

# 使用 HB 廣播機制的點對點串流系統

曾黎明 簡廷宇 陳彥儒 楊宏昌

國立中央大學資訊工程研究所

tsenglm@csie.ncu.edu.tw, bowkaka@dslab.csie.ncu.edu.tw,

oldbig@dslab.csie.ncu.edu.tw, cyht@dslab.csie.ncu.edu.tw

## 摘要

網路的發展，促進了許多資訊交流。近年來，由於網路速度的大幅提升，不僅是網頁，許多音樂、影片等多媒體檔案都可以在網路上互相流通。但是由於影音串流需要的頻寬較大，所以更容易在主從式架構中發生供應原超載來不及處理與供應原路由頻寬不足造成壅塞的現象

因此本論文提出了一個點對點串流的架構 P2PHB，結合了 Harmonic Broadcasting 架構，將原本的分頻道設計適用在點對點網路中，並且以應用層群播的方式進行串流的轉送，對於點對點網路架構中節點可能失效的情形提出了偵測與復原的機制。在進行特定錄放功能如跳轉、倒轉時，較 DirectStream 有更好的片段命中率。並針對熱門度分析各節點提供快取之容量分析，另對 PPBS 與 P2PHB 之異同進行模擬比較分析，可證明本方法在起始延遲具更佳表現。

**關鍵詞：**點對點串流、錄放功能、Harmonic Broadcasting、熱門度

## 1. 前言

隨選視訊系統(Video-On-Demand)[1]則被認為影音串流發展的主流。但是由於影音串流需要的頻寬較大，所以更容易在主從式架構中發生來不及處理等壅塞的現象，使得服務效能降低。所以近年來點對點架構(peer-to-peer scheme)的相關研究也越來越多。點對點架構的優點是隨著使用者人數的增加，還是可以展現其良好的擴充性，由於不會有所有請求必須讓伺服器處理的問題。對於需要大量資料傳輸的多媒體資訊也漸漸有許多相關的軟體如 pppstream, GnuStream, Peercast [2-4]等或是論文被提出。傳統的主從式架構中也有提到許多種廣播或是群播方法：例如 Fast Broadcasting[5,6]、Harmonic Broadcasting[7]等，但是群播直到現在還是沒有辦法被廣泛應用在網路上，因為這樣會增加路由器的負擔。本文結合了 Harmonic Broadcasting 架構與點對點架構，提出一個可以適用在點對點網路架構的影音串流架構，稱為“peer-to-peer streaming system using harmonic broadcasting scheme”(P2PHB)，在這個架構下，我們還提出了是用於本架構所需要的快取策略、節點失效偵測與復原策略、以及錄放功能的支援。

## 2. 相關研究

### 2.1 Harmonic broadcasting

Harmonic Broadcasting(HB)是一種廣播的方式，較以往的廣播方式不同之處在於可以使得來訪者的等待時間大幅減少，除此之外對於頻寬的需求量也不會為頻道數增加而跟著大幅增加，而是隨著頻道數增加而遞減。例如：一個 120 分鐘的影片如果要讓他的等待時間為 4 分鐘時，僅需要 4 個影片頻道的頻寬傳輸量。

Harmonic Broadcasting(HB)的原理如下：如圖 1，當一個影片被切割成等長的  $N$  個等長的片段，這  $N$  個片段對會被分配到  $N$  個頻道廣播，而每個頻道廣播的頻寬會分別遵循 Harmonic Number(HN)的法則，也就是頻道  $i$  的廣播頻寬就是  $b/i$ ，其中  $b$  是影片的播放速率。在這樣的狀況下，當有一個新的客戶端加入這個 HB 架構的網路時，會同時抓取頻道 1 到頻道  $N$  的串流，並且接收的速度為  $b/i$ 。整體頻寬的需求(B)為

