Multimedia im Netz

Online Multimedia

Winter semester 2015/16

Tutorial 12 – Minor Subject
Today’s Agenda

• Discussion of Assignment 11
• Streaming
  – Reading & Discussion: Why YouTube dropped Flash
  – Streaming in HTML
• Quiz
Assignment 11

Segment Image

Is composed of

Component: People

Is composed of

Component: Woman

Right of

Component: Man

In front of

Component: Trashcan

Component: Wall

Is composed of

Component: “Ball and Chain”

above

Component: “Fascist Ronald”

Component: “HappyCat”

Right of
Reading

YouTube drops Flash for HTML5 video as default

http://www.theverge.com/2015/1/27/7926001/youtube-drops-flash-for-html5-video-default
Comprehension

• Discuss with the person next to you:
  – What are the reasons for which YouTube made the switch?
  – What is the idea behind Adaptive Bitrate (ABR)?
  – Why does it reduce buffering?
  – Why is reducing buffering a desirable?
  – What does “heavily congested networks” mean?

• Not mentioned in the text:
  – What kind of streaming models are there?
  – Which model does the HTML5 video player use?
Streaming Media in HTML

• Before the introduction of the <video> element, you needed things like Flash
• Specifying the “type” attribute improves performance. If you do not specify it, the browser will start loading all source files until it finds one that it can play back.

Going Responsive

- It makes sense to use smaller video sources if the output device can’t even display the full size of the original video.

```html
<video controls autoplay>
  <source src="video-small.mp4" type="video/mp4" media="(max-width: 699px)" />
  <source src="video.mp4" type="video/mp4" media="(min-width: 700px)" />
</video>
```
Cross Browser Issue

• Problem:
The solution from the previous slide doesn’t work with the current version of Chrome (and maybe other browsers)

• How do we fix this?

```javascript
var baseURL = 'http://www.medien.ifi.lmu.de/lehre/ws1516/mmn/uebung/material/';
var sources = {
    small : baseURL + 'w3c-webstandards-small.mp4',
    large : baseURL + 'w3c-webstandards.mp4'
};

function updateVideoSource(){
    if(window.innerWidth > 699 && video.src != sources.large){
        video.src = sources.large;
    } else if(window.innerWidth < 700 && video.src != sources.small){
        video.src = sources.small;
    }
}

window.addEventListener('resize',updateVideoSource);
updateVideoSource();
```
Breakout: Media Events

• Goal:
  Understand how the browser fetches and buffers the stream

• What to do:
  1. Attach multiple event listeners to a <video> element
  2. These are the relevant events: canplaythrough, loadstart, canplay, loadedmetadata, loadeddata, ratechange, progress
  3. Produce some kind of output when any of the event occurs (e.g. a console.log, or setting the innerHTML of a <div>

• Time frame: 15 minutes.
Determining the Buffer State

```javascript
video.addEventListener('progress', function(e){
    try{
        if(video.buffered.end(0) != bufferEnd){
            bufferEnd = video.buffered.end(0);
            console.log(bufferEnd);
        }
    }
    catch(e){
        // do nothing
    }
})
```
Breakout: Buffer full

• Inform the user once the buffer is full
• In other words, check if the buffer end has reached the total duration of the video

• Time frame: 15 Minutes
Round-Up Quiz

1. Why is HTTP streaming (DASH) “such a big deal”?
2. What are drawbacks of HTTP streaming?
3. What are drawbacks of the <video> element?
Thanks!
What are your questions?