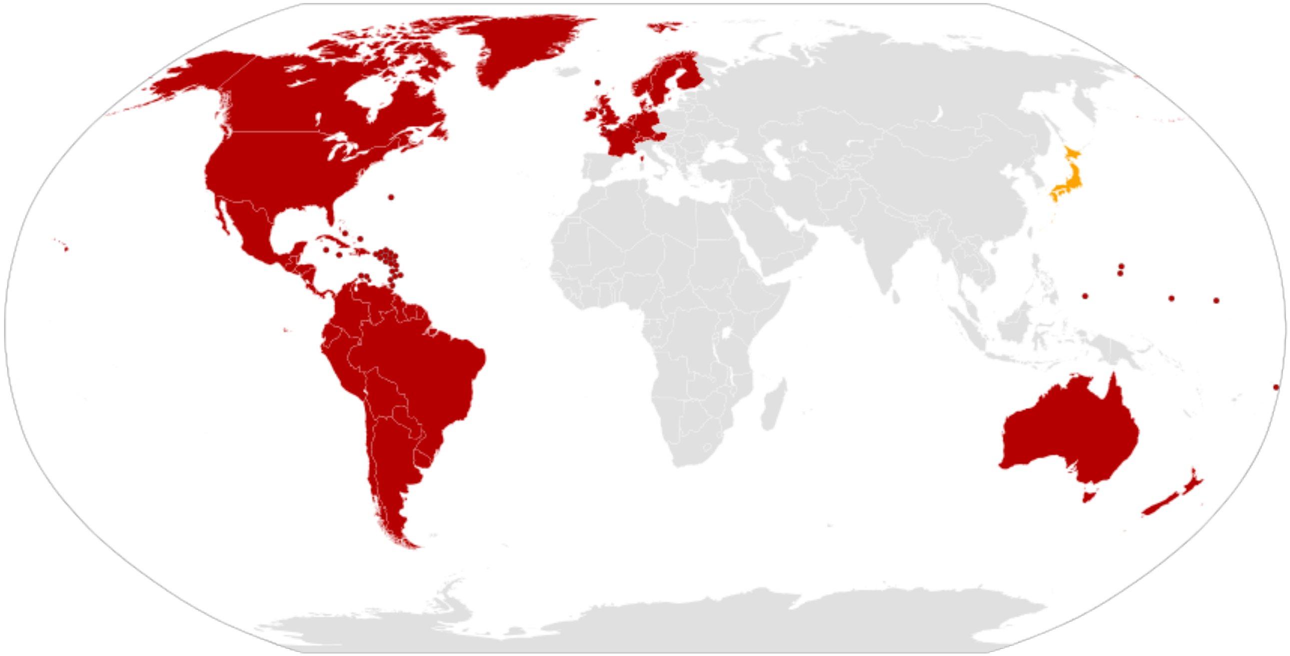
HD 2-stream plan for $8.99, and 4K 4-stream plan for $11.99

1 DVD out at-a-time for $7.99

Blu-rays for an additional $2 per month (US)

Netflix's Expansion to International Markets

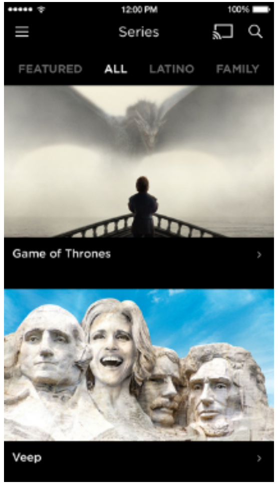
Red: Available, Orange: Announced / Coming Soon

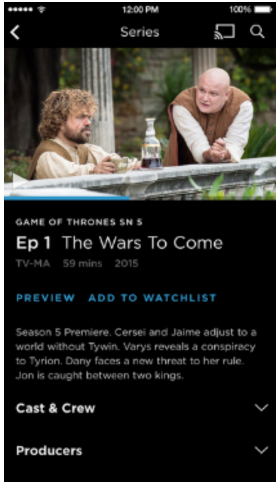


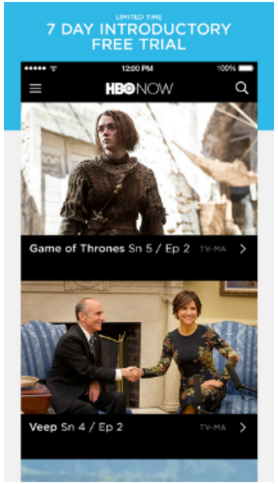
HBO GO and HBO NOW

Delivery of TV Content to IP-Enabled Devices

- Subscribers can watch HBO content thru HBO GO service
  - First launched in Feb. 2010 with Verizon FiOS
  - Later expanded to other provides including AT&T U-Verse, DirecTV, DISH Network, Comcast XFINITY, TWC, etc.
- Starting in April 2015, HBO has been serving consumers directly thru HBO NOW service



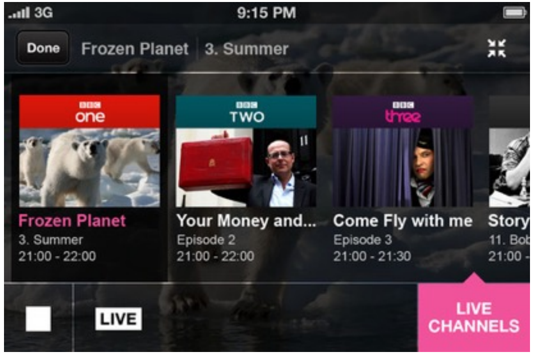




BBC iPlayer

Available (Almost) Globally

- Statistics for April 2015
  - Total Requests
    - 218M for TV programs (9% of the requests were for live streams)
    - 53M for radio programs (73% of the requests were for live streams)
  - Devices
    - 31% computers (~), 23% tablets (~), 24% mobile devices (+), 9% TV platform operators (-), %3 game consoles (~)
- Impact of the World Cup 2014: http://www.bbc.co.uk/mediacentre/latestnews/2014/iplayer-performance-pack-july
- Xmas 2014 Surge: iPlayer had the best month on record with 343M requests (264M for TV) in Jan. 2015



Source: http://downloads.bbc.co.uk/mediacentre/iplayer/iplayer-performance-apr15.pdf

Internet Video (Desktop) in the US

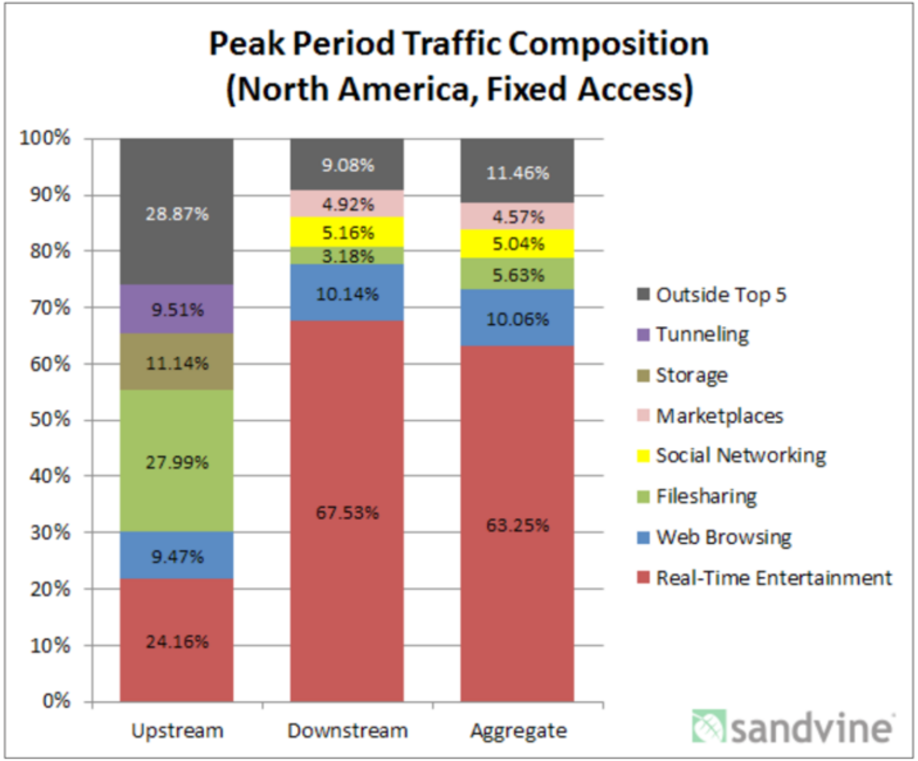
April 2015

	Unique Viewers (x1000)
Google Sites	152,781
Facebook	83,549
Yahoo Sites	55,445
Maker Studios Inc.	43,653
VEVO	43,131
Vimeo	37,365
AOL, Inc.	37,156
Fullscreen	34,200
Comcast NBCUniversal	32,659
CBS Interactive	30,499
Total	**191,019**

Source: comScore Video Metrix

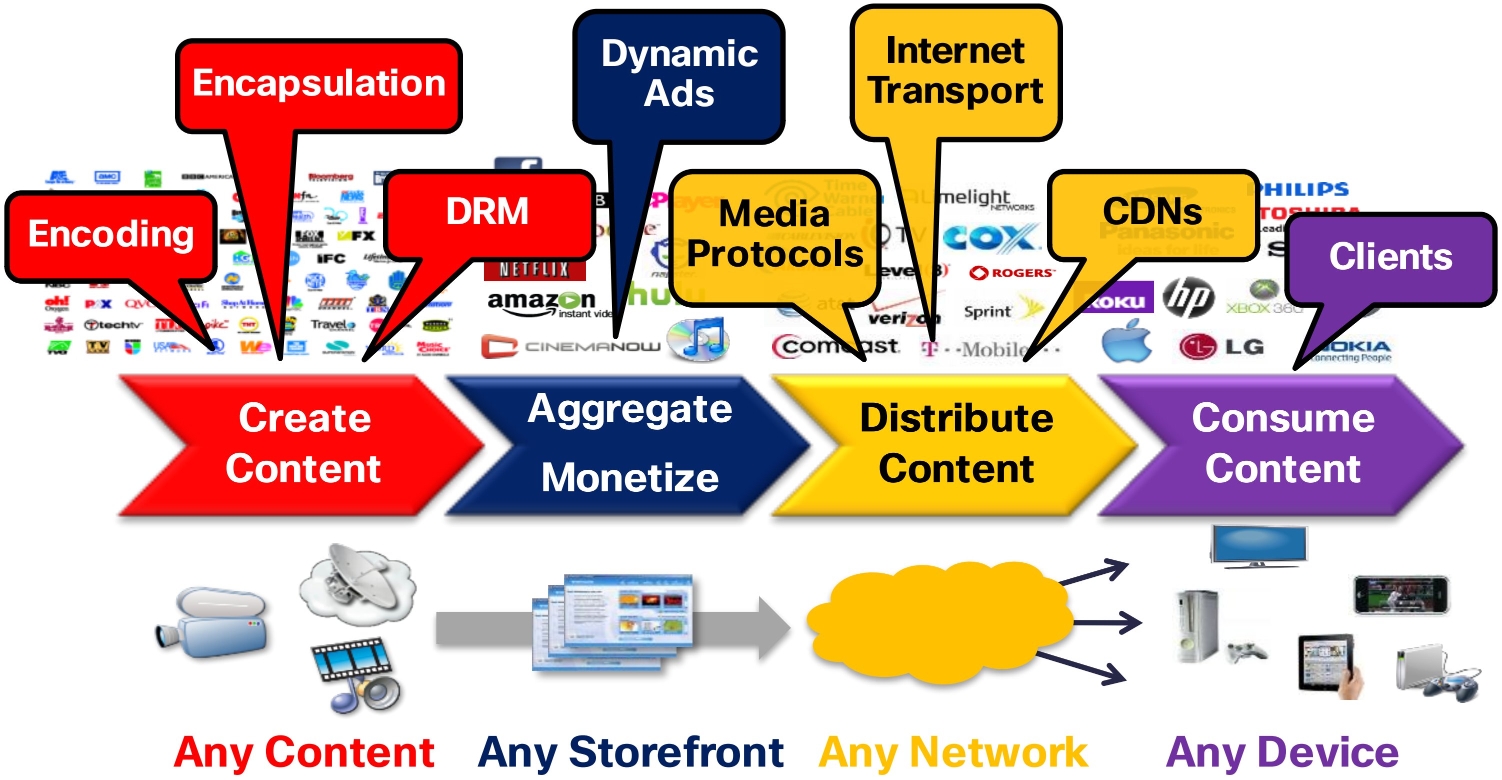
Multimedia is Predominant on the Internet

- Real-time entertainment
  - Streaming video and audio
  - More than 60% of Internet traffic at peak periods
- Popular services
  - YouTube (14.0%), Netflix (34.9%), Amazon Video (2.6%), Hulu (1.4%)
  - All delivered over the top

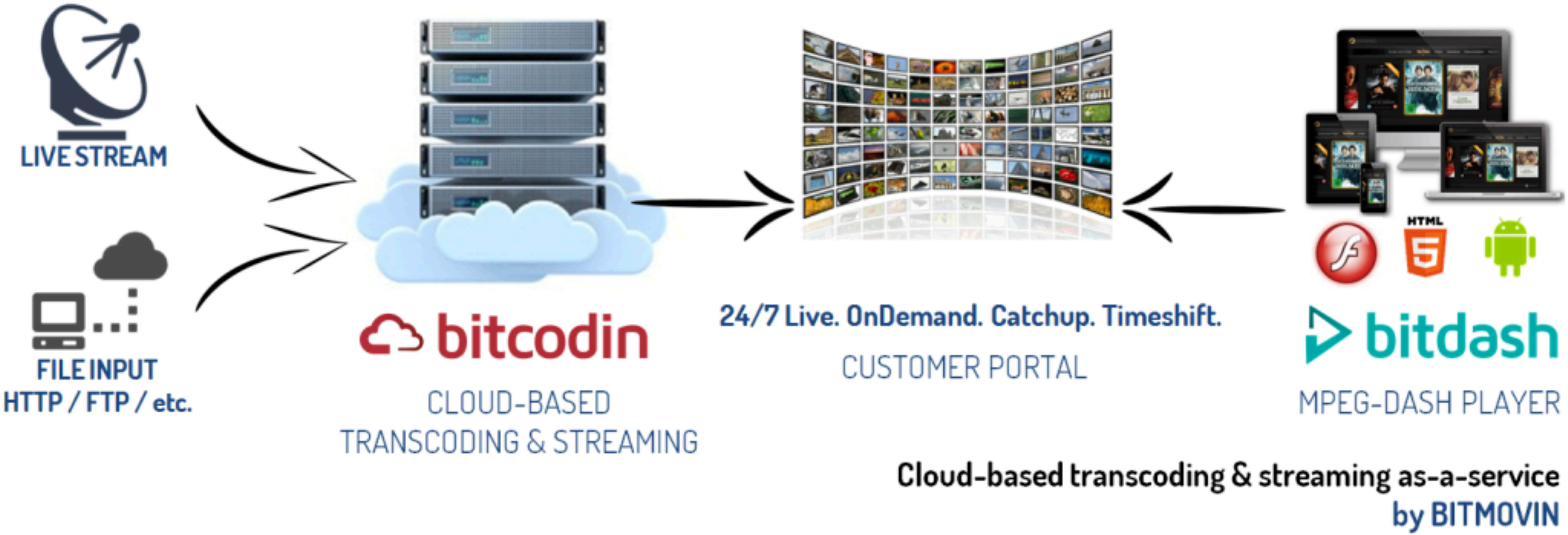


Source: Global Internet Phenomena Report: 2H 2014

Open Digital Media Value Chain



Simplified Example Workflow: bitcodin/bitdash



Source: http://www.bitmovin.net/bitcodin-cloud-based-transcoding-streaming-platform/

Part I: Over-the-Top (OTT) Video and HTTP Adaptive Streaming

- OTT Delivery and Example Services
- Media Delivery over the Internet
- HTTP Adaptive Streaming Building Blocks
- Workflows for Content Generation, Distribution and Consumption
- Overview of the MPEG DASH Standard

Some Background

Broadcast, Broadband, Hybrid Broadcast Broadband

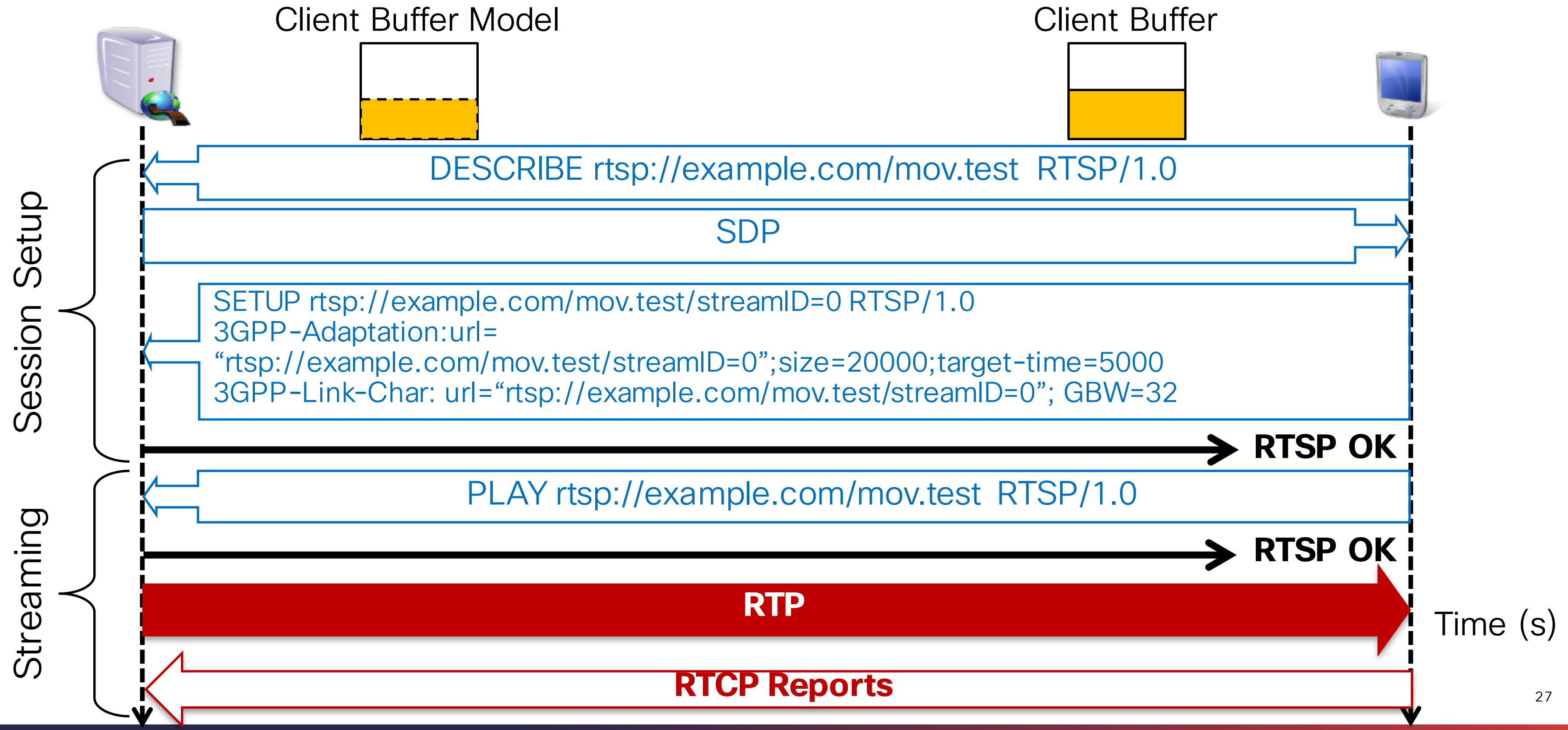
- **Broadcast: MPEG2-TS, DVB, etc.**
- **Broadband, Push-based Streaming**
  - Sender-initiated, content is pushed towards clients (unicast, multicast); intelligent servers, infrastructure, dump clients; typically managed networks
  - Real-time Transport Protocol (RTP) and RTSP, RTCP (sender/receiver reports), SDP, SAP … requires codec-specific payload formats
  - User Datagram Protocol (UDP): simple, connection-less but unreliable
  - Dedicated streaming architecture and corresponding infrastructure
  - Adaptivity through explicit feedback loop, automatic repeat requests, server-based real-time adaptation or stream switching
  - NAT/Firewall issues: requires STUN/TURN/etc. protocols
- **Broadband, Pull-based Streaming**
  - Client-initiated, content is pulled from server (unicast); intelligent clients, existing infrastructure, servers; typically unmanaged networks – OTT streaming
  - Manifest and segments formats (MPEG-2 TS, ISOBMFF)
  - Hypertext Transfer Protocol (HTTP): port 80, no NAT/firewall issues
  - Transmission Control Protocol (TCP): connection-oriented
  - Re-use of existing infrastructure for Web content (server, proxy, cache, CDN)
  - Adaptivity through smart client decisions – adaptation logic
- **Hybrid Broadband Broadcast**
  - Additional synchronization and orchestration issues

Push and Pull-Based Video Delivery

	Push-Based Delivery	Pull-Based Delivery
Source	Broadcasters/servers like Windows Media Apple QuickTime, RealNetworks Helix Cisco VDS/DCM	Web/FTP servers such as LAMP Microsoft IIS Adobe Flash RealNetworks Helix Cisco VDS
Protocols	RTSP, RTP, UDP	HTTP, RTMPx, FTP
Video Monitoring and User Tracking	RTCP for RTP transport	(Currently) Proprietary
Multicast Support	Yes	No
Caching Support	No	Yes for HTTP

Push-Based Video Delivery over RTSP

3GPP Packet-Switched Streaming Service (PSS)



Pull-Based Video Delivery over HTTP

Progressive Download vs. Pseudo and Adaptive Streaming

Progressive Download

- Server sends a file as fast as possible
- Client starts playout after a certain initial buffering

Pseudo Streaming

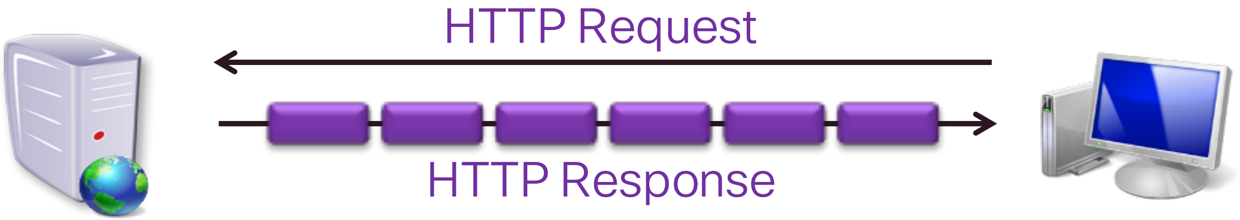
- Server paces file transmission
- Client can seek
- Some metadata is required

Adaptive Streaming

- Client requests small chunks enabling adaptation
- Live streaming and dynamic ads are supported

Progressive Download

One Request, One Response



What is Streaming?

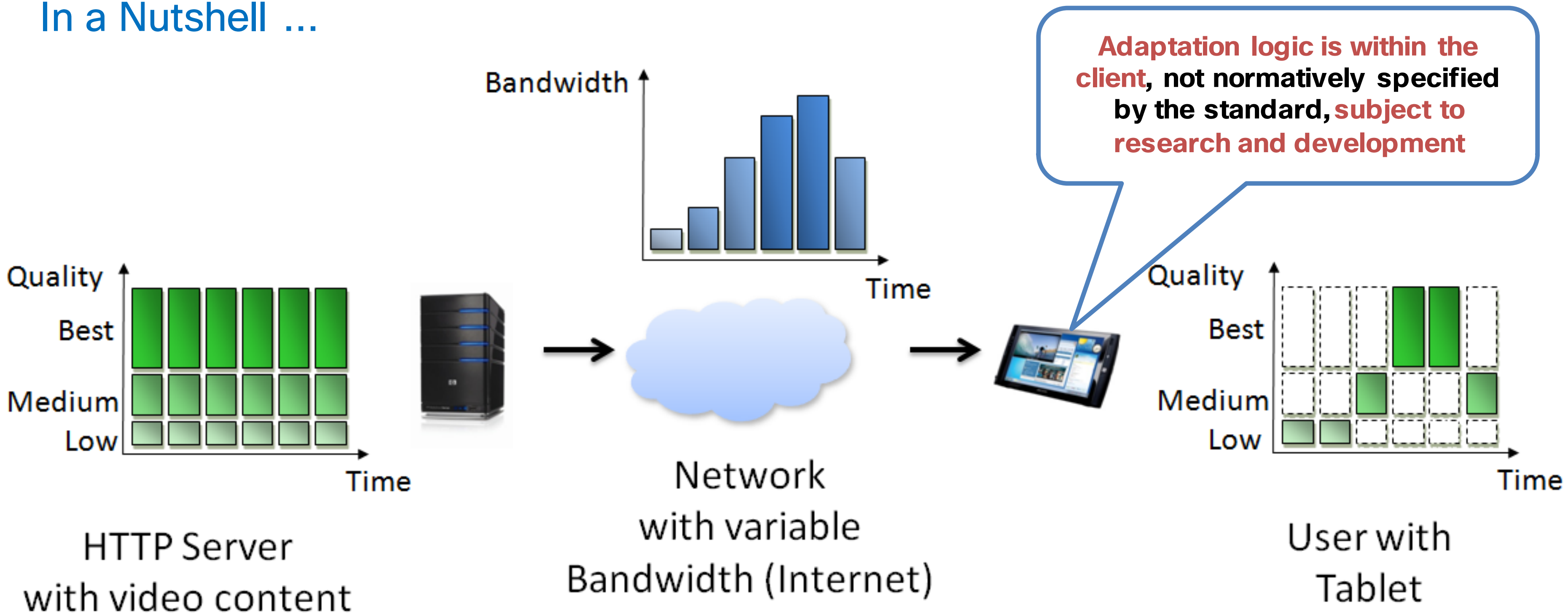
Streaming is transmission of a continuous content from a server to a client and its simultaneous consumption by the client

Two Main Characteristics

1. Client consumption rate may be limited by real-time constraints as opposed to just bandwidth availability
2. Server transmission rate (loosely or tightly) matches to client consumption rate

Over-The-Top - Adaptive Media Streaming

In a Nutshell ...



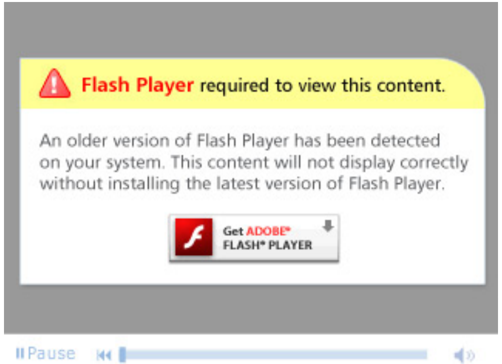
C. Timmerer and C. Griwodz, “Dynamic adaptive streaming over HTTP: from content creation to consumption”, *In Proceedings of the 20th* ***ACM international conference on Multimedia (MM '12)***, Nara, Japan, Oct./Nov. 2012.

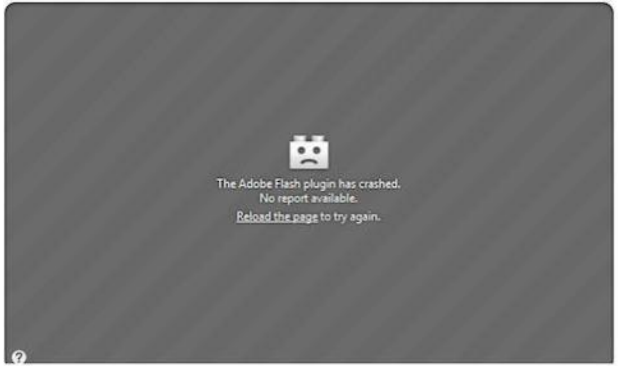
http://www.slideshare.net/christian.timmerer/dynamic-adaptive-streaming-over-http-from-content-creation-to-consumption

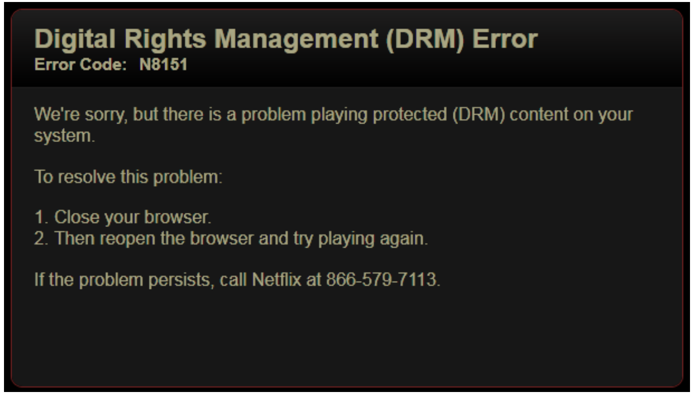
Common Annoyances in Streaming

Stalls, Slow Start-Up, Plug-In and DRM Issues

- Wrong format
- Wrong protocol
- Plugin requirements
- DRM issues
- Long start-up delay
- Poor quality
- Frequent stalls
- Quality oscillations
- No seeking features







Part I: Over-the-Top (OTT) Video and HTTP Adaptive Streaming

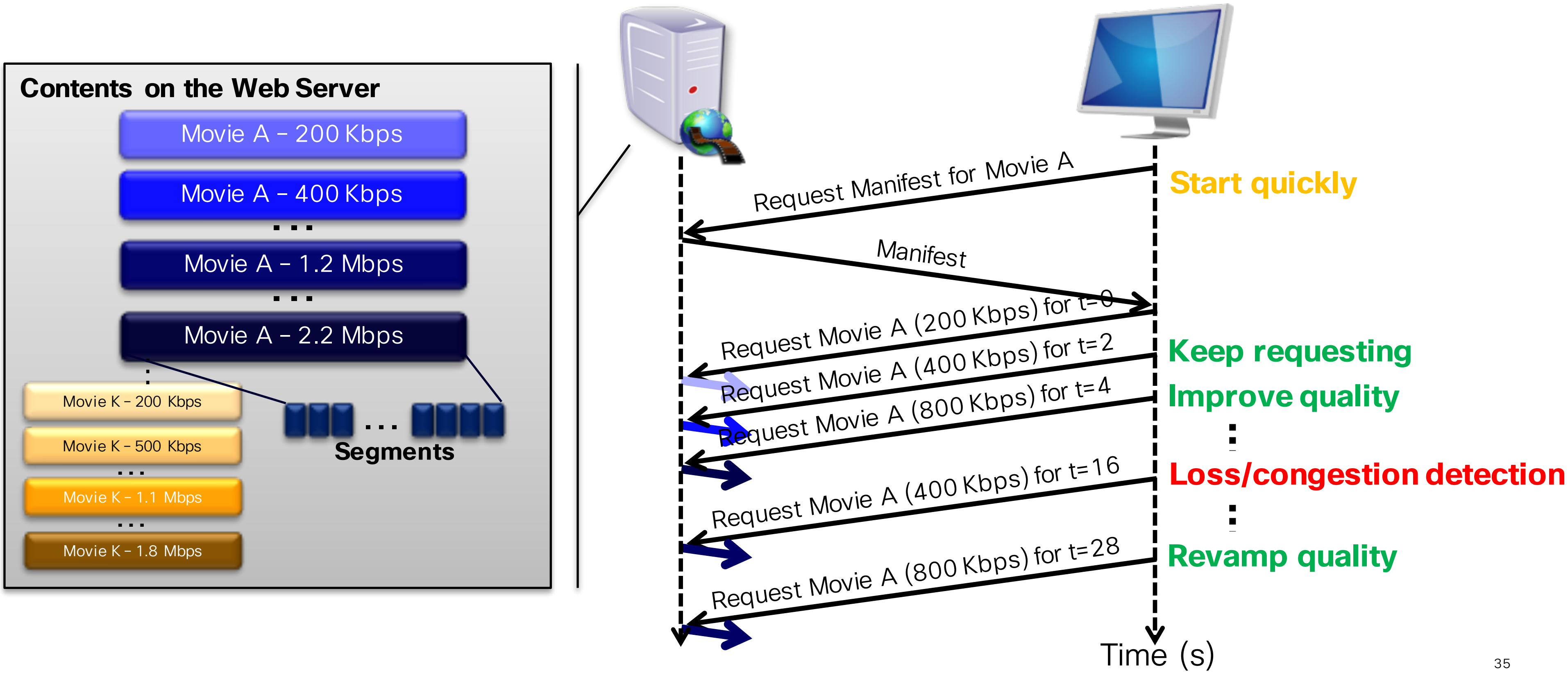
- OTT Delivery and Example Services
- Media Delivery over the Internet
- HTTP Adaptive Streaming Building Blocks
- Workflows for Content Generation, Distribution and Consumption
- Overview of the MPEG DASH Standard

