

第 19 屆行動計算研討會

A Cloud-Assisted P2P Streaming Architecture for Mobile Users

Tz-Heng Hsu(許子衡)

南台科技大學資訊工程系

hsuth@mail.stust.edu.tw

Lin-You Wu(吳麟佑)

南台科技大學資訊工程系

ma0g0106@stust.edu.tw

摘要

雲端運算結合同儕網路成為即時視訊串流與多媒體服務品質的重要研究議題。由於不穩定的無線網路訊號傳輸及同儕節點任意加入與離開，使用者常會遭受長時間的等待和突發性的下載中斷，進而影響觀賞影片的意願。本論文提出一結合雲端運算技術與同儕網路技術的視訊串流傳輸架構，並藉由雲端技術強大的運算能力，加強視訊串流傳輸的彈性，改善網路傳輸的服務品質。

當多媒體提供者上傳視訊內容到雲端儲存空間，雲端服務控制器通知 Master/Reducer 進行視訊編碼，編碼後的視訊資料再由主控節點進行合併傳回雲端儲存空間。當雲端串流伺服器收到不同同儕運算網路子樹根節點(Roots)的視訊傳輸要求時，則會將不同解析度視訊串流依照分層方式，分為視訊基本層(Base Layer, BL)和視訊加強層(Enhancement Layer, EL)傳輸給各個不同同儕子樹根節點(Roots)，同儕運算網路根節點(Roots)再將視訊串流傳輸給其他同儕節點(Peers)，提供高穩定性、高擴展性、及低成本的串流傳輸架構。

關鍵詞：雲端運算、同儕網路、視訊串流

Abstract

To satisfy the busy demand of video transmission of mobile users, it requires a lot of bandwidth and computing resources for video streaming and transcoding. In order to reduce the transmission delay, a cloud-assisted P2P streaming system is proposed. In conjunction with high-power cloud computing resources and low-cost P2P network technologies, the proposed architecture can provide smooth video streaming services to mobile users.

Keywords: Cloud Computing, P2P Network, Video Streaming

一、前言

隨著網際網路和多媒體技術的日益普及，第三代/第四代行動通訊技術(3G/4G)的技術成熟，使用行動裝置觀看視訊串流的使用者日漸增加。個別行動裝置有不同的運算能力、能觀看視訊串流的解析度也不同，使用者的移動也關係著視訊串流傳輸的穩定度，這些因素都會影響使用者觀看視訊的服務品質(Quality of Service)。

為了滿足使用者的視訊觀看需求，必須針對視訊串流進行視訊編碼壓縮，來減少資料量。可擴展視訊編碼(Scalable Video Coding, SVC) [1] [5] [9]，是目前研究視訊串流壓縮技術的重要技術。可擴展視訊編碼是傳統 H.264/MPEG-4 AVC(Advanced Video Coding)編碼的改良，具有三種可擴展的特性：時間性(Temporal)、空間性(Spatial)和品質擴展性(Quality Scalability)。由一個基礎層(Base Layer)及數個加強層(Enhancement Layers)所形成的編碼特性，可依照使用者本身的頻寬條件，選擇適合的解析度來提供視訊。

目前視訊串流傳輸架構以同儕式網路(P2P Network)傳輸為主流，透過共享機制，進行使用者與使用者之間的視訊串流傳輸。由於同儕節點經常性的加入、離開，使得使用者常遭受長時間的等待和突發性的下載中斷，進而影響行動用戶觀賞影片的意願。雲端運算(Cloud Computing)提供了一個異質行動網路下不同裝置的視訊串流解決方案，強大的運算能力、大量的儲存空間與網路頻寬資源，似乎解決了網路視訊串流服務長期以來面對的問題。然而，相較於同儕運算網路節點間的無償資源共享，伴隨著雲端運算強大運算資源而來的是高昂的租用費用。雲端運算廠商主要依用戶使用的處理器時間、儲存空間與網路流量計價。當大量使用者在異質行動網路下觀看視訊串流時，所累積而來的費用極高，不利網路視訊串流應用的發展。因此，如何提供一個結合雲端運算強大資源優勢，但是低成本的支援多媒體服務品質(QoS)視訊串流解決方案，成為

