

How To HLS

HLS et l'Adaptive Streaming

HLS (HTTP Live Streaming) est l'un des formats de streaming vidéo les plus répandus sur les navigateurs de bureau et les navigateurs mobiles.

Les utilisateurs finaux ont des tailles d'écran différentes et des performances réseau différentes : pour pallier à ces problèmes, HLS permet de **découper la vidéo en fragments multiples** de résolution et débit binaire différents. Ces fragments peuvent alors être commutés de façon transparente.

Ce concept est appelé **MBR** (Multi Bit Rate).

Ainsi, un utilisateur ayant des problèmes de réseau ne verra pas sa vidéo interrompue s'il n'est plus en mesure de la télécharger en Haute Définition. Il recevra toujours la vidéo, mais avec une qualité inférieure, ce qui permet d'éviter les interruptions.

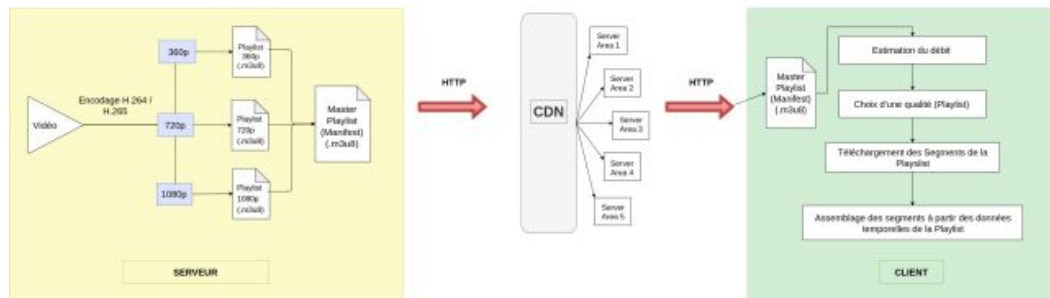
C'est ce qu'on appelle l'**Adaptive Streaming**.

Rôle du serveur HLS : encodage avec ffmpeg

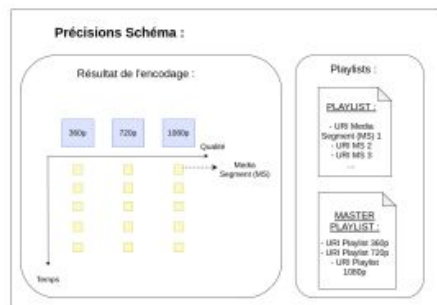
L'étape de création des fragments de vidéo en résolutions diverses s'appelle l'**encodage**. Les fragments créés sont appelés **segments**. Ils sont d'extension .ts.

Afin de réaliser l'encodage, nous utiliserons **ffmpeg**, un outil puissant qui prend en charge la conversion de divers formats vidéo de l'un à l'autre, y compris les formats de fichiers adaptés à HLS, à la fois en entrée et en sortie.

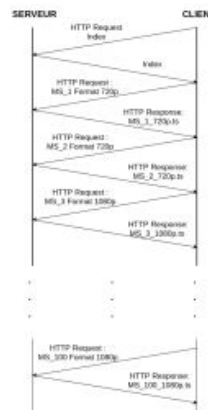
Schéma Fonctionnement Général de HLS :



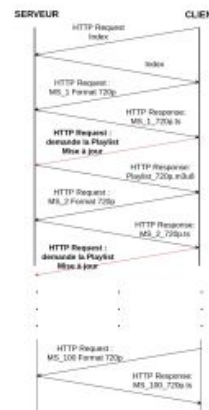
Diagrammes Séquence :



Cas VOD :



Cas Live :

