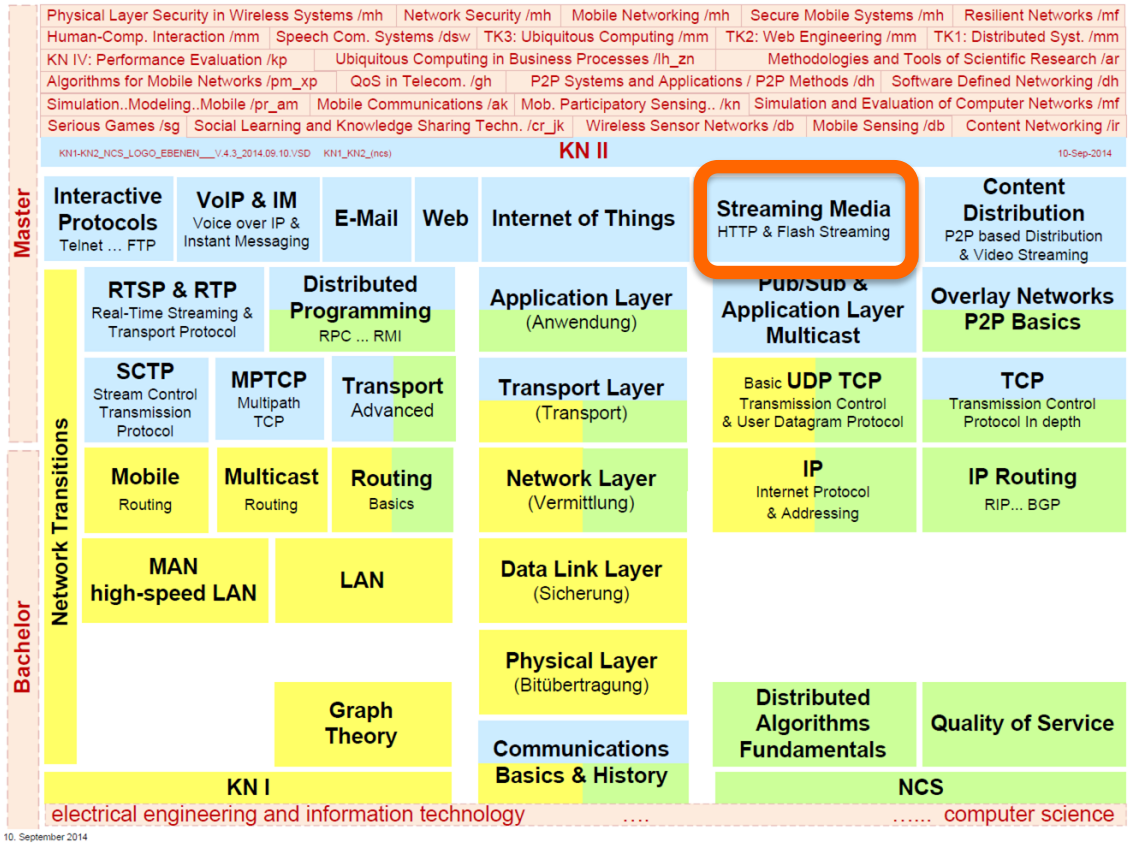
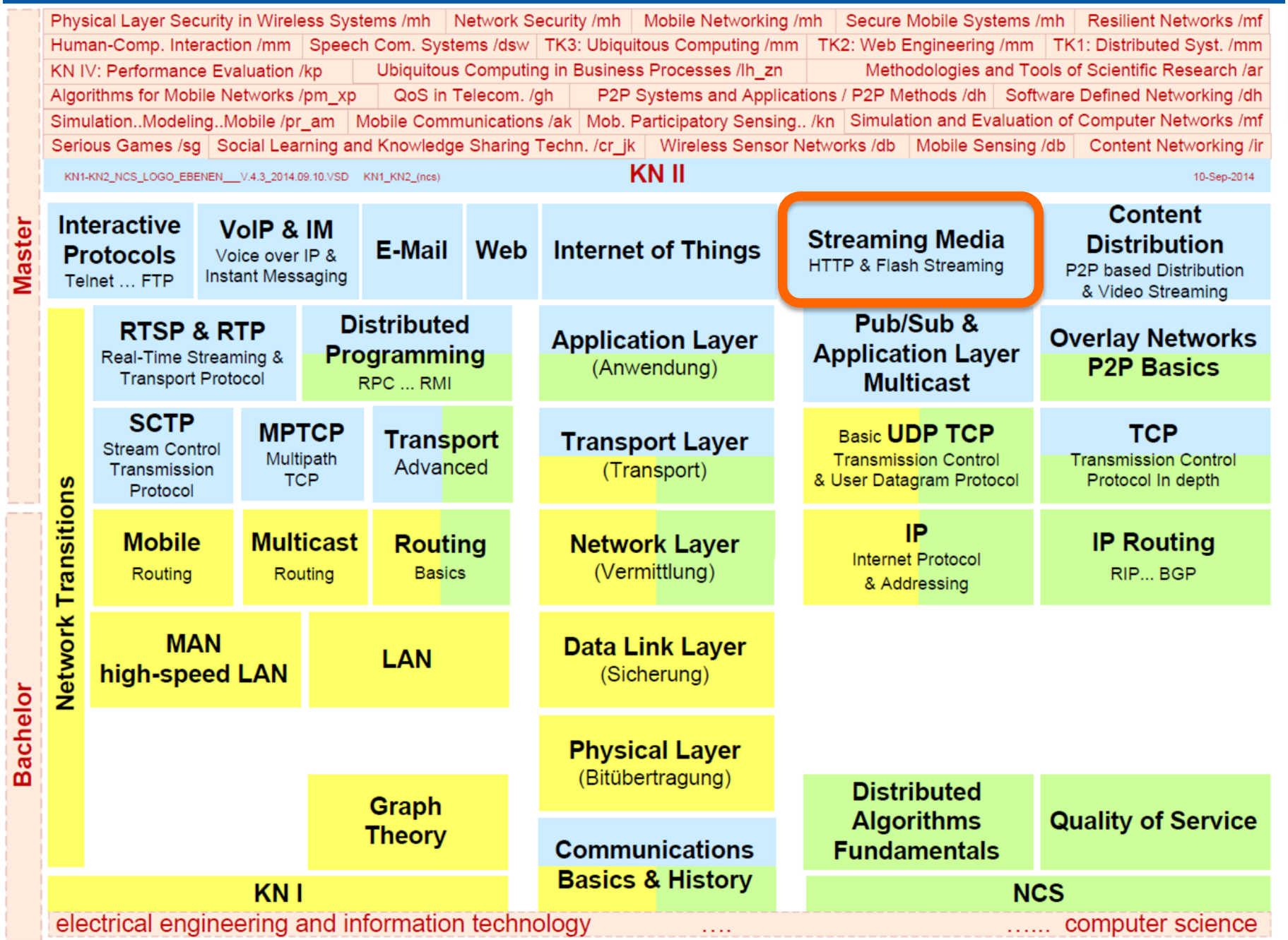
Communication Networks II