異質行動網路環境下視訊串流傳輸的主要問題。

本論文提出的構想是提出一基於雲端輔助之同儕行動視訊串流傳輸架構，並藉由雲端技術強大的運算能力，來協助可擴展視訊編碼，有效縮短可擴展視訊編碼的時間，加強視訊串流傳輸的穩定性，來改善網路傳輸的服務品質(Quality of Service, QoS)，包括：影片播放的連續性(Playback Continuity)和播放延遲(Playback Delay)的問題。當多媒體提供者上傳視訊內容到雲端儲存空間，雲端服務控制器通知 Master/Reducer 進行視訊編碼，編碼後的視訊資料再由主控節點進行合併傳回雲端儲存空間。當雲端串流伺服器收到不同同儕運算網路子樹根節點(Roots)的視訊傳輸要求時，則會將不同解析度視訊串流依照分層方式，分為視訊基本層(Base Layer, BL)和視訊加強層(Enhancement Layer, EL)傳輸給各個不同同儕子樹根節點(Roots)，同儕運算網路根節點(Roots)再將視訊串流傳輸給其他同儕節點(Peers)，同儕式節點也會依據網路狀態、同儕節點之間的傳輸能力，所觀看的視訊串流解析度來動態加入不同的父親節點，形成多個視訊串流傳遞樹(Multiple Delivery Trees)，傳遞資料給其他的同儕節點，當在所設定的最後下載時間門檻值內，使用者遲遲無法從鄰近的同儕節點或同儕根節點下載所欠缺的視訊串流時，則直接向雲端補足所需要的視訊串流，提高視訊串流傳輸的穩定性及提供低成本的串流傳輸架構。

二、相關研究探討

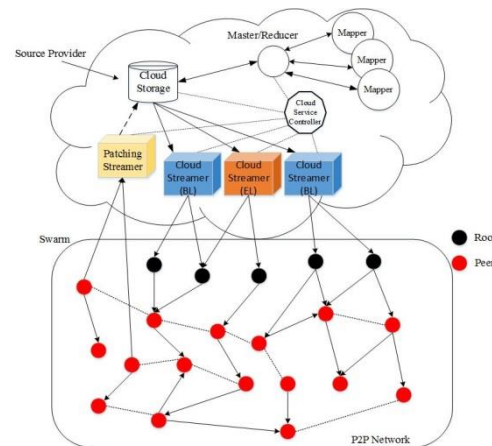
論文[11]中，作者 Zixia Huang 提出 CloudStream 的方法，多媒體提供者，如 YouTube 和 Hulu，在傳輸視訊時都是採用漸進式(Progressive)下載的方式傳送給使用者。但這樣的方式在變異量很大的動態網路中會導致突發性的視訊凍結(Freezes)，而作者利用 H.264/SVC(Scalable Video Coding)的編碼方式，基於使用者和多媒體伺服器之間使用視訊代理伺服器(Proxy)，來適應網路條件的改變。

論文[10]中，作者 Yuanyuan Xu 提出 CoopNet 使用 MDC 編碼的同儕式視訊串流系統，由於同儕節點的加入、離開以及同儕節點頻寬的變異，都會造成同儕式網路的不穩定。作者針對現存的同儕式網路系統採用了多重描述編碼(Multiple Description Coding, MDC)的編碼方式，使用 MDC 的方式讓同儕節點傳輸視訊時提供強大的錯誤恢復能力並支援異質性網路。比較傳統的階層式編碼(Layered Coded)，使用 MDC 編碼具有更高的穩定性。

在某些特殊的事件裡，Flash Crowds 很容易淹沒(Overwhelm)或是使視訊服務崩潰(Crash)導致可擴展性問題的出現，有許多 Cost-Effective 的方式來解決可擴展性的問題，像是利用 P2P 網路運算技術，可有效的利用同儕節點的上傳頻寬來降低伺服器的負擔，利用同儕式網路下載影片部分的內容之後，使用者一邊觀看影片，一邊下載影片剩餘的片段。

