**異質多網多媒體服務**

**期末報告**

**題目:**

**DASH影音串流分析**

**組員姓名:郭靜**

**學號:108598068**

**系級:資工碩一**

* **動機與目的**

設計這個平台，是對於收看電影之前，想先觀看預告片或不知道看什麼電影的群眾設計，可以透過網頁選擇想要觀看的電影預告片，在觀看預告片之前，有一段文字的簡介可提供觀眾閱讀。並且提供不同畫質選擇: 360p、480p、720p，點選按鈕即可切換。網頁末端有影片推薦，可直接點選，切換到想觀看的電影預告頁面。

* **預計完成功能**

1. 選擇影片欄
2. 電影簡介
3. 串流播放影片
4. 選擇畫素鈕
5. 推薦電影欄

* **開發環境**

作業系統 Windows10 語言: nodejs、html5

使用函式庫: ffmpeg、mp4box、dashjs、express、http、css、javascript

* **實踐功能**

自行產生mpd流程

1. 取得mp4檔
2. 透過ffmpeg將影片檔與影音檔分開

取得影片檔指令:

ffmpeg -i video.mp4 -an mute-video.mp4

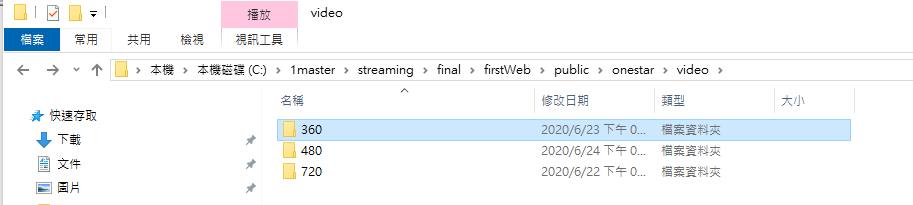
取得影音檔指令:

ffmpeg -i myvideo.mp4 -vn -acodec copy audio.mp4

1. 透過ffmpeg設定qp輸出不同畫質之 .mp4

ffmpeg -i .\intput.mp4 -vcodec libx264 -an -qp 1 video\_720p.mp4

1. 將不同畫質的影片檔分別丟進名為360、480、720的資料夾，以便將影片切割



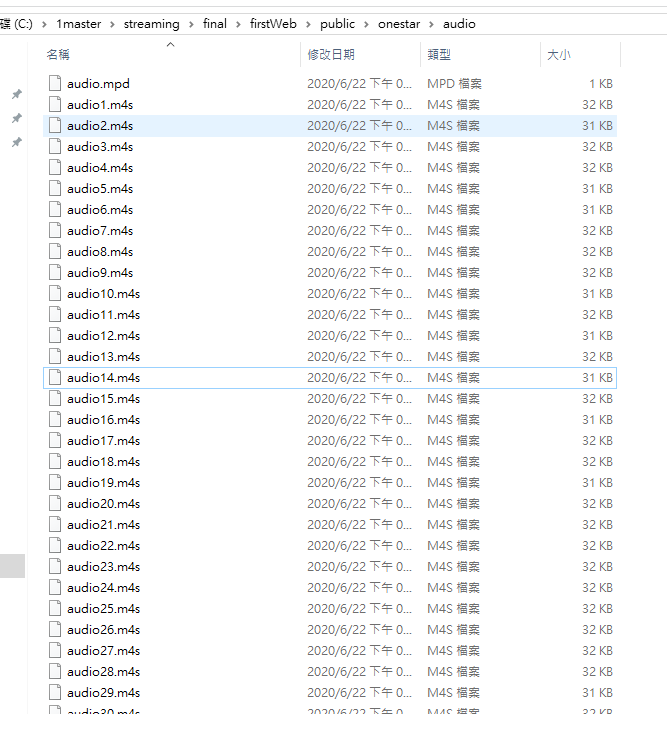
1. 先切割影音檔，透過mp4box

指令為

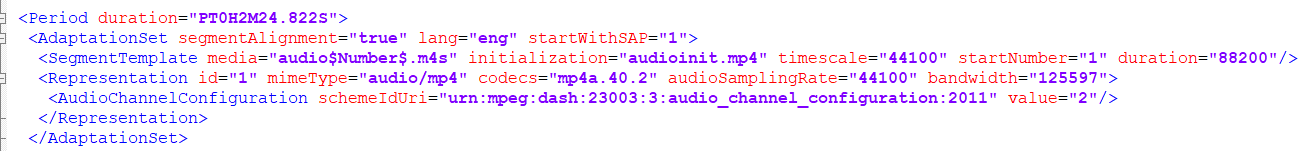
mp4box -dash 2000 -bs-switching no -segment-name audio$Number -out audio.mpd .\audio.mp4

