Dynamic Adaptive Streaming over HTTP: From Content Creation to Consumption

Christian Timmerer

Alpen-Adria-Universität Klagenfurt
Universitätsstraße 65-67
A-9020 Klagenfurt am Wörthersee
+43 463 2700 3621
christian.timmerer@itec.aau.at

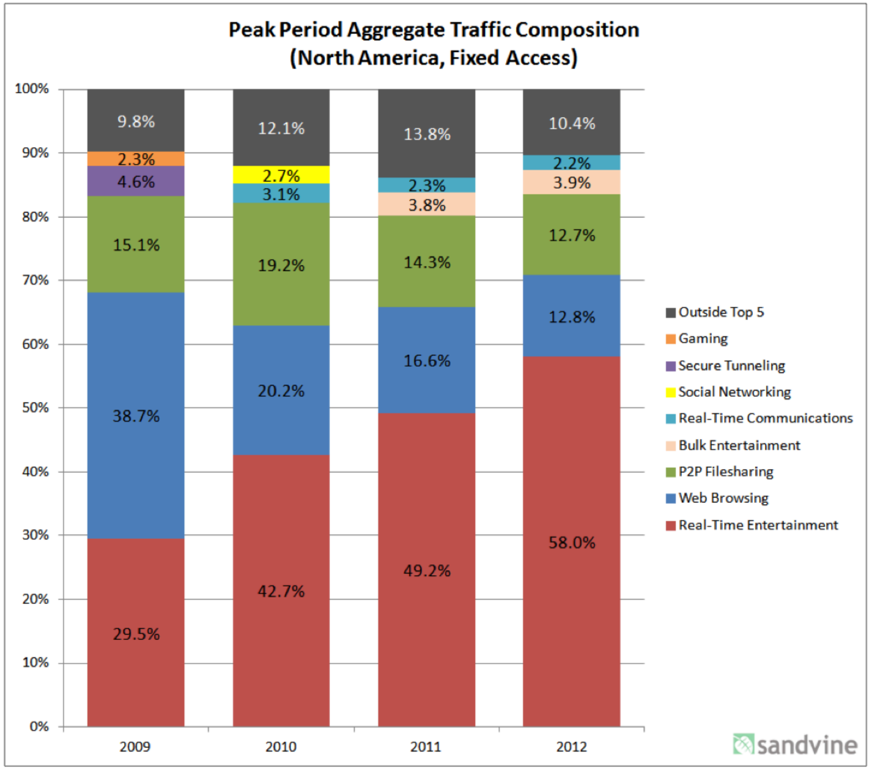
Carsten Griwodz

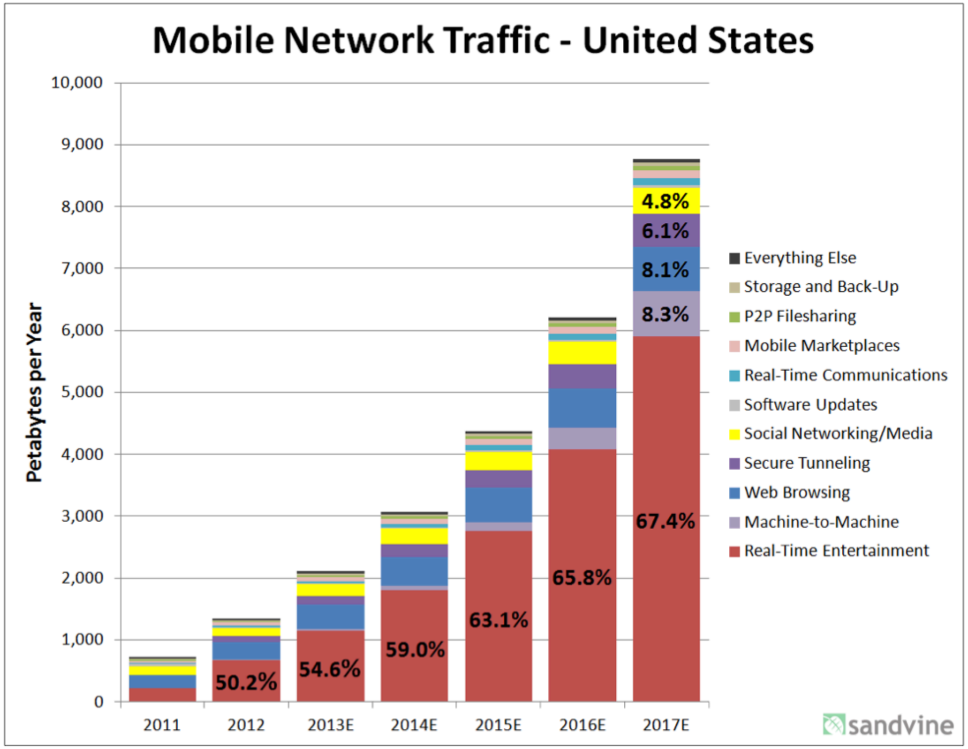
Simula Research Laboratory
P.O.Box 134
1325 Lysaker, Norway
+47 67 82 82 00
griff@simula.no

ACM Multimedia 2012, Nara, Japan
29th October, 2012

Video Predominant on the Internet

- Real-time video is more than 50% of the traffic at peak periods
- Mobile traffic is growing exponentially, all delivered over the top (OTT)



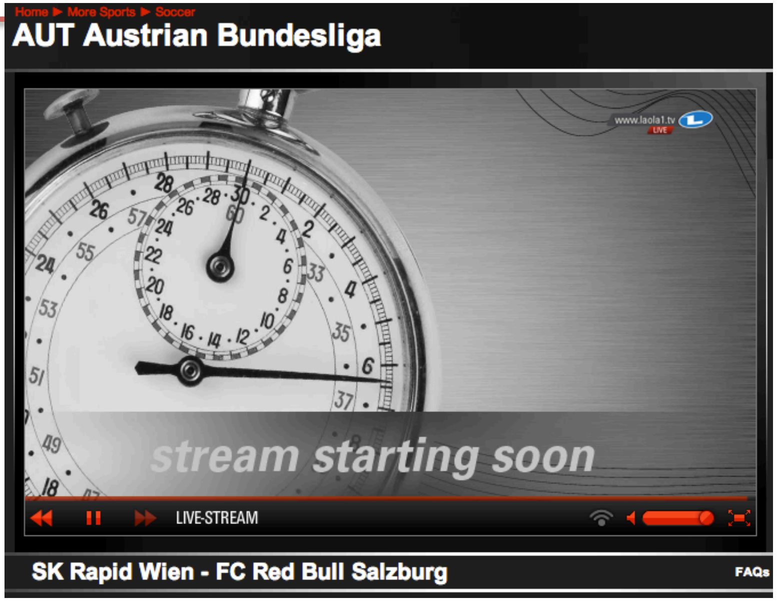


http://www.sandvine.com/downloads/documents/Phenomena_1H_2012/Sandvine_Global_Internet_Phenomena_Report_1H_2012.pdf

... but User Frustration is High!!!

- Wrong format
- Wrong protocol
- Plugin required
- DRM issues
- Long start-up delay
- Low quality
- Frequent stalls
- Bitrate intense
- No DVD/PVR experience

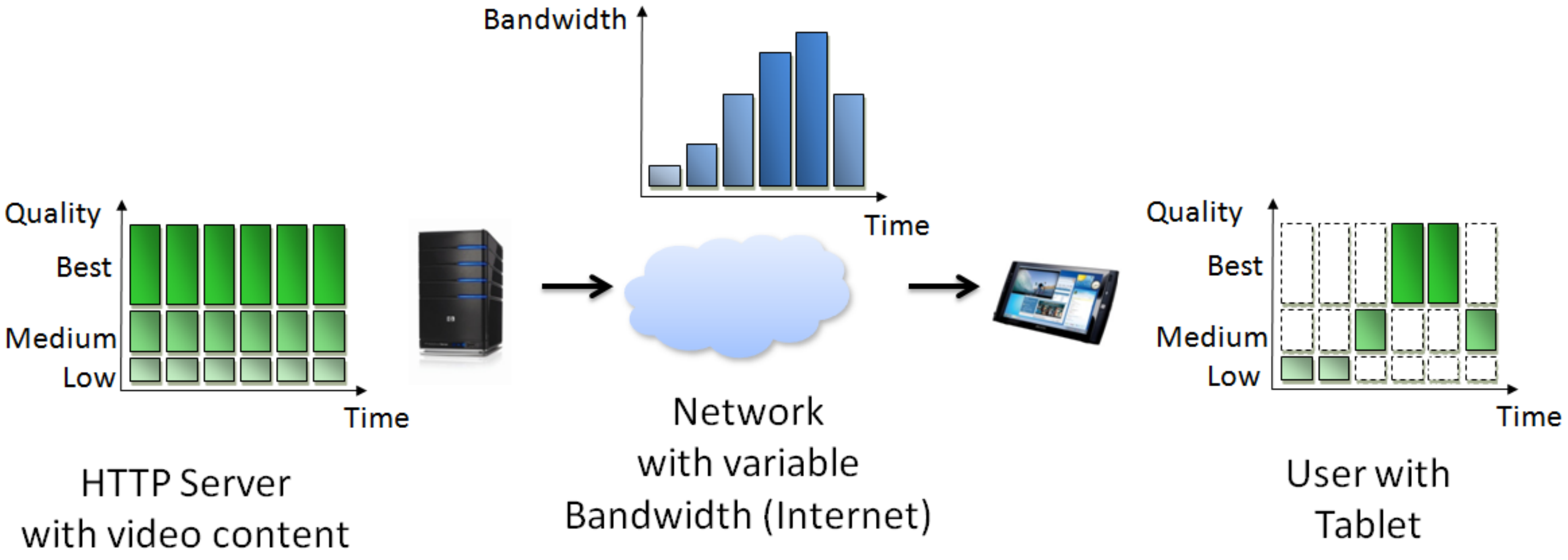
Let’s do something ...



Ack: Thomas Stockhammer

Dynamic Adaptive Streaming over HTTP

- In a nutshell …



HTTP Server
with video content

Network
with variable
Bandwidth (Internet)

User with
Tablet

Ack & ©: Christopher Müller

Presenters

Christian Timmerer

Assistant professor at the Institute of Information Technology (ITEC), Multimedia Communication Group (MMC), Alpen-Adria-Universität Klagenfurt, Austria. His research interests include immersive multimedia communication, streaming, adaptation, and Quality of Experience (QoE). He was the general chair of WIAMIS'08, AVSTP2P'10 (co-located with ACMMM'10), WoMAN'11 (co-located with ICME'11), and TPC co-chair of QoMEX'12. He has participated in several EC-funded projects, notably DANAE, ENTHRONE, P2P-Next, ALICANTE, and SocialSensor. He is an Associate Editor for IEEE Computer Science Computing Now, Area Editor for Elsevier Signal Processing: Image Communication, Review Board Member of IEEE MMTC, editor of ACM SIGMM Records, and member of ACM SIGMM Open Source Software Committee. He also participated in ISO/MPEG work for several years, notably in the area of MPEG-21, MPEG-M, MPEG-V, and DASH (incl. DASH promoters group). He received his PhD in 2006 from the Klagenfurt University. Publications and MPEG contributions can be found under http://research.tim merer.com, follow him on http://www.twitter.com/timse7, and subscribe to his blog http://blog.timmerer.com. Full bio can be found at http://www-itec.uni-klu.ac.at/~timse/cv/.

Carsten Griwodz

Head of the Media department of research company Simula Research Laboratory, and professor of Computer Science at the University of Oslo. He received his Dipl.-Inf. degree from Paderborn University in 1993 and Dr.-Ing. degree from Technische Universität Darmstadt in 2000. He worked for IBM from 1993–98 and participated in the standardization of MHEG. His research is concerned with streaming media, ranging from scalable distribution architectures through operating system and protocol support to subjective visual quality assessment. He was co-chair of ACM NOSSDAV 2008, ACM/IEEE NetGames 2011, SPIE/ACM MMCN 2006 and 2007, Track chair of ACM MM 2008, TPC chair of ACM MMSys 2012 and is general chair of MMSys 2013. He is Associate Editor of ACM TOMCCAP and Editor-in-Chief of the newsletter ACM SIGMM Records. The Media group publishes news at http://mpg.ndlab.net. His publications can be found at http://simula.no/people/griff/bibliography.

Tutorial Outline

- **MPEG Dynamic Adaptive Streaming over HTTP (DASH)**
  - Scope and design principles
  - Data model
  - Profiles
- **DASH "Encoder", Dataset, and Players**
  - GPAC
  - VLC media player plugin, libdash
  - Javascript & HTML5 => DASH-JS
- **Evaluation of DASH**
  - Mobile, heterogeneous environments
  - Quality of Experience (QoE)
  - Synchronization

Dynamic Adaptive Streaming over HTTP (DASH)

Christian Timmerer and Christopher Müller

Alpen-Adria-Universität Klagenfurt (AAU) ◆ Faculty of Technical Sciences (TEWI) ◆ Department of Information Technology (ITEC) ◆ Multimedia Communication (MMC) ◆ Sensory Experience Lab (SELab)

research.timmerer.com ◆ blog.timmerer.com ◆ selab.itec.aau.at ◆ dash.itec.aau.at ◆ stcsn.ieee.net

christian.timmerer@itec.aau.at

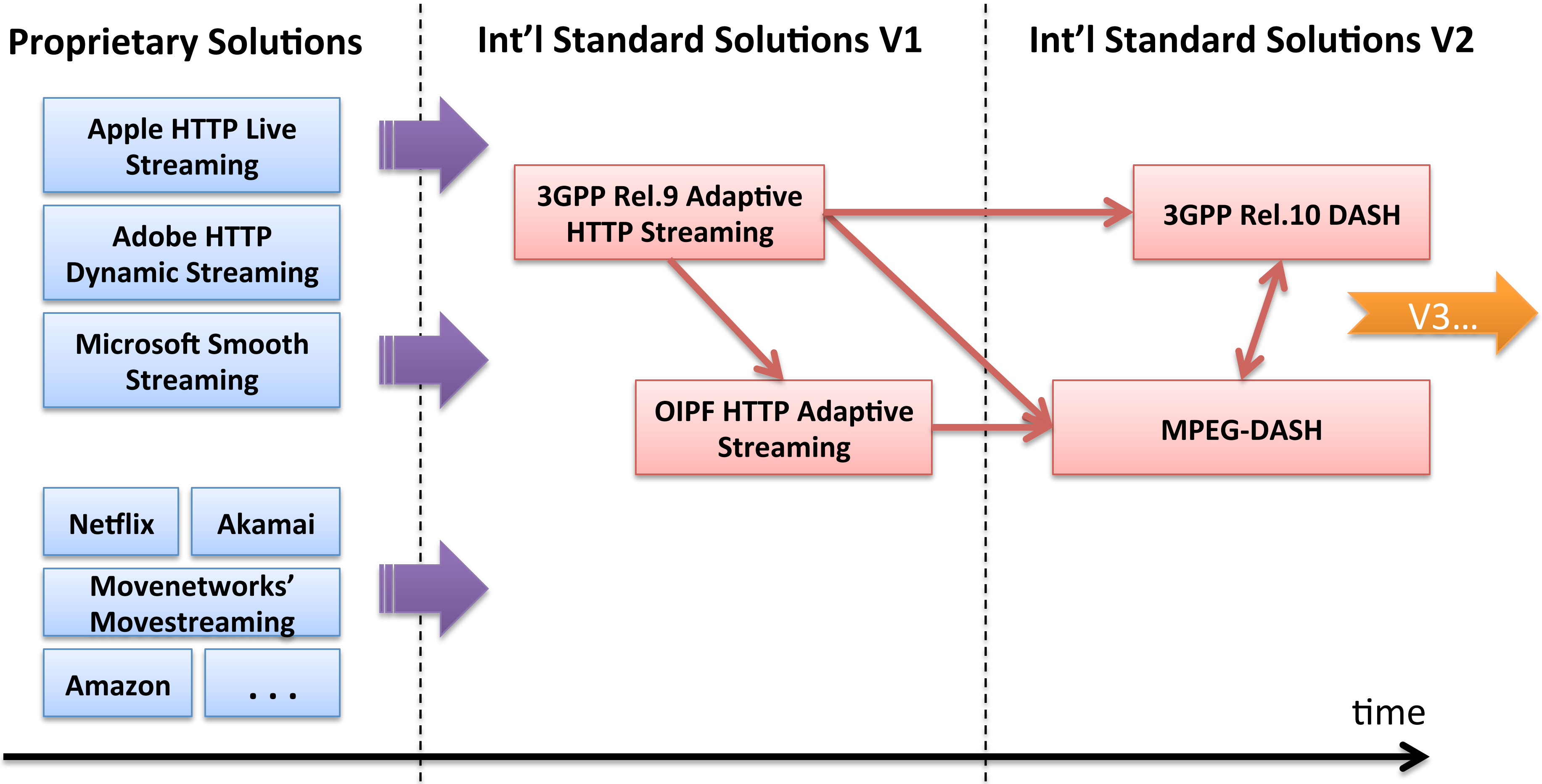
ACM Multimedia 2012, Nara, Japan
29th October, 2012

Acknowledgment: Thomas Stockhammer (QUALCOMM), Mark Watson (Netflix), Iraj Sodagar (Microsoft)

What is DASH?

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

Dynamic Adaptive Streaming over HTTP (DASH)



Proprietary Solutions

Apple HTTP Live Streaming

Adobe HTTP Dynamic Streaming

Microsoft Smooth Streaming

Netflix

Akamai

Movenetworks' Movestreaming

Amazon

. . .

Int'l Standard Solutions V1

3GPP Rel.9 Adaptive HTTP Streaming

OIPF HTTP Adaptive Streaming

Int'l Standard Solutions V2

3GPP Rel.10 DASH

MPEG-DASH

V3…

time

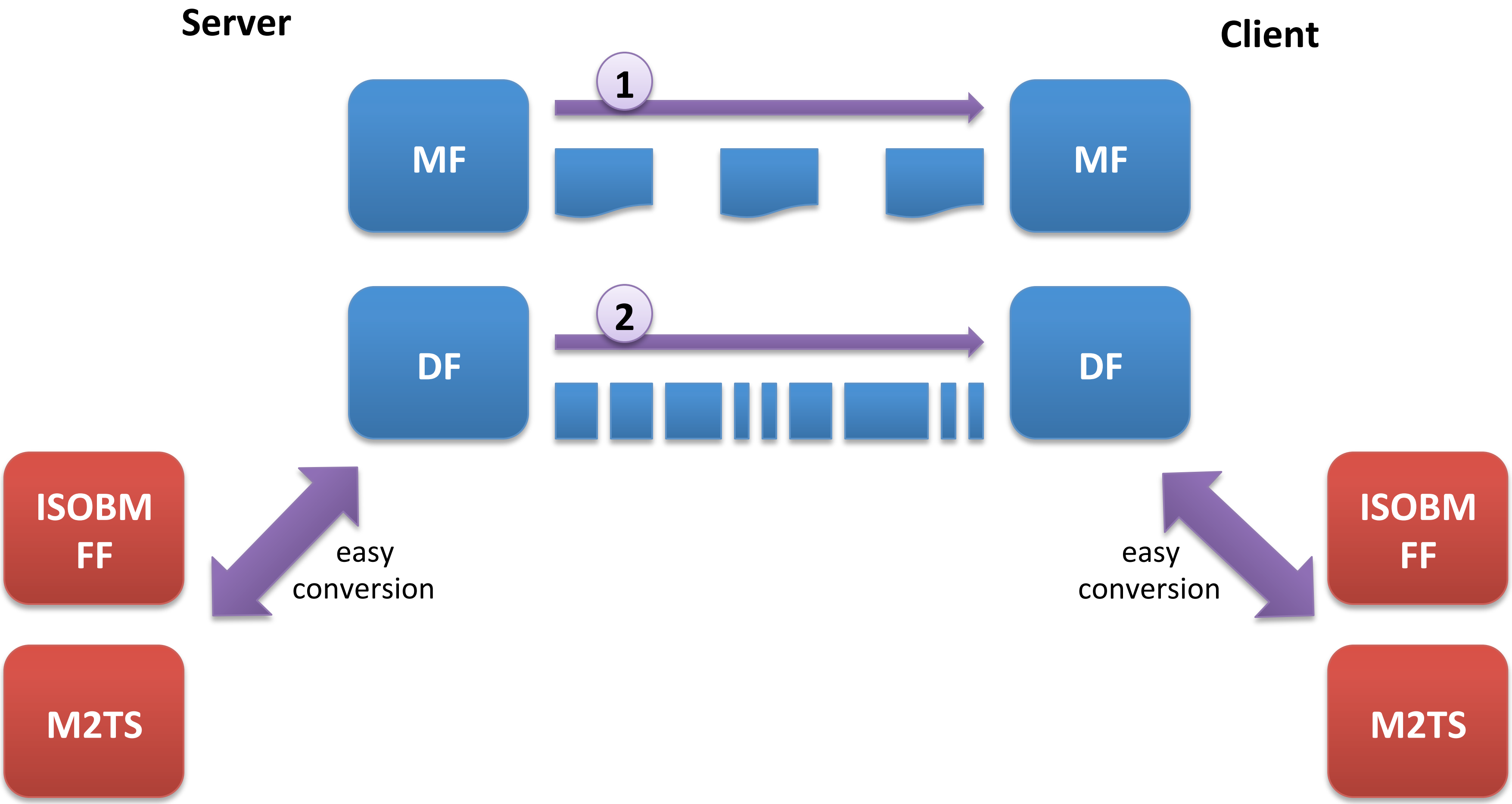
http://multimediacommunication.blogspot.com/2010/05/http-streaming-of-mpeg-media.html

DASH Design Principles

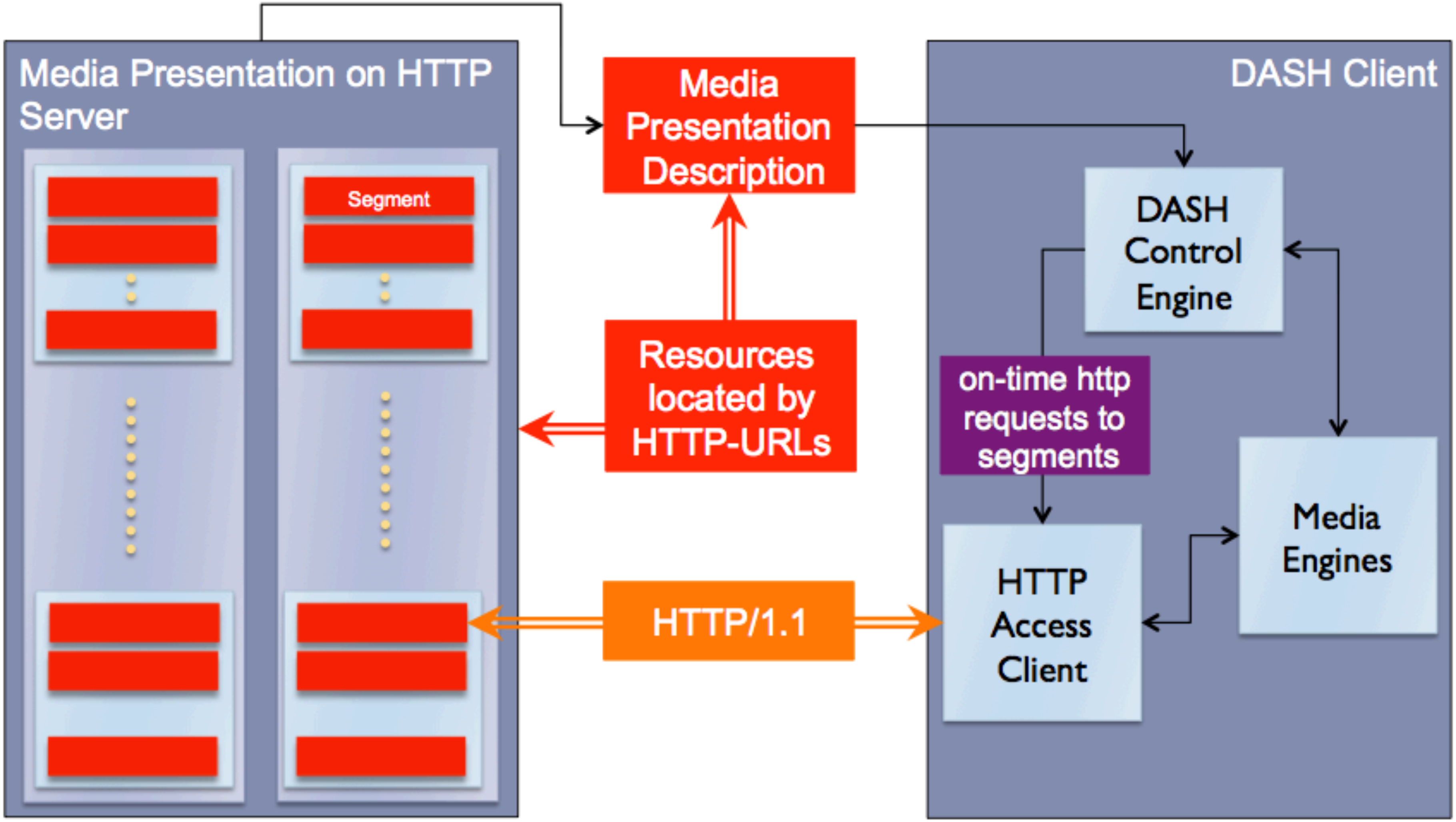
- DASH is not
  - system, protocol, presentation, codec, interactivity, DRM, client specification
- DASH is an enabler
  - It provides formats to enable efficient and high-quality delivery of streaming services over the Internet
  - It is considered as one component in an end-to-end service
  - System 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

Scope of DASH

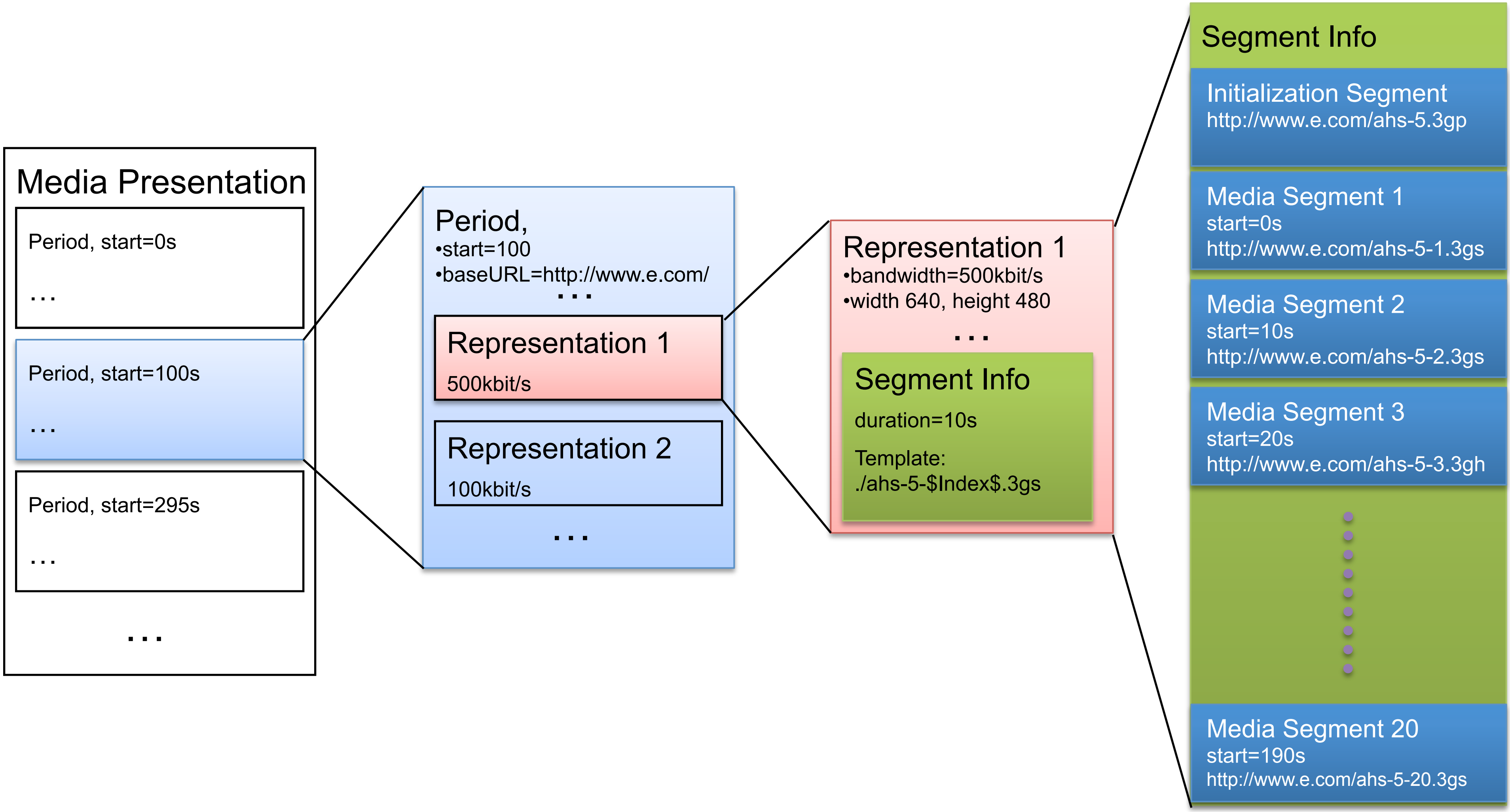


What is specified – and what is not?