$$B = \sum_{i=1}^N \frac{b}{i}$$

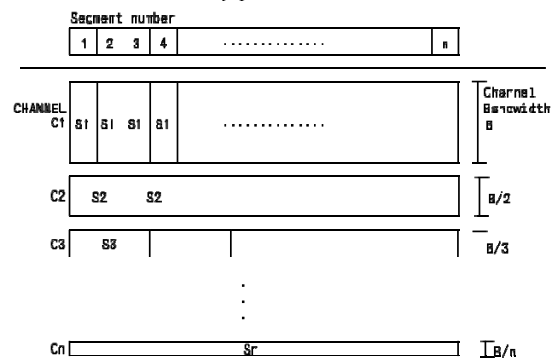


圖 1 摘自[7]， Harmonic Broadcasting 頻寬分配示意圖

### 2.2 Adaptive Live Broadcasting

Adaptive Live Broadcasting(ALB)[8]為一種適應於現場串流(Live streaming)的一種廣播技術，可以讓加入的節點等待時間不超過一個片段的時間長，並且適應在現場串流下，由於現場串流的片段必須要等到一個片段時間長生成後才可以廣播該片段，並且不能已先行下載後面片段的方式進行最佳化。

## 2.3 Fast and lossless Broadcasting

本文[9]提出可以適用於動態流量之影音串流廣播系統，並且提出三種不同的演算法進行實驗，分別為 Smooth Equal-Length Broadcasting(SELB)，每個頻道所廣播的片段節點之影片時間長度為固定，但因為是 VBR 影片，所以片段的大小將不會固定。Smooth Equal-Size Broadcasting(ESLB)則是將影片切割成同樣大小之片段，所以會發生片段大小相同但是片段時間不同。Smooth Equal-Bandwidth Broadcasting(SEBB)為將每個頻道廣播之頻寬為固定大小，每個片段域在都要在片段正式開始播放之前下載完成，而最後實驗結果發現 SEBB 的效果最好。

## 2.4 PPBS

Peer-to-Peer Broadcasting Scheme (PPBS)[10]採用 HB 的架構來進行點對點的廣播影音串流，PPBS 假設網路上的節點距離非常近，且假設所有節點已經被同一個時間計同步。當節點在這個網路上時，就可以幫忙廣播影音串流，所以，在 HB 架構中的每個頻道就可以由一群節點伺服器群(Peer Server Group)替代，參考圖 2，在同一個時間，一個頻道只會有一個節點負責廣播影音串流給接收者，N 個頻道也就會有 N 個不同的節點伺服器進行廣播的動作。

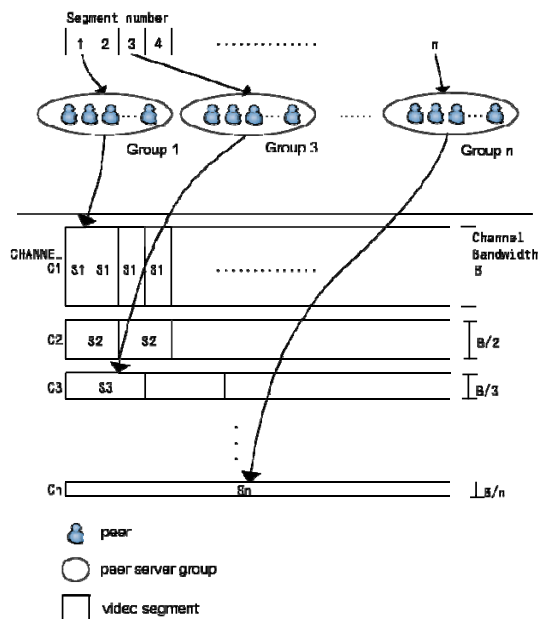


圖 2 摘自[10]，PPBS 架構示意圖

## 2.5 錄放功能

錄放功能為對影片的播放速度進行控制，可分為暫停(pause)、快轉(fast forward)、倒轉(back forward)等，對於點對點串流的錄放功能控制的相關研究有：多重播放流量法[11]，利用 N 台伺服器

