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
   ```

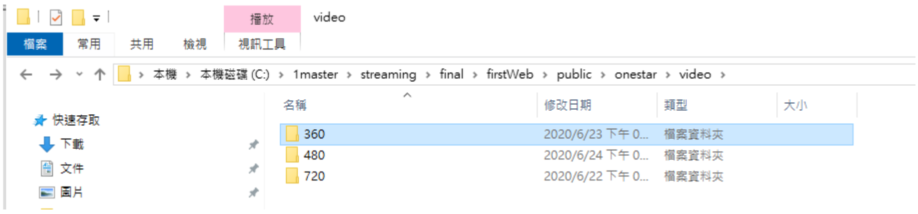
   取得影音檔指令:

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

3. 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
   ```

4. 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



本機 › 本機磁碟 (C:) › 1master › streaming › final › firstWeb › public › onestar › video ›

名稱	修改日期	類型	大小
360	2020/6/23 下午 0...	檔案資料夾	
480	2020/6/24 下午 0...	檔案資料夾	
720	2020/6/22 下午 0...	檔案資料夾	

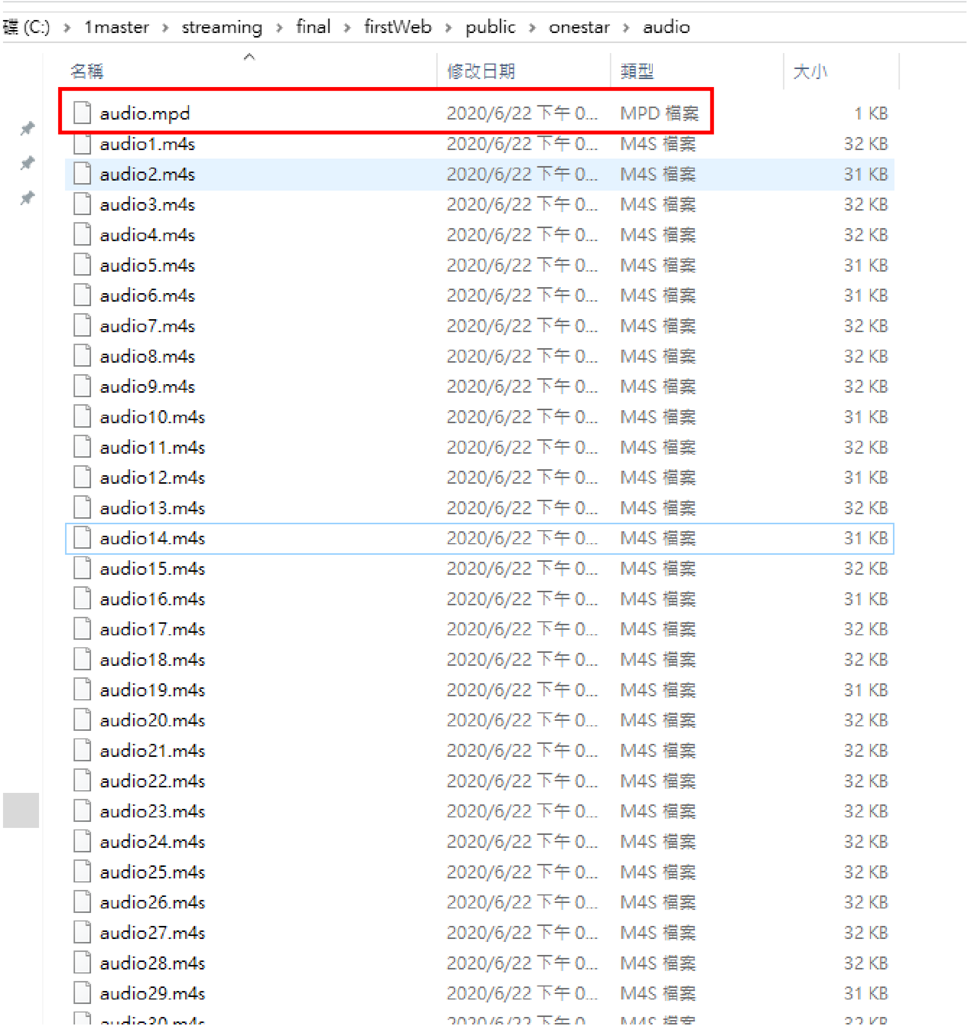
5. 先切割影音檔，透過 mp4box

   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

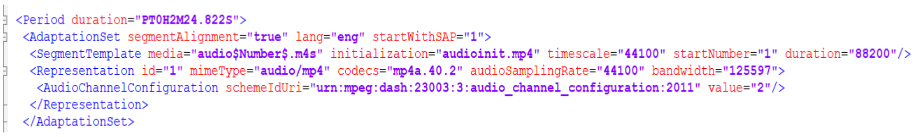
audio.mpd 裡面有我需要的音軌



碟 (C:) › 1master › streaming › final › firstWeb › public › onestar › audio

名稱	修改日期	類型	大小
audio.mpd	2020/6/22 下午 0...	MPD 檔案	1 KB
audio1.m4s	2020/6/22 下午 0...	M4S 檔案	32 KB
audio2.m4s	2020/6/22 下午 0...	M4S 檔案	31 KB
audio3.m4s	2020/6/22 下午 0...	M4S 檔案	32 KB
audio4.m4s	2020/6/22 下午 0...	M4S 檔案	32 KB
audio5.m4s	2020/6/22 下午 0...	M4S 檔案	31 KB
audio6.m4s	2020/6/22 下午 0...	M4S 檔案	31 KB
audio7.m4s	2020/6/22 下午 0...	M4S 檔案	32 KB
audio8.m4s	2020/6/22 下午 0...	M4S 檔案	32 KB
audio9.m4s	2020/6/22 下午 0...	M4S 檔案	32 KB
audio10.m4s	2020/6/22 下午 0...	M4S 檔案	31 KB
audio11.m4s	2020/6/22 下午 0...	M4S 檔案	32 KB
audio12.m4s	2020/6/22 下午 0...	M4S 檔案	31 KB
audio13.m4s	2020/6/22 下午 0...	M4S 檔案	32 KB
audio14.m4s	2020/6/22 下午 0...	M4S 檔案	31 KB
audio15.m4s	2020/6/22 下午 0...	M4S 檔案	32 KB
audio16.m4s	2020/6/22 下午 0...	M4S 檔案	31 KB
audio17.m4s	2020/6/22 下午 0...	M4S 檔案	32 KB
audio18.m4s	2020/6/22 下午 0...	M4S 檔案	32 KB
audio19.m4s	2020/6/22 下午 0...	M4S 檔案	31 KB
audio20.m4s	2020/6/22 下午 0...	M4S 檔案	32 KB
audio21.m4s	2020/6/22 下午 0...	M4S 檔案	32 KB
audio22.m4s	2020/6/22 下午 0...	M4S 檔案	32 KB
audio23.m4s	2020/6/22 下午 0...	M4S 檔案	32 KB
audio24.m4s	2020/6/22 下午 0...	M4S 檔案	32 KB
audio25.m4s	2020/6/22 下午 0...	M4S 檔案	32 KB
audio26.m4s	2020/6/22 下午 0...	M4S 檔案	32 KB
audio27.m4s	2020/6/22 下午 0...	M4S 檔案	32 KB
audio28.m4s	2020/6/22 下午 0...	M4S 檔案	32 KB
audio29.m4s	2020/6/22 下午 0...	M4S 檔案	31 KB
audio30.m4s	2020/6/22 下午 0	M4S 檔案	32 KB

音軌如下圖



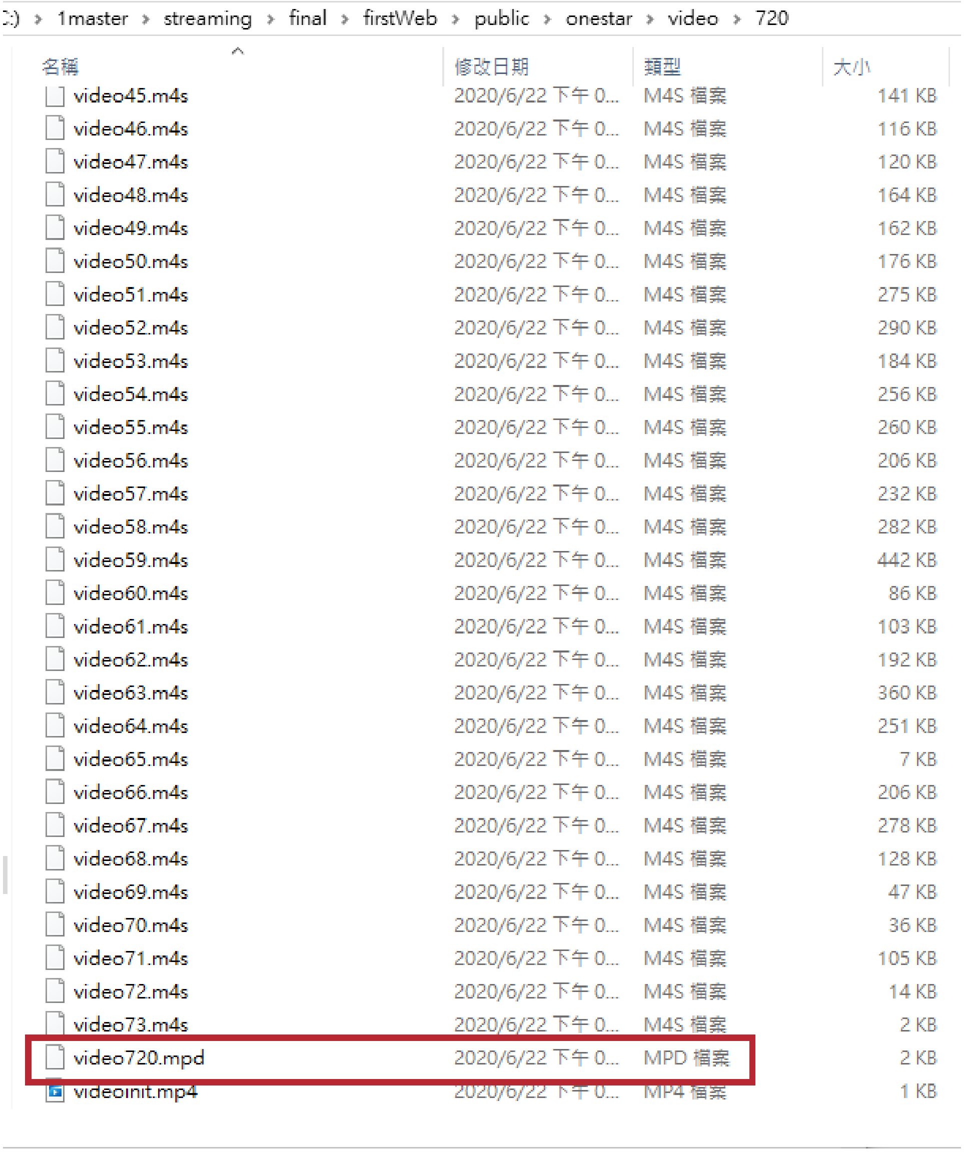
```
<Period duration="PT0H2M24.822S">
 <AdaptationSet segmentAlignment="true" lang="eng" startWithSAP="1">
  <SegmentTemplate media="audio$Number$.m4s" initialization="audioinit.mp4" timescale="44100" startNumber="1" duration="88200"/>
  <Representation id="1" mimeType="audio/mp4" codecs="mp4a.40.2" audioSamplingRate="44100" bandwidth="125597">
   <AudioChannelConfiguration schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="2"/>
  </Representation>
 </AdaptationSet>
```