Streaming Media



Prof. Dr.-Ing. **Ralf Steinmetz**
KOM - Multimedia Communications Lab

Overview



10. September 2014

TECHNISCHE UNIVERSITÄT DARMSTADT

Big Date …

Forecast: Video & Streaming

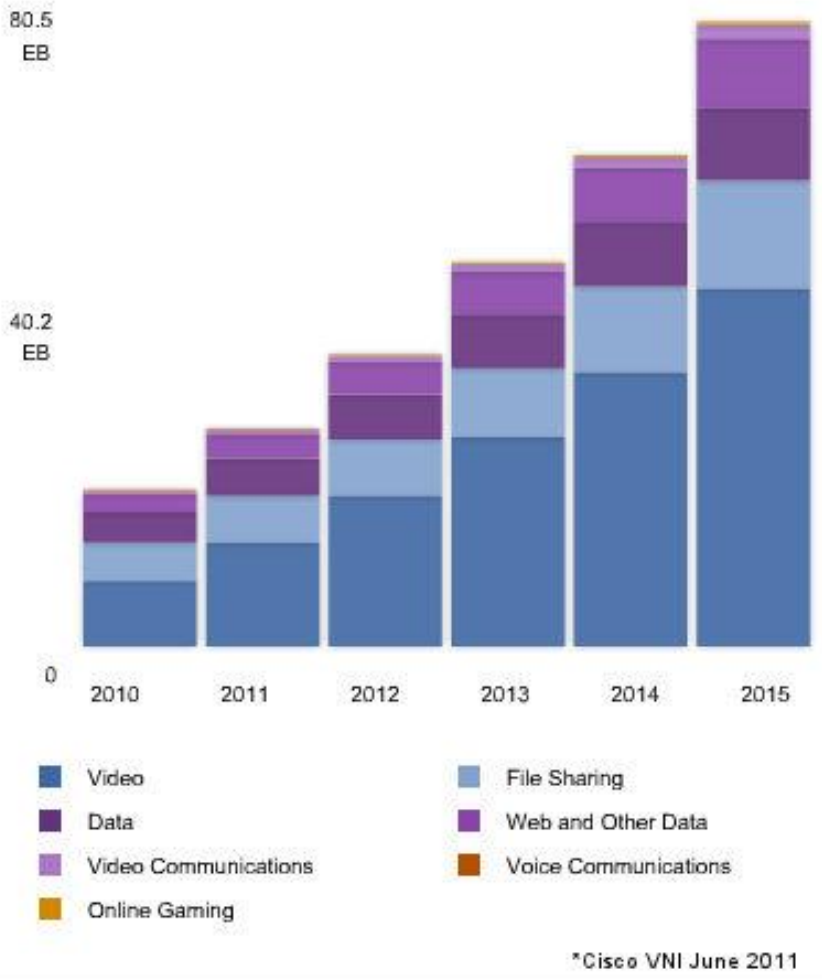
Source: http://www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index-vni/index.html#~overview,

TECHNISCHE UNIVERSITÄT DARMSTADT

Big data ..

Multiples of bytes V · T · E				
SI decimal prefixes		Binary usage	IEC binary prefixes	
Name (Symbol)	Value		Name (Symbol)	Value
kilobyte (kB)	10^3	2^{10}	kibibyte (KiB)	2^{10}
megabyte (MB)	10^6	2^{20}	mebibyte (MiB)	2^{20}
gigabyte (GB)	10^9	2^{30}	gibibyte (GiB)	2^{30}
terabyte (TB)	10^{12}	2^{40}	tebibyte (TiB)	2^{40}
petabyte (PB)	10^{15}	2^{50}	pebibyte (PiB)	2^{50}
exabyte (EB)	10^{18}	2^{60}	exbibyte (EiB)	2^{60}
zettabyte (ZB)	10^{21}	2^{70}	zebibyte (ZiB)	2^{70}
yottabyte (YB)	10^{24}	2^{80}	yobibyte (YiB)	2^{80}
See also: Multiples of bits · Orders of magnitude of data				

Source: http://en.wikipedia.org/wiki/Exabyte



Source: http://www.cisco.com/en/US/netsol/ns827/networking_solutions_sub_solution.html#%7Eforecast