儲存不同流量(bit-rate)的影片，來依照客戶端的頻寬給予不同流量的影片。層次編碼(layer coding)的方法[12][13]，是利用在壓縮影片的時候就進行分層的資訊儲存，例如底層(base layer)僅儲存影音串流的基本資訊，往上層就是加強的資訊，當客戶端選擇進行錄放功能時就僅接收客戶端能夠容許同時進行快轉或倒轉功能的資料。通透性編碼(Transcoding)的方法[14][15]，其中不將影片做重新編碼的處理，但是當客戶端要求進行快轉或是客戶端頻寬不夠的狀況時，才進行影音串流的即時編碼。片斷分割法[16][17]，其中[17]利用代理伺服器動態調整片段被快取的比例，使得錄放功能得以順暢呈現。

## 3 系統設計

本節將介紹我們所提出的 P2PHB 架構，首先介紹系統的組成，我們將介紹在 P2PHB 架構下成員所扮演的角色，接下來會介紹系統分析，對另一個使用點對點廣播 Harmonic Broadcasting 架構的[10]進行功能性比較，之後將詳細敘述系統整體的運作流程，以及各個不同組件的策略。

### 3.1 系統組成

首先我們將介紹 P2PHB 與原本 Harmonic Broadcasting 的差異性，本系統的成員有三：

1. 影音列表網頁伺服器(Streaming list web server)
  - 提供使用者瀏覽影片列表資訊
  - 維護影片列表中所對應的頻道伺服器列表
2. 影音串流伺服器(Streaming server)
  - 維護正在進行串流的節點列表，包含所提出的 HN、節點是否有對其他節點進行影音串流轉送、加入的順序
  - 接受並且分配節點對串流的請求
  - 對請求的節點送出串流計算影片的熱門度
3. 節點(Peer)
  - 接收影音串流
  - 根據給予的熱門度變化快取不同數量片段
  - 轉送影音串流給其他的請求節點

### 3.2 系統分析

對於三種架構，分別為傳統的 HB 架構、以及 PPBS 架構、P2PHB 架構之方法比較。在 HB 原生架構中，是指在同一個區域網路下進行廣播的情形，由於距離非常近，並且頻寬也相仿，所以效能是非常不錯的，並且在此架構下可以得到少許的初始等待時間。且由於為廣播架構，每個節點僅進行接收影音串流的動作，所以對於節點失效時，整體系統不會有影響。而雖然 Harmonic Broadcasting 有

著相當好的效能，但還是會受限於距離上與通訊協定的限制。而後由[10]所提出的 PPBS，提出了當頻道正在廣播的節點伺服器失效或是離開時，馬上會有新的節點接替原先的頻道伺服器的錯誤偵測復原功能。

表 1 HB,PPBS,P2PHB 架構分析表

|       | 網路架構      | 熱門度測<br>量 | 錄放功能   | 起始延遲 | 適用性 | 節點擴充<br>需求 | 節點離開<br>對系統影響 |
|-------|-----------|-----------|--------|------|-----|------------|---------------|
| HB    | 廣播        | X         | X      | 較長   | 區域  | 較低         | 無             |
| PPBS  | 廣播<br>廣播  | X<br>X    | X<br>X | 較長   | 區域  | 較低         | 無             |
| P2PHB | 應用層群<br>播 | 0         | 0      | 短    | 全域  | 較高         | 低             |