6. 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
```

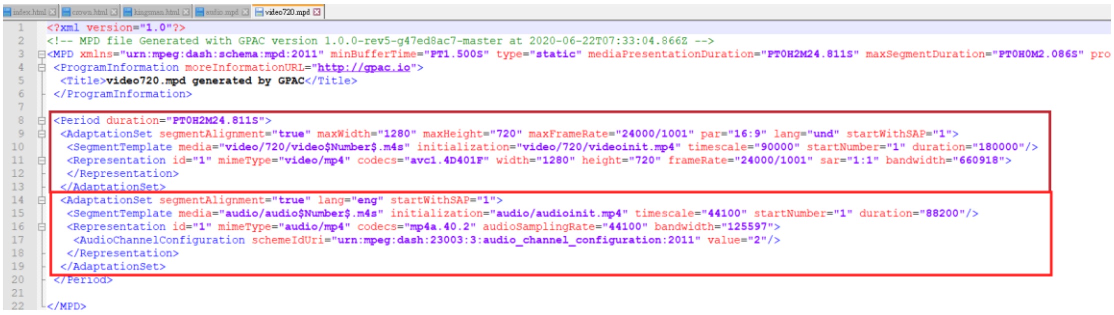


:) › 1master › streaming › final › firstWeb › public › onestar › video › 720

名稱	修改日期	類型	大小
video45.m4s	2020/6/22 下午 0...	M4S 檔案	141 KB
video46.m4s	2020/6/22 下午 0...	M4S 檔案	116 KB
video47.m4s	2020/6/22 下午 0...	M4S 檔案	120 KB
video48.m4s	2020/6/22 下午 0...	M4S 檔案	164 KB
video49.m4s	2020/6/22 下午 0...	M4S 檔案	162 KB
video50.m4s	2020/6/22 下午 0...	M4S 檔案	176 KB
video51.m4s	2020/6/22 下午 0...	M4S 檔案	275 KB
video52.m4s	2020/6/22 下午 0...	M4S 檔案	290 KB
video53.m4s	2020/6/22 下午 0...	M4S 檔案	184 KB
video54.m4s	2020/6/22 下午 0...	M4S 檔案	256 KB
video55.m4s	2020/6/22 下午 0...	M4S 檔案	260 KB
video56.m4s	2020/6/22 下午 0...	M4S 檔案	206 KB
video57.m4s	2020/6/22 下午 0...	M4S 檔案	232 KB
video58.m4s	2020/6/22 下午 0...	M4S 檔案	282 KB
video59.m4s	2020/6/22 下午 0...	M4S 檔案	442 KB
video60.m4s	2020/6/22 下午 0...	M4S 檔案	86 KB
video61.m4s	2020/6/22 下午 0...	M4S 檔案	103 KB
video62.m4s	2020/6/22 下午 0...	M4S 檔案	192 KB
video63.m4s	2020/6/22 下午 0...	M4S 檔案	360 KB
video64.m4s	2020/6/22 下午 0...	M4S 檔案	251 KB
video65.m4s	2020/6/22 下午 0...	M4S 檔案	7 KB
video66.m4s	2020/6/22 下午 0...	M4S 檔案	206 KB
video67.m4s	2020/6/22 下午 0...	M4S 檔案	278 KB
video68.m4s	2020/6/22 下午 0...	M4S 檔案	128 KB
video69.m4s	2020/6/22 下午 0...	M4S 檔案	47 KB
video70.m4s	2020/6/22 下午 0...	M4S 檔案	36 KB
video71.m4s	2020/6/22 下午 0...	M4S 檔案	105 KB
video72.m4s	2020/6/22 下午 0...	M4S 檔案	14 KB
video73.m4s	2020/6/22 下午 0...	M4S 檔案	2 KB
video720.mpd	2020/6/22 下午 0...	MPD 檔案	2 KB
videoinit.mp4	2020/6/22 下午 0...	MP4 檔案	1 KB

產生許多 segment 與一個 mpd 檔，即為影片軌

7. 合併 audio.mpd 與 video.mpd



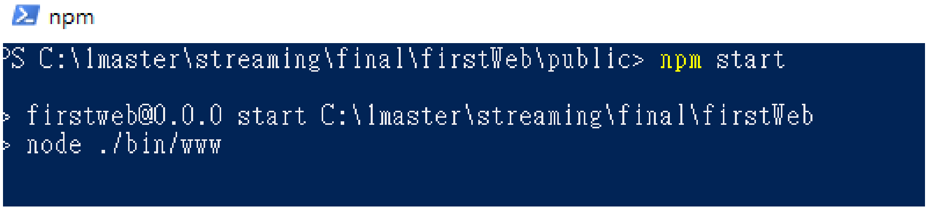
```xml
<?xml version="1.0"?>
<!-- MPD file Generated with GPAC version 1.0.0-rev5-g47ed8ac7-master at 2020-06-22T07:33:04.866Z -->
<MPD xmlns="urn:mpeg:dash:schema:mpd:2011" minBufferTime="PT1.500S" type="static" mediaPresentationDuration="PT0H2M24.811S" maxSegmentDuration="PT0H0M2.086S" pro
 <ProgramInformation moreInformationURL="http://gpac.io">
  <Title>video720.mpd generated by GPAC</Title>
 </ProgramInformation>

 <Period duration="PT0H2M24.811S">
  <AdaptationSet segmentAlignment="true" maxWidth="1280" maxHeight="720" maxFrameRate="24000/1001" par="16:9" lang="und" startWithSAP="1">
   <SegmentTemplate media="video/720/video$Number$.m4s" initialization="video/720/videoinit.mp4" timescale="90000" startNumber="1" duration="180000"/>
   <Representation id="1" mimeType="video/mp4" codecs="avc1.4D401F" width="1280" height="720" frameRate="24000/1001" sar="1:1" bandwidth="660918">
   </Representation>
  </AdaptationSet>
  <AdaptationSet segmentAlignment="true" lang="eng" startWithSAP="1">
   <SegmentTemplate media="audio/audio$Number$.m4s" initialization="audio/audioinit.mp4" timescale="44100" startNumber="1" duration="88200"/>
   <Representation id="1" mimeType="audio/mp4" codecs="mp4a.40.2" audioSamplingRate="44100" bandwidth="125597">
    <AudioChannelConfiguration schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="2"/>
   </Representation>
  </AdaptationSet>
 </Period>

</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。

8. 以上動作即完成需要用的 mpd

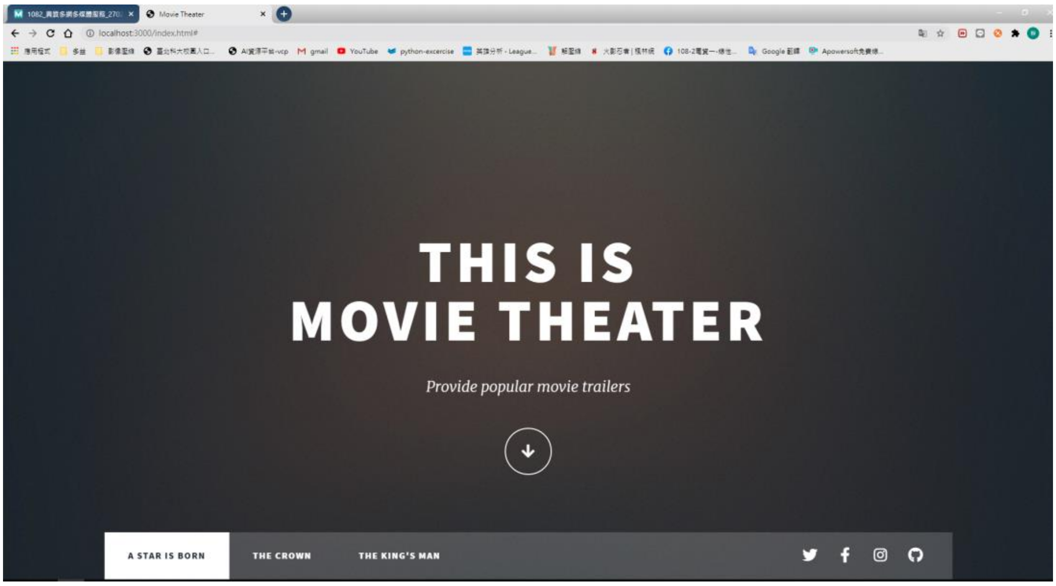
使用 nodejs + express 當作 server 端，html5 當作 client 端

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



```
PS C:\1master\streaming\final\firstWeb\public> npm start

> firstweb@0.0.0 start C:\1master\streaming\final\firstWeb
> node ./bin/www
```

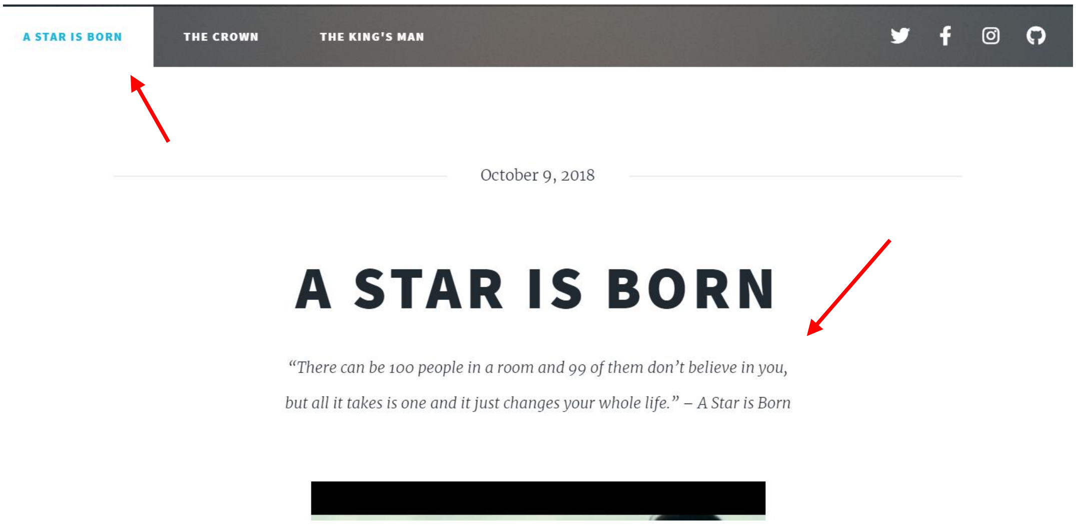
在 chrome 打上 http://localhost:3000/index.html ，即可連上首頁