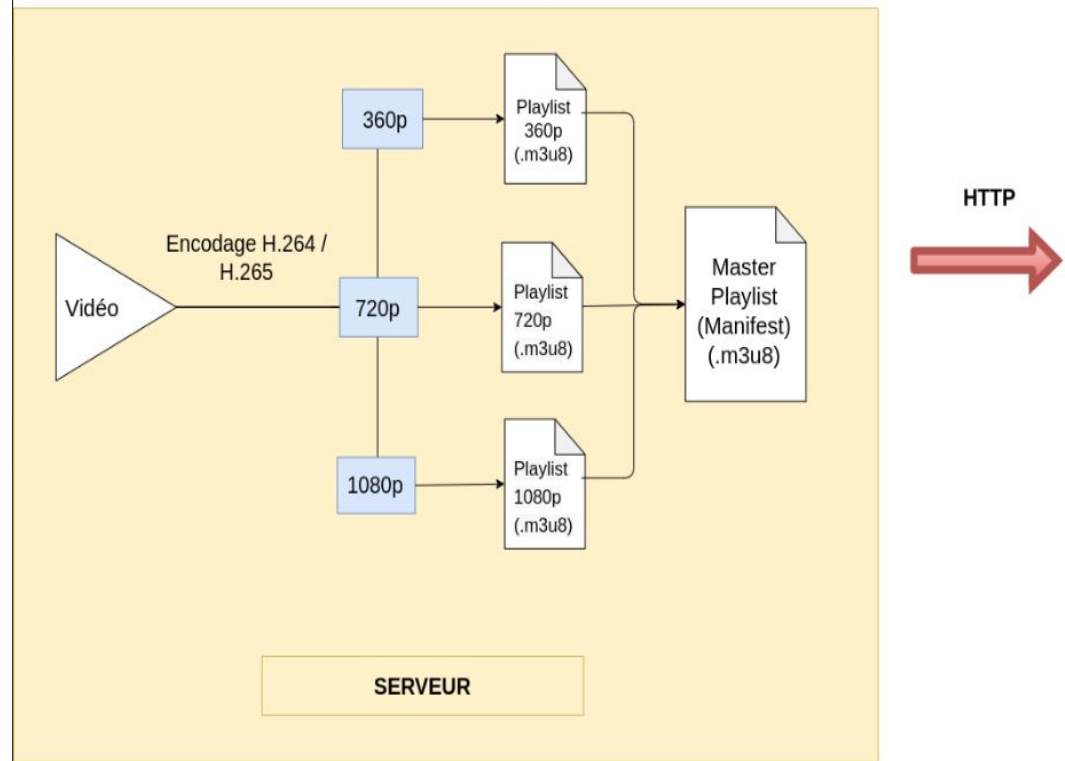


Segmentation de la vidéo et encodage en différentes résolutions

Du côté du serveur HLS, lors de l'encodage, plusieurs segments sont créés pour différentes largeurs de bande et différentes résolutions.

Les URL des segments sont recensés selon la taille de l'écran et la bande passante disponible, dans un fichier d'index au format .m3u8 nommé **Playlist**.

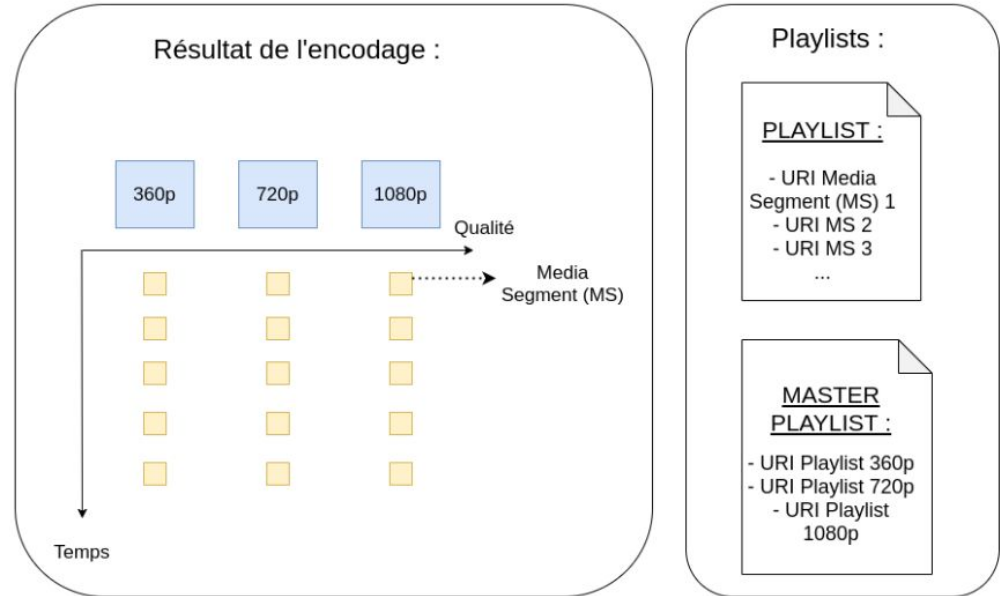
Un fichier nommé **Master Playlist** regroupe les URL des différentes Playlists.