切割之後會產生許多m4s與一個audio.mpd

audio.mpd 裡面有我需要的音軌



音軌如下圖

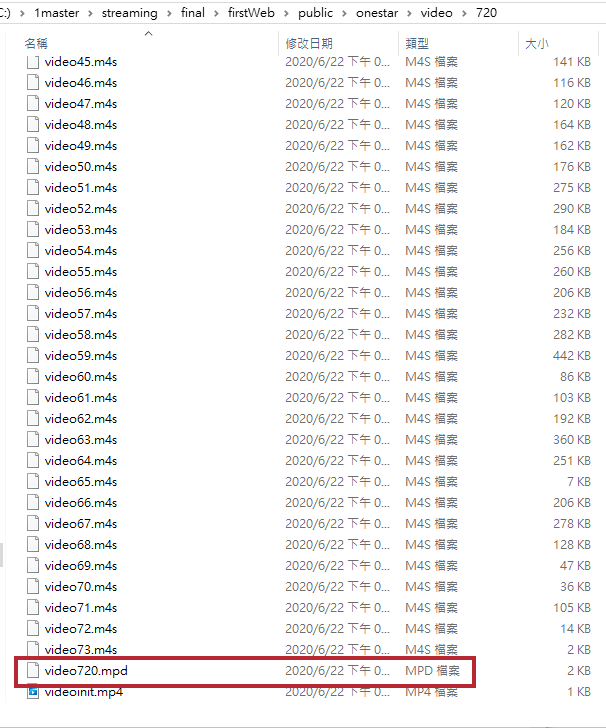


1. 切割影片檔，透過mp4box

因為有不同畫質，而做法一樣，以畫質720p為例

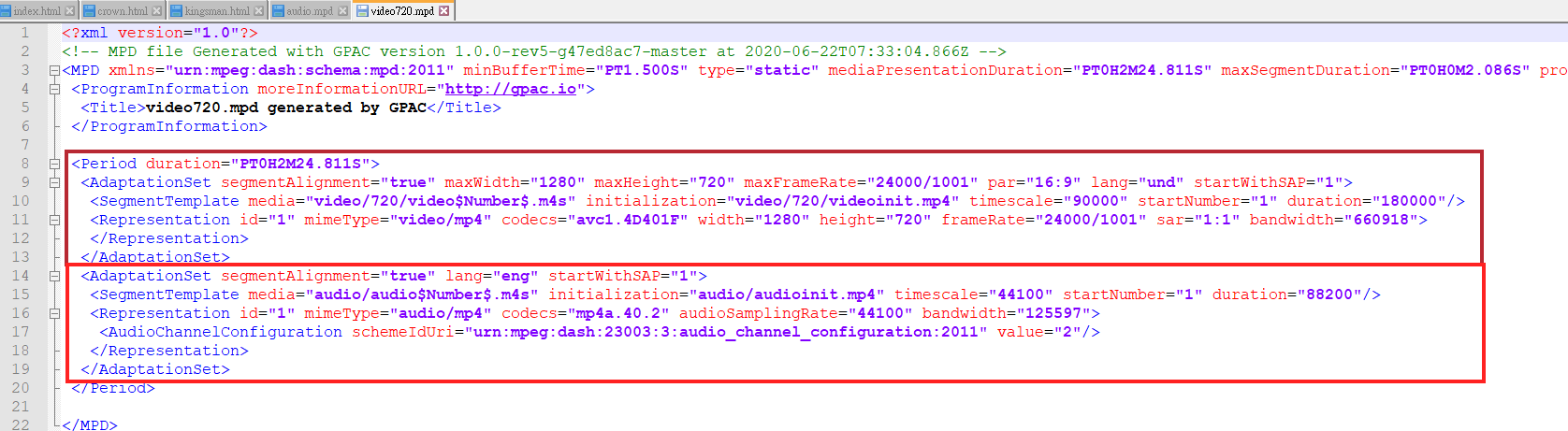
指令為

mp4box -dash 2000 -bs-switching no -segment-name video$Number -out video720.mpd .\video.mp4



