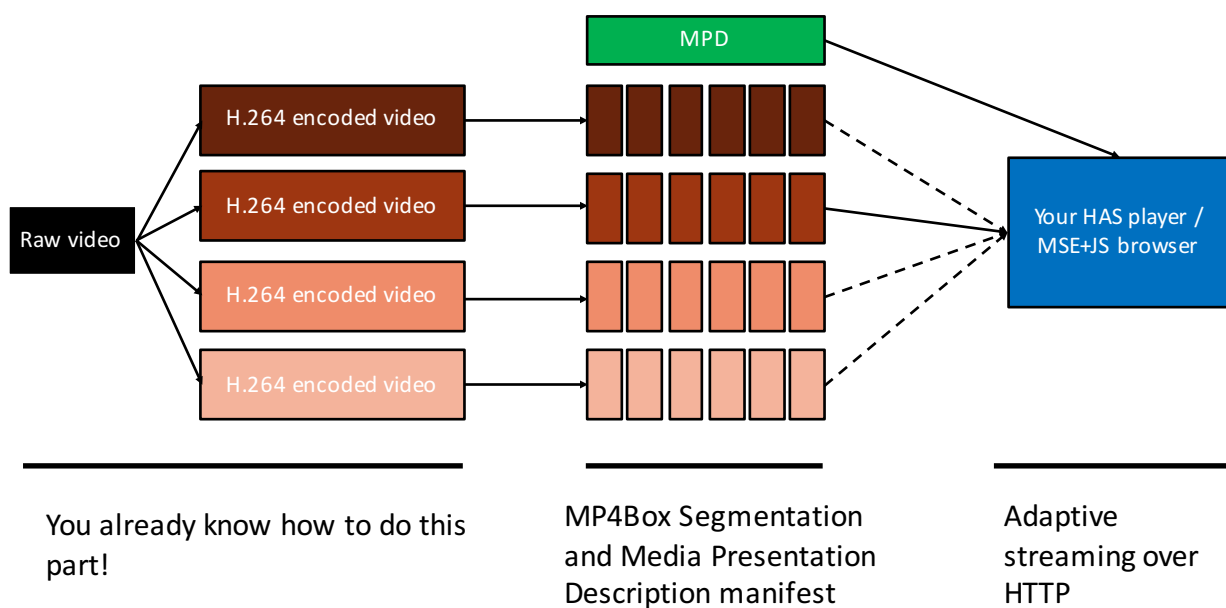


HTTP Adaptive Streaming (HAS) in practice

Architecture (MPEG DASH)



Encode multi-bitrate video streams

Replace "ed.avi" with your source video

```
ffmpeg -i ed.avi -r 24 -b:v 5000k -s 1280x720 -c:v libx264 -x264opts keyint=48:fps=24:min-keyint=48:scenecut=0:no-scenecut:pass=1 h264video720_5000k.mp4
```

```
ffmpeg -i ed.avi -r 24 -b:v 2000k -s 1280x720 -c:v libx264 -x264opts keyint=48:fps=24:min-keyint=48:scenecut=0:no-scenecut:pass=1 h264video720_2000k.mp4
```

```
ffmpeg -i ed.avi -r 24 -b:v 1000k -s 1280x720 -c:v libx264 -x264opts keyint=48:fps=24:min-keyint=48:scenecut=0:no-scenecut:pass=1 h264video720_1000k.mp4
```

```
ffmpeg -i ed.avi -r 24 -b:v 500k -s 1280x720 -c:v libx264 -x264opts keyint=48:fps=24:min-keyint=48:scenecut=0:no-scenecut:pass=1 h264video720_500k.mp4
```

```
ffmpeg -i ed.avi -r 24 -b:v 200k -s 1280x720 -c:v libx264 -x264opts keyint=48:fps=24:min-keyint=48:scenecut=0:no-scenecut:pass=1 h264video720_200k.mp4
```

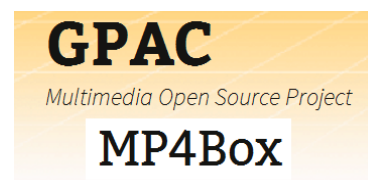
Encode multi-bitrate video streams

x264 configurations to ensure that every 2 seconds of content is independently decodable.