Après l'encodage, on obtient les **Media Segments (.ts)** en différentes résolutions, ainsi que les playlists de chacune.

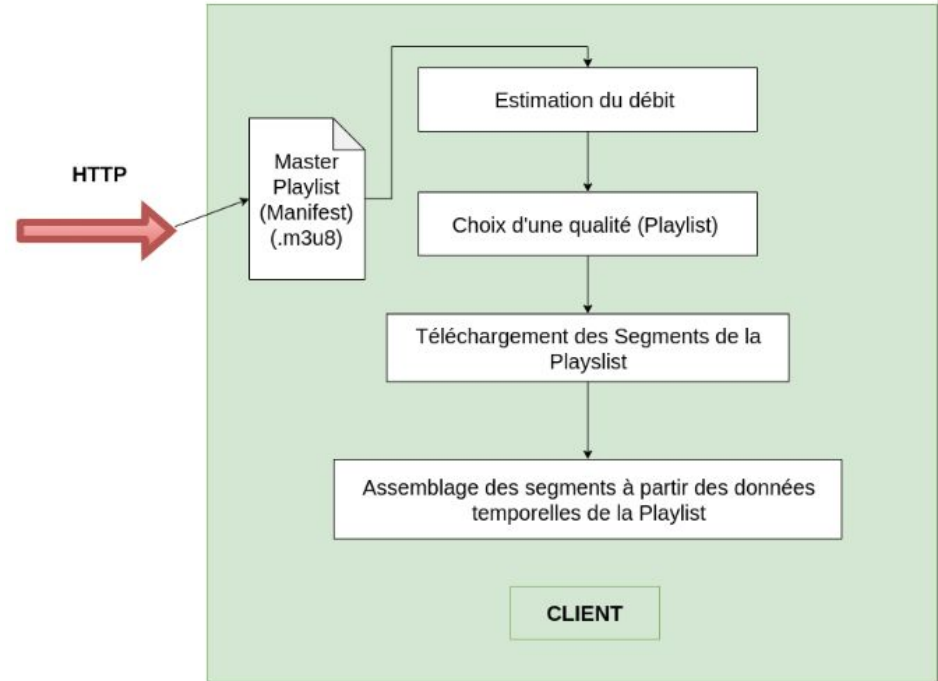
La Master Playlist précisera les résolutions disponibles en fonction de la bande passante de l'utilisateur client.

Précisions Schéma :