THIS IS MOVIE THEATER

Provide popular movie trailers

有電影選擇欄可供選擇，接下來為電影簡介

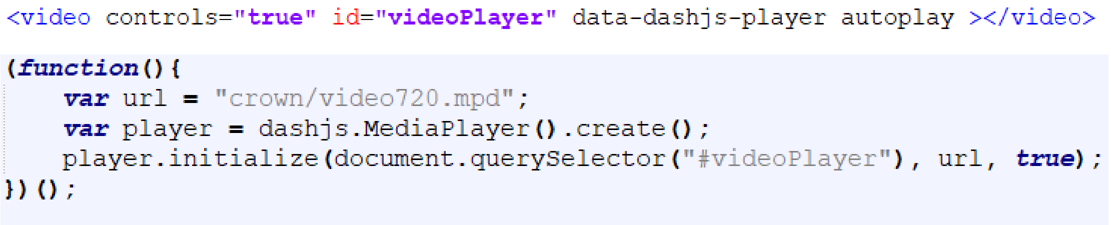


October 9, 2018

A STAR IS BORN

"There can be 100 people in a room and 99 of them don't believe in you, but all it takes is one and it just changes your whole life." – A Star is Born

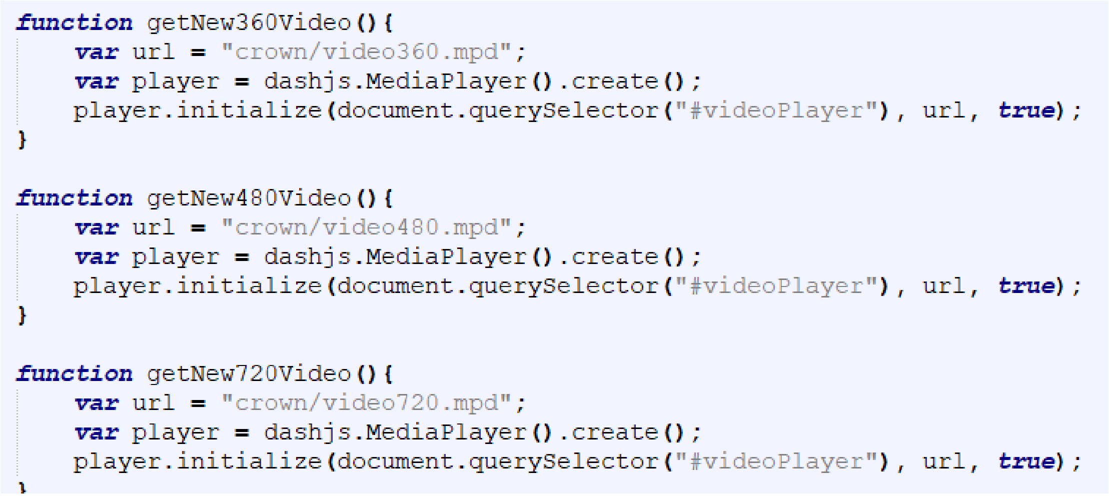
使用 dashjs，結合 JavaScript，在網頁上播放影片



```
<video controls="true" id="videoPlayer" data-dashjs-player autoplay ></video>
```

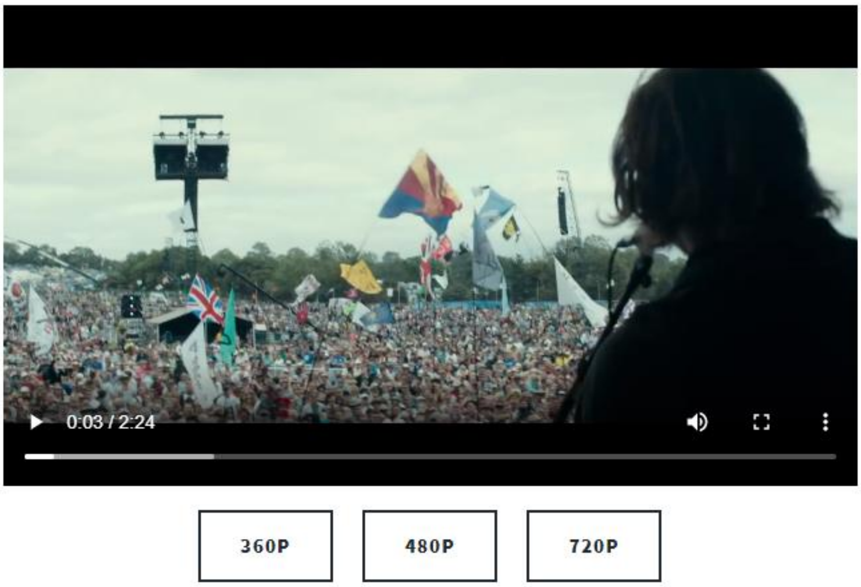
```
(function(){
    var url = "crown/video720.mpd";
    var player = dashjs.MediaPlayer().create();
    player.initialize(document.querySelector("#videoPlayer"), url, true);
})();
```

透過 JavaScript 實踐選擇畫素的 function

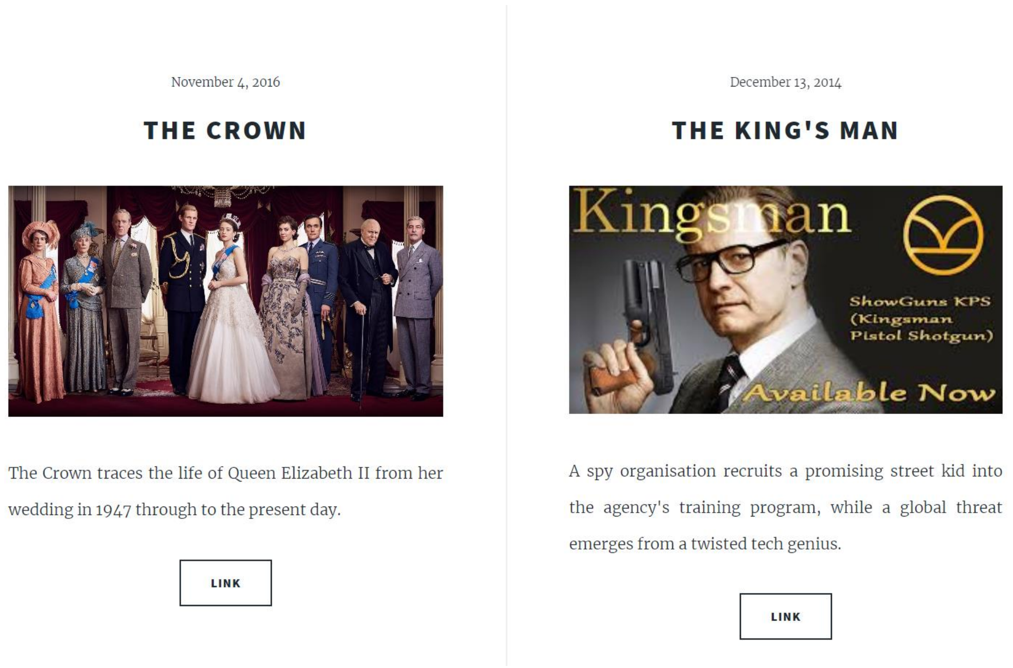


```
function getNew360Video(){
    var url = "crown/video360.mpd";
    var player = dashjs.MediaPlayer().create();
    player.initialize(document.querySelector("#videoPlayer"), url, true);
}

function getNew480Video(){
    var url = "crown/video480.mpd";
    var player = dashjs.MediaPlayer().create();
    player.initialize(document.querySelector("#videoPlayer"), url, true);
}

function getNew720Video(){
    var url = "crown/video720.mpd";
    var player = dashjs.MediaPlayer().create();
    player.initialize(document.querySelector("#videoPlayer"), url, true);
}
```

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



November 4, 2016

THE CROWN

The Crown traces the life of Queen Elizabeth II from her wedding in 1947 through to the present day.

LINK

December 13, 2014

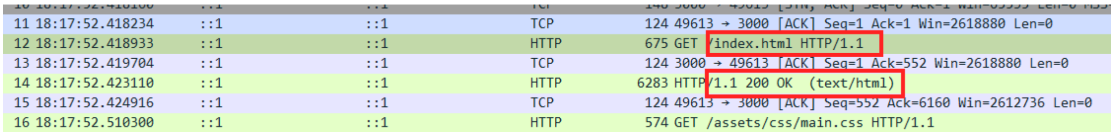
THE KING'S MAN

A spy organisation recruits a promising street kid into the agency's training program, while a global threat emerges from a twisted tech genius.

LINK

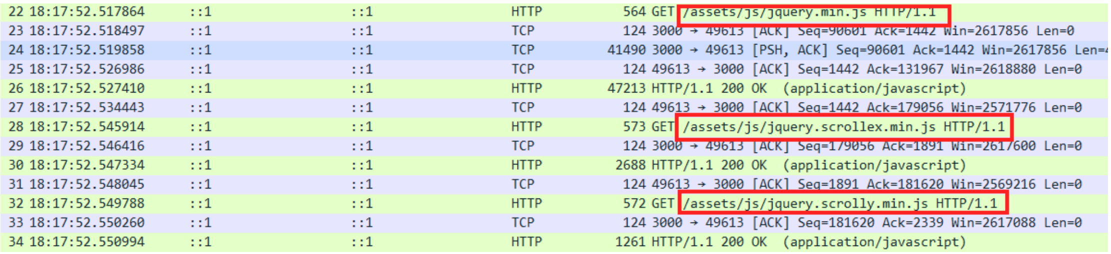
- **Wireshark 之封包觀察結果及分析**

Client 端開啟 index.html，可透過封包觀察到網路連線正常



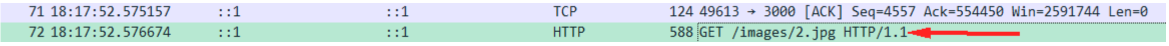
```
10 18:17:52.418100     ::1                ::1                TCP        140 3000 → 49613 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS
11 18:17:52.418234     ::1                ::1                TCP        124 49613 → 3000 [ACK] Seq=1 Ack=1 Win=2618880 Len=0
12 18:17:52.418933     ::1                ::1                HTTP       675 GET /index.html HTTP/1.1
13 18:17:52.419704     ::1                ::1                TCP        124 3000 → 49613 [ACK] Seq=1 Ack=552 Win=2618880 Len=0
14 18:17:52.423110     ::1                ::1                HTTP      6283 HTTP/1.1 200 OK  (text/html)
15 18:17:52.424916     ::1                ::1                TCP        124 49613 → 3000 [ACK] Seq=552 Ack=6160 Win=2612736 Len=0
16 18:17:52.510300     ::1                ::1                HTTP       574 GET /assets/css/main.css HTTP/1.1
```