產生許多segment與一個mpd檔，即為影片軌

1. 合併audio.mpd與video.mpd



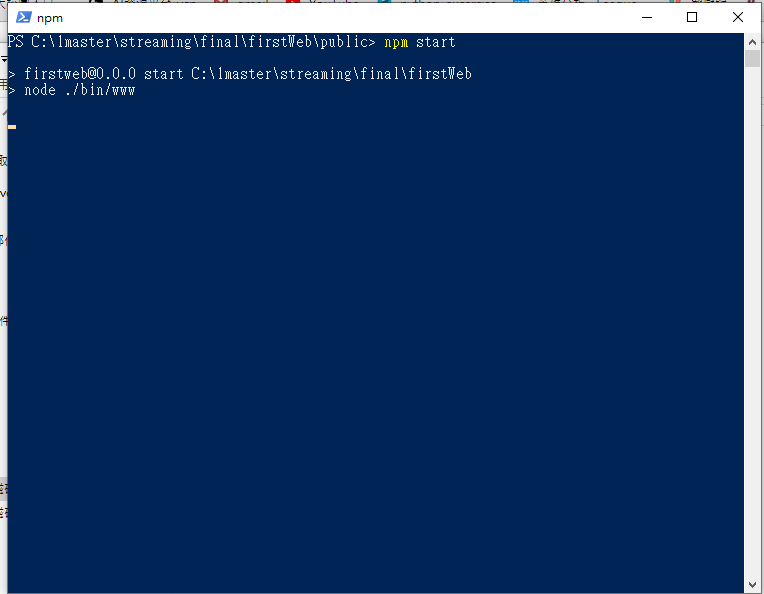
上圖為合併過後的樣子，裡面的路徑需要根據html檔案所在位置而作

改變。

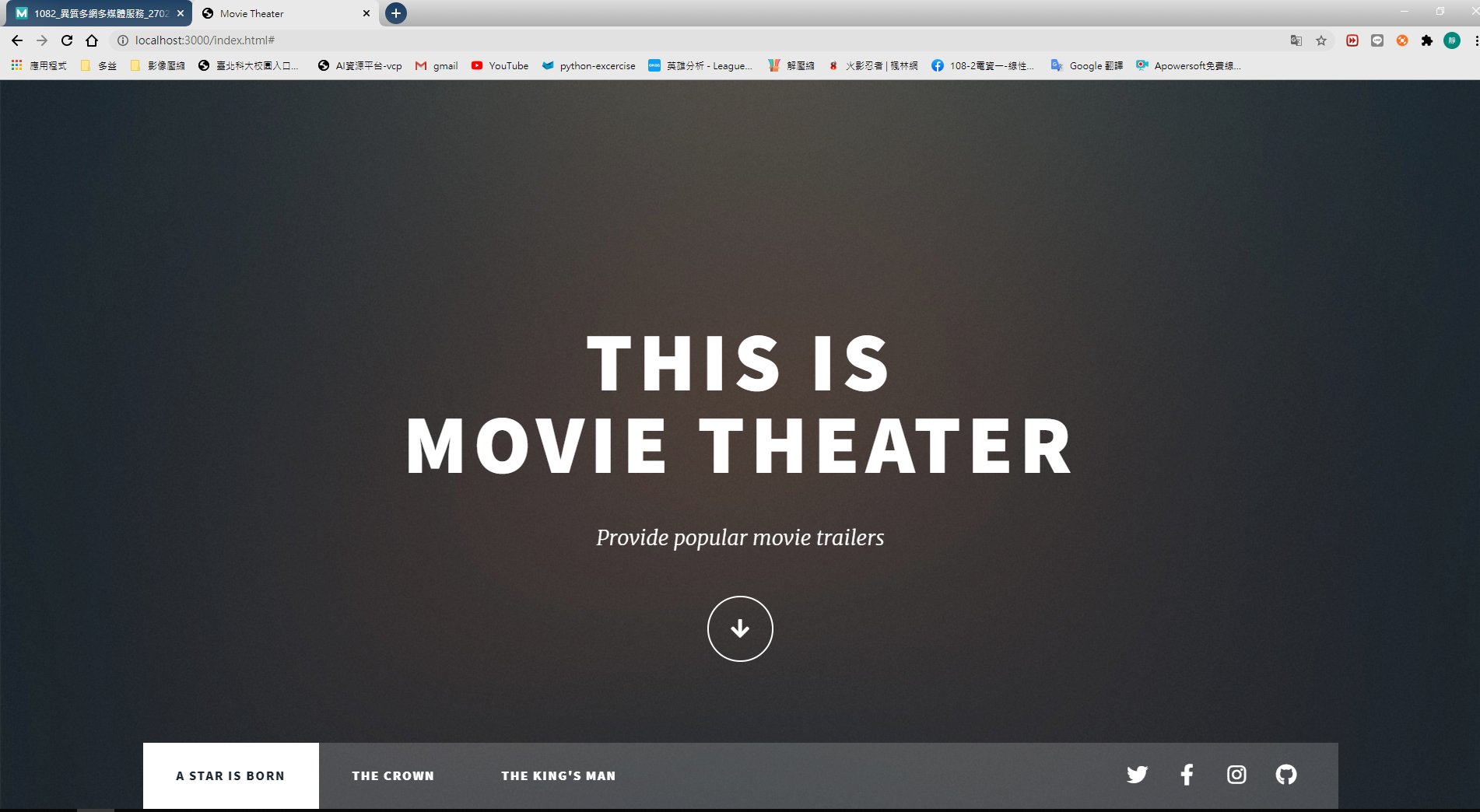
1. 以上動作即完成需要用的mpd

使用nodejs + express 當作server端，html5當作client端

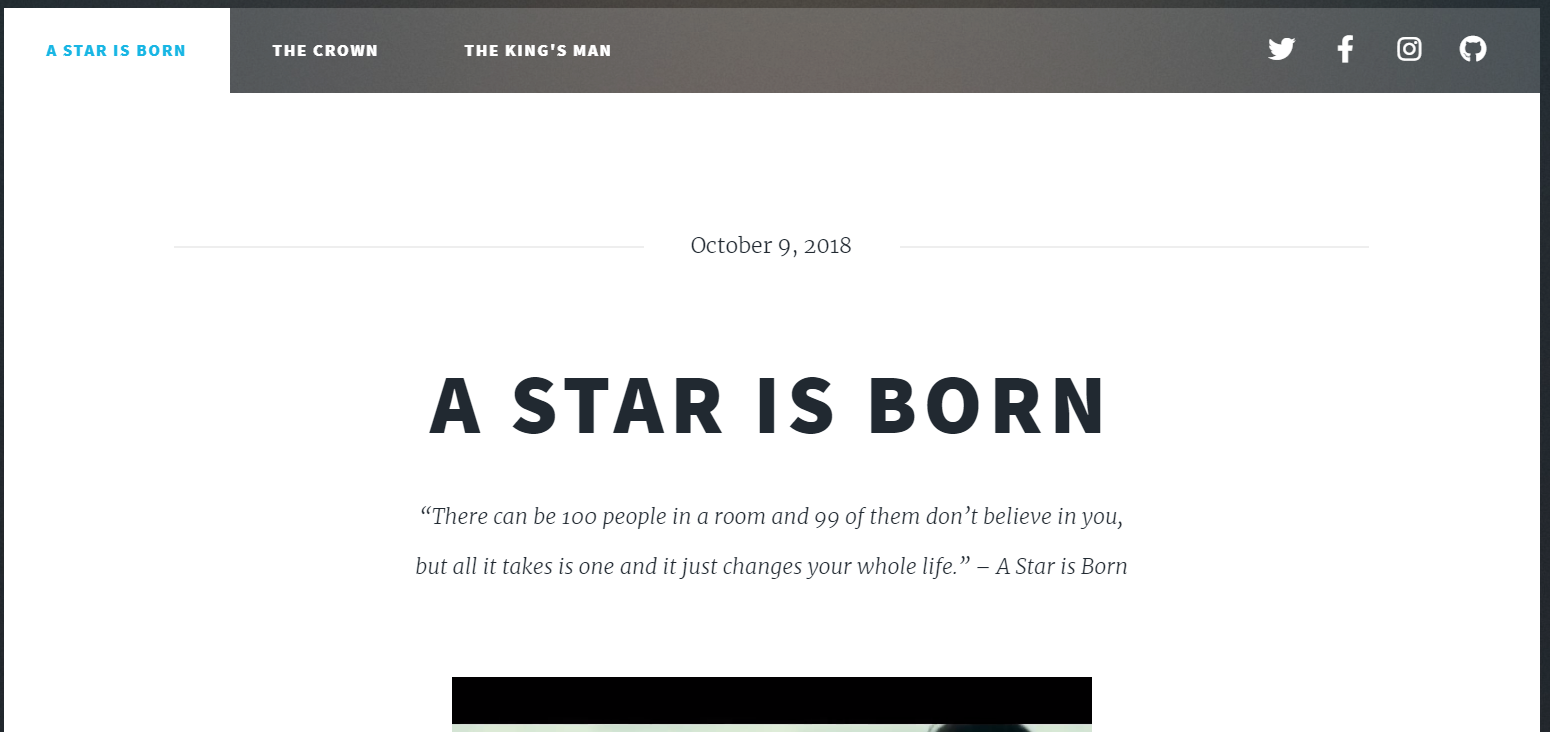
在public資料夾下執行 npm start，網站就架好了，如下圖即連線成功



在chrome打上<http://localhost:3000/index.html> ，即可連上首頁

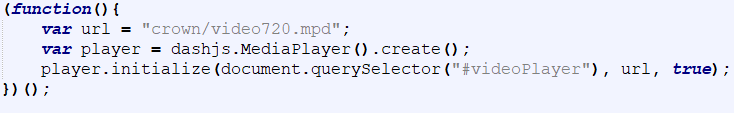


有電影選擇欄可供選擇，接下來為電影簡介

