# 自適式影像編碼 HLS 系統

黃連進　博士
淡江大學
Email:micro@mail.tku.edu.tw

呂建億
淡江大學
Email:700410193@s00.tku.edu.tw

**摘要 ---現今流行的網路串流，Apple HTTP Live Streaming，其缺點無法隨著客戶端與伺服端的頻寬變化，提供相對應的位元率，導致使用者觀看品質不流暢。所以此篇論文是希望能解決以上遇到問題，能利用動態編碼和頻寬管理，達到即時影像可依據頻寬變化，而會改變輸出的影像品質，先以求達到流暢播放為目的，而後求保持最大頻寬使用量下，提供最高的播放品質。**

***關鍵字* ---HTTP Live Streaming, 行動串流, 頻寬偵測, 可適性串流, WebVTT**

## . 一、前言

在室外使用手機上網頻寬變化沒有像室內接上網路線使用一樣穩定，會依據當地訊號的強弱、基地台同時使用人數的變化、當地提供的網路頻寬和移動中所產生的頻寬變化等，各種方面都會影響到播放線上串流地流暢度，在頻寬不穩定的網路環境下，如何讓使用者在畫質和流暢度取得平衡，得到最佳的觀看品質，是很重要可提出來討論的方向 [1] [2]。

傳統提供網路串流，一定需要在電腦安裝同樣的伺服端和客戶端的程式，伺服端壓縮固定位元率 (影像品質) 且相同的影像編碼格式，透過 TCP 或 UDP 或 RTP/RTCP 等傳輸協定，傳送影像和聲音給客戶端，客戶端再對接收到的視訊串流進行解碼且播放。由於使用固定位元率編碼且是預先編碼好，所以如果在頻寬足夠的情況下，可以輕易提供高畫質的影片，但是當現有頻寬無法播放此位元率影片時，就容易發生一直在緩衝影片的狀態。本篇論文在於希望能解決此問題，使用者不需先編碼影片，即可提供使用者即時觀看且高畫質的線上串流影片。

現在網路平台眾多，傳統觀看線上串流方法，使用者一定要安裝可提供播放此線上串流服務相對應的程式，方可播放且觀看影片，需要由使用者去思考播放程式的選擇，不是每個使用者都清楚知道該如何找到相對應的播放程式，應該提供使用者更簡便的方式收看線上串流，不應該由使用者去煩惱該安裝什麼播放程式，該選擇那種播放格式。如果客戶端只需要打開瀏覽器即可使用各種服務，可提供使用者更方便的服務，簡化使用者操作上的問題，即可提升使用者觀看體驗。

## . 二、ADAPTIVE TRANSCODING HTTP LIVE STREAMING SYSTEM

隨選視訊用途在於提供使用者可以觀看預錄影片，所以已經是事先處理好的影片，好處在於可隨時跳轉影片，觀看使用者自行想看的影片位置。不過壞處在於需要大量的儲存空間，存放所有伺服器提供給使用者的影片檔，而且當頻寬不足播放此影片時，容易產生播放停頓，影響使用者體驗。

即時線上串流在於提供給使用者觀看線上轉播等即時輸入訊號，好處是使用者可以即時觀看影像。所以即時線上串流最重要為減少影像延遲問題，就以伺服器角度來講，也可大量減少儲存空間。但伺服器需要提供大量的運算能力，用以應付大量的即時轉碼問題，且當多使用者連入，將大量考驗伺服器運算能力，且最重要為影像輸出位元率為固定值，如果頻寬不足時，完全無法播放即時串流。

以上兩種都產生相似的問題，容易受到頻寬的變化且容易受到多使用者連入，影響播放的流暢度。本篇論文在於解決以上傳統遇到的問題，提出一種合乎現在網路環境狀況的模型。

標準 HLS 適用於隨選視訊上，瀏覽器如需要支援動態調節影片位元率，必須在伺服器為同一個影片準備多個不同位元率的影片檔，且需要在 M3U8 檔案中，設定各種條件，當某個頻寬區間，瀏覽器該選擇下載且播放某個位元率的影片，用以達成調節影片畫質 [3]。如未在 M3U8 檔案中設定頻寬條件，會經由瀏覽器頻寬調整演算法去選擇適合畫質影片。由於其頻寬選擇演算法為 Greedy，所以會產生一種問題，當有新的連線進來時，由 Greedy 演算法 [4]，每個連線都會試圖取得最大畫質設定，但如果伺服器頻寬無法讓全部連線保持現有畫質設定時，每個連線的瀏覽器最後會選擇把畫質都設定為當時播放串流