Adaptive Streaming over HTTP

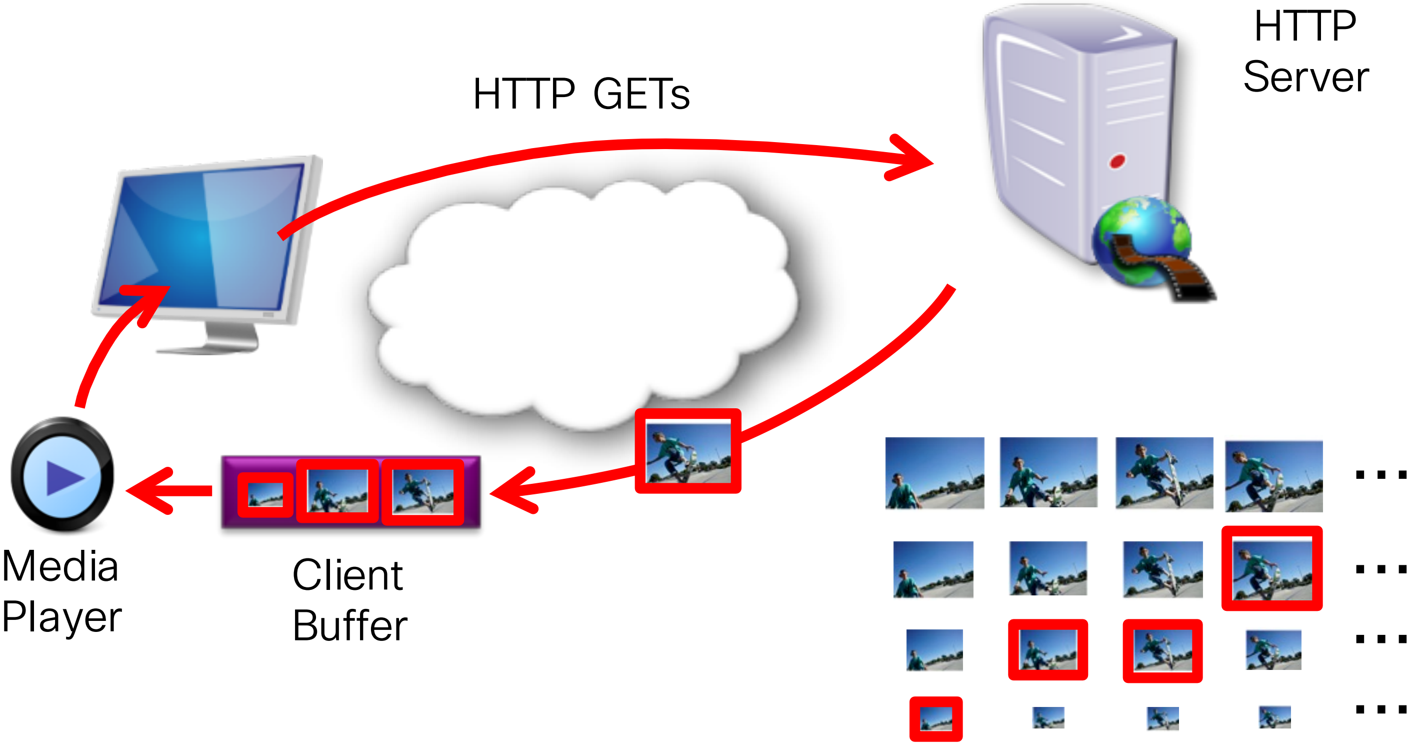
Adapt Video to Web Rather than Changing the Web

- **Imitation of Streaming via Short Downloads**
  - Downloads desired portion in small chunks to minimize bandwidth waste
  - Enables monitoring consumption and tracking clients
- **Adaptation to Dynamic Conditions and Device Capabilities**
  - Adapts to dynamic conditions anywhere on the path through the Internet and/or home network
  - Adapts to display resolution, CPU and memory resources of the client
  - Facilitates “any device, anywhere, anytime” paradigm
- **Improved Quality of Experience**
  - Enables faster start-up and seeking (compared to progressive download), and quicker buffer fills
  - Reduces skips, freezes and stutters
- **Use of HTTP**
  - Well-understood naming/addressing approach, and authentication/authorization infrastructure
  - Provides easy traversal for all kinds of middleboxes (e.g., NATs, firewalls)
  - Enables cloud access, leverages existing HTTP caching infrastructure (Cheaper CDN costs)

Multi-Bitrate Encoding and Representation Shifting



Adaptive Streaming over HTTP



Example Representations

Vancouver 2010

	Encoding Bitrate	Resolution
Rep. #1	3.45 Mbps	1280 x 720
Rep. #2	1.95 Mbps	848 x 480
Rep. #3	1.25 Mbps	640 x 360
Rep. #4	900 Kbps	512 x 288
Rep. #5	600 Kbps	400 x 224
Rep. #6	400 Kbps	312 x 176

Sochi 2014

	Encoding Bitrate	Resolution
Rep. #1	3.45 Mbps	1280 x 720
Rep. #2	2.2 Mbps	960 x 540
Rep. #3	1.4 Mbps	960 x 540
Rep. #4	900 Kbps	512 x 288
Rep. #5	600 Kbps	512 x 288
Rep. #6	400 Kbps	340 x 192
Rep. #7	200 Kbps	340 x 192

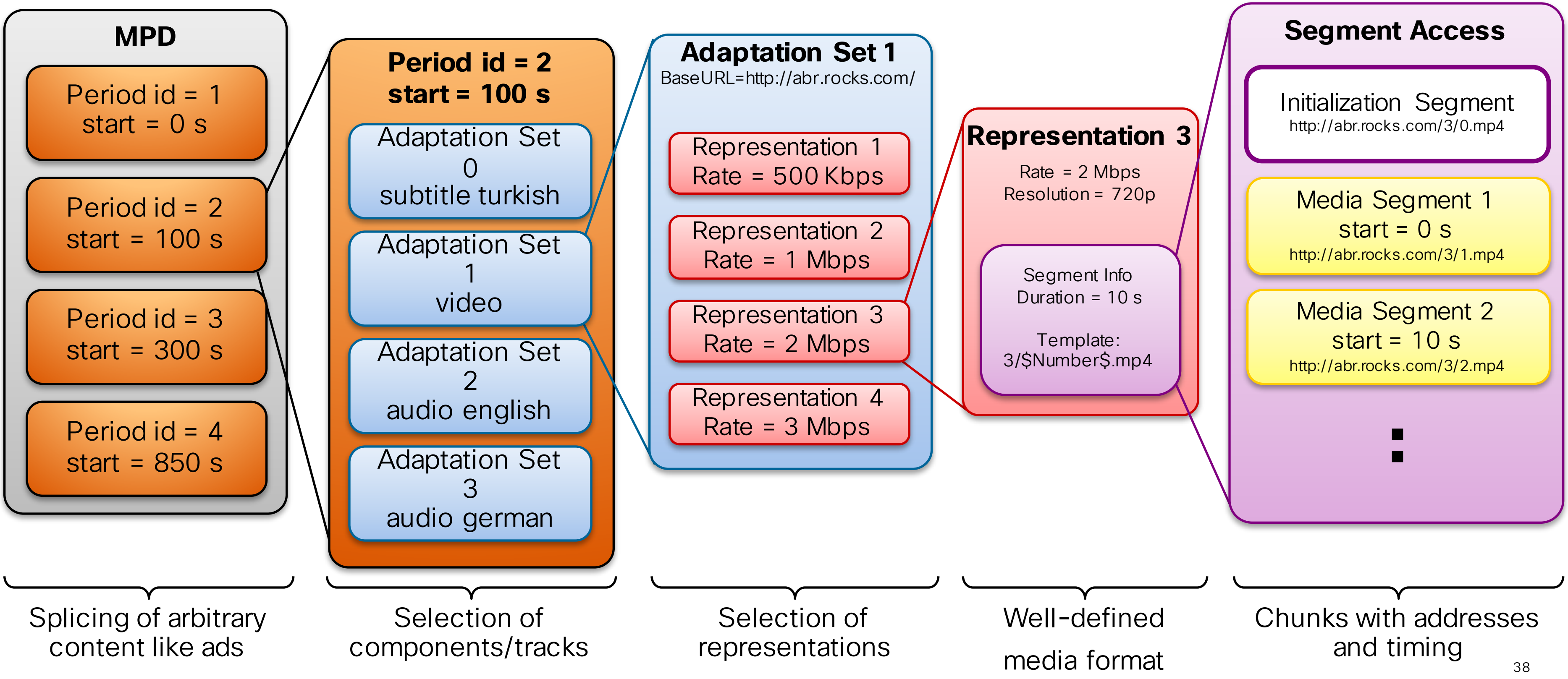
FIFA 2014

	Encoding Bitrate	Resolution
Rep. #1	3.45 Mbps	1280 x 720
Rep. #2	2.2 Mbps	1024 x 576
Rep. #3	1.4 Mbps	768 x 432
Rep. #4	950 Kbps	640 x 360
Rep. #5	600 Kbps	512 x 288
Rep. #6	400 Kbps	384 x 216
Rep. #7	250 Kbps	384 x 216
Rep. #8	150 Kbps	256 x 144

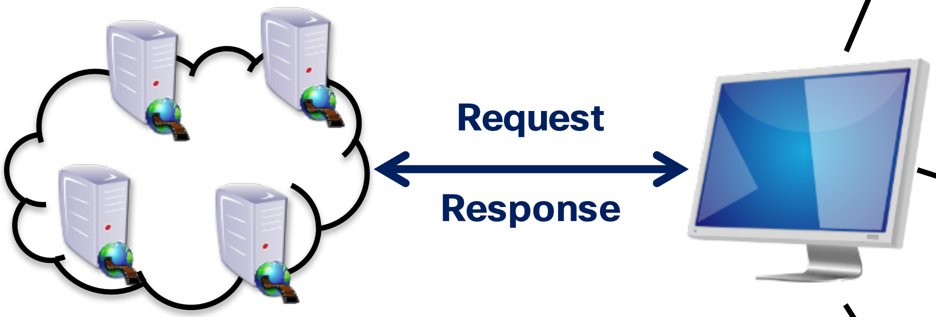
Source: Vertigo MIX10, Alex Zambelli's Streaming Media Blog, Akamai

An Example Manifest Format

List of Accessible Segments and Their Timings



Smart Clients



Client manages

- Manifest(s)
- HTTP transport
- TCP connection(s)

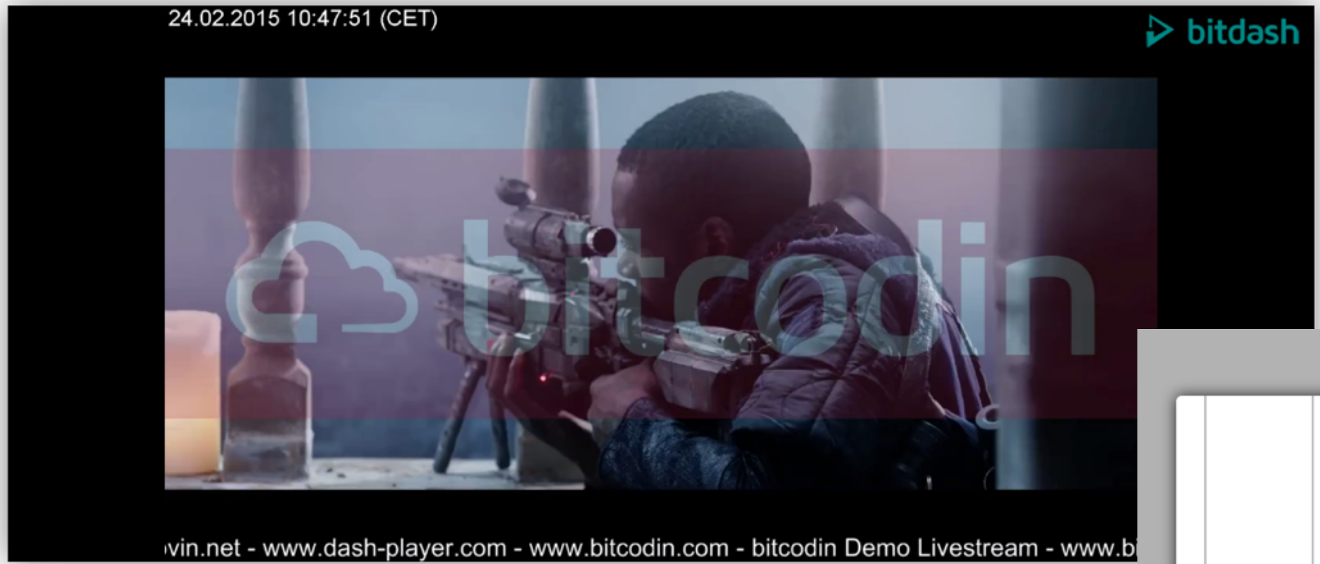
Client monitors/measures

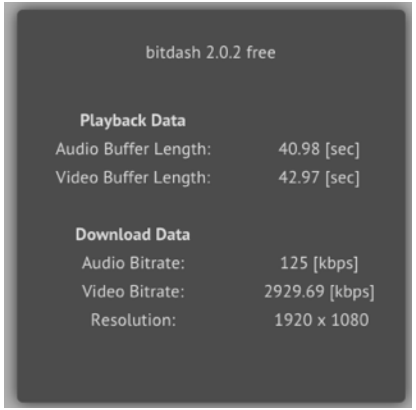
- Playout buffer
- Download times and throughput
- Local resources (CPU, memory, screen, battery, etc.)
- Dropped frames

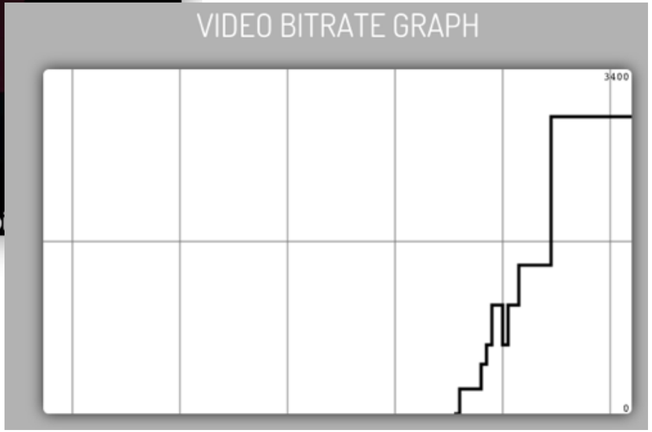
Client performs adaptation

bitdash/bitcodin Showing Adaptation

http://www.dash-player.com/demo/







What about Perceptual Quality?

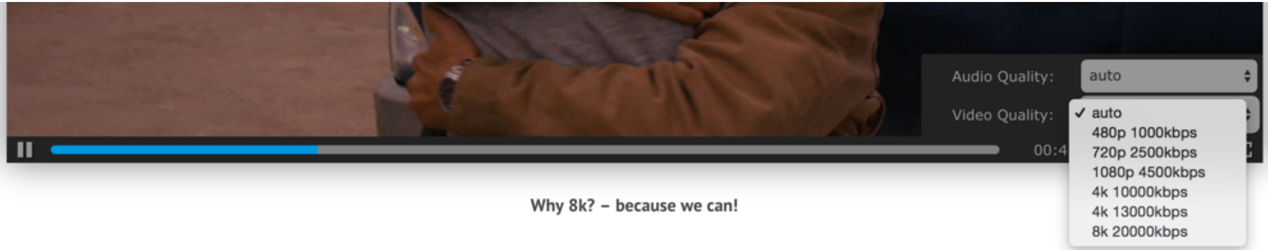
Comparison of two commercially available systems

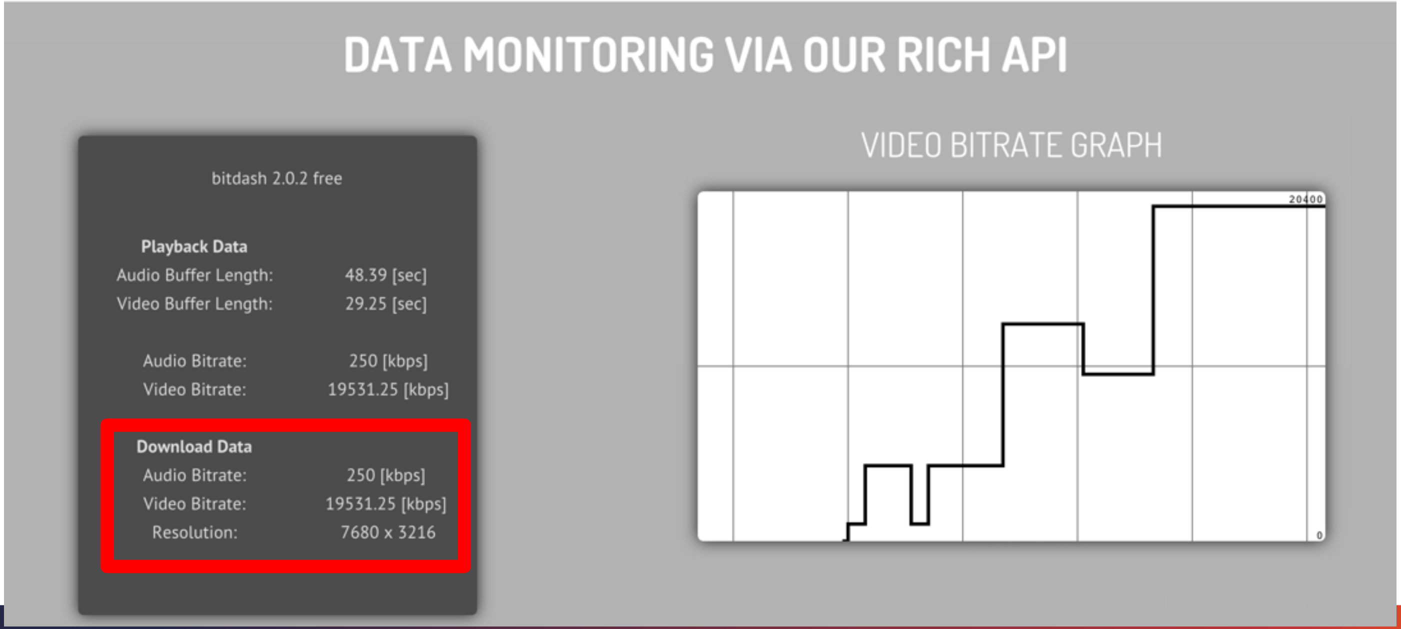
- Feb 19, 2015: Villarreal vs. RB Salzburg

- Quality? Synchronization issues?

8K Streaming! Why? Because we can!

http://www.dash-player.com/demo/8k

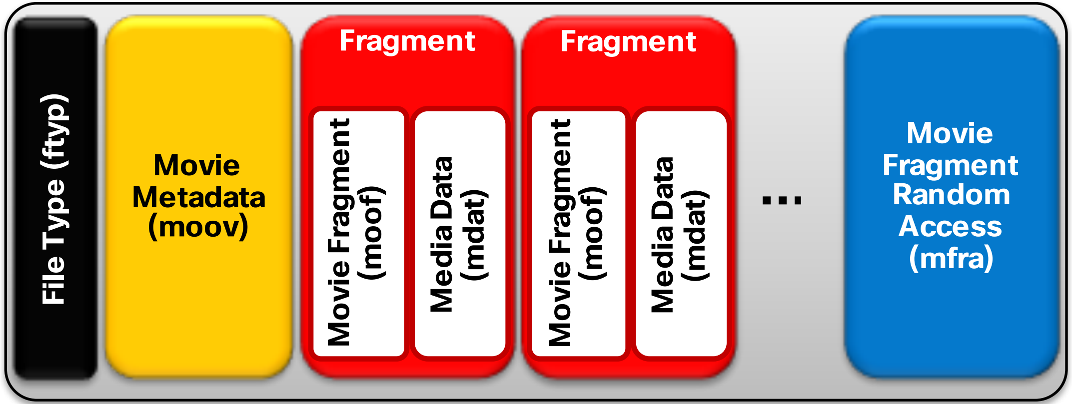




Example Request and Response

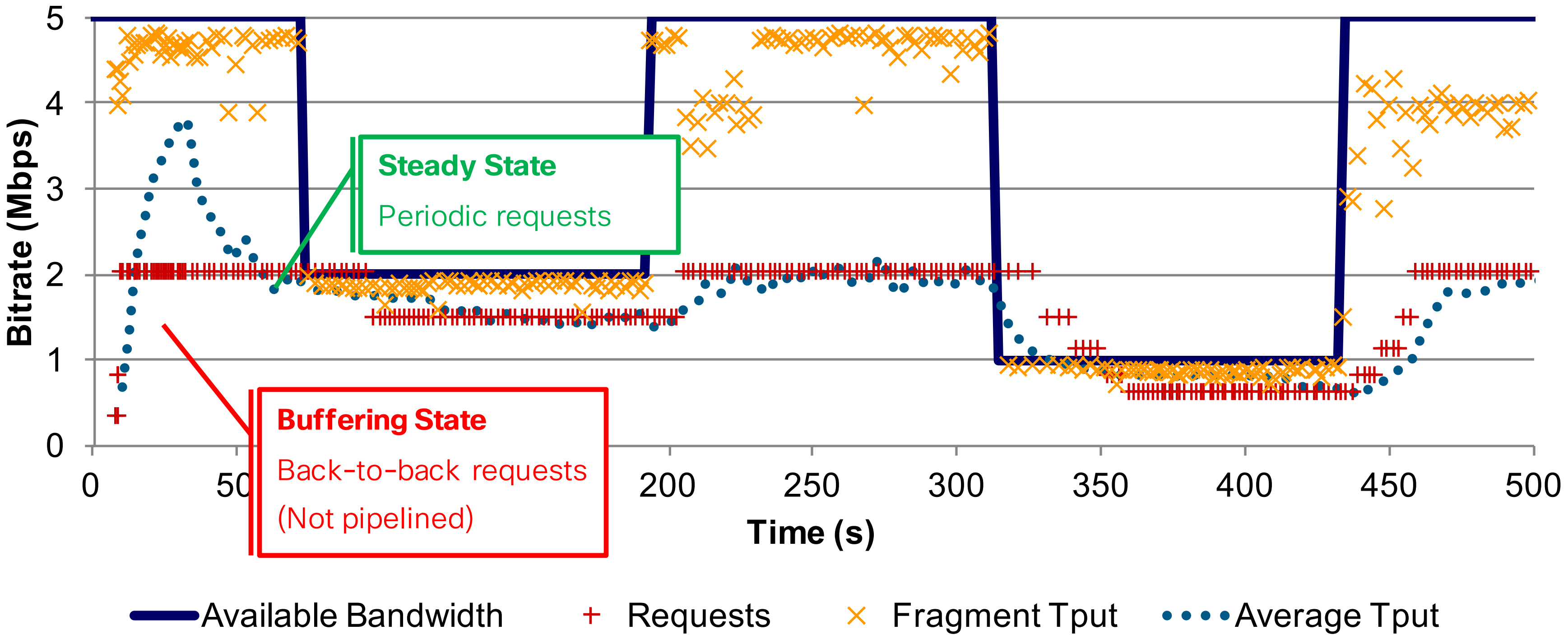
Microsoft Smooth Streaming

- **Client sends an HTTP request**
  - **GET 720p.ism/QualityLevels(572000)/Fragments(video=160577243) HTTP/1.1**
- **Server**
  1. Finds the MP4 file corresponding to the requested bitrate
  2. Locates the fragment corresponding to the requested timestamp
  3. Extracts the fragment and sends it in an HTTP response



Demystifying the Client Behavior

Microsoft Smooth Streaming Experiments



Reading: "An experimental evaluation of rate-adaptation algorithms in adaptive streaming over HTTP," ACM MMSys 2011

Initial and Current Players in the Market

- **Move Adaptive Stream (Now Echostar)**
  - http://www.movenetworks.com
- **Microsoft Smooth Streaming**
  - http://www.iis.net/expand/SmoothStreaming
- **Apple HTTP Live Streaming**
  - http://tools.ietf.org/html/draft-pantos-http-live-streaming
  - http://developer.apple.com/library/ios/#documentation/networkinginternet/conceptual/streamingmediaguide/
- **Netflix**
  - http://www.netflix.com
- **Adobe HTTP Dynamic Streaming**
  - http://www.adobe.com/products/httpdynamicstreaming/
- **bitmovin**
  - bitdash: http://dash-player.com/
- **DASH-IF List of Clients**
  - http://dashif.org/clients/

Where does the Market Stand (Today)?

Fragmented!

Codecs

- AVC (Standard)
- HEVC (Being deployed)
- VC-1 (Almost Dead)
- WebM (Probably hype)

DRM

- Microsoft PlayReady
- Adobe Flash Access
- Marlin
- NDS VGC

Clients

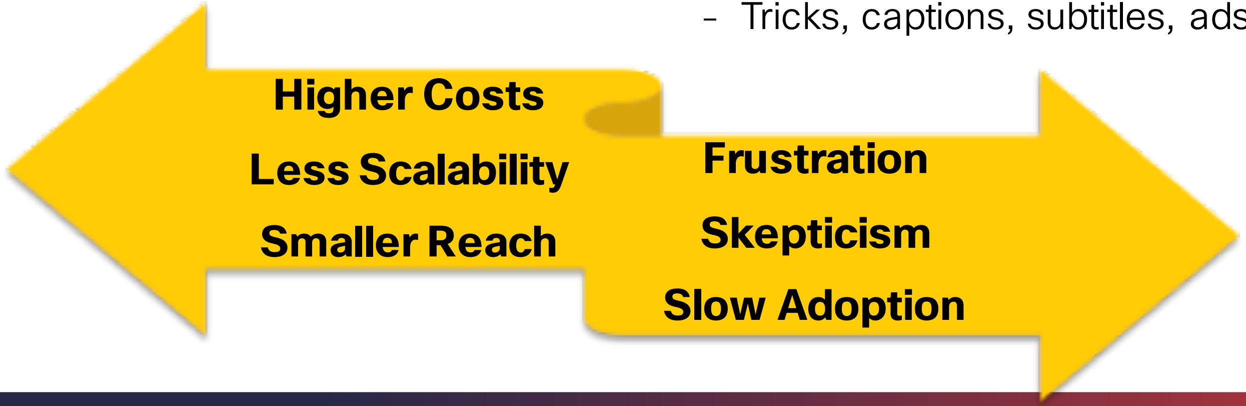
- iOS/Android devices
- HTML5 browsers
- Connected TVs
- Game consoles

Protocols

- Apple HLS
- Microsoft Smooth
- Adobe HDS
- MPEG-DASH?

What does This Mean?

- Fragmented architectures
  - Advertising, DRM, metadata, blackouts, etc.
- Investing in more hardware and software
  - Increased CapEx and OpEx
- Lack of consistent analytics
- Preparing and delivering each asset in several incompatible formats
  - Higher storage and transport costs
- Confusion due to the lack of skills to troubleshoot problems
- Lack of common experience across devices for the same service
  - Tricks, captions, subtitles, ads, etc.



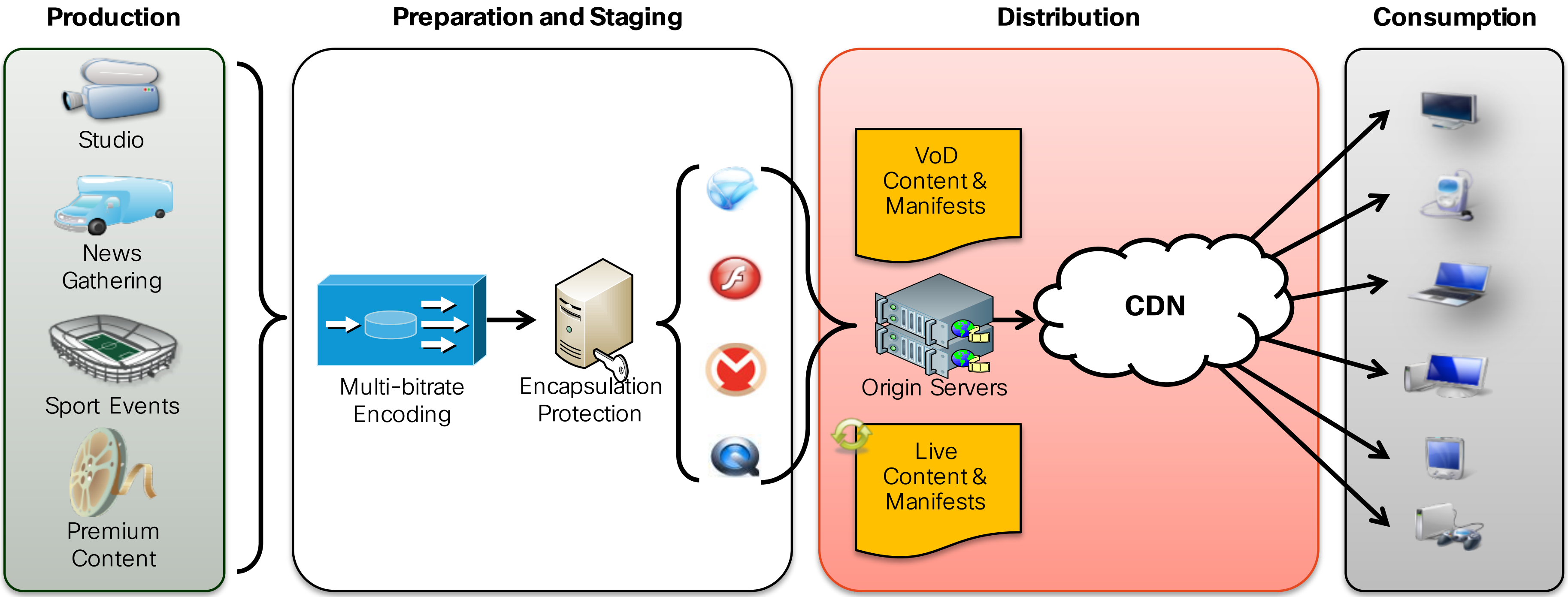
More Details Later...