Media Presentation on HTTP Server
Segment
Media Presentation Description
Resources located by HTTP-URLs
HTTP/1.1
DASH Client
DASH Control Engine
on-time http requests to segments
HTTP Access Client
Media Engines

Ack & ©: Thomas Stockhammer

DASH Data Model



Media Presentation

Period, start=0s
…

Period, start=100s
…

Period, start=295s
…

…

Period,
•start=100
•baseURL=http://www.e.com/
…

Representation 1
500kbit/s

Representation 2
100kbit/s

…

Representation 1
•bandwidth=500kbit/s
•width 640, height 480
…

Segment Info
duration=10s
Template:
./ahs-5-$Index$.3gs

Segment Info

Initialization Segment
http://www.e.com/ahs-5.3gp

Media Segment 1
start=0s
http://www.e.com/ahs-5-1.3gs

Media Segment 2
start=10s
http://www.e.com/ahs-5-2.3gs

Media Segment 3
start=20s
http://www.e.com/ahs-5-3.3gh

Media Segment 20
start=190s
http://www.e.com/ahs-5-20.3gs

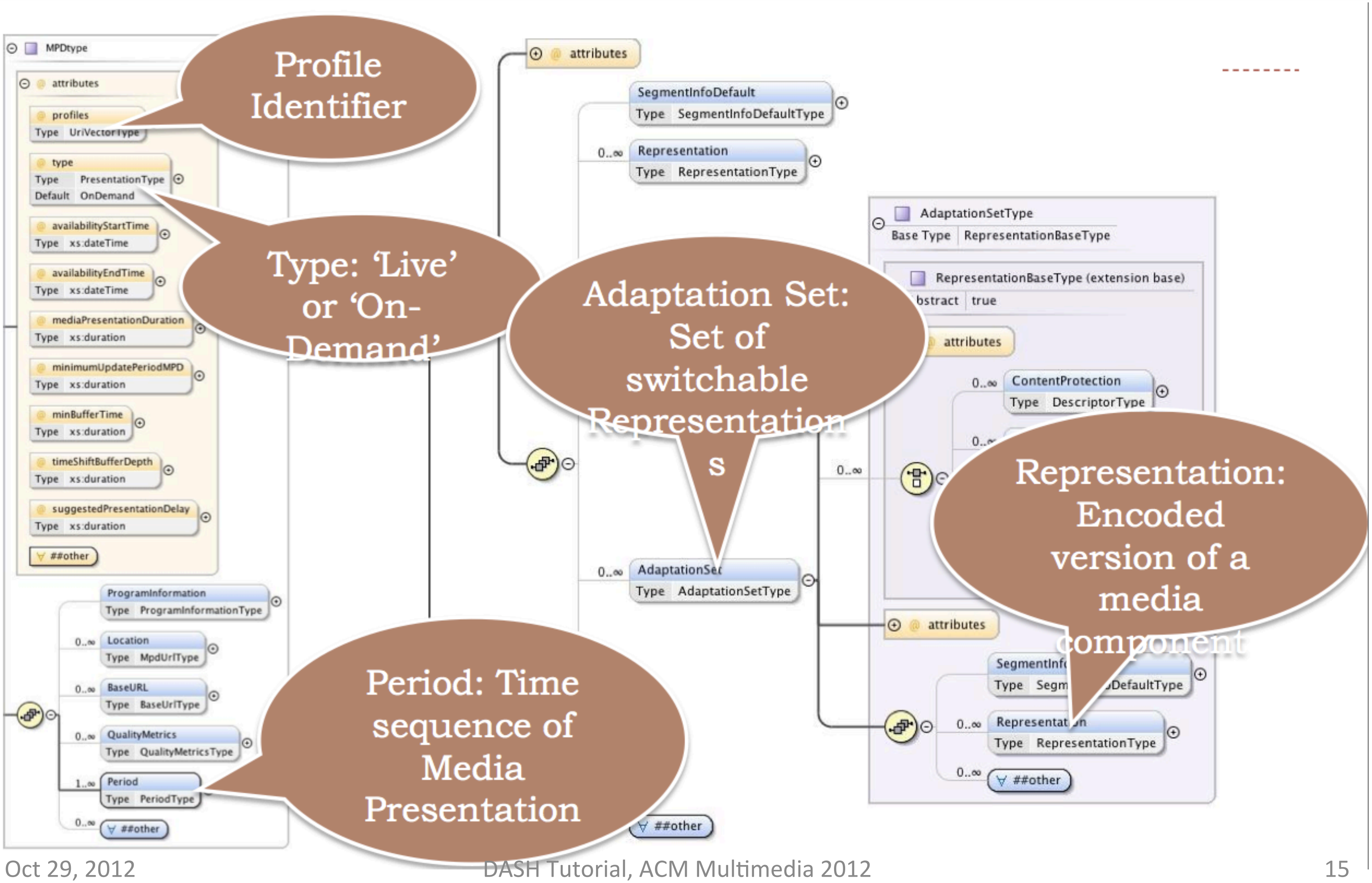
Ack & ©: Thomas Stockhammer

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

Ack & ©: Thomas Stockhammer

MPD Schema Overview



Profile Identifier

Type: 'Live' or 'On-Demand'

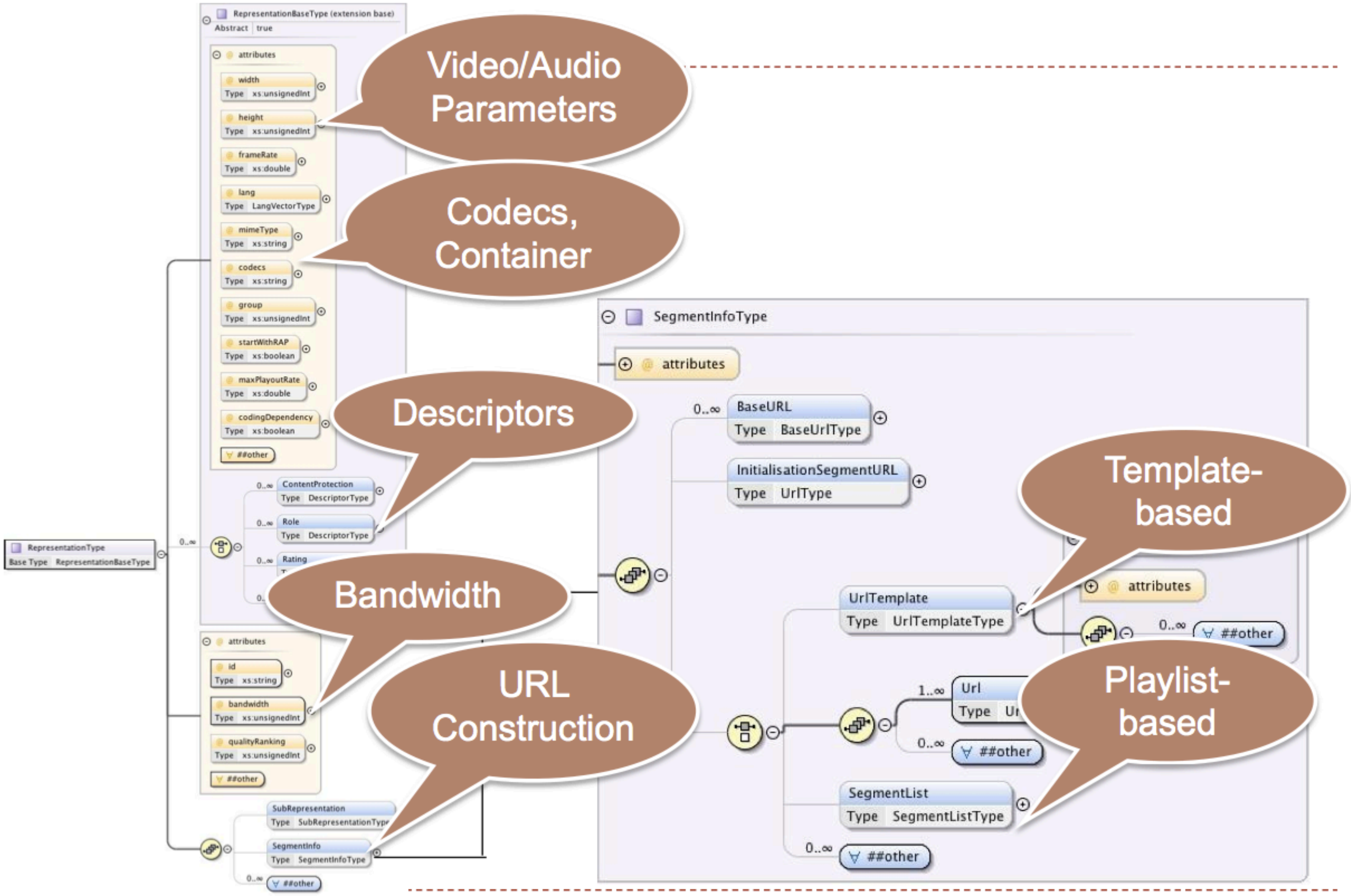
Adaptation Set: Set of switchable Representations

Representation: Encoded version of a media component

Period: Time sequence of Media Presentation

Ack & ©: Iraj Sodagar

MPD Schema - Representation


RepresentationBaseType (extension base)
Abstract true
attributes
width Type xs:unsignedInt
height Type xs:unsignedInt
frameRate Type xs:double
lang Type LangVectorType
mimeType Type xs:string
codecs Type xs:string
group Type xs:unsignedInt
startWithRAP Type xs:boolean
maxPlayoutRate Type xs:double
codingDependency Type xs:boolean
##other
0..∞ ContentProtection Type DescriptorType
0..∞ Role Type DescriptorType
0..∞ Rating
RepresentationType
Base Type RepresentationBaseType
0..∞
attributes
id Type xs:string
bandwidth Type xs:unsignedInt
qualityRanking Type xs:unsignedInt
##other
SubRepresentation Type SubRepresentationType
SegmentInfo Type SegmentInfoType
0..∞ ##other
SegmentInfoType
attributes
0..∞ BaseURL Type BaseUrlType
InitialisationSegmentURL Type UrlType
UrlTemplate Type UrlTemplateType
attributes
0..∞ ##other
1..∞ Url Type Ur
0..∞ ##other
SegmentList Type SegmentListType
0..∞ ##other

Video/Audio Parameters

Codecs, Container

Descriptors

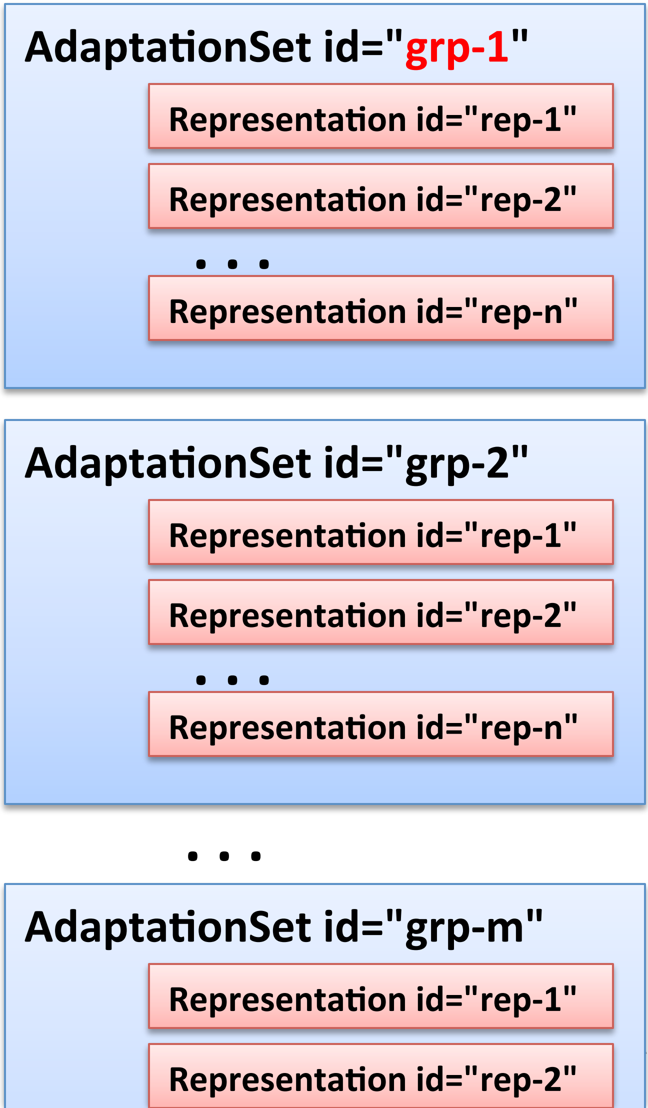
Bandwidth

URL Construction

Template-based

Playlist-based

DASH AdaptationSets & Subsets


AdaptationSet id="grp-1"

Representation id="rep-1"

Representation id="rep-2"

. . .

Representation id="rep-n"

AdaptationSet id="grp-2"

Representation id="rep-1"

Representation id="rep-2"

. . .

Representation id="rep-n"

. . .

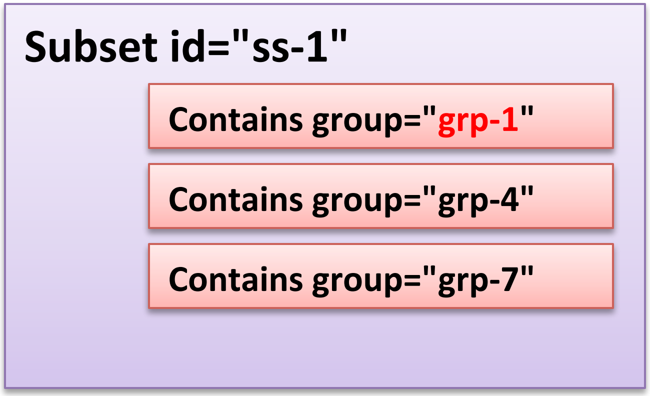
AdaptationSet id="grp-m"

Representation id="rep-1"

Representation id="rep-2"

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

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


Subset id="ss-1"

Contains group="grp-1"

Contains group="grp-4"

Contains group="grp-7"

Subsets

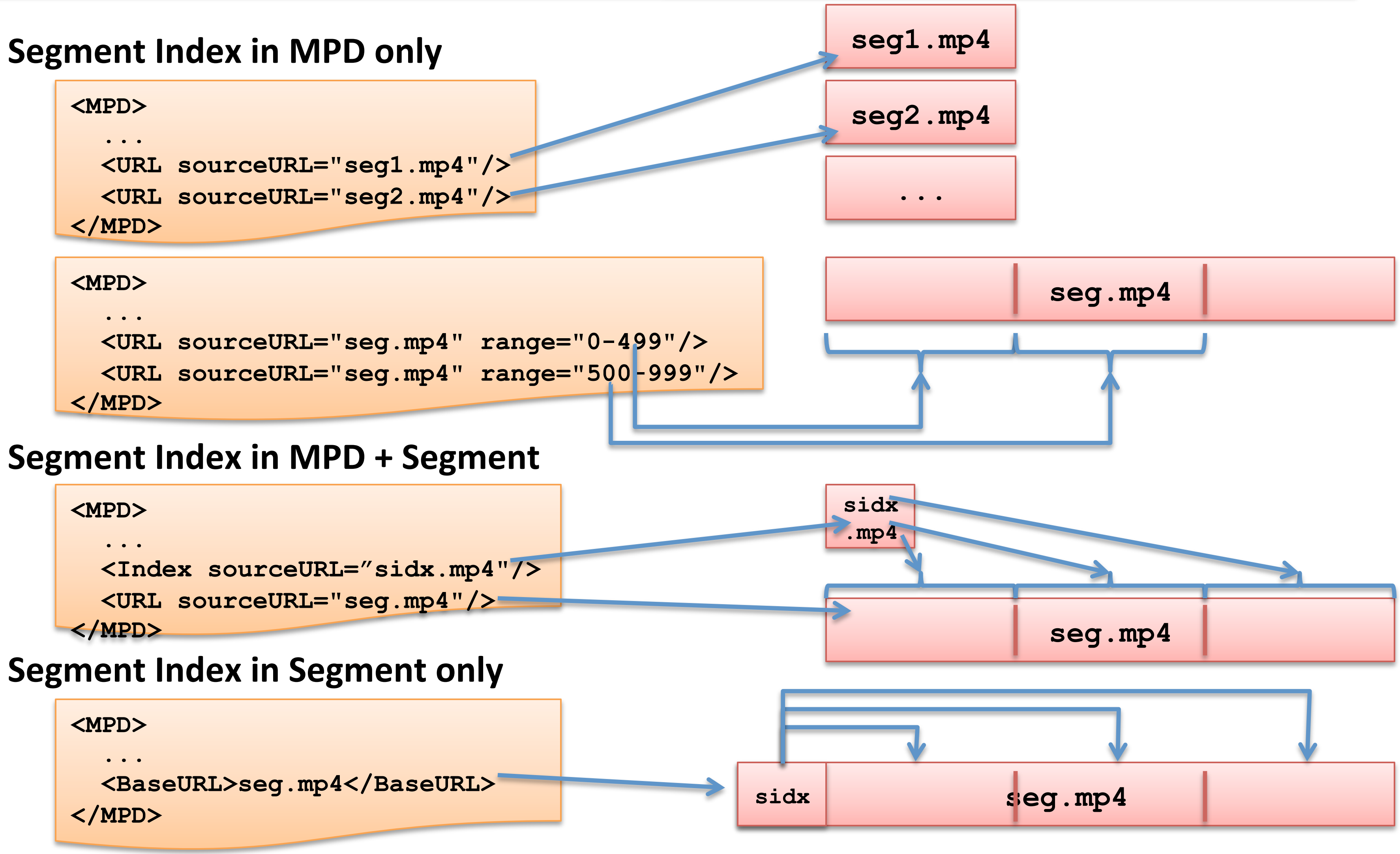
- Mechanism to restrict the combination of *active* Groups
- Expresses the intention of the creator of the Media Presentation

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 Indexing



Segment Index in MPD only

```
<MPD>
  ...
  <URL sourceURL="seg1.mp4"/>
  <URL sourceURL="seg2.mp4"/>
</MPD>
```

```
<MPD>
  ...
  <URL sourceURL="seg.mp4" range="0-499"/>
  <URL sourceURL="seg.mp4" range="500-999"/>
</MPD>
```

Segment Index in MPD + Segment

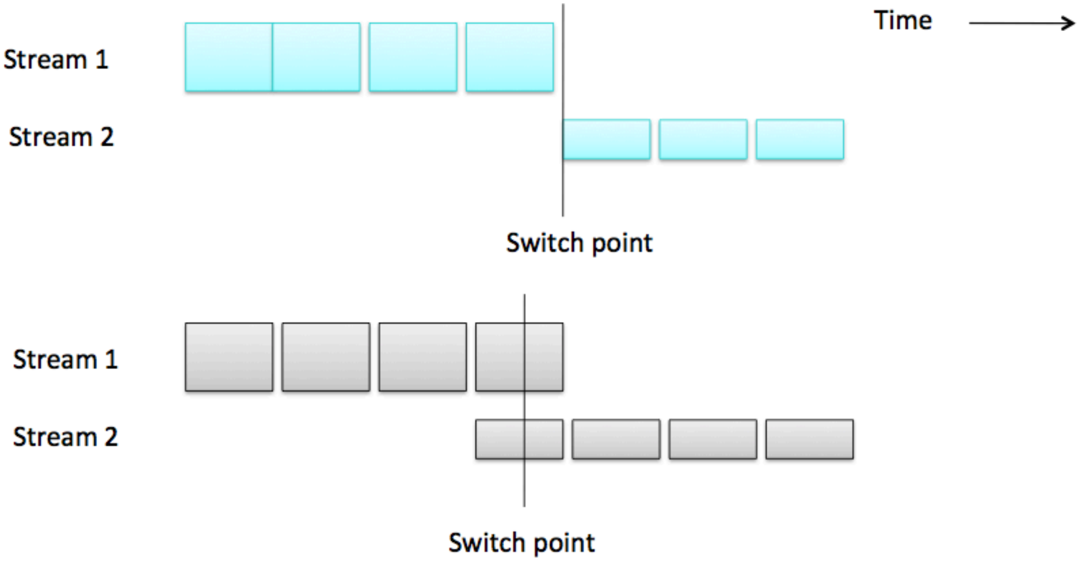
```
<MPD>
  ...
  <Index sourceURL=”sidx.mp4"/>
  <URL sourceURL="seg.mp4"/>
</MPD>
```

Segment Index in Segment only

```
<MPD>
  ...
  <BaseURL>seg.mp4</BaseURL>
</MPD>
```

Switch Point Alignment

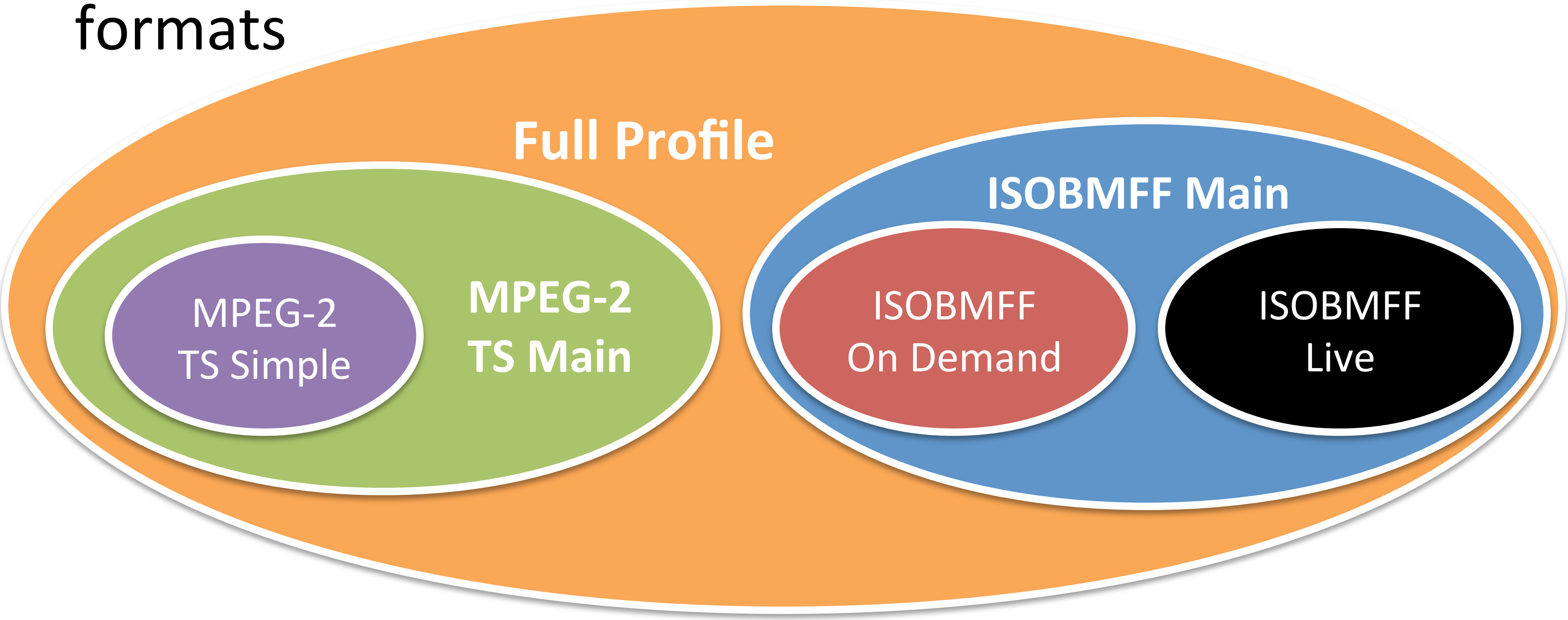
- Segment alignment
  - Permits non-overlapping decoding and presentation of segments from different representations
- Stream Access Points (SAPs)
  - Presentation time and position in segments at which random access and switching can occur
- Bitstream Switching
  - Concatenation of segments from different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



Ack & ©: Mark Watson and Thomas Stockhammer

Profiles

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

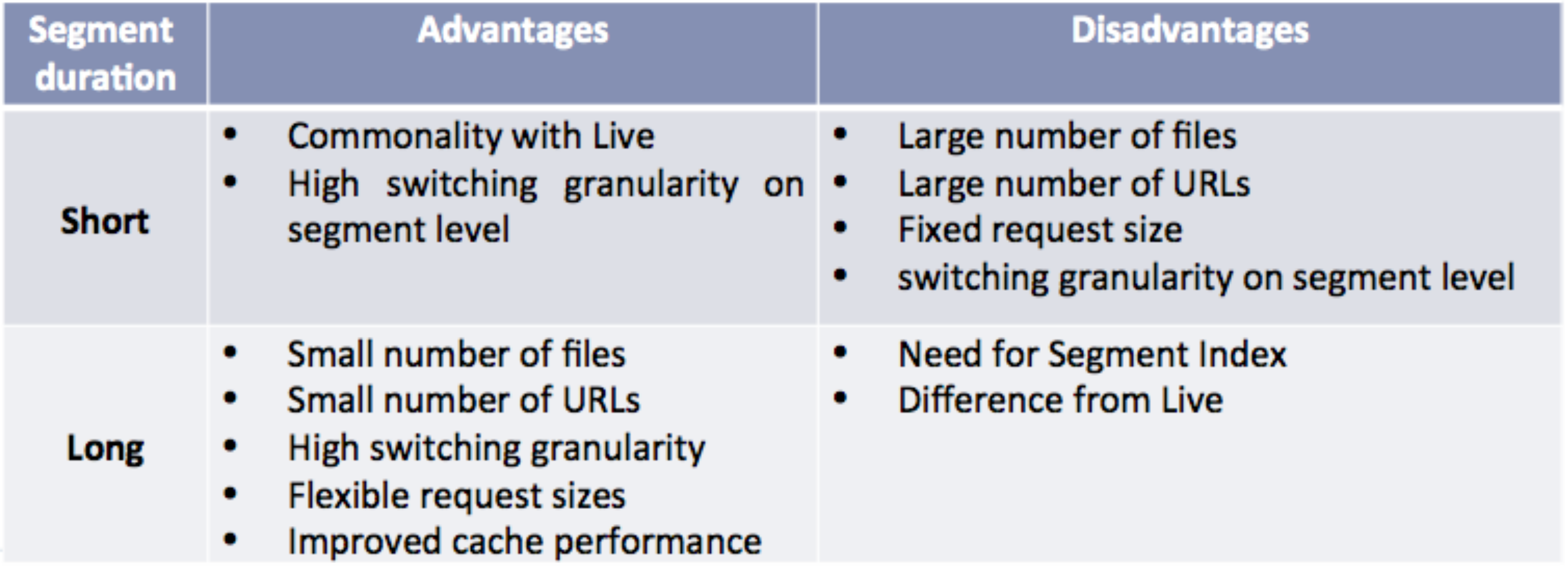


- More restrictions may be added

Ack: Thomas Stockhammer

Adaptive Streaming Summary

- For on demand
  - Chunks are unnecessary and costly
  - Byte range requests have caching and flexibility advantages
  - Separate audio/video essential for language support
- For live
  - Chunks are unavoidable
  - Still value in decoupling request size from chunk size
  - Multiple language audio tracks are rare
  - May need manifest updates
- For both
  - Switch point alignment required for most CE decoding pipelines

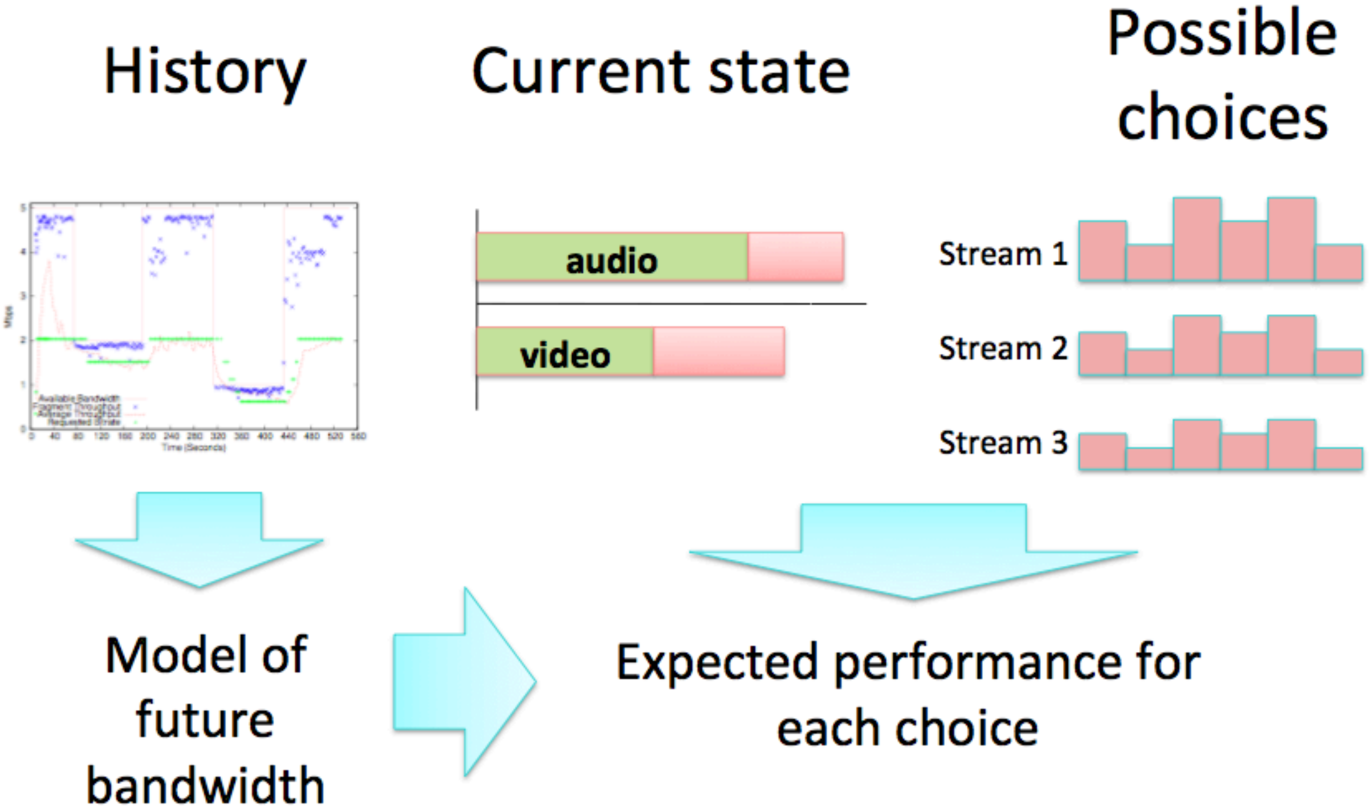


Segment duration	Advantages	Disadvantages
Short	• Commonality with Live • High switching granularity on segment level	• Large number of files • Large number of URLs • Fixed request size • switching granularity on segment level
Long	• Small number of files • Small number of URLs • High switching granularity • Flexible request sizes • Improved cache performance	• Need for Segment Index • Difference from Live

Ack & ©: Mark Watson and Thomas Stockhammer

Adaptation Problem

Choose sequence and timing of requests to minimize probability of re-buffers and maximize quality



History

Current state

Possible choices

Model of future bandwidth

Expected performance for each choice

Ack & ©: Mark Watson

Potential Future Work Items

- MMSys’11 Keynote
  - HTTP Adaptive Streaming in Practice by Mark Watson (Netflix)
  - Future work
    - Good models for future bandwidth
    - Tractable representations of future choices - how to efficiently search the 'choice space'
    - What are the quality goals?
- Call for adaptation logics
  - Efficient implementations of the actual adaptation logic which is responsible for the dynamic and adaptive part of DASH

  http://multimediacommunication.blogspot.com/2011/02/beta-version-of-vlc-dash-plugin.html
- Get it deployed and adopted (e.g. W3C, DVB – what is necessary?)
- Join this activity, everyone is invited – get involved in and exited about DASH!

DASH “Encoder”, Dataset, and Players

Stefan Lederer, Christopher Müller, Benjamin Rainer, and Christian Timmerer

Alpen-Adria-Universität Klagenfurt (AAU) ◆ Faculty of Technical Sciences (TEWI) ◆ Department of Information Technology (ITEC) ◆ Multimedia Communication (MMC) ◆ Sensory Experience Lab (SELab)

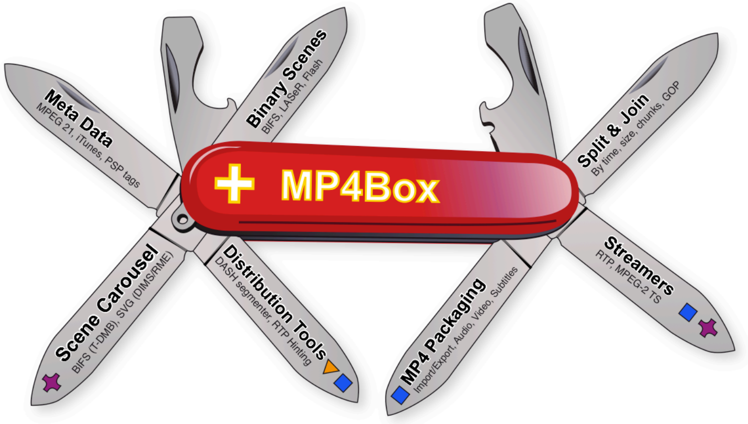
http://research.timmerer.com ◆ http://blog.timmerer.com ◆ http://dash.itec.aau.at/
mailto:christian.timmerer@itec.uni-klu.ac.at

ACM Multimedia 2012, Nara, Japan
29th October, 2012

Acknowledgments. This work was supported in part by the European Commission in the context of the ALICANTE project (FP7-ICT-248652), SocialSensor (FP7-ICT-287975), and the COST Action IC1003 QUALINET.

DASH@GPAC: MP4Box & MP42TS

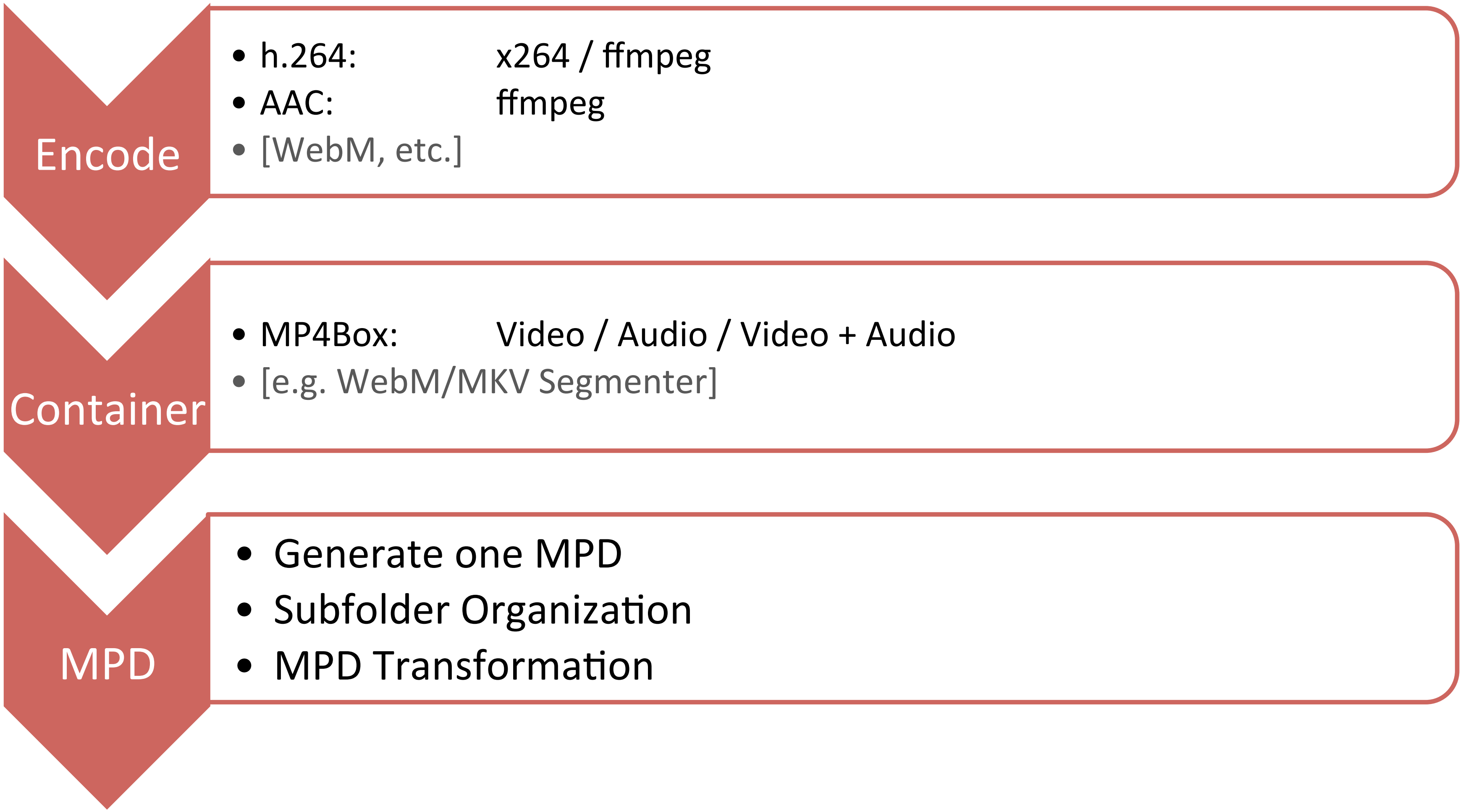
- **Multimedia Packagers**
  - MPEG-2 TS for DASH profiles
  - ISOBMFF Packager & Analyser
- **DASH Segmenter**
  - ISOBMFF and M2TS segments
    - All DASH profiles supported
    - URL-template naming scheme
  - Segment indexing (*SIDX*)
  - GOP-align segments or fragments (*MediaSourceExtension*)
  - Automatic *AdaptationSet* selection
    - Media type, codec, language, PAR
    - Handle groups (same media but not switchable)
- **DASH live simulator**
  - Manages MPD update and timeline continuity


MP4Box
Meta Data
MPEG 21, iTunes, PSP tags
Binary Scenes
BIFS, LASeR, Flash
Split & Join
By time, size, chunks, GOP
Scene Carousel
BIFS (T-DMB), SVG (DIMS/RME)
Distribution Tools
DASH segmenter, RTP Hinting
MP4 Packaging
Import/Export, Audio, Video, Subtitles
Streamers
RTP, MPEG-2 TS

DASHEncoder

- DASH Content Generation Tool
  - Encoding + Multiplexing + MPD generation
  - Generates isoffmain profile compliant MPDs
  - Fully configurable using a config-file
  - Enables batch processing
  - Currently uses x264 and GPAC‘s MP4Box
  - Easy extensible to further encoders & multiplexers
  - http://dash.itec.aau.at/

DASHEncoder



Encode

- h.264: x264 / ffmpeg
- AAC: ffmpeg
- [WebM, etc.]

