

Transmissões de vídeo na Web com taxas de bits adaptáveis (adaptive bitrate)

Por Newton Calegari

Eu acredito que você deve ter notado a diferença em relação ao tipo de conteúdo consumido na Web atualmente. Reflita um pouco e tente notar o que mais se vê atualmente: se o que veio em sua mente foi conteúdo em vídeo, você está certo e isso é mais uma evidência do que é apontado no relatório Cisco Visual Networking Index: Forecast and Methodology, 2015–2020, o qual mostra que o consumo de vídeo online representará cerca 80% do tráfego na Internet, em 2019. É muito vídeo! Esse mesmo relatório da Cisco mostra que levaria mais de 5 milhões de anos para assistir a todo conteúdo em vídeo que passará pelas redes a cada mês em 2020.

E não para por aí, o GlobalWebIndex: Video is the Future of Social diz que o compartilhamento de vídeos já é tão popular quanto o compartilhamento de fotos no Snapchat, e que mais da metade dos usuários do Facebook consome vídeo na plataforma.

Conseguimos perceber também o quanto o modo de produção de conteúdo em vídeo evoluiu nos últimos tempos, sendo possível que o usuário final crie conteúdo interessante com a própria câmera do celular, mas vale a pena ver como as tecnologias para suportar essa demanda de vídeo na Web também evoluíram, como é o caso do streaming.

Para entendermos como isso evoluiu, cabe olharmos um pouco para nossa história recente e nos lembramos do domínio quase que absoluto do Flash como tecnologia principal para streaming na Web. Não há muito tempo, o elemento <video> da especificação HTML5 conseguiu dar um grande passo em direção à mudança do uso das tecnologias de reprodução de vídeo no navegador.

Apesar de o uso do HTML5 para reprodução de vídeo ter possibilitado avanços – como o consumo de vídeo em dispositivos móveis (já que a Apple aboliu o Flash do iOS desde seu início) -, a transmissão via *progressive download* não é uma alternativa viável para sistemas e aplicações de grande escala. A transmissão com taxas de bits adaptáveis (adaptação livre do inglês *adaptive bitrate streaming*) é mais adequada para esses casos.

Vamos entender a diferença entre os dois tipos de transmissão:

Transmissão de mídia com download progressivo



O download progressivo é ideal para arquivos de mídia pequenos e leves, pois há tempo para carregar o buffer e iniciar a reprodução e, por ser pequeno, não afetará a experiência do usuário. Como pode ser visto na imagem 1, o vídeo será reproduzido como se fosse feita a leitura de uma sequência linear de quadros.

Transmissão com taxa de bits adaptável (*adaptive bitrate*)

Já no caso da transmissão com taxa de bits variável, a sequência de reprodução da mídia pode não ser linear, sendo possível ajustar a qualidade do conteúdo exibido sem a necessidade de carregar um arquivo completo (como no caso do *progressive download*).



A imagem 2 mostra diferentes segmentos com diferentes resoluções e sugere que a qualidade sendo reproduzida no player pode ser variável. Para superar as limitações da transmissão com download progressivo, algumas empresas criaram soluções proprietárias para transmissão com *adaptive bitrate*. A Apple possui a tecnologia HTTP Live Stream (HLS), a Adobe tem o HDS e a Microsoft possui o Smooth Streaming.

Com o propósito de diminuir essa fragmentação, surgiu o padrão DASH (no qual essas mesmas empresas também estão envolvidas), que significa Dynamic Adaptive Streaming over HTTP.

Além do padrão DASH, definido pelo consórcio MPEG, surgiu a especificação Media Source Extensions, que se tornou recomendação W3C em novembro de 2016. Essa especificação permite trabalhar com *adaptive bitrate streaming* no navegador sem a necessidade de se utilizar plugins de terceiros.

A especificação MSE integra o padrão MPEG-DASH ao HTML5, permitindo que exploremos mais aspectos da transmissão e consumo de vídeos na Web, como a redução do tempo de início do vídeo, o que melhora significativamente a experiência do usuário, permite “*slicing*” para inserção de anúncios, adaptação do conteúdo de acordo com a banda de conexão do usuário cliente e, por funcionar sobre o protocolo HTTP, não é necessário um servidor específico, podendo utilizar um webserver, como Apache ou nginx, e CDNs (*Content Delivery Network*), aproveitando as vantagens do webserver e configurando com facilidade recursos como HTTPS, proxy e cache.