Motivation

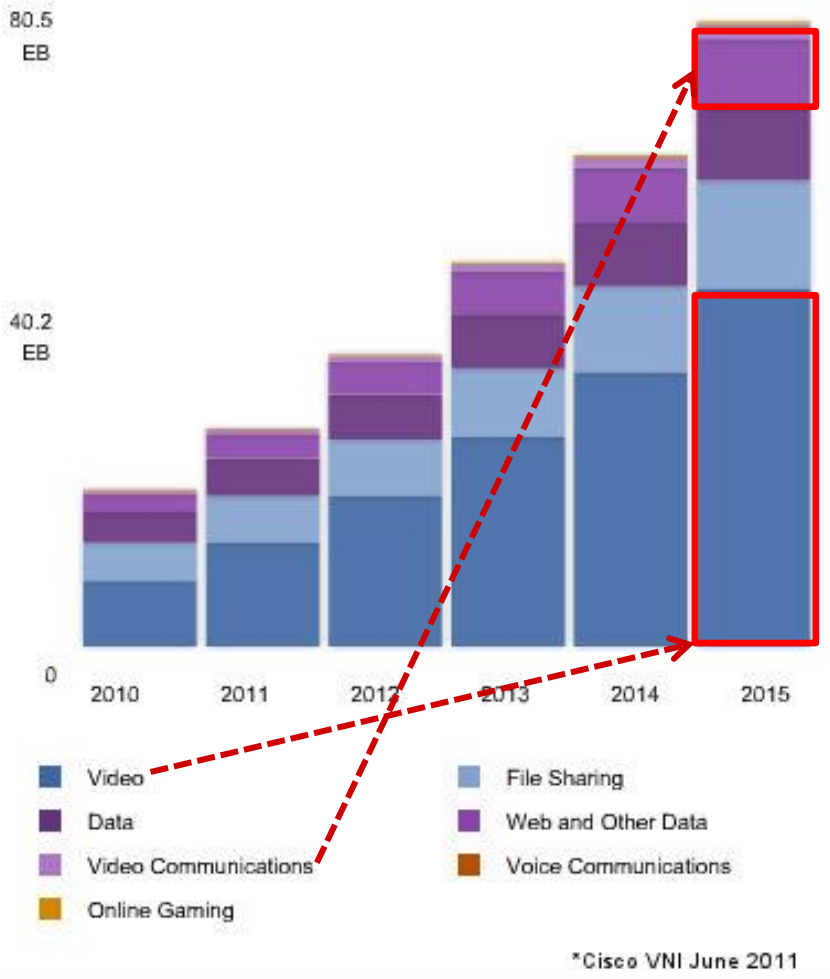
By 2015
ca. 3 billion Internet users

By 2015
ca. 15 billion global network "connections"

- fixed and mobile personal devices
- machine-to-machine "connections"

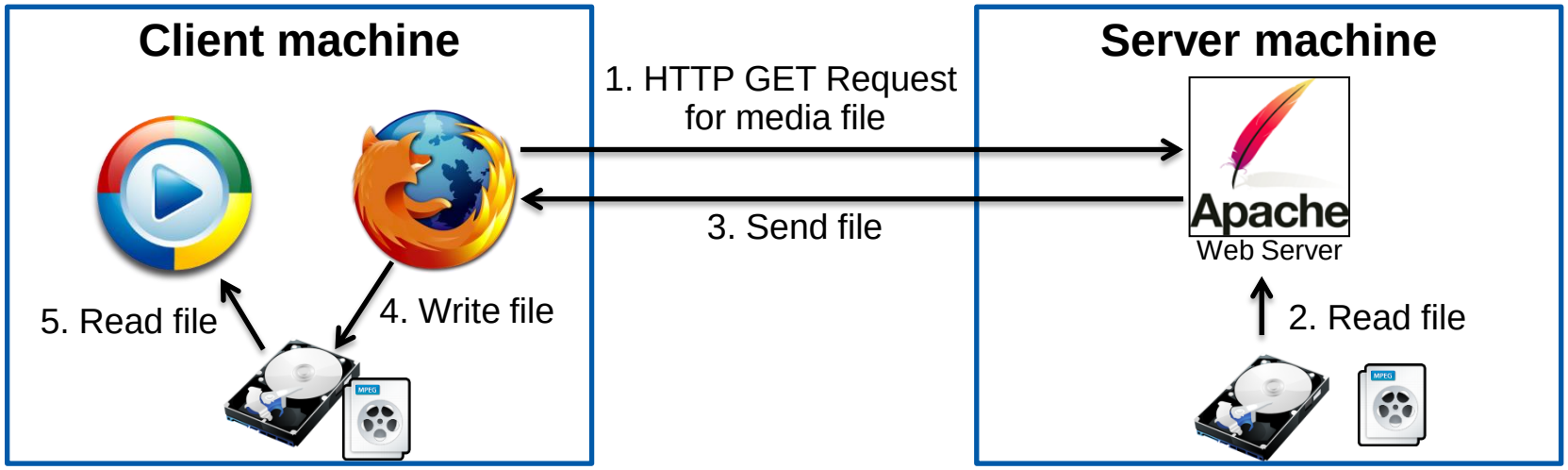
By 2015,
Total monthly traffic ca. 80.5 EB

By 2015,
more than 80% of the traffic
→ video traffic



Source: http://www.cisco.com/en/US/netsol/ns827/networking_solutions_sub_solution.html#%7Eforecast