### 3.3 運作流程

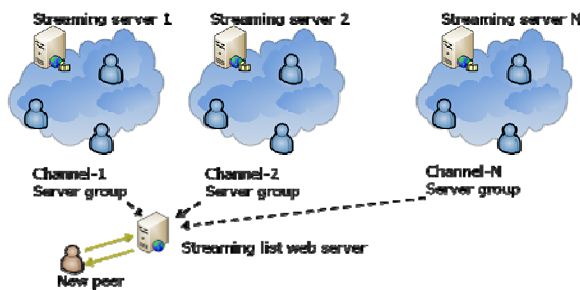


圖 3 P2PHB 使用者加入示意圖

當 P2PHB 網路初始的時候，首先會將影片切成  $N$  個片段，分別儲存到  $N$  個影音列表網頁伺服器，接下來影音串流伺服器會先向影音列表網頁伺服器註冊自己的位置，新的節點開始向每個影音串流伺服器作加入的請求。圖 4 之右方表格為影音串流伺服器維護之串流列表，表格最左方為正在串流之節點 IP 位置，Hs(Harmonic Sequence)欄位則是紀錄節點進行串流的下載速度，在傳統 HB 中的 Hs 就是該片段所在的頻道數，也就是  $b/i$ ，其中  $b$  為影片的播放流量、 $i$  為所在頻道數，但是在這裡要特別紀錄的原因是當有節點使用了 VCR 功能之後，例如一開始使用者就直接從第  $i$  個頻道開始播放，那麼就會與原本的下載架構不符合，後面的錄放功能策略章節將會詳述如何處理這樣的情形。

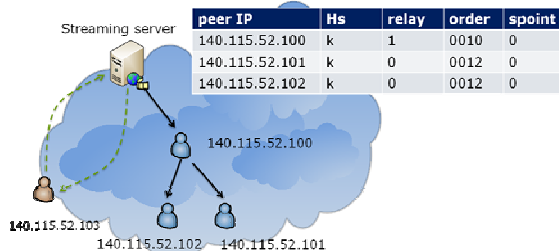


圖 4 頻道 k 使用者加入示意圖

#### 3.3.1 快取策略

首先我們需要了解的是原本的 HB 架構下至少

需要快取多少數量的片段，在[7]中，切割的片段數為 1000 時所需要快取的片段佔全部影片長度不會超過 37%，在一個片段播放間隔  $d$  的時間中，Harmonic Broadcasting 需要快取的片段所佔的容量最多為：

$$B = \left( \frac{S}{N} * \sum_{j=i+1}^N \frac{1}{j} \right) \quad (1)$$

其中  $S$  為片段容量， $N$  為切割的片段數， $i$  為目前正在觀賞的片段。

在有新的節點加入的時候，影音串流伺服器除了會分派父節點給新的節點連接之外，伺服器也會進行相關的熱門度統計，熱門度與節點的快取量是正相關的，也就是說越是熱門的影片，伺服器就會要求新加入節點快取越多的片段，但是當影片很少人觀賞的時候，節點就可以調整自己快取的片段量，不過增加或是減少並不是得到伺服器熱門度資訊後立即調整，而是在節點看到下一個片段的時候才進行調整，以免發生影片變冷門時原本有快取的片段因為被伺服器通知不需要快取而被立即刪除，造成不必要的重新連線延遲。在敘述了策略之後，我們要先對熱門度( $\rho$ )進行定義：

$$\rho = \sum_{k=1}^Q \frac{I_{now-k}}{\tau * k} \quad (2)$$

其中  $\rho$  為影音串流伺服器的統計時間間隔，而  $I$  為在統計時間間隔中所有到訪加入節點的數量， $Q$  為常數，做為熱門度是否與過去統計有相關的調整，當  $Q$  的數值越大就會計算到越多的過去訪客到訪的紀錄，而當  $Q=1$  時就是只紀錄最近一次的來訪量。

根據(1)與(2)式，我們可以定義出需要由節點快取的片段與節點本身播放就需要快取的片段量總和為：

$$B = \left[ \left( \frac{S}{N} * \sum_{j=i+1}^N \frac{1}{j} \right), \left( \frac{S}{N} * \sum_{j=i+1}^N \frac{1}{j} \right) + \alpha \log \sum_{k=1}^Q \frac{I_{now-k}}{\tau * k} \right]$$