Utilizando DASH e MSE na prática

Para realizar uma transmissão com *adaptive bitrate* e explorar os padrões DASH e MSE, vamos primeiro gerar diferentes variações de resolução para um vídeo; depois iremos criar o arquivo de manifesto MPD (*Media Presentation Description*), um arquivo XML contendo informações sobre os diferentes segmentos de vídeo, seus relacionamentos e informações necessárias para escolher quais deles serão reproduzidos em cada momento.

Para essas tarefas, utilizaremos o FFmpeg (www.ffmpeg.org), uma ferramenta excelente para lidar com conversões de conteúdo multimídia, com as bibliotecas *libvpx* e *libvorbis*, além de suporte para WebM. Neste exemplo, estou utilizando um vídeo no container [webm](#).

O primeiro passo é utilizar o FFmpeg para gerar a trilha de áudio desse conteúdo multimídia. Como estamos extraindo o áudio no codec Vorbis, é importante que o FFmpeg esteja com essa biblioteca ativa.

```
1 | ffmpeg -i arquivo.video.principal.webm -vn -acodec libvorbis -ab 128k trilha_audio.webm
```

Agora vamos gerar as diferentes variações de vídeo:

```
1 | ffmpeg -i arquivo.video.principal.webm -c:v libvpx-vp9 -keyint_min 150 \  
2 | -g 150 -tile-columns 4 -frame-parallel 1 -f webm -dash 1 \  
3 | -an -vf scale=160:190 -b:v 250k video_160x90_250k.webm \  
4 | -an -vf scale=320:180 -b:v 500k video_320x180_500k.webm \  
5 | -an -vf scale=640:360 -b:v 750k video_640x360_750k.webm \  
6 | -an -vf scale=640:360 -b:v 1000k video_640x360_1000k.webm \  
7 | -an -vf scale=1280:720 -b:v 1500k video_1280x720_1500k.webm
```

Esse comando no FFmpeg gerará variações do vídeo em diferentes resoluções a partir do arquivo de vídeo principal.

O passo seguinte é a criação do arquivo de manifesto, um arquivo XML que indicaremos com a extensão MPD.

Também utilizaremos o FFmpeg para essa tarefa:

```
1 | ffmpeg \  
2 | -f webm_dash_manifest -i video_160x90_250k.webm \  
3 | -f webm_dash_manifest -i video_320x180_500k.webm \  
4 | -f webm_dash_manifest -i video_640x360_750k.webm \  
5 | -f webm_dash_manifest -i video_640x360_1000k.webm \  
6 | -f webm_dash_manifest -i video_1280x720_1500k.webm \  
7 | -f webm_dash_manifest -i trilha_audio.webm \  
8 | -c copy -map 0 -map 1 -map 2 -map 3 -map 4 -map 5 \  
9 | -f webm_dash_manifest \  
10 | -adaptation_sets "id=0,streams=0,1,2,3,4 id=1,streams=5" \  
11 | manifest.mpd
```

Reprodução do conteúdo no navegador web

Vamos utilizar uma biblioteca chamada [dash.js](#) para a reprodução do vídeo.

O código abaixo mostra como instanciar o player.

```
1 | <video data-dashjs-player src="http://exemplo.org/manifest.mpd" controls>  
2 | </video>  
3 |  
4 | <script src="http://cdn.dashjs.org/latest/dash.all.min.js">  
5 | </Script>
```

Note que o atributo “src” do elemento de vídeo carrega um arquivo de manifesto MPD, e não diretamente a mídia para a reprodução.

Esse foi um exemplo básico de como utilizar técnicas de *adaptive media* no navegador web sem a utilização de plugins de terceiros. Um ponto interessante que vale ressaltar é que a utilização dessa técnica, no passado não muito distante, ficava restrita a poucas aplicações e empresas devido à complexidade e aos custos de desenvolvimento.

Os vídeos na Web estão aumentando

A Web como plataforma de vídeo ainda está evoluindo e crescerá muito mais, por isso é muito importante que fiquemos atentos tanto aos hábitos de consumo de conteúdo quanto às tecnologias que contribuirão para esse crescimento. Como direções futuras, vale observar o consumo de vídeo em 360°, que vem aumentando de modo acelerado, talvez devido às câmeras específicas para esse

tipo de vídeo estarem mais acessíveis ou também pela característica de imersão que vem sendo trazida pelas aplicações de realidade virtual.