提供的最低畫質，在下一個播放區間時 (10 秒)，會把所有連線都往上調高一個畫質，如果還是超過現有總頻寬，又會全部連線設定為最低畫質。一直反覆測試與設定，其產生問題為影響到全部使用者的觀看畫質。

本論文將提出 Adaptive Transcoding HTTP Live Streaming System(AHLS)，以下章節將分為說明以上問題的理論基礎和 AHLS 實作。

*(一) 頻寬偵測*

頻寬偵測在本論文為一重要課題，需要測定出伺服端和客戶端之間頻寬變化，本論文講求即時頻寬偵測且也無需要偵測出每個節點頻寬，只需要能即時偵測出伺服端和客戶端之間頻寬。

HTTP adaptive streaming 有兩個主要重點，提供畫質的最佳化和縮短從影像資源傳入至伺服器，而後傳輸至客戶端且播放，其中間的時間差，此為一種 end-to-end(e2e) delay [5]。標準 HLS 設計，由於視訊需要壓縮切割成時間固定的串流檔，再傳輸至客戶端，不同於 RTSP 架構，無法即時邊壓縮邊傳輸串流，所以一定會產生固定的 e2e delay。如何偵測頻寬，但不提高 e2e delay 數值，其為重點課題。本論文提出在之後實作中，於伺服端和客戶端中，加入一個虛擬中間層，用以替代伺服端與客戶端連接，當客戶端要求下載 HLS 串流檔時，改由虛擬中間層與客戶端連接且傳輸串流，即可測量出每個串流切片伺服端與客戶端傳輸時間。而每個串流檔檔案大小為已知條件，$\varepsilon_s$ 為檔案大小，$\tau_0$ 為串流檔傳輸時間，由公式 1即可計算出伺服端與客戶端現有頻寬 $\beta$。

$$\beta = \frac{\varepsilon_s}{\tau_0}, \tag{1}$$

*(二) 動態調節演算法*

標準 HLS 架構為把串流切割成無數個時間短的影片串流檔，當網路頻寬變化大時，將在下個時間切片切換畫質，用來調整串流位元率大小。本論文提出經由頻寬偵測得出現有頻寬後，在伺服端即時壓縮且切割輸出的串流檔，由於已知每個連線現在頻寬，即可把此數值，當作串流壓縮時的最大參數，確保下個時間點的串流檔位元率不會超過現在頻寬量。

*(三) 編碼壓縮模型*

經由記錄串流檔下載時間，即可測量出現在頻寬。當客戶端開始播放串流檔時，利用此空閒時間，伺服端利用測量出的頻寬，當成下個串流檔壓縮時的轉換參數。

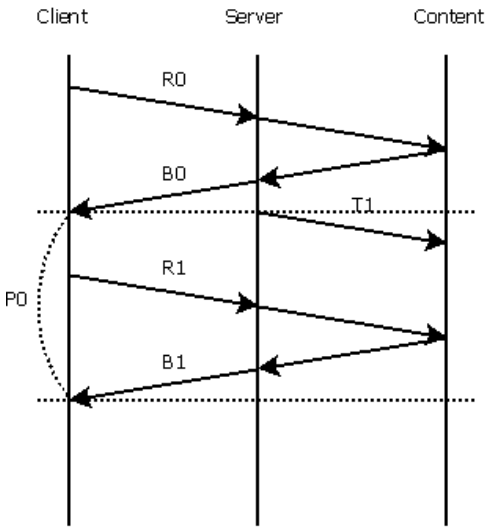


Fig. 1: Encoding model(first version)