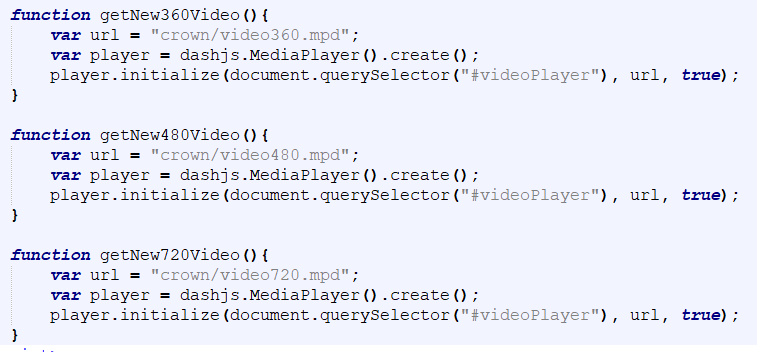


使用dashjs，結合JavaScript，在網頁上播放影片





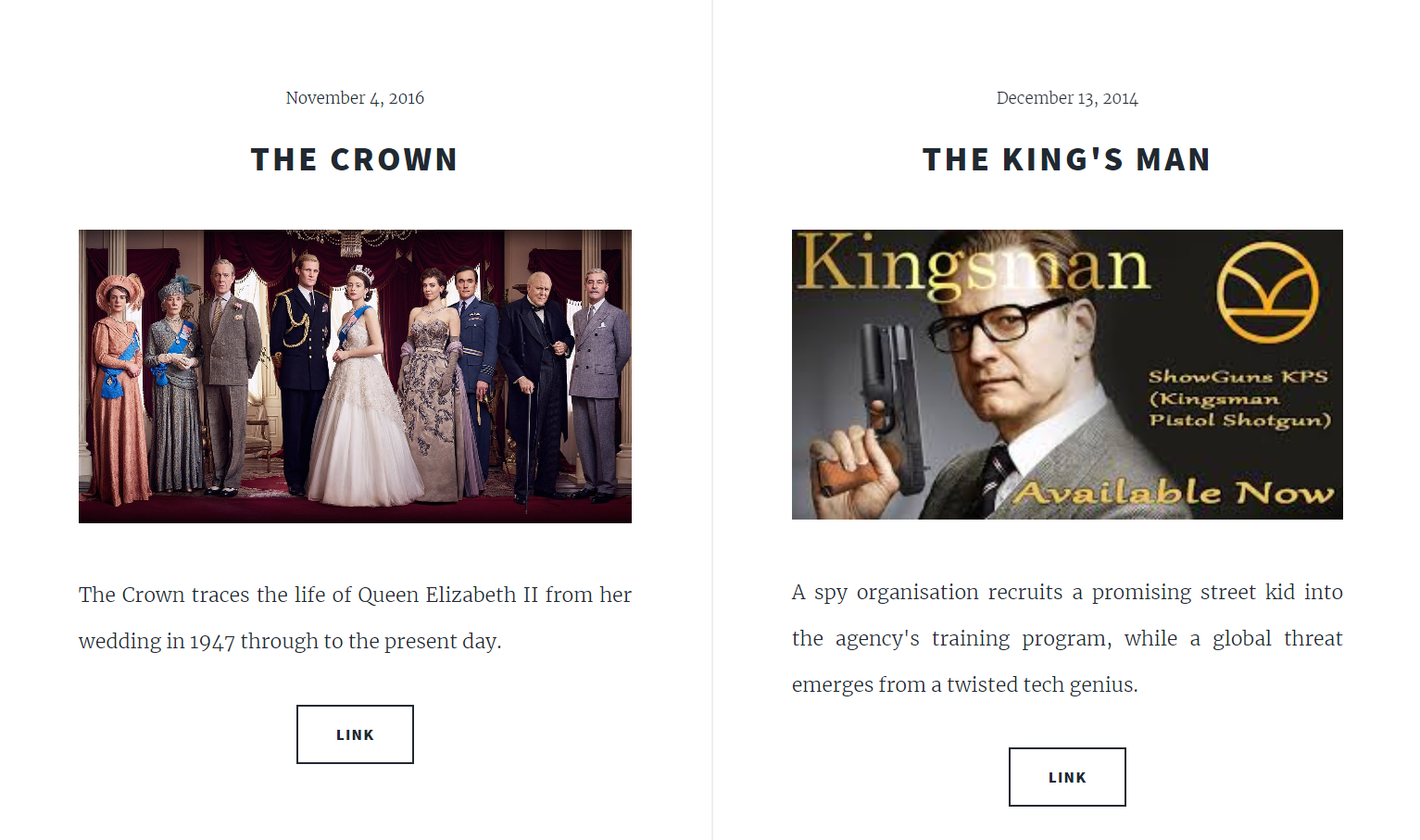
透過JavaScript實踐選擇畫素的function



成果如下圖

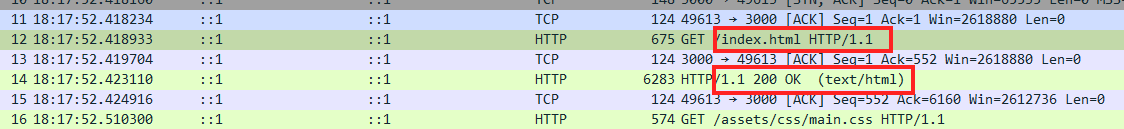


接下來為電影推薦的部分，點選電影名稱及LINK都可以連結到電影網頁

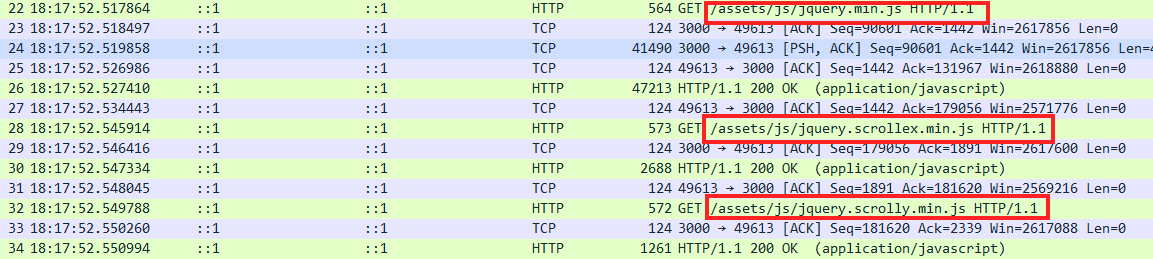


* **Wireshark之封包觀察結果及分析**

Client端開啟index.html，可透過封包觀察到網路連線正常



接下來為網站需要的一些js檔



還有一些需要載入的圖片



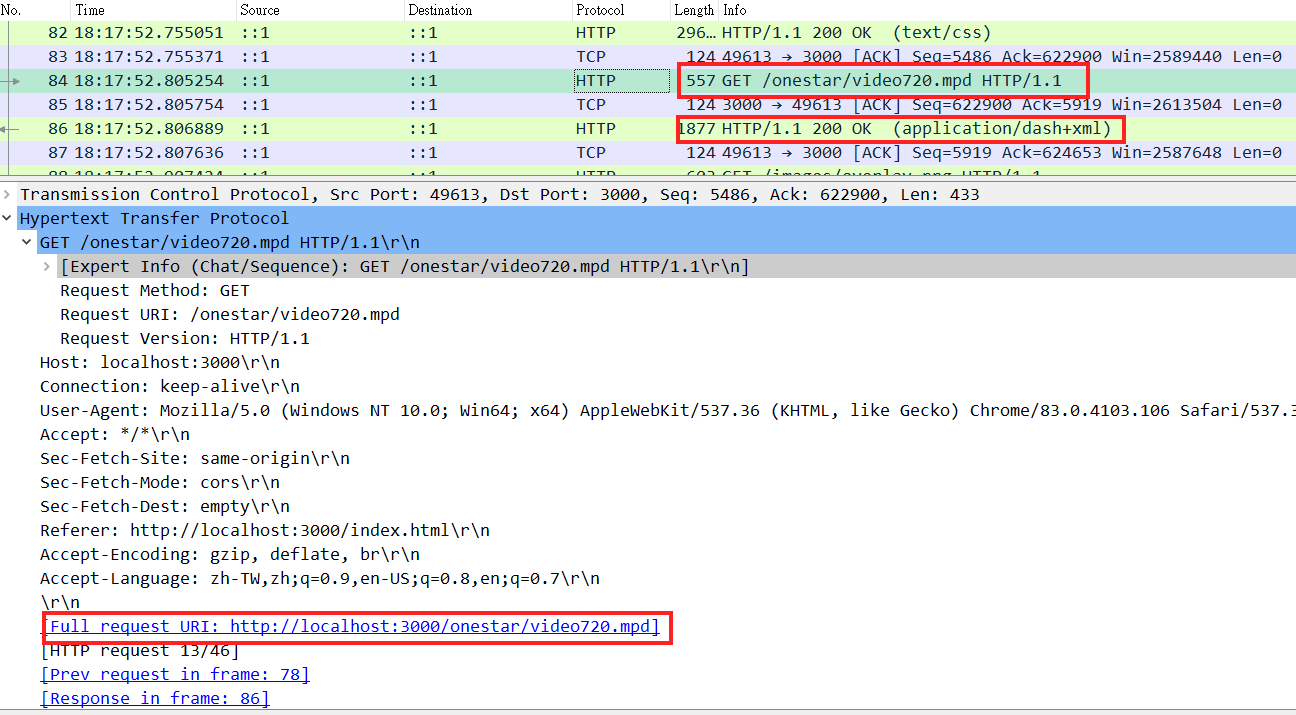
Client端要求畫質為720p的影片(我以720p為例)，video720.mpd為我