1.2 Traditional Approach to Access Media Content

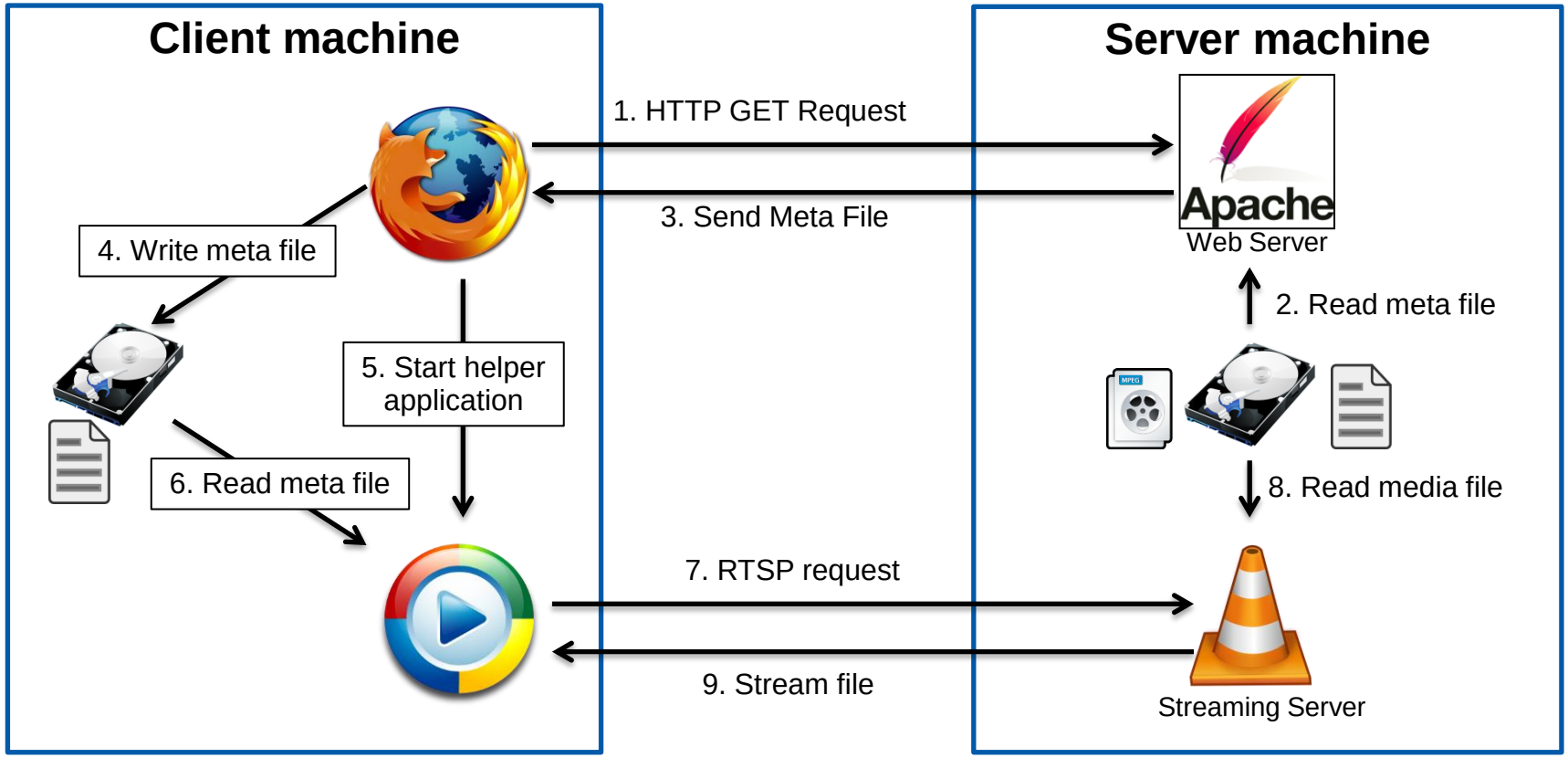


Situation and Problem:

- Phase 1: media content to be downloaded completely (before)
- Phase 2: watching it

- User has to wait a long time

1.3 Streaming Approach to Access Media Content



Advantages:

- Playback starts as soon as playout buffer is filled with data (1 - 10 sec)
- Continuous stream of data

Steps for Streaming Media

TECHNISCHE UNIVERSITÄT DARMSTADT

Compress
audio stream using an audio codec

- MP3, Ogg Vorbis, or AAC

Compress
video stream using a video codec

- H.264, MPEG-4 or V8

Assemble encoded audio and video streams in a container bitstream

- FLV, WebM, ASF or ISMA

Deliver bitstream
from a streaming server to client via transport protocol

- RTP or TCP or UDP

Streaming client may interact with the streaming server
using a control protocol

- HTTP or RTSP

TECHNISCHE UNIVERSITÄT DARMSTADT

Streaming:

- Continuous flow of media data
  - from sender
  - to receiver(s)
- Client starts playback directly after couple of seconds (1-10s)
- Server sends data with speed:
  - prefetching_factor
    - X * media_bit_rate
  - Usually prefetching_factor
    - X = ~2

Two basic types of streaming:

On-demand-streaming

- Complete media already available at the source
- Spooling possible
- Unicast-based communication

Live-streaming

- Data generated in real-time
- No spooling possible
- Multicast-based communication possible

2.1 Classification of Streaming Approaches

TECHNISCHE UNIVERSITÄT DARMSTADT

Type of streaming
- Video on Demand VoD
- Live-Streaming

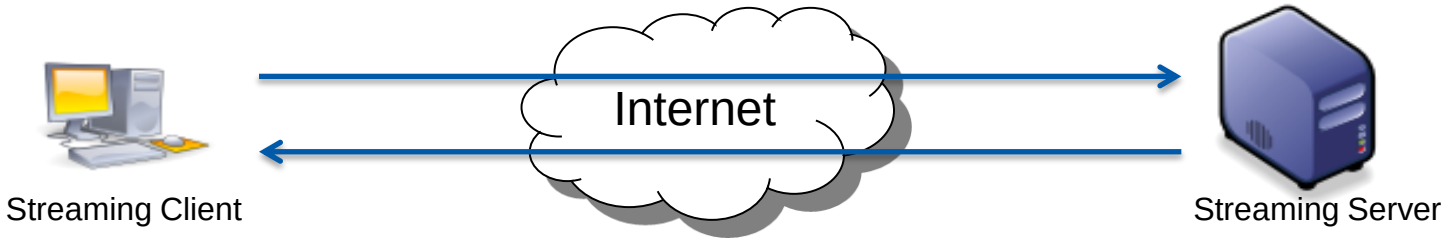
Distribution
- Client/Server
- P2P-Streaming