DASH intends to be to the Internet world ... what MPEG2-TS has been to the broadcast world

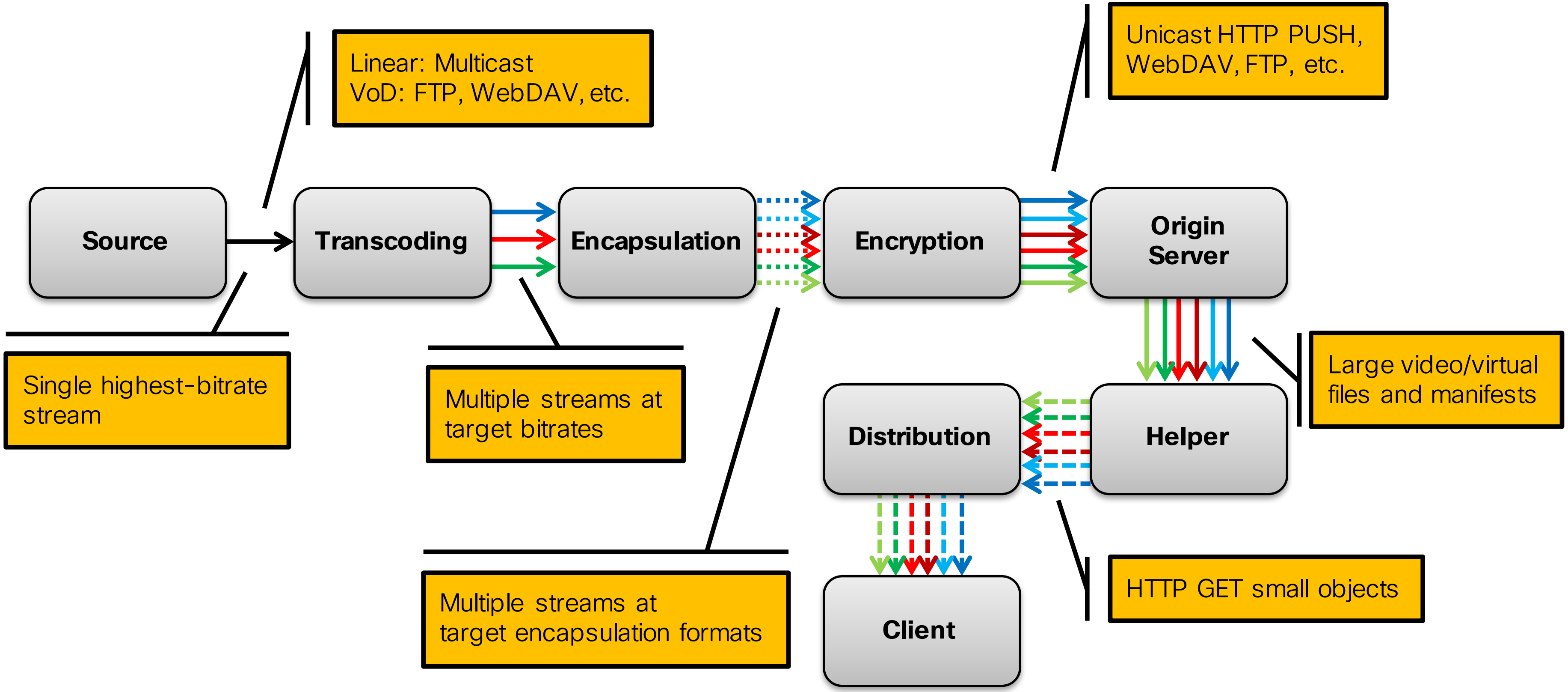
Part I: Over-the-Top (OTT) Video and HTTP Adaptive Streaming

- OTT Delivery and Example Services
- Media Delivery over the Internet
- HTTP Adaptive Streaming Building Blocks
- Workflows for Content Generation, Distribution and Consumption
- Overview of the MPEG DASH Standard

End-to-End Over-the-Top Adaptive Streaming Delivery



Adaptive Streaming Content Workflow



Source Representation

	Container	Manifest	Packaging Tools
Move	2-s chunks (.qss)	Binary (.qmx)	Proprietary
Apple HLS	Fixed-duration MPEG2-TS segments (.ts)	Text (.m3u8)	Several vendors
Adobe Zeri	Aggregated MP4 fragments (.f4f – a/v interleaved)	Client: XML + Binary (.fmf) Server: Binary (.f4x)	Adobe Packager
Microsoft Smooth	Aggregated MP4 fragments (.isma, .ismv – a/v non-interleaved)	Client: XML (.ismc) Server: SMIL (.ism)	Several vendors MS Expression
MPEG DASH	MPEG2-TS and MP4 segments	Client/Server: XML	Several vendors

- **Source containers and manifest files are output as part of the packaging process**
  - These files are staged on to origin servers
  - Some origin server implementations could have integrated packagers
- **Adobe/Microsoft allow to convert aggregated containers into individual fragments on the fly**
  - In Adobe Zeri , this function is called a Helper
  - In Microsoft Smooth, this function is tightly integrated as part of the IIS
- **Server manifest is used by Helper modules to convert the large file into individual fragments**

Staging and Distribution

	Origin Server	Packager → OS Interface	Distribution
Move	Any HTTP server	DFTP, HTTP, FTP	Plain Web caches
Apple HLS	Any HTTP server	HTTP, FTP, CIFS	Plain Web caches
Adobe Zeri	HTTP server with Helper	Integrated packager for live and JIT VoD Offline packager for VoD (HTTP, FTP, CIFS, etc.)	Plain Web caches → Helper running in OS Intelligent caches → Helper running in the delivery edge
Microsoft Smooth	IIS	WebDAV	Plain Web caches Intelligent IIS servers configured in cache mode
MPEG DASH	Any HTTP server	HTTP, FTP, CIFS	Plain Web caches

Delivery

	Client	# of TCP Connections	Transaction Type
Move	Proprietary Move player	3-5	Byte-range requests
Apple HLS	QuickTime X	1 (interleaved)	Whole-segment requests Byte-range requests (iOS5)
Adobe Zeri	OSMF client on top Flash player	Implementation dependent	Whole-fragment access Byte-range access
Microsoft Smooth	Built on top of Silverlight	2 (One for audio and video)	Whole-fragment requests
MPEG DASH	DASH client	Implementation dependent	Whole-segment requests Byte-range requests

- **In Smooth, fragments are augmented to contain timestamps of future fragments in linear delivery**
  - Thus, clients fetch the manifest only once
- **In HLS, manifest is continuously updated**
  - Thus, clients constantly request the manifest

Issues for Content and Service Providers

- Technologies that enabled rapid innovation for IP video delivery to diverse CE endpoints has also created incompatible implementations
  - Players
  - Streaming methods
  - DRM methods
  - Screen sizes, etc.
- Innovation is being driven by CE vendors, not by service or content providers
  - SPs have a significant investment in MPEG2-TS infrastructure and want to leverage existing investments where possible
- Serving each client in its native technology requires creation, storage and delivery of multiple formats and representations

Two high-level options for service delivery

- Transform in the cloud to create media for each client in its native media format
- Serve a common format from the cloud and transform client behavior via apps/plugins

Transform Content in the Cloud

Pros vs. Cons

Pros

- Optimal performance on clients by using their native formats and delivery methods
- Potentially better customer experience through integration with the native player capabilities
- Easier to manage services in the cloud than to manage client app versioning
- Better service velocity (re-use existing client capabilities)
- Ability to transform content for use across multiple client platforms (future-proof)
- Ability to reach across new and legacy systems

Cons

- Additional encode/encapsulation/encrypt processing resources
- Additional storage for multiple representations
- Development effort to support new formats

Transform Content at the Client

Pros vs. Cons

Pros

- Minimized cloud processing resources for encoding, encapsulation and encryption
- Minimized content storage in the cloud due to a single representation
  - Codec
  - Encapsulation
  - Encryption
- More efficient cache utilization in the distribution network
- Potentially, common player ingest from the cloud drives common behavior across client platforms

Cons

- Some target devices will not be using their native player
- Suboptimal rendering quality, battery life, etc. by not using hardware optimizations
- Harder to integrate with native media player features and leverage inter-app capabilities
- Ties the service provider tightly into a third-party relationship
- Third-party tools may not exist across all client platforms
- Unknown willingness of some client manufacturers for approval process

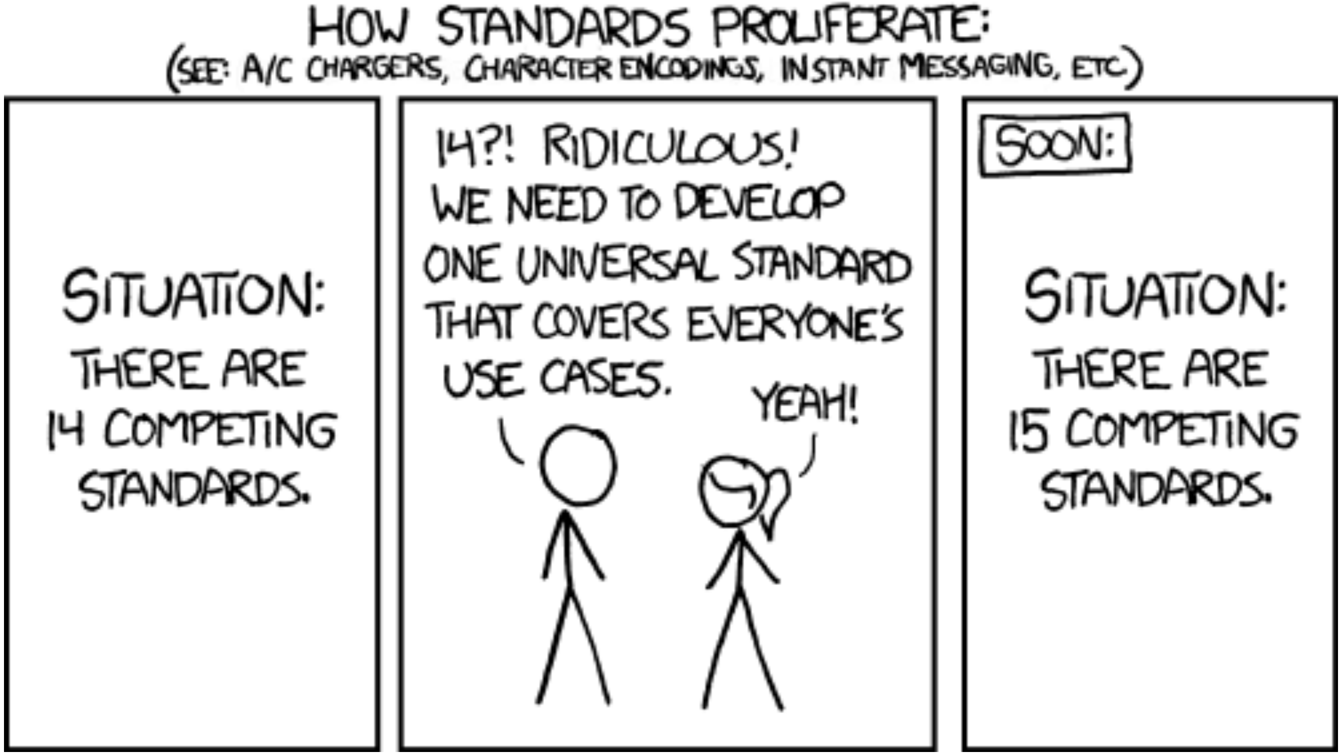
Part I: Over-the-Top (OTT) Video and HTTP Adaptive Streaming

- OTT Delivery and Example Services
- Media Delivery over the Internet
- HTTP Adaptive Streaming Building Blocks
- Workflows for Content Generation, Distribution and Consumption
- Overview of the MPEG DASH Standard

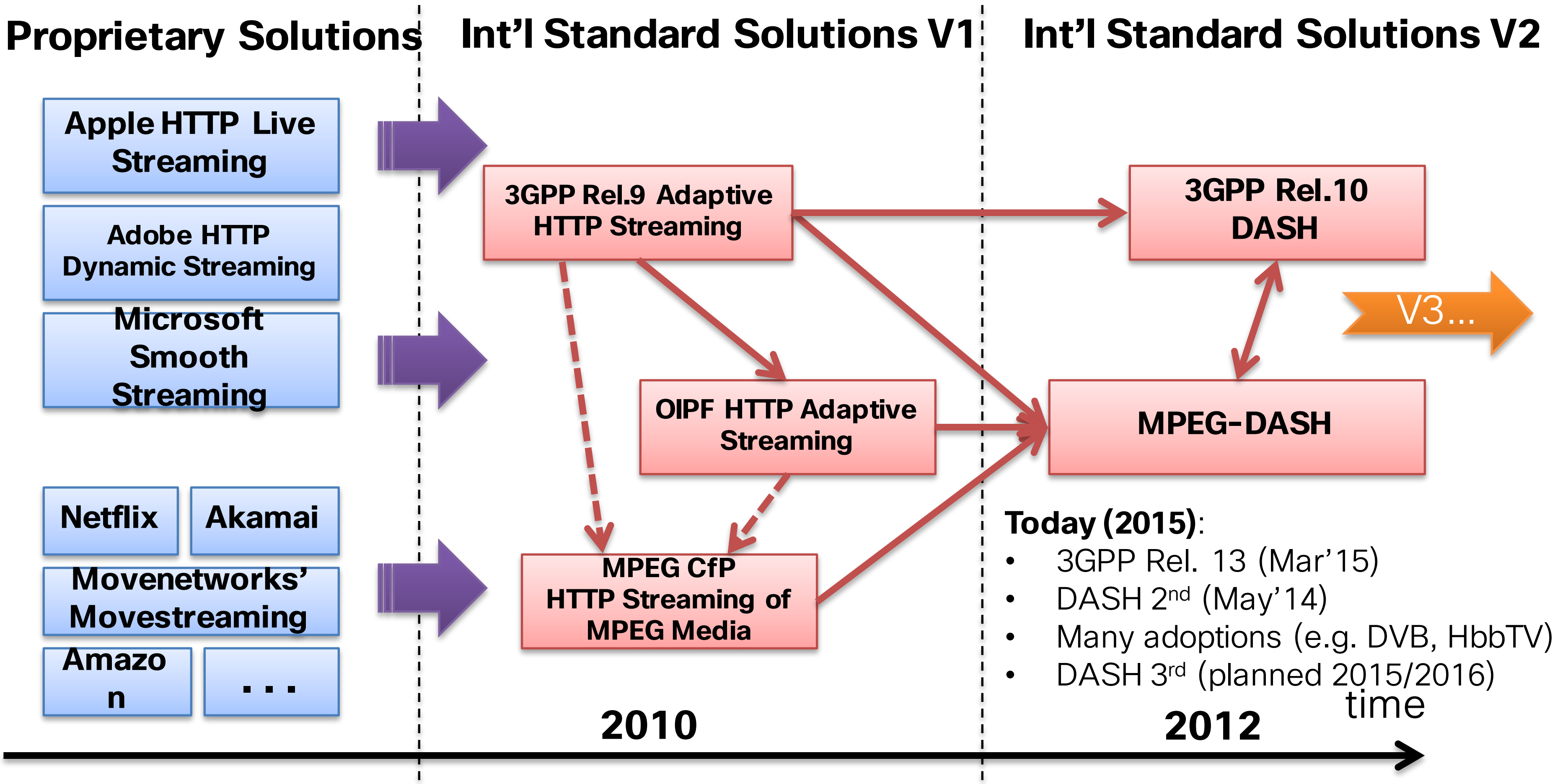
What is DASH?

Reading: http://en.wikipedia.org/wiki/Dash_(disambiguation)

Initial Situation



Source: http://xkcd.com/927/



Today (2015):

- 3GPP Rel. 13 (Mar’15)
- DASH 2nd (May’14)
- Many adoptions (e.g. DVB, HbbTV)
- DASH 3rd (planned 2015/2016)

Reading: http://multimediacommunication.blogspot.com/2010/05/http-streaming-of-mpeg-media.html

MPEG – Dynamic Adaptive Streaming over HTTP

A New Standard: ISO/IEC 23009

- **Goal**
  - Develop an international, standardized, efficient solution for HTTP-based streaming of MPEG media
- **Some Objectives**
  - Do the necessary, avoid the unnecessary
  - Be lazy: reuse what exists in terms of codecs, formats, content protection, protocols and signaling
  - Be backward-compatible (as much as possible) to enable deployments aligned with existing proprietary technologies
  - Be forward-looking to provide ability to include new codecs, media types, content protection, deployment models (ad insertion, trick modes, etc.) and other relevant (or essential) metadata
  - Enable efficient deployments for different use cases (live, VoD, time-shifted, etc.)
  - Focus on formats describing functional properties for adaptive streaming, not on protocols or end-to-end systems or implementations
  - Enable application standards and proprietary systems to create end-to-end systems based on DASH formats
  - Support deployments by conformance and reference software, implementation guidelines, etc.

ISO/IEC 23009-1 Timeline

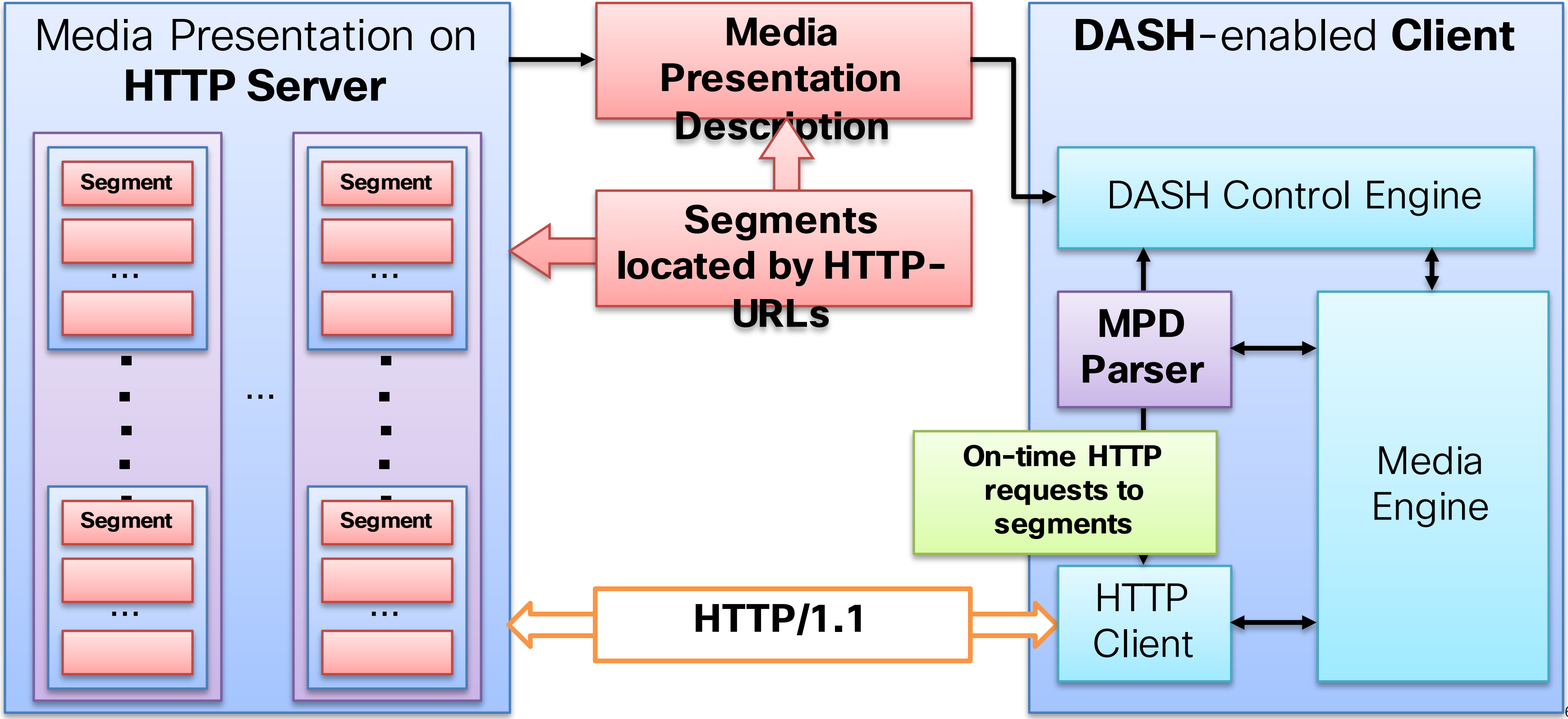
CfP Issued April 2010 → 18 Responses and Working Draft (WD) July 2010 → Committee Draft (CD) Oct. 2010 → Draft International Standard (DIS) Jan. 2011 → Final Draft International Standard Nov. 2011 → Published as International Standard April 2012

Fastest time ever that a standard was developed in MPEG to address the demand of the market

- **Other Relevant Specifications**
  - 14496-12: ISO Base Media File Format
  - 14496-15: Advanced Video Coding (AVC) File Format
  - 23001-7: Common Encryption in ISO-BMFF
  - 23001-8: Coding-Independent Code Points
  - 23001-10: Carriage of Timed Metadata Metrics of Media in ISO Base Media File Format

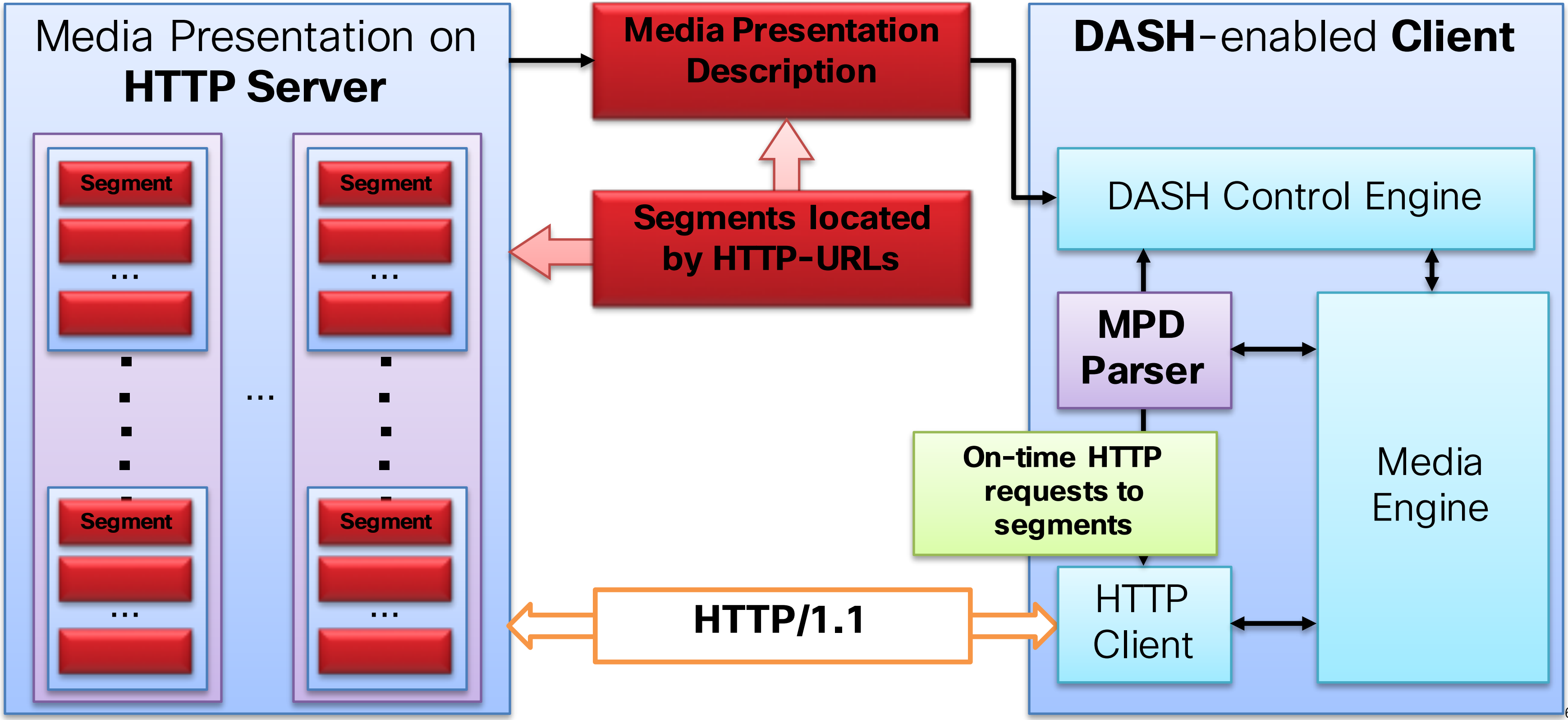
Scope of MPEG DASH

What is specified – and what is not?



Scope of MPEG DASH

What is **specified** – and what is not?

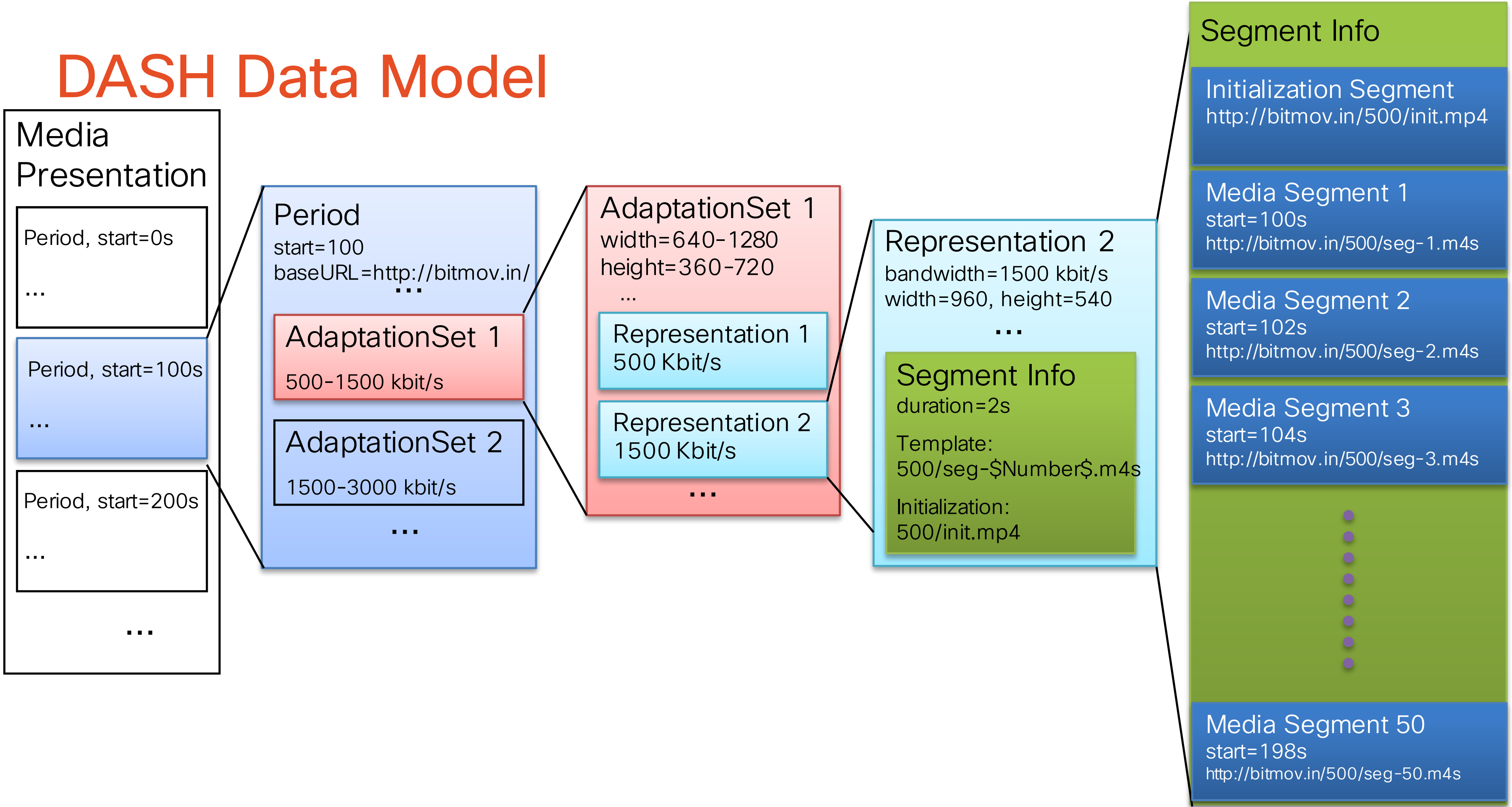


DASH Design Principles

- DASH is not
  - system, protocol, presentation, codec, interactivity, DRM, client specification
- DASH is an enabler
  - It provides formats to enable efficient and high-quality delivery of streaming services over the Internet
  - It is considered as one component in an end-to-end service
  - System definition left to other organizations (SDOs, fora, companies, etc.)
- Design choices
  - Enable reuse of existing technologies (containers, codecs, DRM etc.)
  - Enable deployment on top of HTTP-CDNs (Web Infrastructures, caching)
  - Enable very high user-experience (low start-up, no rebuffering, trick modes)
  - Enable selection based on network and device capability, user preferences
  - Enable seamless switching
  - Enable live and DVD-kind of experiences
  - Move intelligence from network to client, enable client differentiation
  - Enable deployment flexibility (e. g., live, on-demand, time-shift viewing)
  - Provide simple interoperability points (profiles)

Ack & ©: Thomas Stockhammer

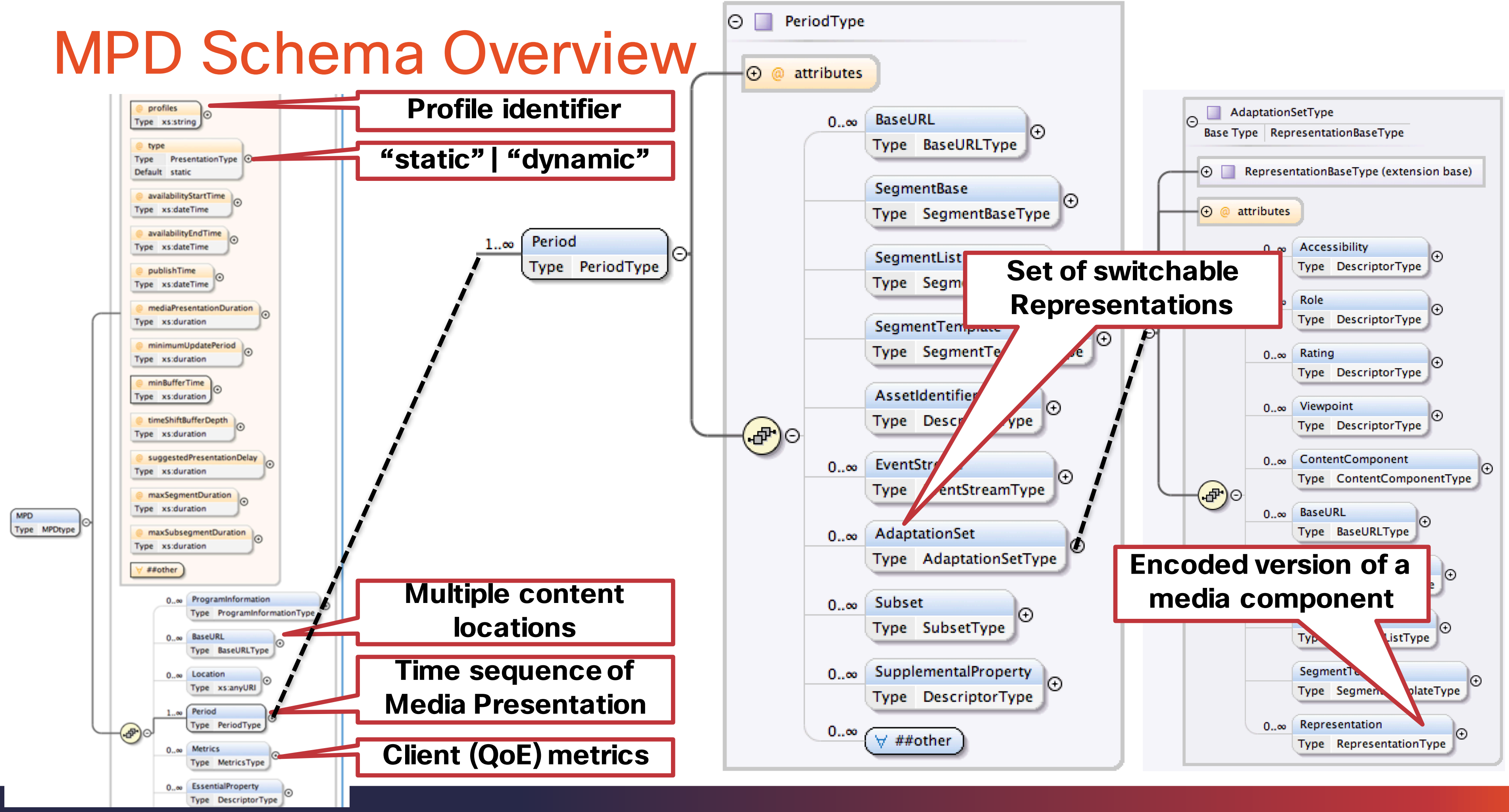
DASH Data Model



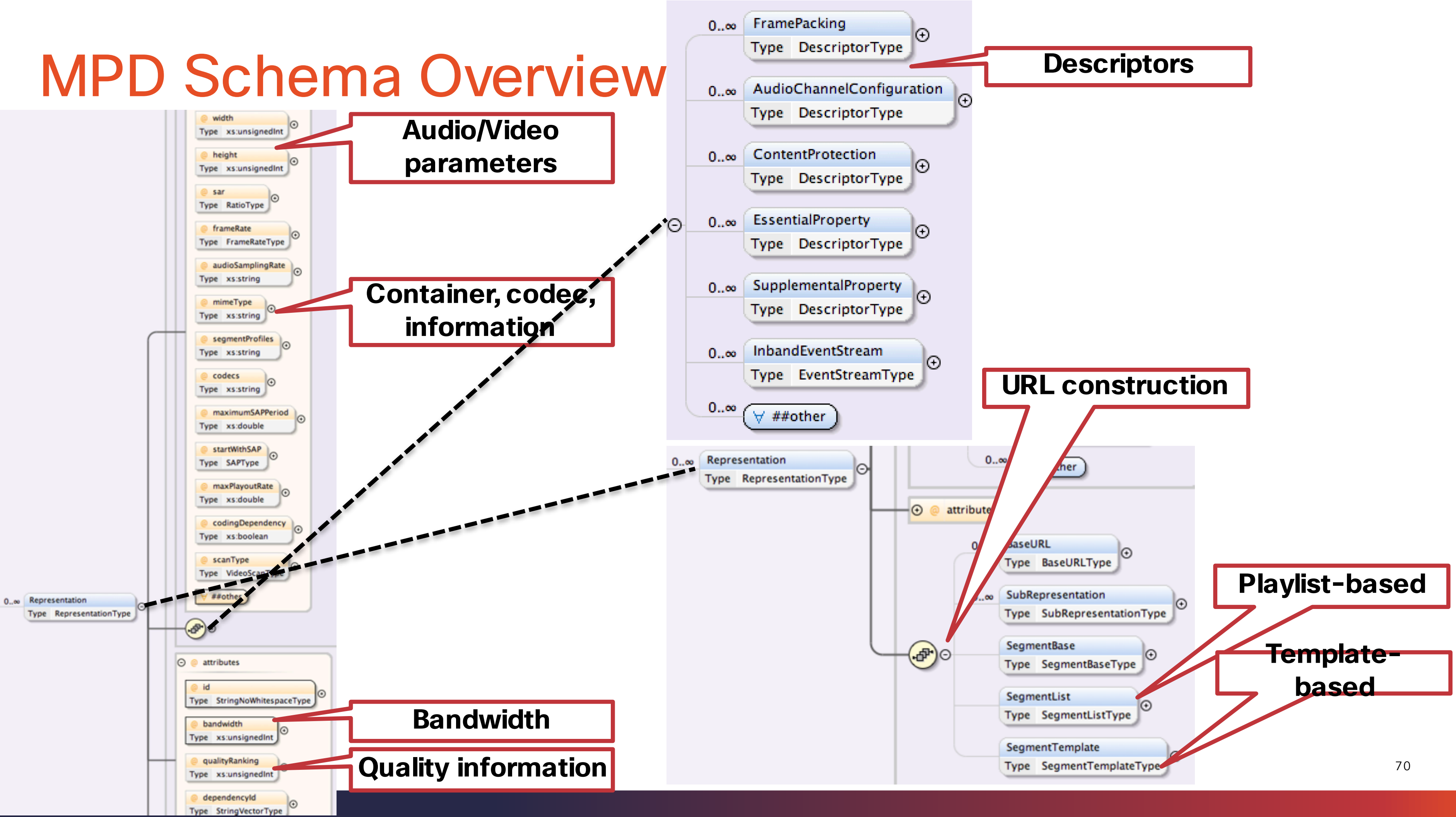
Media Presentation Description

- Redundant information of media streams for the purpose to initially select or reject AdaptationSets of Representations
  - Examples: Codec, DRM, language, resolution, bandwidth
- Access and Timing Information
  - HTTP-URL(s) and byte range for each accessible Segment
  - Earliest next update of the MPD on the server
  - Segment availability start and end time in wall-clock time
  - Approximated media start time and duration of a media segment in the media presentation timeline
  - For live service, instructions on starting playout such that media segments will be available in time for smooth playout in the future
- Switching and splicing relationships across Representations
- Relatively little other information