接下來為網站需要的一些 js 檔



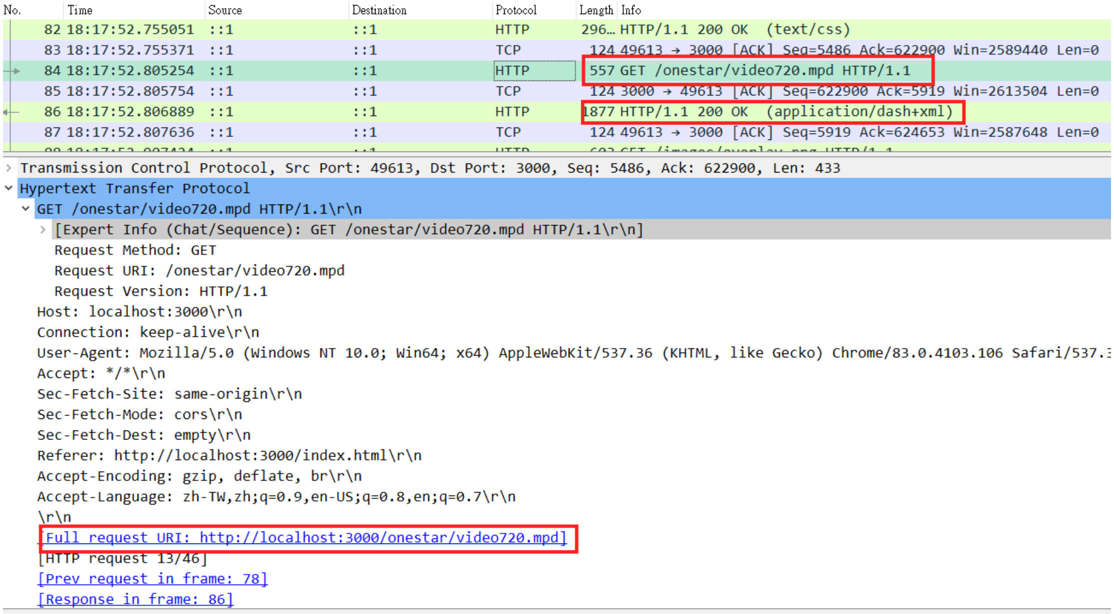
```
22 18:17:52.517864     ::1                ::1                HTTP       564 GET /assets/js/jquery.min.js HTTP/1.1
23 18:17:52.518497     ::1                ::1                TCP        124 3000 → 49613 [ACK] Seq=90601 Ack=1442 Win=2617856 Len=0
24 18:17:52.519858     ::1                ::1                TCP      41490 3000 → 49613 [PSH, ACK] Seq=90601 Ack=1442 Win=2617856 Len=4
25 18:17:52.526986     ::1                ::1                TCP        124 49613 → 3000 [ACK] Seq=1442 Ack=131967 Win=2618880 Len=0
26 18:17:52.527410     ::1                ::1                HTTP     47213 HTTP/1.1 200 OK  (application/javascript)
27 18:17:52.534443     ::1                ::1                TCP        124 49613 → 3000 [ACK] Seq=1442 Ack=179056 Win=2571776 Len=0
28 18:17:52.545914     ::1                ::1                HTTP       573 GET /assets/js/jquery.scrollex.min.js HTTP/1.1
29 18:17:52.546416     ::1                ::1                TCP        124 3000 → 49613 [ACK] Seq=179056 Ack=1891 Win=2617600 Len=0
30 18:17:52.547334     ::1                ::1                HTTP      2688 HTTP/1.1 200 OK  (application/javascript)
31 18:17:52.548045     ::1                ::1                TCP        124 49613 → 3000 [ACK] Seq=1891 Ack=181620 Win=2569216 Len=0
32 18:17:52.549788     ::1                ::1                HTTP       572 GET /assets/js/jquery.scrolly.min.js HTTP/1.1
33 18:17:52.550260     ::1                ::1                TCP        124 3000 → 49613 [ACK] Seq=181620 Ack=2339 Win=2617088 Len=0
34 18:17:52.550994     ::1                ::1                HTTP      1261 HTTP/1.1 200 OK  (application/javascript)
```

還有一些需要載入的圖片



```
71 18:17:52.575157     ::1                ::1                TCP        124 49613 → 3000 [ACK] Seq=4557 Ack=554450 Win=2591744 Len=0
72 18:17:52.576674     ::1                ::1                HTTP       588 GET /images/2.jpg HTTP/1.1
```

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。

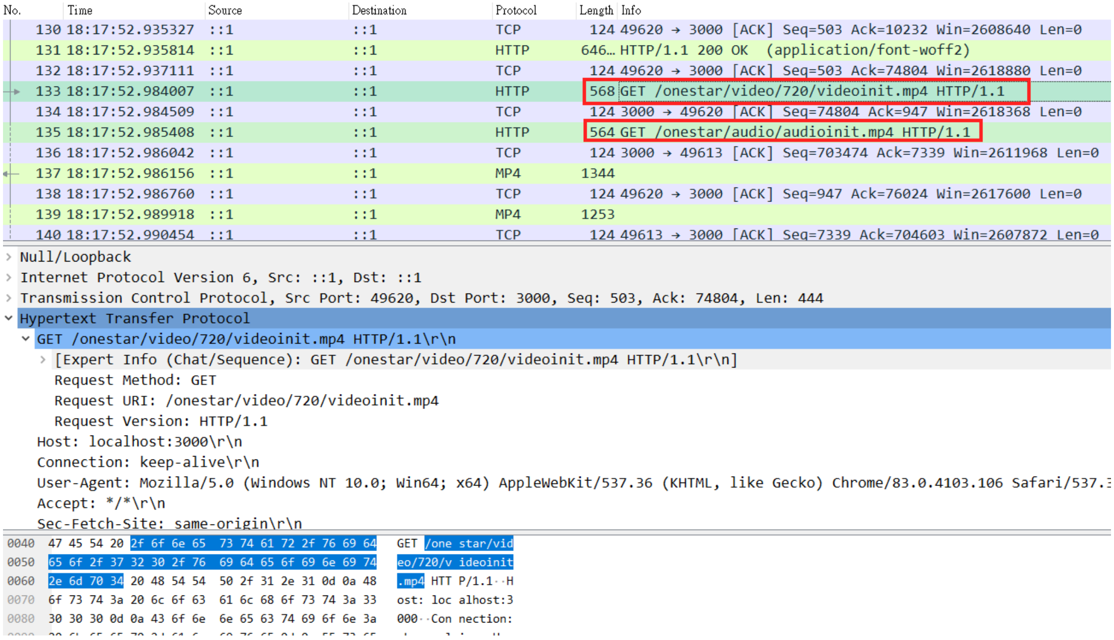


```
No.   Time              Source     Destination   Protocol   Length  Info
   82 18:17:52.755051   ::1        ::1           HTTP       296… HTTP/1.1 200 OK  (text/css)
   83 18:17:52.755371   ::1        ::1           TCP         124 49613 → 3000 [ACK] Seq=5486 Ack=622900 Win=2589440 Len=0
   84 18:17:52.805254   ::1        ::1           HTTP        557 GET /onestar/video720.mpd HTTP/1.1
   85 18:17:52.805754   ::1        ::1           TCP         124 3000 → 49613 [ACK] Seq=622900 Ack=5919 Win=2613504 Len=0
   86 18:17:52.806889   ::1        ::1           HTTP       1877 HTTP/1.1 200 OK  (application/dash+xml)
   87 18:17:52.807636   ::1        ::1           TCP         124 49613 → 3000 [ACK] Seq=5919 Ack=624653 Win=2587648 Len=0
```

```
> Transmission Control Protocol, Src Port: 49613, Dst Port: 3000, Seq: 5486, Ack: 622900, Len: 433
v Hypertext Transfer Protocol
  v GET /onestar/video720.mpd HTTP/1.1\r\n
    > [Expert Info (Chat/Sequence): GET /onestar/video720.mpd HTTP/1.1\r\n]
      Request Method: GET
      Request URI: /onestar/video720.mpd
      Request Version: HTTP/1.1
    Host: localhost:3000\r\n
    Connection: keep-alive\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/83.0.4103.106 Safari/537.3
    Accept: */*\r\n
    Sec-Fetch-Site: same-origin\r\n
    Sec-Fetch-Mode: cors\r\n
    Sec-Fetch-Dest: empty\r\n
    Referer: http://localhost:3000/index.html\r\n
    Accept-Encoding: gzip, deflate, br\r\n
    Accept-Language: zh-TW,zh;q=0.9,en-US;q=0.8,en;q=0.7\r\n
    \r\n
    [Full request URI: http://localhost:3000/onestar/video720.mpd]
    [HTTP request 13/46]
    [Prev request in frame: 78]
    [Response in frame: 86]
```