其中  $\alpha$  為常數，我們可以藉由控制調整因為影片的熱門度對於片段快取需求量的依賴程度。另外因為節點的快取量是有限的，所以節點可以視自己的儲存空間是否足夠來決定是否要多快取這些片段。

#### 3.3.2 節點失敗偵測與復原

在點對點網路中，所有的節點隨時都有可能加入或是離開，提出節點失敗偵測與復原之前，我們需要在一次回顧節點在加入系統時會做的其中一件事情，就是紀錄影音串流伺服器、父系節點及祖父系節點的 IP 位置資訊。

這個時候後來加入的節點如果為子節點的話，就只要與父節點要求相關的 IP 位置資訊，而且由於父節點有其兄弟及祖父節點的 IP 位置資訊，所

以不需要再費工夫跟其他節點請求資訊。

節點互相擁有 IP 清單的用意是在於當父節點發生失敗時，可以利用清單上的列表進行修復，當節點超過一個時間門檻( $\delta$ )沒有收到來自父節點的串流，或是沒有收到子節點的回應時，就會認定該節點已經失效，對於已經失效的父節點，採取的策略是看看清單中的其他父節點是否可以提供串流，將會採取同時詢問所有父節點的策略以節省時間，如果可以的話，就往新的父節點接上，回復原本的串流。但如果所有的父節點都已經失效，這個時候就會轉向尋找祖父節點的串流，如果祖父節點依然健在，就捨棄父節點向祖父節點連接。但是如果連所有祖父節點也失效的時候，由於經過的時間不可以過長，再重新尋找頻道伺服器群中可以連接的節點太沒有效率，所以節點就只好緊急從影音串流伺服器直接串流，維持播放的品質，經過每一個 $\tau$ (影音串流伺服器的統計時間間隔)就會重新檢查這些節點是否可以重新接上其他適合的節點，讓伺服器的頻寬得以節省，也可以讓節點可以有更好的串流品質，所以我們可以把該流程化成以下流程圖：

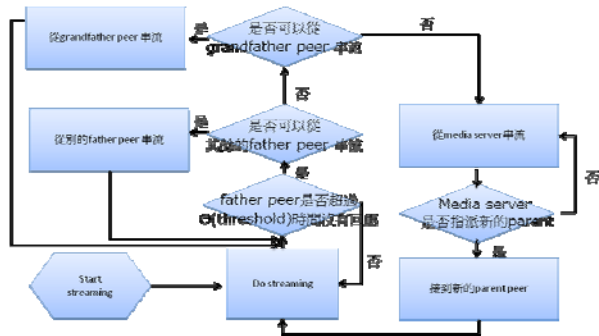


圖 5 節點失效復原流程圖

接下來要分析修復時間的長度，我們可以分析當一個節點從察覺到找到新的可串流節點所需要的平均修復時間為： $R_{AVG} = \delta + [2(1-\varepsilon^k)d_{ipff}] + [(1-l^k)\varepsilon^k(\delta + 2d_{ipff} + 2d_{ipgm})] + \{2[l^k\varepsilon^k(\delta + d_{ipff} + d_{ipgm})] + D\}$

表 2 修復時間代碼對照表

| 符號            | 代表意義                            |
|---------------|---------------------------------|
| $\delta$      | 節點等待判定失效時間門檻                    |
| $d_{ij}$      | 從節點 <i>i</i> 到節點 <i>j</i> 的連線延遲 |
| $D$           | 影音串流伺服器到任何節點的平均延遲時間             |
| $P_{fi}$      | 父節點 <i>i</i>                    |
| $P_{gi}$      | 祖父節點 <i>i</i>                   |
| $k$           | 父節點數量                           |
| $l$           | 祖父節點數量                          |
| $\varepsilon$ | 節點失效的機率                         |
| $P_i$         | 節點 <i>i</i>                     |
| $S_k$         | 伺服器 <i>k</i>                    |
| $R_{AVG}$     | 平均修復時間                          |