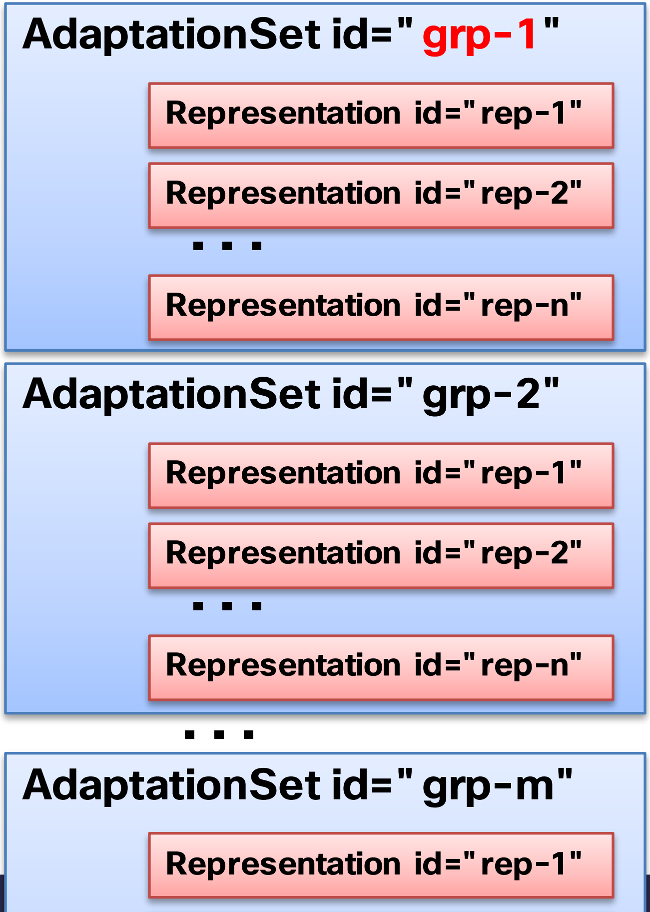
MPD Schema Overview



MPD Schema Overview

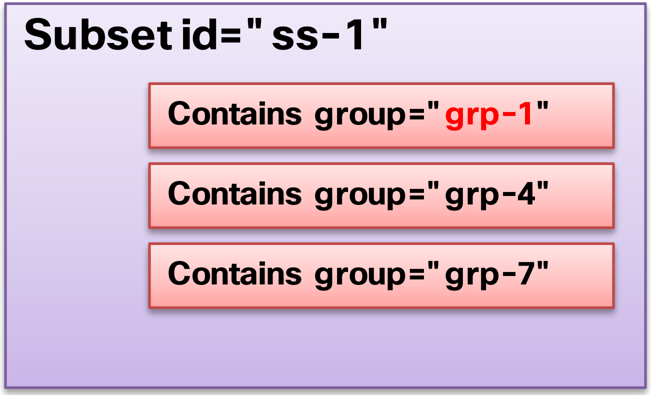


DASH AdaptationSets & Subsets



AdaptationSet by codec, language, resolution, bandwidth, views, etc. – very flexible (in combination with xlink)!

- Ranges for the @bandwidth, @width, @height and @frameRate



Subsets

- Mechanism to restrict the combination of *active* Groups
- Expresses the intention of the creator of the Media Presentation

Segment Indexing

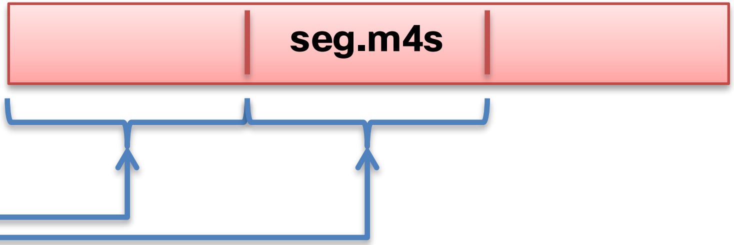
- Provides binary information in ISO box structure on
  - Accessible units of data in a media segment
  - Each unit is described by
    - Byte range in the segments (easy access through HTTP partial GET)
    - Accurate presentation duration (seamless switching)
    - Presence of representation access positions, e.g. IDR frames
- Provides a compact bitrate-over-time profile to client
  - Can be used for intelligent request scheduling
- Generic Data Structure usable for any media segment format, e.g. ISO BMFF, MPEG-2 TS, etc.
- Hierarchical structuring for efficient access
- May be combined with media segment or may be separate

Ack & ©: Thomas Stockhammer

Segment Index in MPD only

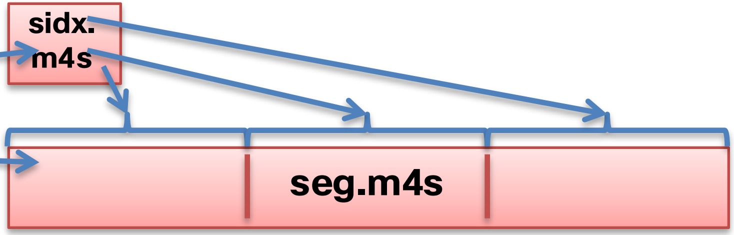
```
<MPD>
  ...
  <URL sourceURL="seg1.m4s"/>
  <URL sourceURL="seg2.m4s"/>
</MPD>
```


```
<MPD>
  ...
  <URL sourceURL="seg.m4s"  range="0-499"/>
  <URL sourceURL="seg.m4s"  range="500-999"/>
</MPD>
```



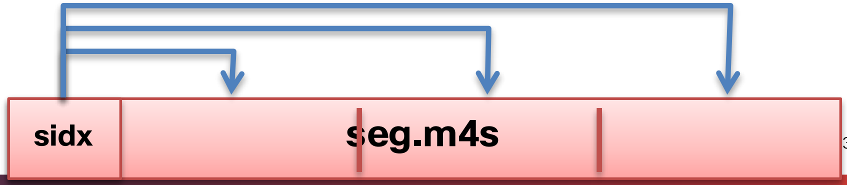
Segment Index in MPD + Segment

```
<MPD>
  ...
  <Index sourceURL="sidx.mp4"/>
  <URL sourceURL="seg.m4s"/>
</MPD>
```



Segment Index in Segment only

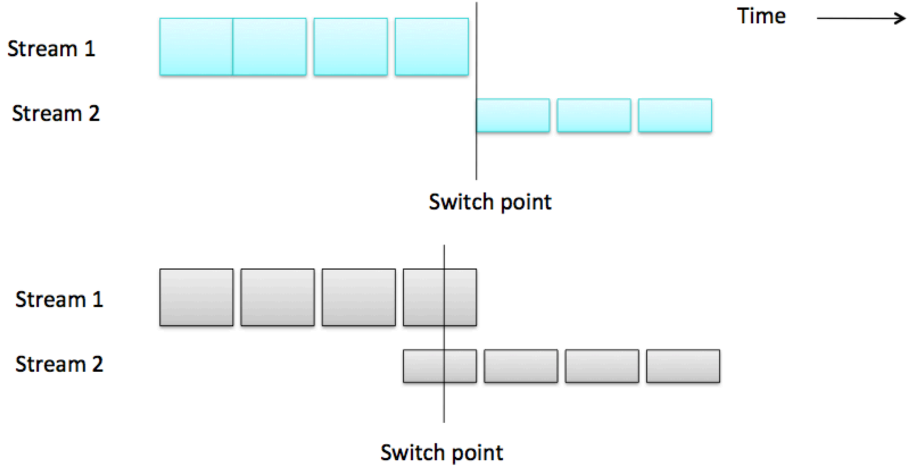
```
<MPD>
  ...
  <BaseURL>seg.m4s</BaseURL>
</MPD>
```



Switch Point Alignment

- Segment alignment
  - Permits non-overlapping decoding and presentation of segments from different representations
- Stream Access Points (SAPs)
  - Presentation time and position in segments at which random access and switching can occur
- Bitstream Switching
  - Concatenation of segments from different representations results in conforming bitstream
- Alignment and SAPs can also apply for subsegments

- Preferable switching points are segment/subsegment boundaries for which
  - Alignment holds across representations
  - The switch-to representation starts with a SAP



```xml
▼<MPD xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="urn:mpeg:dash:schema:mpd:2011" xsi:sc
 http://standards.iso.org/ittf/PubliclyAvailableStandards/MPEG-DASH_schema_files/DASH-MPD.xsd" type="sta
 maxSegmentDuration="PT5S" minBufferTime="PT1S" mediaPresentationDuration="PT0H12M14.00S" profiles="urn:
 ▼<Period id="1" start="PT0S">
    <BaseURL>http://demo.bitmovin.net/data/4k/tearsofsteel/</BaseURL>
  ▼<AdaptationSet mimeType="text/vtt" lang="en">
    ▼<Representation id="caption_en" bandwidth="256">
      ▼<BaseURL>
          http://demo.bitmovin.net/data/4k/tearsofsteel/TOS-de.vtt
        </BaseURL>
      </Representation>
    </AdaptationSet>
  ▼<AdaptationSet group="1" mimeType="audio/mp4" minBandwidth="65439" maxBandwidth="193714" segmentAli
    ▼<Representation id="1" bandwidth="193714" codecs="mp4a.40.2" audioSamplingRate="44100">
        <SegmentTemplate timescale="1000" duration="3993" media="audio/192k/segment_$Number$.m4s" initia
      </Representation>
    ▼<Representation id="2" bandwidth="129714" codecs="mp4a.40.2" audioSamplingRate="44100">
        <SegmentTemplate timescale="1000" duration="3993" media="audio/128k/segment_$Number$.m4s" initia
      </Representation>
    ▼<Representation id="3" bandwidth="97714" codecs="mp4a.40.2" audioSamplingRate="44100">
        <SegmentTemplate timescale="1000" duration="3993" media="audio/96k/segment_$Number$.m4s" initial
      </Representation>
    ▼<Representation id="4" bandwidth="65439" codecs="mp4a.40.2" audioSamplingRate="44100">
        <SegmentTemplate timescale="1000" duration="3993" media="audio/64k/segment_$Number$.m4s" initial
      </Representation>
    </AdaptationSet>
  ▼<AdaptationSet group="2" mimeType="video/mp4" par="1:1" minBandwidth="353461" maxBandwidth="1020806
    startWithSAP="1">
    ▼<Representation id="350k" frameRate="24" bandwidth="353461" codecs="avc1.64001e" width="806" heigh
        <SegmentTemplate timescale="1000" duration="4000" media="video/350k/segment_$Number$.m4s" initia
      </Representation>
    ▼<Representation id="500k" frameRate="24" bandwidth="501297" codecs="avc1.64001f" width="1075" heig
        <SegmentTemplate timescale="1000" duration="4000" media="video/500k/segment_$Number$.m4s" initia
      </Representation>
    ▼<Representation id="1m" frameRate="24" bandwidth="1024626" codecs="avc1.64001f" width="1613" heigh
        <SegmentTemplate timescale="1000" duration="4000" media="video/1mbit/segment_$Number$.m4s" initi
      </Representation>
    ▼<Representation id="2m" frameRate="24" bandwidth="2037737" codecs="avc1.640032" width="2419" heigh
        <SegmentTemplate timescale="1000" duration="4000" media="video/2mbit/segment_$Number$.m4s" initi
      </Representation>
    ▼<Representation id="10m" frameRate="24" bandwidth="10208069" codecs="avc1.640033" width="3840" hei
        <SegmentTemplate timescale="1000" duration="4000" media="video/10mbit/segment_$Number$.m4s" init
      </Representation>
    </AdaptationSet>
  </Period>
</MPD>
```

type=static typically, for on demand content

Base URL of the segments

Subtitles

Audio adaptation set with different representations (bw)

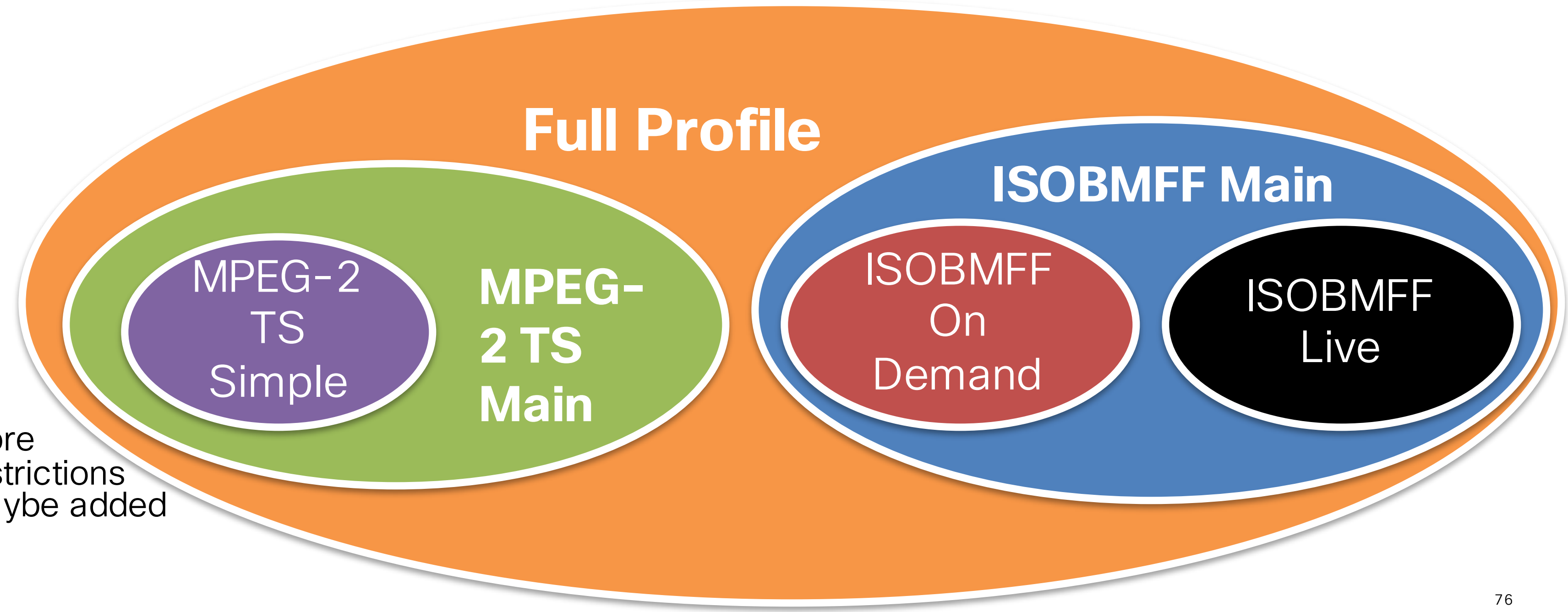
Video adaptation set with different representations (bw)

Different codecs (profiles)

Segment URL constructed with template and base URL

Profiles

- Subset (restrictions) of the functionality, target specific applications/domains
- As of now, mainly related to supported segment formats



- More restrictions maybe added

Major Functional Components – Data Model

- Provide information to a client, where and when to find the data that composes A/V → **MPD**
- Provide the ability to offer a service on the cloud and HTTP-CDNs → **HTTP-URLs and MIME Types**
- Provide service provider the ability to combine/splice content with different properties into a single media presentation → **Periods**
- Provide service provider to enable the client/user selection of media content components based on user preferences, user interaction device profiles and capabilities, using conditions or other metadata → **Adaptation Sets**
- Provide ability to provide the same content with different encodings (bitrate, resolution, codecs) → **Representations**
- Provide extensible syntax and semantics for describing representation/adaptation set properties → **Descriptors**
- Provide ability to access content in small pieces and do proper scheduling of access → **Segments and Subsegments**
- Provide ability for efficient signaling and deployment optimized addressing → **Playlist, Templates, Segment Index**
- Provide ability to enable reuse of existing encapsulation and parsing tools → **MPEG2-TS and ISO-BMFF**

Major Functional Components – Timing

- **Common Media Presentation Time**
  - Provide ability to present content from different adaptation sets synchronously
  - Provide ability to support seamless switching across different representations
- **Switching Support Features**
  - Signaling of Stream Access Points
  - Segment Alignment to avoid overlap downloading and decoding
- **Playout Times per Segment and Track Fragment Decode Times**
  - Provide ability to randomly access and seek in the content
- **Segment Availability Time**
  - Mapped to wall-clock time
  - Expresses when a segment becomes available on the server and when ceases it to be available
  - Provide ability to support live and time-shift buffer services with content generated/removed on the fly

Major Functional Components – Operations

- Provide ability for personalized access to media presentation, e.g. targeted advertisement → **MPD Assembly with xlink**
- Provide ability to provide redundant content offering → **Multiple Base URLs**
- Provide ability to announce unforeseen/unpredictable events in live services → **MPD Updates**
- Provide ability to send events associated with media times → **In-band and MPD-based Event Messages**
- Provide the ability to log and report client actions → **DASH Metrics**
- Provide ability to efficiently support trick modes → **Dedicated IDR-frame Representations and Sub-representations**
- Provide ability to signal collection of a subset/extension of tools → **Profiles and Interoperability Points**

ISO/IEC 23009 Parts

- 23009-1: Media Presentation Description and Segment Formats
  - 2nd edition has been published
  - 1st amendment (high profile and availability time synchronization)
  - 2nd amendment (spatial relationship description, generalized URL parameters, etc.)
  - 3rd amendment (authentication, MPD linking, callback event, period continuity, etc.)
- 23009-2: Conformance and Reference Software
  - 1st edition has been published
  - WD for 2nd edition is in progress, incl. support for ISOBMFF, M2TS and Web-based conformance checking provided by DASH-IF

ISO/IEC 23009 Parts

- 23009-3: Implementation Guidelines
  - 1st edition is done, will be published soon
  - 2nd edition is in progress
  - 1st amendment is in progress
- 23009-4: Segment Encryption and Authentication
  - Published by ISO in 2013
- 23009-5: Server and Network Assisted DASH (SAND)
  - CD is in progress
- 23009-6: DASH over Full Duplex HTTP-based Protocols (FDH)
  - WD is in progress

Ongoing Work in MPEG DASH (as of MPEG 112)

...as of Last week!

- **Currently Running Core Experiments**
  - SAND: Server and Network Assisted DASH
  - FDH: DASH over Full Duplex HTTP-based Protocols
  - SISSI: SAP-Independent Segment Signaling
  - CAPCO: Content Aggregation and Playback Control
- **Technologies under Consideration**
  - Service-level Service Protection Using Segment Encryption
  - Support for 3DV with Depth
  - Support for Controlled Playback in DASH
  - Editorial Adaptation Set Continuity across Periods
  - Playout Continuity of Adaptation Sets across Periods

Server and Network Assisted DASH (SAND)

All Started with a Workshop in July 2013

- A half-day workshop was held on this subject and Cisco gave a joint presentation with Qualcomm
- Program, contributions and slides are available at:
  - http://multimediacommunication.blogspot.co.at/2013/05/mpeg-workshop-on-session-management-and.html

Possible Control Points in the Ecosystem

Content Provider	Advertiser	Service Provider	Content Delivery Network	Operator ISP	Device/ Client	Consumer

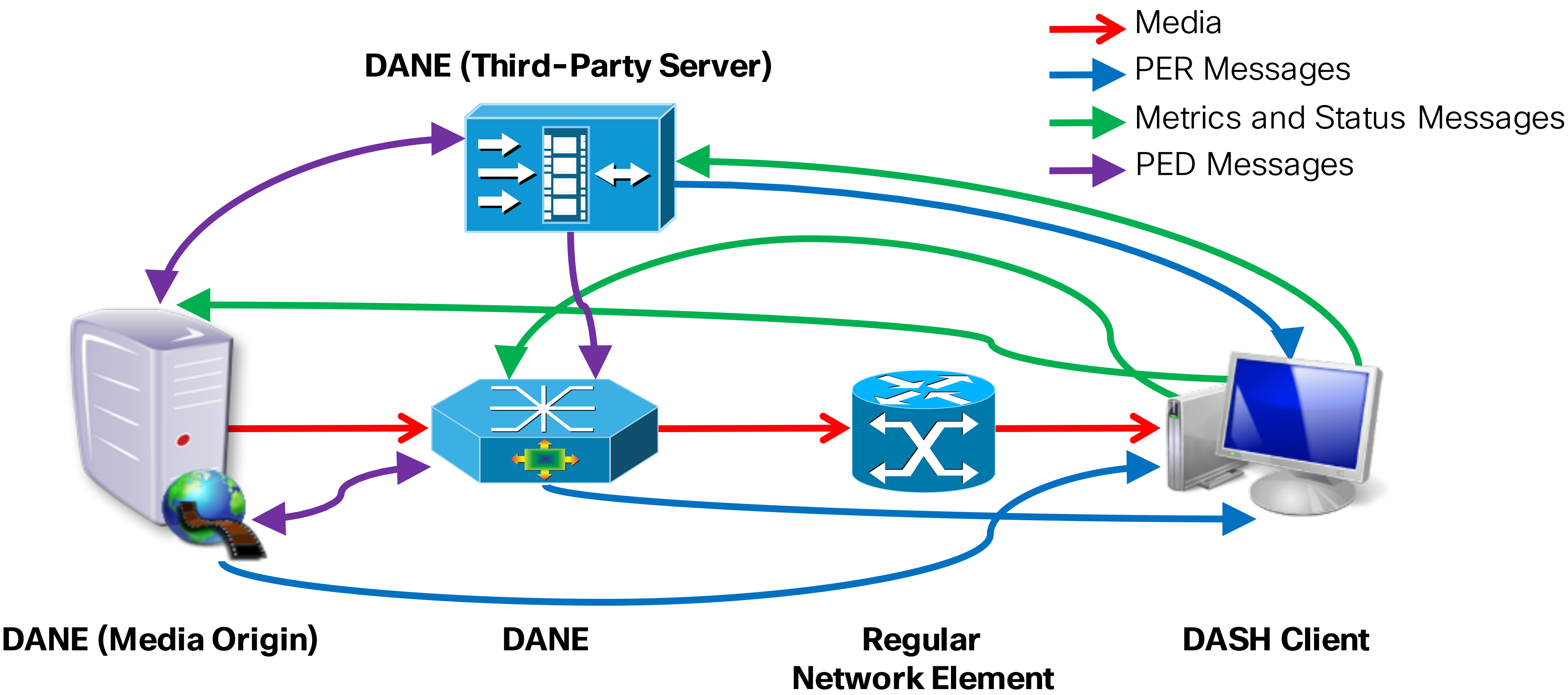
- I want to make sure that I provide the best possible video quality
- I want to control the general quality-of-experience of all my subscribers, potentially differentiate and avoid overload and congestion situations
- I want to make sure that my cheaper distribution is used when it is available
- I want to make sure that my content is protected and does not leak
- I want to make sure that my ad is viewed and I know that it is viewed
- I want to make sure that the servers in the network are properly used

How to Control (Actually Assist) the Streaming Clients?

- (Blind) Bandwidth throttling ✗
- Manifest offerings, manipulations and updates ✗
- Event signaling ✗
- HTTP operation (Redirects) ✗
- Control plane and session management ✓

Architecture for SAND

DANE: DASH-assisting network element
PER: Parameters for enhancing reception
PED: Parameters for enhancing delivery



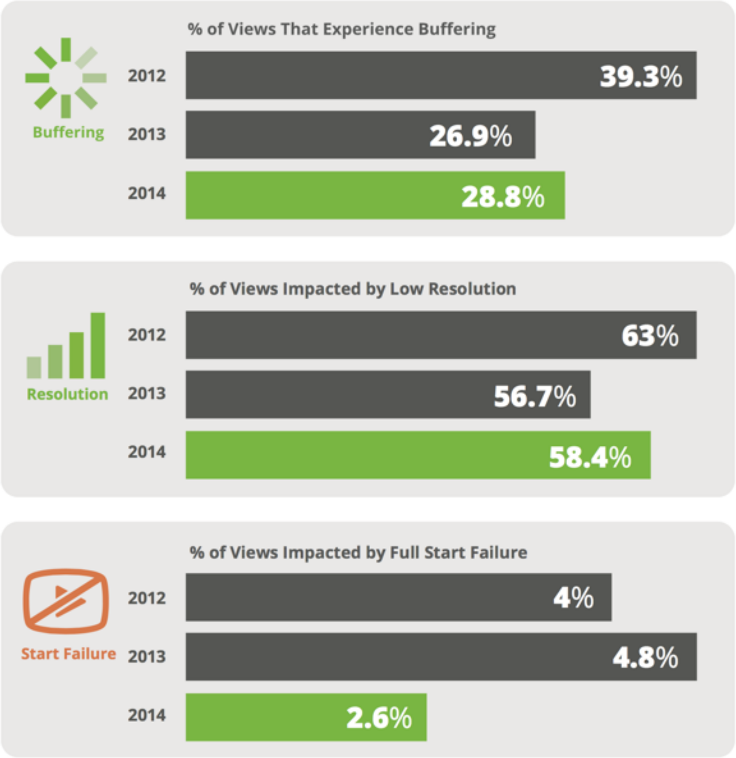
Organizations Working on DASH

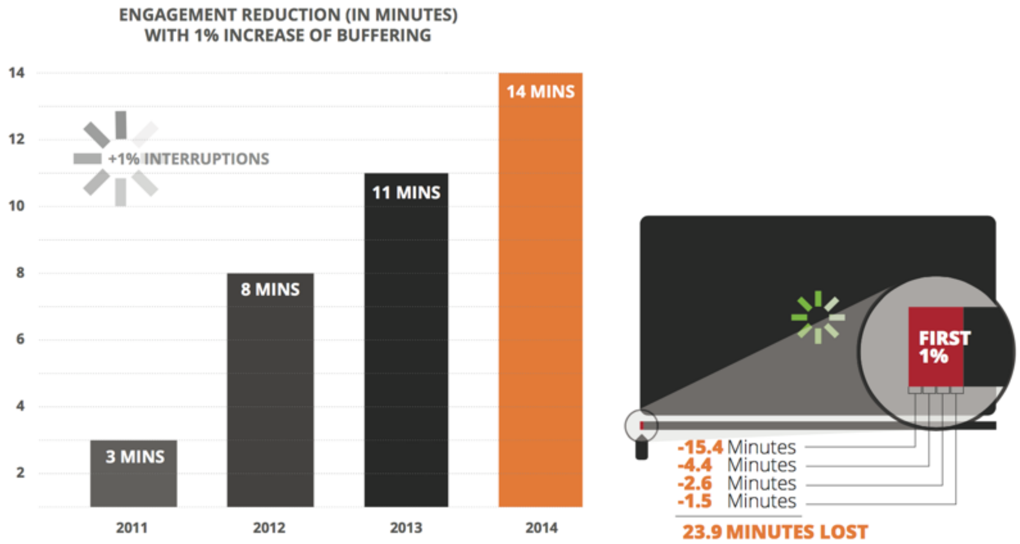
- **MPEG DASH**
  - http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html
  - Mailing List: http://lists.uni-klu.ac.at/mailman/listinfo/dash
- **DASH Industry Forum**
  - http://dashif.org
- **3GPP PSS and DASH**
  - http://ftp.3gpp.org/specs/html-info/26234.htm
  - http://ftp.3gpp.org/specs/html-info/26247.htm
- **DECE – UltraViolet**
  - http://www.uvvu.com/
- **HbbTV (Hybrid Broadcast Broadband TV)**
  - http://www.hbbtv.org/pages/about_hbbtv/specification.php
- **Digital TV Group (DTG)**
  - http://www.dtg.org.uk/publications/books.html
- **Digital Video Broadcasting (DVB)**
  - http://www.dvb.org

Part II: Common Problems in HTTP Adaptive Streaming

- Multi-Client Competition Problem
- Consistent-Quality Streaming
- QoE Optimization
- Inter-Destination Media Synchronization

Some Interesting Stats from Conviva





Source: Conviva Viewer Experience Report, 2015

Part II: Common Problems in HTTP Adaptive Streaming

- Multi-Client Competition Problem
- Consistent-Quality Streaming
- QoE Optimization
- Inter-Destination Media Synchronization

Streaming over HTTP – The Promise

- Leverage tried-and-true Web infrastructure for scaling
  - Video is just ordinary Web content!
- Leverage tried-and-true TCP
  - Congestion avoidance
  - Reliability
  - No special QoS for video

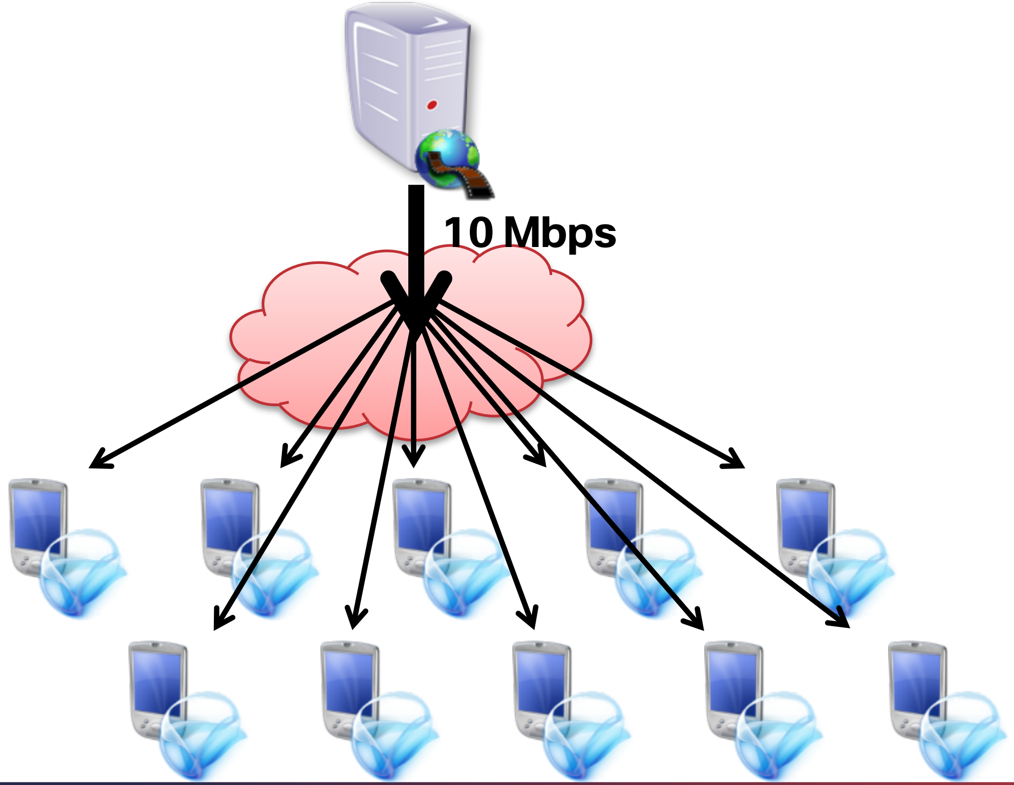
It should all "just work" ☺

Does Streaming over HTTP Scale?