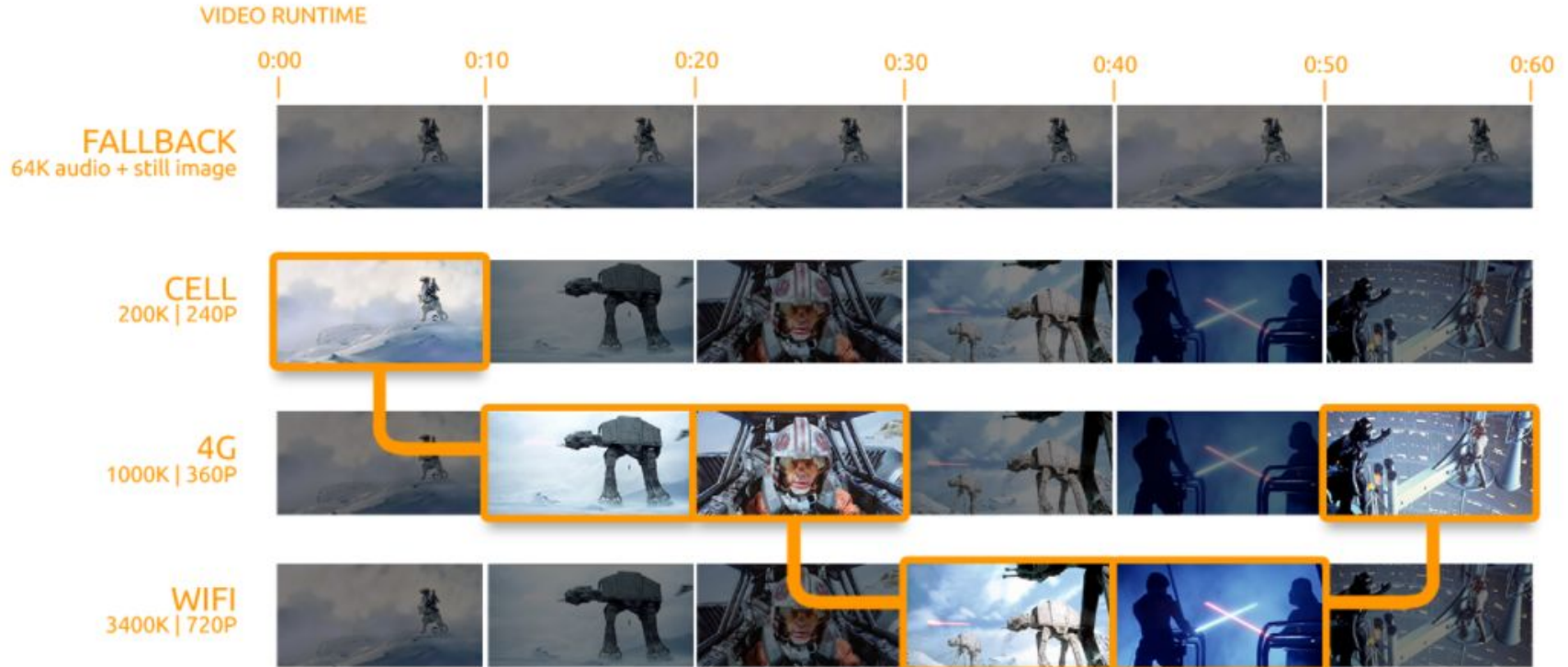


Le client, grâce à une requête HTTP, **récupère la Master Playlist** qui lui permet de choisir les différentes résolutions en fonction de son débit.

Après avoir estimé son débit, il peut désormais **choisir la qualité adaptée**, et télécharger les segments avec cette qualité au fur et à mesure et les assembler.

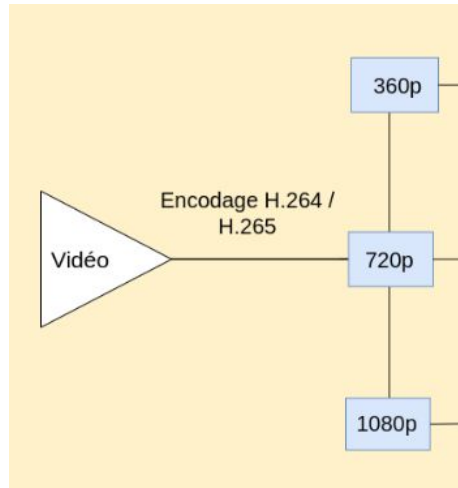


Exemple qui montre l'aspect adaptative streaming du protocole HLS



Mise en place d'un serveur et d'un client HLS

Première partie: Encodage de la vidéo côté serveur grâce à ffmpeg



I] La création de la master playlist:

- ❏ Encodage de la vidéo sur trois qualités :
 - 360p
 - 480p
 - 720p

*Cette playlist permet au client de choisir les résolutions dont il a besoin en fonction de sa bande passante. Les segments sont dans les dossiers **test360**, **test480** et **test720***

```
1  #EXTM3U
2  #EXT-X-VERSION:3
3  #EXT-X-STREAM-INF:BANDWIDTH=800000,RESOLUTION=640x360
4  ./test360/360p.m3u8
5  #EXT-X-STREAM-INF:BANDWIDTH=1400000,RESOLUTION=842x480
6  ./test480/480p.m3u8
7  #EXT-X-STREAM-INF:BANDWIDTH=2800000,RESOLUTION=1280x720
8  ./test720/720p.m3u8
```