TABLE I: The parameter of encoding model(first version)

| 參數 | 說明 |
|---|---|
| $R_0$ | Request first slice |
| $B_0$ | Response first slice and detect bandwidth |
| $T_1$ | Using bandwidth to request transcoding |
| $P_0$ | Play slice |
| $R_1$ | Request next slice |
| $B_1$ | Response next slice and detect bandwidth |

轉碼器 (Transcoder)，其可分為兩種功能，一為編碼 (Encode)，一為解碼 (Decode)。如果影片來源不是 HLS 原生支援的 MPEG2-TS 或 MPEG4-TS (H.264)，一定要先經由解碼器 (Decoder) 還原成未壓縮格式，然後編碼器 (Encode) 才可轉換為串流檔。本章節將提出兩種轉碼器的模型，HLS 設計是每十秒編碼成一個 TS 檔。客戶端播放影片時，傳輸下一個 TS 檔給客戶端。

影片傳輸給客戶端時，經由計算下載時間，用以測量出頻寬變化。由圖 1和表 I得知，每次下個串流檔開始壓縮的時間點，一定為每次傳輸完串流檔才能開始壓縮，瀏覽器在播放到影片的一半時間點 (5 秒)，將會請求下一個串流檔，每次轉碼下一個串流檔時間為影片時間的一半 (5 秒)，所以轉碼與傳輸串流檔至客戶端，總共只有 5 秒的處理時間。

## 三、實作

現今行動裝置興起，使用者常使用行動網路或無線網路上網，除了網路瀏覽外，利用行動網路觀看線上串流，也逐漸普及。但行動網路容易受很多因素影響傳輸速度，導致在線上串流傳輸不穩定，不定時的播放停格，影響使用者觀看感受，不合乎在外快速觀看影片的習慣。

在實作方面，在行動裝置觀看線上串流，大多需要安裝播放程式，用於連線到伺服器播放影片。如果使用者只需利用瀏覽器播放影片，是否更為方便？讓線上串流提供商，不用考慮各平台相容性問題和是否需要花時間撰寫 App，專心撰寫網頁和提供更多的線上串流服務。

本章節將會分別針對以下幾種方向，將會分別介紹實作中，所有會使用到的軟體，該如何架設應用於此論文實作且會遇到些什麼問題，該如何解決此問題，提出所有設計時遇到問題且該如何改善，用以合乎使用者需求的設計。

### *(一) 系統流程*

本論文不需修改網頁伺服器，最大好處為可使用任何一種現今常用的網頁伺服器。本篇論文使用在 Linux 下最常見的 Apache 網頁伺服器，且需要安裝 PHP 的外掛模組，網頁伺服器需要支援 PHP 網頁。實作程式是架設於 PHP 架構上，利用 PHP 撰寫跟客戶端溝通的網頁程式。本論文主要實作兩個 PHP 程式，其特性為執行於伺服端，所以對客戶端負載不會有影響，而且只有當每次客戶端需要使用時，才會執行 PHP 程式一次。第一個 PHP 程式用於動態產生 M3U8 檔；當客戶端要求下載串流檔時，第二個 PHP 程式用於傳輸串流檔給客戶端，而且當傳輸完串流檔，即可得知傳輸時間且計算出現在頻寬。把頻寬參數代入 FFmpeg，用以設定下個串流切片檔的串流品質。

程式運作流程為使用者經由程式播放介面 (如圖 2)，選擇出想觀看的影片後，客戶端會試著跟伺服端取得新的 M3U8 播放清單。當客戶端取得 M3U8 檔時，即可知道該如何下載串流檔，即會開始嘗試下載串流檔。客戶端下載串流檔後，即可開始播放影片，之後客戶端會一直重覆取得串流檔且播放串流檔。

原生 HLS 的 M3U8 播放清單會預先產生，HTML5 只需讀取載入 M3U8 播放清單即可 (如圖 3)，適用於隨選視訊環境，由於本論文需要達到即時線上串流功能性，需要支援 HLS 的線上串流格式，每個串流檔下載前，載入的播放清單皆不相同，所以實現每次客戶端請求 M3U8 檔時，可動態調整 M3U8 檔的序列號 (SEQUENE) 且儲存於資料庫中。