- When streaming clients compete with other traffic, mostly yes
- But when streaming clients compete with each other for bandwidth, we begin to see problems:
  - The clients' adaptation behaviors interact with each other:
    - One client upshifts → Other clients get less bandwidth and may downshift
    - One client downshifts → Other clients get more bandwidth and may upshift
  - The competing clients form an "accidental" distributed control-feedback system
    - Such systems often exhibit unanticipated behaviors
    - A variety of such behaviors can be seen with widely deployed streaming clients
- Unless adaptation mechanisms are carefully designed to work when competing with other clients, unexpected behaviors will result in places like
  - Multiple screens within a household
  - ISP access and aggregation links
  - Small cells in stadiums and malls

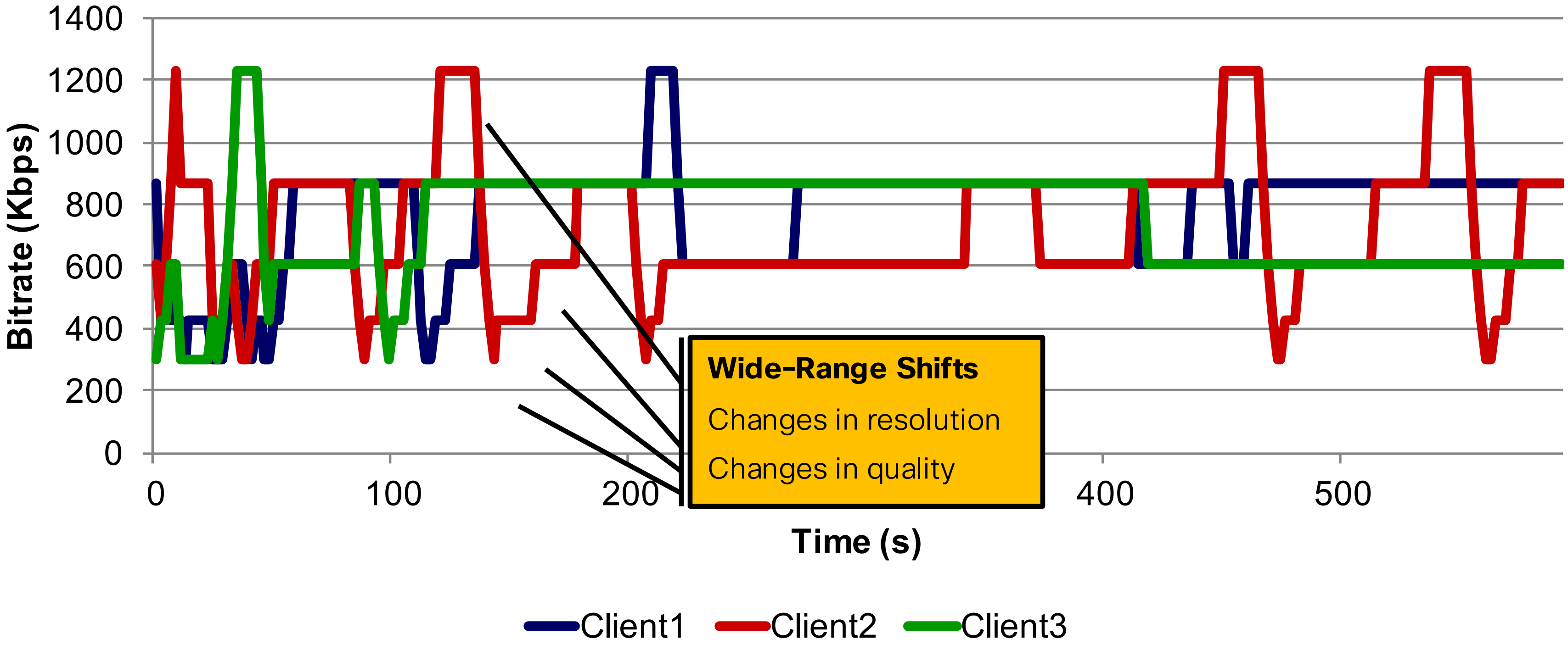
Simple Competition Experiment

10 Microsoft Smooth Clients Sharing 10 Mbps Link



10 Microsoft Smooth Clients Sharing 10 Mbps Link

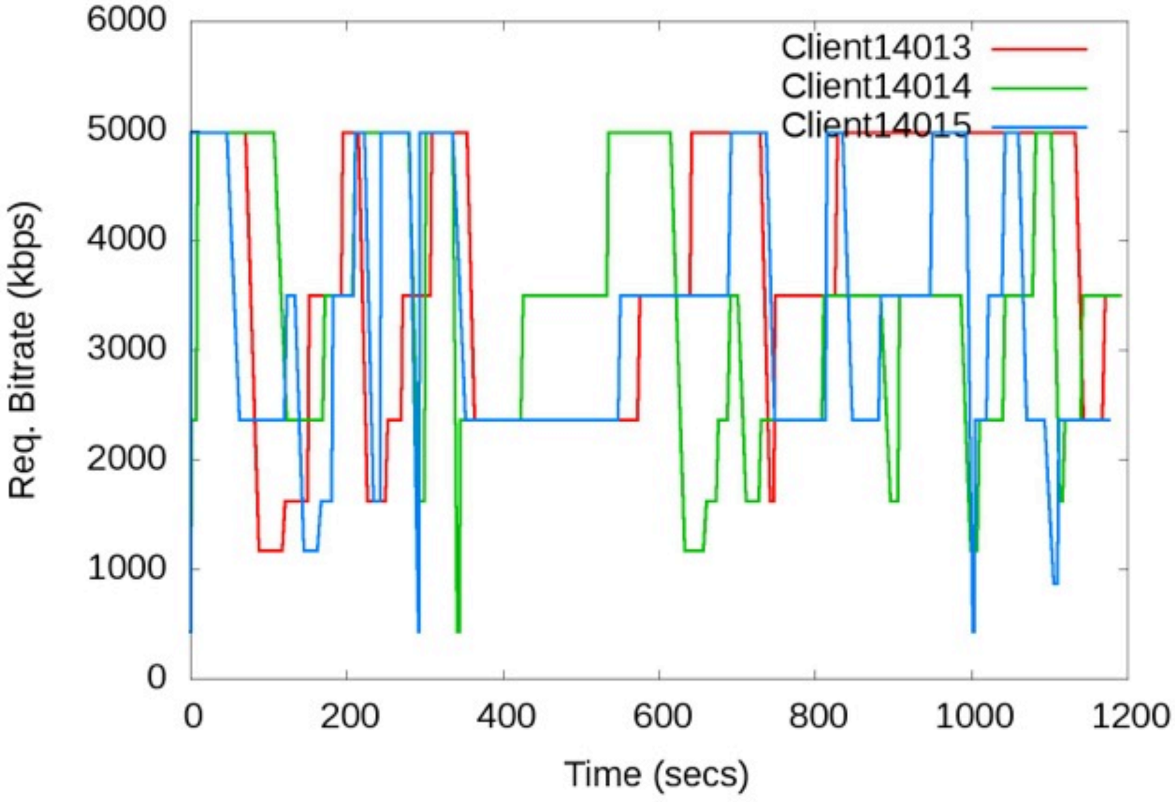
Streaming “Big Buck Bunny” (Three Clients are Shown)



Available Representations: 300, 427, 608, 866, 1233, 1636, and 2436 Kbps

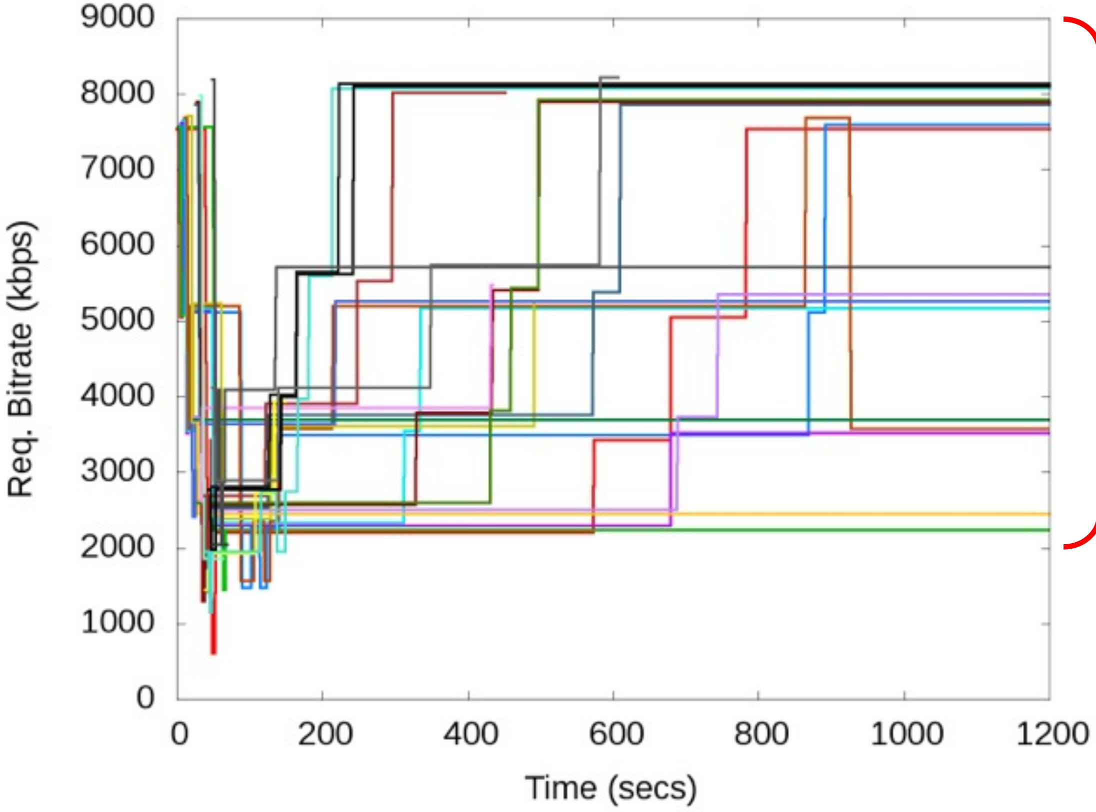
30 Apple Clients (Lion) Sharing 100 Mbps Link

50 ms RTT, Single RED Queue



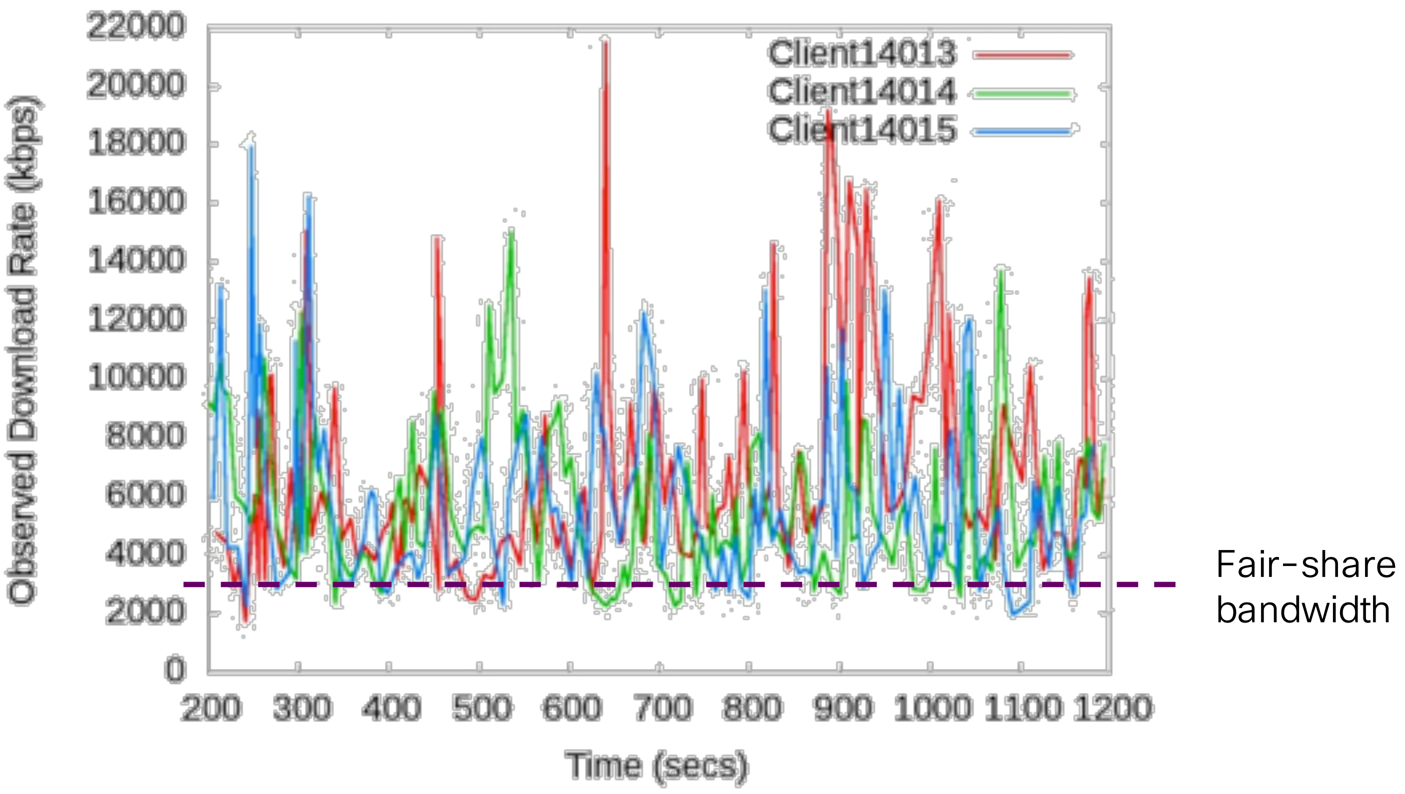
22 Apple Clients (Mavericks) Sharing 100 Mbps Link

50 ms RTT, Single RED Queue



Clients seem to “lock in” after a while; persistent unfairness?

Download Rates Experienced by Individual Clients

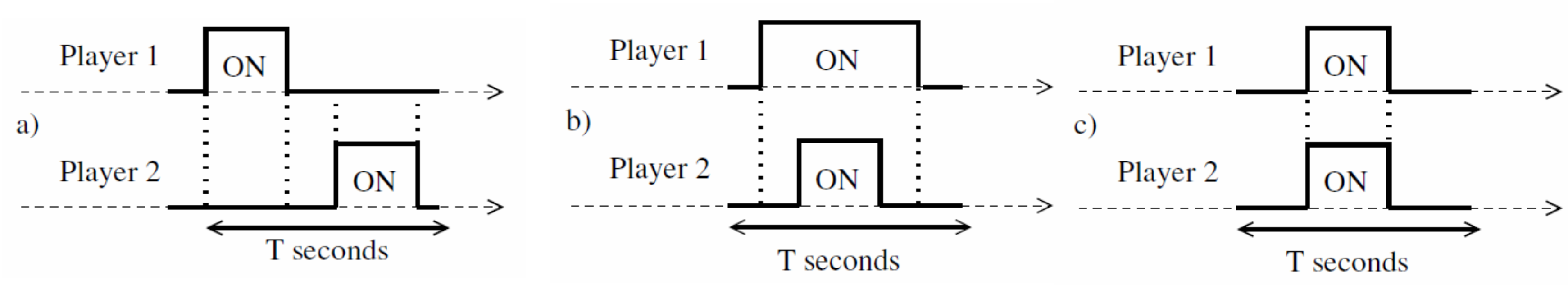


Understanding the Root Cause

Two Competing Clients

- Depending on the timing of the ON periods:
  - Unfairness, underutilization and/or instability may occur
  - Clients may grossly overestimate their fair share of the available bandwidth

Clients cannot figure out how much bandwidth to use until they use too much

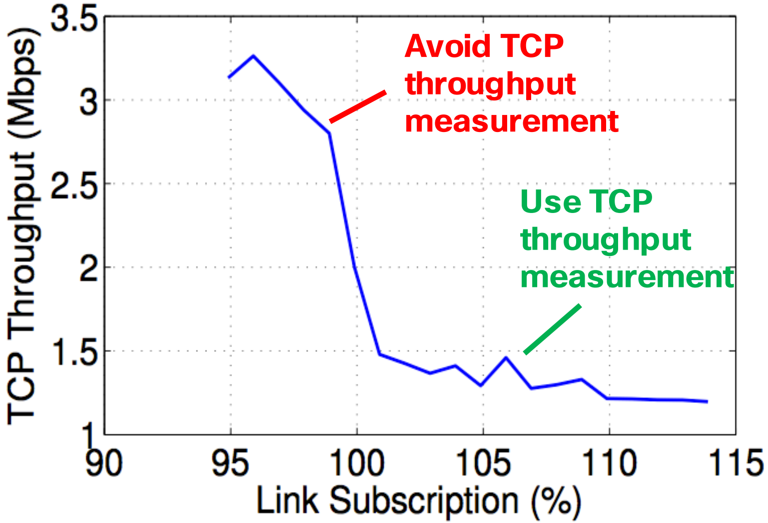


Reading: “What happens when HTTP adaptive streaming players compete for bandwidth?,” ACM NOSSDAV 2012

Client-Side Approaches

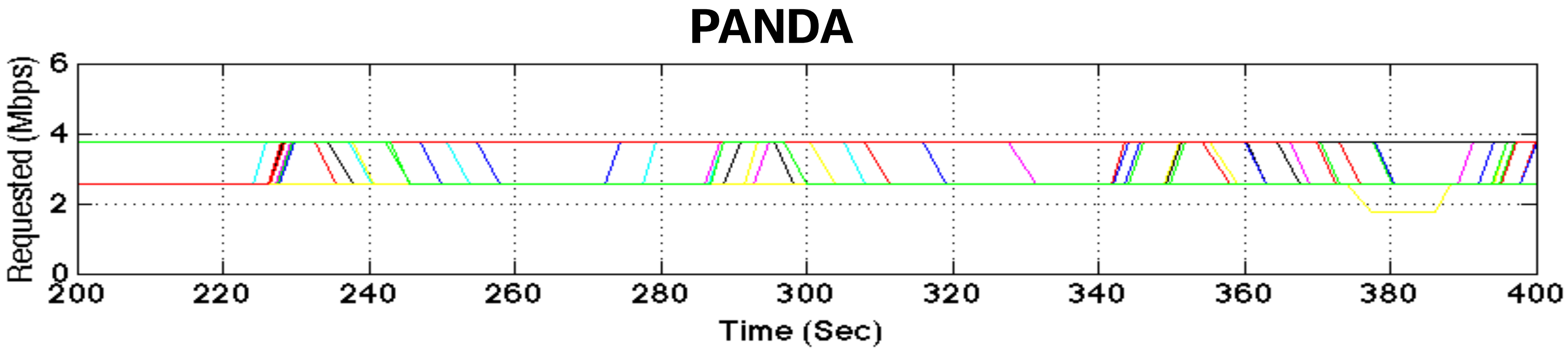
The PANDA Algorithm

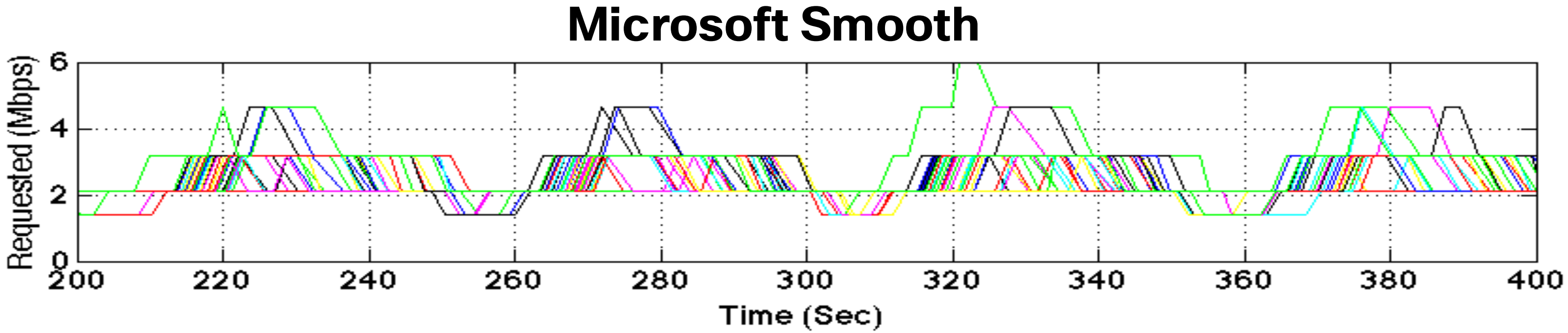
- Avoid the root cause that triggers bitrate oscillation
  - Use the TCP throughput measurement only when the link is over-subscribed
- How to tell when the link is under/over-subscribed?
  - Apply “probing” (i.e., small increments of data rate)
  - Additive-Increase, Multiplicative-Decrease (AIMD) for probing (similar to TCP)
- How to continuously vary the data rate (the video bitrate is discrete)?
  - Fine-tune the inter-request time



Reading: ”Probe and adapt: rate adaptation for HTTP video streaming at scale," IEEE JSAC, Apr. 2014

36 PANDA vs. Smooth Clients Sharing 100 Mbps





PANDA players can effectively stop oscillations!

Network-Based Approaches

Could Network QoS in the Core and Edge Help?

- Idea: Apply QoS to downstream streams to stabilize client rate selections
- Questions:
  - What QoS policy will help?
  - How to recognize which service flows carry adaptive streaming traffic?
  - Can the solution fit within existing platform QoS mechanisms?
  - Can solution work with existing clients?
- We are actively investigating these questions

Control Plane Approaches

Server(s) and Network Providing Assistance to Clients

- Control plane that enables to exchange messages between the client and other elements
  - Control plane typically has 1:1 correspondence and is bi-directional
  - Control plane carries operational data in both directions
  - Control plane is independent from the media/manifest distribution

Part II: Common Problems in HTTP Adaptive Streaming

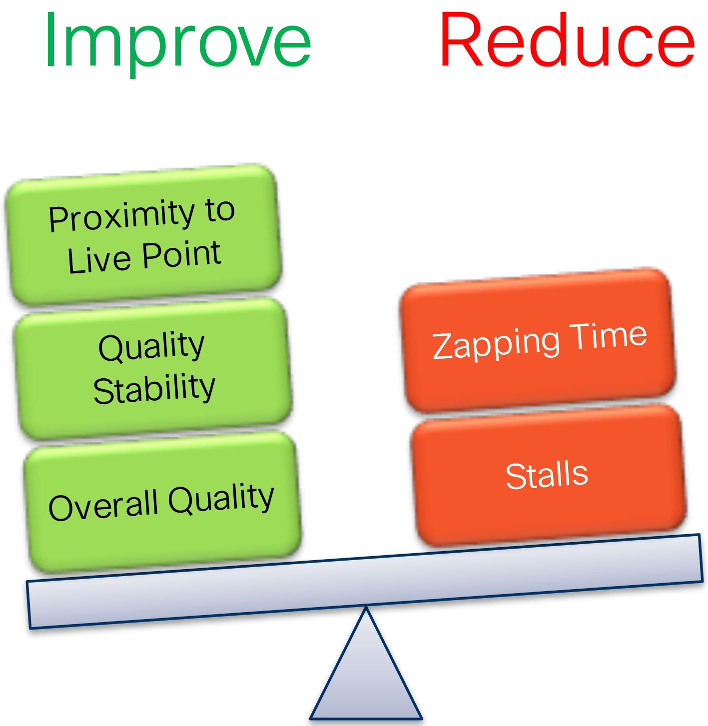
- Multi-Client Competition Problem
- Consistent-Quality Streaming
- QoE Optimization and Measurement
- Inter-Destination Media Synchronization

Nature of the Video Content

- Using constant-bitrate (CBR) encoding scheme has pros and cons:
  - ✓ Encoding is easier and cheaper
  - ✓ Client adaptation algorithms are simpler
  - ✗ Viewer experience deteriorates between low-motion/complexity vs. high-motion/complexity scenes

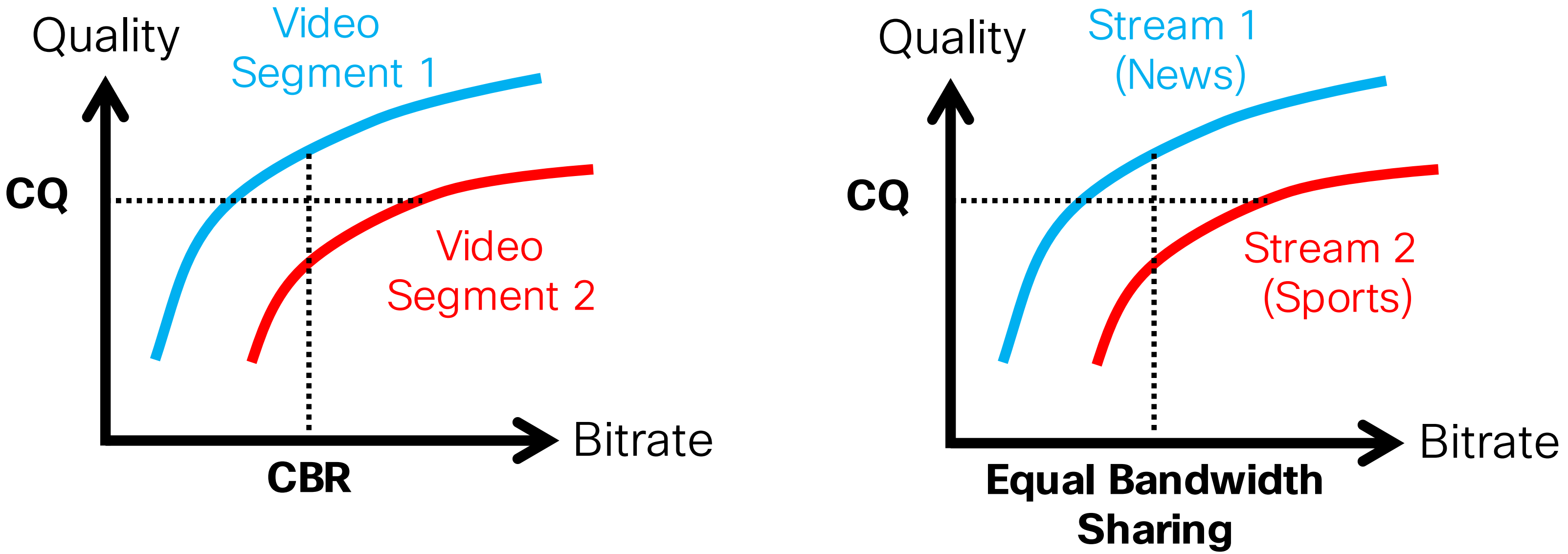
- Allocate bits among segments intelligently to yield an optimal overall quality

Tradeoffs in Adaptive Streaming

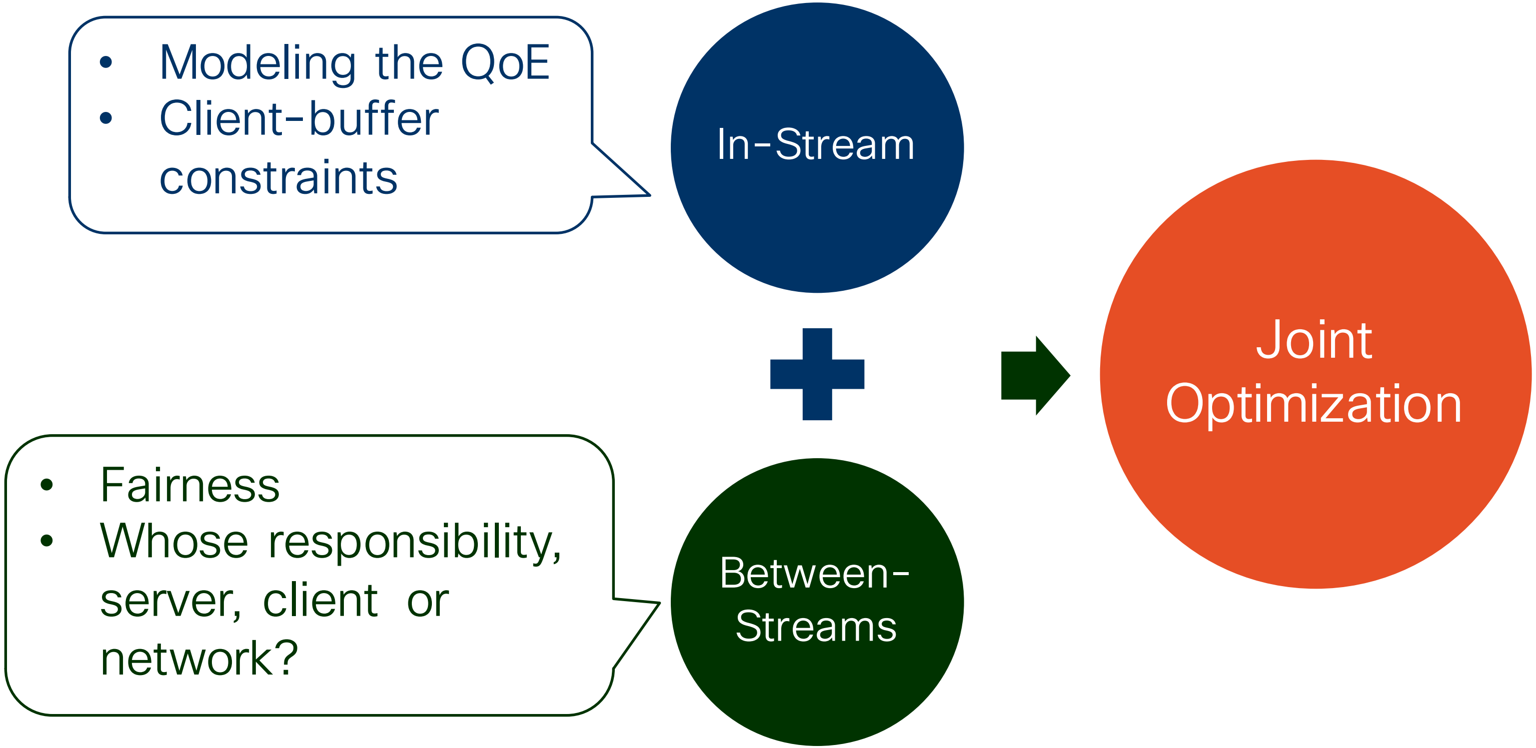


Dimensions – In-Stream vs. Between-Streams

- Same principle applies to both:
  - In-stream Case: Temporal bit shifting between segments
  - Between-streams Case: Bit shifting between streams sharing a bottleneck link



Scope of Optimization



In-Stream Bitrate Allocation: Challenges

- Challenge 1: How to measure video quality?
  - How to measure the quality of each segment?
  - Temporal pooling – How a viewer forms an overall impression over a sequence of segments?
- Challenge 2: We must meet client-buffer constraints
  - We must not drain the buffer
  - We must maintain buffer below an upper bound, too
- Challenge 3: Optimization is myopic
  - Client does not know available bandwidth in the future
  - Only a finite horizon of video information might be available

Video Quality – A Generic Framework

- Quality score for a segment: PSNR, -MSE, SSIM, JND, …
- Temporal Pooling: Possible objective functions
  - Max-Sum: Maximize the sum of (or average) quality over segments
  - Max-Min: Maximize the worst-case quality over segments
- Temporal pooling using α–fairness utility function [Srikant’04]