Container

- MP4Box: Video / Audio / Video + Audio
- [e.g. WebM/MKV Segmenter]

MPD

- Generate one MPD
- Subfolder Organization
- MPD Transformation

Dataset

- **Dataset with DASH Content**
  - Long sequences in high quality
  - Various segment-length versions
  - Free available for DASH experiments
  - PSNR values per frame
- **Problem: Content Rights**
  - CC-Attribution 2.0 Generic (**CC-BY 2.0**) License or similar
  - Free to Share, Free to Remix
  - **Note**: YouTube introduces CC-BY in June 2011!
- **Negotiation with content owner**

Dataset Sequences

Name	Source Quality	Length	Genre
Big Buck Bunny	1080p YUV	09:46	Animation
Elephants Dream	1080p YUV	10:54	Animation
Red Bull Playstreets	1080p, 6 Mbit H.264	01:37:28	Sport
The Swiss Account	1080p, 6 Mbit H.264	57:34	Sport
Valkaama	1080p, 6 Mbit H.264	01:33:05	Movie
Of Forest and Men	SD	10:53	Movie

DASH Dataset Sequences

Bitrates and Resolutions

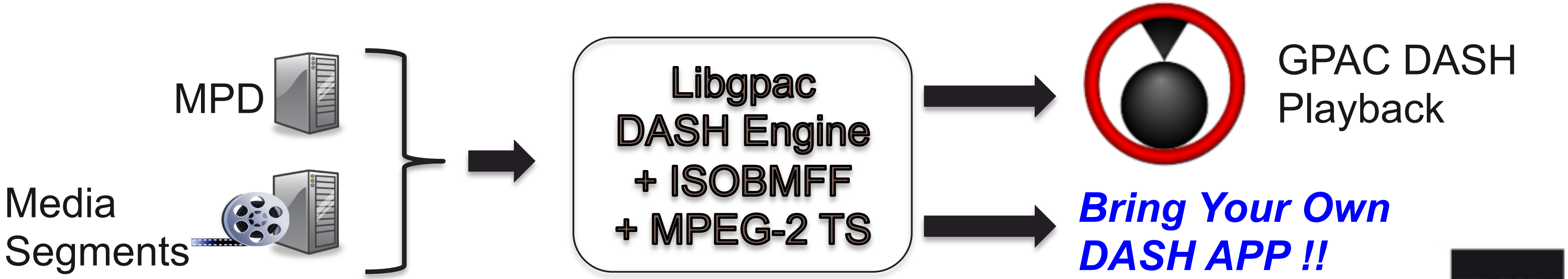
#	Animation	Sport	Movie
1	50 kbit/s, 320x240	100 kbit/s, 320x240	50 kbit/s, 320x240
2	100 kbit/s, 320x240	150 kbit/s, 320x240	100 kbit/s, 320x240
3	150 kbit/s, 320x240	200 kbit/s, 480x360	150 kbit/s, 320x240
4	200 kbit/s, 480x360	250 kbit/s, 480x360	200 kbit/s, 480x360
5	250 kbit/s, 480x360	300 kbit/s, 480x360	250 kbit/s, 480x360
6	300 kbit/s, 480x360	400 kbit/s, 480x360	300 kbit/s, 480x360
7	400 kbit/s, 480x360	500 kbit/s, 854x480	400 kbit/s, 480x360
8	500 kbit/s, 480x360	700 kbit/s, 854x480	500 kbit/s, 854x480
9	600 kbit/s, 854x480	900 kbit/s, 854x480	600 kbit/s, 854x480
10	700 kbit/s, 854x480	1,2 Mbit/s, 854x480	700 kbit/s, 854x480
11	900 kbit/s,1280x720	1,5 Mbit/s,1280x720	900 kbit/s,1280x720
12	1,2 Mbit/s,1280x720	2,0 Mbit/s,1280x720	1,2 Mbit/s,1280x720
13	1,5 Mbit/s,1280x720	2,5 Mbit/s,1280x720	1,5 Mbit/s,1280x720
14	2,0 Mbit/s,1280x720	3,0 Mbit/s,1920x1080	2,0 Mbit/s,1920x1080
15	2,5 Mbit/s,1920x1080	4,0 Mbit/s,1920x1080	2,5 Mbit/s,1920x1080
16	3,0 Mbit/s,1920x1080	5,0 Mbit/s,1920x1080	3,0 Mbit/s,1920x1080
17	4,0 Mbit/s,1920x1080	6,0 Mbit/s,1920x1080	4,0 Mbit/s,1920x1080
18	5,0 Mbit/s,1920x1080		5,0 Mbit/s,1920x1080
19	6,0 Mbit/s,1920x1080		6,0 Mbit/s,1920x1080
20	8,0 Mbit/s,1920x1080		

DASH Content Types

- **Segment Size**:
  - Seconds: 1, 2, 4, 6, 10, 15
- **File Organization**
  - Segmented
  - One file per representation, Byte Range Requests
- **e.g.: Big Buck Bunny**
  - **120 Encodings** needed
  - Only **6 DASH Encoder** runs

DASH@GPAC: Playback

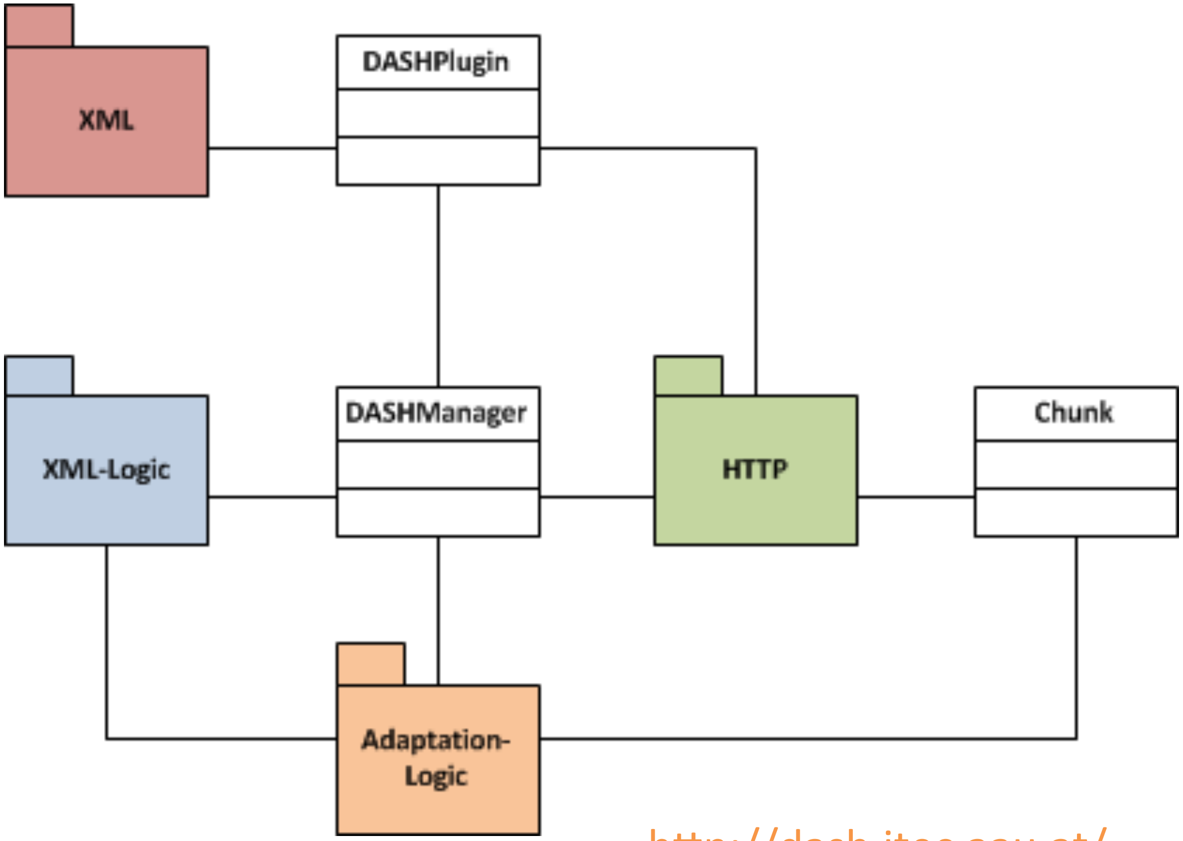
- **DASHClient**
  - DASH ISOBMFF, M2TS (+ HLS)
    - With or without bitstreamSwitching
    - Support for multiple Periods
  - All profiles except *onDemand* (ongoing)
    - VoD through « live » or « main »
  - Local files and http(s) playback
  - Various download policies
- **Integrated in Osmo4**
  - Many input formats and codecs
  - Composition engine (SVG, BIFS, X3D)
- **Try it!**
  - Included in libgpac
  - Independent from player



Bring Your Own DASH APP !!

DASH VLC Plugin Architecture

- Four major components and two controller classes
- Easy Adaptation Logic Interface for Researchers and Developers
- Flexible HTTP structure for further improvements e.g. persistent connections



http://dash.itec.aau.at/

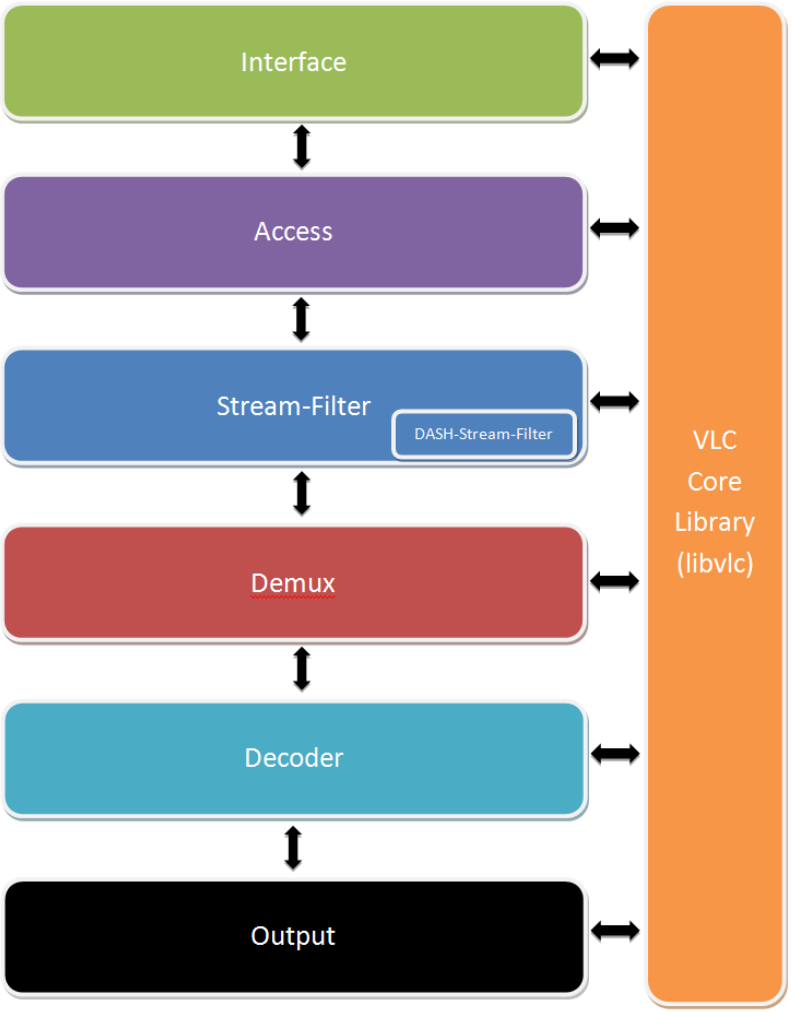
DASH VLC Plugin Features

- Officially part of VLC and as library (libdash)
- Provides a simple interface to integrate new Adaptation Logics
- Dynamic adaptation to the available bandwidth
- Flexible for further improvements, e.g., profiles, persistent connections and pipelining
- Source code is available through the VLC git repository and at:

http://www-itec.aau.at/dash

VLC Architecture

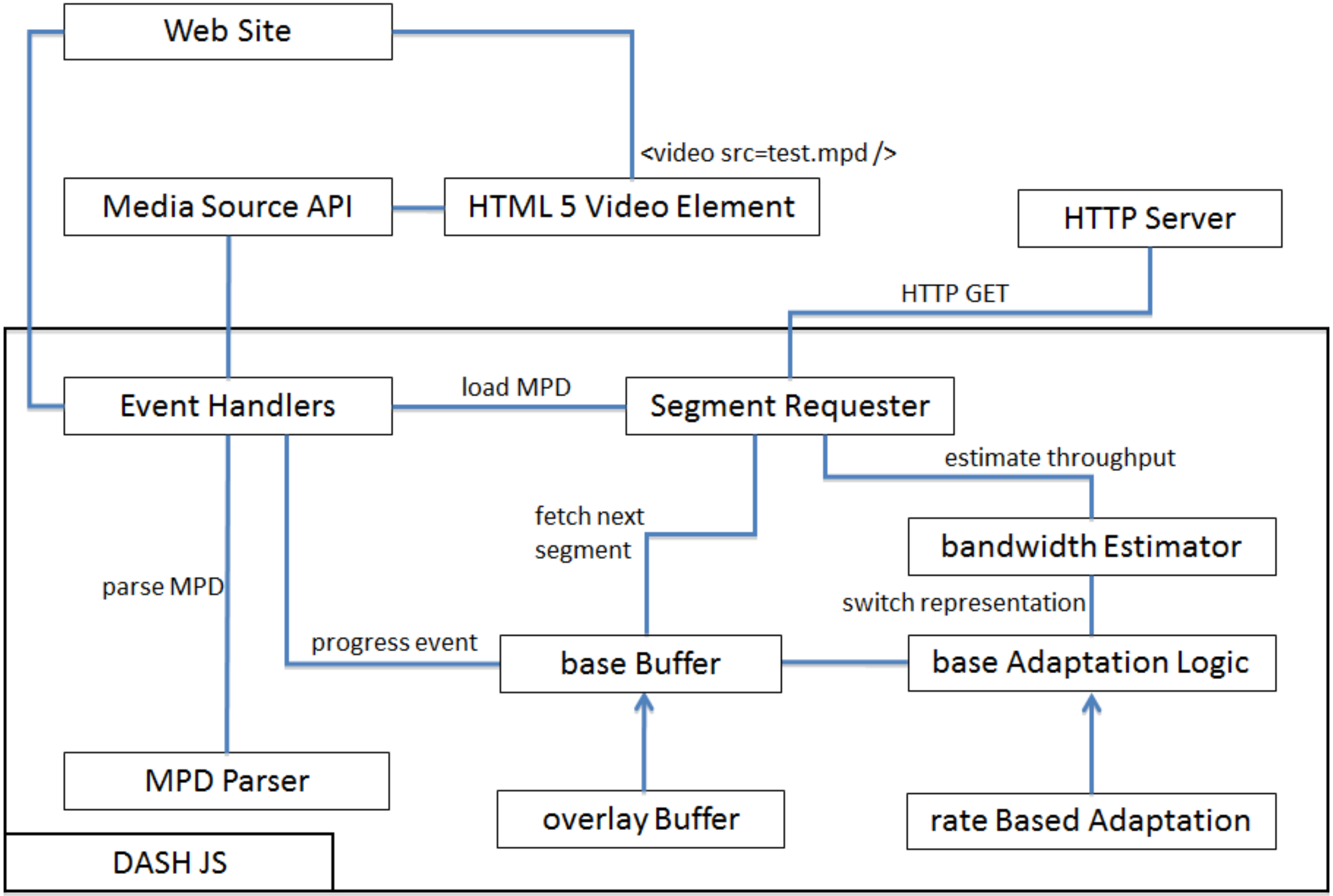
- Interface: User interaction e.g. stop, play etc.
- Access: HTTP, RTP etc.
- Stream-Filter: Recording, Dynamic Streaming
- Demux: MP4, M2TS, MKV
- Decoder: H264, VP8 etc.



DASH in JavaScript (DASH-JS)

- Completely implemented in JavaScript – no (3rd party) plugins required
- Makes use of the Media Source API provided by Google Chrome
  - Support for WebM and ISOBMFF
- Provides time based and byte based buffers
  - E.g., use as input for adaptation logics
- Flexible adaptation logics
  - Easy to extend existing ones or integrate your own

DASH-JS Architecture



DASH-JS (cont'd)

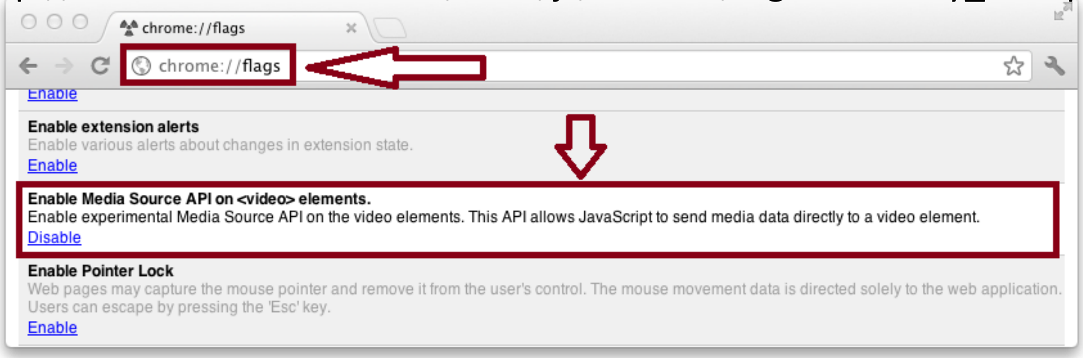
- Bandwidth / throughput estimation
  - … is done each time a segment is retrieved
  - At the beginning the MPD is used to have an educated guess on the bandwidth
  - To bypass proxy caching "no-cache" is set in the HTTP Request Header (will influence the throughput estimation)
- Representation selection is based on:

$$b_n = \frac{w_1 b_{n-1} + w_2 b_m}{w_1 + w_2}$$

  - b_{n-1} denotes the throughput calculated at the n-1th segment
  - b_m depicts the throughput measured with the nth segment
  - b_n is used to decide which representation should be selected
  - The weights (w_1 and w_2) are used to mimic optimistic or pessimistic behavior
- Simple adaptation logic, easy to extend, modify…

Showcases

- Sintel Trailer @ 480p
  - 5 representation from 200 kbps to 2000 kbps video bitrate
  - 128 kbps audio for all representations
  - Showcase: http://www-itec.uni-klu.ac.at/dash/js/dashtest.html
  - MPD: http://www-itec.uni-klu.ac.at/dash/js/content/sintel_multi_rep.mpd
- Big Buck Bunny @ 480p
  - 7 representations from 200 kbps to 4700 kbps video bitrate
  - 128 kbps audio for all representations
  - Showcase: http://www-itec.uni-klu.ac.at/dash/js/dashtest-bunny.html
  - MPD: http://www-itec.uni-klu.ac.at/dash/js/content/bigbuckbunny.mpd
- Big Buck Bunny @ 1080p
  - 7 representations from 1000 kbps to 8000 kbps
  - 128 kbps audio for all representations
  - Showcase: http://www-itec.uni-klu.ac.at/dash/js/dashtest-bunny1080p.html
  - MPD: http://www-itec.uni-klu.ac.at/dash/js/content/bigbuckbunny_1080p.mpd



Enable

Enable extension alerts
Enable various alerts about changes in extension state.
Enable

Enable Media Source API on <video> elements.
Enable experimental Media Source API on the video elements. This API allows JavaScript to send media data directly to a video element.
Disable

Enable Pointer Lock
Web pages may capture the mouse pointer and remove it from the user's control. The mouse movement data is directed solely to the web application. Users can escape by pressing the 'Esc' key.
Enable

Conclusions

- End-to-end DASH tools available
  - GPAC provides support for ISOBMFF, M2TS, and beyond
  - DASH VLC plugin and libdash (world first DASH player)
  - DASH-JS for easy Web integration (HTML5, Javascript)
- Flexible architecture, easy to extend, e.g.:
  - Add your own profile (!!!)
  - Add your own buffer model
  - Add your own bandwidth estimation, adaptation logic
- Open source: http://dash.itec.aau.at | http://gpac.sourceforge.net
  - Feel free to use it, please acknowledge/reference us

An Evaluation of Dynamic Adaptive Streaming over HTTP in Vehicular Environments

Christopher Müller, Stefan Lederer, and Christian Timmerer

Alpen-Adria-Universität Klagenfurt (AAU) ◆ Faculty of Technical Sciences (TEWI) ◆ Department of Information Technology (ITEC) ◆ Multimedia Communication (MMC) ◆ Sensory Experience Lab (SELab)

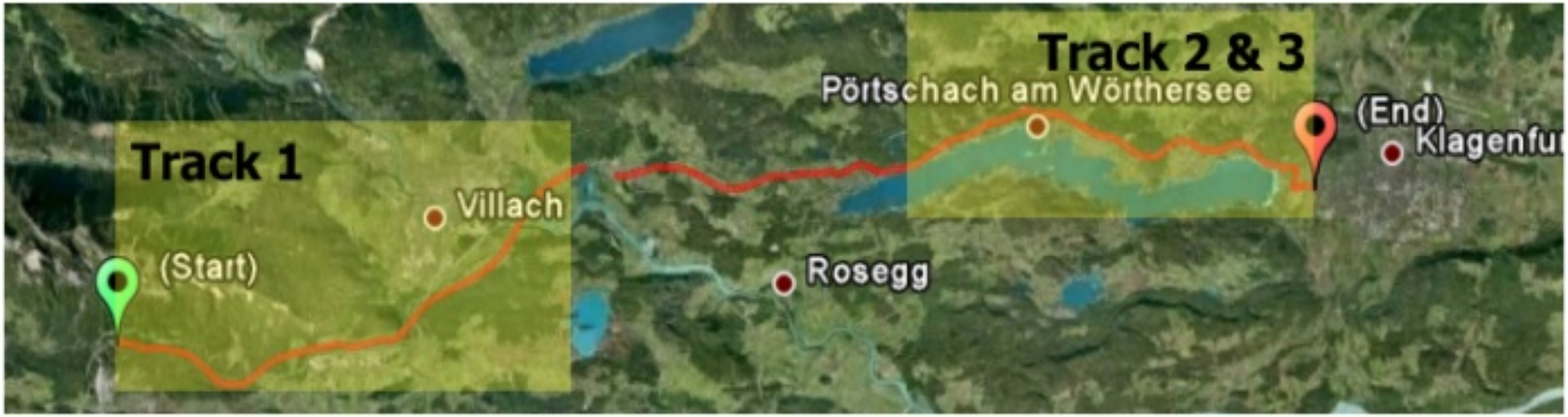
http://research.timmerer.com ◆ http://blog.timmerer.com ◆ http://dash.itec.aau.at/
mailto:christian.timmerer@itec.uni-klu.ac.at

ACM Multimedia 2012
29th October, 2012

Acknowledgments. This work was supported in part by the European Commission in the context of the ALICANTE project (FP7-ICT-248652), SocialSensor (FP7-ICT-287975), and the COST Action IC1003 QUALINET.

Methodology

- **Experiment 1 / Track 1 (601 seconds)**
  - Drive on the freeway A2, passing by the city of Villach in the direction to Klagenfurt.
- **Experiment 2 / Track 2 (575 seconds)**
  - From the Alpen-Adria-Universität Klagenfurt on the freeway A2 until the service area around Techelsberg.
- **Experiment 3 / Track 3 (599 seconds)**
  - From the service area around Techelsberg on the freeway A2 to the exit of Klagenfurt.

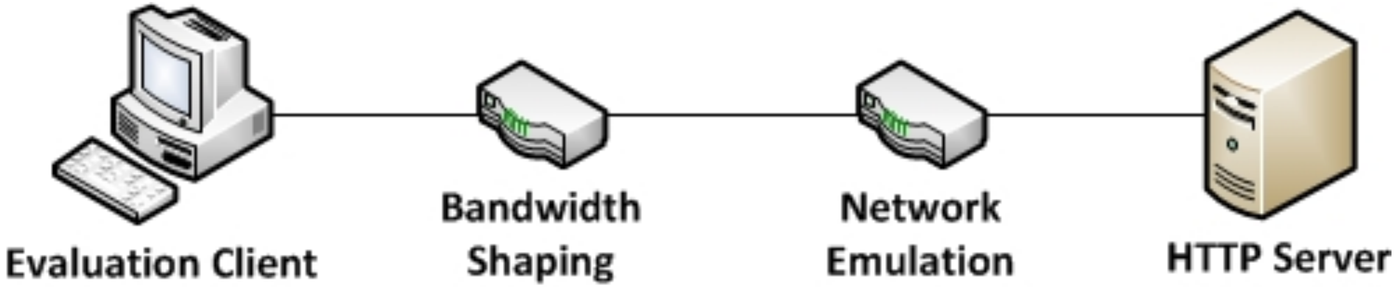


Metrics

- Average bitrate
  - Overall performance for the entire session
- Number of quality switches
  - Different representation due available bandwidth
- Buffer level
  - Estimated with download timestamp (DTS) and presentation timestamp (PTS)
- Number of unsmooth seconds
  - Buffer empty

Experimental Setup

- Bandwidth Shaping
  - Ubuntu 11.04 w/ Linux hierarchical token bucket (htb)
  - Available bandwidth will be adjusted every 2s due to the recorded traces and 2s segment length
- Network Emulation
  - Emulates a round trip time of 150ms
- HTTP Server
  - Server based on Windows Server 2008 and IIS / Ubuntu 11.04 and Apache Web Server
- Evaluation Client
  - Windows or Linux depending on the evaluation system



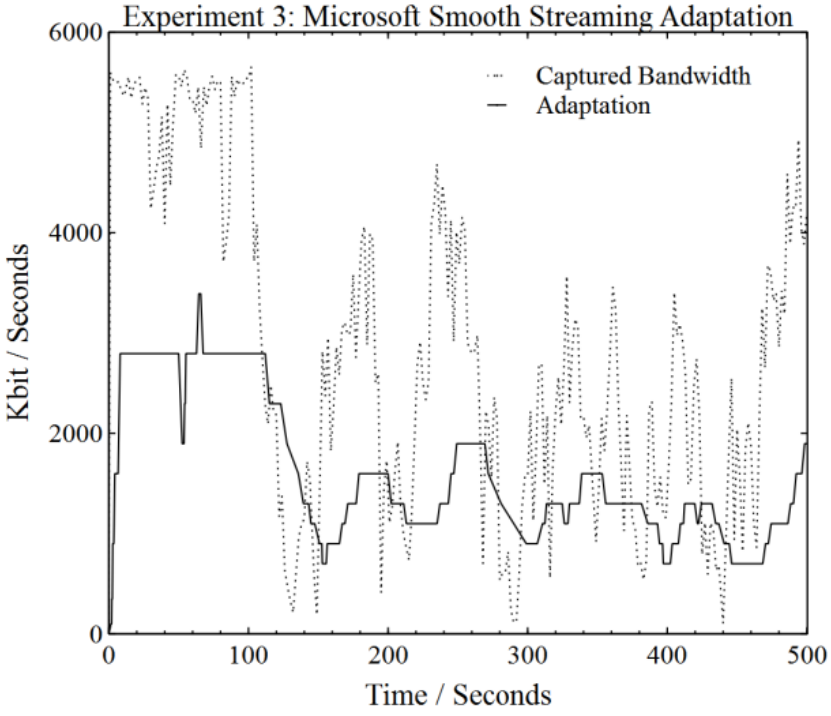
Dataset

- All experiments have been performed with the same content based on [Lederer2012]
- The content has been encoded with x264
- 14 different bitrates from 100kbps to 4500kbps
- Segments with a length of 2 seconds
  - Restricted by Microsoft Smooth Streaming
- That content has been used for all three scenarios

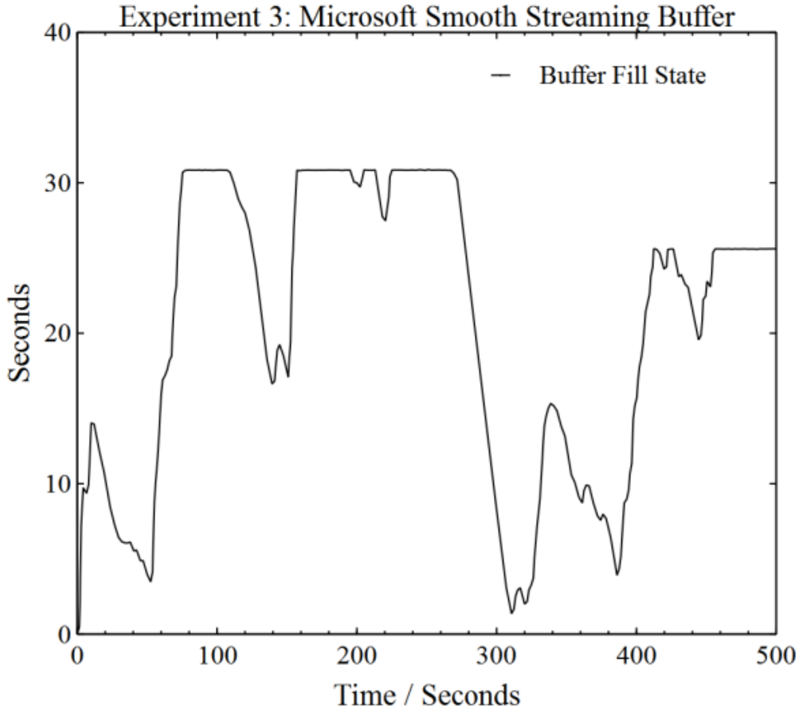
Microsoft Smooth Streaming

- Client based on Windows 7, Microsoft Silverlight and Firefox 7
- Server based on Windows Server 2008 and IIS with Media Services 4.0
- Content has been multiplexed with IIS Transform Manager 1.0 Beta
- PTS has been taken from the request URL
- DTS comes from the bandwidth emulation node

Microsoft Smooth Streaming (cont'd)



(a)



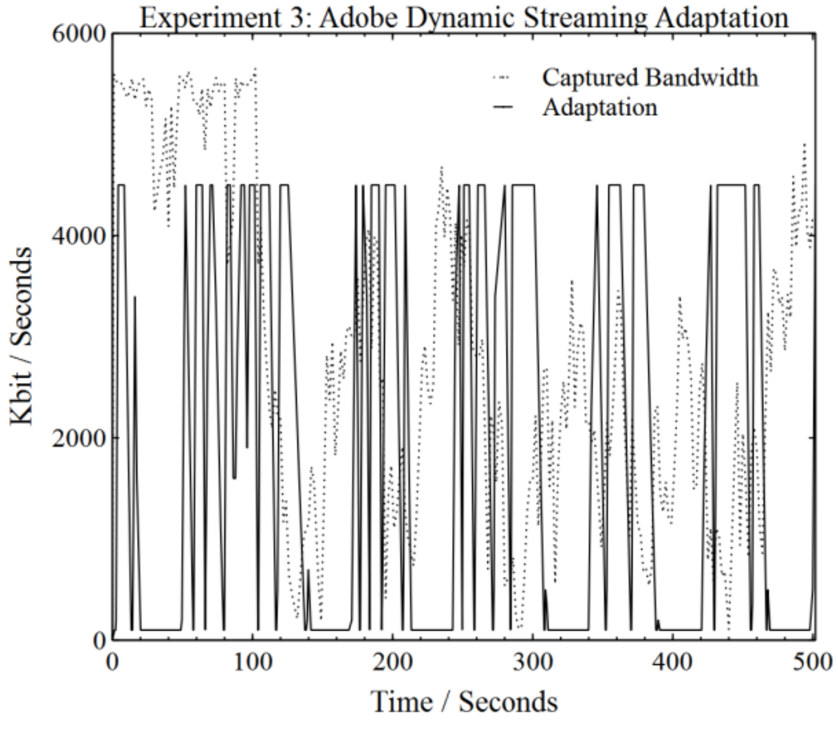
(b)

- Few switches with a good average bitrate
- Nevertheless close to unsmoothness at second 300