SEQUENE 用於記錄客戶端播放至第幾個串流檔，且令客戶端判斷是否有新的串流檔可下載且播放，而回應給各個連線客戶端的 M3U8 檔內容都是獨立不相同的，回傳的檔名也會由客戶端

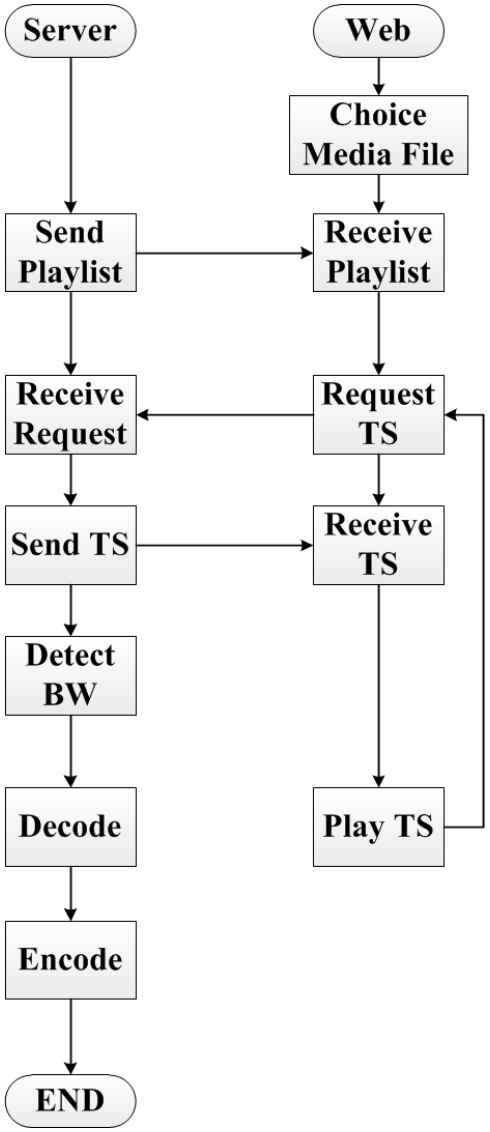


Fig. 2: 程式流程圖

```
<html>
<body>
<video src="http://ip/hls.m3u8" controls autoplay>
</video>
</body>
</html>
```

Fig. 3: Standard M3U8 playlists of HLS

```
<html>
<body>
<video src="http://genm3u8.php" controls autoplay>
</video>
</body>
</html>
```

Fig. 4: The code of dynamic generate M3U8

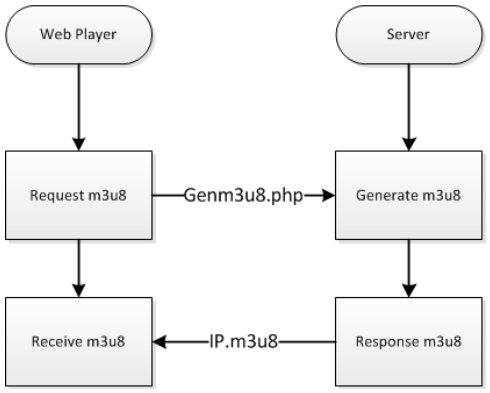


Fig. 5: The flowchart of dynamic generate M3U8

實體 IP 命名而成，經由 PHP 實現即時產生播放清單，當客戶端經由 HTML5 請求播放清單時，實際上為執行 PHP 程式，PHP 程式產生出播放清單且回傳給客戶端 (如圖 4)，圖 5為 PHP 程式如何動態產生 M3U8 檔的流程圖。

當客戶端取得 M3U8 播放清單後，會嘗試去播放播放清單內設定的串流檔，而程式同樣利用 PHP 當作和客戶端溝通的媒介。當客戶端要求取得可下載的串流檔時，其實是執行一次用以取得串流檔的 PHP，PHP 會先去跟伺服端下載可播放的串流檔，傳輸給客戶端，且測量出傳輸給客戶端的上傳時間，用以計算出伺服端和客戶端的頻寬變化。當取得頻寬變化量，經由公式 1計算出下次串流檔應該調整的位元率，伺服端即可開始動態編碼下一個串流檔，當作下一次被呼叫到時，傳輸給客戶端的暫存線上串流檔 (如圖 2)。