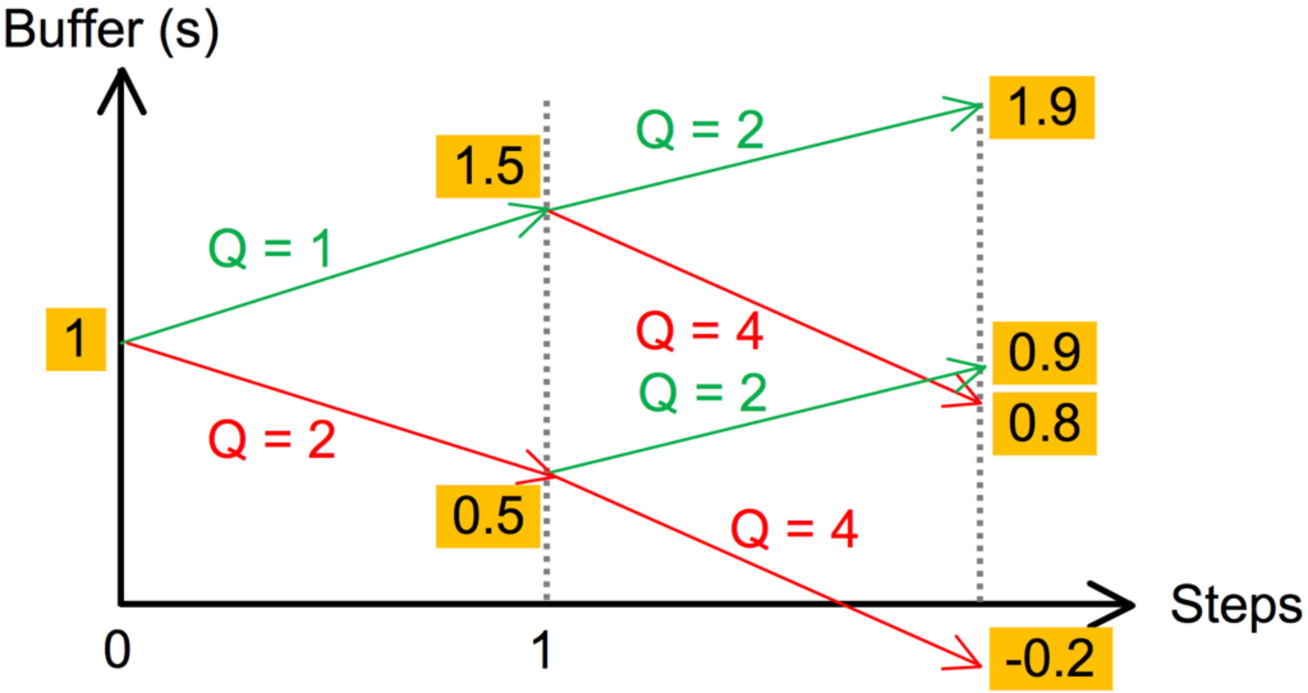
$$\max \sum_n U_\alpha(Q(n)), \text{ where } U_\alpha(q) := \frac{q^{1-\alpha}}{1-\alpha}$$

- Special cases
  - Max-Sum (α=0)
  - Max-Min (α=∞)
  - Proportional fairness (α=1)

Bandwidth and Video Bitrate Variability

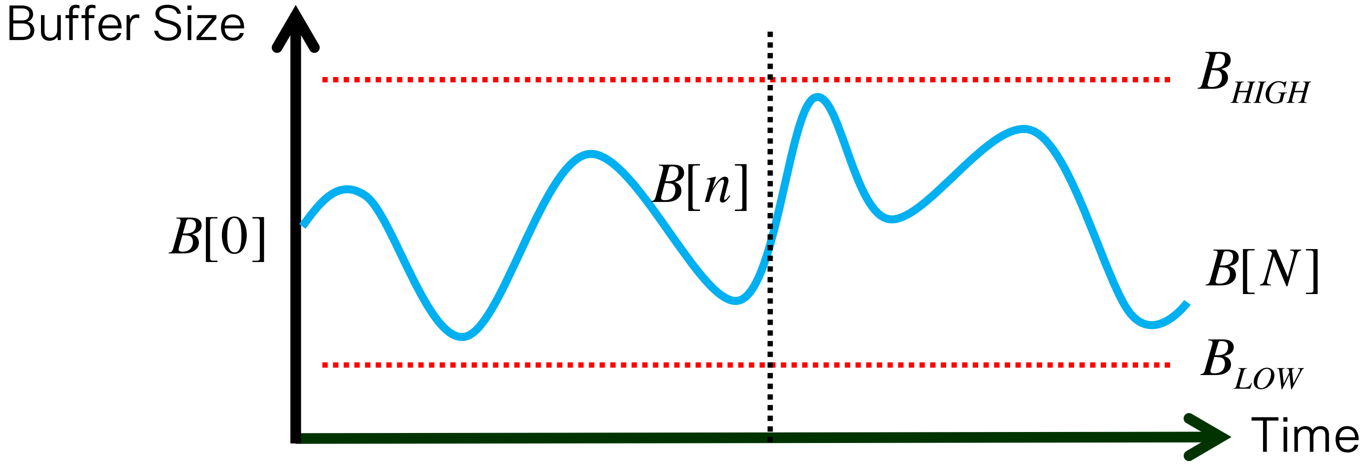
- Quality optimization poses higher risk of buffer underrun/overshoot than conventional streaming
- We need to
  - Impose lower and upper bounds on buffer evolution
  - Have a fast algorithm to detect bandwidth drops
  - Have proper balance between these two
- Proposed Solution
  - Use a fast algorithm (e.g., PANDA) to quickly detect bandwidth changes
  - Apply an online algorithm to adapt to network bandwidth step by step
  - Use dynamic programming (DP) to program buffer evolution within a sliding window

A Toy Example



Dynamic Programming Solution

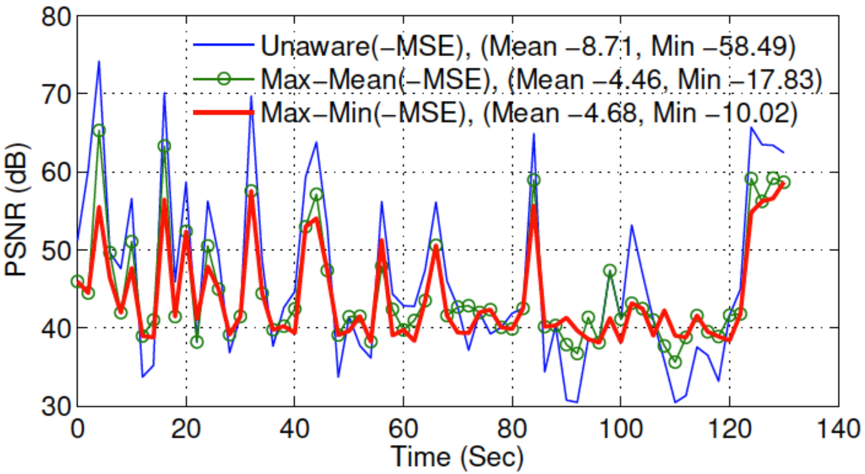
- Brute-force search has exponential complexity
  → Dynamic programming reduces processing time to polynomial time

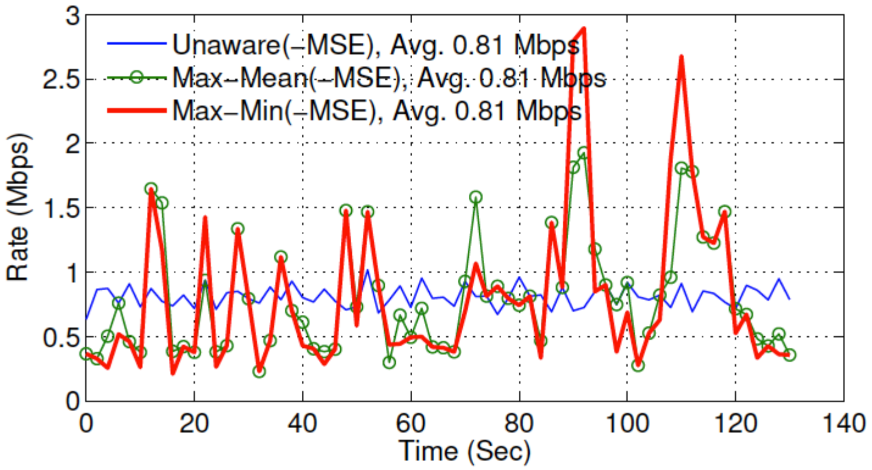


$$Q^*(B[0] \rightarrow B[N]) = \max_{B[n]} \{ Q^*(B[0] \rightarrow B[n]) + Q^*(B[n] \rightarrow B[N]) \}$$

Simulation Results – Elysium

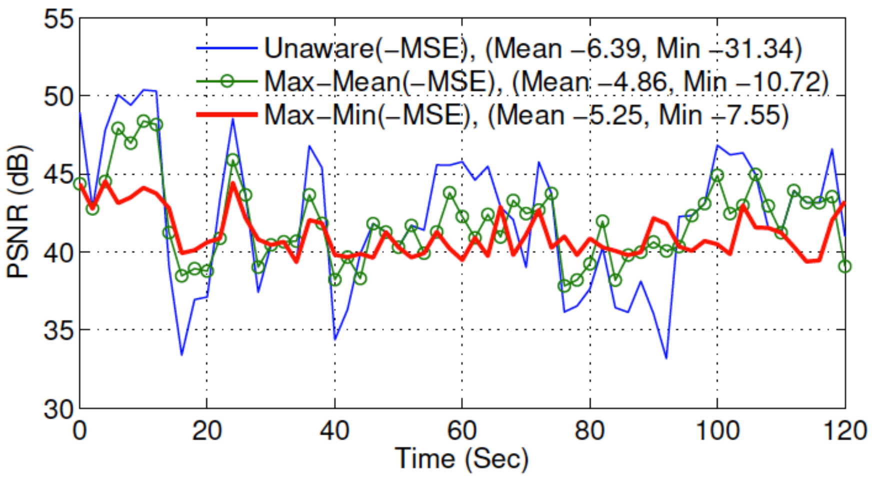
Quality-Unaware vs. Mean Quality Optimized vs. Minimum Quality Optimized

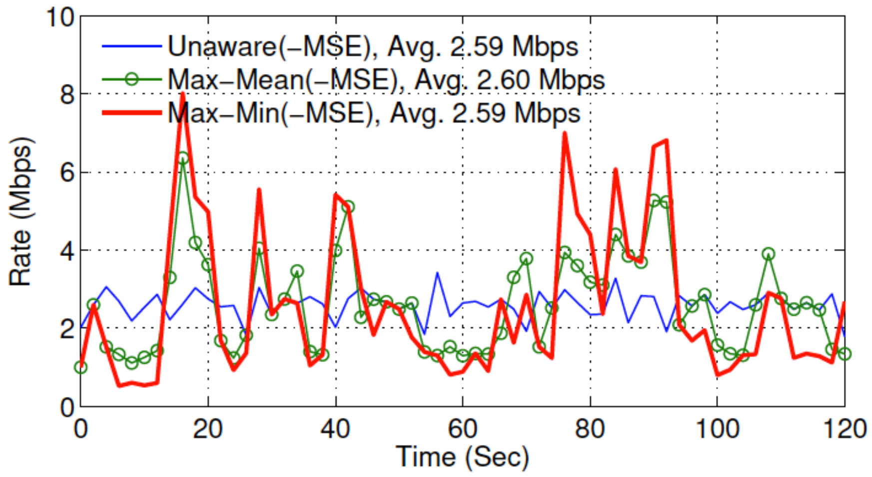




Simulation Results – Avatar

Quality-Unaware vs. Mean Quality Optimized vs. Minimum Quality Optimized





Demo

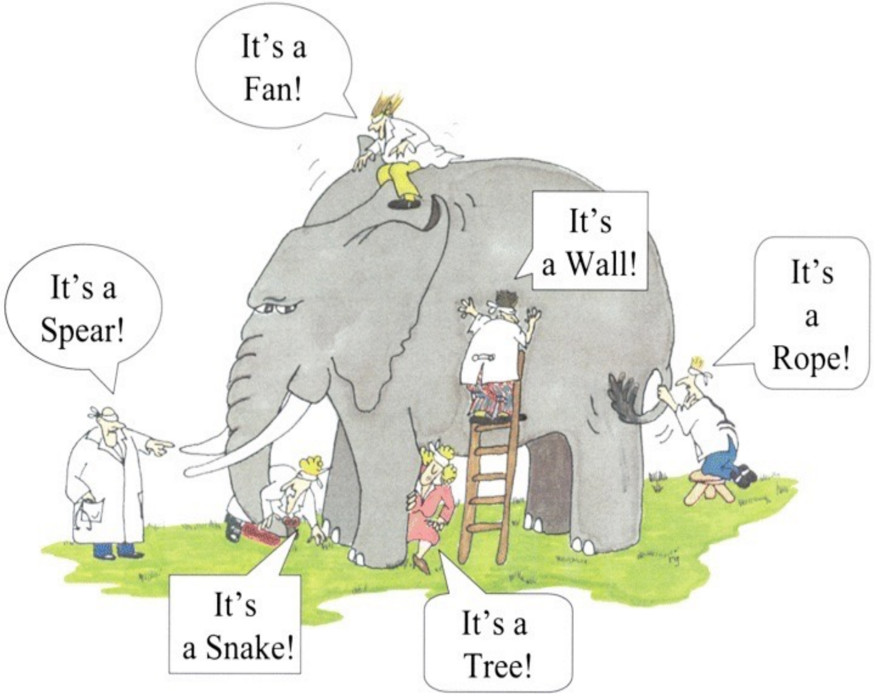
- Sample 1: CBR encoded, quality-unaware streaming at 800 Kbps
- Sample 2: VBR encoded, quality-unaware streaming at 800 Kbps
- Sample 3: VBR encoded, consistent-quality streaming at 800 kbps
- Also available at https://sites.google.com/site/cqhttpstreaming

Part II: Common Problems in HTTP Adaptive Streaming

- Multi-Client Competition Problem
- Consistent-Quality Streaming
- QoE Optimization and Measurement
- Inter-Destination Media Synchronization

What is Quality?

Many definitions but in general, it's like an elephant

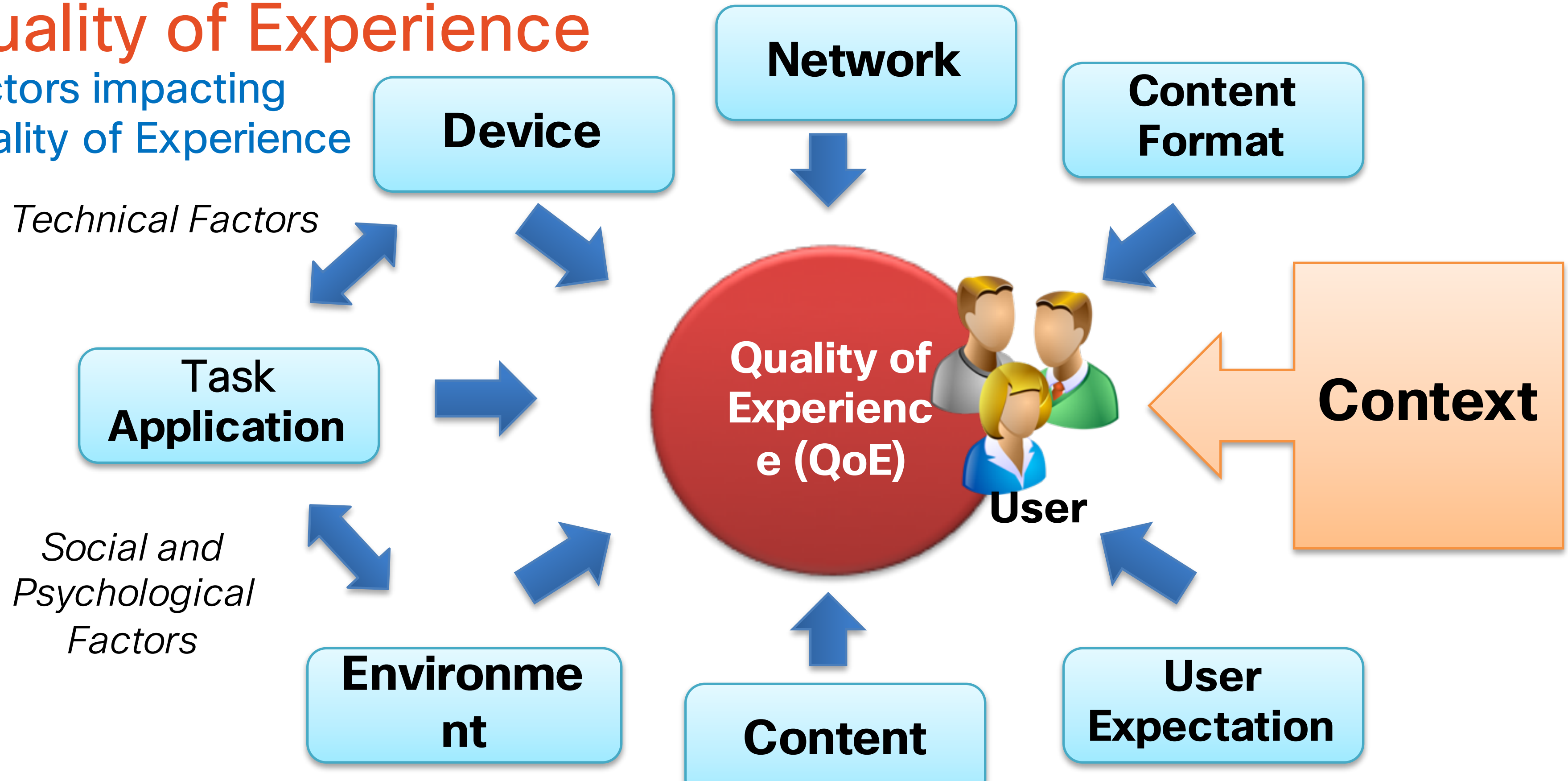


The blind men and the elephant, Poem by John Godfrey Saxe

➜ see also F. Pereira, "On Quality of Multimedia Experiences", QUALINET Final Workshop, Delft, The Netherlands, Oct. 2014.

Quality of Experience

Factors impacting Quality of Experience



T. Ebrahimi, “Quality of Multimedia Experience: Past, Present and Future”, Keynote at ACM Multimedia 2009, Beijing, China, Oct 22, 2010. http://www.slideshare.net/touradj_ebrahimi/qoe

Quality of Experience

http://www.qualinet.eu/

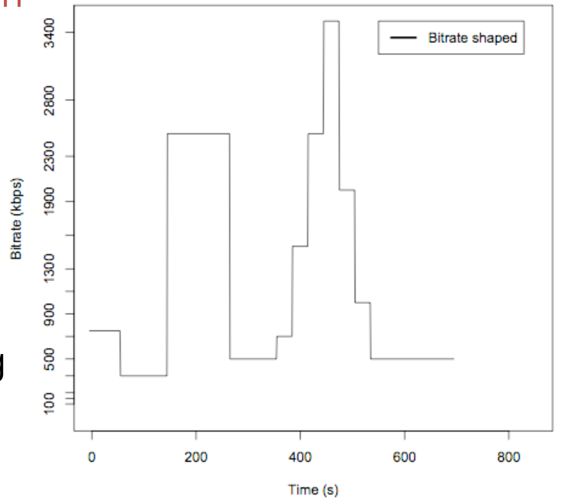
- **Quality of Experience (QoE)**
  - is the **degree of delight or annoyance** of the user (persona) of an application or service
  - results from the **fulfillment of his or her expectations** with respect to the **utility and/or enjoyment of the application or service** in the light of the **user's personality and current state** (context)
- Different application domains have different QoE requirements
  - Need to provide specializations of the general QoE definition
  - Take into account requirements formulated by means of influence factors and features of QoE
  - Domain-/service-/application-specific QoE definition
- Need to identify HAS/ABR/DASH QoE influence factors

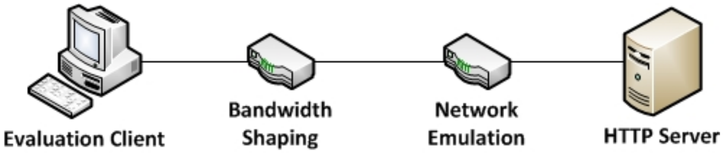
How to evaluate DASH?

bitcodin
http://www.bitcodin.com

bitdash
http://www.dash-player.com

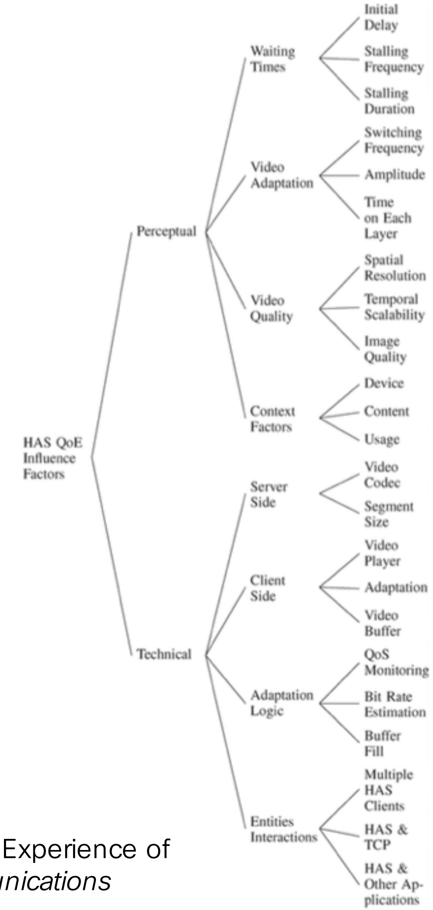
- Test sequence
  - Dataset, tools (see backup slides for details)
  - Adopted Tears of Steel / Big Buck Bunny & DASHed it with bitcodin
- Players
  - Proprietary solutions (smooth, HLS, HDS)
  - YouTube, dash.js, DASH-JS
  - bitdash
  - ...and compare it with ten different adaptation algorithms
- Objective evaluation
  - Common test setup using network emulation & bandwidth shaping
  - Predefined bandwidth trajectory (or real network traces)
- Subjective evaluation
  - Lab [ITU-T B.500 / P.910] vs. crowdsourcing with special platforms or social networks





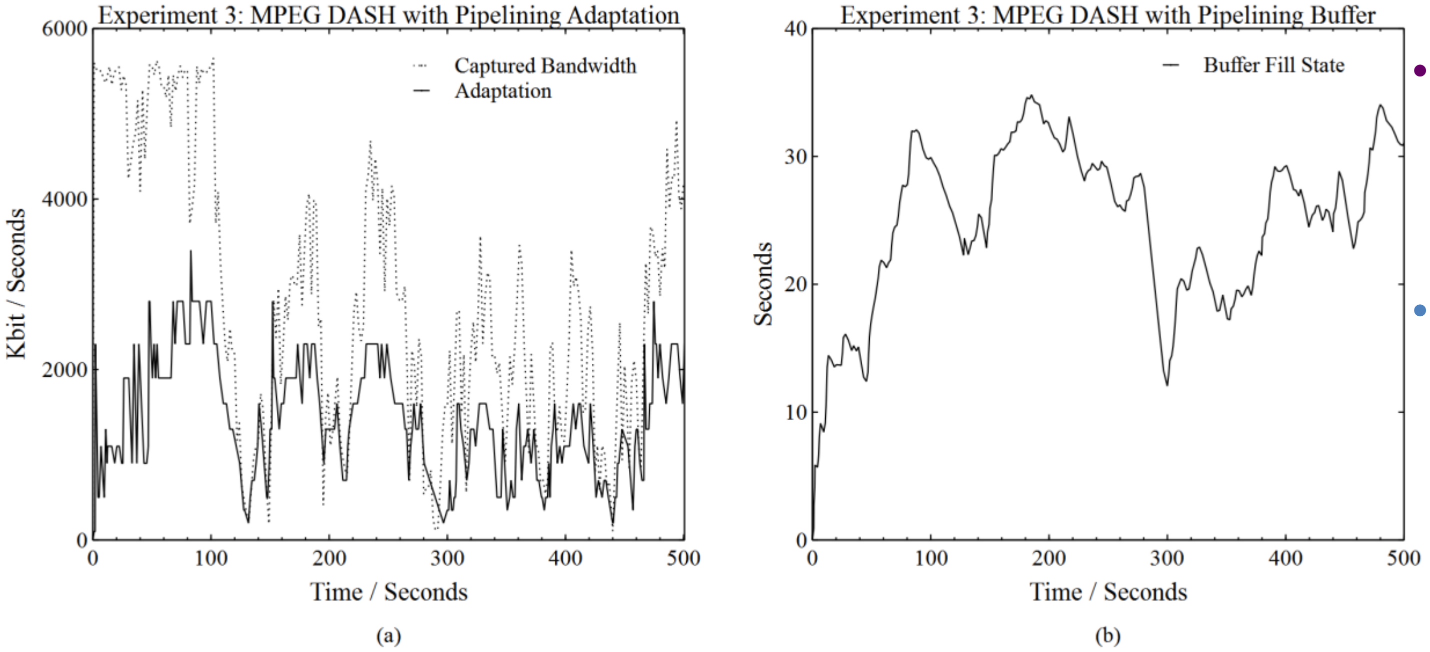
Quality of Experience for DASH

- Objective
  - Initial or startup delay (low)
  - Buffer underrun / stalls (zero)
  - Quality switches (low)
  - Media throughput (high)
  - [Other media-related configuration: encoding, representations, segment length, …]
- Subjective
  - Mean Opinion Score (MOS) – various scales
  - Various methodologies (e.g., DSCQS, DSIS, ACR, PC, …)



M. Seufert, et al., "A Survey on Quality of Experience of HTTP Adaptive Streaming," . *IEEE Communications Surveys & Tutorials*, vol. 2014 (2014). doi:10.1109/COMST.2014.2360940

DASH VLC Implementation



- Real-world network traces captured while driving on the highway
- Mobile environment considered very challenging for DASH services

- Simple throughput-based adaptation logic
- Non stepwise switching
- Good average bitrate and stable buffer

Summary of the Results

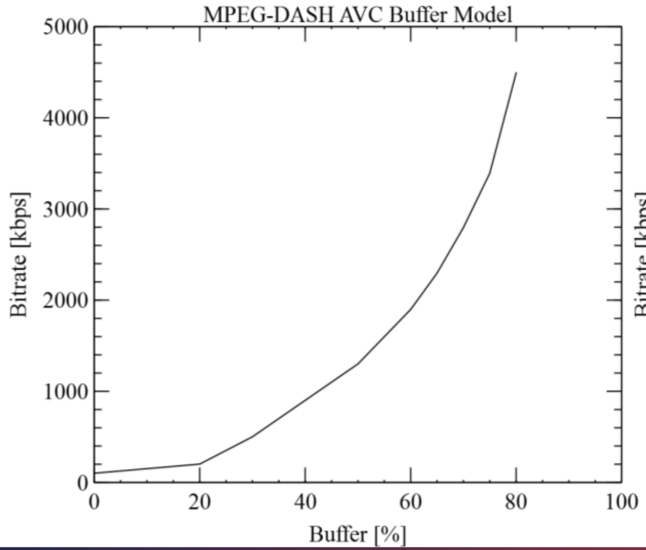
Name	Average Bitrate [kbps]	Average Switches [Number of Switches]	Average Unsmoothness [Seconds]
Microsoft	1522	51	0
Adobe	1239	97	64
Apple	1162	7	0
DASH VLC	1045	141	0
DASH VLC Pipelined	1464	166	0

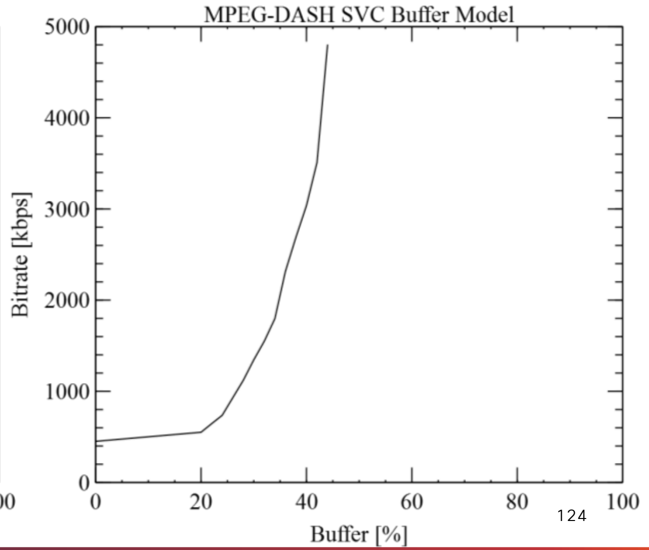
- Similar results for Web-based DASH player (DASH-JS)

C. Mueller, S. Lederer, C. Timmerer, "An Evaluation of Dynamic Adaptive Streaming over HTTP in Vehicular Environments", *In Proceedings of the Fourth Annual* ***ACM SIGMM Workshop on Mobile Video (MoVid12)***, Chapel Hill, North Carolina, February 2012.

Improving the Adaptation Logic

- Adaptation based on the buffer model with exponential characteristic
  - to reduce the number of quality switches
  - to enable a smooth playback
- SVC model more aggressive due to layered coding scheme
- Different characteristics
  - Exponential
  - Logarithmic
  - Linear

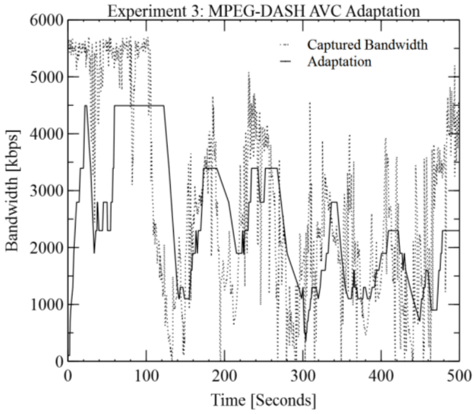




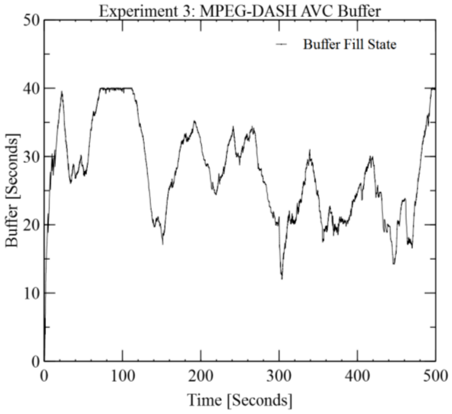
DASH AVC vs. SVC

- AVC – smooth playback
- Increased throughput compared to prev. implementations
- Stable adaptation process and buffer

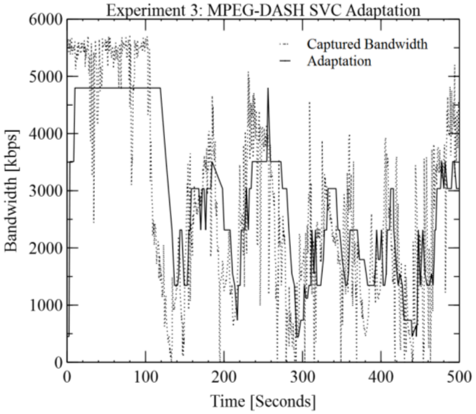
- SVC – better bandwidth utilization than AVC
- Accurate reaction to bandwidth changes
- Still stable buffer



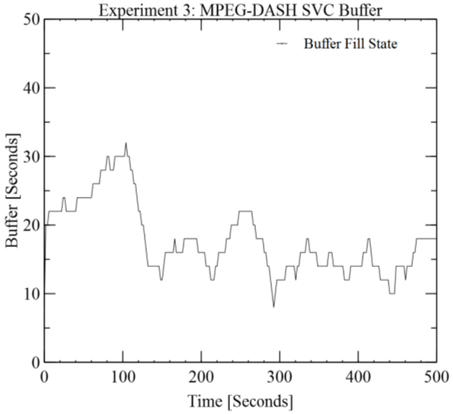
(a)



(b)



(a)



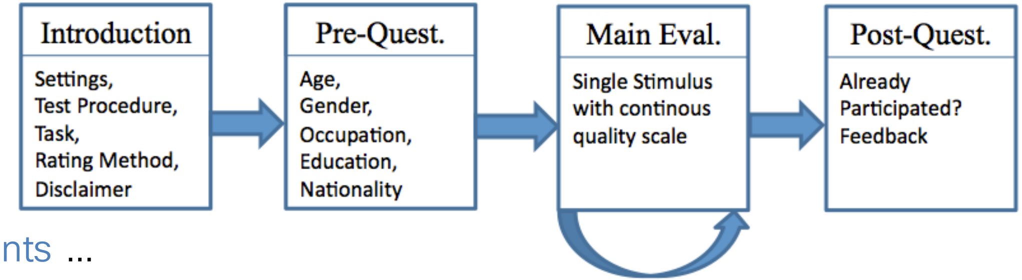
(b)

Summary of the Results

Name	Average Bitrate [kbps]	Average Switches [Number of Switches]	Average Unsmoothness [Seconds]
Microsoft	1522	51	0
Adobe	1239	97	64
Apple	1162	7	0
DASH VLC	1045	141	0
DASH VLC Pipelined	1464	166	0
bitdash-AVC	**2341**	**81**	**0**
bitdash-SVC	**2738**	**101**	**0**

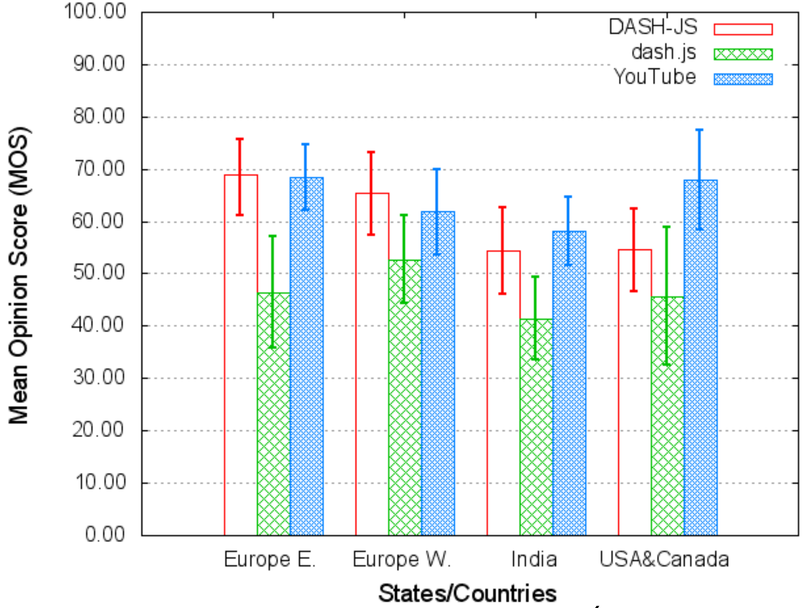
C. Mueller, D. Renzi, S. Lederer, S. Battista, C. Timmerer, "Using Scalable Video Coding for Dynamic Adaptive Streaming over HTTP in Mobile Environments", *In Proceedings of the 20th* ***European Signal Processing Conference (EUSIPCO12),*** Bucharest, Romania, August 2012.