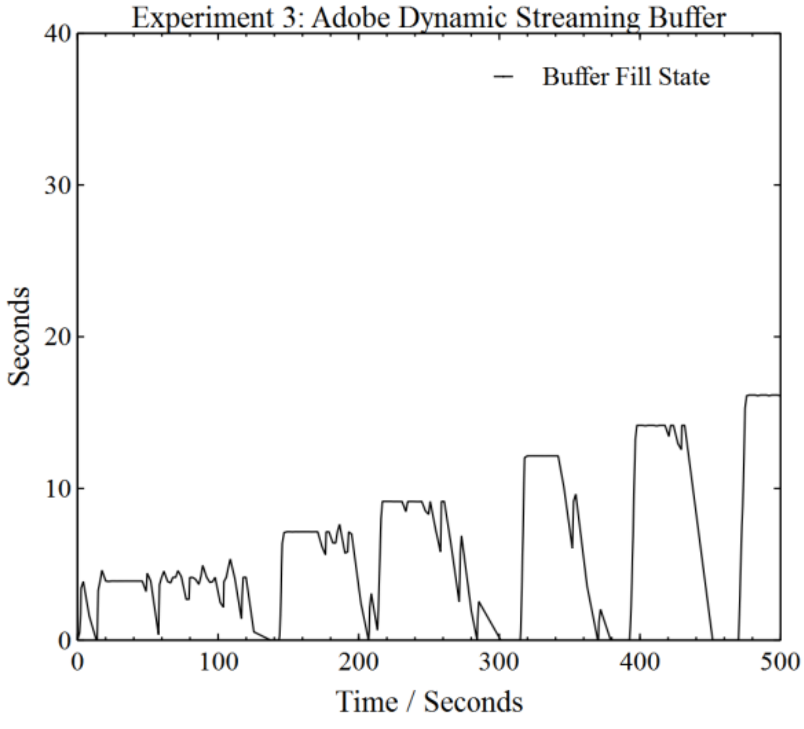
Adobe Dynamic HTTP Streaming

- Client is based on Ubuntu 11.04, Firefox 7 and the Open Source Media Framework player
- The server component hosts the Flash Media Server in development edition
- The content has been generated with the Adobe File Packager for Adobe Dynamic Streaming

Adobe Dynamic HTTP Streaming (cont'd)



(a)



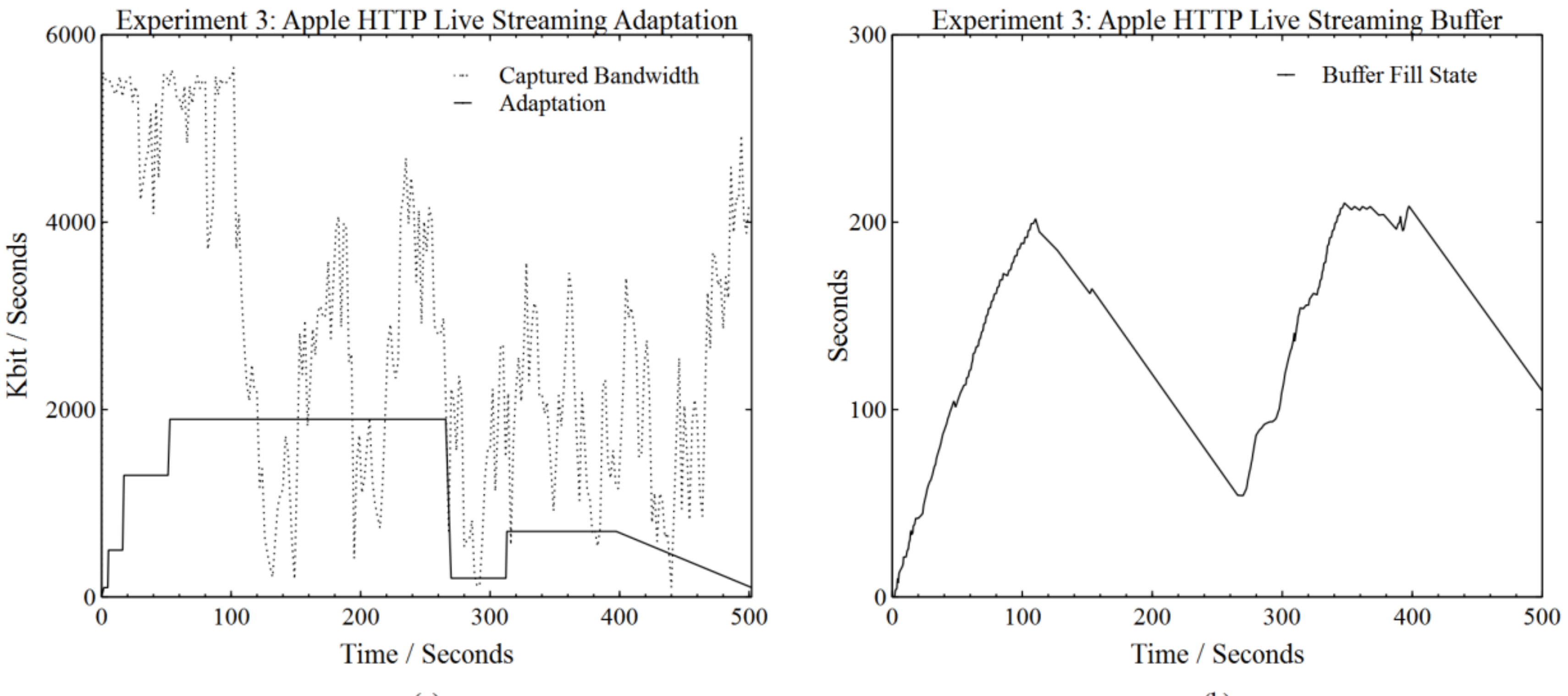
(b)

- High number of unsmooth seconds
- Rather binary and unpredictable

Apple HTTP Live Streaming

- Client is based on Mac OS X Snow Leopard 10.6 and Safari 5
- Content has been generated with Microsoft Transform Manager
  - Transmultiplexing of mp4 to MPEG-2 TS
  - Chops the transport stream into segments of 2 seconds length
- The only system that uses MPEG-2 TS

Apple HTTP Live Streaming (cont'd)



(a)

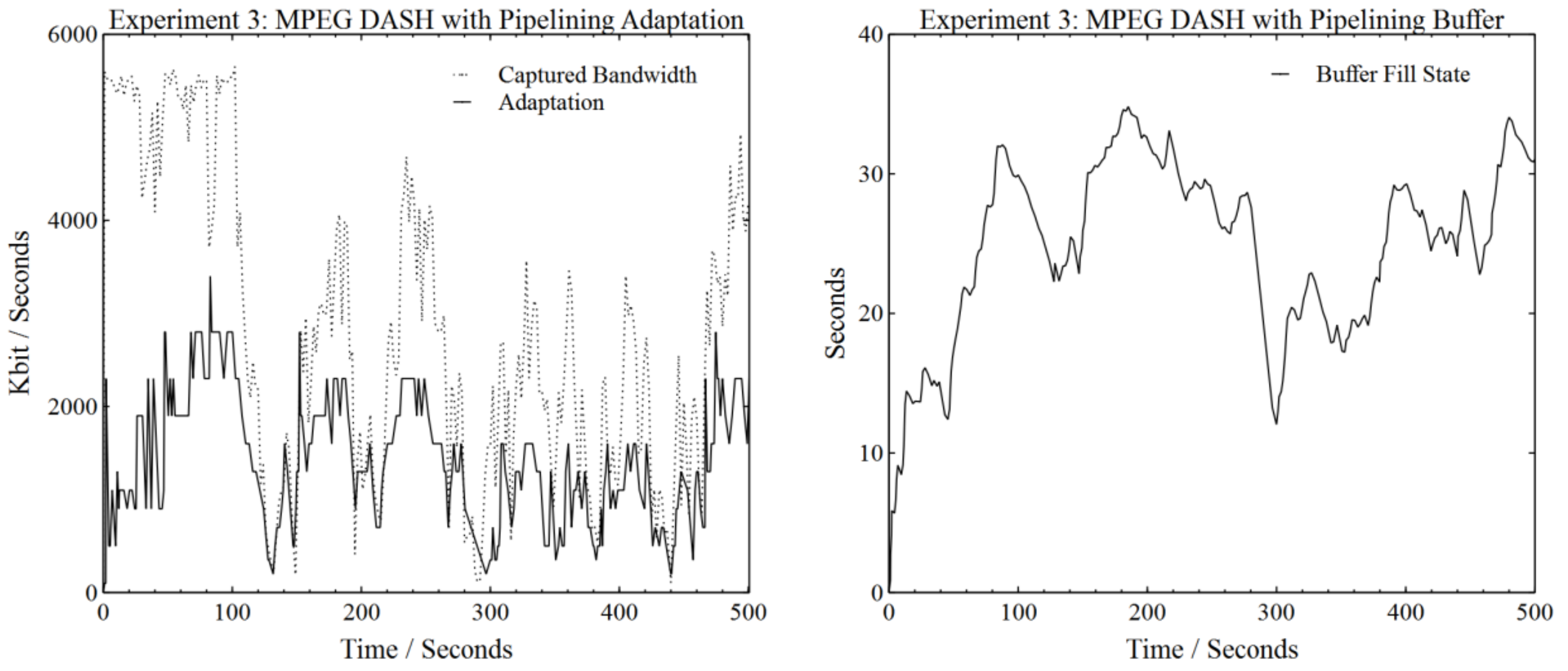
(b)

- Very few switches with a lower bitrate
- Large buffer for energy awareness

Our MPEG-DASH Implementation

- Client: DASH VLC Plugin [Mueller2011] on Ubuntu 11.04
- Server: Ubuntu 11.04 which hosts an Apache Web server
- Content based on DASH dataset generated with DASHEncoder
- Simple (naïve) adaptation logic

Our MPEG-DASH Implementation (cont'd)

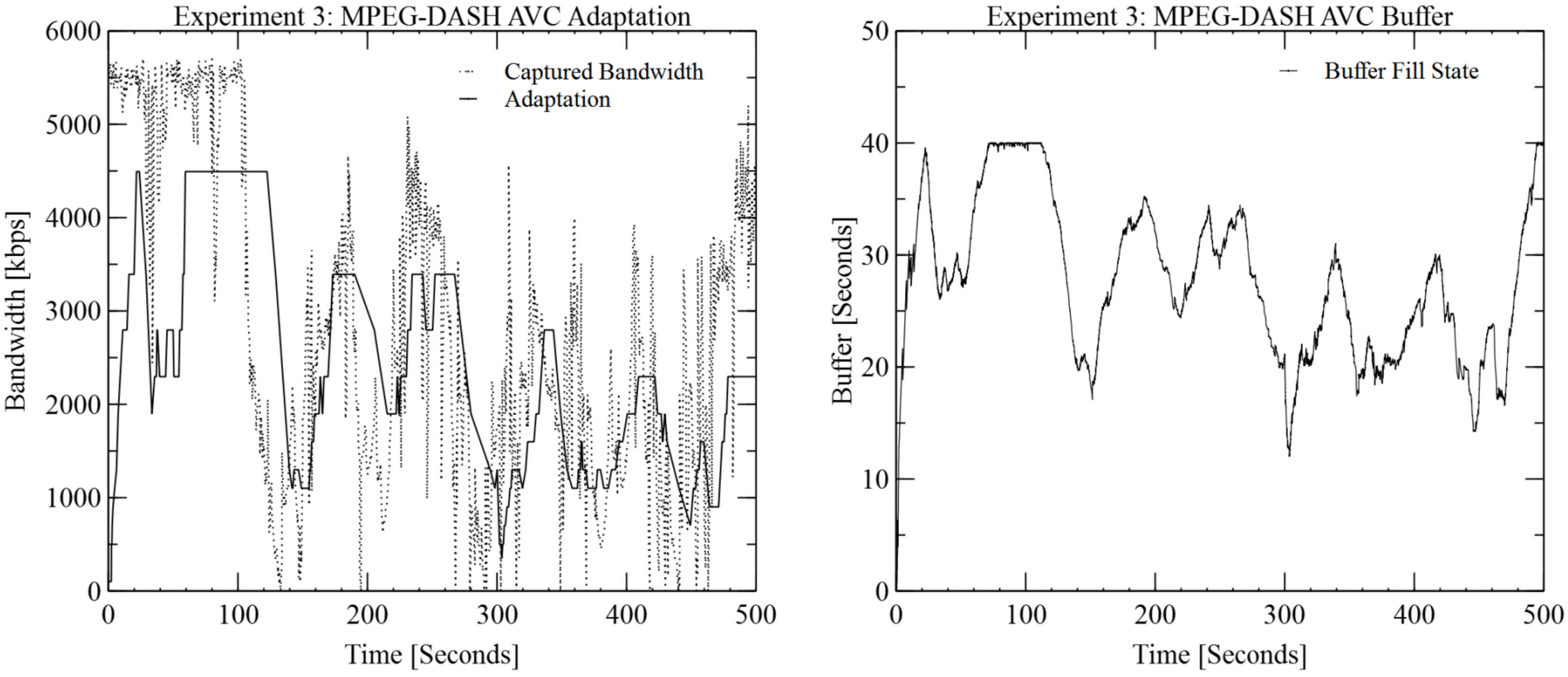


(a)

(b)

- Non stepwise switching
- Good average bitrate and stable buffer

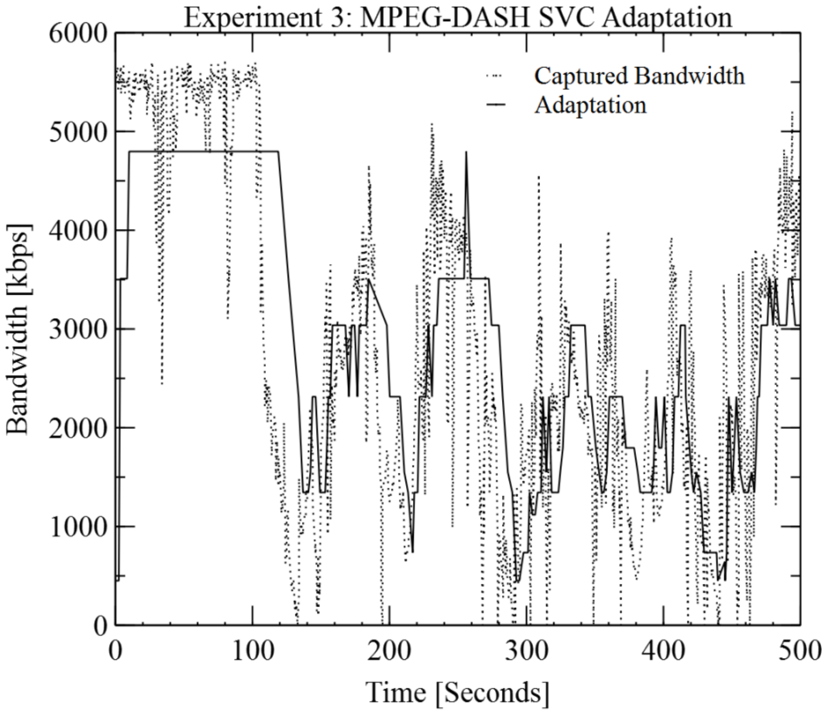
MPEG-DASH AVC



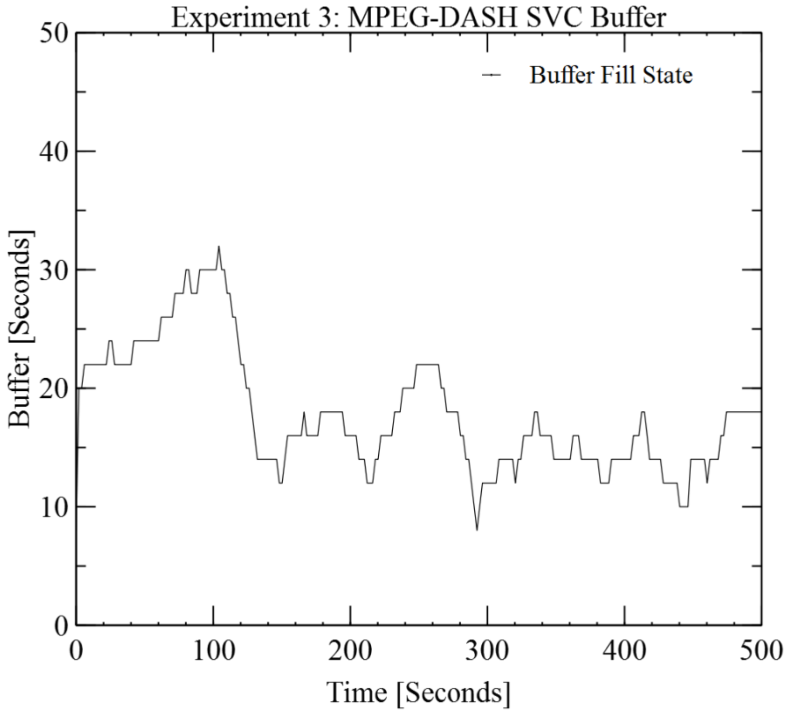
(a)

(b)

MPEG-DASH SVC



(a)



(b)

Summary

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

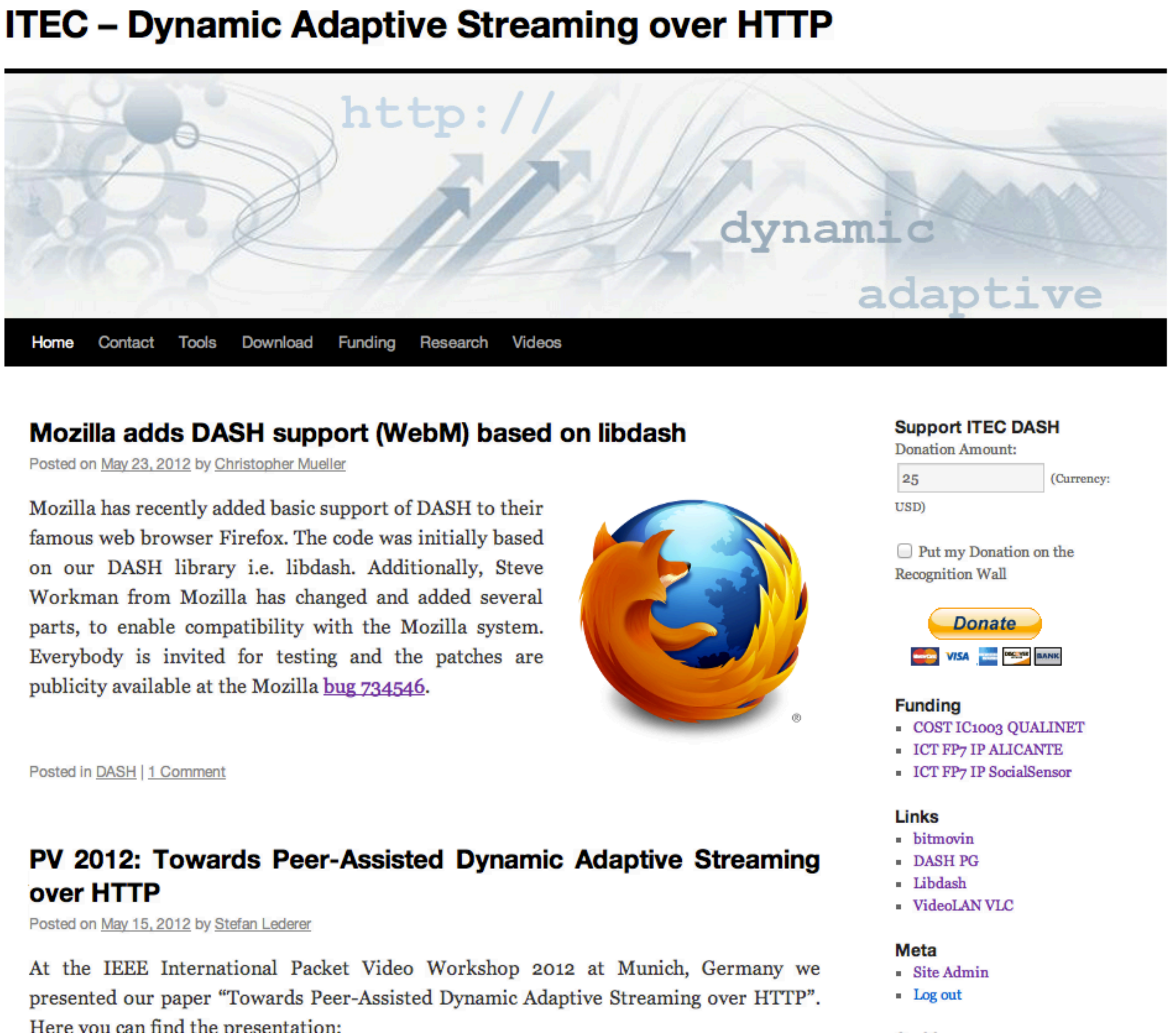
* … MoVid/MMSys, February 2012

** … EUSIPCO, August 2012

Conclusions

- Microsoft Smooth Streaming
  - Performs very well w.r.t. average bitrate
  - Yes, deserves the name smooth streaming
- Apple HLS
  - Less quality switches due to large buffer
  - Designed for mobile devices, energy awareness
- Adobe HDS
  - Binary decision between the highest and the lowest representation
  - Stalls, re-buffering => low QoE
- Our MPEG-DASH implementation
  - Achieves a good/superior average bitrate
  - In striking distance to the top, space for improvements though (e.g., buffer management)
- Disclaimer: comparison of specific client implementations, not formats (manifest/segment), not technology

DASH @ AAU/ITEC



ITEC – Dynamic Adaptive Streaming over HTTP

Mozilla adds DASH support (WebM) based on libdash

Posted on May 23, 2012 by Christopher Mueller

Mozilla has recently added basic support of DASH to their famous web browser Firefox. The code was initially based on our DASH library i.e. libdash. Additionally, Steve Workman from Mozilla has changed and added several parts, to enable compatibility with the Mozilla system. Everybody is invited for testing and the patches are publicity available at the Mozilla bug 734546.

Posted in DASH | 1 Comment

PV 2012: Towards Peer-Assisted Dynamic Adaptive Streaming over HTTP

Posted on May 15, 2012 by Stefan Lederer

At the IEEE International Packet Video Workshop 2012 at Munich, Germany we presented our paper "Towards Peer-Assisted Dynamic Adaptive Streaming over HTTP". Here you can find the presentation:

Support ITEC DASH

Donation Amount:

25 (Currency: USD)

□ Put my Donation on the Recognition Wall

Funding

- COST IC1003 QUALINET
- ICT FP7 IP ALICANTE
- ICT FP7 IP SocialSensor

Links

- bitmovin
- DASH PG
- Libdash
- VideoLAN VLC

Meta

- Site Admin
- Log out

http://dash.itec.aau.at/

DASH VLC Plugin

DASHEncoder

libdash

Dataset

DASH-JS

Join this activity, everyone is invited – get involved in and exited about DASH!

Acknowledgments

- EC projects for partially funding this activity
  - ALICANTE project (FP7-ICT-248652)
    - http://www.ict-alicante.eu
  - SocialSensor project (FP7-ICT-287975)
    - http://www.socialsensor.org
  - COST ICT Action IC1003
    - QUALINET – European Network on Quality of Experience in Multimedia Systems and Services
    - http://www.qualinet.eu/
- DASH Industry Forum
  - http://www.dashif.com
- Christopher Müller: VLC Plugin, libdash
- Stefan Lederer: DASHEncoder, dataset, DASH-JS
- Benjamin Rainer: DASH-JS
- Hermann Hellwagner for his advice and feedback
- ISO/IEC MPEG and its participating members for their constructive feedback during the standardization process

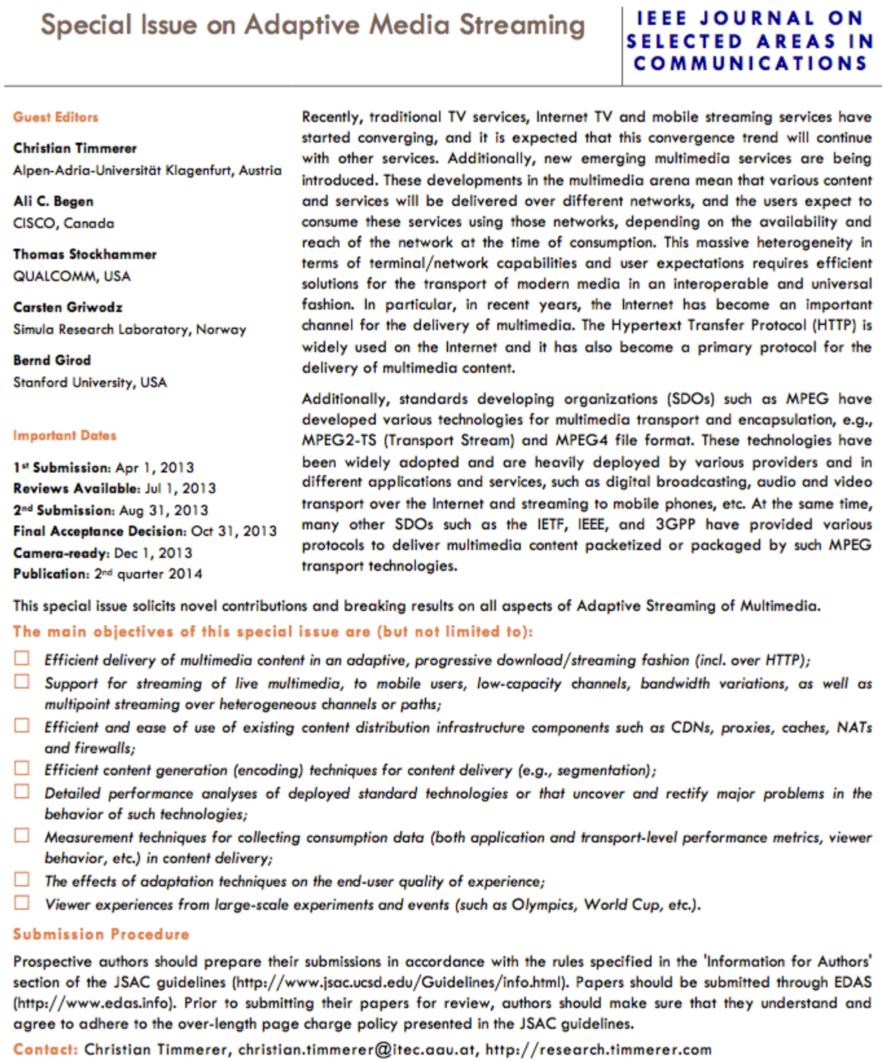
References

- Christopher Müller, Daniele Renzi, Stefan Lederer, Stefano Battista and Christian Timmerer, "Using Scalable Video Coding for Dynamic Adaptive Streaming Over HTTP in Mobile Environments", In Proceedings of the 20th European Signal Processing Conference 2012, Bucharest, Romania, August 27-31, 2012.
- Benjamin Rainer, Stefan Lederer, Christopher Müller and Christian Timmerer, "A Seamless Web Integration of Adaptive HTTP Streaming", In Proceedings of the 20th European Signal Processing Conference 2012, Bucharest, Romania, August 27-31, 2012.
- Stefan Lederer, Christopher Müller and Christian Timmerer, "Peer-Assisted Dynamic Adaptive Streaming over HTTP – System Design and Evaluation", In Proceedings of the IEEE International Packet Video Workshop 2012, Munich, Germany, May 10-11, 2012.
- Christopher Müller, Stefan Lederer and Christian Timmerer, "An Evaluation of Dynamic Adaptive Streaming over HTTP in Vehicular Environments", In Proceedings of the 4th ACM Workshop on Mobile Video, Chapel Hill, North Carolina, February 24, 2012.
- Stefan Lederer, Christopher Müller and Christian Timmerer, "Dynamic Adaptive Streaming over HTTP Dataset", In Proceedings of the ACM Multimedia Systems Conference 2012, Chapel Hill, North Carolina, February 22-24, 2012.
- Christopher Müller and Christian Timmerer, "A VLC Media Player Plugin enabling Dynamic Adaptive Streaming over HTTP", In Proceedings of the ACM Multimedia 2011, Scottsdale, Arizona, November 28, 2011.
- Christopher Müller and Christian Timmerer, "A Test-Bed for the Dynamic Adaptive Streaming over HTTP featuring Session Mobility", In Proceedings of the ACM Multimedia Systems Conference 2011, San Jose, California, February 23-25, 2011.
- Christian Timmerer and Christopher Müller, "HTTP Streaming of MPEG Media", In Proceedings of the Streaming Day 2010, Udine, Italy, September 16-17, 2010.

IEEE JSAC: Adaptive Media Streaming

- Guest Editors
  - Christian Timmerer, Alpen-Adria-Universität Klagenfurt, Austria
  - Ali C. Begen, CISCO, Canada
  - Thomas Stockhammer, QUALCOMM, USA
  - Carsten Griwodz, Simula Research Laboratory, Norway
  - Bernd Girod, Stanford University, USA
- Important Dates
  - 1st Submission: Apr 1, 2013
  - Reviews Available: Jul 1, 2013
  - 2nd Submission: Aug 31, 2013
  - Final Acceptance Decision: Oct 31, 2013
  - Camera-ready: Dec 1, 2013
  - Publication: 2nd quarter 2014

http://multimediacommunication.blogspot.com/2012/07/jsac-special-issue-adaptive-media.html



Special Issue on Adaptive Media Streaming

IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS

Guest Editors

Christian Timmerer
Alpen-Adria-Universität Klagenfurt, Austria

Ali C. Begen
CISCO, Canada

Thomas Stockhammer
QUALCOMM, USA

Carsten Griwodz
Simula Research Laboratory, Norway

Bernd Girod
Stanford University, USA

Important Dates

1st Submission: Apr 1, 2013
Reviews Available: Jul 1, 2013
2nd Submission: Aug 31, 2013
Final Acceptance Decision: Oct 31, 2013
Camera-ready: Dec 1, 2013
Publication: 2nd quarter 2014

Recently, traditional TV services, Internet TV and mobile streaming services have started converging, and it is expected that this convergence trend will continue with other services. Additionally, new emerging multimedia services are being introduced. These developments in the multimedia arena mean that various content and services will be delivered over different networks, and the users expect to consume these services using those networks, depending on the availability and reach of the network at the time of consumption. This massive heterogeneity in terms of terminal/network capabilities and user expectations requires efficient solutions for the transport of modern media in an interoperable and universal fashion. In particular, in recent years, the Internet has become an important channel for the delivery of multimedia. The Hypertext Transfer Protocol (HTTP) is widely used on the Internet and it has also become a primary protocol for the delivery of multimedia content.

Additionally, standards developing organizations (SDOs) such as MPEG have developed various technologies for multimedia transport and encapsulation, e.g., MPEG2-TS (Transport Stream) and MPEG4 file format. These technologies have been widely adopted and are heavily deployed by various providers and in different applications and services, such as digital broadcasting, audio and video transport over the Internet and streaming to mobile phones, etc. At the same time, many other SDOs such as the IETF, IEEE, and 3GPP have provided various protocols to deliver multimedia content packetized or packaged by such MPEG transport technologies.

This special issue solicits novel contributions and breaking results on all aspects of Adaptive Streaming of Multimedia.

The main objectives of this special issue are (but not limited to):

- ☐ *Efficient delivery of multimedia content in an adaptive, progressive download/streaming fashion (incl. over HTTP);*
- ☐ *Support for streaming of live multimedia, to mobile users, low-capacity channels, bandwidth variations, as well as multipoint streaming over heterogeneous channels or paths;*
- ☐ *Efficient and ease of use of existing content distribution infrastructure components such as CDNs, proxies, caches, NATs and firewalls;*
- ☐ *Efficient content generation (encoding) techniques for content delivery (e.g., segmentation);*
- ☐ *Detailed performance analyses of deployed standard technologies or that uncover and rectify major problems in the behavior of such technologies;*
- ☐ *Measurement techniques for collecting consumption data (both application and transport-level performance metrics, viewer behavior, etc.) in content delivery;*
- ☐ *The effects of adaptation techniques on the end-user quality of experience;*
- ☐ *Viewer experiences from large-scale experiments and events (such as Olympics, World Cup, etc.).*

Submission Procedure

Prospective authors should prepare their submissions in accordance with the rules specified in the 'Information for Authors' section of the JSAC guidelines (http://www.jsac.ucsd.edu/Guidelines/info.html). Papers should be submitted through EDAS (http://www.edas.info). Prior to submitting their papers for review, authors should make sure that they understand and agree to adhere to the over-length page charge policy presented in the JSAC guidelines.