論文[2]中，因為多媒體提供者與同儕式網路中可用上傳頻寬的瓶頸，會影響到視訊串流服務品質(QoS)。作者 Amir H. Payberah 提出 CLIVE 架構，是一個雲端輔助系統使用同儕式視訊串流架構，來解決動態行為網路(Dynamic Behavior Network)，提供良好的服務品質(QoS)。作者考慮到 helpers(雲端伺服器)的能力、租用的成本以及如何選擇正確的 helpers 的類型，並規範 helpers 的數量與動態行為的使用者，假如目前有太少的 helpers，就會無法達到 QoS 的最低期望的標準。利用 CLIVE 中所定義的 CLIVE Manager(CM)，透過 gossip-based 聚合協定與其他的同儕節點找出目前的 swarm 狀態。CLIVE 也包含被動的 helper，它的功能就是提供同儕節點的最後請求，也就是說在最後規定的時間內同儕節點無法提供視訊資料時，則直接向被動的 helper 下載資料。

三、模型架構



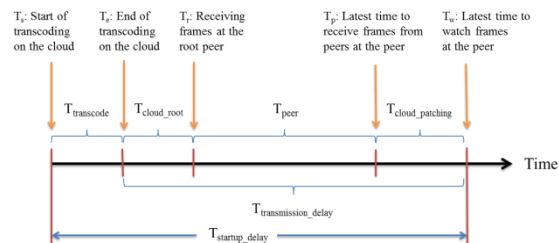
圖一 系統架構圖

圖一所示為本系統之架構，結合雲端運算技術與網狀覆蓋網路(Mesh Overlays Network)為基礎的同儕樹狀網路。首先，多媒體提供者(Source Provider)，上傳視訊內容到雲端儲存空間(Cloud Storage)，接著，雲端服務控制器(Cloud Service Controller)通知主控節點(Master/Reducer)將影片進行不同大小的分段處理，分別傳送給不同的 Mapper 伺服器，進

行可擴展視訊編碼(SVC-coding)，採用起始播放延遲(Startup Delay)優先的即時排程編碼演算法進行視訊編碼，編碼後的視訊資料再由主控節點進行合併，儲存回雲端儲存空間。雲端服務控制器接著建立雲端串流伺服器(Cloud Streamer)準備進行視訊串流傳輸，雲端串流伺服器依所傳輸的視訊資料層分為視訊基礎層雲端串流伺服器(Base Layer Cloud Streamer)及視訊加強層雲端串流伺服器(Enhancement Layer Cloud Streamer)。

當雲端串流伺服器收到不同同儕運算網路子樹根節點(Roots)的視訊傳輸要求時，則會將不同解析度視訊串流依照分層方式，分為視訊基本層(Base Layer, BL)和視訊加強層(Enhancement Layer, EL)傳輸給各個不同同儕子樹根節點(Roots)，同儕運算網路根節點再將視訊串流傳輸給其他同儕節點(Peers)。網路子樹根節點是由同儕網路中，運算資源(例如網路頻寬、運算能力)最高的同儕節點擔任。

其他同儕節點需要下載不同解析度的串流視訊時，會優先向各個同儕子樹根節點要求建立連線，下載所需視訊串流。雲端串流伺服器的數量是由雲端服務控制器依不同解析度視訊連線需求來動態產生，以降低網路傳輸延遲成本及運算成本。



圖二 即時視訊串流傳輸架構時序圖

圖二展示本系統的即時視訊串流傳輸架構時序圖，我們定義了 $T_{startup_delay}$ 變數代表同儕節點第一次接收串流資料時所需要等待的時間； $T_{transmission_delay}$ 變數代表編碼後可調性視訊編碼資料由雲端串流伺服器經由同儕運算網路傳送到末端同儕網路節點所需要的傳輸時間。 $T_{transcode}$ 變數代表原始視訊檔案進行可調性視訊所消耗的時間； T_s 變數代表雲端平台開始進行起始視訊片段(Startup Segment)編碼的時間； T_e 變數代表雲端平台結束起始視訊片段編碼的時間。 T_{cloud_root} 變數代表雲端串流伺服器將雲端視訊串流傳輸給同儕運算網路子樹根節點的傳輸時間； T_r 變數代表同儕運算網路子樹根節點收到視訊串流封包的時間。