### 3.3.3 錄放功能

在 P2PHB 系統架構中，也可以順利的提供如暫停、快轉、倒轉等不同的錄放功能，以下將針對不同的錄放功能在 P2PHB 架構下如何運作以及相關流程。

跳轉(jump forward)：跳轉的步驟如下，首先快轉可以視為一個節點離開原本正在進行串流的串流伺服器群，接下來再以一個新加入節點的身分重加入頻道伺服器群中，即便是節點希望快轉的部分依然在同一個片段中，還是要進行重新加入的動作，這個時候加入的節點與一般正常撥放速度的節點不同，對於以快轉模式加入頻道伺服器群的節點，取得串流的速度便會與一般的使用者不同，所以在影音串流伺服器中的  $H_s$  欄位就是要紀錄因為快轉加入的節點重新計算過的  $H_s$  值。對於一般加入節點的第  $k$  個頻道來說，取得串流的速率是  $S/k$ ，但是對於快轉過來第  $k$  頻道的使用者，對他們來說是第一個頻道，所以在這裡就要把  $H_s$  欄位紀錄為 1，而不是原本的  $k$ ，其中：

$$\begin{cases} S_{new} = (S_{old} + 1) \bmod N - 1, S_{P_{old}} > S_{new} \\ S_{new} = (S_{old} + 1) \bmod N, S_{P_{old}} = S_{new} \end{cases}$$

這個時候對節點來說，片段  $k$  就是節點現在要看的第一個片段，而第二個片段就是  $k+1$ ，就可以讓重新加入的節點正常取得串流而不會因為進行錄放功能而導致無法正常撥放影片。在影片進行快轉之後，由於  $H_s$  欄位不同的節點並不會在同一支轉送路徑上。

跳倒轉(jump backward)：跳倒轉與跳轉相仿，不同之處在於快轉是向後面的片段移動，跳倒轉是向前面的片段移動。而前面的片段可能是還被保存在快取空間中。

快轉(fast forward)：進行快轉時，首先要節點跟串流伺服器取得已經快取完整影片片段的節點來作為父節點才行，如圖 6，在這裡的節點 D，目前正在看第 3 個片段，並且他也快取了第 1 個與第 2 個片段，如果此時節點 D 想要進行快轉的功能，節點 D 就會直接向其父節點 C 要求編號為偶數之與必要之 I 與 P 頁框。且由於 2 倍速快轉的原因，播放的時間為原本一個片段長度時間的 1/2，所以在 1/2 個片段播放時間後，節點 D 就會轉向往節點 B 要求片段 4，繼續進行 2 倍速快轉。3 倍速快轉的部分則是取編號 3k 的影片頁框，而快取的需求則更高。

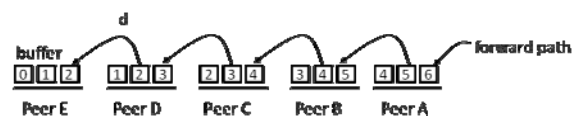


圖 6 快轉節點快取狀態示意圖

倒轉(backward)：倒轉與快轉類似。快轉為漸往右移，而倒轉則為往左移。

暫停(pause)：暫停的功能就是停止原本正在播放的影片，所以也不需要繼續接收資料。但是對於 P2PHB 的架構中並不會立即停止影片繼續串流，因為可能還有子節點正在接收自身發出的串流，所以這個時候還是繼續進行串流，直到快取空間已滿無法繼續維持現在的暫停播放點，這個時候才會真正的離開頻道伺服器播放群。