--fps 24	Specifies the framerate which shall be used, here 24 frames per second.
--keyint 48	Sets the maximum interval between keyframes. This setting is important as we will later split the video into segments and at the beginning of each segment should be a keyframe. Therefore, --keyint should match the desired segment length in seconds multiplied with the frame rate. Here: 4 seconds * 24 frames/seconds = 96 frames.
--min-keyint 48	Sets the minimum interval between keyframes. See --keyint for more information. We achieve a constant segment length by setting minimum and maximum keyframe interval to the same value and furthermore by disabling scenecut detection with the --no-scenecut parameter.
--no-scenecut	Completely disables adaptive keyframe decision.

Preparing media segments - MP4Box

- The multimedia packager available in GPAC is called **MP4Box**.
- It can be used for performing many manipulations on multimedia files like AVI, MPG, TS.
- Manipulating ISO files like MP4, 3GP: adding, removing, multiplexing audio, video and presentation data (including subtitles) from different sources and in different formats,
- for performing encryption of streams
- for attaching metadata to individual streams or to the whole ISO file to produce MPEG-21 compliant or hybrid MPEG-4/MPEG-21 files
- **for preparation of HTTP Adaptive Streaming content**



Preparing media segments - MP4Box

Download binary from: <https://gpac.wp.mines-telecom.fr/downloads/gpac-nightly-builds/>

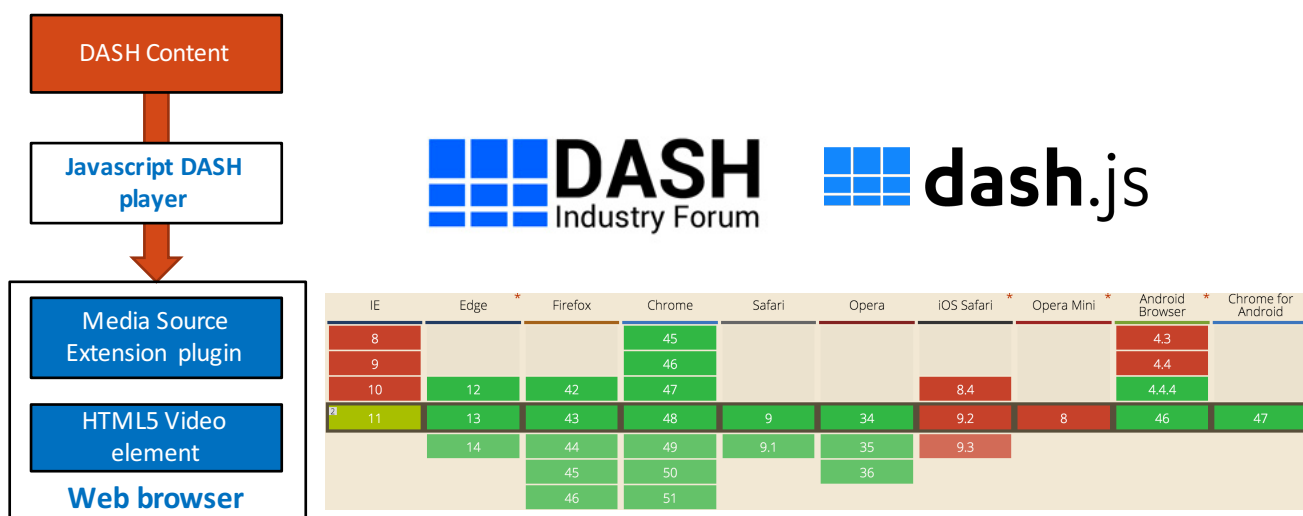
MP4Box -dash 2000 -frag 2000 -rap -segment-name video720_5000k_ h264video720_5000k.mp4

-dash 2000	Segments the given file into 2000ms chunks.
-frag 2000	Creates subsegments within segments. By setting it to the same value, there will only one subsegment per segment.
-rap	Forces segments to start random access points, i.e. keyframes. Segment duration may vary due to where keyframes are in the video – that's why we (re-) encoded the video before with the appropriate settings!
-segment-name segment_	The name of the segments. An increasing number and the file extension is added automatically. So in this case, the segments will be named like this: segment_1.m4s, segment_2.m4s, ...
h264video720_5000k.mp4	The video we have created just before which should be segmented.