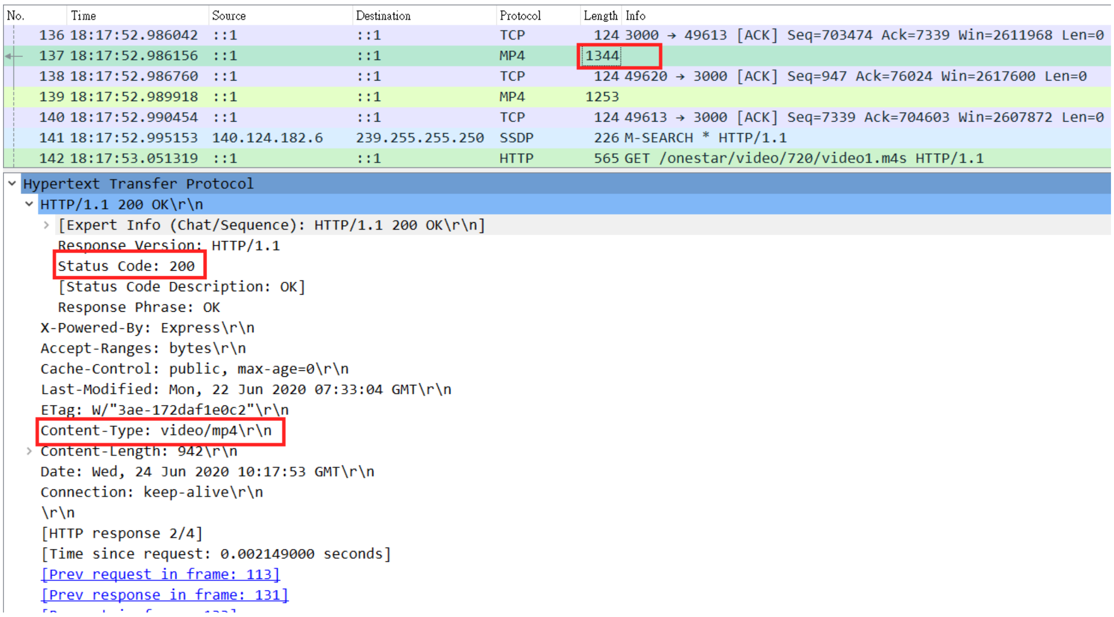
在開始串流播放影片之前，我們要會先載入 videoinit.mp4 與

audioinit.mp4，這兩個檔案在切割 segment 時產生的



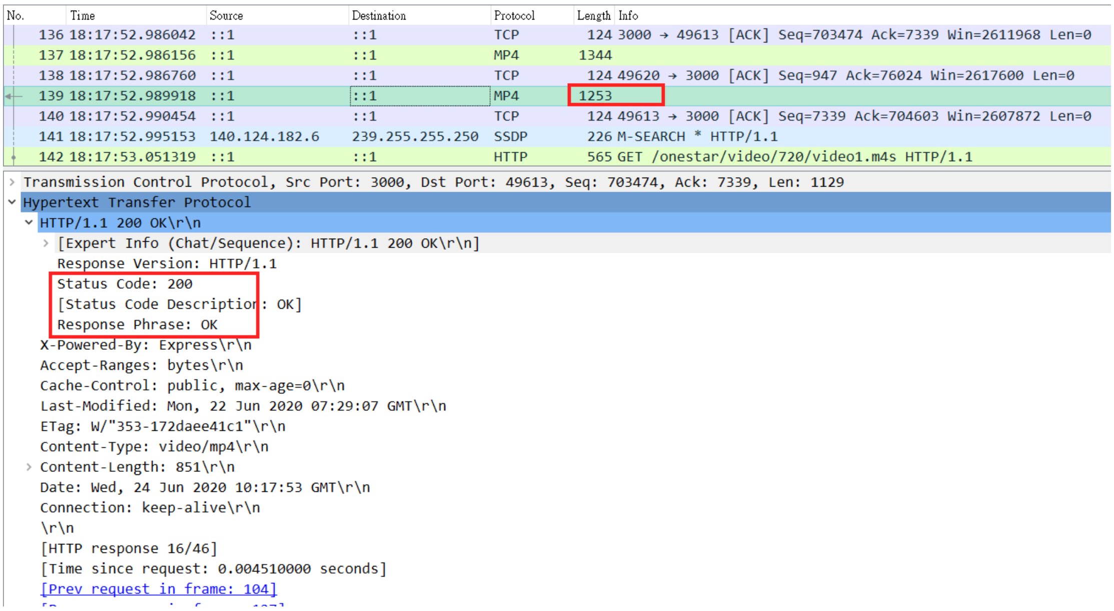
```
No.  Time             Source  Destination  Protocol  Length Info
 130 18:17:52.935327  ::1     ::1          TCP        124 49620 → 3000 [ACK] Seq=503 Ack=10232 Win=2608640 Len=0
 131 18:17:52.935814  ::1     ::1          HTTP      646… HTTP/1.1 200 OK  (application/font-woff2)
 132 18:17:52.937111  ::1     ::1          TCP        124 49620 → 3000 [ACK] Seq=503 Ack=74804 Win=2618880 Len=0
 133 18:17:52.984007  ::1     ::1          HTTP       568 GET /onestar/video/720/videoinit.mp4 HTTP/1.1
 134 18:17:52.984509  ::1     ::1          TCP        124 3000 → 49620 [ACK] Seq=74804 Ack=947 Win=2618368 Len=0
 135 18:17:52.985408  ::1     ::1          HTTP       564 GET /onestar/audio/audioinit.mp4 HTTP/1.1
 136 18:17:52.986042  ::1     ::1          TCP        124 3000 → 49613 [ACK] Seq=703474 Ack=7339 Win=2611968 Len=0
 137 18:17:52.986156  ::1     ::1          MP4       1344
 138 18:17:52.986760  ::1     ::1          TCP        124 49620 → 3000 [ACK] Seq=947 Ack=76024 Win=2617600 Len=0
 139 18:17:52.989918  ::1     ::1          MP4       1253
 140 18:17:52.990454  ::1     ::1          TCP        124 49613 → 3000 [ACK] Seq=7339 Ack=704603 Win=2607872 Len=0

> Null/Loopback
> Internet Protocol Version 6, Src: ::1, Dst: ::1
> Transmission Control Protocol, Src Port: 49620, Dst Port: 3000, Seq: 503, Ack: 74804, Len: 444
v Hypertext Transfer Protocol
  v GET /onestar/video/720/videoinit.mp4 HTTP/1.1\r\n
    > [Expert Info (Chat/Sequence): GET /onestar/video/720/videoinit.mp4 HTTP/1.1\r\n]
      Request Method: GET
      Request URI: /onestar/video/720/videoinit.mp4
      Request Version: HTTP/1.1
    Host: localhost:3000\r\n
    Connection: keep-alive\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/83.0.4103.106 Safari/537.3
    Accept: */*\r\n
    Sec-Fetch-Site: same-origin\r\n

0040  47 45 54 20 2f 6f 6e 65  73 74 61 72 2f 76 69 64   GET /one star/vid
0050  65 6f 2f 37 32 30 2f 76  69 64 65 6f 69 6e 69 74   eo/720/v ideoinit
0060  2e 6d 70 34 20 48 54 54  50 2f 31 2e 31 0d 0a 48   .mp4 HTT P/1.1··H
0070  6f 73 74 3a 20 6c 6f 63  61 6c 68 6f 73 74 3a 33   ost: loc alhost:3
0080  30 30 30 0d 0a 43 6f 6e  6e 65 63 74 69 6f 6e 3a   000··Con nection:
```

此為 videoinit 回傳的封包，可以看到 200 回傳成功



```
No.  Time             Source         Destination      Protocol  Length Info
 136 18:17:52.986042  ::1            ::1              TCP        124 3000 → 49613 [ACK] Seq=703474 Ack=7339 Win=2611968 Len=0
 137 18:17:52.986156  ::1            ::1              MP4       1344
 138 18:17:52.986760  ::1            ::1              TCP        124 49620 → 3000 [ACK] Seq=947 Ack=76024 Win=2617600 Len=0
 139 18:17:52.989918  ::1            ::1              MP4       1253
 140 18:17:52.990454  ::1            ::1              TCP        124 49613 → 3000 [ACK] Seq=7339 Ack=704603 Win=2607872 Len=0
 141 18:17:52.995153  140.124.182.6  239.255.255.250  SSDP       226 M-SEARCH * HTTP/1.1
 142 18:17:53.051319  ::1            ::1              HTTP       565 GET /onestar/video/720/video1.m4s HTTP/1.1

v Hypertext Transfer Protocol
  v HTTP/1.1 200 OK\r\n
    > [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
      Response Version: HTTP/1.1
      Status Code: 200
      [Status Code Description: OK]
      Response Phrase: OK
    X-Powered-By: Express\r\n
    Accept-Ranges: bytes\r\n
    Cache-Control: public, max-age=0\r\n
    Last-Modified: Mon, 22 Jun 2020 07:33:04 GMT\r\n
    ETag: W/"3ae-172daf1e0c2"\r\n
    Content-Type: video/mp4\r\n
  > Content-Length: 942\r\n
    Date: Wed, 24 Jun 2020 10:17:53 GMT\r\n
    Connection: keep-alive\r\n
    \r\n
    [HTTP response 2/4]
    [Time since request: 0.002149000 seconds]
    [Prev request in frame: 113]
    [Prev response in frame: 131]
```

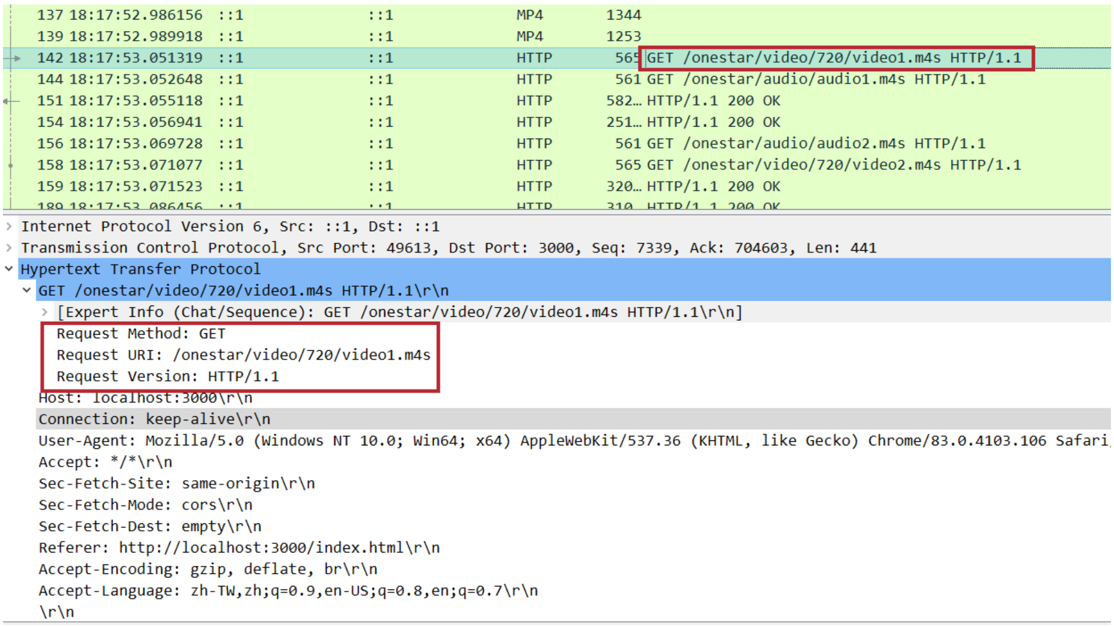
此為 audioinit 回傳的封包，可以看到 200 回傳成功



```
No.  Time             Source         Destination      Protocol  Length Info
 136 18:17:52.986042  ::1            ::1              TCP        124 3000 → 49613 [ACK] Seq=703474 Ack=7339 Win=2611968 Len=0
 137 18:17:52.986156  ::1            ::1              MP4       1344
 138 18:17:52.986760  ::1            ::1              TCP        124 49620 → 3000 [ACK] Seq=947 Ack=76024 Win=2617600 Len=0
 139 18:17:52.989918  ::1            ::1              MP4       1253
 140 18:17:52.990454  ::1            ::1              TCP        124 49613 → 3000 [ACK] Seq=7339 Ack=704603 Win=2607872 Len=0
 141 18:17:52.995153  140.124.182.6  239.255.255.250  SSDP       226 M-SEARCH * HTTP/1.1
 142 18:17:53.051319  ::1            ::1              HTTP       565 GET /onestar/video/720/video1.m4s HTTP/1.1
```

```
> Transmission Control Protocol, Src Port: 3000, Dst Port: 49613, Seq: 703474, Ack: 7339, Len: 1129
v Hypertext Transfer Protocol
  v HTTP/1.1 200 OK\r\n
    > [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
      Response Version: HTTP/1.1
      Status Code: 200
      [Status Code Description: OK]
      Response Phrase: OK
    X-Powered-By: Express\r\n
    Accept-Ranges: bytes\r\n
    Cache-Control: public, max-age=0\r\n
    Last-Modified: Mon, 22 Jun 2020 07:29:07 GMT\r\n
    ETag: W/"353-172daee41c1"\r\n
    Content-Type: video/mp4\r\n
  > Content-Length: 851\r\n
    Date: Wed, 24 Jun 2020 10:17:53 GMT\r\n
    Connection: keep-alive\r\n
    \r\n
    [HTTP response 16/46]
    [Time since request: 0.004510000 seconds]
    [Prev request in frame: 104]
```

