**Department of Computer Science National Tsing Hua University** 

### CS 5263: Wireless Multimedia Networking Technologies and Applications

### **DASH Streaming and WebRTC**

**Instructor: Cheng-Hsin Hsu** 

Some slides adopted from ACM Multimedia 2012 DASH Tutorial. We thank Christian and Carsten for sharing the slides

## **Different Ways to Deliver Multimedia Contents**



### **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

# AUT Austrian Bundesliga w.laola1.tv 💶 stream starting soon LIVE-STREAM SK Rapid Wien - FC Red Bull Salzburg Live: Fußball Arena: Bundesliga × **■**))

# **Dynamic Adaptive Streaming over HTTP**

- Cut video into segments (each lasts for a few seconds)
  - Every segment is encoded in multiple quality (general sense) levels
- Receiver requests the quality level of each segment based on the network conditions and queue status
  - Scalable, as servers are stateless
- Video segments are sent over the HTTP connections
- Widely used nowadays for 3 main reasons
  - Enable NAT/Firewall traversals
  - Capitalize existing HTTP cache/CDNs
  - TCP streaming is no longer an issue because of broadband networks ← even if we can only use ½ of the capacity, we are fine

### **DASH in a Nutshell**



# **History of DASH**



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 (standardization bodies, forums, companies,...)

#### 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)

### **Manifest and Data Files**



### What is Specified – And What is Not?



### **DASH Data Model**



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

### **Media Presentation Description**

- Meta 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
- Some other information

### **MPD Schema Overview**



### **MPD Schema - Representation**



### **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

## **Segment Indexing**



## **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



### **Profiles**

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



#### More restrictions may be added

# **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 consumer electronics decoders

| Segment<br>duration | Advantages  | Disadvantages   |
|---------------------|---|---|
| Short               | <ul> <li>Commonality with Live</li> <li>High switching granularity on segment level</li> </ul>  | <ul> <li>Large number of files</li> <li>Large number of URLs</li> <li>Fixed request size</li> <li>switching granularity on segment level</li> </ul> |
| Long                | <ul> <li>Small number of files</li> <li>Small number of URLs</li> <li>High switching granularity</li> <li>Flexible request sizes</li> <li>Improved cache performance</li> </ul> | <ul> <li>Need for Segment Index</li> <li>Difference from Live</li> </ul>  |

### **Adaptation Problem**

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



choices

### **DASH Encoders, Datasets, and Players**

- "Encoder": GPAC
- Datasets: Big Buck Bunny and so on...
- Players: VLC media player plugin, libdash



# 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



jean.lefeuvre@telecom-paristech.fr

### **DASH Encoder**

### 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/

### **Three Steps of DASH Encoder**



### Datasets

#### 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

# **Popular 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     |

### **Popular Sequences (cont.)**



walkaama ch

### **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 Clients**

### 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



jean.lefeuvre@telecom-paristech.fr

http://gpac.sourceforge.net

## **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



### **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.



### **Summary: Pull-Based DASH Streaming**

### 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: <u>http://dash.itec.aau.at</u> | <u>http://gpac.sourceforge.net</u>

### **WebRTC: Real-Time Communications**

### **<u>Web</u>** Browsers with <u>Real-Time-Communication</u>

- Audio/Video Chat on the web
- Accessed through Javascript API
- Does not require plugins, downloads or installs
- Multiple browsers, multiple platforms
- Good NAT/Firewall traversal supports
- Based on UDP streaming

http://www.webrtc.org/faq



## **The Origin of WebRTC**



Source: jimmylee.info

## **WebRTC Lowers the Barriers**



- Circuit-switched
- Electronic devices
- Dedicated lines



- SIP and IP
- Standard protocols
- IMS core for carriers
- Complex infrastructure

P2P

- **IP** 



- Client software
- Walled garden
- WebRTC



- HTML5
- No plug-in
- No client software
- Hopefully interoperate

### **WebRTC Signaling Triangle**



### **WebRTC Signaling Trapezoid**



### Architectures

- Peer-to-peer: phone call
- Full mesh: (small) conference call
- Star: (medium) conference call
- MCU: (large) conference call





### **WebRTC Protocols**



### **WebRTC Related RFCs**

- ICE: Interactive Connectivity Establishment (RFC 5245)
- STUN: Session Traversal Utilities for NAT (RFC 5389)
- TURN: Traversal Using Relays around NAT (RFC 5766)
- SDP: Session Description Protocol (RFC 4566)
- XMPP: Extensible Messaging and Presence Protocol (RFC 3921)
- DTLS: Datagram Transport Layer Security (RFC 6347)
- SCTP: Stream Control Transport Protocol (RFC 4960)
- SRTP: Secure Real-Time Transport Protocol (RFC 3711)

### **Interactive Connectivity Establishment: ICE**

- A framework for connecting peers, it tries to find the best path for each call
  - Direct
  - STUN (Session Traversal Utilities for NAT)
  - TURN (Traversal Using Relays around NAT)



### **How NAT Traversal Works**



### **Three Main Tasks and JavaScript APIs**

### Main Tasks

- Acquire audio and video
- Transferring audio and video
- Transferring arbitrary data

### JavaScript APIs

- MediaStream (getUserMedia)
- **RTCPeerConnection**
- RTPDataChannel
- Details on APIs are left as exercise....

### **Summary: Push-Based WebRTC**

- Web Real-Time Communications
- Standards to enable <u>browser</u> based sessions (voice, video, collaborations, ...) without the need of custom clients or plugins
- Builds on HTLM5 capabilities with JavaScript
- Standardized by W3C and IETF
  - IETF RTCWeb WG (Internet world, IP protocols)
  - W3C WebRTC WG (web world, Browsers etc.)
- Intended for all browsers to support
  - Microsoft being problematic
    - Have their own CU-RTC-Web framework
  - Apple (Safari) not at the table