II] Encodage de la vidéo pour les différentes résolutions

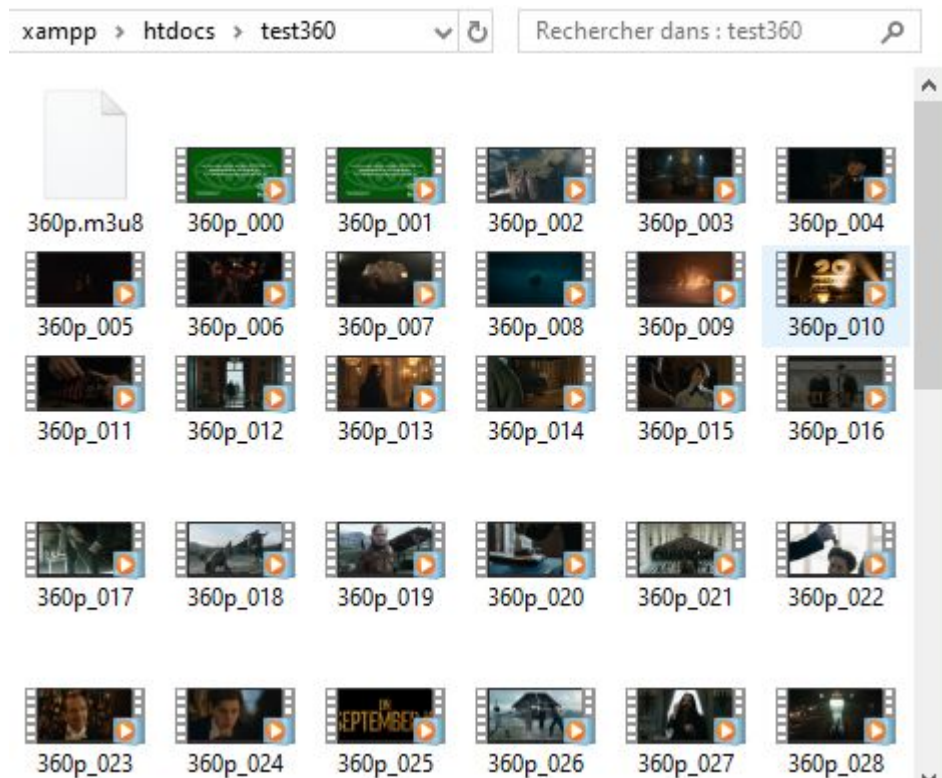
il est très important qu'en plus des paramètres de résolution et de débit binaire, les commandes soient identiques afin que les rendus soient correctement alignés, ce qui signifie que les images clés seront placées exactement aux mêmes positions pour permettre de passer d'une image à l'autre en fluidité.

La fonction ci-dessous permet de faire l'encodage de la vidéo :