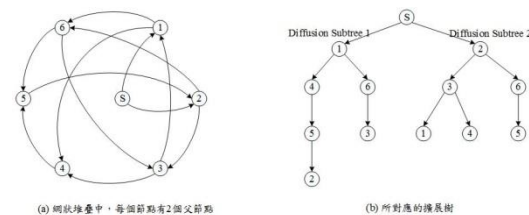
T_{peer} 變數代表同儕運算網路節點跟節點之間的視訊串流封包傳輸時間； T_p 變數代表同

儕運算網路節點收到其他同儕節點最後所傳輸的視訊串流封包的時間。 $T_{cloud_patching}$ 變數代表雲端串流輔助伺服器(Patching Streamer)補足視訊串流封包給同儕運算網路節點所需要的傳輸時間； T_w 變數代表同儕運算網路節點開始觀看視訊的最後時間。

3.1 模型流程

分析架構時序圖，從雲端平台開始進行起始視訊片段的編碼時間 T_s 到雲端平台結束起始視訊片段編碼時間 T_e 。原始視訊影片進行可調性視訊編碼的時間 $T_{transcode}$ ，假設 Cloud Streamer 到 Root peer 頻寬為 BW_{cloud_root} ，假設每個 Frame 的平均大小 F_{size} ，對應到 T_{cloud_root} 時間為 F_{size}/BW_{cloud_root} ，當行動節點要求視訊串流時，資料在傳輸的過程中可能會有遺失，在 T_{peer} 定義時間內，相鄰不同資料的同儕節點，可以用推拉方式(Push-pull Approach)，向其他同儕節點要求提供所缺少的視訊資料，一個同儕節點可以向其父親節點及其他階層的同儕節點要求資料。

本論文提出的網狀覆蓋網路(Mesh Overlays Network)為基礎的同儕樹狀中，同時會有多個視訊串流傳播樹(Multiple Delivery Trees)傳送不同的資料給不同的同儕節點。雲端串流輔助伺服器則是用來提供當行動節點播放視訊串流時，當目前的下載點(Downloading Point)離撥放時間(Playout Point)小於評估最後下載時間閾值(Threshold)時，補足所缺少的視訊資料，避免因為緩衝區資料不足所造成的撥放中斷跟傳輸延遲。



(a) 網狀堆疊網路，每個節點有2個父節點

(b) 所對應的擴展樹

圖三 網狀堆疊網路映射至最佳擴展樹[3]

3.2 Tree-Based View of Mesh Overlays

網狀堆疊網路是非常易變的網路架構。因此，很難去分析網狀堆疊網路，相反的，樹狀架構網路是很容易理解的，也很簡單推導出樹狀的特性。如圖三(a)網狀堆疊網路中，每個節點都有2個父節點，對應到圖三(b)的擴展樹。如圖三所示，可將網狀堆疊網路對應成 K 多重擴展樹， $K=2$ 。

3.3 分析

本論文參考作者 Bartosz Biskupski 提出的

同儕串流架構及公式[3]，用在本論文同儕串流網路中，已知一個網狀堆疊網路可以看成一個 K 多重擴展樹的架構；在一個網狀堆疊網路中，每一個同儕節點可以從 K 個父節點下載串流視訊。假設，此網狀堆疊網路滿足論文[3]中的最佳化公式，則可將此網狀堆疊網路映射至一最佳擴展樹，如圖四所示。

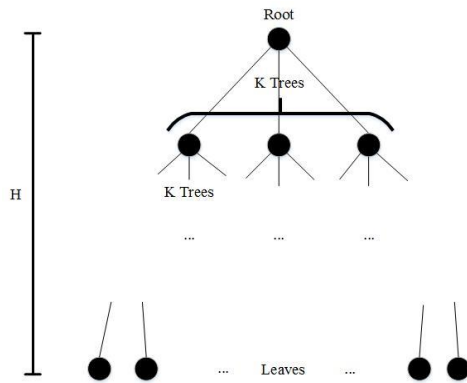
因此，在此最佳擴展樹裡，每個內部節點的分支(Out-degree, d)會等於 K 。在網狀堆疊網路中，假設節點數量為 M (包含來源端)，當網狀堆疊網路映射成最佳擴展樹後，總節點數量 N 等於分支度乘以網狀堆疊網路的節點數量，所以 $N=d*(M-1)+1$ 。在 K 多重擴展樹的高度就等於 $H(d,N)$ ，而最佳擴展樹的高度、分支 d 與擴展樹的階層 i 之間的關係，如公式(1)：