的mpd檔名稱，client使用GET的方式請求video720.mpd，我們server

收到請求之後回傳200 OK，代表回傳成功，並且可以看到Content-Type變

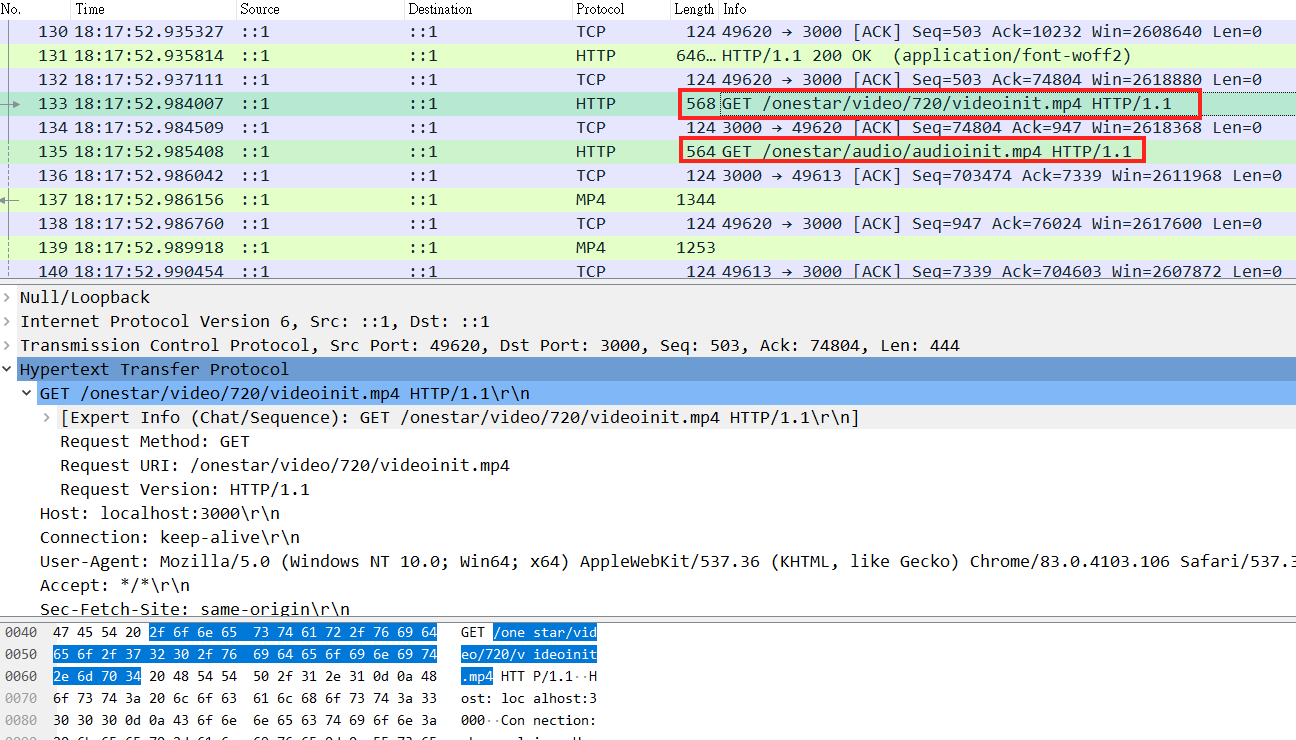
成application/dash+xml這說明我們目前的檔案傳輸格式application且檔