pour chaque résolution, on place les différentes trames .ts ainsi que la playlist correspondante (voir page suivante) à cette résolution dans le dossier.

```
function convert(req,res) {
  fs.stat('test360/360p.m3u8', function(err) {
    if (!err) {
      console.log('file already exists');
    }
    else if (err.code === 'ENOENT') {
      console.log('file or directory does not exist');
      child = exec(`ffmpeg -hide_banner -y -i data/out.mp4 \
        -vf "pad=ceil(iw/2)*2:ceil(ih/2)*2" -c:a aac -ar 48000 -c:v h264 -profile:v main -crf 20 -sc_threshold 0 -g 48 -keyint_min 48 -hls_time \
        4 -hls_playlist_type vod -b:v 800k -maxrate 856k -bufsize 1200k -b:a 96k -hls_segment_filename test360/360p_%03d.ts test360/360p.m3u8 \
        -vf "pad=ceil(iw/2)*2:ceil(ih/2)*2" -c:a aac -ar 48000 -c:v h264 -profile:v main -crf 20 -sc_threshold 0 -g 48 -keyint_min 48 -hls_time \
        4 -hls_playlist_type vod -b:v 1400k -maxrate 1498k -bufsize 2100k -b:a 128k -hls_segment_filename test480/480p_%03d.ts test480/480p.m3u8 \
        -vf "pad=ceil(iw/2)*2:ceil(ih/2)*2" -c:a aac -ar 48000 -c:v h264 -profile:v main -crf 20 -sc_threshold 0 -g 48 -keyint_min 48 -hls_time \
        4 -hls_playlist_type vod -b:v 2800k -maxrate 2996k -bufsize 4200k -b:a 128k -hls_segment_filename test720/720p_%03d.ts test720/720p.m3u8`,
        function (error, stdout, stderr) {
          if (error !== null) {
            console.log('exec error: ' + error);
          }
        });
    }
  });
}
```

Résultat: contenu du dossier 360p



A la fin de l'encodage, on obtient les trames de la vidéo dans le répertoire correspondant à leur résolution comme montré sur la figure ci-contre

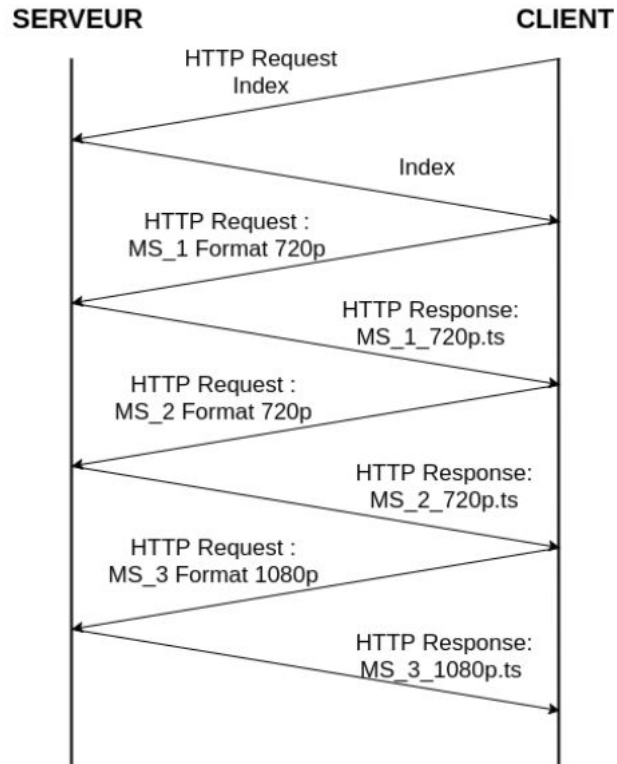
Playlist correspondante à la résolution 360: fichier 360p.m3u8

```
1 #EXTM3U
2 #EXT-X-VERSION:3
3 #EXT-X-TARGETDURATION:4
4 #EXT-X-MEDIA-SEQUENCE:0
5 #EXT-X-PLAYLIST-TYPE:VOD
6 #EXTINF:4.000000,
7 360p_000.ts
8 #EXTINF:4.000000,
9 360p_001.ts
10 #EXTINF:4.000000,
11 360p_002.ts
12 #EXTINF:4.000000,
13 360p_003.ts
14 #EXTINF:4.000000,
15 360p_004.ts
16 #EXTINF:4.000000,
17 360p_005.ts
18 #EXTINF:4.000000,
19 360p_006.ts
20 #EXTINF:4.000000,
21 360p_007.ts
22 #EXTINF:4.000000,
23 360p_008.ts
24 #EXTINF:4.000000,
25 360p_009.ts
26 #EXTINF:4.000000,
27 360p_010.ts
28 #EXTINF:4.000000,
29 360p_011.ts
30 #EXTINF:4.000000,
31 360p_012.ts
32 #EXTINF:4.000000,
33 360p_013.ts
34 #EXTINF:4.000000,
35 360p_014.ts
```

La playlist ci-contre représente les différents segments (les fichiers .ts) contenus dans ce dossier qui ont pour résolution 360p.