$$\sum_{i=0}^{H(d,N)} d^i = N(1)$$

公式(1)的幾何序列(Geometric Sequence)，得出方程式可以算出一棵樹最佳的平均高度，如公式(2)：

$$H(d,N) = \log_d((d-1)*N+1) - 1(2)$$

公式(2)中，最後一次的減一，是樹的最後一層為葉子節點，葉子節點並沒有分支，故減去一層。



圖四 K 棵擴展樹示意圖

在一個同儕擴展樹中，假設每個節點可提供的平均上傳率為 $Upload_{avg}$ ， N 為同儕擴展樹節點的總數量，包含 Root 端(由雲端串流伺服器下載視訊串流封包，再上傳給同儕樹中其它節點)，將節點總數量 N 的平均可上傳頻寬 $Upload$ 全部加總除以節點總個數 $(N-1)$ ，可以算出所有節點的上傳資料傳輸率。利用平均可上傳頻寬除以節點總數量得知，如公式(3)：

$$Upload_{avg} = \frac{\sum_i N upload_i}{(N-1)}(3)$$

當所有節點的上傳頻寬都達到飽和(Saturated)以及每條連結(Link)的上傳速率都相同，假設 K 為原網狀堆疊網路中，每個同儕子節點的父節點數量，可算出父親節點所能提供給子節點的傳輸速率 $Parent_{avg}$ 。如公式(4)：

$$Parent_{avg} = \frac{\sum_i N upload_i}{(N-1)*K}(4)$$

公式(3)、(4)利用每個節點的上傳頻寬加總除以節點的總數量，可以求出整體的傳輸速率。 F_{size} 為 Frame 的大小，反過來說，利用每個節點的總數量除以每個節點的上傳頻寬總和，就能得出每一個 Frame 傳輸在一條連結所需的傳輸時間 T_{frame} ，如公式(5)：

$$T_{frame} = \frac{F_{size} * (N-1) * K}{\sum_i N upload_i}(5)$$

根據公式(5)能算出父節點經由一條連結傳給子節點所需要的傳輸時間。進而得知，根節點(Root)產生一個新的 Frame 時間為 $T_{generate}$ 的單位傳輸時間(s)，如公式(6)：

$$T_{generate} = \frac{F_{size} * (N-1)}{\sum_i N upload_i}(6)$$

根據上述之公式(3)~(6)，可以算出同儕式網路的總傳輸速率以及總傳輸時間。根據公式(2)， $H(d,N)$ 為平衡樹的高度及公式(5)，傳輸在每一條連結中所需要的傳輸時間，資料從根節點傳輸到葉節點的總傳輸延遲時間 Delay，如公式(7)：

$$Delay = \frac{H(d,N) * F_{size} * K * (N-1)}{\sum_i N upload_i}(7)$$

依據資料的總傳輸量，從起始時間 T_{start} 到結束時間 T_{end} 接收到的封包總傳輸量 Pkt_{num} 乘上封包平均大小 Pkt_{size} 的吞吐量 Throughput，來評估整體的同儕樹狀網路效能，如公式(8)：

$$Throughput = \frac{Pkt_{num} * Pkt_{size}}{T_{end} - T_{start}}(8)$$

3.5 雲端串流伺服器之增減

同儕式網路中，網路的傳輸情況會隨著時間的改變與節點的移動性會有不一樣的變異(Variation)，在頻寬不足時，同儕節點可能會無法下載完整的視訊串流資料。此時，不同視訊解析度的視訊串流傳播樹有不同的根節點(Roots)，負責向雲端串流伺服器下載不同視訊解析度的即時串流後轉送(Forwarding)給其他同儕子節點。

當行動同儕節點播放串流視訊時，若緩衝

區資料不足，則採用推拉方式(Push-pull Approach)，向其他同儕節點要求提供缺少的視訊資料。當目前下載點(Downloading Point)離播放時間(Playout Point)小於評估最後下載時間門檻值(Threshold)時，且其他的同儕節點無法提供缺少的視訊資料，該節點可在 Tcloud_patching 定義的時間內直接向雲端上的 Patching Streamer 要求下載缺少的視訊資料，來補足因為緩衝區資料不足所造成的播放中斷以及傳輸延遲。