Streaming protocol
- HTTP
- RTSP
- Flash Streaming

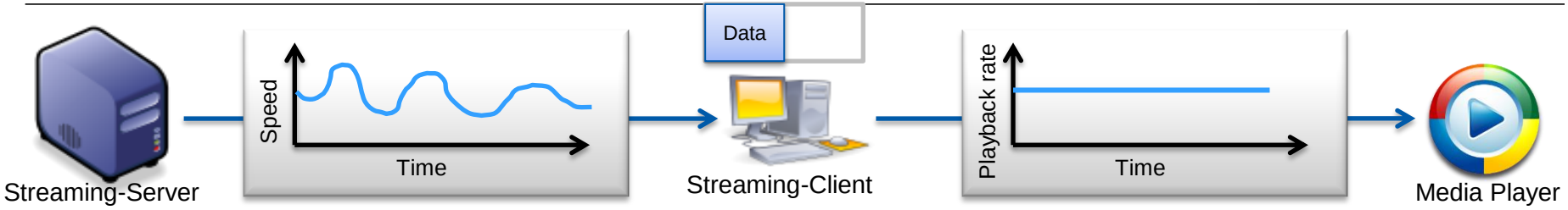
Type of Media
- Audio
- Video

Data delivery
- UDP
- TCP

Client / Device
- PC
- Mobile Device
- Set-Top-Box



Alternatives for the Delivery of Streaming Data



Via UDP with constant sending rate = playback speed

- Client directly starts playback of media data
- Drawbacks: buffer underrun possible
- Packet loss possible
  → retransmission at application layer (if possible)

Via UDP with buffer time of 2-5s

- To compensate jitter and delay in transmission
- Packet loss possible
  → retransmission at application layer (if possible)

Via TCP

- To transmit data as fast as possible
- Retransmission of data done by TCP → better QoE for user
- Starvation possible due to in order delivery of data
  → underrun of playout buffer

Mixed-Media Websites

- HTML websites with media content
  - E.g.: news sites: Spiegel.de, CNN.com

Online Media Libraries

- Video / Audio Archives
  - E.g.: ZDF-mediathek
- Copyright protected content

Video Portals

- User generated content
  - E.g.: YouTube.com, MyVideo.de

Internet Radio

- Shoutcast.com
- Phonostar

Internet Video Stores

- Flat rate
  - E.g. Amazon prime
- Pay-per view
  - E.g.: videoload.de

IP-TV

- Via Set-Top-Box
  - E.g.: T-Home Entertain

Most visited Websites of the World

TECHNISCHE UNIVERSITÄT DARMSTADT

www.KOM.tu-darmstadt.de

Google

Facebook

?

Example YouTube

Video platform for exchanging videos

History

- Founded by Chad Hurley, Steve Chen und Jawed Karim
  - in 2005
- Bought by Google
  - in 2006
  - for 1.3 billion EUR

Data - amount

- More than 1 billion unique users visit YouTube each month
- Over 6 billion hours of video are watched each month on YouTube
  - that's almost an hour for every person on Earth, and
  - 50% more than last year
- 100 hours of video are uploaded to YouTube every minute
- 70% of YouTube traffic comes from outside the US

YouTube is localized in 56 countries and across 61 languages

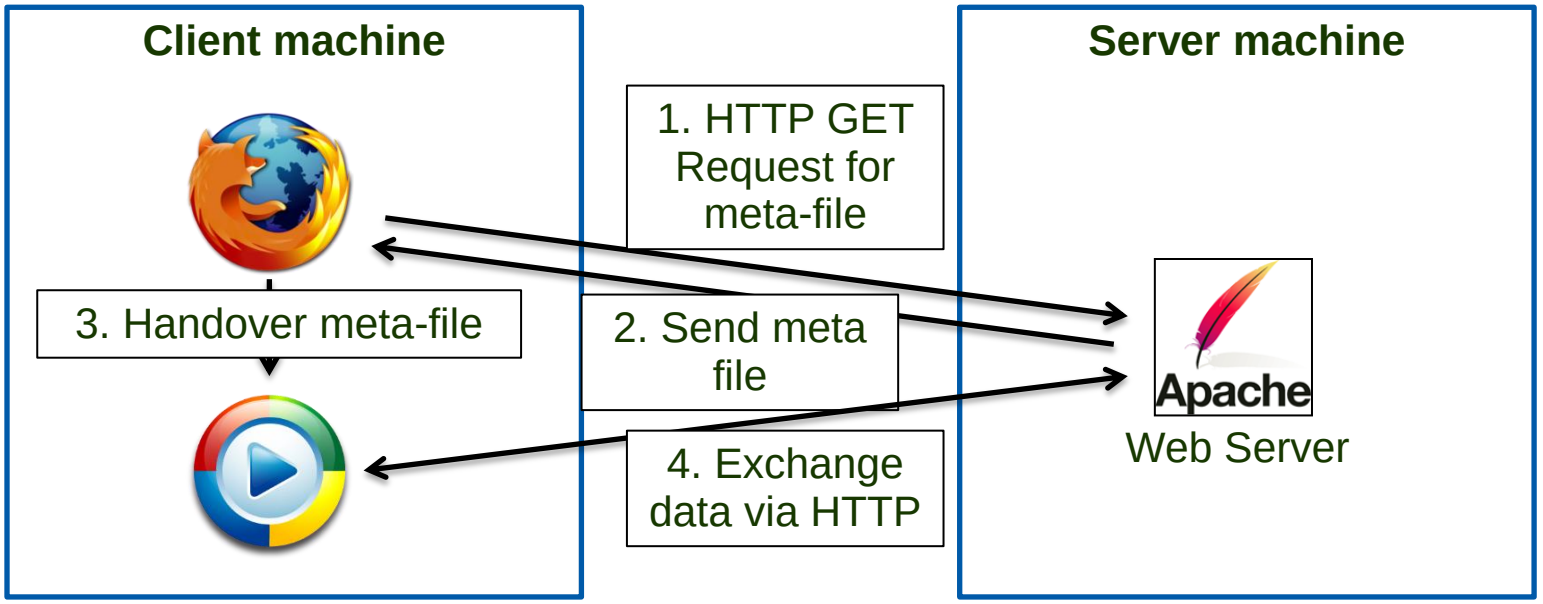
Source: http://www.youtube.com/t/press_statistics
Source: http://de.wikipedia.org/wiki/YouTube#Statistik