## 4 效能評估

### 4.1 實驗環境

- ◆ 模擬軟體：NS2(Network Simulation 2)
- ◆ 影片長度：60 分鐘
- ◆ 影片播放速率：平均 1147Kbps(140KB/S)
- ◆ 平均 I/P/B frame 容量：22.26K/7.508K/2.031K
- ◆ 影片切割片段數：120 個
- ◆ Harmonic Number：5.369
- ◆ 使用者到訪問隔時間：遵循 lognormal 分佈

$$\left( \frac{1}{\sqrt{2\pi}\sigma} \exp \left\{ -\frac{(\log(x) - \mu)^2}{2\sigma^2} \right\} \right)$$

其中  $\mu=3.59$ ， $\sigma=1.52$

- ◆ 節點個數：1000 個節點
- ◆ 在這裡我們以 GT-ITM[18]拓模產生器產生隨機的 transit-stub，並由 GT-ITM 決定每個節點的傳送延遲時間。由兩個 transit domain 組成，其中兩個 transit domain 中又分布 8 個 stub domain。
- ◆ 節點頻寬：10Mbps/100Mbps 節點的接收頻寬為 10Mbps，而影音串流伺服器與路由器則為 100Mbps

### 4.2 實驗數據與分析

#### 4.2.1 實驗一：起始延遲時間

在實驗一中，我們將 1000 個點依照 lognormal 分佈加入網路進行串流，並且紀錄每個點的啟始延遲時間，起始延遲時間指節點加入頻到伺服器群時，先與影音串流伺服器取得連接的節點後，在與新的父節點溝通，直到正式接收串流可以解出第一張 I frame 為止的時間。如圖 7，在 P2PHB 的表現(藍色,下方的線條)中，1000 個點來說的平均延遲時間只有 1.495 秒，另外在 PPBS(紅色,上方的線條)中，因受限於廣播的周期性，所以新加入的節點必須要等下一個周期的廣播起始後才可以正式開始播放，造成起始等待時間上升的情形，在這裡的 d，也就是片段與片段之間的時間間隔為 30 秒鐘，實驗結果得到 1000 個節點的平均等待時間為 12.75

秒。

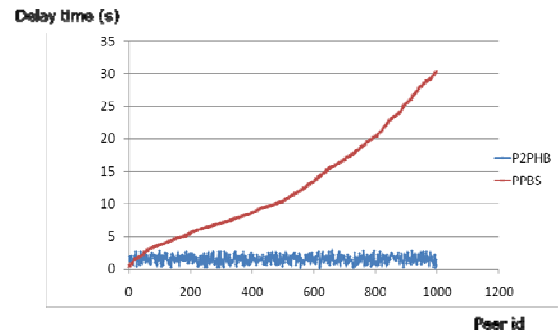


圖 7 起始延遲時間比較圖

#### 4.2.1 實驗二：失效復原時間

實驗二中，我們假設點對點串流中的節點失效情形發生(如圖 8)，針對點對點網路中，節點時常會有來來去去的狀況，所以在此實驗 1000 個節點中，如果在同一時間有 100 個節點同時失效時，整個每個節點進行復原的時間需要多少。在本實驗我們可以得到 PPBS 的平均延遲時間 1.804 秒，而 P2PHB 的延遲時間為 2.32 秒，此外在這裡需要討論的是 PPBS 的延遲時間會隨著失效的節點不同有很大的差異，因為 PPBS 的架構為廣播，所以當一個節點失效的時候，系統的修復、所有的節點能不能夠繼續看都仰賴備援節點的補上，如果第二優先的節點也失效的話，那麼延遲時間就會再繼續增長。