當同儕節點向雲端串流伺服器 BL Streamer、EL Streamer 與 Patching Streamer 下載視訊串流資料時，要如何去分配 Cloud Streamer 的數量來達到最高的成本效益是非常不容易的，分配過多的 Cloud Streamer 可以降低 Patching Streamer 的負擔>Loading)，這樣成本消耗太高，分配過少的 Cloud Streamer，會提高 Patching Streamer 的負擔，最後也會提高使用雲端運算的成本。

四、效能分析

4.1 模擬環境

本系統模擬同儕式網路結合雲端運算之效能分析，本文比較 Client-Sever、Pure P2P 與本文提出的 Cloud-Assisted P2P 的架構。在模擬系統環境中，進行伺服器負載量、傳輸延遲時間、吞吐量、服務品質與使用雲端服務的成本來進行比較。

效能分析利用 JSVM 軟體架構[7] 來擷取(Extract)可擴展視訊編碼中視訊基礎層(Base Layer)以及視訊加強層(Enhance Layer)的資訊。JSVM 針對視訊串流進行可擴展視訊編碼，可以編碼出一個高品質視訊的位元串流，其中也包含一個或多個視訊品質較低的子位元串流，使用者可以根據自身的頻寬條件抓取適合的視訊位元串流來觀看，藉此達到可擴展視訊編碼的目的。

並把擷取出來的視訊基礎層和視訊加強層利用 FFmpeg 提供的 ffprobe 工具進行視訊串流的分析，取得所需要的視訊串流資料加以運算。FFmpeg 是一套自由軟體，針對聲音和視訊多種格式進行轉檔和切割。本次實驗使用 4CIF(704x576)和 CIF(352x288)兩種解析度來進行可擴展視訊編碼(H.264/SVC bit-stream)，使用 GOP 為 8 的編碼長度，如圖六。

	bitrate	Min-bitr	Y-PSNR	U-PSNR	V-PSNR
352x288 @ 3.7500	261.7366	261.7366	35.9664	42.6486	44.2174
352x288 @ 7.5000	289.0944	289.0944	35.8676	42.6721	44.2636
352x288 @ 15.0000	319.0067	319.0067	35.6587	42.6518	44.2337
352x288 @ 30.0000	353.0150	353.0150	35.4233	42.6191	44.2302
704x576 @ 3.7500	1135.8474	1135.8474	36.3892	42.9568	44.8667
704x576 @ 7.5000	1268.4184	1268.4184	35.7660	42.8820	44.8602
704x576 @ 15.0000	1397.2687	1397.2687	35.3409	42.7832	44.8204
704x576 @ 30.0000	1548.0604	1548.0604	35.0480	42.6681	44.7754

Encoding speed: 4927.919 ms/frame, Time:1468520.000 ms, Frames: 298
ftz@ubuntu:~/58e5

圖六 可擴展視訊編碼

將編碼完成的位元串流率利用 JSVM 所提供的 BitStreamExtractorStatic 工具進行視訊串流的提取(Extract)，將所提取出來的視訊基礎層以及視訊加強層進行資料量的運算，並取得視訊加強層的資料量，藉由利用 FFmpeg 提供的工具 ffprobe 抓取視訊的檔頭資訊、畫面的種類和畫面的資料量，將分析結果進行資料量的計算。

模擬環境中，如圖七。同儕節點數量 5,000~35,000、Upload 為 100,000kbps、同儕節點的上、下傳頻寬 500kbps、BL Frame 大小約為 13kb(低解析度)、EL Frame 大小約為 38kb(高解析度)、所有同儕節點下載通道 8 條、三台串流伺服器(兩台 BL Streamer、一台 EL Streamer)，根據頻寬條件每台 BL 伺服器可服務 5,000 個節點，而每台 EL 伺服器可服務 1,667 個節點。

