

RFC 8216 : HTTP Live Streaming

Stéphane Bortzmeyer

<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 1 septembre 2017

Date de publication du RFC : Août 2017

<https://www.bortzmeyer.org/8216.html>

Ce RFC décrit la version 7 du protocole "*HTTP Live Streaming*", qui permet de distribuer des flux multimédia, typiquement de la vidéo.

Utiliser HTTP, c'est bien, ça passe quasiment tous les pare-feux, et ça permet de réutiliser l'infrastructure existante, comme les serveurs, et les caches. (Au contraire, un protocole comme RTP, RFC 3550¹, a souvent du mal à passer les différentes "*middleboxes*".) La manière triviale de distribuer une vidéo avec HTTP est d'envoyer un fichier vidéo au bon format, avec des méthodes HTTP standard, typiquement en réponse à un GET. Mais cette méthode a des tas de limites. Par exemple, elle ne permet pas de s'adapter à une capacité <<https://www.bortzmeyer.org/capacite.html>> variable du réseau. Par contre, le protocole décrit dans ce RFC, quoique plus compliqué qu'un simple GET, permet cet ajustement (et plein d'autres).

"*HTTP Live Streaming*" (ou simplement "*Live Streaming*", ou encore HLS) existe depuis longtemps, la première spécification a été publiée (par Apple) en 2009, mais c'est la première fois qu'il se retrouve dans un RFC, ce qui devrait permettre d'améliorer encore l'interopérabilité.

Donc, brève description de "*HTTP Live Streaming*" : une ressource multimédia qu'on veut diffuser est représentée par un URI. Ceui-ci désigne une "*playlist*", qui est un fichier texte contenant les URI des données multimedia et diverses informations. Les données multimédia figurent dans plusieurs segments ; en jouant tous les segments successivement, on obtient la vidéo souhaitée. Le client n'a pas ainsi à tout charger, il peut récupérer les segments au fur et à mesure de ses besoins. (Au lieu de « segments », `mmu_man` <https://m.g3l.org/@mmu_man> suggère de parler d'un « émincé <<https://fr.wiktionary.org/wiki/%C3%A9minc%C3%A9>> de fichiers ».) Voici l'exemple de "*playlist*" que donne le RFC :

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc3550.txt>

```
#EXTM3U
#EXT-X-TARGETDURATION:10

#EXTINF:9.009,
http://media.example.com/first.ts
#EXTINF:9.009,
http://media.example.com/second.ts
#EXTINF:3.003,
http://media.example.com/third.ts
```

Ce format de *"playlist"* est dérivé du format M3U et est généralement nommé « M3U étendu ». On y trouve successivement l'identificateur du format (si ça commence par #EXTM3U, c'est bien une *"playlist"* de notre RFC), la durée maximale d'un segment en secondes, puis les trois segments, avec la durée de chacun. Cette durée, indiquée par l'étiquette #EXTINF, est un des rares éléments obligatoires dans une *"playlist"*. Il y a bien sûr plein d'autres choses qu'on peut mettre dans un fichier *"playlist"*, la section 4 du RFC en donne la liste complète. Ainsi, #EXT-X-BYTERANGE permettra d'indiquer qu'on ne veut jouer qu'une partie d'une vidéo. Des exemples plus complets sont donnés dans la section 8 du RFC.

Ça, c'était la *"media playlist"*, qui se limite à lister des ressources multimédias. La *"master playlist"*, dont le format est le même, est plus riche, et permet de spécifier des versions alternatives d'un même contenu. Ces variantes peuvent être purement techniques (« *"variant stream"* », deux encodages d'une même vidéo avec différents formats) ou porter sur le fond : par exemple deux pistes audio dans des langues différentes, ou deux vidéos d'un même événement filmé depuis deux points de vue distincts. Les variantes techniques servent au client à s'ajuster aux conditions réelles du réseau, les variantes de fond (nommées *"renditions"*) servent au client à donner à l'utilisateur ce qu'il préfère. Ici, par exemple, on a dans cette *"master playlist"* trois variantes techniques (trois résolutions, de la plus basse, à la plus élevée, qui va nécessiter une bonne capacité réseau), et trois « *"renditions"* », correspondant à trois points de vue (*"centerfield"* et *"dugout"* sont des termes de baseball, ce qui est logique vu l'employeur d'un des deux auteurs du RFC) :

```
#EXTM3U
#EXT-X-MEDIA:TYPE=VIDEO,GROUP-ID="low",NAME="Main", \
  DEFAULT=YES,URI="low/main/audio-video.m3u8"
#EXT-X-MEDIA:TYPE=VIDEO,GROUP-ID="low",NAME="Centerfield", \
  DEFAULT=NO,URI="low/centerfield/audio-video.m3u8"
#EXT-X-MEDIA:TYPE=VIDEO,GROUP-ID="low",NAME="Dugout", \
  DEFAULT=NO,URI="low/dugout/audio-video.m3u8"

#EXT-X-STREAM-INF:BANDWIDTH=1280000,CODECS="...",VIDEO="low"
low/main/audio-video.m3u8

#EXT-X-MEDIA:TYPE=VIDEO,GROUP-ID="mid",NAME="Main", \
  DEFAULT=YES,URI="mid/main/audio-video.m3u8"
#EXT-X-MEDIA:TYPE=VIDEO,GROUP-ID="mid",NAME="Centerfield", \
  DEFAULT=NO,URI="mid/centerfield/audio-video.m3u8"
#EXT-X-MEDIA:TYPE=VIDEO,GROUP-ID="mid",NAME="Dugout", \
  DEFAULT=NO,URI="mid/dugout/audio-video.m3u8"

#EXT-X-STREAM-INF:BANDWIDTH=2560000,CODECS="...",VIDEO="mid"
mid/main/audio-video.m3u8

#EXT-X-MEDIA:TYPE=VIDEO,GROUP-ID="hi",NAME="Main", \
  DEFAULT=YES,URI="hi/main/audio-video.m3u8"
#EXT-X-MEDIA:TYPE=VIDEO,GROUP-ID="hi",NAME="Centerfield", \
  DEFAULT=NO,URI="hi/centerfield/audio-video.m3u8"
#EXT-X-MEDIA:TYPE=VIDEO,GROUP-ID="hi",NAME="Dugout", \
  DEFAULT=NO,URI="hi/dugout/audio-video.m3u8"

#EXT-X-STREAM-INF:BANDWIDTH=7680000,CODECS="...",VIDEO="hi"
hi/main/audio-video.m3u8
```