案類型為dash+xml。

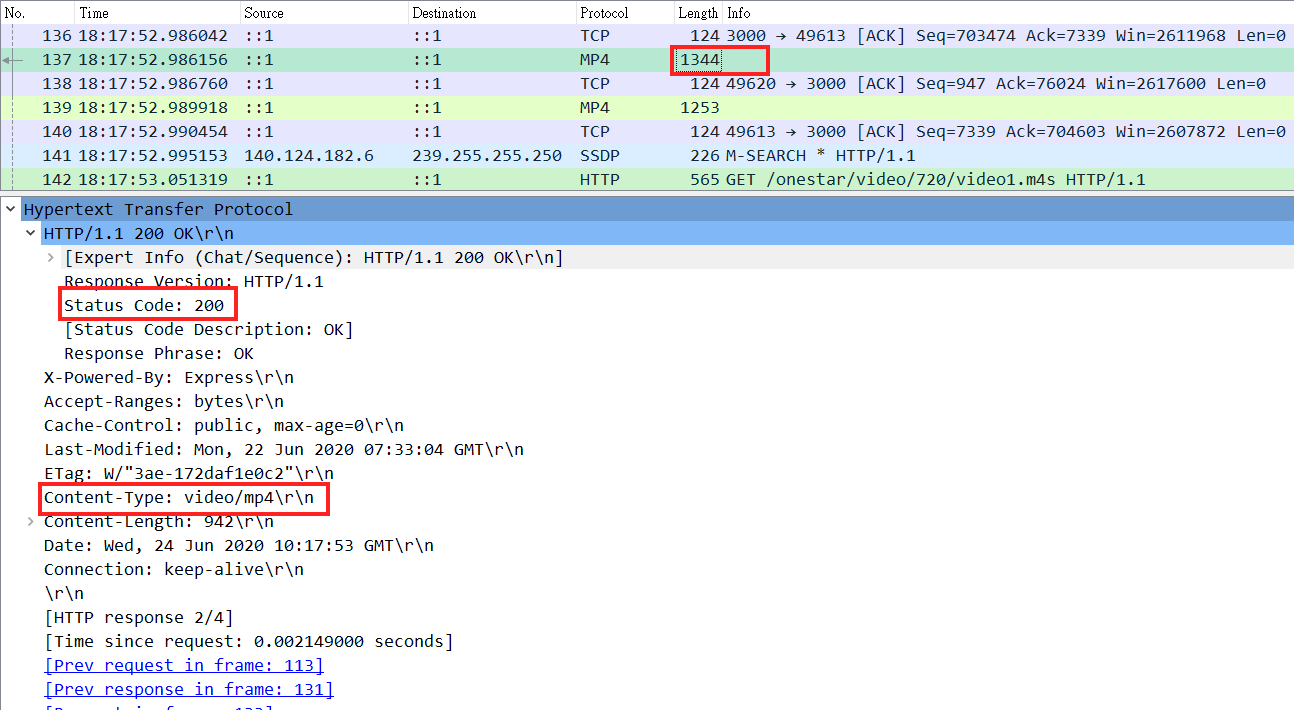


在開始串流播放影片之前，我們要會先載入videoinit.mp4與

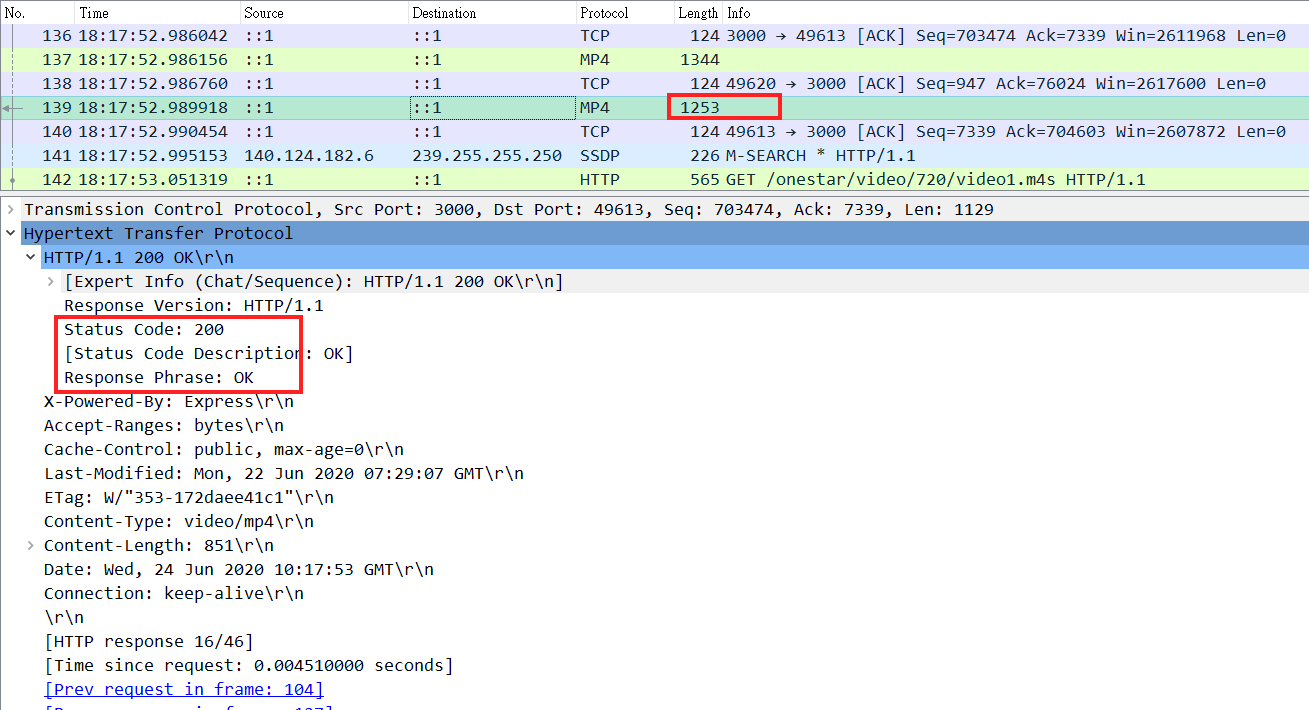
audioinit.mp4，這兩個檔案在切割segment時產生的



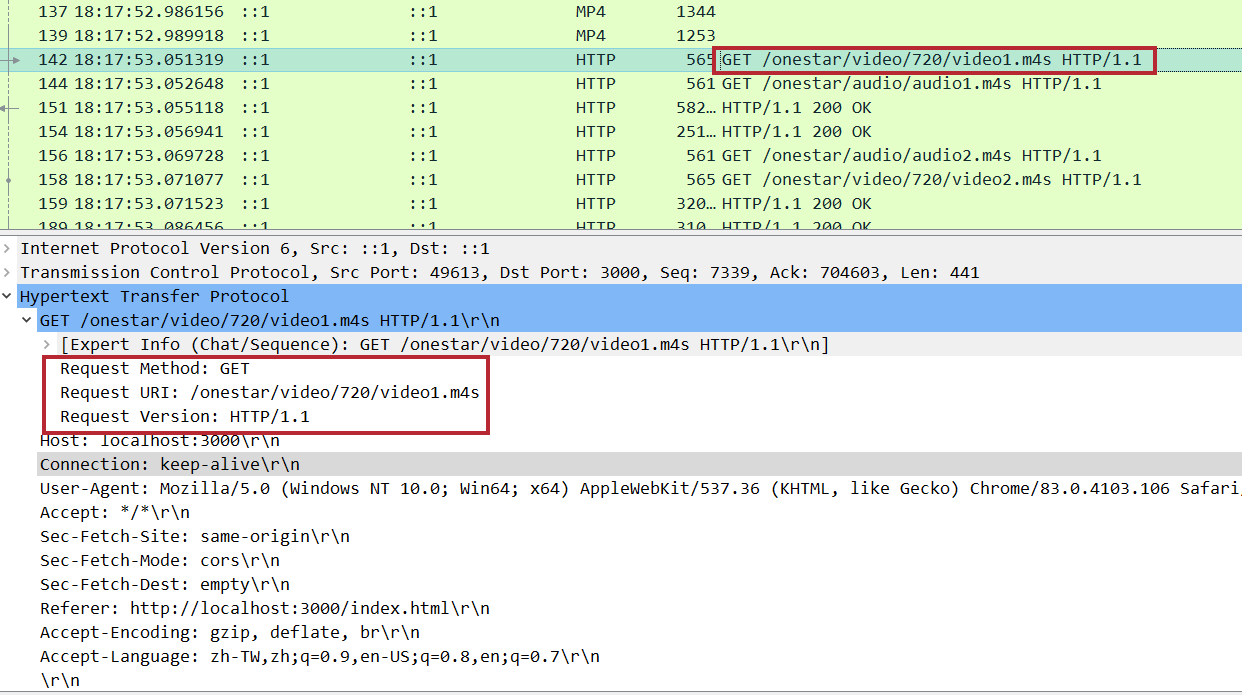
此為videoinit 回傳的封包，可以看到200回傳成功



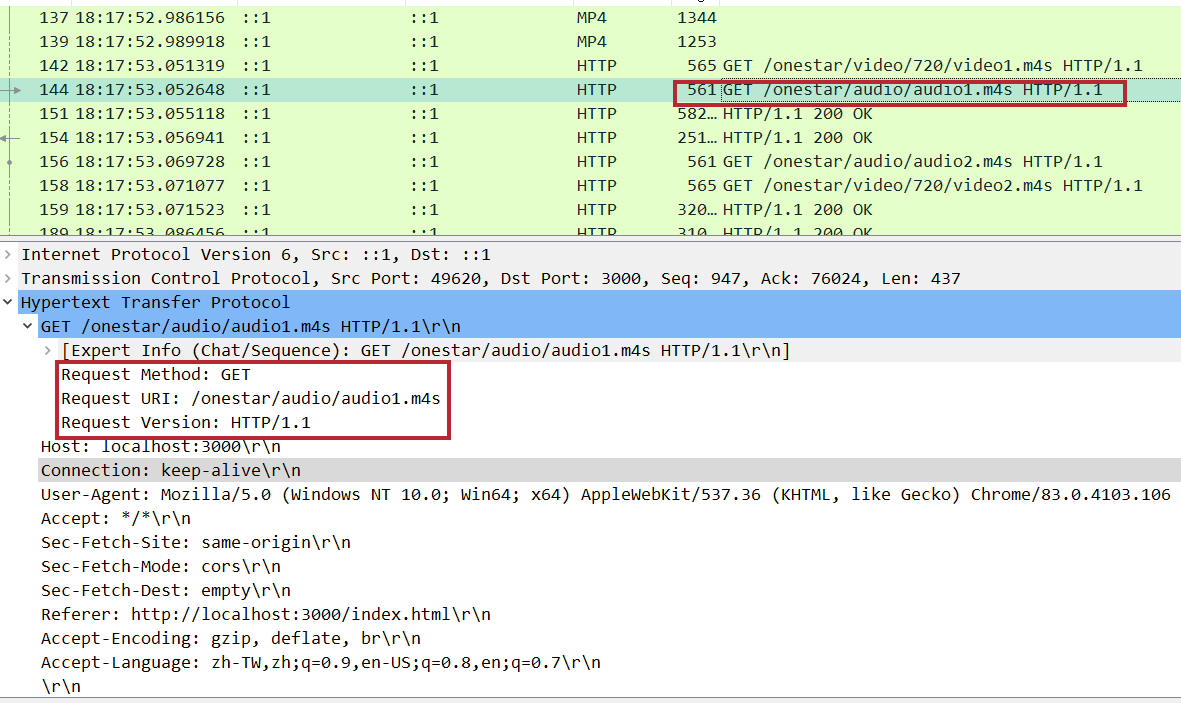
此為audioinit 回傳的封包，可以看到200回傳成功



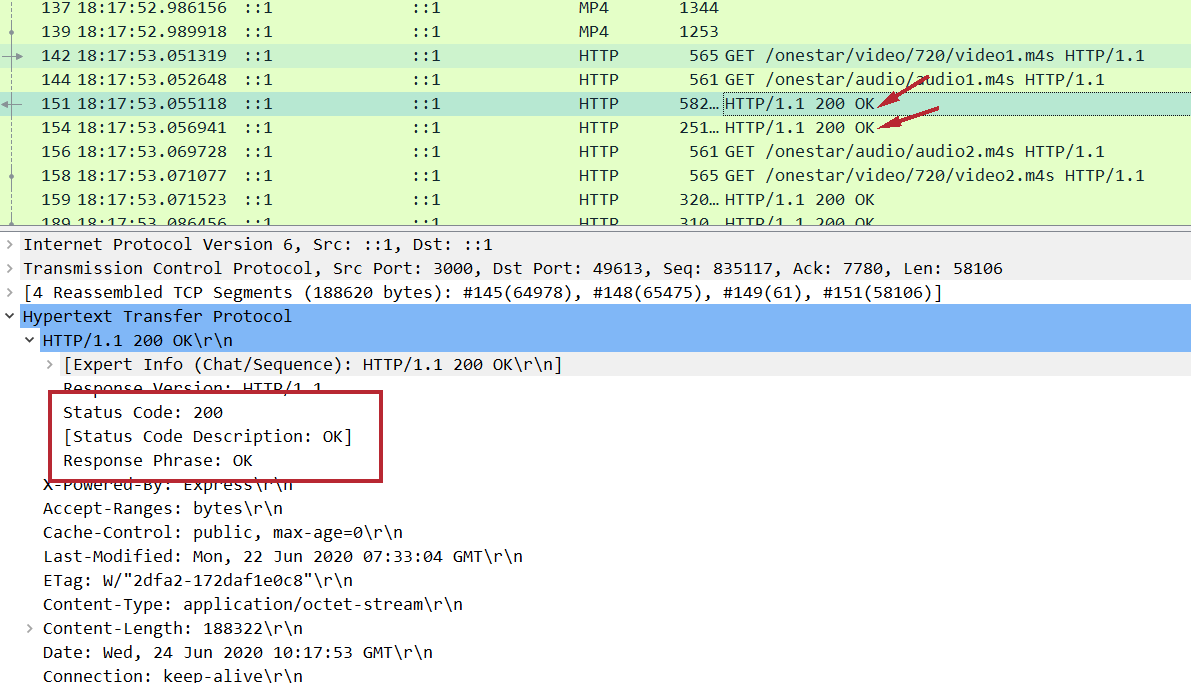
Client使用GET方式請求video1.m4s



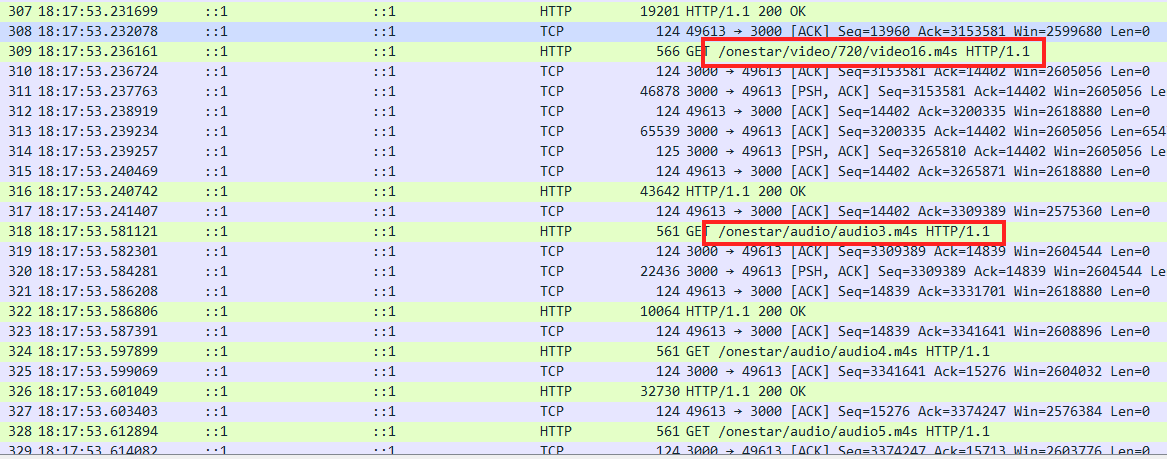