參數設定	
同儕節點數量	5,000~35,000
Upload	100,000kbps
同儕節點的上、下傳頻寬	500kbps
BL Frame大小	13kb(低解析度)
EL Frame大小	38kb(高解析度)
同儕節點下載通道	8條
串流伺服器	兩台BL Streamer、一台EL Streamer
一台BL Streamer	根據頻寬可服務5,000個節點
一台EL Streamer	根據頻寬可服務1,667個節點

圖七 模擬環境設定

4.2 效能分析

4.2.1 傳輸延遲時間

依據公式(7)之定義，針對三種架構 Client-Sever、Pure P2P 以及 Cloud-Assisted P2P。不同的架構中，需要調整公式內部的變數來符合各個不同的架構。為了方便討論，我們將 BL Frame 與 EL Frame 分別計算，達到簡單化。Client-Sever 環境中，使用者直接向工作伺服器要求視訊串流封包，公式(7)中每個節點的分支(Out-degree)會等於節點總數量減一，其中， T_{sever_client} 為 RTT 時間，公式調整如下：

$$Delay = \frac{frame\ size * K}{Upload_{Sever_client}} + T_{sever_client}, (K = N - 1)$$

Pure P2P 環境中,不必做任何調整,其中, T_{sever_root} 為 RTT 時間,如公式(7):

$$Delay = \frac{tree\ height * frame\ size * K * (N - 1)}{\sum_i^N upload_i} + T_{sever_root}$$

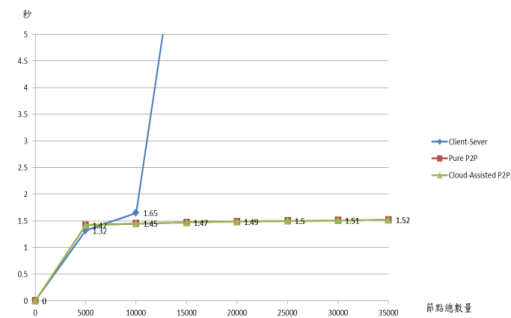
Cloud-Assisted P2P 環境中,當目前的下載點(Downloading Point)離撥放時間(Playout Point)小於評估最後下載時間門檻值(Threshold)時,則觸發緊急事件,直接向 Patching Streamer 補足(Patching)所缺少的視訊資料,避免因為緩衝區資料不足所造成的撥放中斷跟傳輸延遲,參考圖二。其中, T_{cloud_root} 為 RTT/2 時間,公式調整如下:

$$Delay = \min \left(\frac{tree\ height * frame\ size * K * (N - 1)}{\sum_i^N upload_i} + T_{cloud_root}, T_{transmission_delay} \right)$$

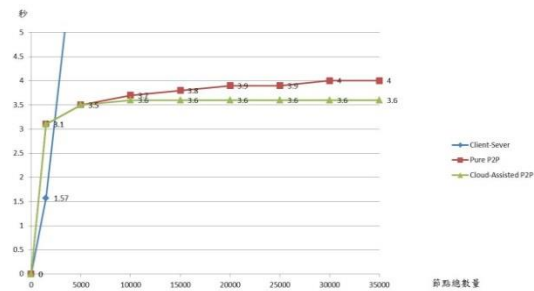
圖八為 BL Frame 在三種不同架構中的傳輸延遲時間。菱形線條為 Client-Sever,在節點總數量已達 10,000 個時,因為超過兩台 BL Streamer 加總起來可以服務的節點總數量,所以系統已超過節點可需求的數量。方形線條為 Pure P2P,當同儕根節點(Root)收到雲端視訊串流封包時,利用“推拉”的方式將視訊串流封包傳遞給其他的同儕節點。三角形線條為 Cloud-Assisted P2P,傳遞視訊串流封包的方式跟 Pure P2P 一樣,而且也尚未達到所設定的最後下載時間門檻值,不用直接向雲端串流伺服器需求下載,故 Pure P2P 與 Cloud-Assisted P2P 為同一條線上。

圖九為 EL Frame 在三種不同架構中的傳輸延遲時間。菱形線條為 Client-Sever,在節點總數量已達 1,667 個時,因為超過一台 EL Streamer 可以服務的節點總數量,所以系統已超過節點可需求的數量。方形線條為 Pure P2P,當同儕根節點(Root)收到雲端串流封包時,利用“推拉”的方式將視訊串流封包傳遞給其他的同儕節點。三角視訊形線條為 Cloud-Assisted P2P,在傳遞延遲時間與最後下載時間的門檻值(Threshold),超過所設定的門檻值時,則直接向雲端串流伺服器(PS)補足所缺少的視訊串流封包,所以線條曲線在門檻值