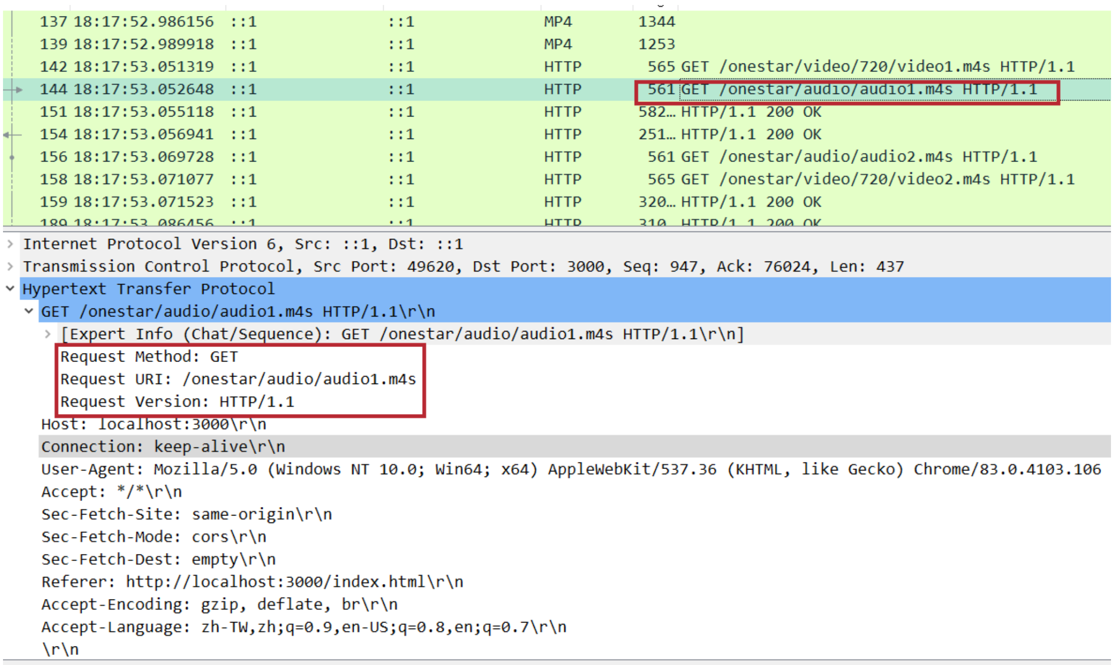
Client 使用 GET 方式請求 video1.m4s



```
 137 18:17:52.986156  ::1   ::1   MP4    1344
 139 18:17:52.989918  ::1   ::1   MP4    1253
 142 18:17:53.051319  ::1   ::1   HTTP    565 GET /onestar/video/720/video1.m4s HTTP/1.1
 144 18:17:53.052648  ::1   ::1   HTTP    561 GET /onestar/audio/audio1.m4s HTTP/1.1
 151 18:17:53.055118  ::1   ::1   HTTP   582… HTTP/1.1 200 OK
 154 18:17:53.056941  ::1   ::1   HTTP   251… HTTP/1.1 200 OK
 156 18:17:53.069728  ::1   ::1   HTTP    561 GET /onestar/audio/audio2.m4s HTTP/1.1
 158 18:17:53.071077  ::1   ::1   HTTP    565 GET /onestar/video/720/video2.m4s HTTP/1.1
 159 18:17:53.071523  ::1   ::1   HTTP   320… HTTP/1.1 200 OK
 180 18:17:53.086456  ::1   ::1   HTTP   310… HTTP/1.1 200 OK
```

```
> Internet Protocol Version 6, Src: ::1, Dst: ::1
> Transmission Control Protocol, Src Port: 49613, Dst Port: 3000, Seq: 7339, Ack: 704603, Len: 441
v Hypertext Transfer Protocol
  v GET /onestar/video/720/video1.m4s HTTP/1.1\r\n
    > [Expert Info (Chat/Sequence): GET /onestar/video/720/video1.m4s HTTP/1.1\r\n]
      Request Method: GET
      Request URI: /onestar/video/720/video1.m4s
      Request Version: HTTP/1.1
    Host: localhost:3000\r\n
    Connection: keep-alive\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/83.0.4103.106 Safari
    Accept: */*\r\n
    Sec-Fetch-Site: same-origin\r\n
    Sec-Fetch-Mode: cors\r\n
    Sec-Fetch-Dest: empty\r\n
    Referer: http://localhost:3000/index.html\r\n
    Accept-Encoding: gzip, deflate, br\r\n
    Accept-Language: zh-TW,zh;q=0.9,en-US;q=0.8,en;q=0.7\r\n
    \r\n
```

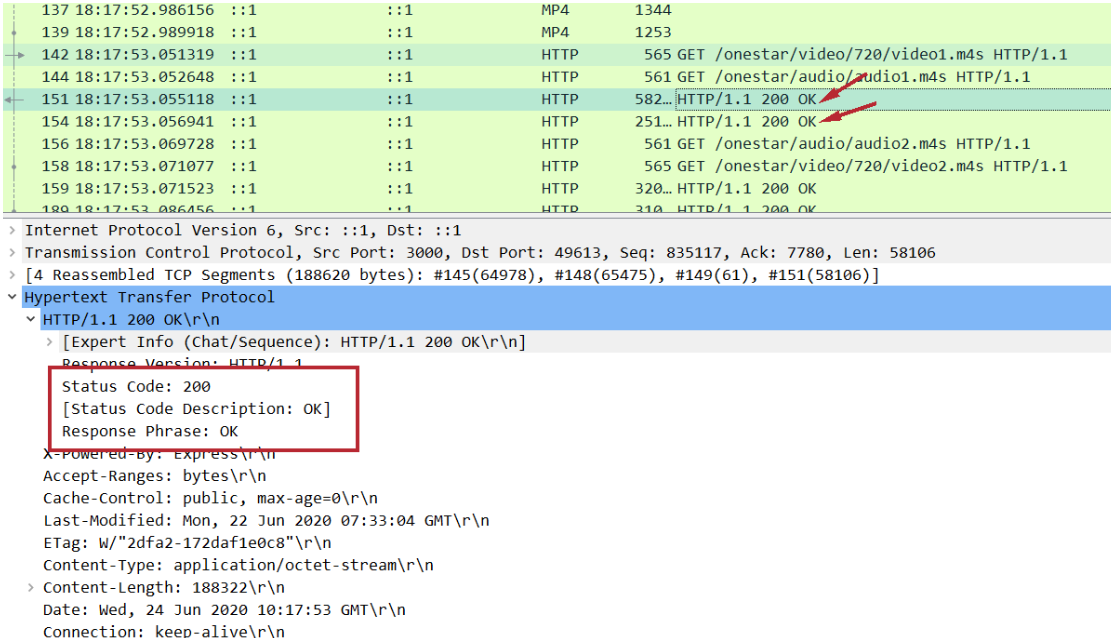
Client 使用 GET 方式請求 audio1.m4s，



```
137 18:17:52.986156  ::1             ::1             MP4       1344
139 18:17:52.989918  ::1             ::1             MP4       1253
142 18:17:53.051319  ::1             ::1             HTTP       565 GET /onestar/video/720/video1.m4s HTTP/1.1
144 18:17:53.052648  ::1             ::1             HTTP       561 GET /onestar/audio/audio1.m4s HTTP/1.1
151 18:17:53.055118  ::1             ::1             HTTP      582… HTTP/1.1 200 OK
154 18:17:53.056941  ::1             ::1             HTTP      251… HTTP/1.1 200 OK
156 18:17:53.069728  ::1             ::1             HTTP       561 GET /onestar/audio/audio2.m4s HTTP/1.1
158 18:17:53.071077  ::1             ::1             HTTP       565 GET /onestar/video/720/video2.m4s HTTP/1.1
159 18:17:53.071523  ::1             ::1             HTTP      320… HTTP/1.1 200 OK
180 18:17:53.086456  ::1             ::1             HTTP      310… HTTP/1.1 200 OK
```

```
> Internet Protocol Version 6, Src: ::1, Dst: ::1
> Transmission Control Protocol, Src Port: 49620, Dst Port: 3000, Seq: 947, Ack: 76024, Len: 437
v Hypertext Transfer Protocol
  v GET /onestar/audio/audio1.m4s HTTP/1.1\r\n
    > [Expert Info (Chat/Sequence): GET /onestar/audio/audio1.m4s HTTP/1.1\r\n]
      Request Method: GET
      Request URI: /onestar/audio/audio1.m4s
      Request Version: HTTP/1.1
    Host: localhost:3000\r\n
    Connection: keep-alive\r\n
    User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/83.0.4103.106
    Accept: */*\r\n
    Sec-Fetch-Site: same-origin\r\n
    Sec-Fetch-Mode: cors\r\n
    Sec-Fetch-Dest: empty\r\n
    Referer: http://localhost:3000/index.html\r\n
    Accept-Encoding: gzip, deflate, br\r\n
    Accept-Language: zh-TW,zh;q=0.9,en-US;q=0.8,en;q=0.7\r\n
    \r\n
```