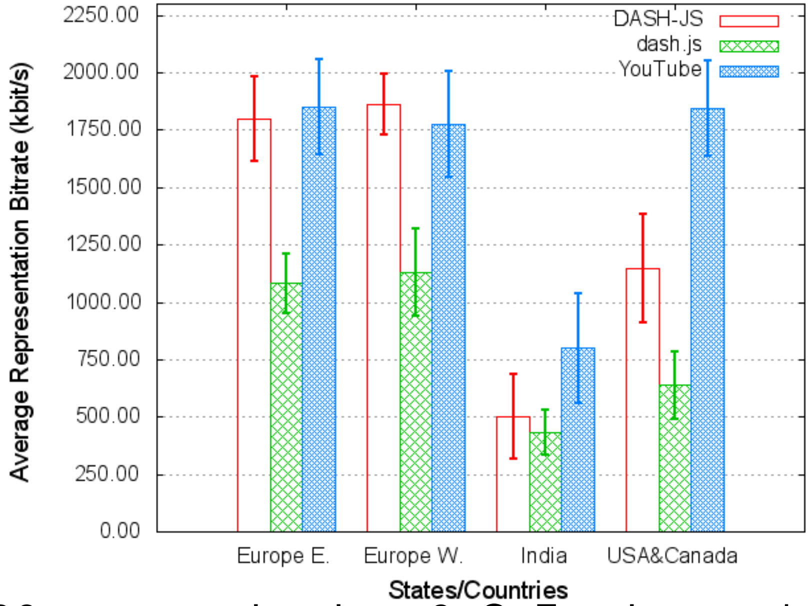
Crowdsourced QoE Evaluation



- Quality of Experience ...
  - Mean Opinion Score [0..100]
  - [other objective metrics:
    start-up time, throughput, number of stalls]
- ... Web-based Adaptive HTTP Streaming Clients ...
  - HTML5+MSE: DASH-JS (dash.itec.aau.at), dash.js (DASH-IF, v1.1.2), YouTube
- ... Real-World Environments ...
  - DASH-JS, dash.js hosted at ITEC/AAU (~ 10Gbit/s)
  - YouTube hosted at Google data centers
  - Content: Tears of Steel @ 144p (250 kbit/s), 240p (380 kbit/s), 360p (740 kbit/s), 480p (1308 kbit/s), and 720p (2300 kbit/s); segment size: 2s
  - Users access content over the open Internet (i.e., real-world environment)
- ... Crowdsourcing
  - Campaign at Microworker platform (others also possible: Mechanical Turk, social networks) limited to Europe, USA/Canada, India
  - Screening Techniques: Browser fingerprinting, stimulus presentation time, QoE ratings and pre-questionnaire

Mean Opinion Score vs. Average Bitrate

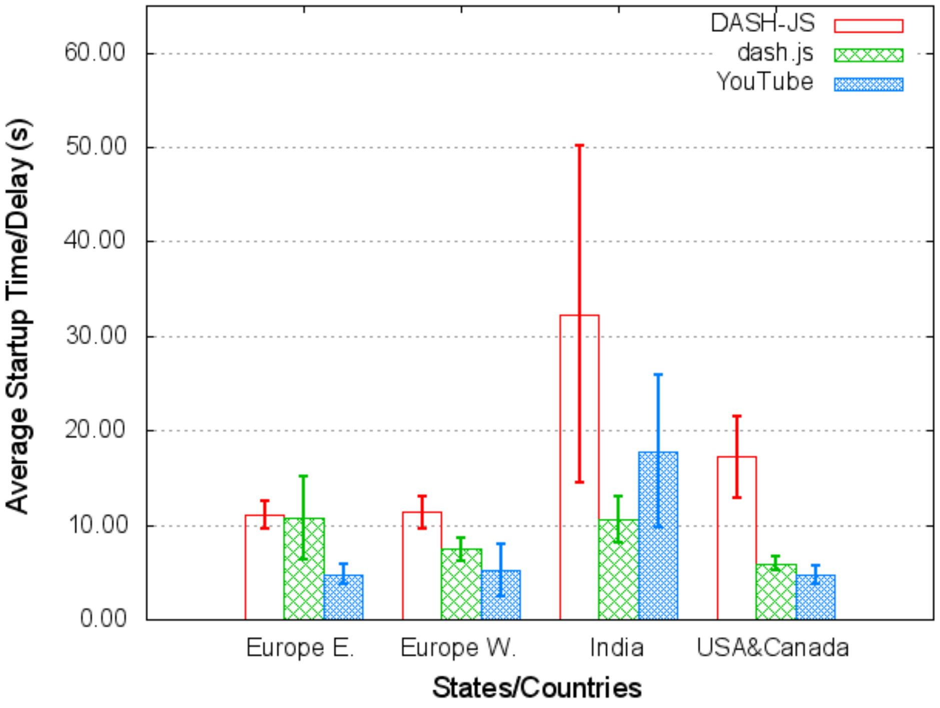


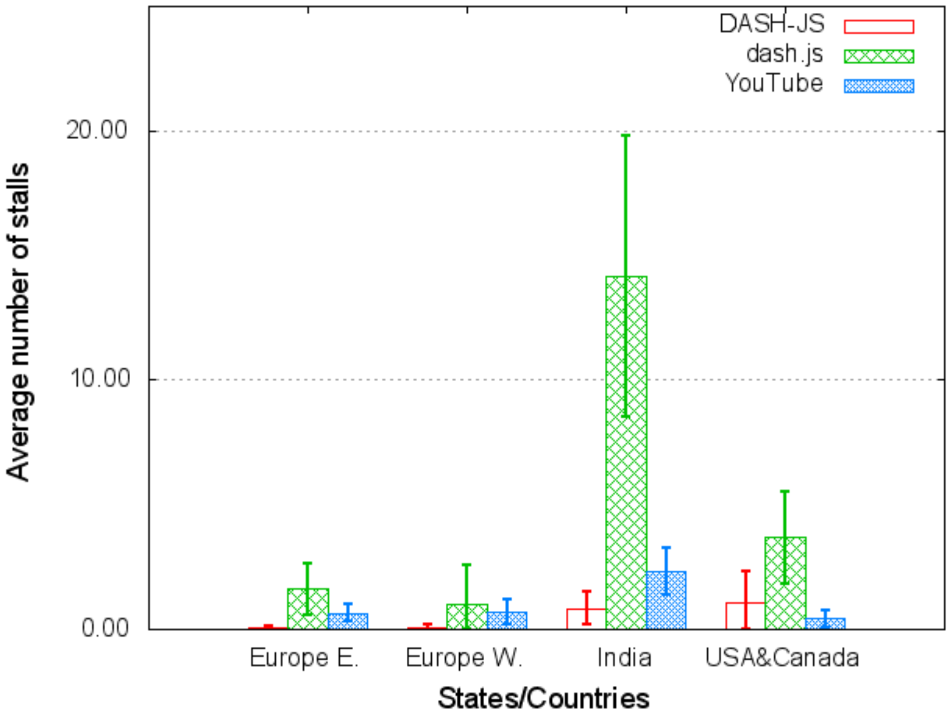


288 microworkers, 33 screened (Fingerprinting: 20, presentation time: 6, QoE ratings and pre- questionnaire: 7), 175 male and 80 female, majority (80%) is aged between 18 and 37

B. Rainer, C. Timmerer, "Quality of Experience of Web-based Adaptive HTTP Streaming Clients in Real-World Environments using Crowdsourcing", *Proceedings of International Workshop on VideoNext: Design, Quality and Deployment of Adaptive Video Streaming*, Sydney, Australia, Dec. 2014.

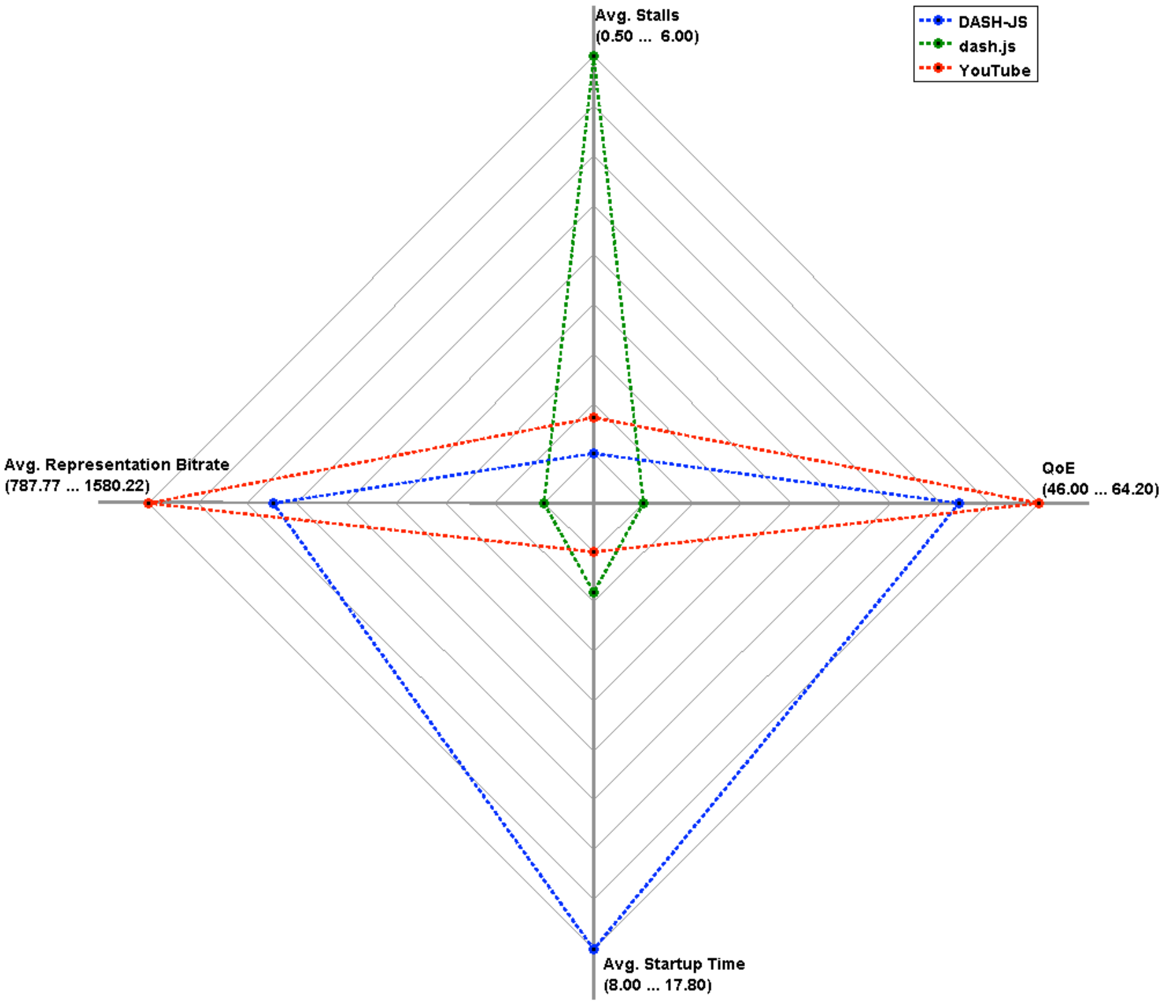
Average Startup Times – Stalls





Results Summary

- DASH-JS
  - High start-up time
  - Low number of stalls
  - Good throughput, QoE
- dash.js
  - Low start-up time
  - High # stalls
  - Low throughput
  - Low QoE
- YouTube
  - Low start-up time
  - Low number of stalls
  - Best throughput, QoE



Now,10 different adaptation logics ..

- Adaptation logics well-known in research literature
- Predefined bandwidth trajectory and test setup
- Different segment sizes, RTTs, HTTP/2, etc.

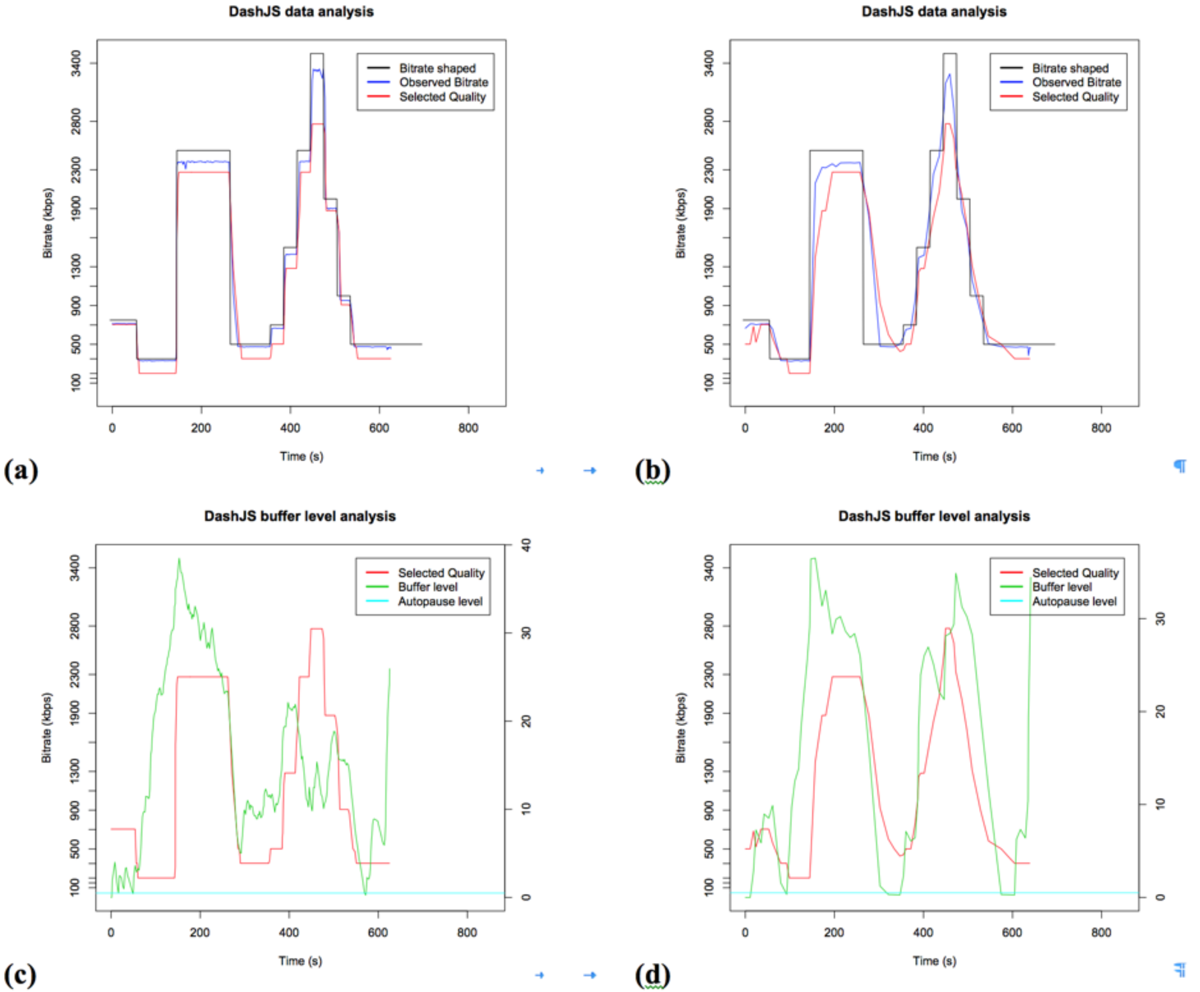


Figure 1. Bandwidth Adaptation for DASH-JS with 2s and 10s segment length: (a) adaptation logic 2s, (b) adaptation logic 10s, (c) buffer level 2s, (d) buffer level 10s. The buffer file state is provided in seconds on the right side.

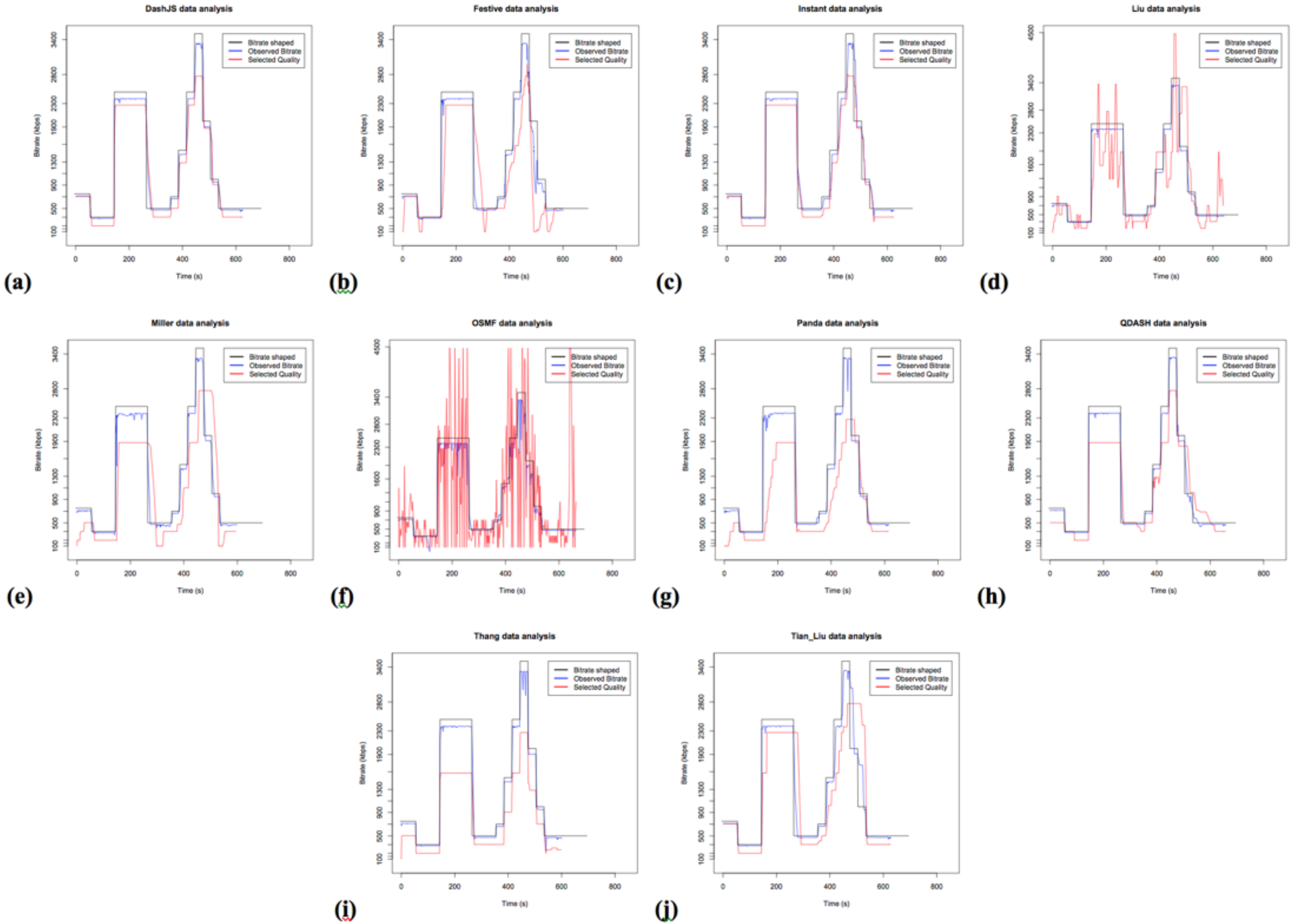


Figure 4. Bandwidth Adaptation with 2s segment size for (a) DASH-JS, (b) FESTIVE, (c) Instant, (d) Liu, (e) Miller, (f) OSMF, (g) PANDA, (h) QDASH, (i) Thang, and (j) Tian-Liu.

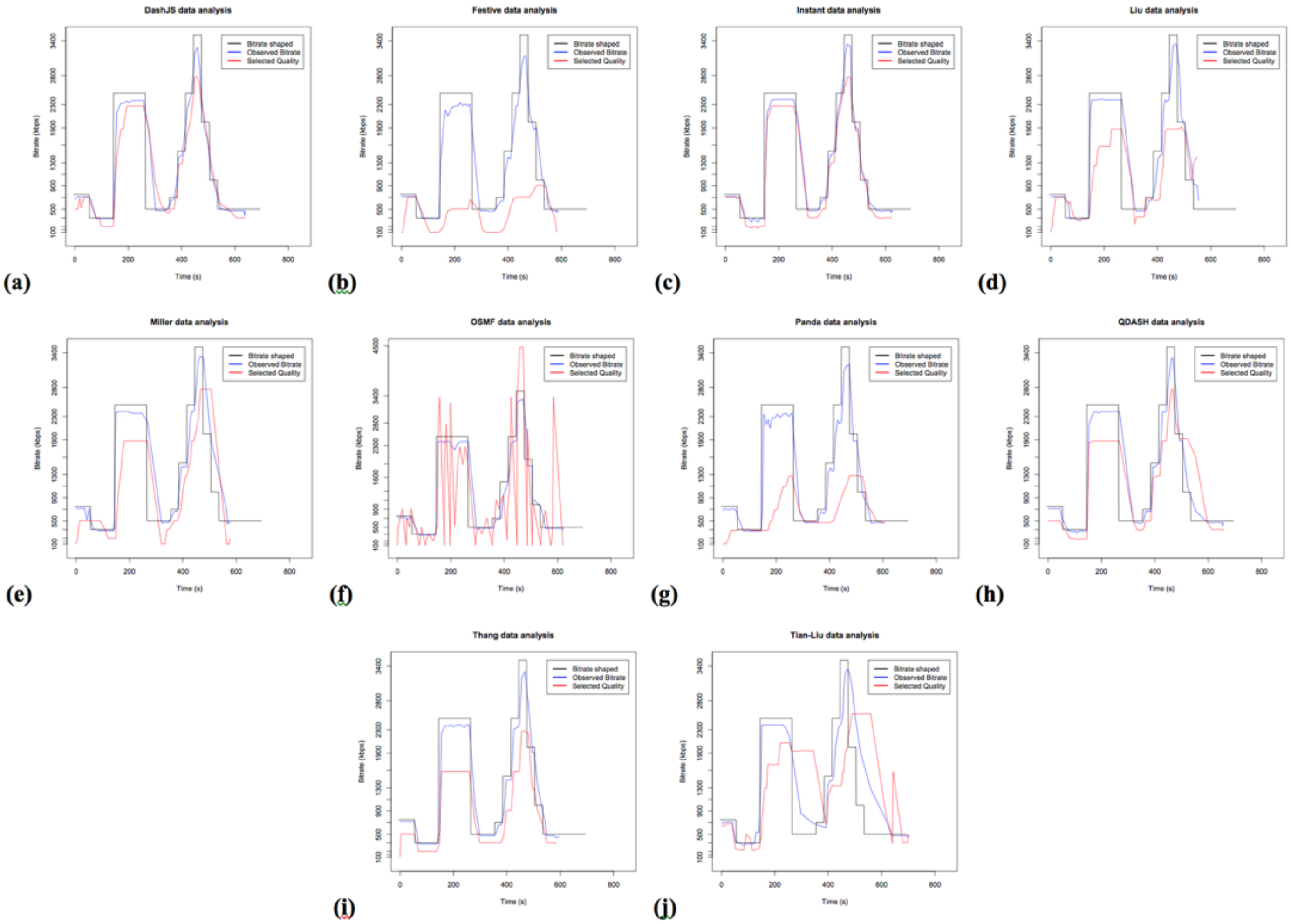
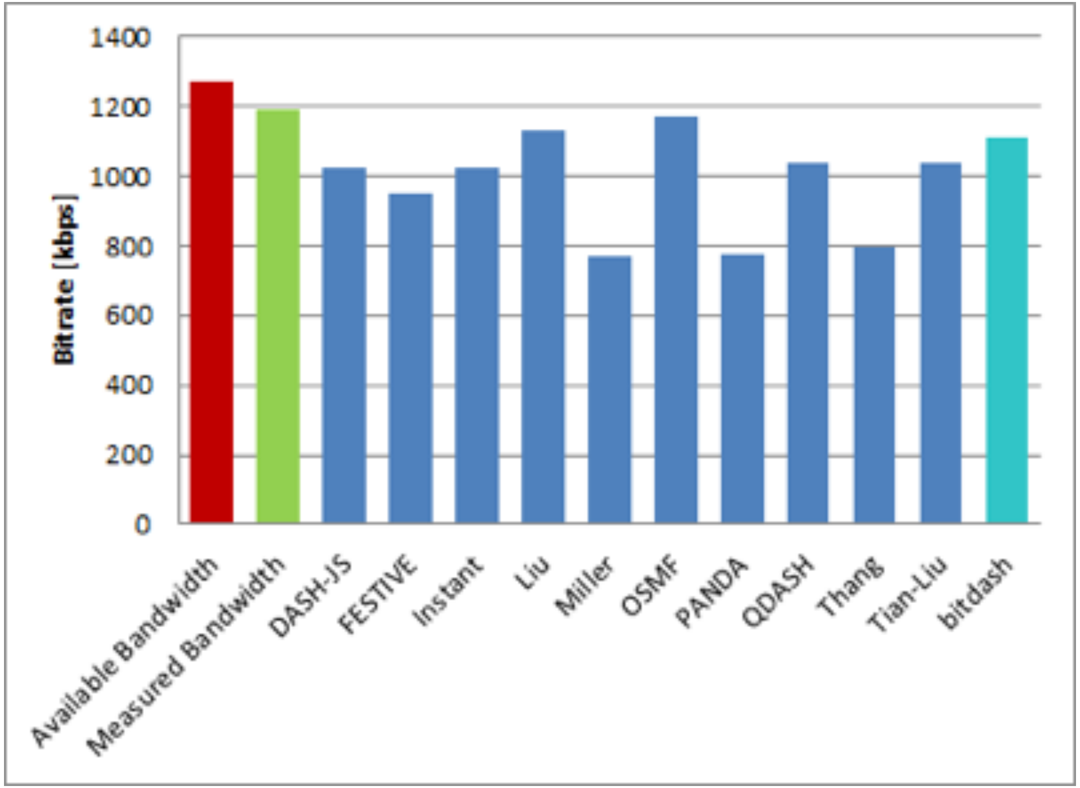
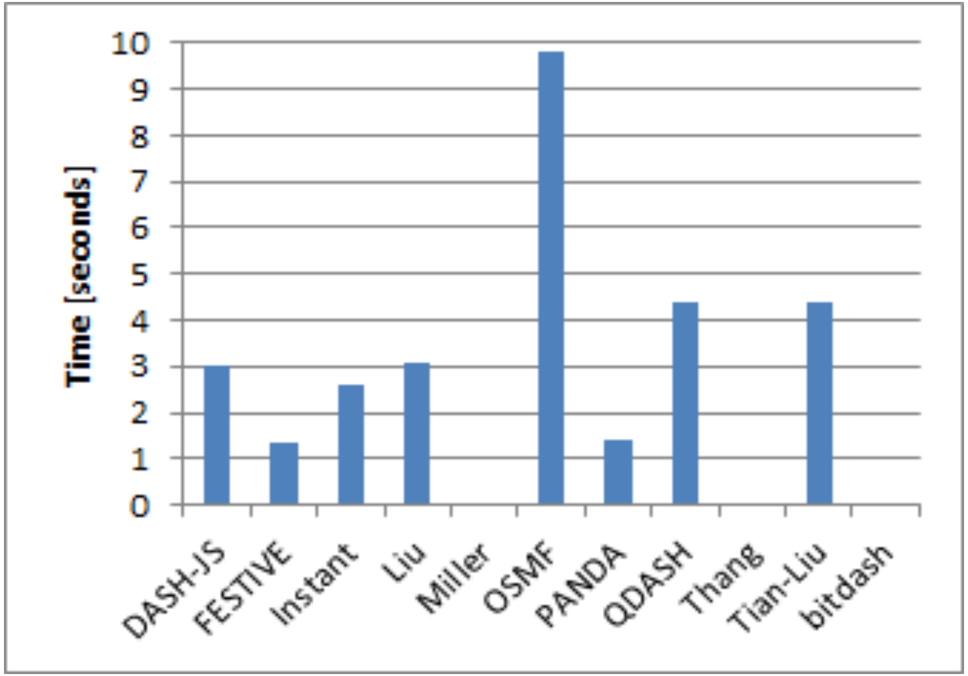


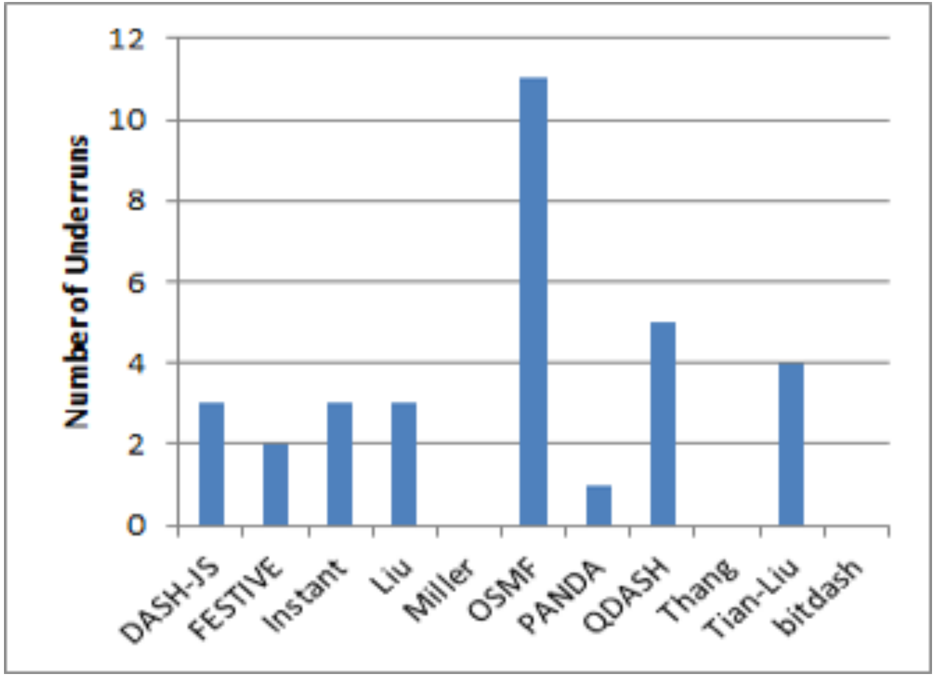
Figure 5. Bandwidth Adaptation with 10s segment size for (a) DASH-JS, (b) FESTIVE, (c) Instant, (d) Liu, (e) Miller, (f) OSMF, (g) PANDA, (h) QDASH, (i) Thang, and (j) Tian-Liu.

Average Bitrate (higher is better)



Stalls (lower is better)





Concluding Remarks

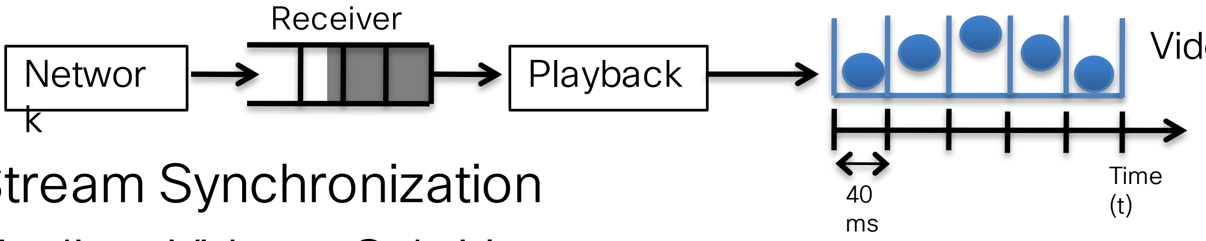
- Objective evaluation
  - Real-world network traces to emulate mobile environments with fluctuating bandwidth
  - Use “standard” metrics: startup delay, throughput, stalls, ...
- Subjective evaluation
  - Crowdsourcing & real-world environments a promising, cheap, fast, and reliable approach (also suitable for product development/testing)
  - Results indicate that the delivered representation bitrate (media throughput) and (the number of) stalls are the main influence factors on the QoE
- Still an issue, how to automate everything for live/on-demand services in real-time...