### *(二) 影片字幕*

視訊串流有時會觀賞些外語影片，因為語言隔閡影響觀看者對影片的理解，對使用者的觀看體驗將會降低。如果視訊串流能支援字幕功能，將可提供翻譯串流內容等資訊，也可提供串流提供者，加入一些需要提供給使用者的文字資訊，例如字幕廣告、跑馬燈字幕，用以提供各種文字資訊。實作方法將會使用 WebVTT 技術，此技術為 W3C 制定用來解決 HTML5 串流的字幕問題，本篇論文將會修改 W3C 所制定的規格，用以合乎本論文系統架構。

```
#EXT-X-MEDIA:TYPE=SUBTITLES,GROUP-ID="subs",NAME="
    Chinese",AUTOSELECT=YES,FORCED=YES,URI="gensubm3u8.
    “php
```

Fig. 6: The tag of HLS subtitles

```
printf("#EXTM3U\n");
printf("#EXT-X-MEDIA-SEQUENCE:%d\n", $seq);
printf("#EXT-X-TARGETDURATION:30\n");
printf("#EXTINF:30,\n");
printf("subtitles.webvtt\n");
```

Fig. 7: Dynamic generate subtitle

提供字幕功能，需要修改 M3U8 檔的輸出格式，所以需要修改產生 M3U8 檔的 PHP 程式，加入字幕的標籤 (如圖 6)，#EXT-X-MEDIA:TYPE=SUBTITLES 在 HLS 用以設定且提供字幕功能；GROUP-ID="subs" 用以和相對應的串流檔配對；NAME=" 日本人" 可設定顯示在播放器中，用以選擇字幕的說明文字；AUTOSELECT=YES 用以設定是否會自動選擇；FORCED=NO 用以設定預設是否會掛載此字幕；URI="subtitles.m3u8" 用以設定應該掛載那個字幕的 M3U8 檔。

字幕實作將會修改 URI 標籤中的內容，不掛載 WebVTT 字幕檔且修改為掛載 PHP 程式，此 PHP 程式將會自動產生與播放串流相對應的字幕 M3U8 檔 (如圖 7)，串流即可順利讀取到配對的字幕檔 (WebVTT 檔)，其程式將會讀取字幕時間軸長度，用以設定 DURATION 數值為字幕時間軸長度，動態設定 DURATION 值即可達成只需掛載單一字幕檔。標準 WebVTT 需要為每個串流影片檔皆提供一個字幕檔，其方法可讓串流提供者在串流播放中，動態更新顯示字幕，在本篇論文實作可不需達成此目的，且此方法會影響字幕掛載的複雜度。

### *(三) 偵測頻寬*

測量頻寬在本篇論文是個很重要的課題，測量伺服端和客戶端之間頻寬，如果能越準確，對動態編碼影片品質正確性將大為提升。有效利用現有網路頻寬，同時提供最高影片品質。本章節提出了三種測量頻寬的方法。

1) 利用 JavaScript 在網頁介面撰寫測速下載程式，於每次測試頻寬時，會從伺服端下載一

個測試檔案，用於測試現在頻寬。但會產生一個問題，在測試時會額外佔用現有頻寬，線上串流可使用頻寬會變少且影響播放的流暢度，而且無法準確測量出客戶端和伺服端真實頻寬變化。

2) 修改網頁伺服器的程式碼，於伺服器建立 Socket，傳輸影片串流時測量傳輸時間，用以計算出頻寬變化。但會產生一個問題，架設伺服器時，無法自由選擇適合的網頁伺服器，也難以修改全部市面上常見網頁伺服器的程式碼。
3) 最佳找出一個可解決以上問題的解法，利用 PHP 撰寫程式當作伺服端和客戶端的下載的中間媒介。當客戶端要求下載時，實際上是執行 PHP 程式，所以可輕易取得客戶端何時要求下載，可以很方便計算下載花費時間，以測量頻寬，可提供更多設計需求。