Preparing media segments - MP4Box

- Create a new folder in **public_html** and name it **ed**
- Copy *.mpd, *_init.mp4, and *.m4s generated by MP4Box to **ed**
- Now the content is ready
- Let's setup the player

MPEG DASH playback

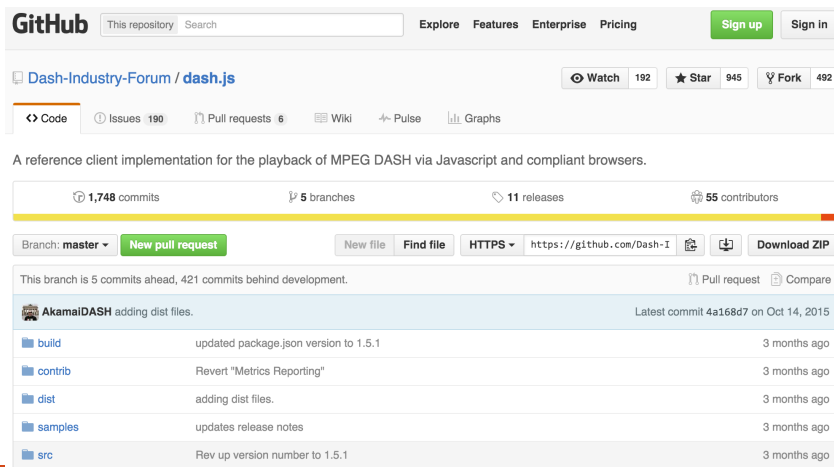


<http://caniuse.com/#search=mse>

DASH IF player



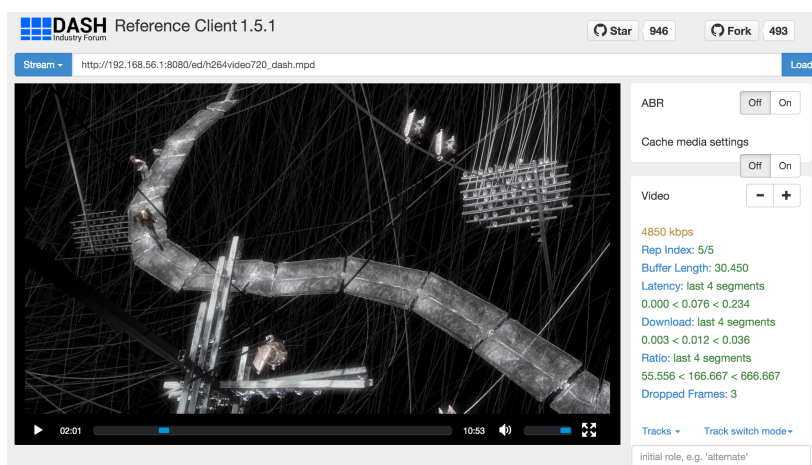
- Download from Github:
- <https://github.com/Dash-Industry-Forum/dash.js/tree/master>
- Extract everything from the downloaded package and copy them to folder **ed**.



Play the DASH Content

OR <http://192.168.56.2>

- Make sure your virtual server is running (*Vagrant up*), check <http://192.168.56.1:8080>
- Go to <http://192.168.56.1:8080/ed/samples/dash-if-reference-player/index.html>
- Type [http://192.168.56.1:8080/ed/\[the name of your MPD file\].mpd](http://192.168.56.1:8080/ed/[the name of your MPD file].mpd) in the textbox and click Load.



What does your MPD file look like?

```
<?xml version="1.0"?>
<!-- MPD file Generated with GPAC version 0.5.2-DEV-rev1165-g2fcb210-master at 2016-01-24T01:04:21.683Z-->
<MPD xmlns="urn:mpeg:dash:schema:mpd:2011" minBufferTime="PT1.500S" type="static"
mediaPresentationDuration="PT0H10M53.792S" maxSegmentDuration="PT0H0M2.000S" profiles="urn:mpeg:dash:profile:full:2011">
  <ProgramInformation moreInformationURL="http://gpac.sourceforge.net">
    <Title>h264video720_200k_dash.mpd generated by GPAC</Title>
  </ProgramInformation>
  <Period duration="PT0H10M53.792S">
    <AdaptationSet segmentAlignment="true" maxWidth="1280" maxHeight="720" maxFrameRate="24" par="16:9" lang="und">
      <Representation id="1" mimeType="video/mp4" codecs="avc3.64001f" width="1280" height="720" frameRate="24" sar="1:1"
startWithSAP="1" bandwidth="195903">
        <SegmentList timescale="12288" duration="24576">
          <Initialization sourceURL="video_720_200k_init.mp4"/>
          <SegmentURL media="video_720_200k_1.m4s"/>
          ...
          <SegmentURL media="video_720_200k_327.m4s"/
        </SegmentList>
      </Representation>
    </AdaptationSet>
  </Period>
</MPD>
```