Si vous voulez le voir en vrai, et que vous êtes abonné à Free, la Freebox distribue sa "playlist" sous ce format (mais avec le mauvais type MIME) :

```
% curl -s http://mafreebox.freebox.fr/freeboxtv/playlist.m3u
#EXTM3U
#EXTINF:0,2 - France 2 (bas débit)
rtsp://mafreebox.freebox.fr/fbxtv_pub/stream?namespace=1&service=201&flavour=ld
#EXTINF:0,2 - France 2 (HD)
rtsp://mafreebox.freebox.fr/fbxtv_pub/stream?namespace=1&service=201&flavour=hd
#EXTINF:0,2 - France 2 (standard)
rtsp://mafreebox.freebox.fr/fbxtv_pub/stream?namespace=1&service=201&flavour=sd
#EXTINF:0,2 - France 2 (auto)
rtsp://mafreebox.freebox.fr/fbxtv_pub/stream?namespace=1&service=201
#EXTINF:0,3 - France 3 (standard)
rtsp://mafreebox.freebox.fr/fbxtv_pub/stream?namespace=1&service=202&flavour=sd
...
```

(On me dit qu'avec certaines Freebox, il faut ajouter `-L` à `curl`, en raison d'une redirection HTTP.) Un client comme `vlc` peut prendre comme argument l'URI de la "playlist", la télécharger et jouer ensuite les vidéos.

(Tous les moyens sont bons pour télécharger la "playlist", le RFC n'impose pas un mécanisme particulier.) Télécharger chacun des segments se fait a priori avec HTTP, RFC 7230, mais vous voyez plus haut que, sur la Freebox, c'est RTSP, RFC 7826. Le serveur, lui, est un serveur HTTP ordinaire. L'auteur a « juste » eu à découper sa vidéo en segments (ou la garder en un seul morceau si elle est assez courte), à fabriquer un fichier "playlist", puis à le servir avec le type MIME `application/vnd.apple.mpegurl`. (La section 6 du RFC décrit plus en détail ce que doivent faire client et serveur.)

Il est recommandé que chaque segment soit auto-suffisant, c'est-à-dire puisse être décodé sans avoir besoin des autres segments. Ainsi, si les segments sont en H.264, chacun doit avoir un "Instantaneous Decoding Refresh (IDR)".

Plusieurs formats sont évidemment possibles dans une "playlist" "HTTP Live Streaming", MPEG-2, MPEG-4 fragmenté, WebVTT pour avoir les sous-titres, et un format pour l'audio (qui accepte entre autres MP3).

"HTTP Live Streaming" permet aussi de servir du direct. Prenons par exemple le flux vidéo de l'Assemblée Nationale, qui permet de suivre les débats parlementaires :

```
% curl -s http://videos-diffusion-live.assemblee-nationale.fr/live/live1/stream1.m3u8
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-ALLOW-CACHE:NO
#EXT-X-TARGETDURATION:11
#EXT-X-MEDIA-SEQUENCE:382
#EXTINF:9.996,
media_1_4805021_382.ts
#EXTINF:11.483,
media_1_4805021_383.ts
...
```

Un peu plus tard, les fichiers ont changé, le client HLS doit recharger la "playlist" régulièrement (section 6.2.2 du RFC), et charger de nouveaux fichiers vidéo :

```
% curl -s http://videos-diffusion-live.assemblee-nationale.fr/live/live1/stream1.m3u8
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-ALLOW-CACHE:NO
#EXT-X-TARGETDURATION:11
#EXT-X-MEDIA-SEQUENCE:390
#EXTINF:10.039,
media_1_4805021_390.ts
#EXTINF:11.242,
media_1_4805021_391.ts
...
```

(Notez également l'incrémementation du #EXT-X-MEDIA-SEQUENCE.) Ainsi, l'utilisateur a l'impression d'une vidéo continue, alors que le logiciel client passe son temps à charger des segments différents et à les afficher à la suite.

Notez que "HTTP Live Streaming" n'est pas le seul mécanisme dans ce domaine, son principal concurrent est MPEG-DASH.

Si vous voulez lire davantage :

- Un article d'introduction <<https://larryjordan.com/articles/basics-of-http-live-streaming/>>, orienté vers l'auteur de vidéos,
- Un article plus technique d'Apple <<https://developer.apple.com/library/content/documentation/NetworkingInternet/Conceptual/StreamingMediaGuide/Introduction/Introduction.html>>.