*(四)HLS 伺服器*

HLS 當初設計時，即講求可快速架設出一個線上串流的系統。當安裝好網頁伺服器後，只需調整伺服端的 MIME 設定，用以支援 HLS 所需要支援的檔案副檔名。以架設平台為 Ubuntu 且安裝 Apache 網頁伺服器為例子，需要修改"/etc/apache2/mods-enabled/mime.conf"，增加幾筆支援的檔案格式。

*(五) 資料庫設計*

本程式還需要安裝資料庫軟體，用以儲存在程式運作中，所需要的各種數據，用以提供給程式完成全部功能的參考資料。本程式將利用 MySQL 架設資料庫環境，資料庫儲存可分為兩個儲存表格，一為 media_list，另一為 session。

media_list 表格用以儲存伺服端可播放的所有檔案清單，表格內記錄了檔案名稱 (Name)，時間長度 (Duration) 和影片原始位元率 (BitRate)，如表 II。當使用者開啟網頁介面時，會讀取資料庫取得可播放的影片列表，使用者可以隨意選擇想要播放的影片。而且介面也提供搜尋影片功能，當影片增加越來越多時，檔案清單會過長，無法在有限的網頁介面顯示出來，使用者可利用搜尋功能，找出想播放的影片。

Session 表格用以儲存各個使用者現在播放的狀態。表格內利用 Address 儲存了每個使用者的 IP 位址，以用區別每個使用者的儲存資料；Bandwidth 儲存每位使用者現在所測量出的頻寬變化，用以提供給下次編碼時，轉碼的參數設定；File 儲存每位使用者現在選擇播放的影片檔名，用以當作編碼時，編碼程式輸入檔名的參數設定；Sequence 儲存每位使用者現在播放進度，可和 File 合併使用，當使用者不小心關閉視窗時，可利用此參數回復狀態，知道使用者上次播放檔案進度。

TABLE II: The table of media_list

| item | type | description |
|---|---|---|
| Name | CHAR | Record video file name |
| Duration | BIGINT | Record video length(second) |
| BitRate | BIGINT | Record video bitrate(kbps) |

*(六) 介面設計*

本章節利用 jQuery 撰寫 Web UI，其為市面上其中一種 JavaScript 函式庫，用於快速方便的設計出所需得功能，然後經由資料庫取得可播放影片的各種資料，顯示於播放清單，例如影片名稱、影片原始位元率和影片時間長度。希望提供 Web UI 給使用者操作網頁型播放程式，跟資料庫查找伺服器內可播放影片，讓使用者可選擇喜歡影片播放，提供在行動裝置和電腦上相同的操作體驗。

## 四、功能分析與效能比較

表 III針對字幕、播放平台和瀏覽器需要額外安裝外掛程式進行比較。目前經由瀏覽器可順利掛載 WebVTT 字幕的平台方面，於 W3C 規範上，目前只支援 HTTP 漸近式下載協定，所以 mp4 和 WebM 可順利經由 HTML5 掛載字幕。而 HLS 和 AHLS 在字幕支援方面，由於 HLS 和 AHLS 於 iOS 平台上支援 WebVTT 字幕功能，所以字幕可於 iOS 平台上順利掛載字幕。而 HLS 只可支援於隨選視訊上使用字幕，AHLS 可順利掛載於即時串流。外掛程式方面，目前自適式串流技術只有 HLS 和 AHLS 可完全支援 HTML5 的瀏覽器中串流播放，其他漸近式串流皆需要額外安裝外掛程式，不然無法於瀏覽器播放。而漸近式下載技術中，mp4 和 WebM 都可經由 HTML5 播放串流。

表 IV和表 V將針對 HLS 和 AHLS 效能比較分析。HLS 設計在播放串流中，會開始下載下一個串流切片檔案，如果串流切片時間設定為 10 秒，瀏覽器將會在播放串流切片於 5 秒時，開始下載下一個串流切片檔案。由表 IV分析顯示，由於 HLS 其設計位元率是固定，所以串流切片檔案將會固定大小，當頻寬越來越小時，所需傳輸時間越來越長，當頻寬 0.5Mbps 時，已經無