Contact: Christian Timmerer, christian.timmerer@itec.aau.at, http://research.timmerer.com

Thank you for your attention

... questions, comments, etc. are welcome …

Ass.-Prof. Dipl.-Ing. Dr. Christian Timmerer
Klagenfurt University, Department of Information Technology (ITEC)
Universitätsstrasse 65-67, A-9020 Klagenfurt, AUSTRIA
christian.timmerer@itec.uni-klu.ac.at
http://research.timmerer.com/
Tel: +43/463/2700 3621 Fax: +43/463/2700 3699

A Comparison of Quality Scheduling in Commercial Adaptive HTTP Streaming Solutions on a 3G Network

Haakon Riiser, Håkon S. Bergsaker, Paul Vigmostad,
Pål Halvorsen, Carsten Griwodz

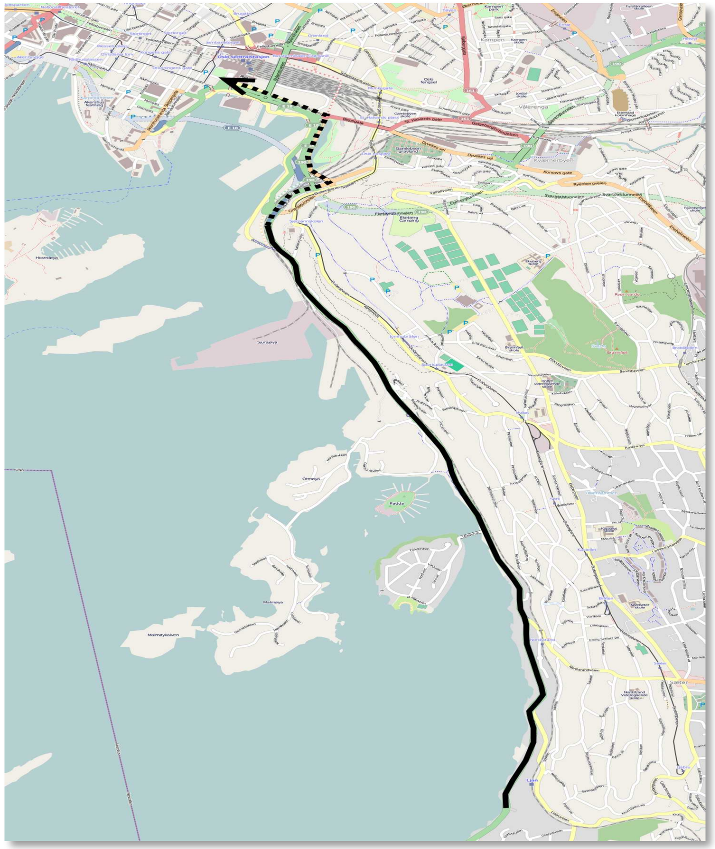
Netview Technology

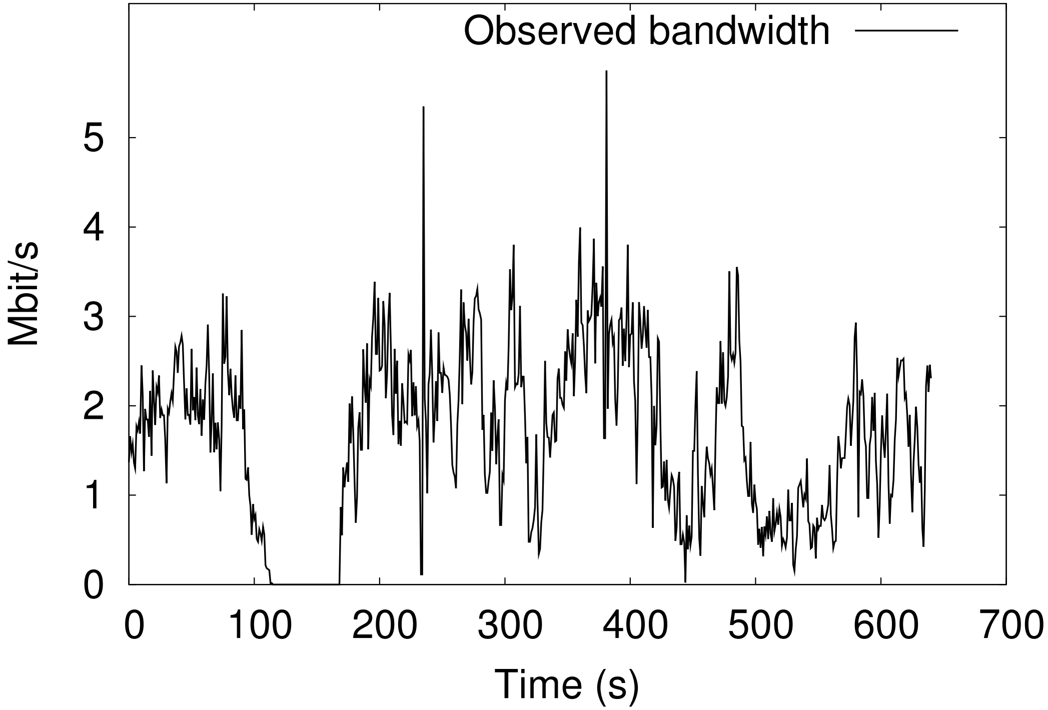
[simula . research laboratory]

IAD INFORMATION ACCESS DISRUPTIONS

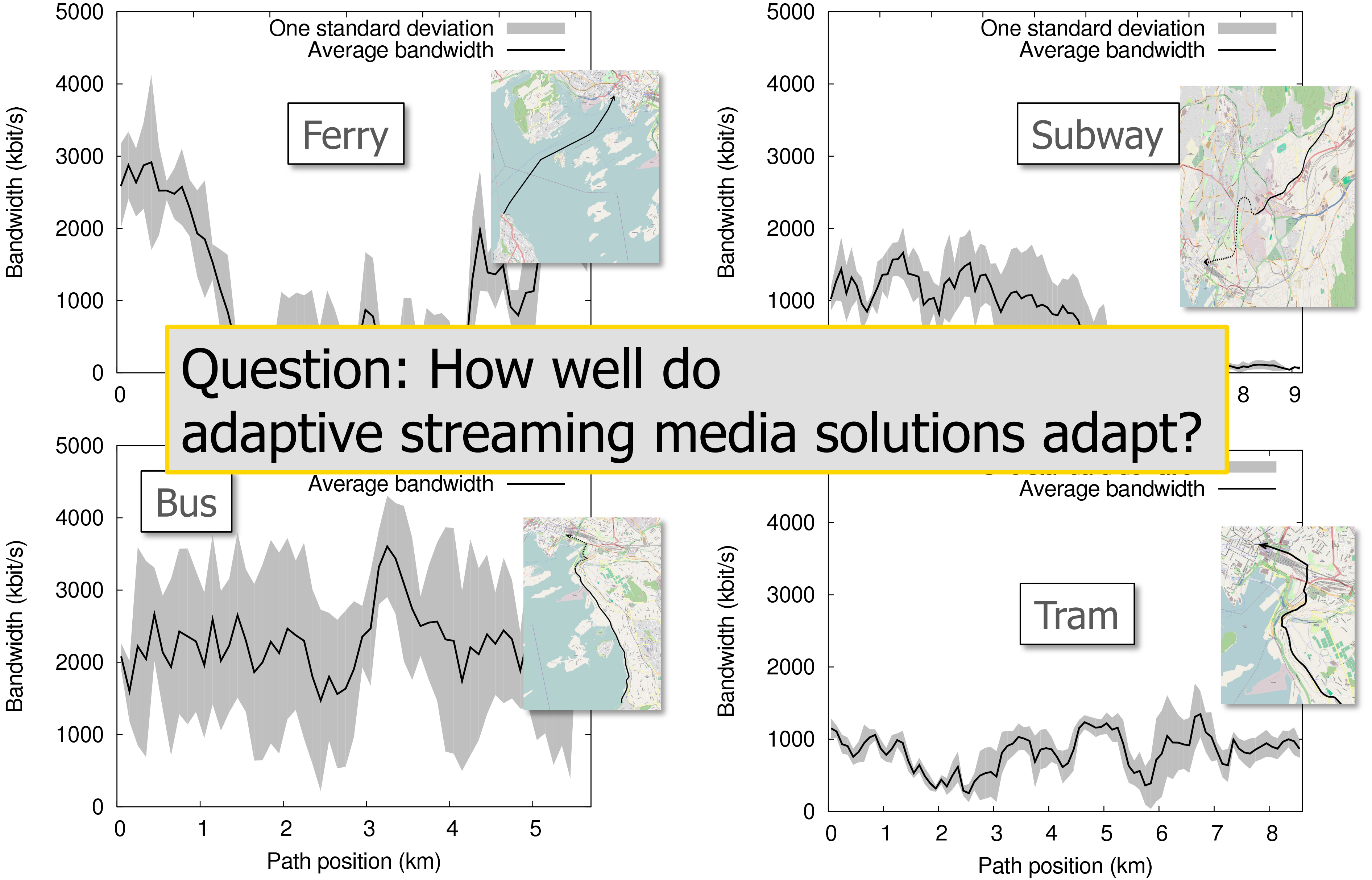
UNIVERSITAS OSLOENSIS MDCCCXI

Fluctuating Bandwidth Problem





Fluctuating Bandwidth Problem



Question: How well do adaptive streaming media solutions adapt?

Adaptive Delivery: Tested Systems

- Adobe Strobe Media Playback (v1.6.328 for Flash 10.1)
  using HTTP Dynamic Streaming Format
- Apple's native iPad player (iOS v4.3.3)
  using native HLS format
- Microsoft Silverlight/IIS Smooth (v4.0.60531.0 on Win7)
  using native Smooth format and default desktop scheduler
- Netview Media Client (v2011-10-10)
  using Apple HLS format (worst case) and Netview 3G scheduler

HTTP streaming

HTTP live streaming

smooth HD

Netview Technology

Server

- Apache v2.2.14
- CodeShop Unified Streaming Platform v1.4.25
- Throttling module for trace-based bandwidth limitation, 1sec resolution

Adaptive Delivery: Test Content

- Video: Norwegian football game (i.e. soccer)
- H.264/AVC
- 2 second segments

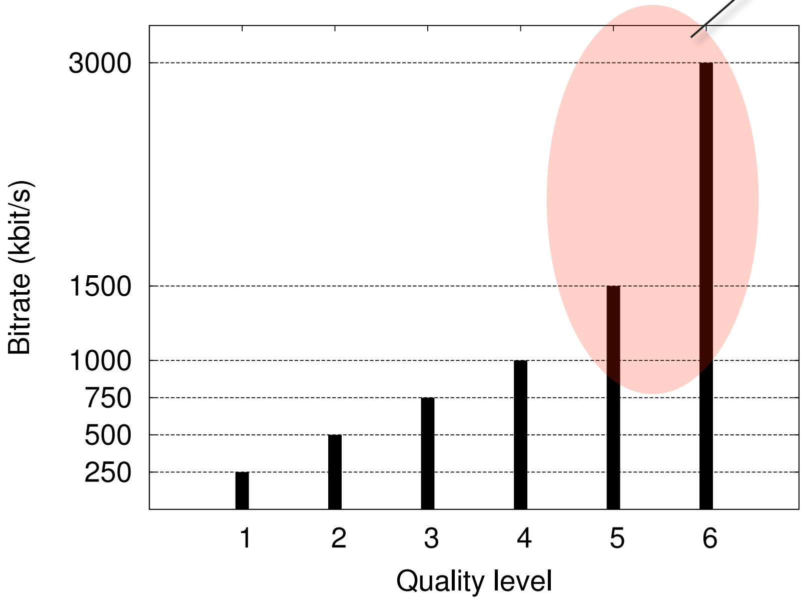
Akamai recommendations

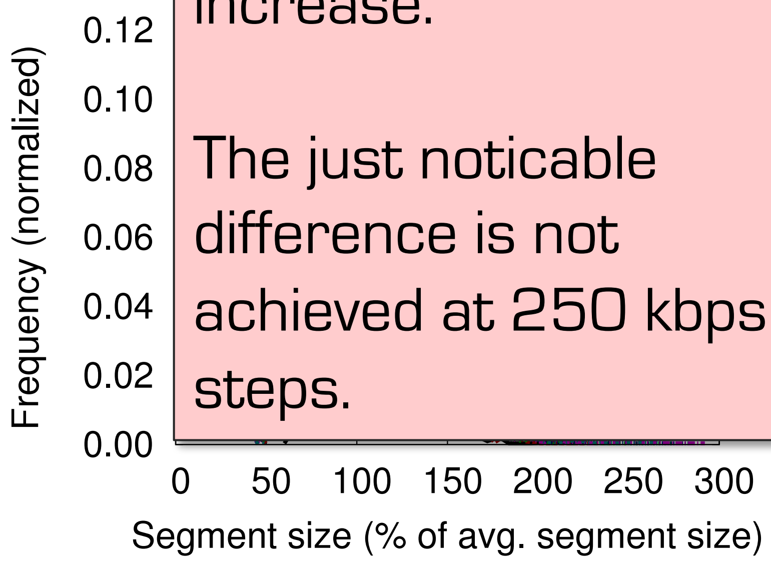
- 6 quality levels: 250, 500, 750, 1000,

Why is the bandwidth increase non-linear?

Bitrate increase provides dimishing return on visual quality increase.

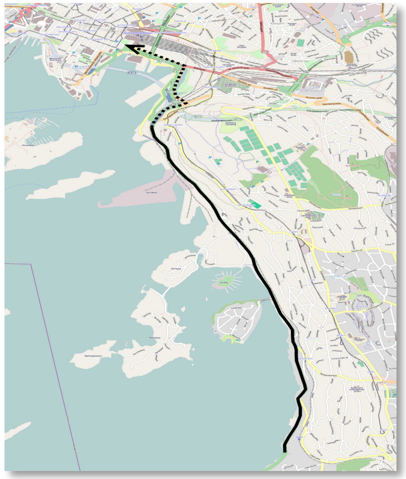
The just noticable difference is not achieved at 250 kbps steps.

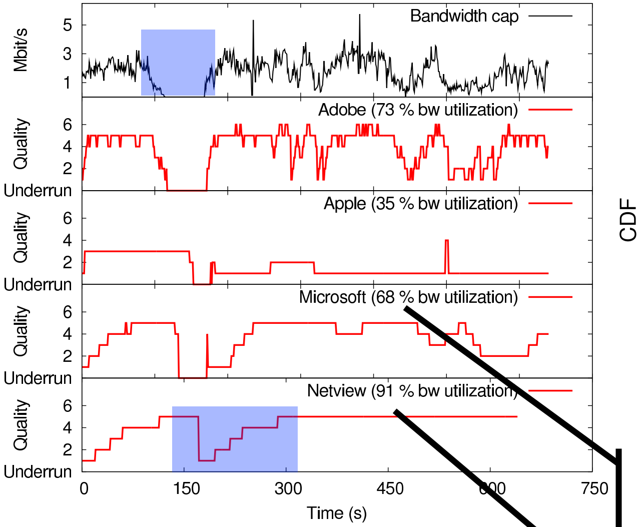


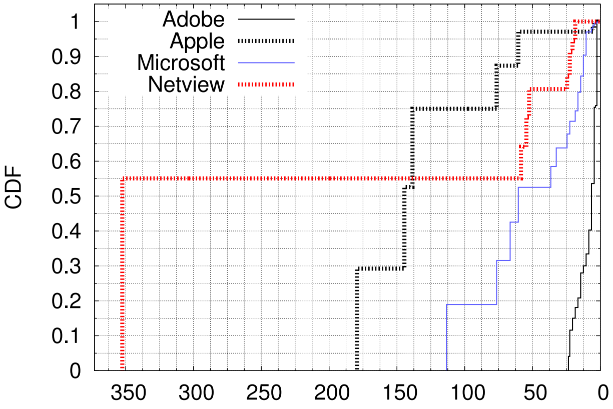


Comparison of Existing Quality Schedulers

Bus:



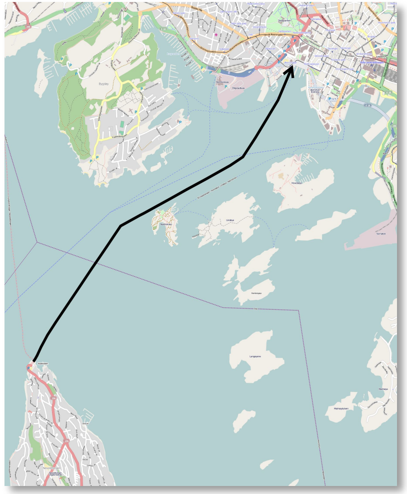

Mbit/s
5
3
1
Bandwidth cap
Adobe (73 % bw utilization)
Apple (35 % bw utilization)
Microsoft (68 % bw utilization)
Netview (91 % bw utilization)
Quality
6
4
2
Underrun
0
150
300
450
600
750
Time (s)

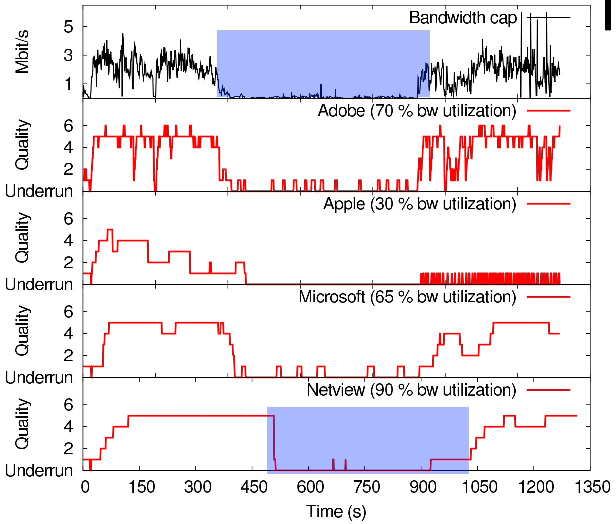


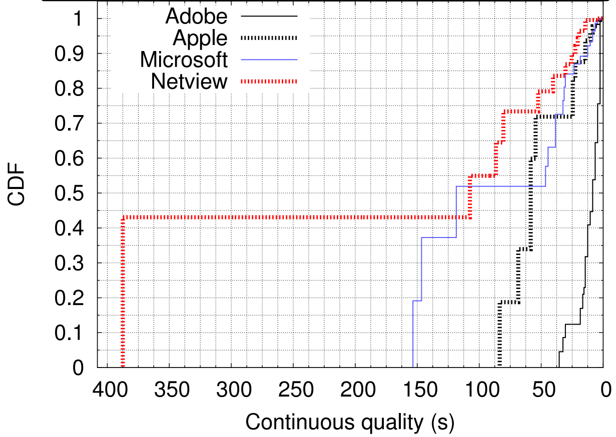
Configured for desktops

Configured for mobiles

Ferry:

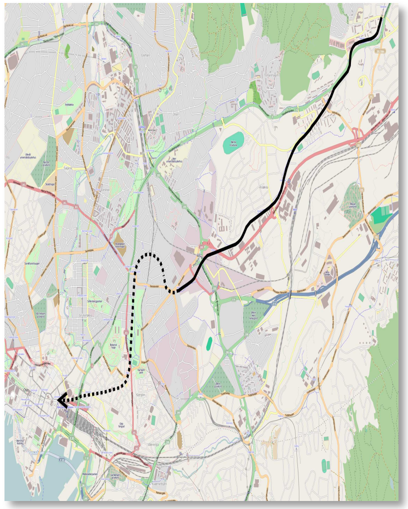


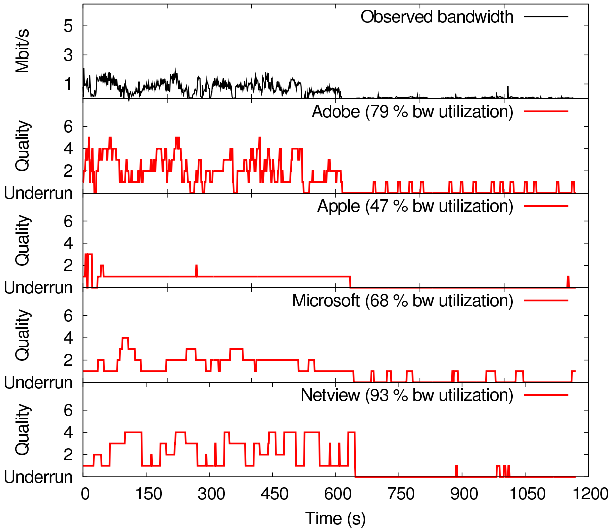

Mbit/s
5
3
1
Bandwidth cap
Adobe (70 % bw utilization)
Apple (30 % bw utilization)
Microsoft (65 % bw utilization)
Netview (90 % bw utilization)
Quality
6
4
2
Underrun
0
150
300
450
600
750
900
1050
1200
1350
Time (s)

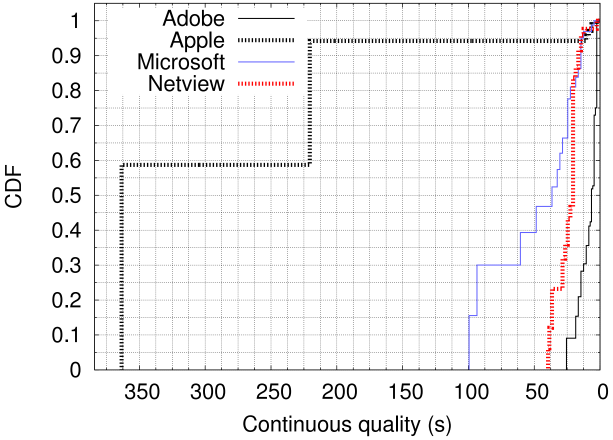


Comparison of Existing Quality Schedulers

Metro:




Observed bandwidth
Mbit/s
5
3
1
Adobe (79 % bw utilization)
Apple (47 % bw utilization)
Microsoft (68 % bw utilization)
Netview (93 % bw utilization)
Quality
6
4
2
Underrun
0
150
300
450
600
750
900
1050
1200
Time (s)



For the Netview algorithm see:
"Video Streaming Using a Location-based Bandwidth-Lookup Service for Bitrate Planning", Haakon Riiser, Tore Endestad, Paul Vigmostad, Carsten Griwodz, Pål Halvorsen, *TOMCCAP, July 2012*

Conclusion

Difference between players' quality scheduler dominates the overhead

Adobe: maximizes aggressiveness at the expense of stability

Apple: maximize stability at the expense of aggressiveness

Microsoft Silverlight: good compromise

Netview: similar to Microsoft, outperforms in this configuration

- Many people commute using the same route
- Many mobile devices have GPS receivers
- What about crowdsourcing the throughput on the commute paths at various times of day?

A Comparison of Quality Scheduling in Commercial Adaptive HTTP Streaming Solutions on a 3G Network

Haakon Riiser, Håkon S. Bergsaker, Paul Vigmostad,
Pål Halvorsen, Carsten Griwodz

Netview Technology

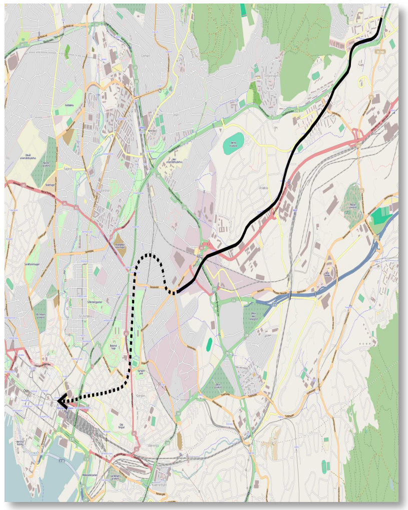
[simula . research laboratory]

IAD
INFORMATION ACCESS DISRUPTIONS

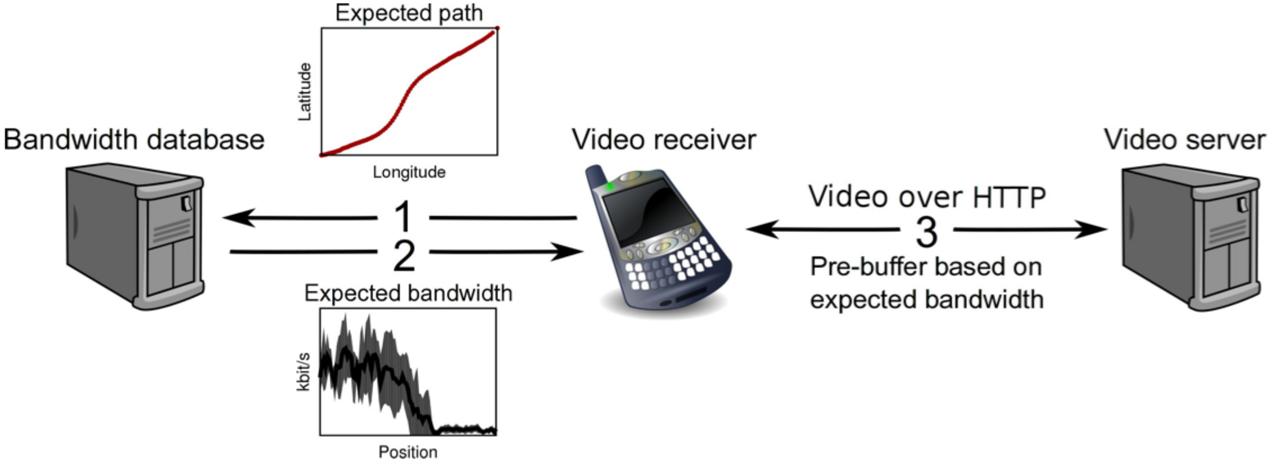
UNIVERSITAS OSLOENSIS MDCCCXI

Prediction

Metro:



Location-based bandwidth-lookup service for bitrate (video quality) planning:



Expected path

Bandwidth database

Video receiver

Video server

1

2

Expected bandwidth

Video over HTTP

3

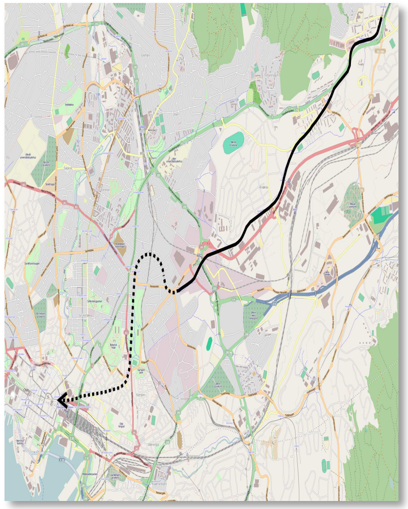
Pre-buffer based on expected bandwidth

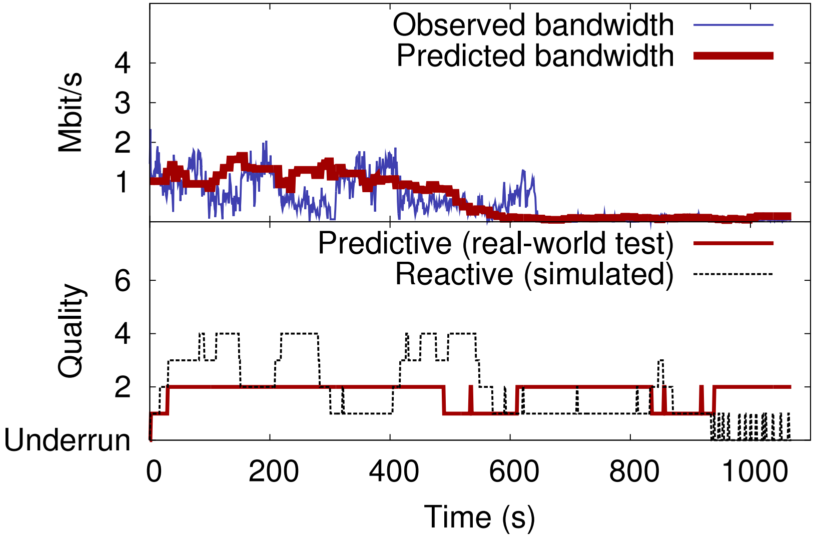
Algorithm:

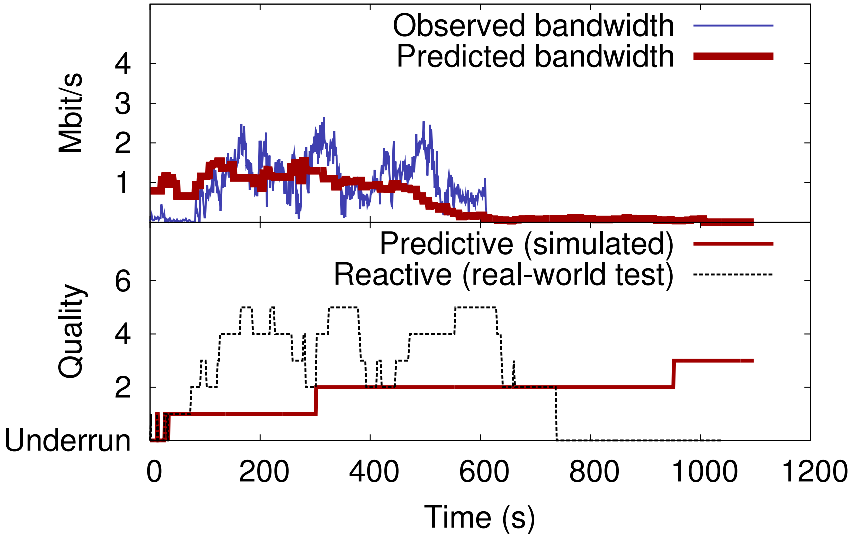
- Calculate the predicted amount of data that historically can be downloaded
- Calculate maximum steady quality without getting buffer-underruns
- Safety: reactive algorithm fallback

Prediction: Metro

Metro:







From:
"Video Streaming Using a Location-based Bandwidth-Lookup Service for Bitrate Planning",
Haakon Riiser, Tore Endestad, Paul Vigmostad, Carsten Griwodz, Pål Halvorsen,
TOMCCAP, July 2012

Quality-of-Experience Considerations with Relevance for Adaptive HTTP Streaming

Pengpeng Ni, Ragnhild Eg, Alexander Eichhorn,
Pål Halvorsen, Carsten Griwodz

[simula . research laboratory]

INFORMATION ACCESS DISRUPTIONS

UNIVERSITAS OSLOENSIS · MDCCCXI ·

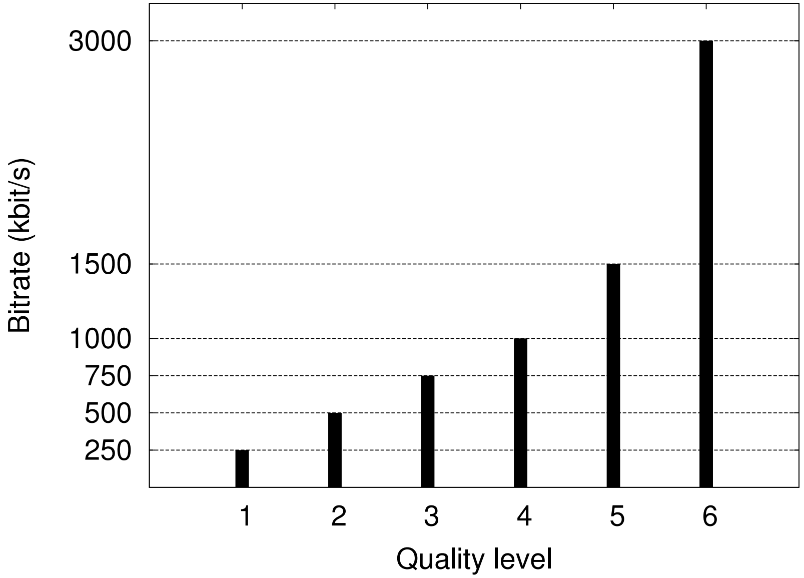
Adaptive Delivery: Test Content

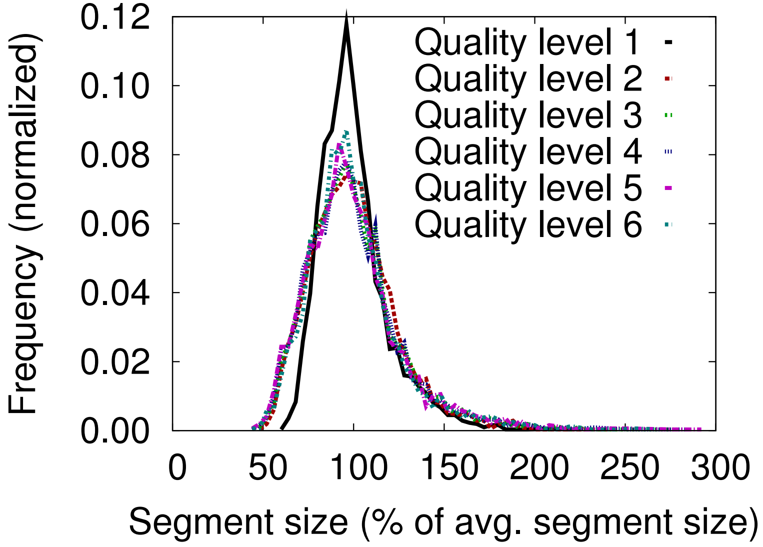
- Video: Norwegian football game (i.e. soccer)
- H.264/AVC
- 2 sec

Question: What is "good" adaptation?

Akamai recommendations

- 6 quality levels: 250, 500, 750, 1000, 1500, 3000 kbps





Why visual quality studies?

UDP-based streaming

- resists packets loss
- random loss

Applications

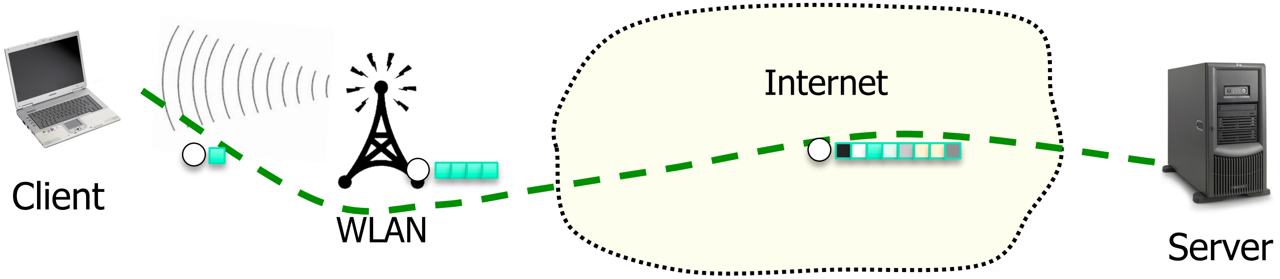
- IPTV
- DVD-H
- video conferencing
- classical RTSP/RTP servers

DASH & similar

- scales to available bandwidth
- congestion loss

Applications

- Commercial VoD: Netflix, Akamai, Microsoft, Apple, Comoyo, ...
- MPEG DASH
- Free VoD: Youtube, Metacafe, Dailymotion, Vimeo, Revver, Flixya ...