More representations

- Its not adaptive streaming if we have only one video representation and no audio track.
- Add video in 5 different bitrates

MP4Box -dash 2000 -frag 2000 -rap -segment-name video720_200k_ h264video720_200k.mp4

MP4Box -dash 2000 -frag 2000 -rap -segment-name video720_500k_ h264video720_500k.mp4

MP4Box -dash 2000 -frag 2000 -rap -segment-name video720_1000k_ h264video720_1000k.mp4

MP4Box -dash 2000 -frag 2000 -rap -segment-name video720_2000k_ h264video720_2000k.mp4

MP4Box -dash 2000 -frag 2000 -rap -segment-name video720_5000k_ h264video720_5000k.mp4

- Add audio (audio.mp4 is the one you encoded before in AAC codec).

MP4Box -dash 2000 -frag 2000 -rap -segment-name audio_ audio.mp4

Let's put everything together

- Copy *.mpd, *_init.mp4, and *.m4s generated by MP4Box to **ed**
- Duplicate 200k.mpd and rename it ed.mpd
- Open ed.mpd using a text editor
- Within the <AdaptationSet></AdaptationSet> block, you'll find one <Representation></Representation> block.
- Now open 500k.mpd, copy the <Representation></Representation> block and add it after the original <Representation></Representation> block in ed.mpd (give it a new ID). Do the same for 1000k.mpd, 2000k.mpd, and 5000k.mpd
 <Representation id="2" mimeType="video/mp4" codecs="avc3.64001f" width="1280" height="720" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="487536">...</Representation>
- Open audio.mpd, copy the <AdaptationSet></AdaptationSet> block, and add it after the original <AdaptationSet></AdaptationSet> block in ed.mpd

What does your MPD file look like?

```
...
<Period duration="PT0H10M53.792S">
  <AdaptationSet segmentAlignment="true" maxWidth="1280" maxHeight="720" maxFrameRate="24" par="16:9" lang="und">
    <Representation id="1" mimeType="video/mp4" codecs="avc3.64001f" width="1280" height="720" frameRate="24" sar="1:1"
    startWithSAP="1" bandwidth="195903">... </Representation>
    <Representation id="2" mimeType="video/mp4" codecs="avc3.64001f" width="1280" height="720" frameRate="24" sar="1:1"
    startWithSAP="1" bandwidth="487536">... </Representation>
    <Representation id="3" mimeType="video/mp4" codecs="avc3.64001f" width="1280" height="720" frameRate="24" sar="1:1"
    startWithSAP="1" bandwidth="976717">... </Representation>
    <Representation id="4" mimeType="video/mp4" codecs="avc3.64001f" width="1280" height="720" frameRate="24" sar="1:1"
    startWithSAP="1" bandwidth="1953805">... </Representation>
    <Representation id="5" mimeType="video/mp4" codecs="avc3.64001f" width="1280" height="720" frameRate="24" sar="1:1"
    startWithSAP="1" bandwidth="4849802">... </Representation>
  </AdaptationSet>
  <AdaptationSet segmentAlignment="true" lang="und">
    <Representation id="1" mimeType="audio/mp4" codecs="mp4a.40.2" audioSamplingRate="48000" startWithSAP="1"
    bandwidth="198163">...</Representation>
  </AdaptationSet>
</Period>
</MPD>
```

Let's put everything together

- Now we have a ed.mpd file which contains two adaptation set: one for video and one for audio.
- The video adaption set contains five representations.
- Go to <http://192.168.56.1:8080/ed/samples/dash-if-reference-player/index.html>
- Type <http://192.168.56.1:8080/ed/ed.mpd> in the textbox and click Load
- Turn off ABR, manually adjust the video representation
- See the video quality go up and down.

See how it works.

- User Developer Tools to emulate slower networks and see how HAS adapts.

The screenshot shows the Chrome Developer Tools Network tab. A context menu is open over the 'Network' tab, displaying various network emulation presets. The 'DSL (2 Mb/s 5ms RTT)' preset is selected. Below the menu, the network timeline shows several requests. The table below lists the details of these requests.

	Method	Status	Type	Initiator	Size	Time	Time
0k_39.m4s	GET	200	xhr	FragmentLoader.js:174	182 KB	719 ms	
	GET	200	xhr	FragmentLoader.js:174	(from cache)	3 ms	
0k_40.m4s	GET	200	xhr	FragmentLoader.js:174	178 KB	702 ms	