Alexa traffic ranking

- In late 2014
  - Ranked #3
  - Third most visited website in the world
  - Producing ~10% of the world´s data traffic in the Internet

Techniques used

- Apache Web Server / MySQL
- H.264 / Flash-Video / HTML 5 Video



Delivery of media stream via HTTP

Properties:

Delivery of media stream via HTTP

- Split up media content into a set of chunks of equal size
- Request video stream chunk by chunk

Cheap streaming alternative

- No dedicated streaming server required
- No problems with NATs because of the usage of port 80

Stream played within browser plugin

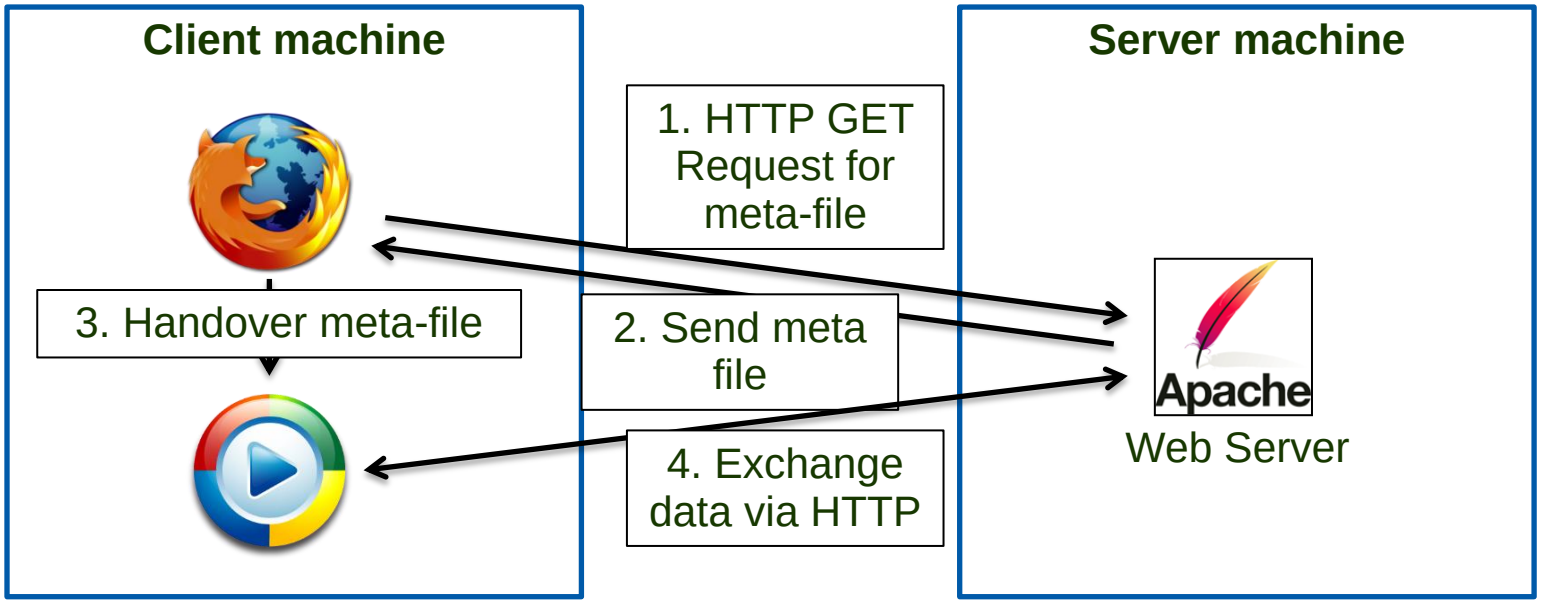
Different representations of same video content

- to be kept in separate files

Media data delivered via TCP

- using the open HTTP connection

Steps in HTTP Streaming



User clicks on link
- → **1. request metadata**
- → **2. metadata-file send back**

Browser determines suitable application based on content type
- → **start corresponding player**
- → **3. handover metadata-file**

Media Player sends HTTP-GET message
- → **4. to receive media content in response**

TECHNISCHE UNIVERSITÄT DARMSTADT

Since version 5 HTML supports streaming of media content

- Added <video> tag
- Additional optional fields

```
<video controls="controls">
  <source src="movie.mp4" type="video/mp4" />
  <source src="movie.ogg" type="video/ogg" />
  <source src="movie.webm" type="video/webm" />
Your browser does not support the video tag.
</video>
```