Why visual quality studies?

UDP-based streaming

Resilience to packet loss

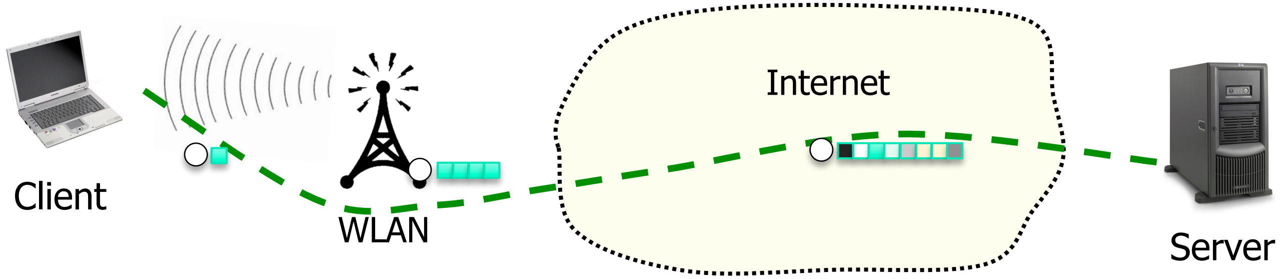
Possibly resilience to bit errors

Possibly active adaptation (server-side decision)

DASH & similar

Resilience to buffer underruns

Active adaptation (client-side decision)



PSNR

Peak Signal-to-Noise Ratio
A prevalent video quality metric

$$PSNR = 10\log_{10}\frac{(2^B - 1)^2}{MSE}$$

where:

$$MSE = \frac{1}{MN}\sum_{y=1}^{M}\sum_{x=1}^{N}[Im_a(x,y) - Im_b(x,y)]^2$$

M, N = image dimensions

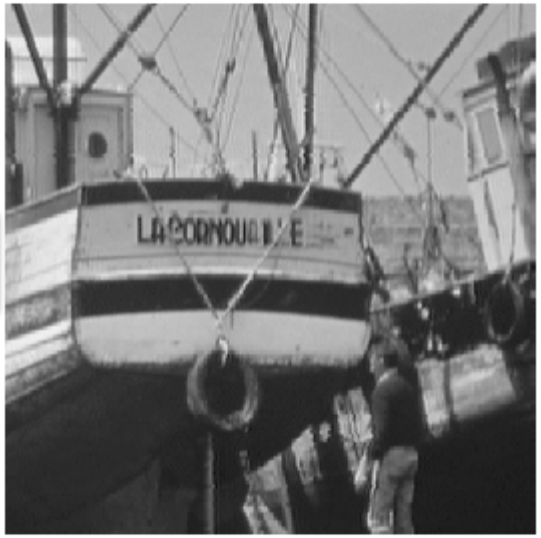
Im_a , Im_b = pictures to compare

B= bit depth

User experience: reduced PSNR

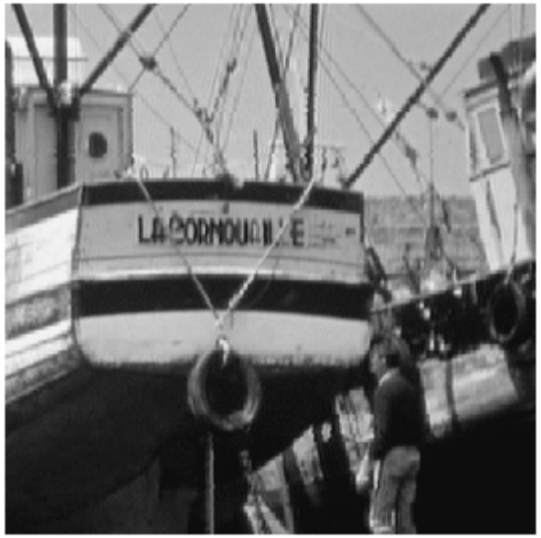
PSNR

Reference

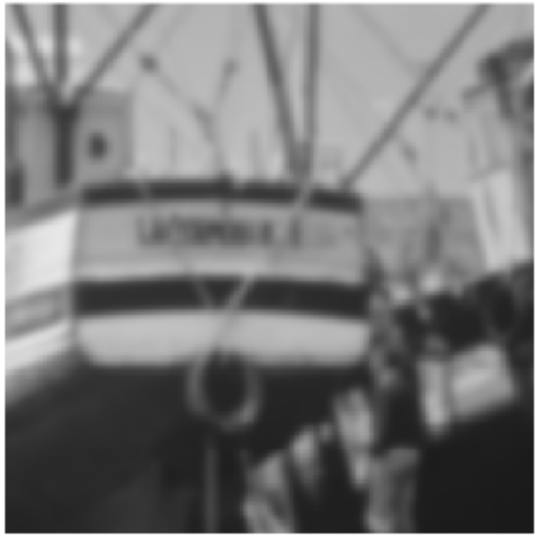


Example from
Prof. Touradj Ebrahimi,
ACM MM'09 keynote

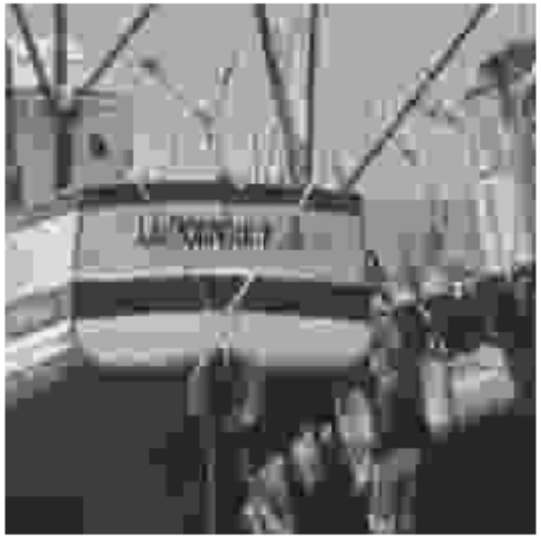
PSNR = 24.9 dB



PSNR = 24.9 dB



PSNR = 24.9 dB



PSNR

Peak Signal-to-Noise Ratio
A prevalent video quality metric

In addition to this:

- several different PSNR computations for color images
- different PSNR for different color spaces (RGB,YUV)
- visible influence of the encoding format

These problems hurts all metrics that are based on PSNR

Improved by image quality metrics such as

- SSIM variants
- rate distortion metrics

Conclusion

You encounter Quality-of-Experience measurements based on PSNR everywhere in the literature.

- it is very easy to compute
- it is probably working to assess the quality of images with bit errors

Be very careful when you encounter

1. PSNR-based objective metrics
2. adaptation mechanisms whose effectiveness is proven by PSNR

- It's easy to complain.
- But how to do better?

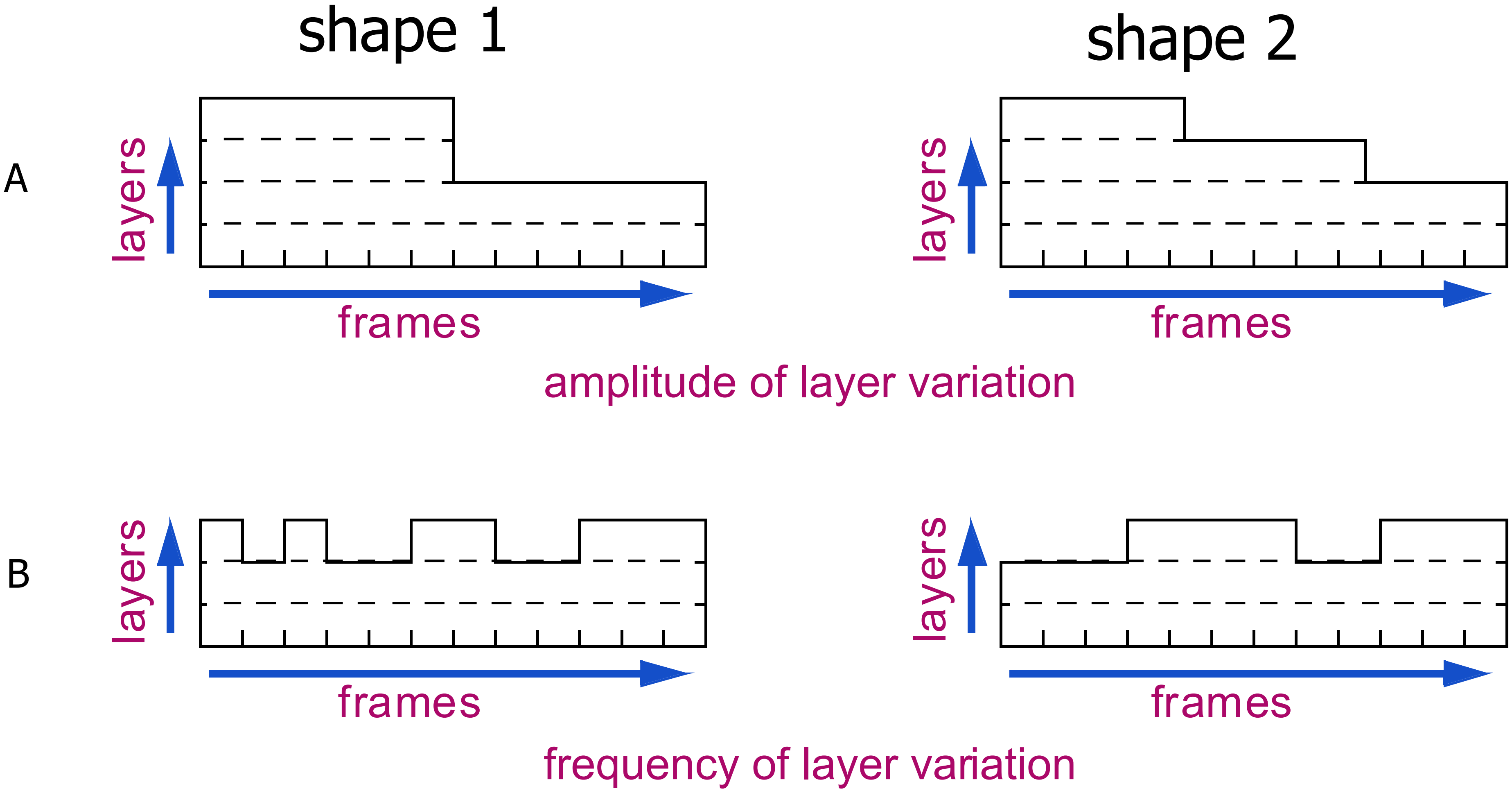
QoE Effects of Changing Layers in Adaptive HTTP Streaming

Pengpeng Ni, Ragnhild Eg, Alexander Eichhorn,
Pål Halvorsen, Carsten Griwodz, Michael Zink

Image-based metrics can fail badly: Flickering

Layered content

Used SPEG as layer encoded video format [avail. from C. Krasic, UBC]



Layered content

Comparing PSNR

with layer change counting: $S(v) = \sum |v_i - v_{i+1}|$

wh rs of a

fra

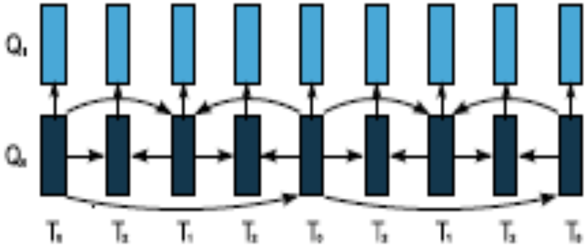
These observations discredit PSNR,
but they don’t provide a real alternative.

These observations indicate that frequent change is problematic.

But SPEG is outdated.
SPEG scaling decisions were based on intuition.
Adaptive HTTP Streaming today is based on H.264.

Sh							G
PS							29.84
PS							64.30
S(v							0
S(v) of shape 2	2	4	1	0	0	0.5	2
Subjective assessment	0.35	0.55	0.73	1.18	1.02	2.18	-0.24

Layer fluctuation in scalable video streaming



(a) Global Bitstream (G)

We need guidelines for video adaptation strategies

To cope with the bandwidth fluctuation, which scalability dimension is generally preferable for video adaptation?

Within each dimension, which scaling pattern generates the least annoying flicker effect?

Is it possible to control the annoyance of flicker effects?

How is subjective video quality related to other factors, such as content, devices?

We investigate visual artifact issues via several subjective tests

Previous Findings

- Human perception of video quality are content and context dependent
- There is no general preference order of scaling dimensions
- High frequency may cause flicker effect that is regarded as worse than constant low image quality
- In mobile scenario, a frame rate above 12 fps does not cause noticeable visual artifacts
- In HDTV scenario, a frame rate of 12 fps does generate noticeable motion jerkiness

Challenges of subjective tests:

- Efficient and reliable experimental design that avoids fatigue and learning effect

Flickering effects: the visual artifacts caused by layer fluctuation in scalable video

Flicker arises from recurrent changes in spatial or temporal quality, some so rapid that the human visual system only perceives fluctuations within the video.

Noise flicker

Compression scaling

Blur flicker

Resolution scaling

Motion flicker

Frame rate scaling

Blurriness flicker example

Blur flicker
Amplitude: 480x320px – 120x80px
Frequency: 10f / 3 Hz

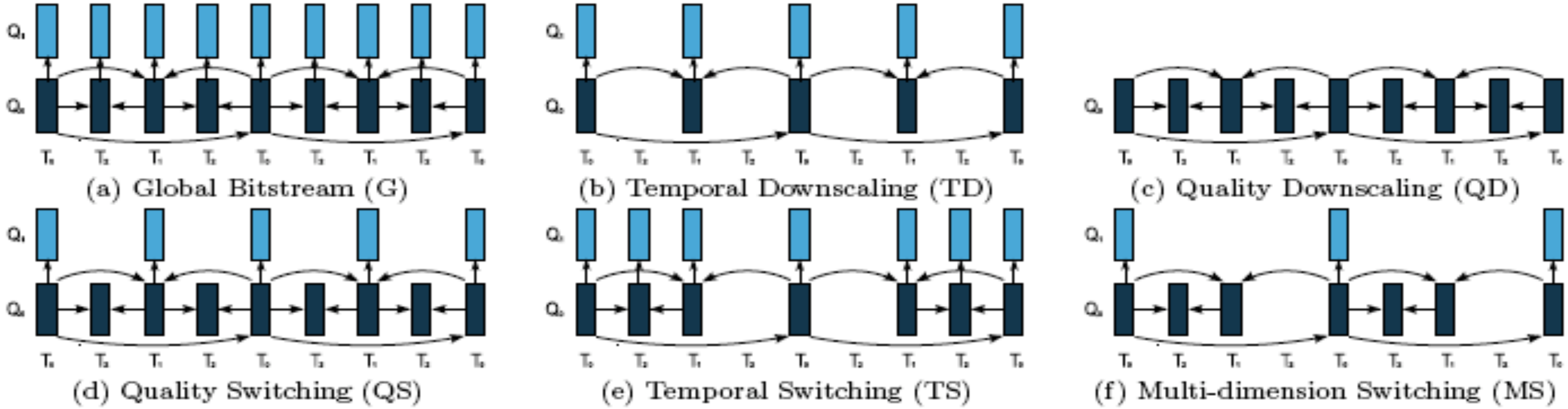
Noise flicker example

Noise flicker
Amplitude: QP24 – QP40
Frequency: 90f / 0.33 Hz

Motion flicker example

Motion flicker
Amplitude: 30fps – 3fps
Frequency: 6f / 5 Hz

How to describe different layer fluctuations?



(a) Global Bitstream (G)

(b) Temporal Downscaling (TD)

(c) Quality Downscaling (QD)

(d) Quality Switching (QS)

(e) Temporal Switching (TS)

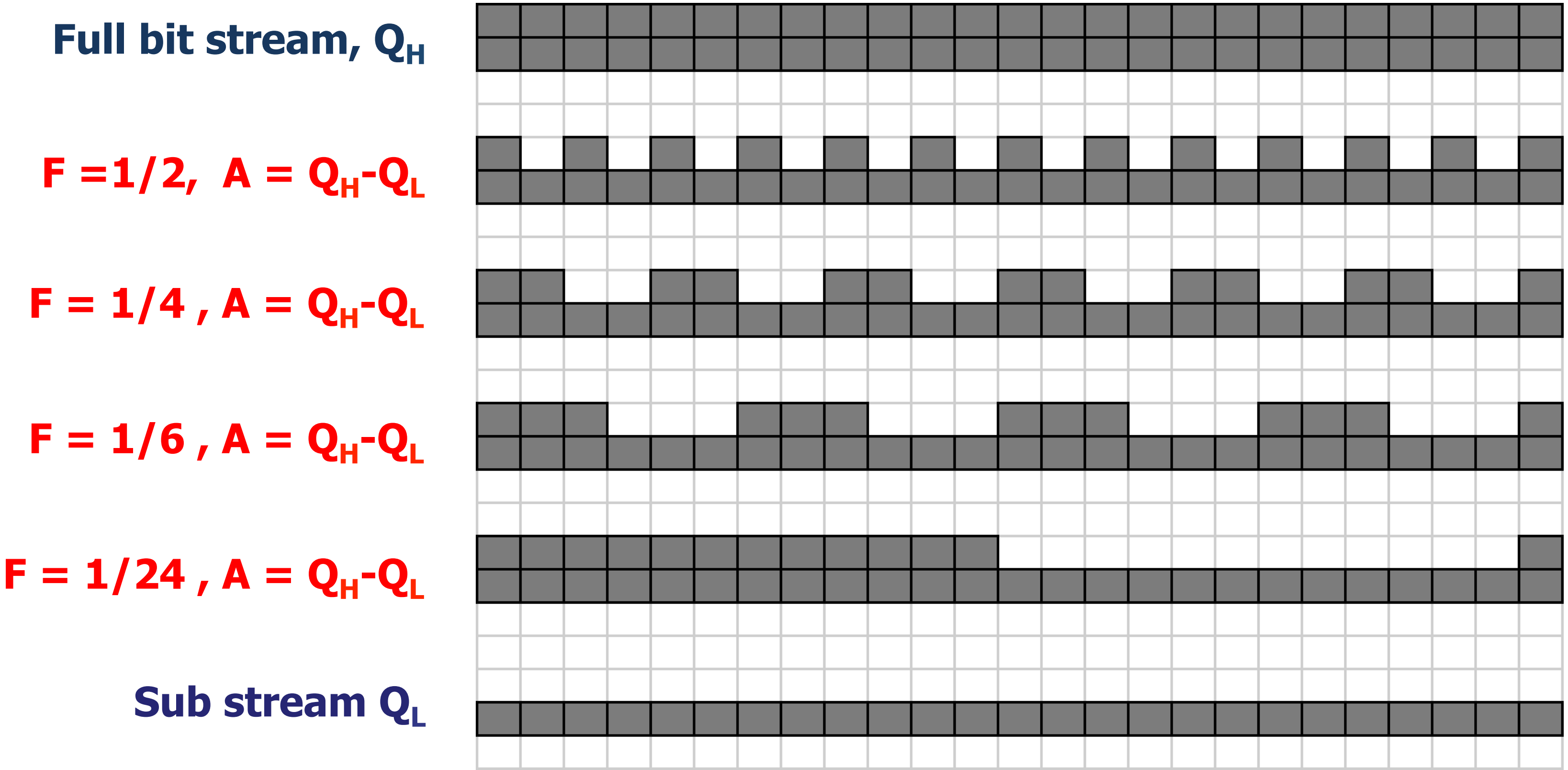
(f) Multi-dimension Switching (MS)

Layer fluctuation pattern

- Frequency: The time interval it takes for a video sequence return to its previous status
- Amplitude: The quality difference between the two layers being switched
- Dimension: Spatial or temporal, artifact type

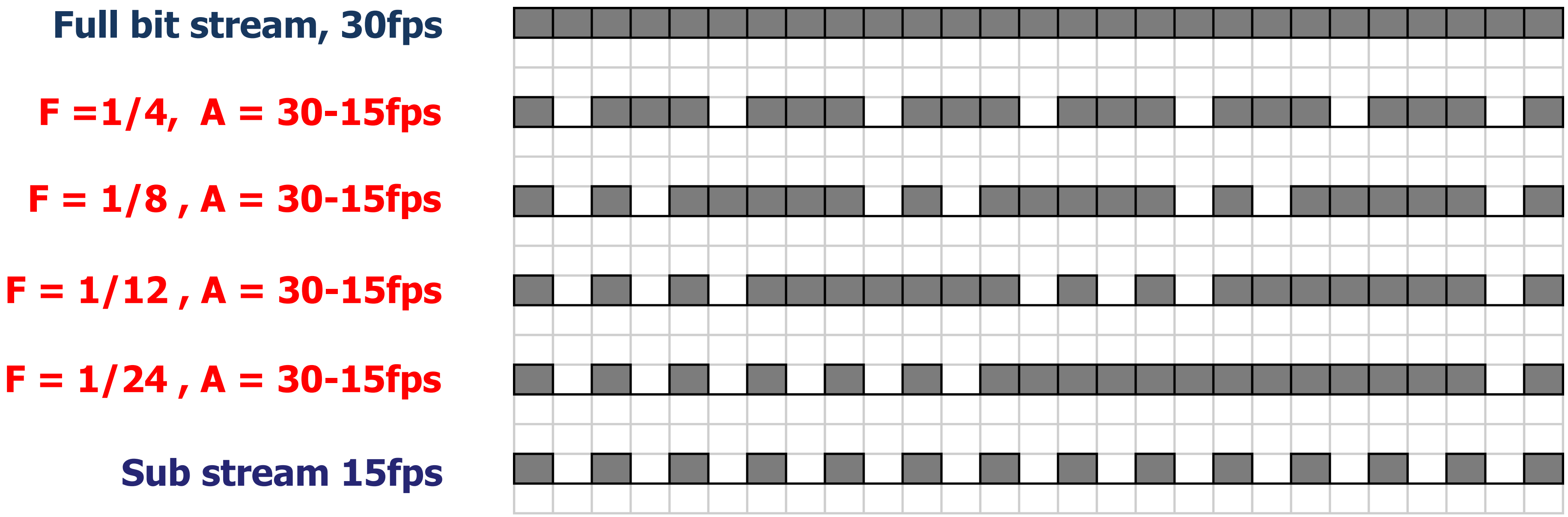
Layer Frequency and Amplitude are the interesting factors in our subjective test

Layer fluctuation pattern in Spatial dimension


Full bit stream, Q_H
F =1/2, A = Q_H-Q_L
F = 1/4 , A = Q_H-Q_L
F = 1/6 , A = Q_H-Q_L
F = 1/24 , A = Q_H-Q_L
Sub stream Q_L

Bandwidth consumption in all of the **changing** patterns is the same, due to the same amplitude. References Q_H and Q_L differ.

Layer fluctuation pattern in Temporal dimension



Although the average bit-rate is the same, the visual experience of different patterns may not be identical.

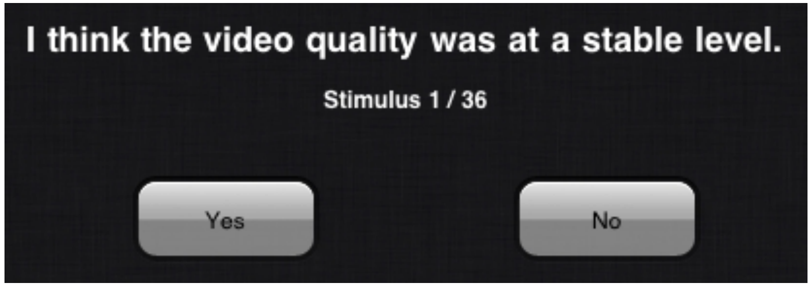
Method

Participants

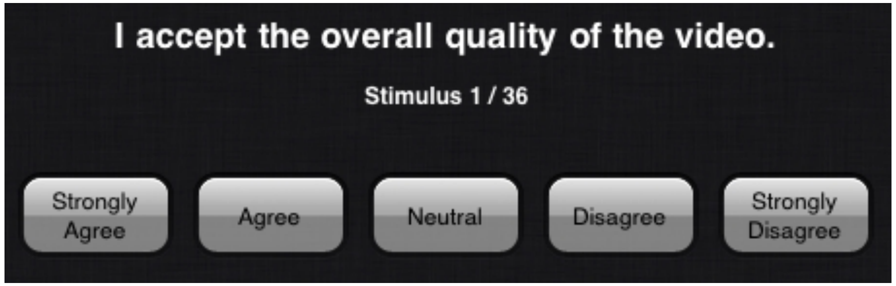
- 28 paid, voluntary participants
- 9 females, 19 males
- Age 19 – 41 years (mean 24)
- Self-reported normal hearing, and normal/corrected vision

Procedure

- Field study at university library
- Presented on iPod touch devices
  - Resolution 480x320
  - Frame rate 30 fps
- 12 sec video duration
- Random presentations
- Optional number of blocks

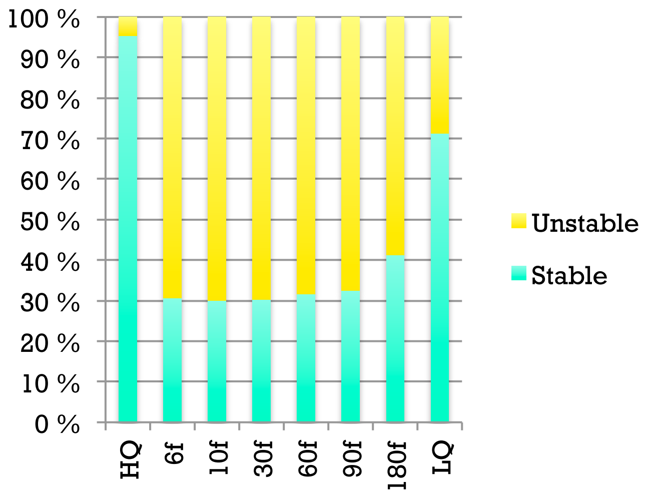

I think the video quality was at a stable level.

Stimulus 1 / 36

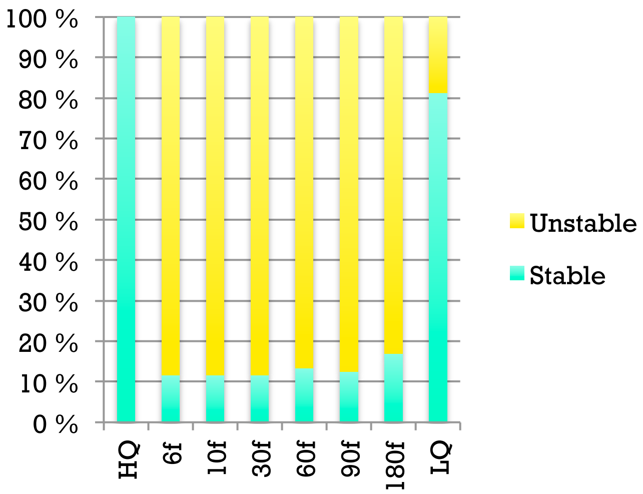

I accept the overall quality of the video.