接下來兩個封包為成功回傳之封包

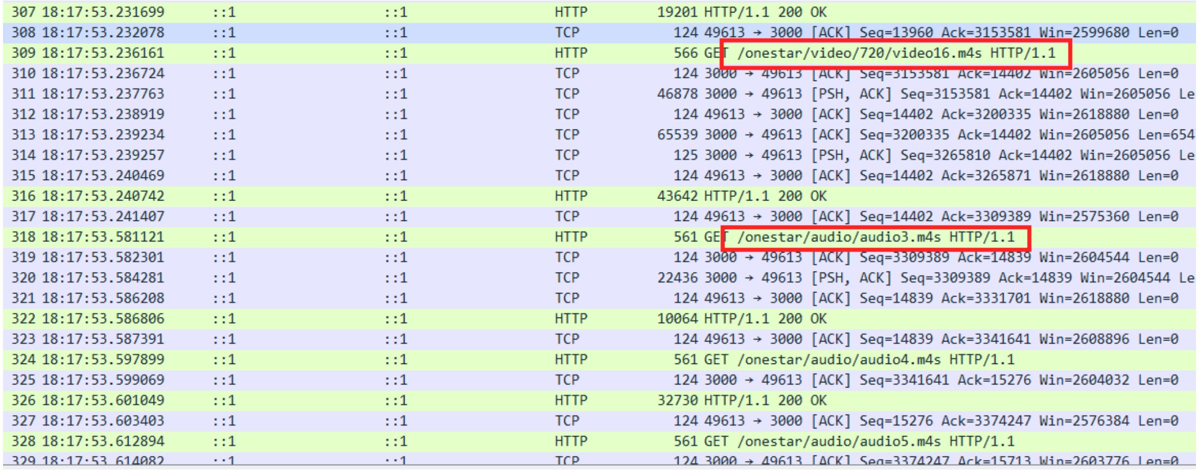


```
137 18:17:52.986156  ::1             ::1             MP4       1344
139 18:17:52.989918  ::1             ::1             MP4       1253
142 18:17:53.051319  ::1             ::1             HTTP       565 GET /onestar/video/720/video1.m4s HTTP/1.1
144 18:17:53.052648  ::1             ::1             HTTP       561 GET /onestar/audio/audio1.m4s HTTP/1.1
151 18:17:53.055118  ::1             ::1             HTTP      582… HTTP/1.1 200 OK
154 18:17:53.056941  ::1             ::1             HTTP      251… HTTP/1.1 200 OK
156 18:17:53.069728  ::1             ::1             HTTP       561 GET /onestar/audio/audio2.m4s HTTP/1.1
158 18:17:53.071077  ::1             ::1             HTTP       565 GET /onestar/video/720/video2.m4s HTTP/1.1
159 18:17:53.071523  ::1             ::1             HTTP      320… HTTP/1.1 200 OK
180 18:17:53.086456  ::1             ::1             HTTP      310… HTTP/1.1 200 OK
```

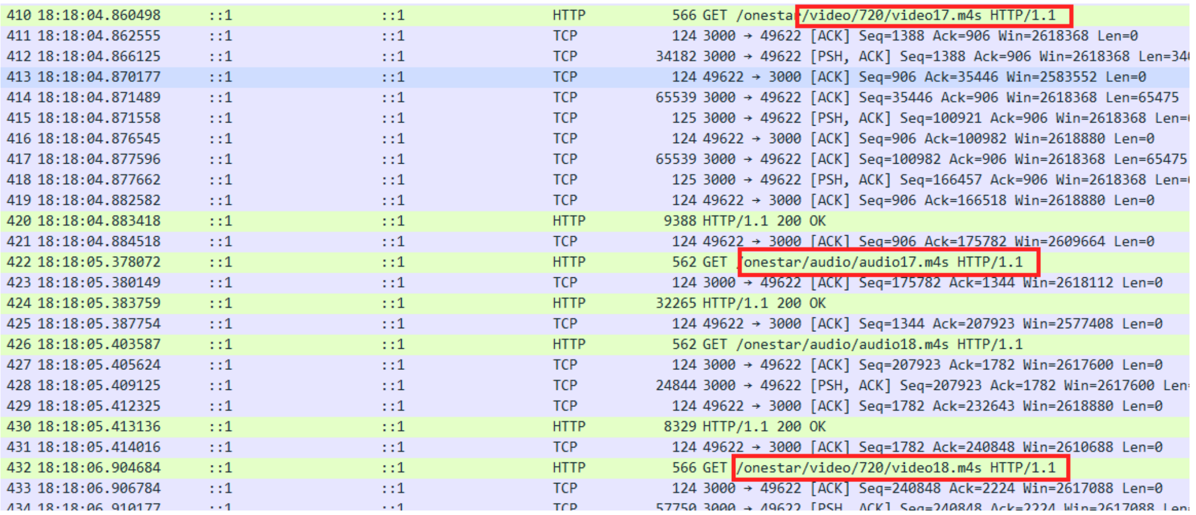
```
> Internet Protocol Version 6, Src: ::1, Dst: ::1
> Transmission Control Protocol, Src Port: 3000, Dst Port: 49613, Seq: 835117, Ack: 7780, Len: 58106
> [4 Reassembled TCP Segments (188620 bytes): #145(64978), #148(65475), #149(61), #151(58106)]
v Hypertext Transfer Protocol
  v HTTP/1.1 200 OK\r\n
    > [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
      Response Version: HTTP/1.1
      Status Code: 200
      [Status Code Description: OK]
      Response Phrase: OK
    X-Powered-By: Express\r\n
    Accept-Ranges: bytes\r\n
    Cache-Control: public, max-age=0\r\n
    Last-Modified: Mon, 22 Jun 2020 07:33:04 GMT\r\n
    ETag: W/"2dfa2-172daf1e0c8"\r\n
    Content-Type: application/octet-stream\r\n
  > Content-Length: 188322\r\n
    Date: Wed, 24 Jun 2020 10:17:53 GMT\r\n
    Connection: keep-alive\r\n
```

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



```
307 18:17:53.231699    ::1        ::1         HTTP     19201 HTTP/1.1 200 OK
308 18:17:53.232078    ::1        ::1         TCP        124 49613 → 3000 [ACK] Seq=13960 Ack=3153581 Win=2599680 Len=0
309 18:17:53.236161    ::1        ::1         HTTP       566 GET /onestar/video/720/video16.m4s HTTP/1.1
310 18:17:53.236724    ::1        ::1         TCP        124 3000 → 49613 [ACK] Seq=3153581 Ack=14402 Win=2605056 Len=0
311 18:17:53.237763    ::1        ::1         TCP      46878 3000 → 49613 [PSH, ACK] Seq=3153581 Ack=14402 Win=2605056 Le
312 18:17:53.238919    ::1        ::1         TCP        124 49613 → 3000 [ACK] Seq=14402 Ack=3200335 Win=2618880 Len=0
313 18:17:53.239234    ::1        ::1         TCP      65539 3000 → 49613 [ACK] Seq=3200335 Ack=14402 Win=2605056 Len=654
314 18:17:53.239257    ::1        ::1         TCP        125 3000 → 49613 [PSH, ACK] Seq=3265810 Ack=14402 Win=2605056 Le
315 18:17:53.240469    ::1        ::1         TCP        124 49613 → 3000 [ACK] Seq=14402 Ack=3265871 Win=2618880 Len=0
316 18:17:53.240742    ::1        ::1         HTTP     43642 HTTP/1.1 200 OK
317 18:17:53.241407    ::1        ::1         TCP        124 49613 → 3000 [ACK] Seq=14402 Ack=3309389 Win=2575360 Len=0
318 18:17:53.581121    ::1        ::1         HTTP       561 GET /onestar/audio/audio3.m4s HTTP/1.1
319 18:17:53.582301    ::1        ::1         TCP        124 3000 → 49613 [ACK] Seq=3309389 Ack=14839 Win=2604544 Len=0
320 18:17:53.584281    ::1        ::1         TCP      22436 3000 → 49613 [PSH, ACK] Seq=3309389 Ack=14839 Win=2604544 Le
321 18:17:53.586208    ::1        ::1         TCP        124 49613 → 3000 [ACK] Seq=14839 Ack=3331701 Win=2618880 Len=0
322 18:17:53.586806    ::1        ::1         HTTP     10064 HTTP/1.1 200 OK
323 18:17:53.587391    ::1        ::1         TCP        124 49613 → 3000 [ACK] Seq=14839 Ack=3341641 Win=2608896 Len=0
324 18:17:53.597899    ::1        ::1         HTTP       561 GET /onestar/audio/audio4.m4s HTTP/1.1
325 18:17:53.599069    ::1        ::1         TCP        124 3000 → 49613 [ACK] Seq=3341641 Ack=15276 Win=2604032 Len=0
326 18:17:53.601049    ::1        ::1         HTTP     32730 HTTP/1.1 200 OK
327 18:17:53.603403    ::1        ::1         TCP        124 49613 → 3000 [ACK] Seq=15276 Ack=3374247 Win=2576384 Len=0
328 18:17:53.612894    ::1        ::1         HTTP       561 GET /onestar/audio/audio5.m4s HTTP/1.1
329 18:17:53.614082    ::1        ::1         TCP        124 3000 → 49613 [ACK] Seq=3374247 Ack=15713 Win=2603776 Len=0
```