No further plugin required,

- codec integration in browser needed

Attribute	Value	Description
autoplay	autoplay	If present, then the video will start playing as soon as it is ready
controls	controls	If present, controls will be displayed, such as a play button
height	pixels	Sets the height of the video player
loop	loop	If present, the video will start over again, every time it is finished
muted	muted	Specifies that the audio output of the video should be muted
poster	URL	Specifies the URL of an image representing the video
preload	auto, metadata, none	Specifies whether or not the video should be loaded when the page loads
src	URL	Specifies the URL of the video file
width	pixels	Sets the width of the video player

4 HTTP Live Streaming

Problem:
Normal HTTP streaming not suitable for Live Streaming

- Data rate cannot be adjusted to meet requirements of devices
  → might lead to starving peers
  → no automatic fallback
- Quality level has to be selected manually at the beginning
- Codec / parameter change requires to restart the streaming

Solution:
Adaptive streaming solution over HTTP

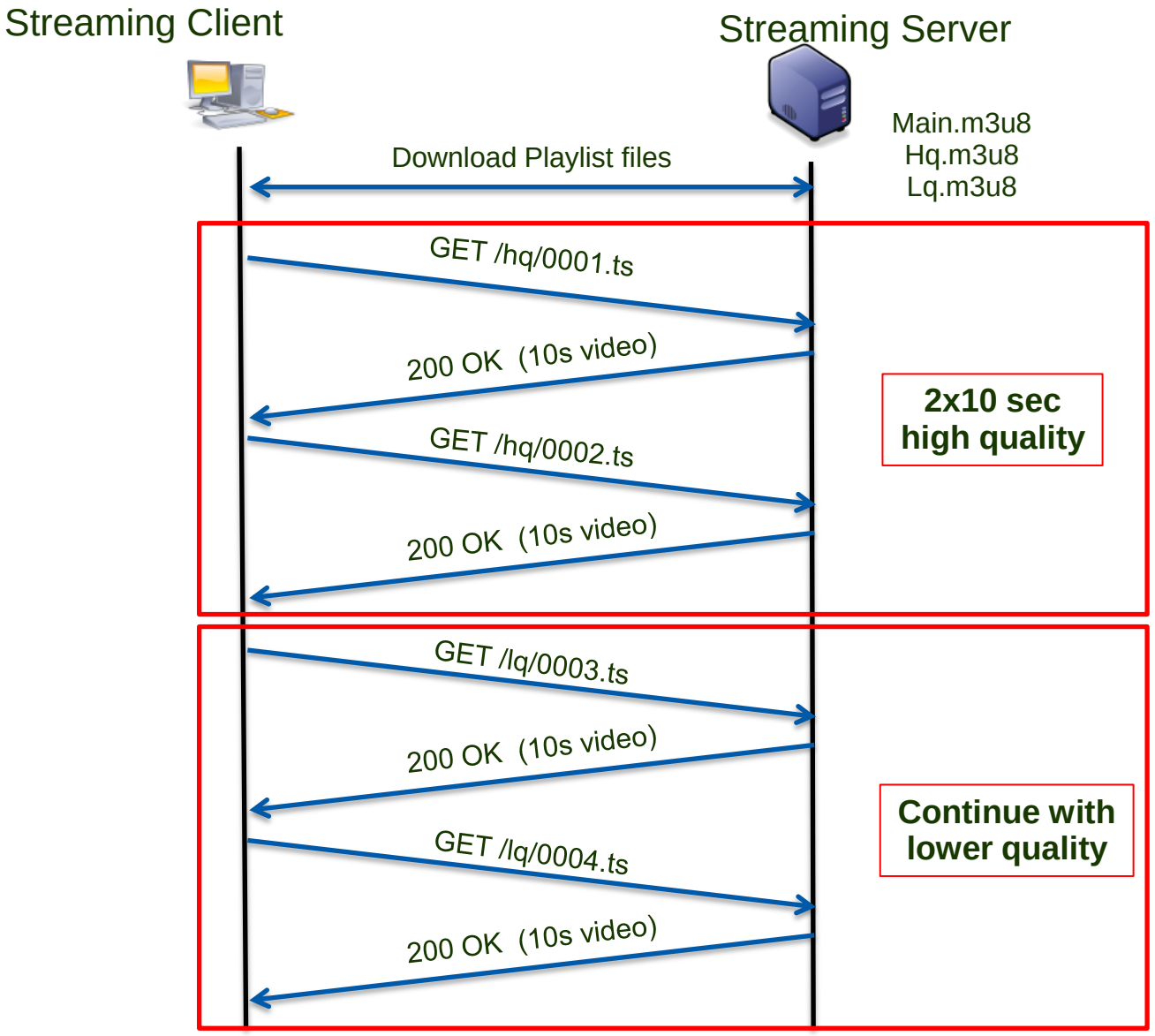
- Developed by Apple for iOS-Streaming
- Works with HTML4 and 5

Idea:
To break overall stream into a sequence of small HTTP-based file downloads

- client may select from a number of alternate streams
  - containing the same content
- same material encoded at a variety of data rates
- adapt stream to the available data rate
- start of streaming session
  - download an extended M3U
  - M3U:
    format of data for a playlist containing the metadata
- All streams transmitted as MPEG-2 transport stream

http://tools.ietf.org/html/draft-pantos-http-live-streaming-01

4.1 Interaction Between Client and Server



4.2 HTTP Live Streaming Playlist

Extends current playlist format (m3u)

- Now playlists should end with *.m3u8

e.g.

```
#EXTM3U
#EXT-X-MEDIA-SEQUENCE:0
#EXT-X-TARGETDURATION:10
#EXTINF:10,
http://a.de/hq/0001.ts
#EXTINF:10,
http://a.de/hq/0002.ts
...
#EXT-X-ENDLIST
```

HQ.m3u8

Specifies a set of new tags:

- EXT-X-TARGETDURATION
  - approx. duration of next media file
- EXT-X- MEDIA-SEQUENCE
  - unique sequence number
- EXT-X-KEY
  - information to decrypt the media (if necessary)
- EXT-X-PROGRAM-DATE-TIME
  - associates the beginning of the next media file with an absolute date/time
- EXT-X-ALLOW-CACHE
  - indicates whether the client may cache downloaded media files for later replay
- EXT-X-ENDLIST
  - End of Playlist
- EXT-X-STREAM-INF
  - next URI in the Playlist file identifies another Playlist file

EXT-X-STREAM-INF Tag

TECHNISCHE UNIVERSITÄT DARMSTADT

General notation:

- #EXT-X-STREAM-INF:[attribute=value][,attribute=value]* <URI>

Following attributes are allowed:

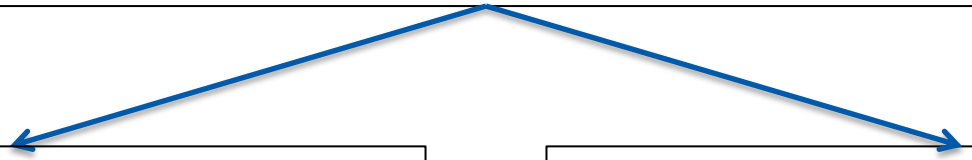
- BANDWIDTH=<n>
  - Max bandwidth of the video
- PROGRAM-ID=<i>
  - Program needed to play the media content
- CODECS="[format][,format]*„
  - The format of each media sample
  - Each media file must be

HTTP Live Streaming Playlist Example

TECHNISCHE UNIVERSITÄT DARMSTADT

Main.m3u8

```
#EXTM3U
#EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=256000
http://a.de/hq/video1.m3u8
#EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=100000
http://a.de/lq/video1.m3u8
#EXT-X-ENDLIST
```

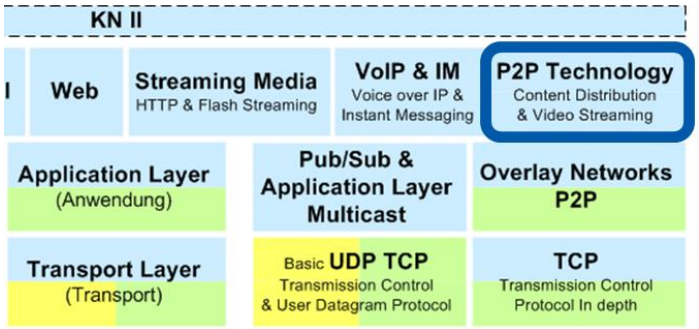


HQ.m3u8

```
#EXTM3U
#EXT-X-MEDIA-SEQUENCE:0
#EXT-X-TARGETDURATION:10
#EXTINF:10,
http://a.de/hq/0001.ts
#EXTINF:10,
http://a.de/hq/0002.ts
...
#EXT-X-ENDLIST
```

LQ.m3u8

```
#EXTM3U
#EXT-X-MEDIA-SEQUENCE:0
#EXT-X-TARGETDURATION:10
#EXTINF:10,
http://a.de/lq/0001.ts
#EXTINF:10,
http://a.de/lq/0002.ts
...
#EXT-X-ENDLIST
```



Some P2P and overlay basics needed …

- → see separate section

P2P video streaming has become

- increasingly popular approach for streaming (live) content
- many receivers, similar to IPTV

Two approaches for P2P video streaming

- Tree-based overlay
  - Push content delivery
  - Single or multiple tree
- Mesh-shaped overlay
  - Pull content delivery (swarming)
  - Like BitTorrent with modified chunk selection

Research issues

- Streaming topology
- Scalable video codecs (more bandwidth, better quality)

Additional Real-Time Messaging Protocol (RTMP) used for communication between

- Flash-Player and Flash-Server
- Transport of media content done like in HTTP Streaming
- Enables adaptive streaming like RTP / RTCP

Advantages over HTTP-Streaming

- Better support for live streaming
- Browser-independent due to the use as a plugin
- Support for user interactions
- Embedded audio- and video codecs

Drawbacks:

- Flash Plugin needs to be installed
- Proprietary (Adobe) but specification publicly available since 2008
- Today, Flash-Streaming is a de-facto standard
- In 2009:
  - 98% of the Internet users have Flash installed
  - 80% of all Internet videos are based on Flash

Flash Video - FLV

TECHNISCHE UNIVERSITÄT DARMSTADT

Container format developed by Adobe

Audio Tags

- Audio-Stream
- Codecs: MP3, Speex, Nellymoser, G.711, AAC (MP4), PCM
- Up to 44 kHz, 16 Bit

Video Tags

- Video-Stream
- Codecs: On2 VP6, AVC (H.264), Screen Video, Sorensen H.263

Data Tags (Script)

- Flash ActionScript

Meta Data

- duration, width, height, videodatarate, framerate

Representation within FLV file

TECHNISCHE UNIVERSITÄT DARMSTADT

FLV file contains different representations of the media content

- Different bitrate, codec

Selection of the representation via HTTP

- Usually via some AJAX mechanism

```
<encodings>
<type1>
<filename>48817_180x100_VP6_388.flv</filename>
<fileitem>4f41dcbf-69a2-47d2-9f57
8a53249ac7a_180x100_VP6_388.flv</fileitem>
<codec>VP6</codec>
<totalbitrate>388</totalbitrate>
<averagebitrate>384</averagebitrate>
<videobitrate>260</videobitrate>
<audiobitrate>128</audiobitrate>
<duration>116</duration>
<framerate>25.0</framerate>
</type1>
<type2>
[…]
</type9>
<encodings>
```

Flash Streaming Session (YouTube)