Stimulus 1 / 36

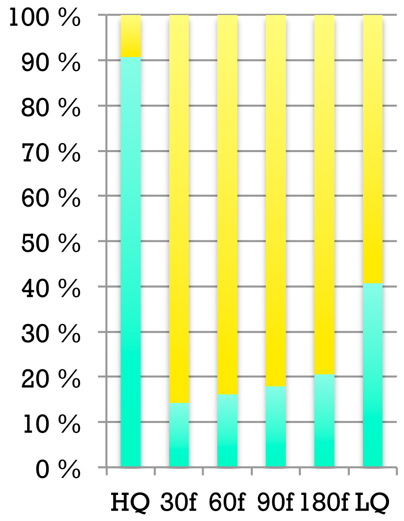
Stability scores - Period



Perceived quality stability across period levels for *Noise flicker*

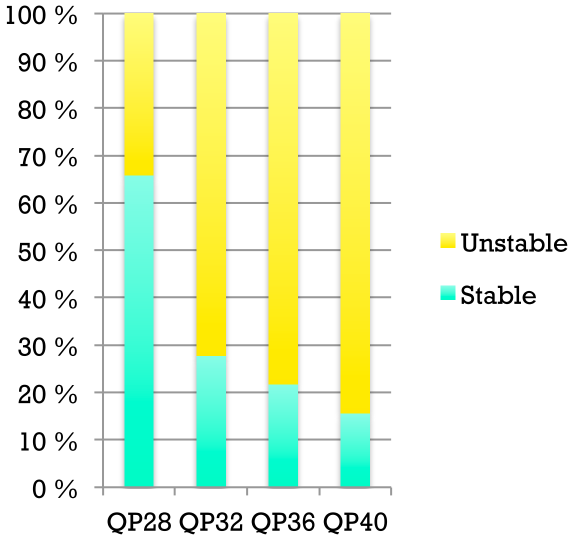


Perceived quality stability across period levels for *Blur flicker*

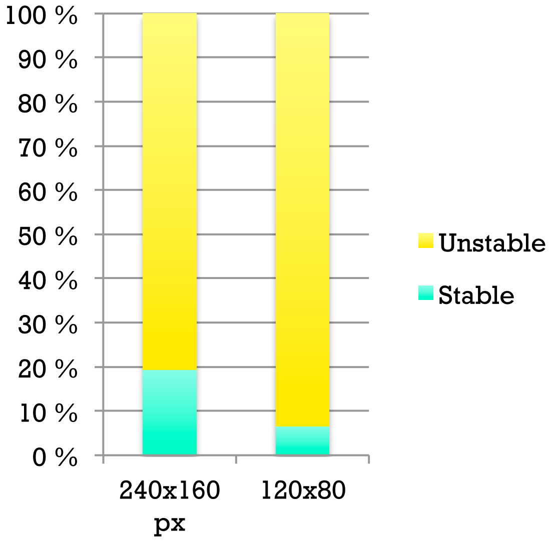


Perceived quality stability across period levels for *Motion flicker*

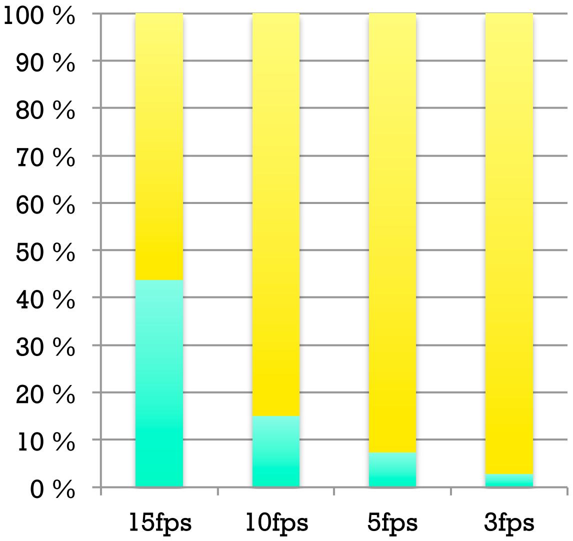
Stability scores - Amplitude



Perceived quality stability across amplitude levels for *Noise flicker*

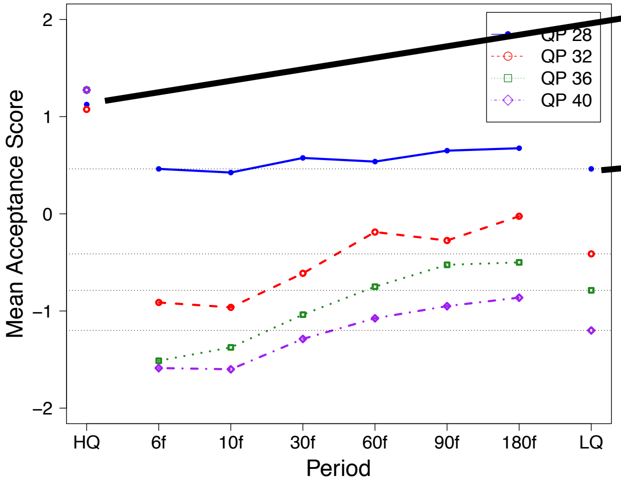


Perceived quality stability across amplitude levels for *Blur flicker*



Perceived quality stability across amplitude levels for *Motion flicker*

Video quality



Constant high quality references

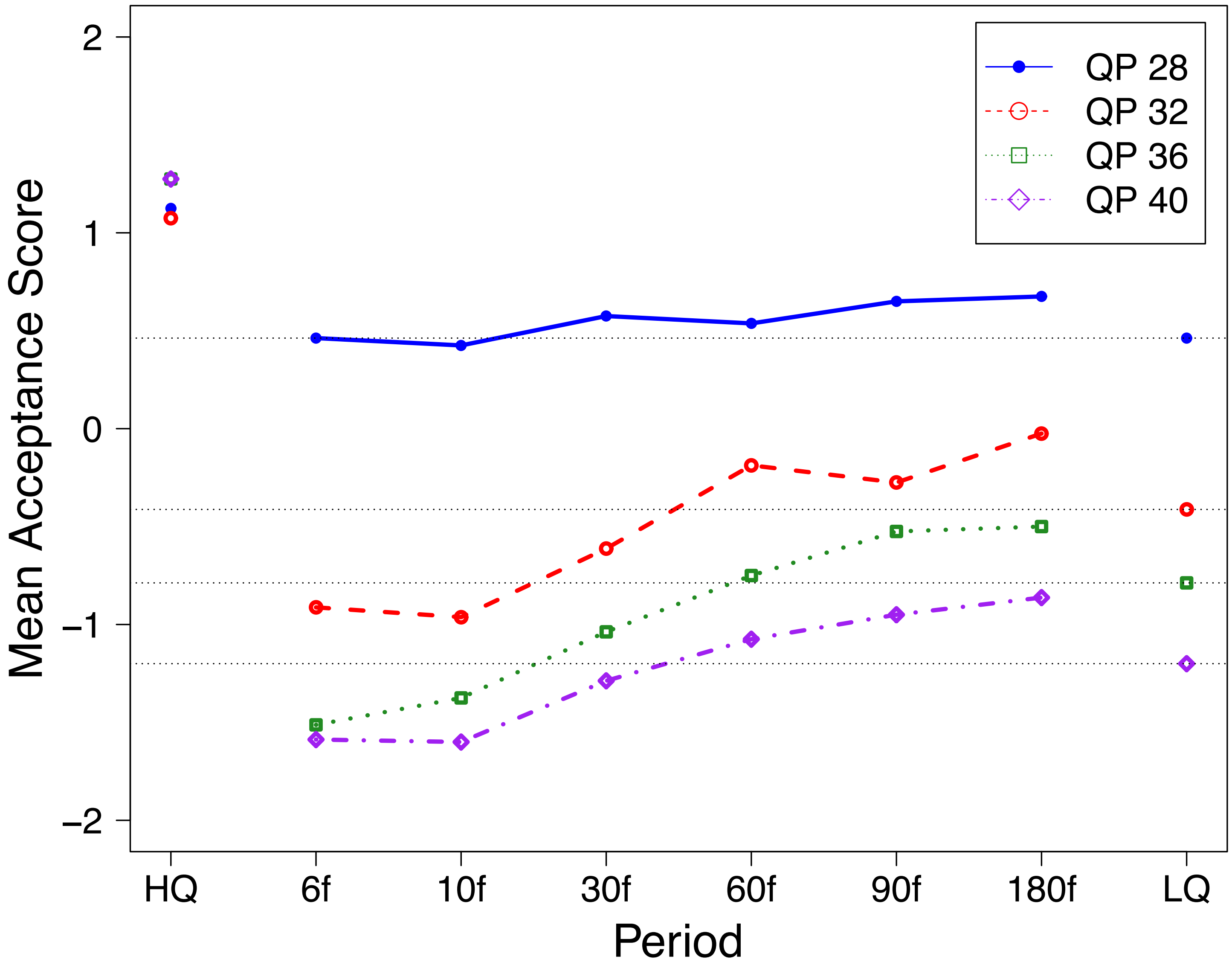
Constant low quality reference, QP28

Not investigated here: relation between qualities

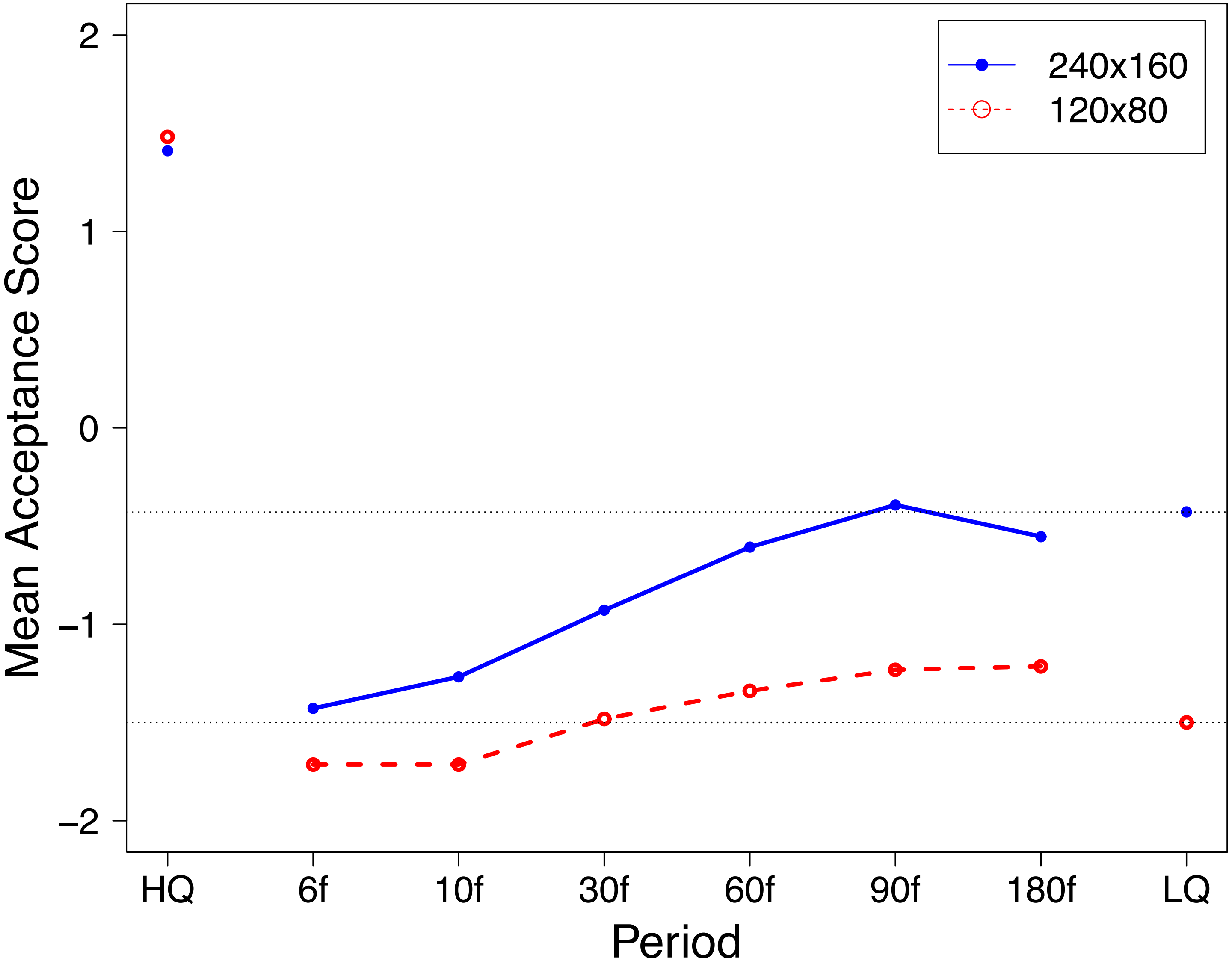
Noise	
L1	QP24
L0	QP28, QP32, QP36, QP40
Period	1/5s, 1/3s, 1s, 2s, 3s, 6s
Content	4 mid/long distance shots

From:
"Flicker Effects in Adaptive Video Streaming to Handheld Devices",
Pengpeng Ni, Ragnhild Eg, Alexander Eichhorn, Carsten Griwodz, Pål Halvorsen
ACM Multimedia 2011

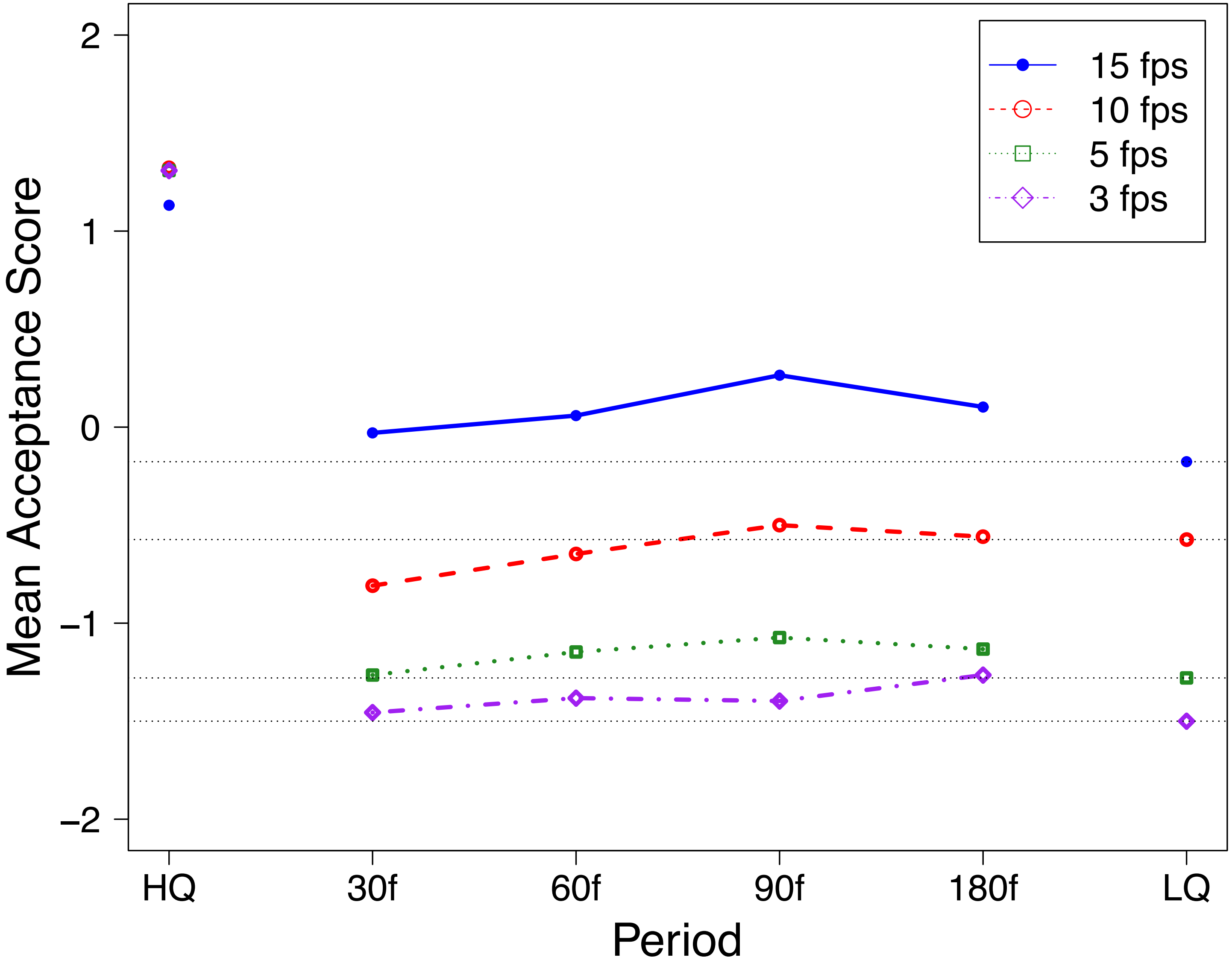
Acceptance - Noise flicker



Acceptance – Blur flicker



Acceptance – Motion flicker



Conclusions

With longer flicker frequencies (high periods), acceptance of video quality increases in the spatial dimension

Amplitude (quality difference) has larger effect than frequency, both for stability and acceptance

For noise flicker, large quality differences are rated more acceptable with less frequent quality shifts.

For blur flicker, improved acceptance with less frequent shifts is more pronounced for the smallest quality difference.

The flicker effect varies across contents, particularly for motion flicker.

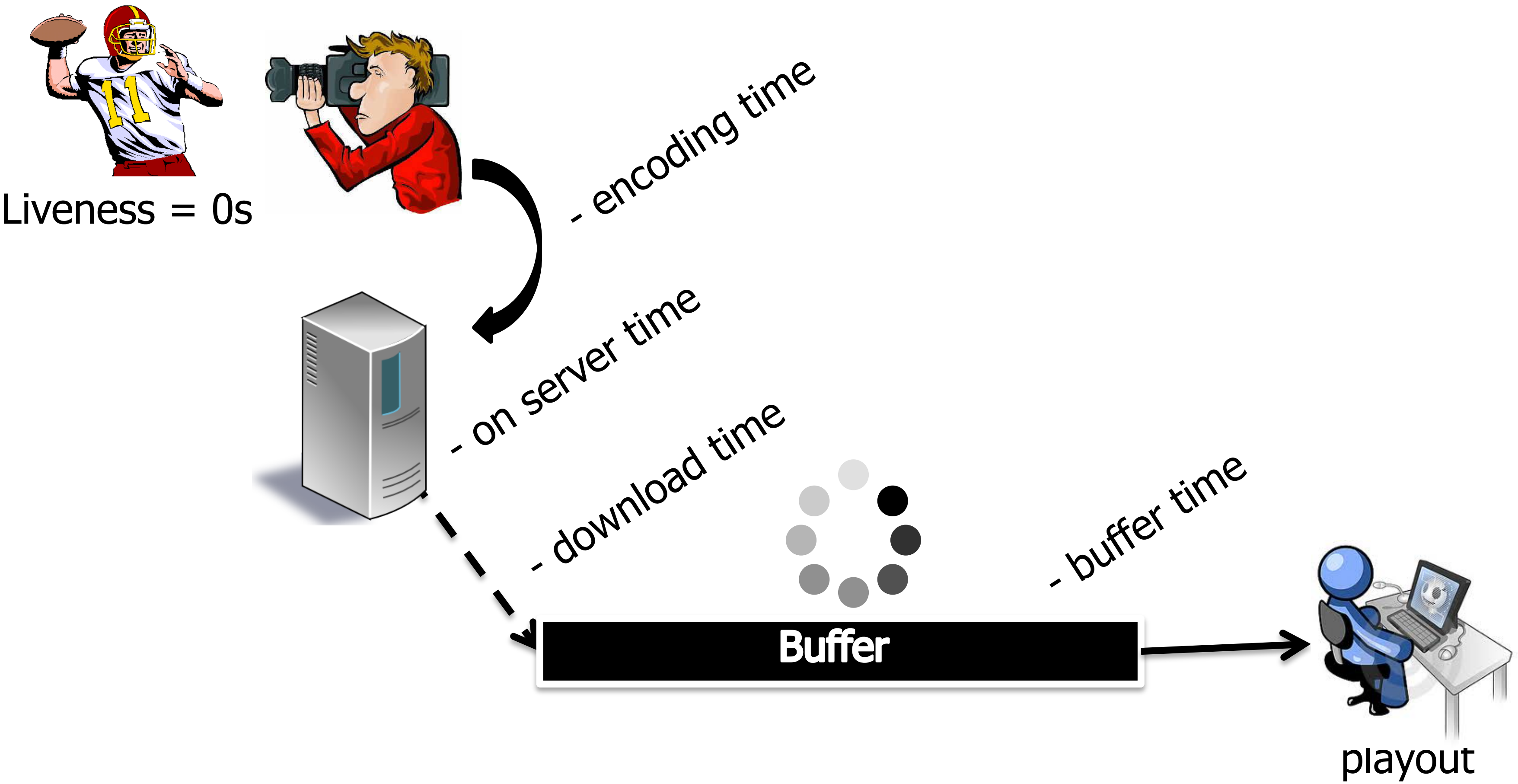
The three types of flicker have different influences on stability and quality acceptance scores. Scores are generally lower for blur flicker.

Network Behaviour of DASH Traffic

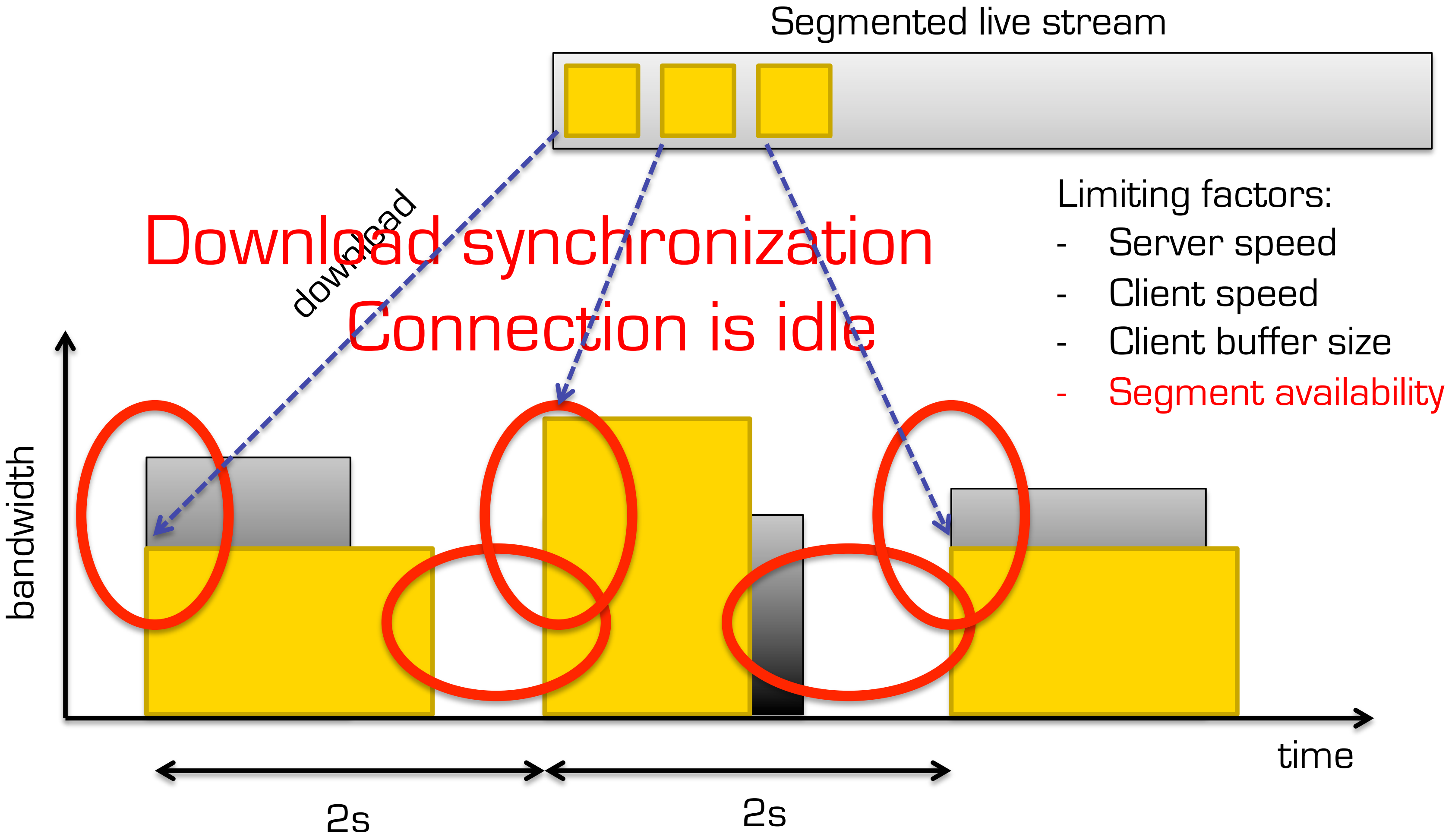
Tomas Kupka, Pål Halvorsen, Carsten Griwodz

Stream liveness

- The time it takes for an event to appear on the screen



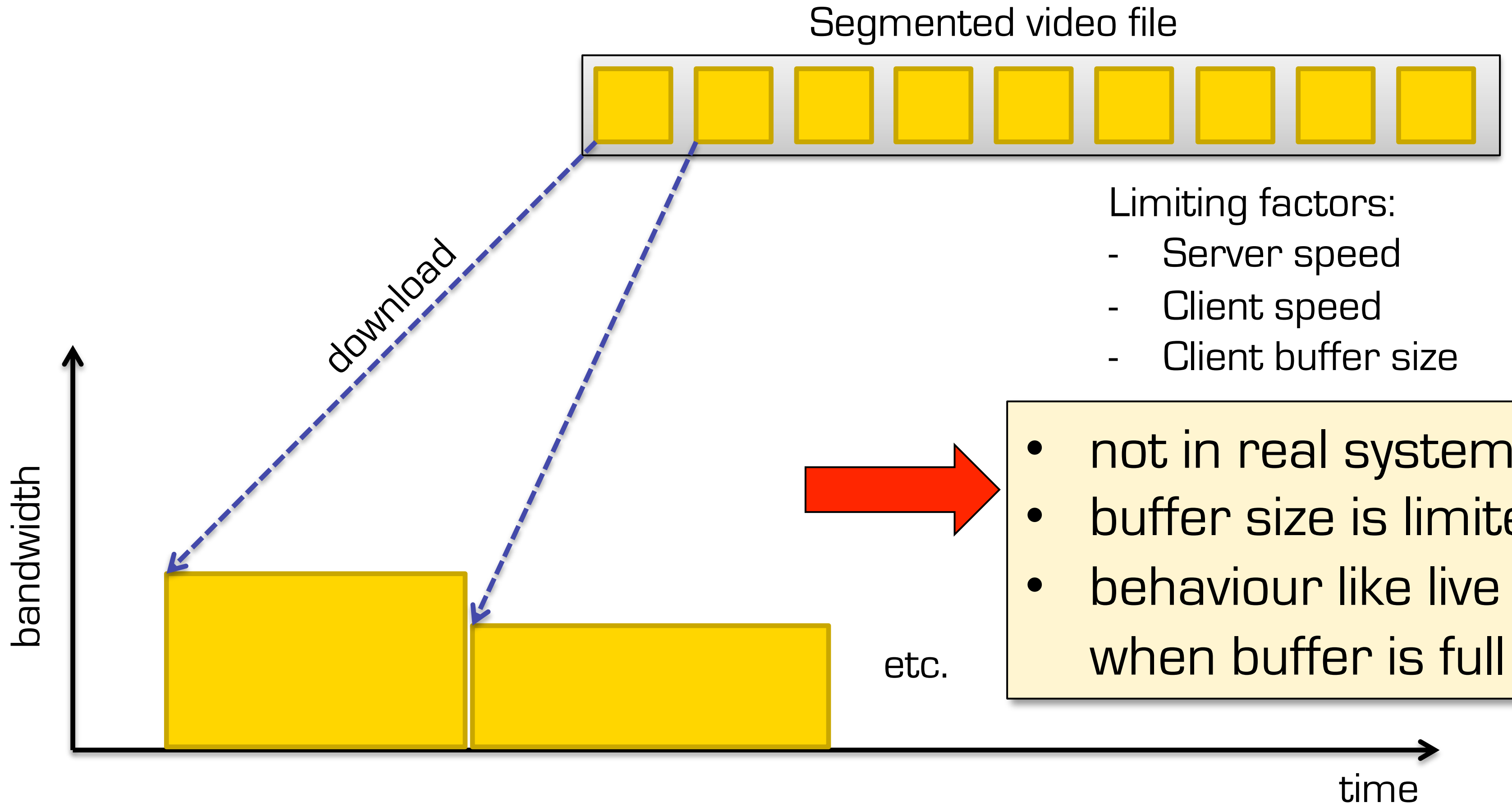
Live traffic pattern



Limiting factors:

- Server speed
- Client speed
- Client buffer size
- Segment availability

VoD traffic pattern



Limiting factors:
- Server speed
- Client speed
- Client buffer size

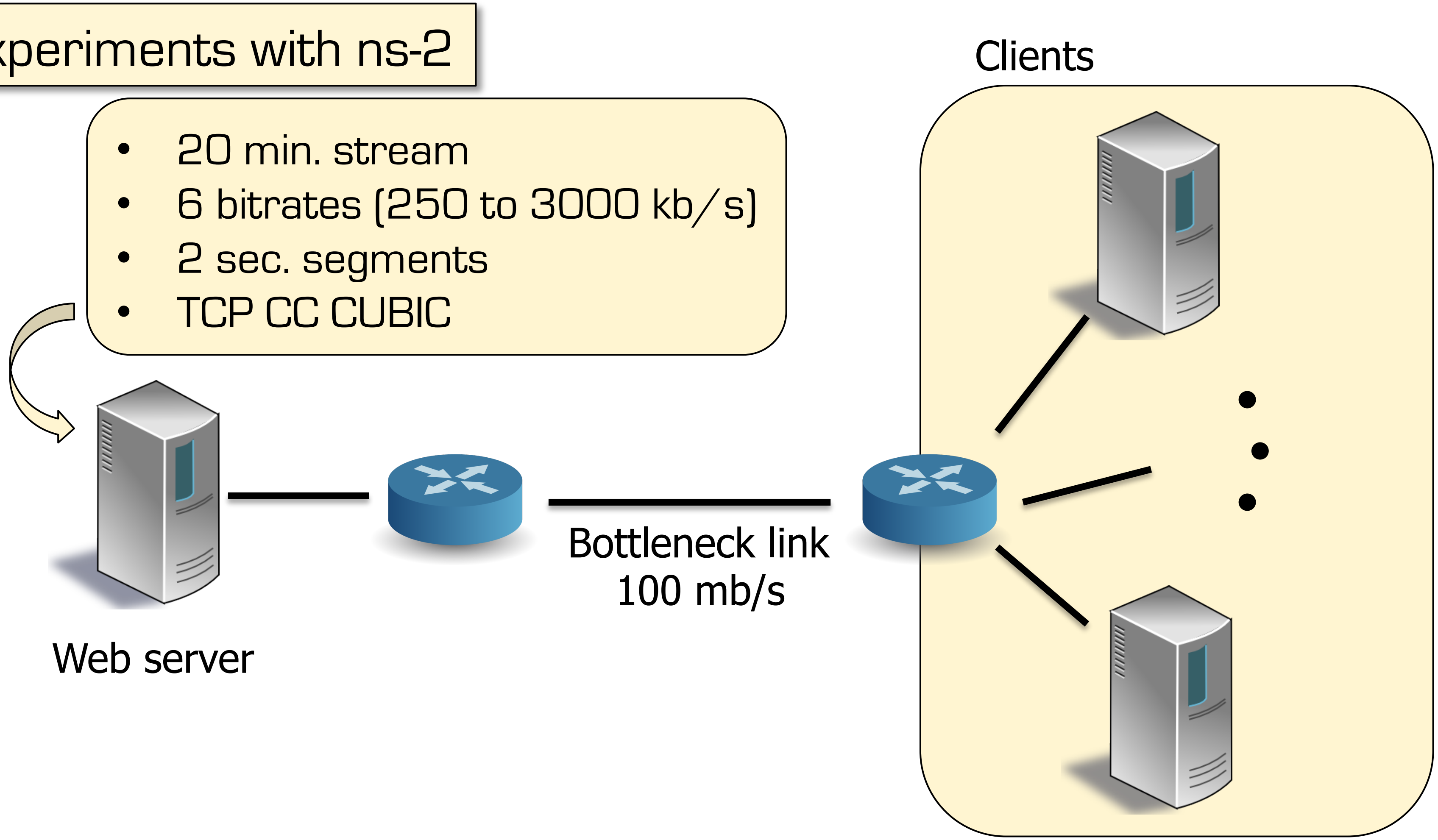
- not in real systems
- buffer size is limited
- behaviour like live when buffer is full

Transporting DASH

1. Does the request synchronization lead to problem?
2. Can we reduce such problem?

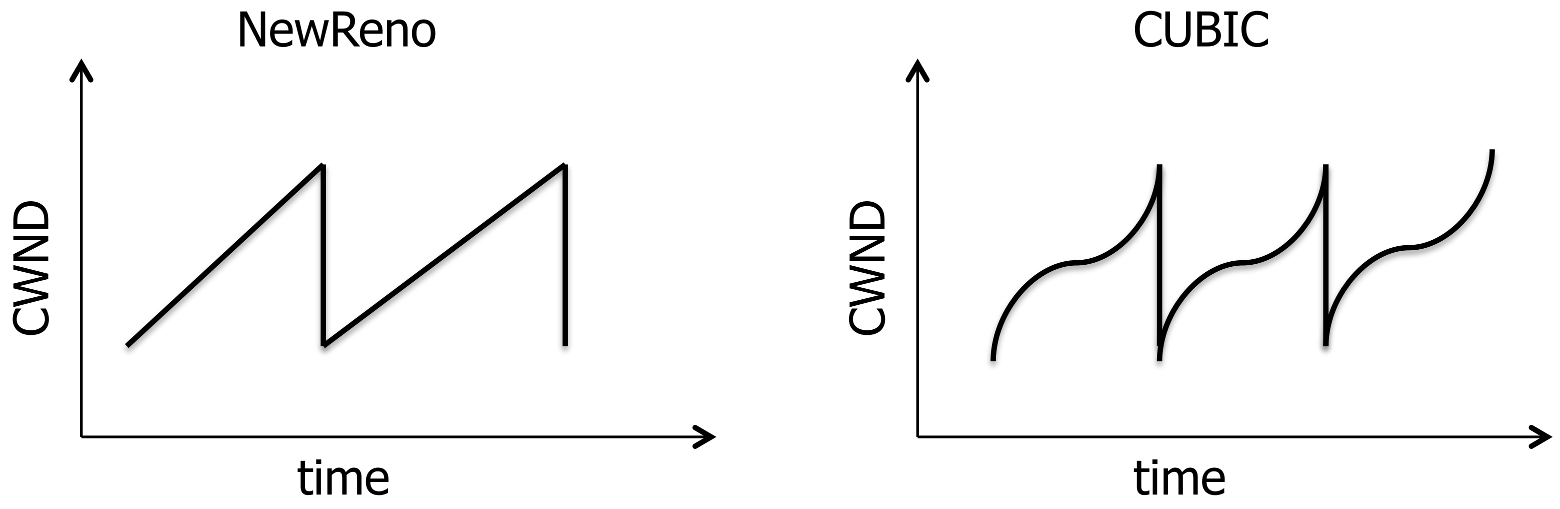
Experiments with ns-2

- 20 min. stream
- 6 bitrates (250 to 3000 kb/s)
- 2 sec. segments
- TCP CC CUBIC



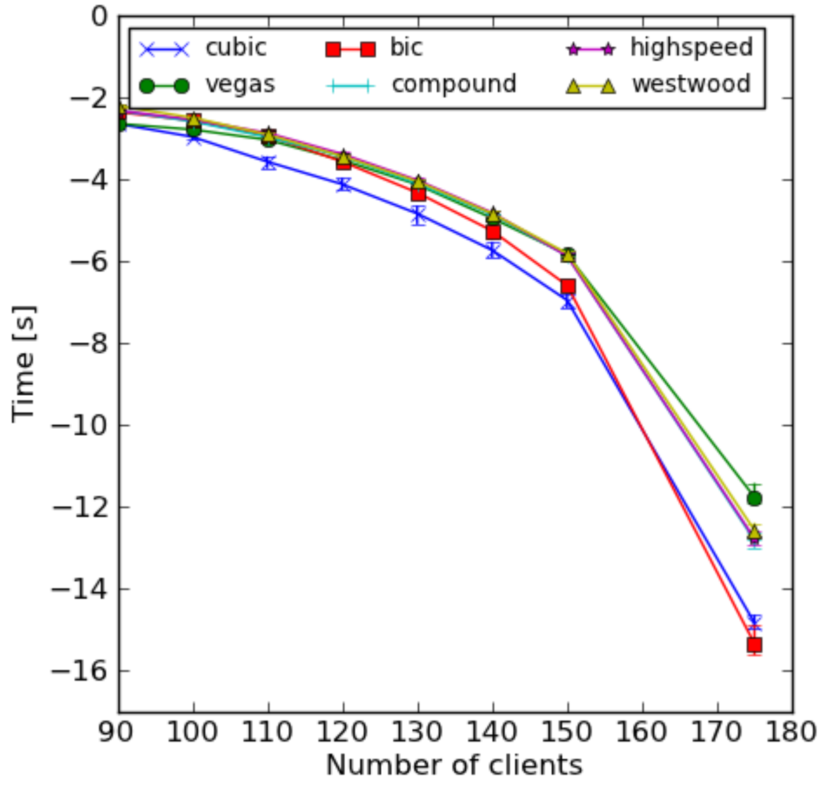
Evaluated modifications

- TCP congestion control alternatives
  - TCP's behave differently for short On traffic