Part II: Common Problems in HTTP Adaptive Streaming

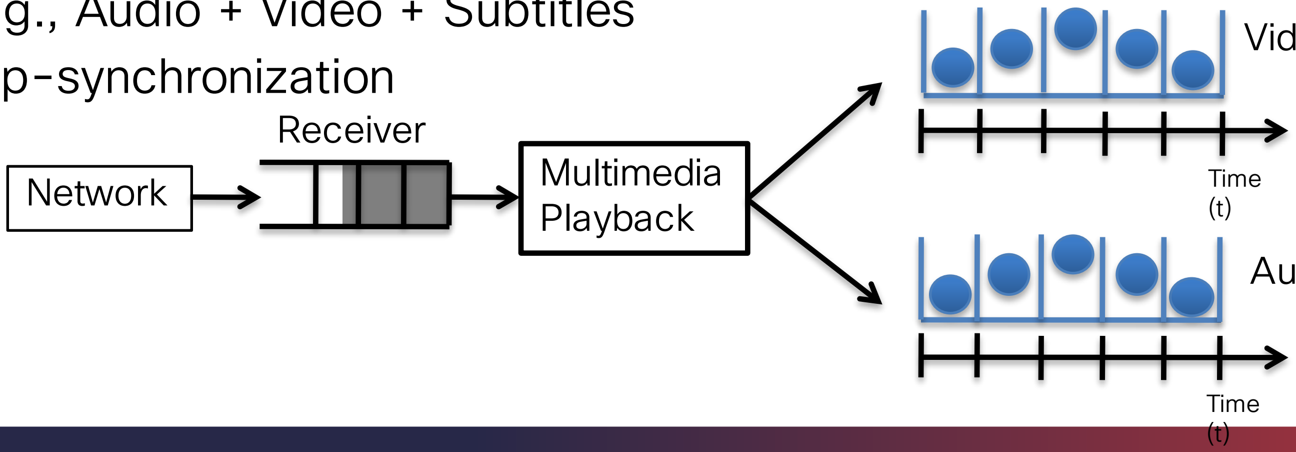
- Multi-Client Competition Problem
- Consistent-Quality Streaming
- QoE Optimization and Measurement
- Inter-Destination Media Synchronization

Types of Synchronization

- Intra-Stream Synchronization
  - Avoid jitter between the presentation of two consecutive media units

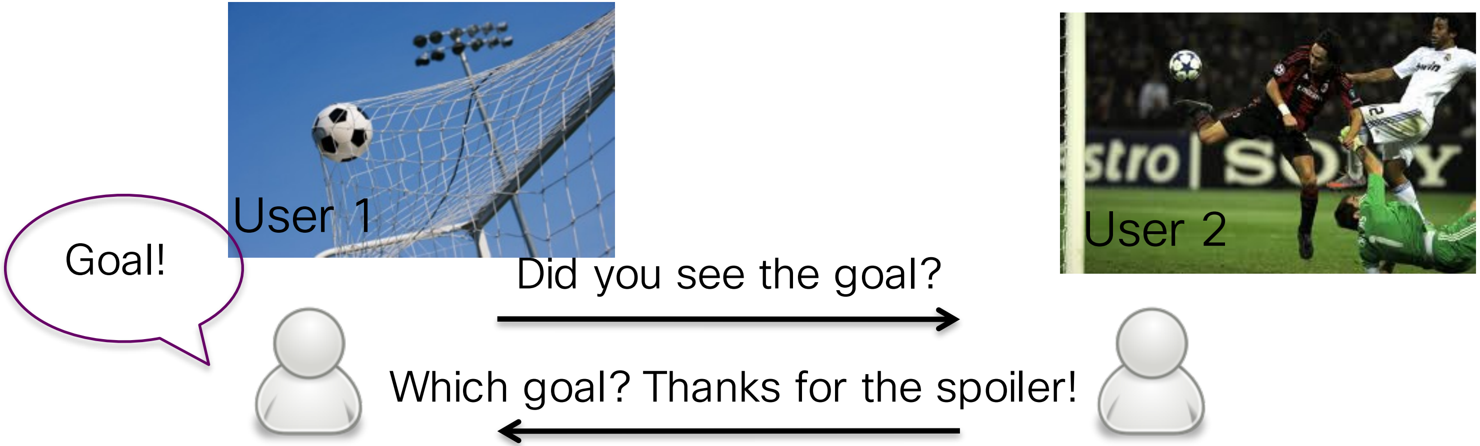


- Inter-Stream Synchronization
  - E.g., Audio + Video + Subtitles
  - Lip-synchronization



Inter-Destination Media Synchronization

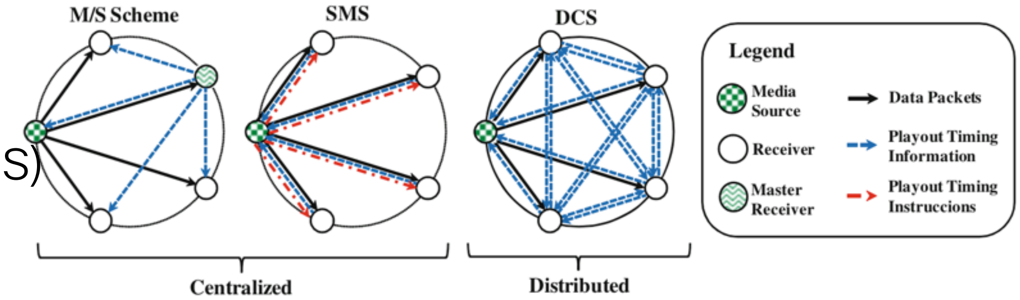
- IDMS == the playout of media streams at two or more geographically distributed locations in a time synchronized manner [draft-ietf-avtcore-idms-13]
- Watching multimedia content online together while geographically distributed, e.g., sport events, Twitch, online quiz shows, …
- SocialTV scenario featuring real-time communication via text, voice, video



C. Timmerer, B. Rainer, " The Social Multimedia Experience," *IEEE Computer*, vol. 47, no. 3, pp. 67-69, Mar., 2014

IDMS Building Blocks

- Building blocks
  - Session management
  - Identify the synchronization point and threshold of asynchronism
  - Signal timing and control information among the participating entities
  - Adapt the media playout to establish or restore synchronism
- IDMS schemes
  - Server/client (aka master/slave, MS)
  - Synchronization maestro scheme (SMS)
  - Distributed control scheme (DCS)



M. Montagud, F. Boronat, H. Stokking, R. van Brandenburg, " Inter-destination multimedia synchronization: schemes, use cases and standardization", *Multimedia Systems* (2012), 18:459-482.

Adaptive Media Playout

- Initially introduced for compensating the impact of error prone communication channels on the smoothness of the multimedia playout to avoid buffer under-/overruns
- Static, simple, naïve approach
  - Skip/pause content sections
  - Easy to implement, non-negligible QoE impact
- Dynamic Adaptive Media Playout (AMP)
  - Dynamically increase/decrease the playout rate for certain content sections
  - Find appropriate content sections where the media playout rate can be modified without significant impact on the QoE

QoE for IDMS

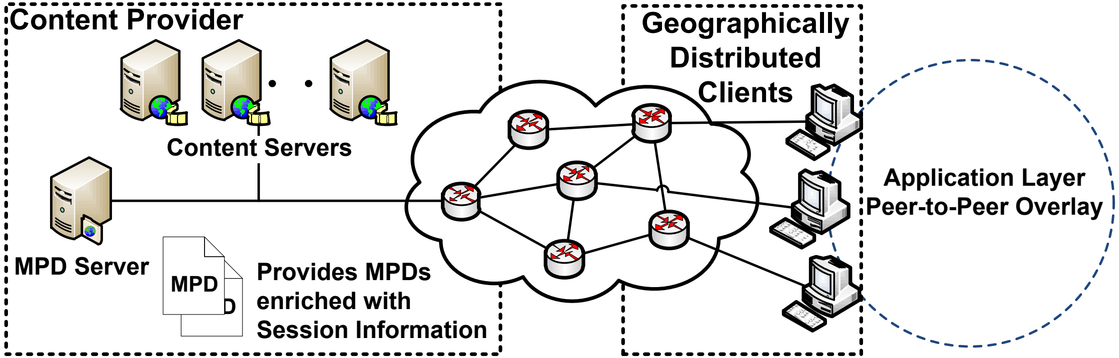
- Increasing/decreasing media playout rate

  ➪perceptual distortion in audio and/or video
- Select appropriate metrics, e.g.:
  - Audio: the spectral energy of an audio frame
  - Video: the average length of motion vectors between two consecutive frames
- How do metrics correlate with QoE? Find out…

  ➪Subjective quality assessments (w/ crowdsourcing)
- Define a utility model and incorporate into the media client to carry out the IDMS

B. Rainer, C. Timmerer, "A Quality of Experience Model for Adaptive Media Playout", *In Proceedings of QoMEX 2014*, Singapore, Sep 2014. B. Rainer, C. Timmerer, "Self-Organized Inter-Destination Multimedia Synchronization for Adaptive Media Streaming", *ACM Multimedia 2014*, Orlando, Florida, Nov. 2014.

Self-Organized IDMS for Adaptive Media Streaming



- Include IDMS Session Object (ISO) within MPEG-DASH Media Presentation Description
  - Time bounded entity to which a set of peers is assigned to
  - Unique identifier for a certain multimedia content
- P2P overlay construction & coarse synchronization
  - UDP & predefined message format; start segment for new peers
- Self-organized fine synchronization
  - Merge & Forward: flooding-based algorithm & bloom filters

Session Management

- IDMS Session Object (ISO)
  - Time bounded entity, contains a set of peers, uniquely identifiable
  - (IP, port) and the type of the Network Address Translator
  - Every peer numbers the peers in the ISO strict monotonically increasing beginning with one
- ISO is identified by session key
  - Provided by (3rd party) application or the user
- Integrated into the MPD of MPEG-DASH thanks to its extensibility
- Server includes the corresponding ISO when requested
  - E.g., a peer requests the MPD with a session key

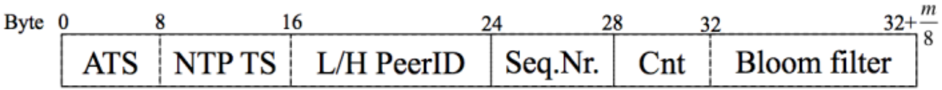
```xml
<MPD xmlns="urn:mpeg:dash:schema:mpd:2011" xmlns:iso="http://www
      .aau.at/DASH/Session" type="static"
      mediaPresentationDuration="PT3256S" minBufferTime="PT1.2S"
      profiles="urn:mpeg:dash:profile:isoff-on-demand:2011">
<BaseURL>http://www.example.com/</BaseURL>
<Period>
    <AdaptationSet>
        <Representation id="0" mimeType="video/mp4" codecs="avc1,
              mp4a" startWithSAP="1" bandwidth="1713804">
            <!-- ... representation info -->
        </Representation> <!-- ... more representations -->
    </AdaptationSet>
</Period>
<iso:IDMSSessionObject>
    <iso:PeerList>
        <iso:Peer>
            <iso:Identifier nat="NoNAT">
                <iso:IP>143.205.122.242</iso:IP>
                <iso:Port>8029</iso:Port>
            </iso:Identifier>
            <iso:Identifier nat="FullCone">
                <iso:IP>143.205.199.149</iso:IP>
                <iso:Port>8030</iso:Port>
            </iso:Identifier>
        </iso:Peer> <iso:Peer>
            <iso:Identifier nat="PortRestricted">
                <iso:IP>10.0.0.5</iso:IP>
                <iso:Port>8029</iso:Port>
            </iso:Identifier>
        </iso:Peer> <!-- ... more peers -->
     </iso:PeerList> <iso:TTL>2014-07-26T21:32:52</iso:TTL>
</iso:IDMSSessionObject>
</MPD>
```

Synchronization

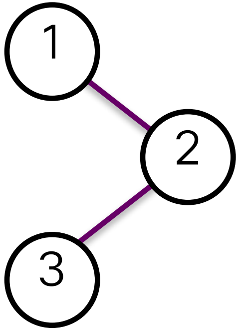
- Two phase synchronization using non-reliable communication (UDP) – reliable communication is not essential
  - Coarse synchronization for overlay creation and educated guess where to start downloading segments
  - Fine synchronization to negotiate reference playback timestamp in a self-organized way based on the constructed overlay
- Coarse synchronization
  - Periodically request PTS + NTP timestamp from known peers
  - Add unknown peers to list of peers upon receiving requests
  - Calculate start segment using *i)* maximum, *ii)* minimum, *iii)* average, or *iv)* weighted average of the received timestamps
  - Goal is to start with a segment that is as closest as possible to the segment the other peers are currently playing

Self-organized Fine Synchronization

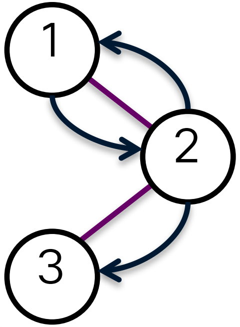
- Merge & Forward (M&F)
  - Distributed, self-organized
  - Flooding-based algorithm: periodically forwards information to neighbors
  - Calculates the average playback timestamp among the peers (merge)
- Bloom filter
  - Track which and how many peers have contributed to the average playback timestamp
  - Each peer uses the same set of hash functions
  - Fixed-length packet (i.e., scalability in terms of message size)



Merge & Forward – Example



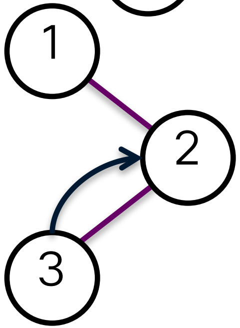
Initial state:
{BF, LID, HID, Cnt, ATS}
1: {{1}, 1, 1, 1, P1}
2: {{2}, 2, 2, 1, P2}
3: {{3}, 3, 3, 1, P3}



$\tau = 1$

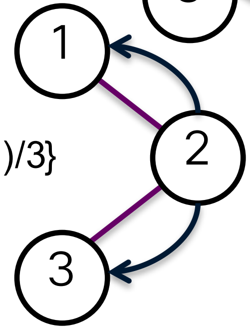
1: {{1,2}, 1, 2, 2, (P1+P2)/2}
2.1: {{1,2}, 1, 2, 2, (P1+P2)/2}
3: {{2,3}, 2, 3, 2, (P2+P3)/2}

$\tau = 1$



1: {{1,2}, 1, 2, 2, (P1+P2)/2}
2: {{1,2,3}, 1, 3, 3, (P3+2*(P1+P2/)/2)/3}
3: {{2,3}, 2, 3, 2, (P2+P3)/2}

$\tau = 2$



1: {{1,2,3}, 1, 3, 3, (P1+P2+P3)/3}
2: {{1,2,3}, 1, 3, 3, (P1+P2+P3)/3}
3: {{1,2,3}, 1, 3, 3, (P1+P2+P3)/3}

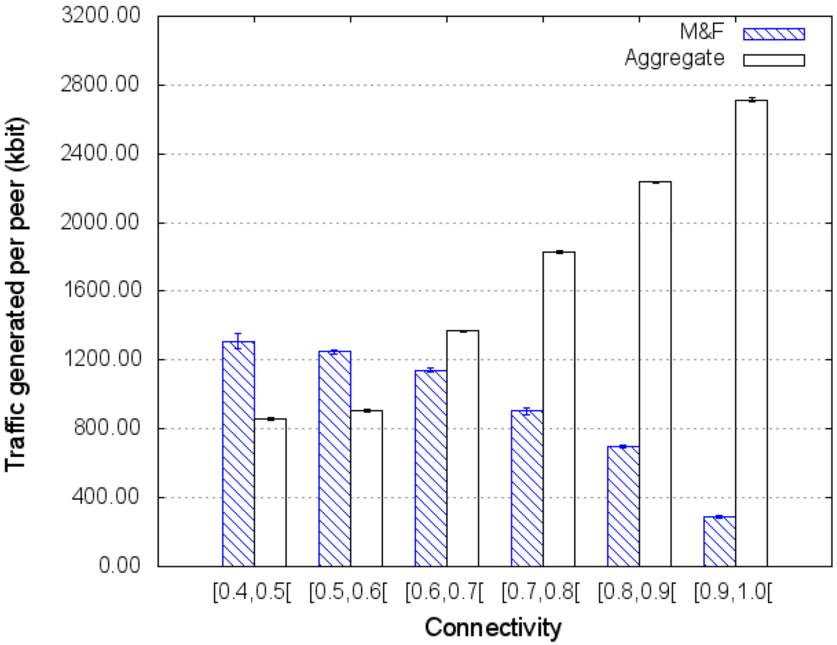
BF ... Bloom filter
H/LID ... Lowest/Highest Peer ID
Cnt ... Cumulative Count
ATS ... (Weighted) Average TS

Merge & Forward – Evaluation

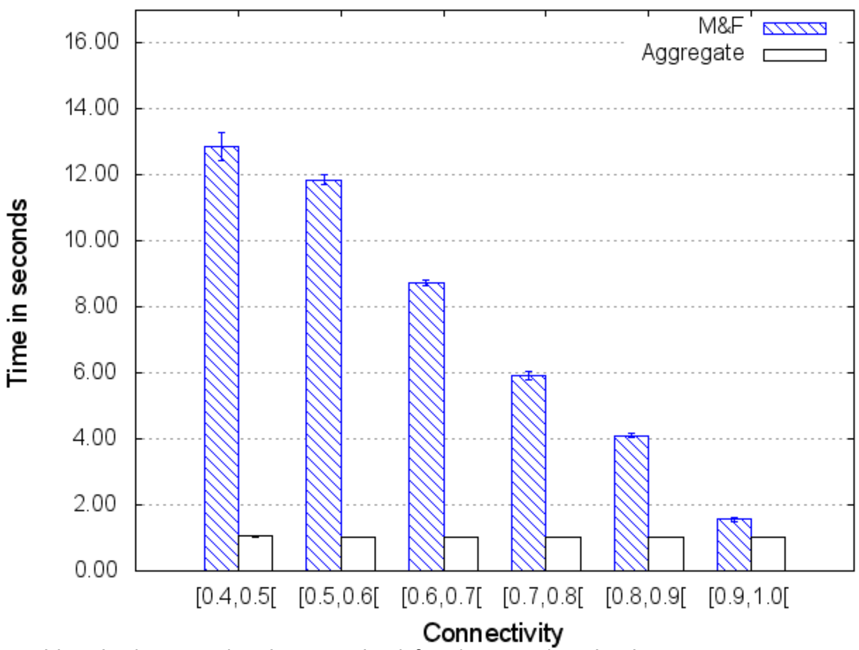
- Simulation environment OMNeT++ with INET framework
  - Random networks (Erdős-Rényi)
  - 40, 60, and 80 peers
  - Probabilities for creating connections between peers: 0.1 to 0.9 (uniformly distributed)
  - Period of 250ms and RTT of 300ms between peers
  - 30 simulation runs and take the average

- Compared to “Aggregate” which
  - Each peer aggregates all received playback timestamps into a single message (scalability!)
  - Periodically sends this aggregate to its neighbors

Merge & Forward – Evaluation



- Y-axis denotes the average traffic generated per peer in kbit
- X-axis denotes the connectivity of the overlay network
- For 80 peers



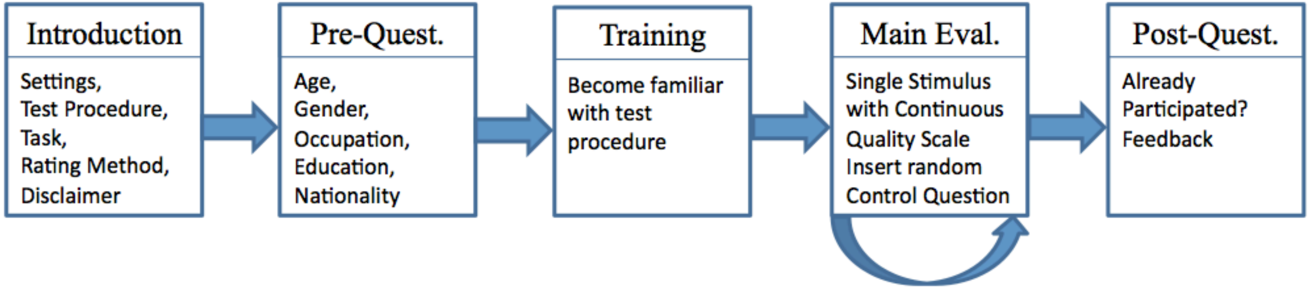
- Y-axis denotes the time required for the synchronization process
- X-axis denotes the connectivity of the overlay network
- For 80 peers

Dynamic Adaptive Media Playout

- Works on the content in the buffer (actually agnostic to the delivery format) for a given content section with a given duration (e.g., 2s)
- Goal: overcome asynchronism by increasing or decreasing the playback rate
  - Select those content sections which mask the playback rate variation
  - I.e., doesn't impact QoE significantly
- Content features for measuring the distortion caused by AMP
  - Audio: spectral energy of audio frames
  - Video: motion intensity (length of motion vectors) between consecutive video frames
- Metrics for the distortion
  - Difference between the impaired (increased/decreased playback rate) and the unimpaired case (nominal playback rate) for both modalities (audio/video)

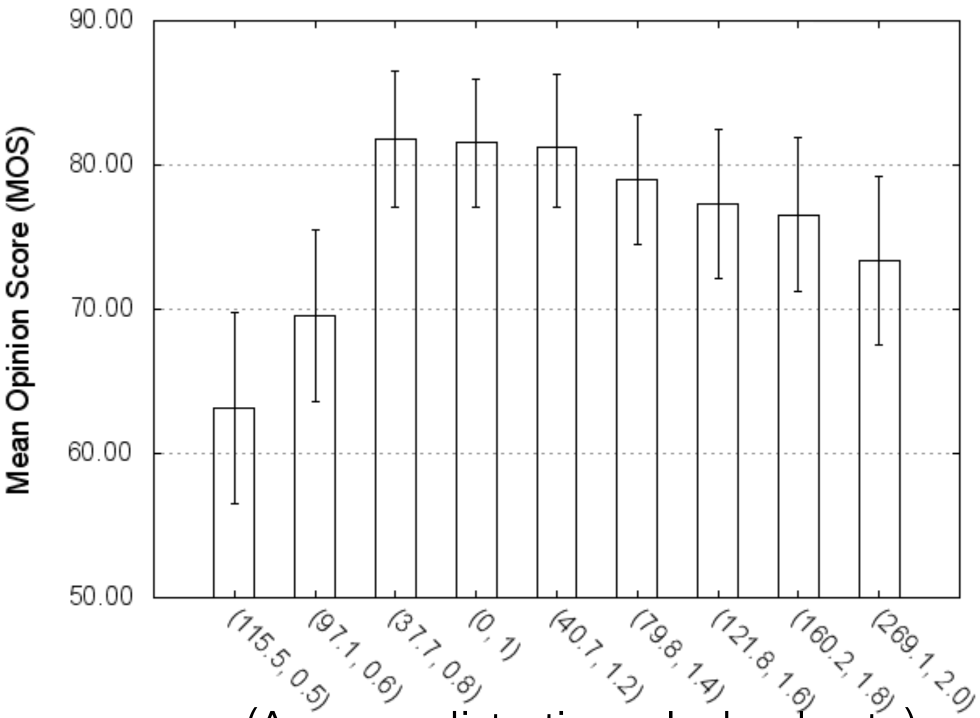
Dynamic AMP – Evaluation

- Subjective Quality Assessment using Crowdsourcing
  - Microworkers platform, 80 participants, 55 used for evaluation
  - Duration of the test: 15 minutes
  - Reward/compensation: $0.25
- Sequences
  - Babylon A.D. for training
    - {1, 0.5, 2} times the nominal playback rate
  - Big Buck Bunny for the main evaluation
    - {0.5, 0.6, 0.8, 1,1.2, 1.4, 1.6, 1.8, 2} times the nominal playback rate
    - Selected content sections according to our approach which are presented to the participants



Dynamic AMP – Evaluation Results

- No significant difference in QoE degradation for playback rate in range of [0.8, 1.8]
- Significant QoE degradation for other playback rates, i.e., 0.5, 0.6, 2.0
- Strong linear correlation between our distortion metric and the MOS assessed by the subjective quality assessment



(Average distortion, playback rate)

Concluding Remarks

- Introduced IDMS to over-the-top streaming including but not limited to MPEG-DASH

- Distributed Control Scheme that scales with the number of peers
  - Can be combined with any streaming protocol
  - Not coupled with the session management or the overlay creation

- Dynamic AMP for carrying out the actual synchronization
  - (General) Optimization problem that aims on finding appropriate content sections
  - Reduce QoE impact

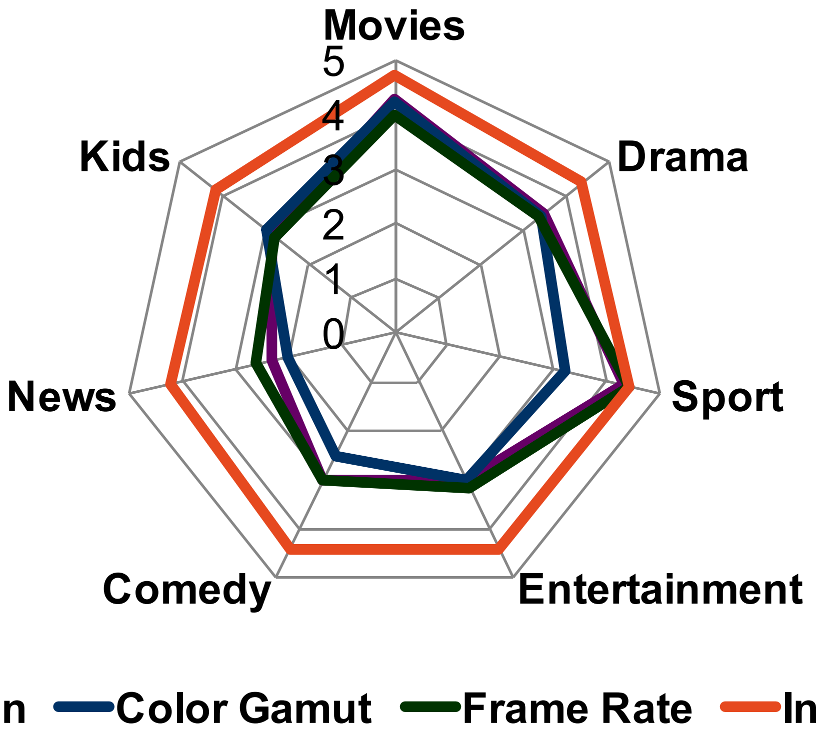
- Demo video available on YouTube
  - https://www.youtube.com/watch?v=2V9rO5SbI7A
  - Search for: MergeAndForward
  - Source Code available at: https://github.com/grishnagkh/mf

Part III: Open Issues and Future Research Directions

Four Major Areas of Focus
Things We Assume We Know All about

- Content Preparation
  - Choosing target bitrates/resolutions to make switching as seamless as possible
  - Determining segment durations
  - Encoding the content so that the perceived quality is stable and good even in the case of frequent up/downshifts
- Distribution and Delivery
  - Current approaches treat network as a “black box”
    - Intuitively, exchange of information should provide improvement
  - Can or should we provide controlled unfairness on the server or in the network?
  - Would better caching/replication/pre-positioning content avoid the overload?
  - Is there a better transport than TCP, maybe MPTCP, DCCP, SCTP, or QUIC?
  - Should we consider IP multicast to help reduce bandwidth usage?
- Quality-of-Experience (QoE) Modeling and Client Design
- Analytics, Fault Isolation and Diagnostics

One Strategy may not Work for All Content Types



Source: Screen Digest (Higher value indicates more importance)

Modeling and Measuring Quality of Experience

Understanding the Impact of QoE on Viewer Engagement

- How can we
  - Model adaptive streaming dynamics such as rate/resolution shifting for different genres?
  - Take into account shorter buffering and faster trick modes in this model?
- Does QoE impact viewer engagement?
  - If yes, how?

We need to be able to answer these questions for:

- Designing a client that takes QoE into account
- Keeping viewers happy and engaged, subsequently increasing ad revenues

Research Directions in Streaming

Reading

“Probe and adapt: rate adaptation for HTTP video streaming at scale,”
IEEE JSAC, Apr. 2014

“Streaming video over HTTP with consistent quality,”
ACM MMSys, 2014

“Caching in HTTP adaptive streaming: friend or foe?,”
ACM NOSSDAV, 2014

“Self-organized inter-destination multimedia synchronization for adaptive media streaming,”
ACM Multimedia, 2014

“The social multimedia experience,”
IEEE Computer, 2014

“Crowdsourcing quality-of-experience assessments,”
IEEE Computer, 2014

Cisco Research Seeking Proposals

http://research.cisco.com

- Several RFPs about video delivery, though RFP-2014-010 is specifically designed for adaptive streaming research
- Interest Areas
  - Design of server-side, client-side, and network-based adaptation methods and hybrids of the three
  - Comparison of reliable multicast distribution vs. adaptive unicast streaming for broadcast (live) content
  - Investigation of the impact of adaptive transport in large-scale deployments
  - Development of instrumentation needed to assess the effectiveness of adaptive transport

Further Reading and References

Further Reading and References

Adaptive Streaming

- **Overview Articles**
  - “Watching video over the Web, part 2: applications, standardization, and open issues,” IEEE Internet Computing, May/June 2011
  - “Watching video over the Web, part 1: streaming protocols,” IEEE Internet Computing, Mar./Apr. 2011
- **VideoNext workshop in ACM CoNEXT 2014**
  - http://conferences2.sigcomm.org/co-next/2014/Workshops/VideoNext/
- **Special Issue on Adaptive Media Streaming**
  - IEEE JSAC – Apr. 2014
- **Special Session in Packet Video Workshop 2013**
  - Technical program and slides: http://pv2013.itec.aau.at/
- **Special Sessions in ACM MMSys 2011**
  - Technical program and slides: at http://www.mmsys.org/?q=node/43
  - VoDs of the sessions are available in ACM Digital Library
    - http://tinyurl.com/mmsys11-proc (Requires ACM membership)
- **Multimedia Communication Blog**
  - http://multimediacommunication.blogspot.co.at
- **W3C Web and TV Workshops**
  - http://www.w3.org/2013/10/tv-workshop/
  - http://www.w3.org/2011/09/webtv
  - http://www.w3.org/2010/11/web-and-tv/

Further Reading and References

Source Code for Adaptive Streaming Implementations

- **DASH Industry Forum**
  - http://dashif.org/software/
- **Open Source Implementations/Frameworks**
  - http://dash.itec.aau.at/
  - http://gpac.wp.mines-telecom.fr/
  - https://github.com/bitmovin/libdash
  - https://github.com/google/shaka-player
- **Microsoft Media Platform: Player Framework**
  - http://playerframework.codeplex.com/
- **Adobe OSMF**
  - http://sourceforge.net/adobe/osmf/home/Home/
- **JW Player**
  - https://github.com/jwplayer/jwplayer

Further Reading and References

Adaptive Streaming Demos

- **DASH**
  - http://dash-mse-test.appspot.com/dash-player.html
  - http://dashif.org/reference/players/javascript/index.html
  - https://github.com/google/shaka-player
- **Akamai HD Network**
  - http://wwwns.akamai.com/hdnetwork/demo/flash/default.html
  - http://wwwns.akamai.com/hdnetwork/demo/flash/hds/index.html
  - http://wwwns.akamai.com/hdnetwork/demo/flash/hdclient/index.html
  - http://bit.ly/testzeri
- **Microsoft Smooth Streaming**
  - http://www.iis.net/media/experiencesmoothstreaming
  - http://www.smoothhd.com/
- **Adobe OSMF**
  - http://blogs.adobe.com/osmf/
- **Apple HTTP Live Streaming (Requires QuickTime X or iOS)**
  - http://devimages.apple.com/iphone/samples/bipbopall.html
- **bitdash**
  - http://www.dash-player.com/

MMSYS 2016

May 10-13, 2016
Klagenfurt am Wörthersee
AUSTRIA

Submission deadline: November 27, 2015

http://www.mmsys.org/ | http://mmsys2016.itec.aau.at/ | @mmsys2015