時,呈現一直線。



圖八 BL Frame 傳遞延遲時間



圖九 EL Frame 傳遞延遲時間

五、結論

因為不穩定的無線網路會造成訊號的中斷,再加上同儕節點經常性的加入、離開,使得同儕節點會遭受長時間的等待和突發性的下載中斷。當大量的使用者在行動網路下觀看視訊串流時,所累積的資源租用成本費用極高,而資料來回傳輸的過程中,也會導致延遲成本的增加。

本文提出的結合雲端運算技術與網狀覆蓋網路為架構支援多媒體服務品質視訊串流傳輸架構,透過雲端運算進行可擴展視訊編碼,同時利用同儕式網路的共享機制,讓使用者根據自身的網路頻寬條件來選擇適合當時的狀況來觀看視訊串流。同時考慮到視訊串流最後下載的時間門檻值內,若超過時間門檻值,則直接向雲端下載所缺少的視訊串流,來避免長時間的下載等待或經常性的下載中斷。

本文提出的架構租用雲端運算資源,只需部份費用就能改善同儕(P2P)運算網路傳輸的服務品質,針對 Sever-Client、Pure P2P 與本論文的架構在模擬環境中的傳輸延遲時間進行比較,實驗結果顯示本論文的架構都比 Sever-Client 和 Pure P2P 來的優異。

誌謝

本研究計劃執行所須之研究經費係由科技部所提供。計劃編號: (MOST

102-2221-E-218-016 -)

六、參考文獻

- [1] A. Segall and G. J. Sullivan, "Spatial Scalability Within the H.264/AVC Scalable Video Coding Extension," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1121–1135, Sep. 2007.
- [2] A.H. Payberah, H. Kavalionak, V. Kumareshan, A. Montresor, and S. Haridi, "CLive: Cloud-Assisted P2P Live Streaming," In *Peer-to-Peer Computing (P2P)*, pp. 79–90, IEEE 12th International Conference, 2012.
- [3] B. Bartosz, S. Marc, F. Pascal, M. Rene, "Tree-Based Analysis of Mesh Overlays for Peer-to-Peer Streaming," In *Distributed Applications and Interoperable Systems*, pp. 126–139. Springer, 2008.
- [4] B. Pan, X. Wang, C. Hong, S. Kim "AMVP-Cloud: A Framework of Adaptive Mobile Video Streaming and User Behavior Oriented Video Pre-Fetching in the Clouds," *IEEE*, pp.1-8, October. 2012
- [5] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103–1120, Sep. 2007.
- [6] J. Sacha, J. Napper, C. Stratan, and G. Pierre, "Adam2: Reliable distribution estimation in decentralized environments," in *Proc. of the 30th IEEE International Conference on Distributed Computing Systems. (ICDCS'10)*, pp. 697-707, June. 2010.
- [7] JSVM Software Manual, <http://ube.ege.edu.tr/~boztok/JSVM/SoftwareManual.pdf>.
- [8] M. Jelasity, A. Montresor, and O. Babaoglu, "Gossip-based aggregation in large dynamic networks," *ACM Transactions on Computer Systems.(TOCS)*, vol. 23, 2005.
- [9] T. Wiegand, G. J. Sullivan, J. Reichel, H. Schwarz, and M. Wien, *Joint Draft 11 of SVC Amendment, Joint Video Team*, Doc. JVT-X201, Jul. 2007.
- [10] Y. Xu, C. Zhu, W. Zeng, X. J. Li, "Multiple description coded video streaming in peer-to-peer networks," *ADVANCES IN 2D/3D VIDEO STREAMING OVER P2P NETWORKS*, pp. 412-429, vol. 27, May. 2012.
- [11] Z. Huang, C. Mei, L. E. Li, and T. Woo, "CloudStream: delivering high-quality streaming videos through a cloud-based SVC proxy," *IEEE INFOCOM*, pp. 1–5, April. 2011.