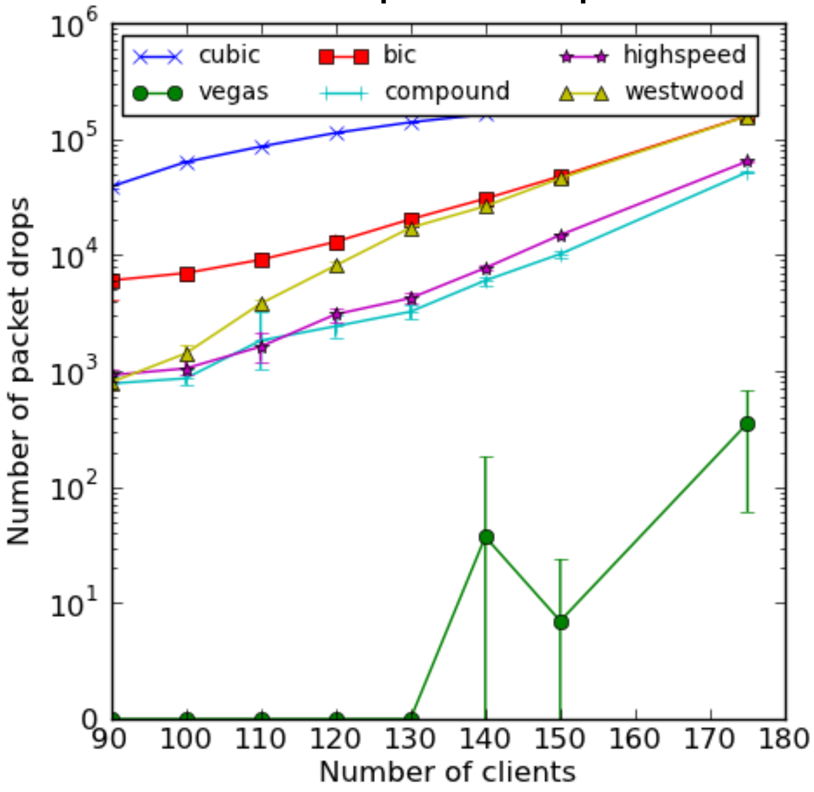


TCP congestion control alternatives

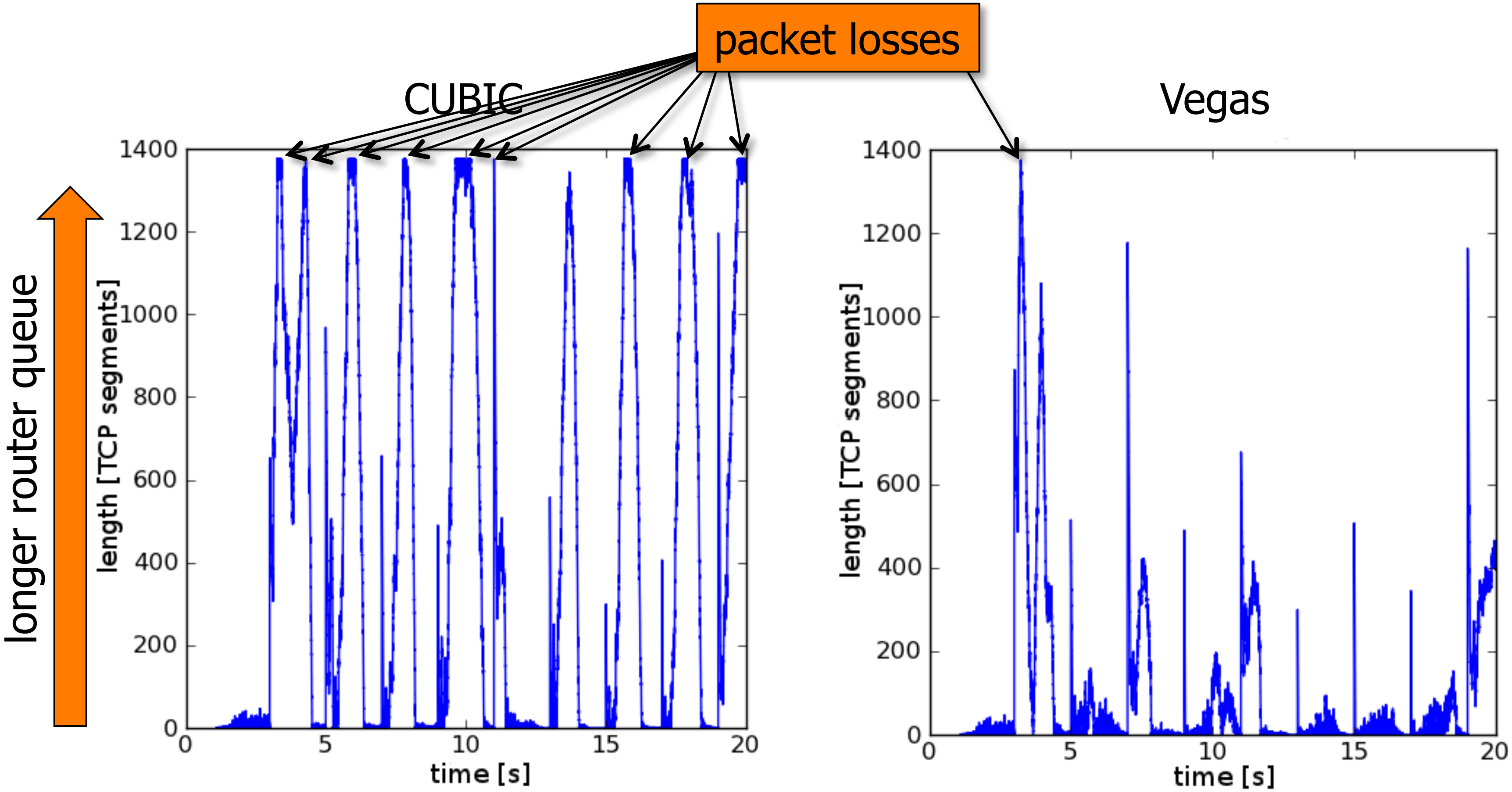
Liveness



Router queue drops



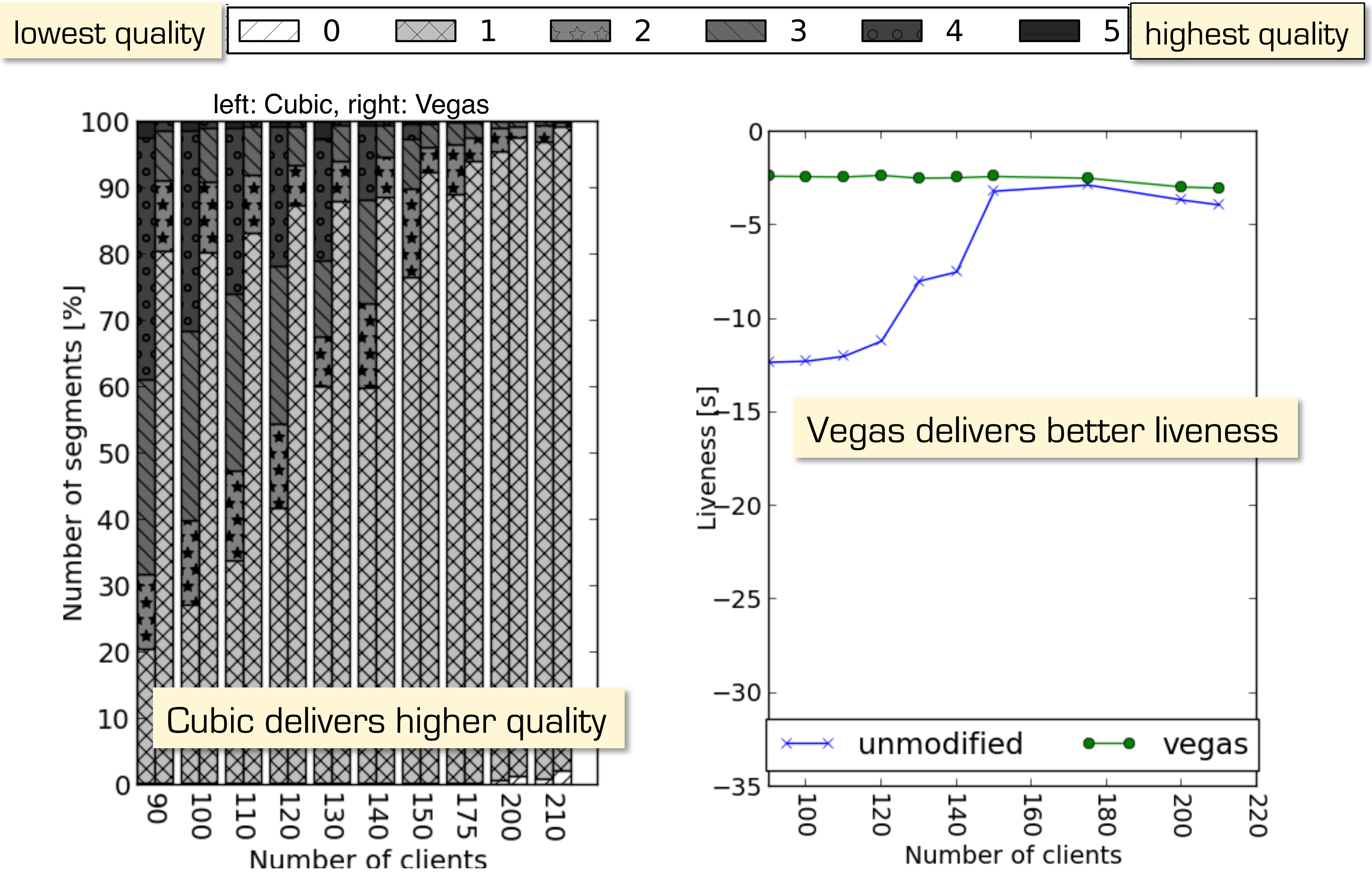
TCP congestion control alternatives



TCP Vegas shares the network much better than TCP CUBIC

Unfortunately, TCP Vegas looses to all other TCPs in sharing

TCP congestion control alternatives

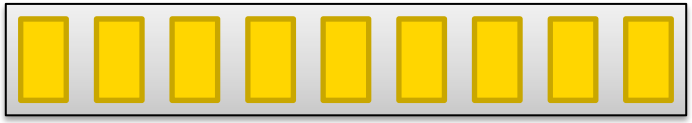


Vegas delivers better liveness

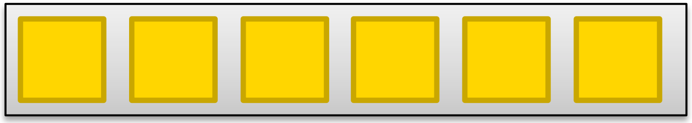
Cubic delivers higher quality

Evaluated modifications

Increasing segment length: give TCP more time



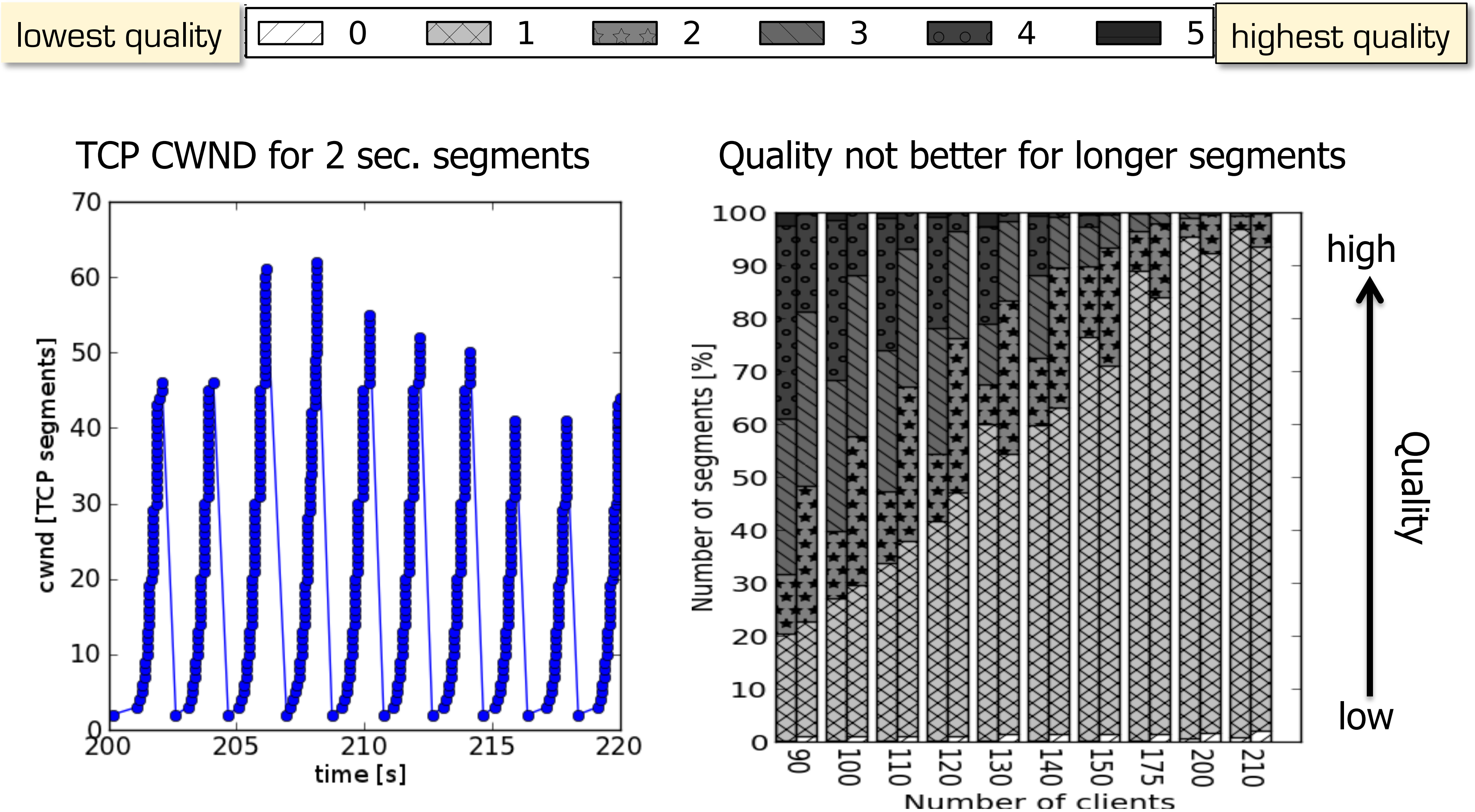
vs.



2 second segments

10 second segments

2 second vs. 10 second segments

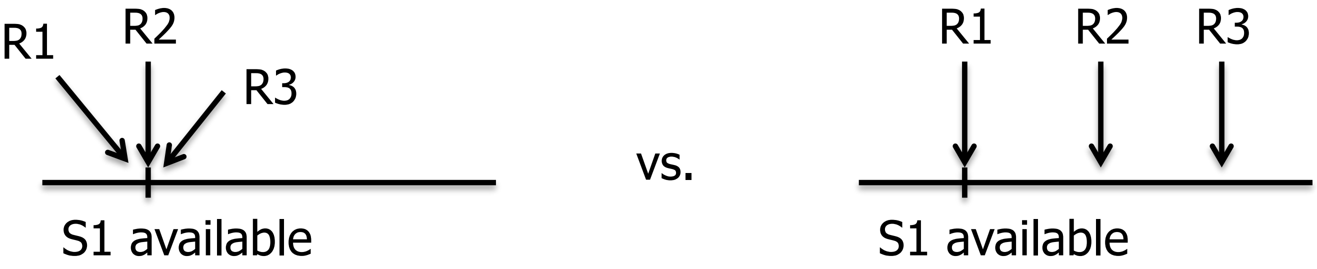


TCP CWND for 2 sec. segments

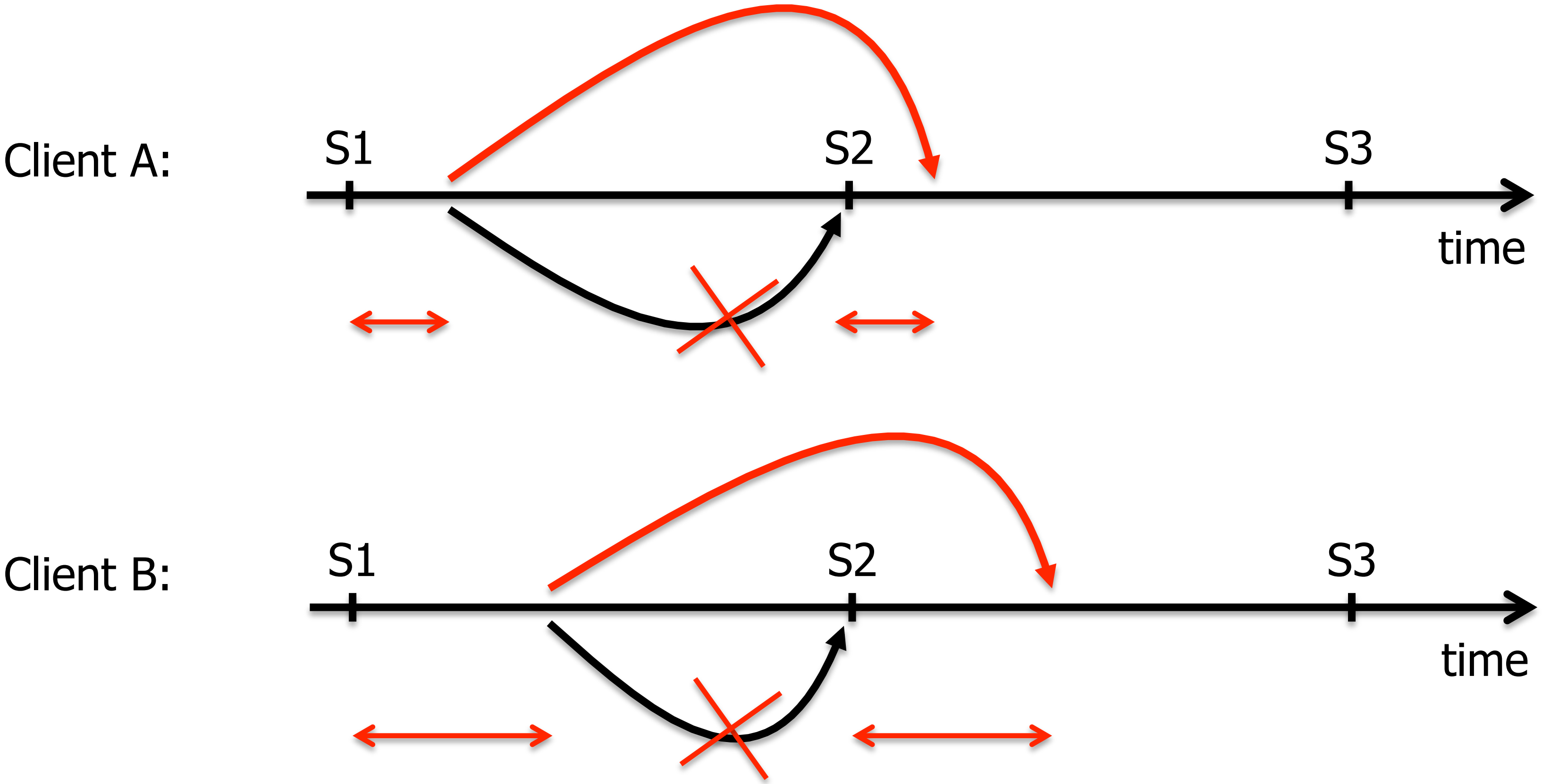
Quality not better for longer segments

Evaluated modifications

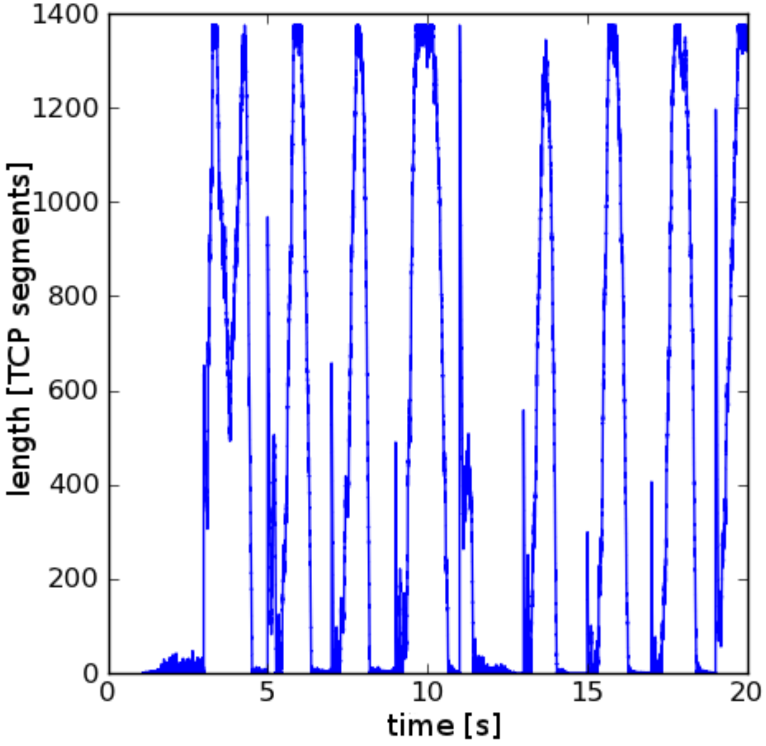
Request distribution: avoid request synchronization

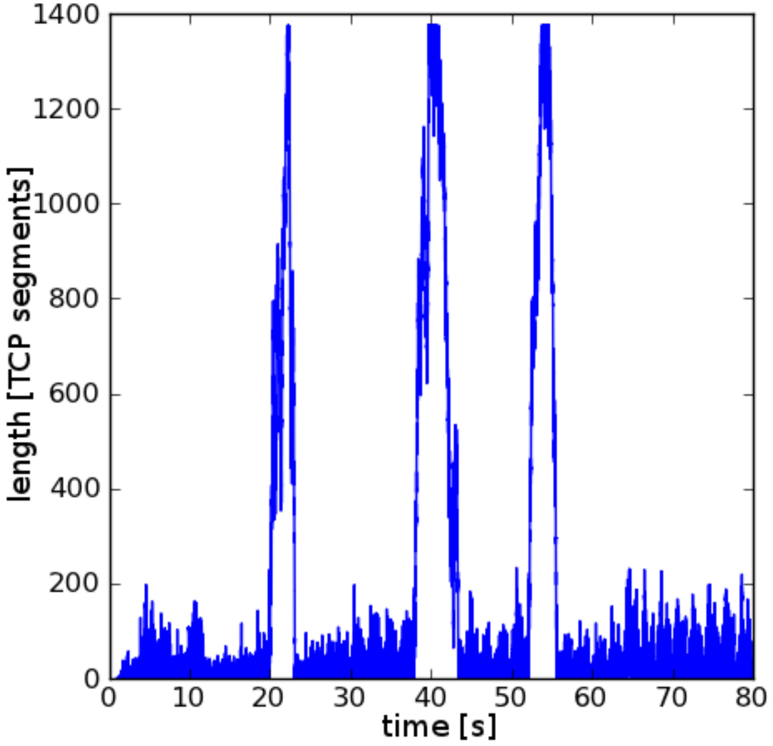


Client request distribution



Client request distribution

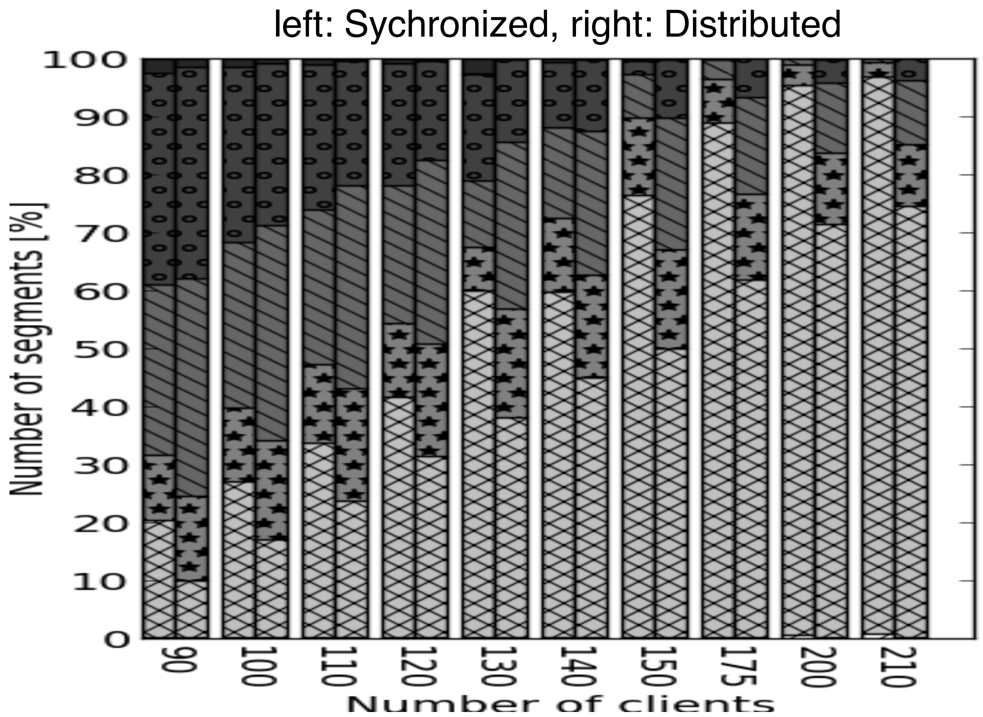

Synchronous requests


Distributed requests

Client request distribution

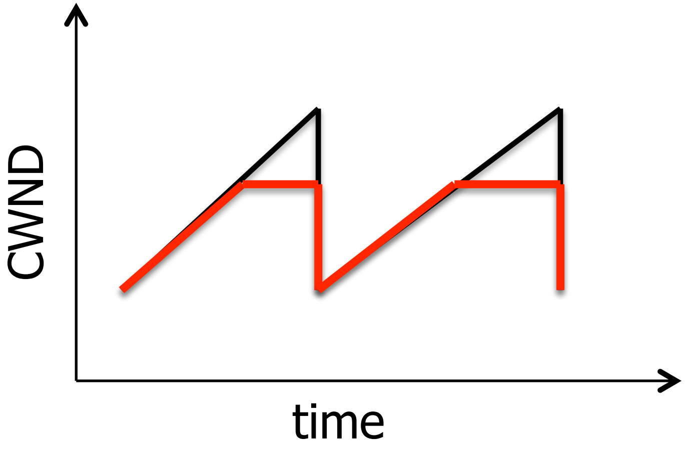
lowest quality ▨ 0 ▨ 1 ▨ 2 ▨ 3 ▨ 4 ■ 5 highest quality

Distribution of the number of segments of different quality

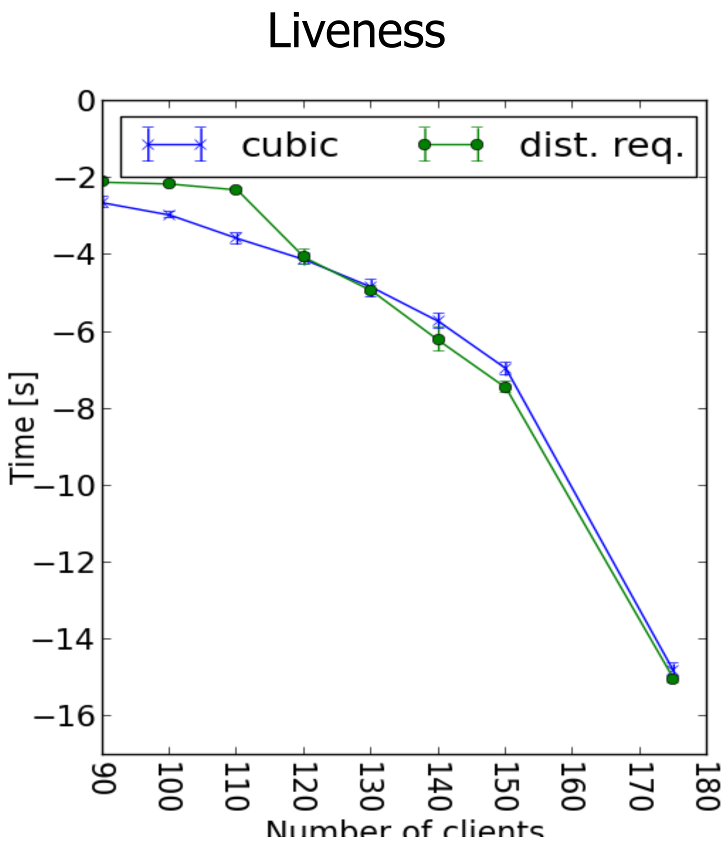


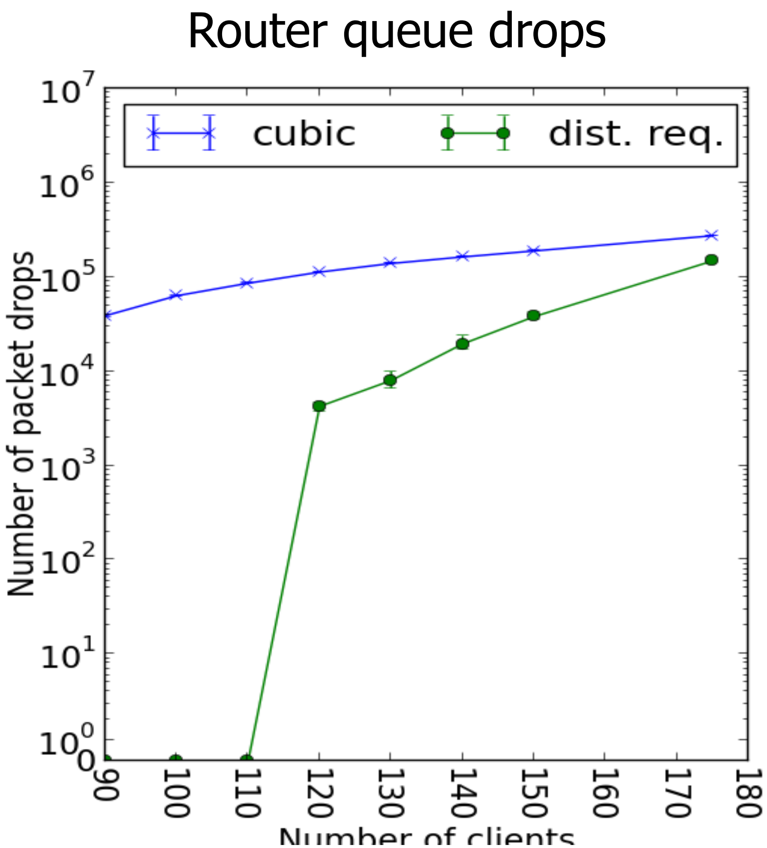
Evaluated modifications

CWND limit: restrict CWND to the max. bitrate available

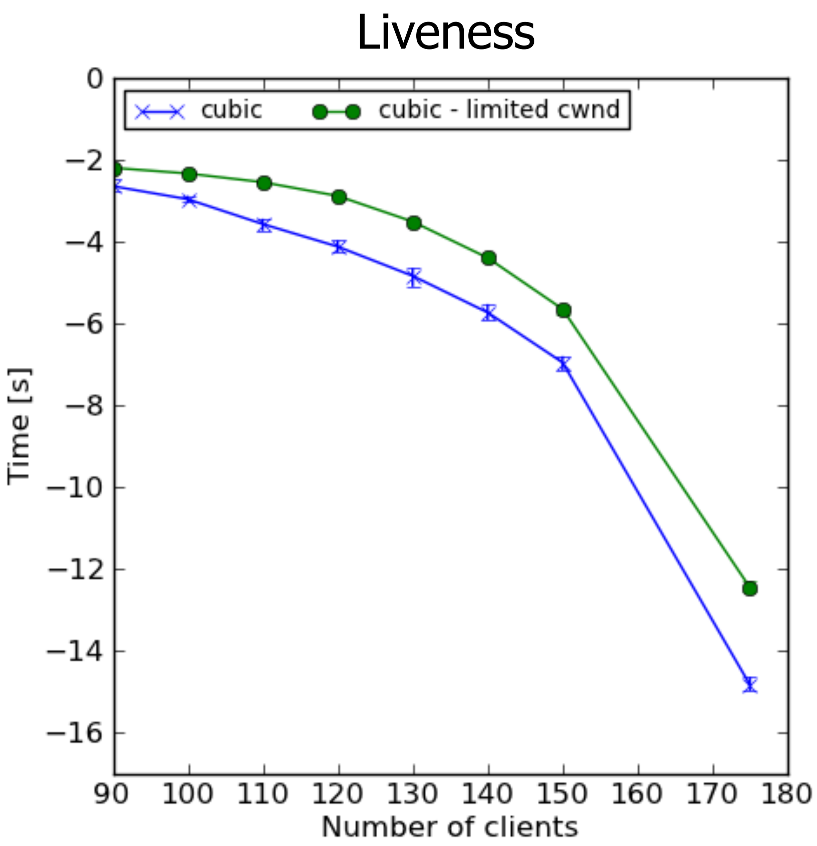


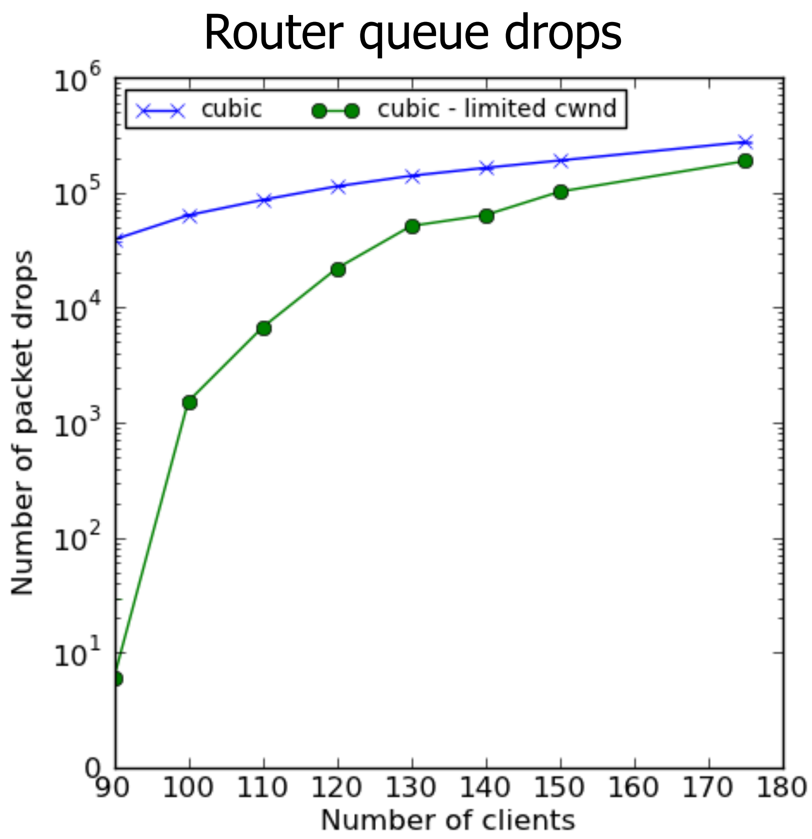
Client request distribution





CWND limitation





Conclusion

TCP CC alternatives: TCP Vegas is good, but not practical

Segment duration:
No evidence of longer segments being better (network view)

Client request distribution leads to good quality and liveness

CWDN limitation leads to better fairness

DASH Tutorial: Summary

Introduction to MPEG DASH

DASH tools

Examples of DASH (and DASH-like) systems' performance

QoE considerations

Interaction of DASH and TCP