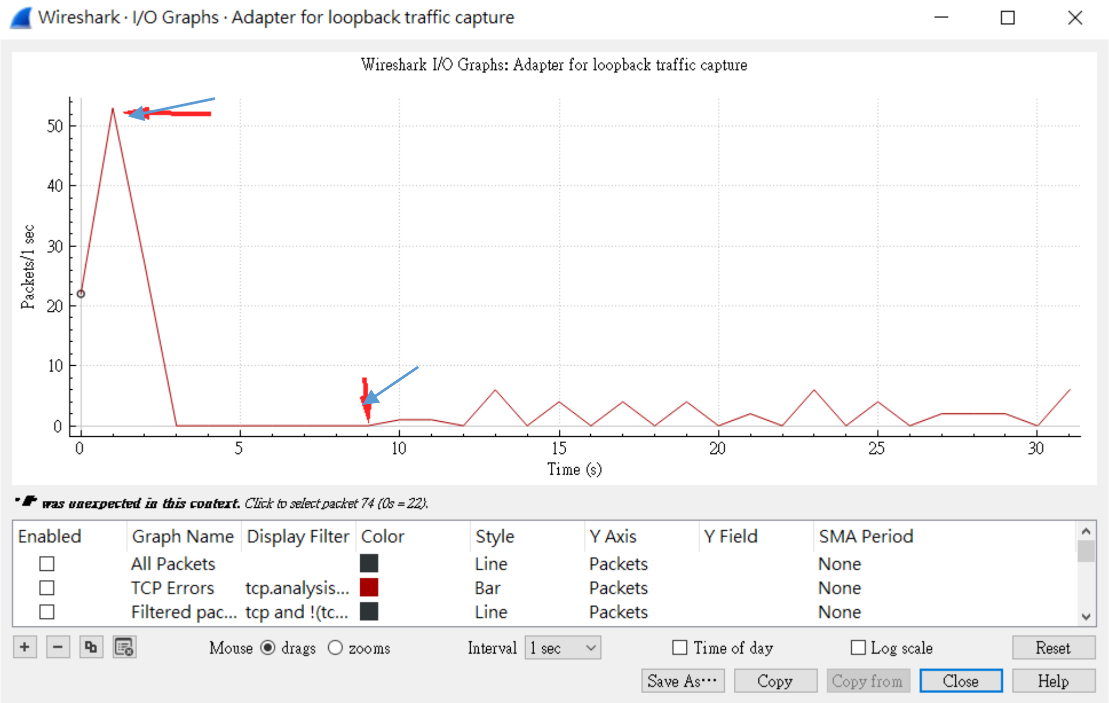


```
410 18:18:04.860498    ::1        ::1         HTTP       566 GET /onestar/video/720/video17.m4s HTTP/1.1
411 18:18:04.862555    ::1        ::1         TCP        124 3000 → 49622 [ACK] Seq=1388 Ack=906 Win=2618368 Len=0
412 18:18:04.866125    ::1        ::1         TCP      34182 3000 → 49622 [PSH, ACK] Seq=1388 Ack=906 Win=2618368 Len=34
413 18:18:04.870177    ::1        ::1         TCP        124 49622 → 3000 [ACK] Seq=906 Ack=35446 Win=2583552 Len=0
414 18:18:04.871489    ::1        ::1         TCP      65539 3000 → 49622 [ACK] Seq=35446 Ack=906 Win=2618368 Len=65475
415 18:18:04.871558    ::1        ::1         TCP        125 3000 → 49622 [PSH, ACK] Seq=100921 Ack=906 Win=2618368 Len=
416 18:18:04.876545    ::1        ::1         TCP        124 49622 → 3000 [ACK] Seq=906 Ack=100982 Win=2618880 Len=0
417 18:18:04.877596    ::1        ::1         TCP      65539 3000 → 49622 [ACK] Seq=100982 Ack=906 Win=2618368 Len=65475
418 18:18:04.877662    ::1        ::1         TCP        125 3000 → 49622 [PSH, ACK] Seq=166457 Ack=906 Win=2618368 Len=
419 18:18:04.882582    ::1        ::1         TCP        124 49622 → 3000 [ACK] Seq=906 Ack=166518 Win=2618880 Len=0
420 18:18:04.883418    ::1        ::1         HTTP      9388 HTTP/1.1 200 OK
421 18:18:04.884518    ::1        ::1         TCP        124 49622 → 3000 [ACK] Seq=906 Ack=175782 Win=2609664 Len=0
422 18:18:05.378072    ::1        ::1         HTTP       562 GET /onestar/audio/audio17.m4s HTTP/1.1
423 18:18:05.380149    ::1        ::1         TCP        124 3000 → 49622 [ACK] Seq=175782 Ack=1344 Win=2618112 Len=0
424 18:18:05.383759    ::1        ::1         HTTP     32265 HTTP/1.1 200 OK
425 18:18:05.387754    ::1        ::1         TCP        124 49622 → 3000 [ACK] Seq=1344 Ack=207923 Win=2577408 Len=0
426 18:18:05.403587    ::1        ::1         HTTP       562 GET /onestar/audio/audio18.m4s HTTP/1.1
427 18:18:05.405624    ::1        ::1         TCP        124 3000 → 49622 [ACK] Seq=207923 Ack=1782 Win=2617600 Len=0
428 18:18:05.409125    ::1        ::1         TCP      24844 3000 → 49622 [PSH, ACK] Seq=207923 Ack=1782 Win=2617600 Len
429 18:18:05.412325    ::1        ::1         TCP        124 49622 → 3000 [ACK] Seq=1782 Ack=232643 Win=2618880 Len=0
430 18:18:05.413136    ::1        ::1         HTTP      8329 HTTP/1.1 200 OK
431 18:18:05.414016    ::1        ::1         TCP        124 49622 → 3000 [ACK] Seq=1782 Ack=240848 Win=2610688 Len=0
432 18:18:06.904684    ::1        ::1         HTTP       566 GET /onestar/video/720/video18.m4s HTTP/1.1
433 18:18:06.906784    ::1        ::1         TCP        124 3000 → 49622 [ACK] Seq=240848 Ack=2224 Win=2617088 Len=0
434 18:18:06.910177    ::1        ::1         TCP      57750 3000 → 49622 [PSH, ACK] Seq=240848 Ack=2224 Win=2617088 Len
```

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



Wireshark · I/O Graphs · Adapter for loopback traffic capture
Wireshark I/O Graphs: Adapter for loopback traffic capture
Packets/1 sec
50
40
30
20
10
0
0
5
10
15
20
25
30
Time (s)

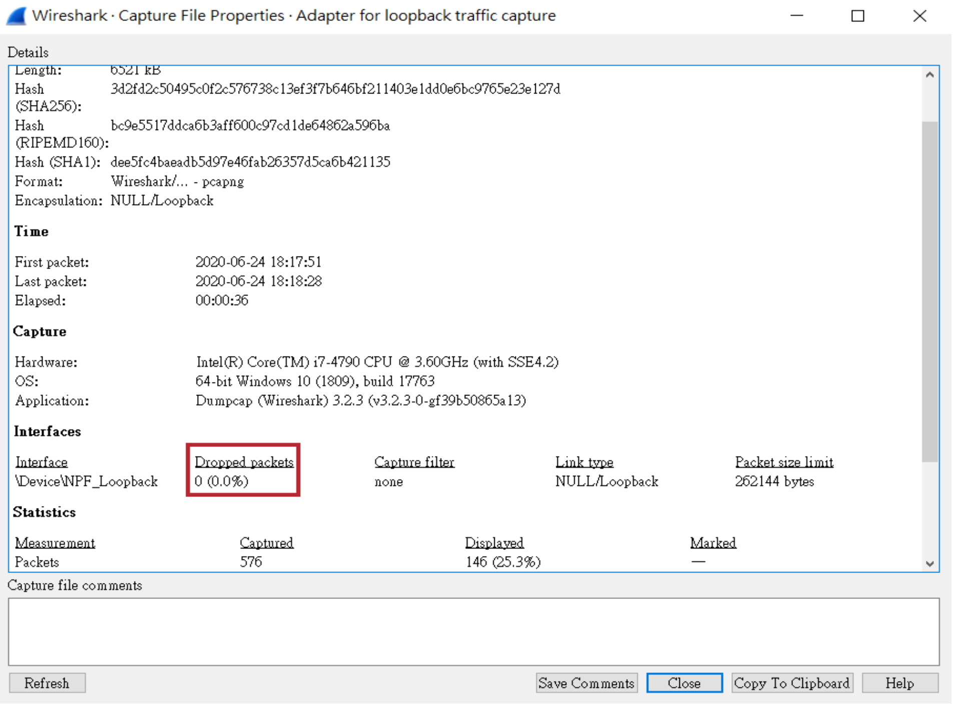
• was unexpected in this context. Click to select packet 74 (0s = 22).

Enabled	Graph Name	Display Filter	Color	Style	Y Axis	Y Field	SMA Period
□	All Packets		■	Line	Packets		None
□	TCP Errors	tcp.analysis...	🟥	Bar	Packets		None
□	Filtered pac...	tcp and !(tc...	■	Line	Packets		None

+ − Mouse ● drags ○ zooms Interval 1 sec □ Time of day □ Log scale

Reset
Save As…
Copy
Copy from
Close
Help

透過 capture file properties 顯示沒有封包丟失



Wireshark · Capture File Properties · Adapter for loopback traffic capture

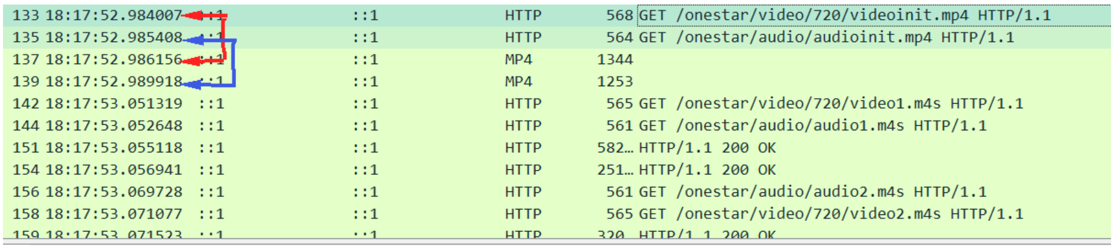
Details

```
Length:        6521 kB
Hash           3d2fd2c50495c0f2c576738c13ef3f7b646bf211403e1dd0e6bc9765e23e127d
(SHA256):
Hash           bc9e5517ddca6b3aff600c97cd1de64862a596ba
(RIPEMD160):
Hash (SHA1):   dee5fc4baeadb5d97e46fab26357d5ca6b421135
Format:        Wireshark/... - pcapng
Encapsulation: NULL/Loopback

Time

First packet:              2020-06-24 18:17:51
Last packet:               2020-06-24 18:18:28
Elapsed:                   00:00:36

Capture

Hardware:                  Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz (with SSE4.2)
OS:                        64-bit Windows 10 (1809), build 17763
Application:               Dumpcap (Wireshark) 3.2.3 (v3.2.3-0-gf39b50865a13)

Interfaces

Interface                Dropped packets   Capture filter   Link type       Packet size limit
\Device\NPF_Loopback     0 (0.0%)          none             NULL/Loopback   262144 bytes

Statistics

Measurement              Captured          Displayed        Marked
Packets                  576               146 (25.3%)      —
```

Capture file comments

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



```
133 18:17:52.984007 ::1        ::1        HTTP       568 GET /onestar/video/720/videoinit.mp4 HTTP/1.1
135 18:17:52.985408 ::1        ::1        HTTP       564 GET /onestar/audio/audioinit.mp4 HTTP/1.1
137 18:17:52.986156 ::1        ::1        MP4       1344
139 18:17:52.989918 ::1        ::1        MP4       1253
142 18:17:53.051319 ::1        ::1        HTTP       565 GET /onestar/video/720/video1.m4s HTTP/1.1
144 18:17:53.052648 ::1        ::1        HTTP       561 GET /onestar/audio/audio1.m4s HTTP/1.1
151 18:17:53.055118 ::1        ::1        HTTP      582… HTTP/1.1 200 OK
154 18:17:53.056941 ::1        ::1        HTTP      251… HTTP/1.1 200 OK
156 18:17:53.069728 ::1        ::1        HTTP       561 GET /onestar/audio/audio2.m4s HTTP/1.1
158 18:17:53.071077 ::1        ::1        HTTP       565 GET /onestar/video/720/video2.m4s HTTP/1.1
159 18:17:53.071523 ::1        ::1        HTTP      320  HTTP/1.1 200 OK
```

- **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。