TABLE III: The different of streaming technology

| Streaming | Subtitle | Environment | Plug-in |
|---|---|---|---|
| Adobe HDS | No Supported | IE, Safari, Chrome | Flash |
| MS Smooth | No Supported | IE, Safari, Chrome | SliverLight |
| DASH | No Supported | IE, Safari, Chrome | DASH VLC plugin |
| Apple HLS | Only Support VOD | iOS, Safari, Android | No need |
| mp4 | Supported | All devices | No need |
| WebM | Supported | Chrome, Android 2.3.3 and up | No need |
| AHLS | Supported | iOS, Mac Safari, Android 4.0 and up | No need |

TABLE IV: HLS performance

| Bandwidth(Mbps) | 2 | 1 | 0.5 | 0.25 |
|---|---|---|---|---|
| Slice file size(Mbytes) | 4.1 | 4.1 | 4.1 | 4.1 |
| Transmit time(秒) | 2.05 | 4.1 | 8.2 | 16.4 |

TABLE V: AHLS performance

| Bandwidth(Mbps) | 2 | 1 | 0.5 | 0.25 |
|---|---|---|---|---|
| Slice file size(Mbytes) | 5.688 | 3.535 | 1.247 | 0.803 |
| Transmit time(秒) | 2.844 | 3.535 | 2.493 | 3.229 |

法在播放串流時，於有限時間 5 秒內傳輸完成，將會導致視訊串流開始停頓，視訊串流無法流暢播放，將會影響使用者觀看體驗不佳。由表 V分析顯示，AHLS 設計會依照當前頻寬變化，自適式轉碼下一個串流切片檔的位元率，就算頻寬越來越小，AHLS 會調整位元率，使得串流切片檔案縮小，用以提升傳輸時間。在測試環境條件下，皆可於有限時間內下載完串流切片檔，視訊串流將可流暢播放，用以提升使用者觀看體驗，串流播放流暢而影像品質有些損失是可以接受。AHLS 設計是可以適應於各種頻寬變化，使用者不必擔心播放環境是否足夠播放串流。

## . 五、結論與未來研究

本篇論文目標為希望能依據現有頻寬，動態調整影片位元率，且在可限制的影像編碼參數，提供最佳畫質的線上串流影像，經過前面章節討論後，可得到以下結論。

1) 經由測試結果已得知，影片位元率確實可依據客戶端與伺服端之間頻寬，動態調整編碼時位元率，可讓使用者能順利在各種不同網路環境 (有線網路、無線網路和行動網路)，順利播放線上串流影片 。
2) 實作出得網頁操作介面，確實可適用於 iPhone、iPad、Android 4.0 手機和 Mac 電腦等，都可提供相同的操作介面，且可流暢播放，成功達成去除客戶端程式，且去除使用 Adobe Flash 等外掛程式於瀏覽器播放。
3) 更可以擴充應用範圍，可利用平板連接電視或投影機，即可達到隨身電影院的應用功能。

在取得現在頻寬後，如何有效調整位元率為一種很重要的課題，利用現有公式可處理大部份頻寬變化情況，希望在未來可提出預測頻寬變化機制，當在取得頻寬之前，就可知道自適式編碼應該使用位元率來調整影片，除了可不必受制於等待頻寬偵測的機制，可更接近和影片播放時間同步的即時編碼，且可以更有效的調整位元率，更加不容易受到頻寬變化量大的網路環境所影響。

由於目前只支援 HLS 播放，使得可播放平台受限制，程式當初設計有考慮未來跨平台的通用性，希望未來可隨時增加支援的平台，而可達到完全跨平台的線上串流系統。

## 參考文獻


[1] C. Y. Hung, ``Live broadcast and vod system based on http live streaming," 清華大學資訊工程學系學位論文, 2012.

[2] S. K. Hsieh, ``Research and design of video adaption algorithm based on http live streaming," 東華大學, 2012.

[3] Apple http live streaming. [Online]. Available: https://developer.apple.com/resources/http-streaming/

[4] K. J. Ma and R. Bartos, ``Http live streaming bandwidth management using intelligent segment selection," in Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE, 2011, pp. 1--5.

[5] T. Kupka, P. Halvorsen, and C. Griwodz, ``An evaluation of live adaptive http segment streaming request strategies," in Local Computer Networks (LCN), 2011 IEEE 36th Conference on, 2011, pp. 604--612.