Client使用GET方式請求audio1.m4s，

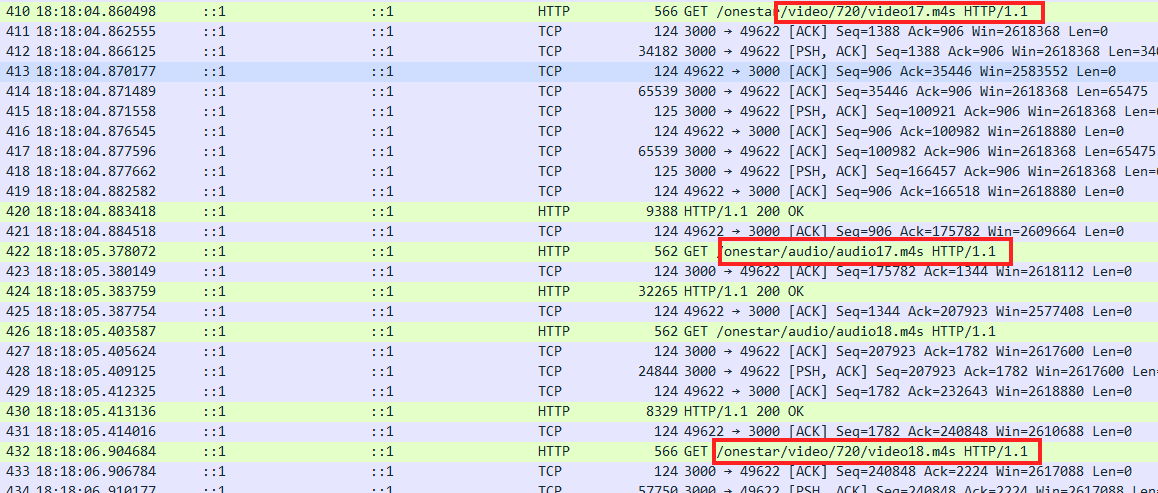


接下來兩個封包為成功回傳之封包



接下來為串流到client端的封包，陸續有video$number$.m4s與audio$number$.m4s之封包





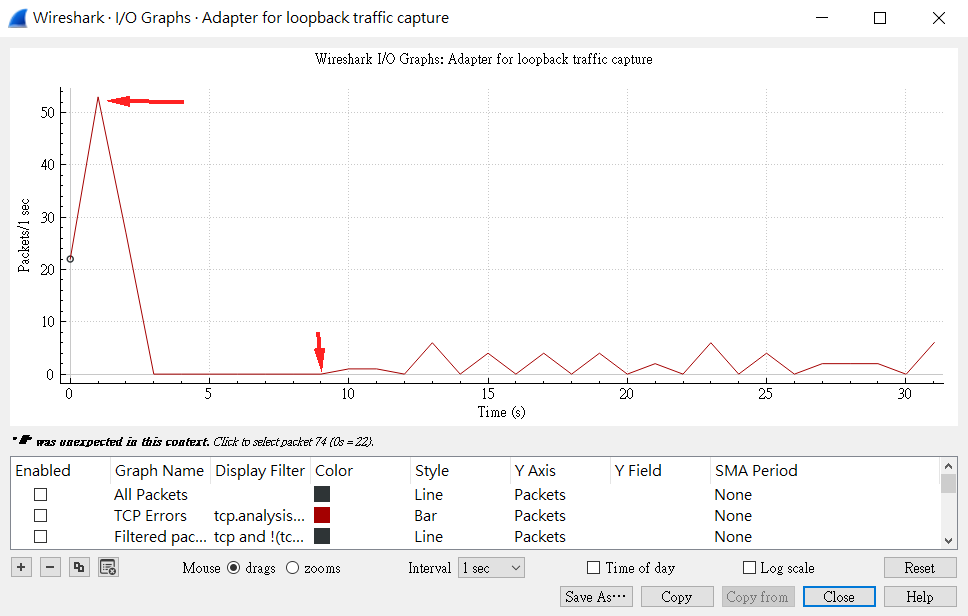
0~3秒封包數最多的地方為開啟網頁時，client要求載入網頁資訊時產生

的，包括網頁原始碼、圖片…等

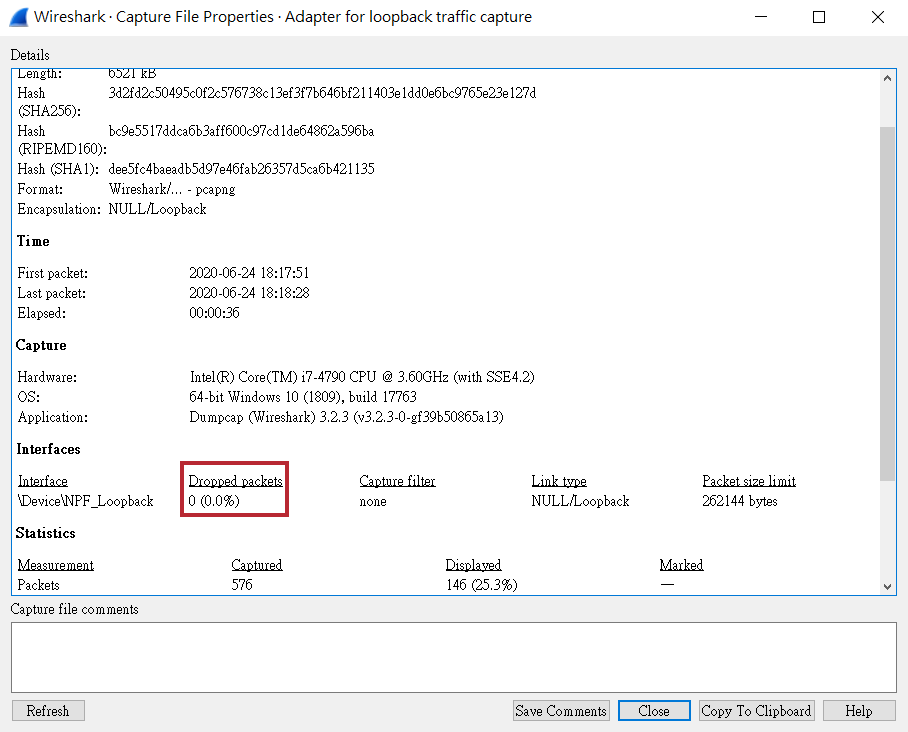
3~9秒封包數趨近為0，代表client端沒有特別要求載入資訊

9秒開始陸續有擷取到封包，代表client端點選了預告片，開始要求

mpd、videom4s、audiom4s…等檔案

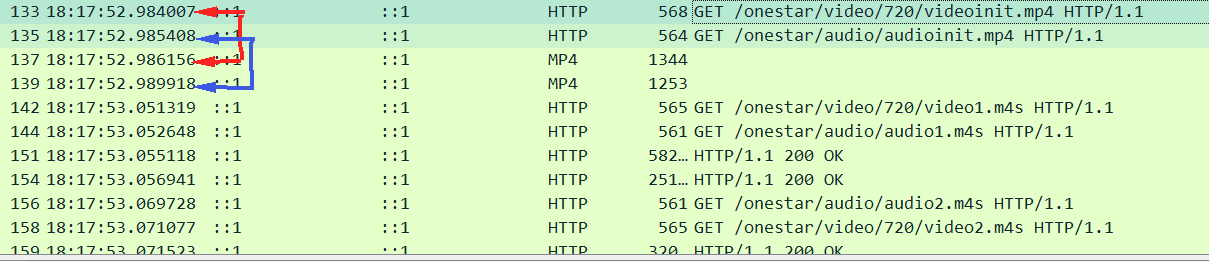


透過capture file properties顯示沒有封包丟失



wireshark可以簡單地看到client送出請求到server回傳的delay時間，下

圖以請求videoinit.mp4與audioinit.mp4為例



* **心得感想**

這次的期末報告花了許多時間完成，但也學到了很多，從一開始對一個原始mp4檔開始做處理，分割影片與影音檔，分別對他們切segment，在串流到網頁上去，都一步一步的完成了，對我來說非常有成就感，中途除了對影片作處理之外，透過dashjs在網頁上播放影片的用法，我也是第一次接觸到，當我要建立三個按鈕來切換畫質的過程遇到了困難，我用javascrip的function來切換src，一開始都沒有成功，後來換了寫法，影片就能成功的切換畫質了。除了切換畫質的部分，其他都還蠻順利的就完成了，學到的知識都很受用，會希望透過接下來暑假的時間有更深的研究。