C'est une diffusion de type vidéo à la demande (VOD) : le fichier est de taille fixe, tous les segments sont déjà créés.

Mise en place d'un serveur



L'opération d'encodage est réalisée par le **serveur**.

La Master Playlist, les playlists et les segments recensés par les playlists sont hébergés par le serveur.

Ainsi, lorsque le client réalise une requête afin d'obtenir un fichier, le serveur lui répond en envoyant le fichier demandé.

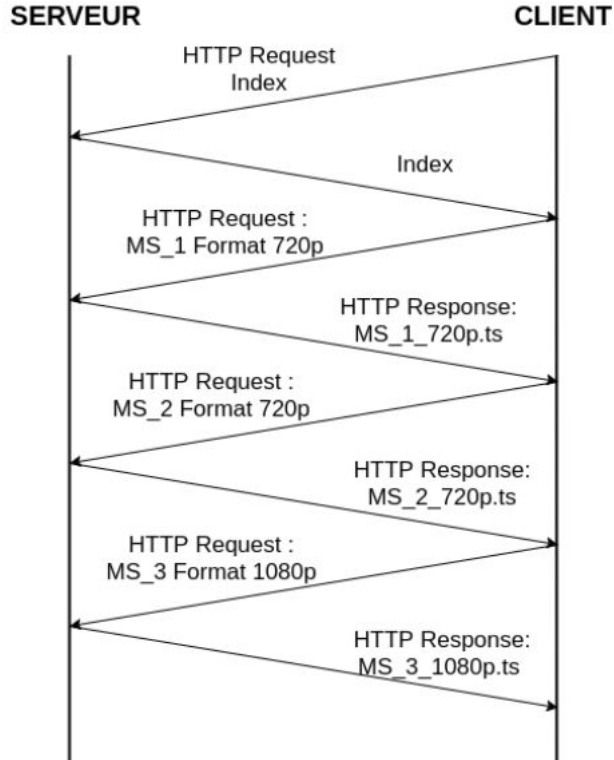
Code permettant la mise en place du serveur

```
var http = require('http');  
var fs=require('fs')  
var exec = require('child_process').exec;
```

```
27 //2. Express.js  
28 var express = require('express');  
29 var app = express();  
30 app.use(express.static('.'));  
31  
32  
33 //Create a route for the first http connection  
34 app.get('/', function (request, response) {  
35     response.sendFile(__dirname + '/indexconnect.html');  
36     console.log(request.url);  
37     convert(request,response);  
38 }  
39 });  
40  
41 //3. Create a server that will serve both http and socket connection using the app function of Express.js  
42 var server = http.createServer(app);  
43  
44  
45  
46 //6. Listen to the "shared" server (not the Express.js app)  
47 server.listen(3000, console.log("Listening to port 3000"));  
48
```

Fonction permettant
l'encodage de la vidéo

Mise en place d'un client HLS



Dans le cas du streaming via le protocole HLS, **le client récupère la Master Playlist** grâce à une requête HTTP.

Ce fichier regroupe les liens vers des playlists de différentes qualité.

Chaque Playlist regroupe les segments permettant au client de rassembler la vidéo.

Par la suite, il **télécharge les segments** un à un, **en choisissant la qualité** de segment qui lui convient.

Mise en place d'un client HLS

← → ↺ ⓘ localhost:3000

2. Objectif du protocole:

3. Standardisation

II. Description du Protocole pour VOD

-----Schémas décrivant le protocole

III. Description du Protocole pour Live Videos

IV. Avantages et inconvénients du protocole

V. Demonstration



Elements Console Sources Network Performance

Filter ☐ Hide data URLs ☒ All XHR JS CSS Img

☐ Has blocked cookies ☐ Blocked Requests

2000 ms 4000 ms 6000 ms 8000 ms

Name	Status	Type
localhost	304	document
hls.js@latest	200	script
bootstrap.min.css	200	stylesheet
playlist.m3u8	304	xhr
360p.m3u8	304	xhr
360p_000.ts	304	xhr
blob:http://localhost:3000/0dff3cb5...	200	text/jav...
480p.m3u8	304	xhr
360p_001.ts	304	xhr
360p_002.ts	304	xhr

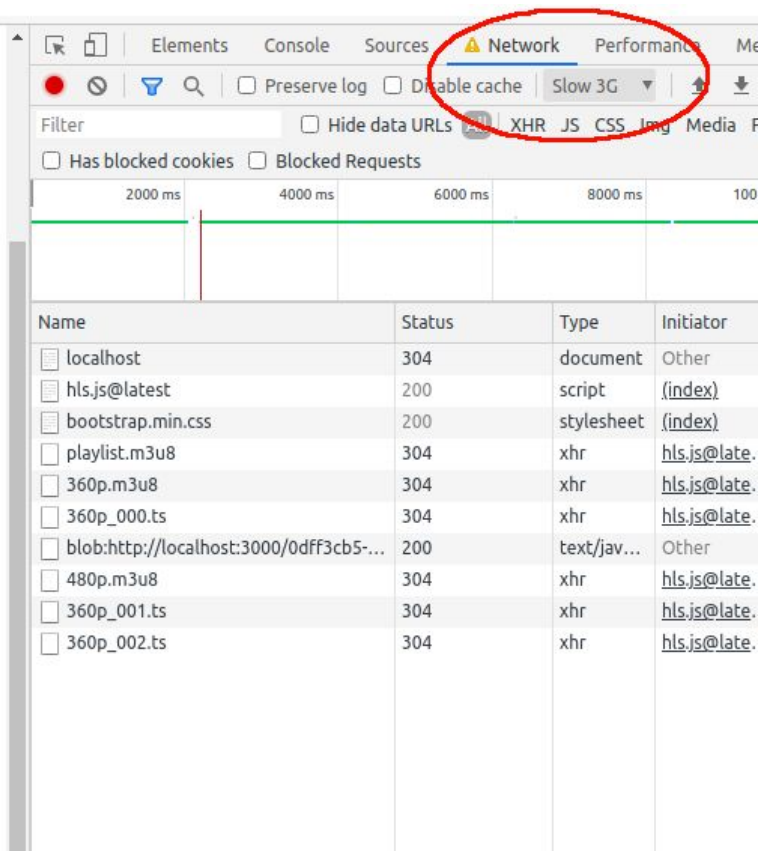
10 requests 1.9 kB transferred 1.6 MB resources Finish: 14.48 s

Console What's New

top Filter

⚠ DevTools failed to load SourceMap: Could not load content for HTTP error: status code 404, net::ERR_HTTP_RESPONSE_CODE_FAIL

Mise en place d'un client HLS



Le client HLS choisit la playlist en fonction du débit de son réseau et commence le téléchargement des segments.

Depuis le navigateur, en tapant CTRL+C+Shift, il a la possibilité de passer en mode “Inspect” et de **changer les caractéristiques réseau** (onglet Network).

Il peut décider ainsi de changer sa bande passante, ce qui va influencer sur la qualité des segments reçus.

Mise en place d'un client HLS

```
<!doctype html>
<html>
<head>
  <meta charset="UTF-8">
  <title>HLS basic example</title>
</head>

<body>
  <h1>Bunny</h1>
  <script src="https://cdn.jsdelivr.net/npm/hls.js@latest"></script>
  <video id="player" muted =true style="width: 900px; height: 507px;" controls></video>
  <script>
    var video = document.getElementById('player');
    var hls = new Hls();
    hls.loadSource('playlist.m3u8');
    hls.attachMedia(video);
    hls.on(Hls.Events.MANIFEST_PARSED,function() {
      video.play();
    });
  </script>
</body>

</html>
```

A vous de jouer !

Dans un terminal, tapez “git clone <https://github.com/Countertermatt/rsc-video.git>”

Puis, suivez les instructions du fichier **README.me** situé dans le **répertoire hls**.

Vous pouvez changer les performances réseau de votre navigateur, comme indiqué précédemment, pour observer vous-même l’effet sur le téléchargement !