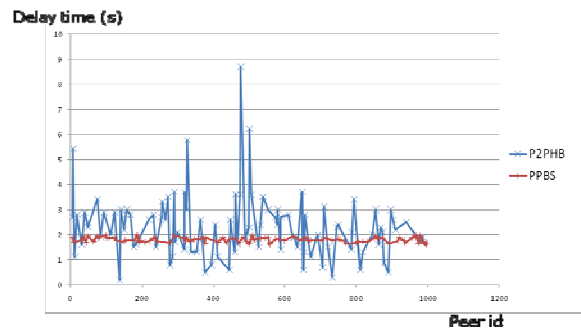


圖 8 節點失效復原時間延遲圖

#### 4.2.1 實驗三：錄放功能快取命中率

實驗三(如圖 9)，我們假設使用者做跳轉或倒轉的功能距離與行為是均勻分布的，可以是播放時間的任何一個點，並且假設使用者自己的快取空間為最大，最多可以快取所有的影片空間。在這裡我們以 Directstream 與 P2PHB 架構進行比較，原因是因為 PPBS 架構上並不支援錄放功能，而實際上 PPBS 的運作環境也是在可廣播環境下，因此我們以 Directstream 進行效能評估的比較方向。我們可以發現 P2PHB 有著比 Directstream 平均高 1.46 倍的快取命中率，尤其在影片剛播放的時間，快取的命中率較 Directstream 有著更顯著的表現。原因是 P2PHB 在播放初起會取 HN 的頻道下載量，在之後的所有頻道都

會依照 b/i 的速度進行快取，而 Directstream 則無，所以造成快取命中率的差異。

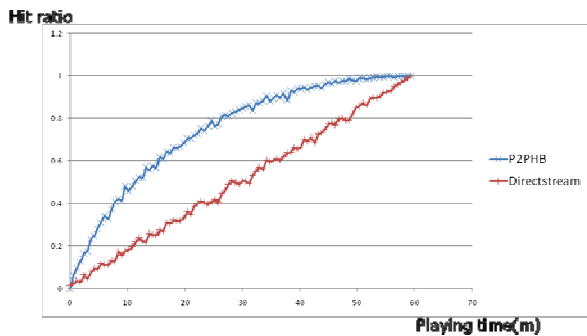


圖 9 錄放功能命中率與播放時間圖

## 5 結論

本論文提出了一個點對點串流的架構 P2PHB，結合了 Harmonic Broadcasting 架構，將原本的分頻道設計適用在點對點網路中，並且以應用層群播的方式進行串流的轉送，並且對於點對點網路架構中節點可能失效的情形提出了偵測與復原的機制。

最後，本文提出模擬實驗，比較 P2PHB 與 PPBS 架構在起始延遲時間與節點失效時的復原時間進行比較，我們可以看到在同樣的狀況下，P2PHB 可以得到更少的初始延遲時間，減少節點剛加入時的等待，而針對錄放功能時的命中率也提出對 Directstream 的比較，結果發現可以有平均高於 1.46 倍的表現，且在影片播放初期的時候可以有更明顯的差異。

## 參考文獻

- [1] V. Rangan, H. Vin and S. Ramanathan, "Designing an on-demand multimedia service", IEEE Communications Magazine, vol. 30, no. 7, pp.56-65, Jul 1992.
- [2] ppstream, <http://www.pppstream.com>
- [3] GnuStream, <http://gnustream.com/>
- [4] Peercast, <http://www.peercast.org/>
- [5] L-S. Juhn and L-M. Tseng, "Fast broadcasting for hot video access", RTCSA'97: the proceedings of the 4th international workshop on real-time computing systems and applications, pp.237-243, Oct 1997.
- [6] L-S. Juhn and L-M. Tseng, "Fast data broadcasting and receiving scheme for popular video service", IEEE Trans. Broadcasting, vol. 44, no. 1, pp. 100-105, Mar 1998.
- [7] Li-Shen Juhn and Li-Ming Tseng, "Harmonic Broadcasting for Video-on-Demand Service," IEEE TRANSACTIONS ON BROADCASTING, Sept 1997
- [8] Huang-Chang Yang, Hsiang-Fu Yu, and Li-Ming Tseng, "Adaptive Live Broadcasting for Highly-Demanded Videos," Journal of Information Science and Engineering, vol. 19, no. 3, pp. 531-549, May 2003
- [9] Huang-Chang Yang, Hsiang-Fu Yu, Li-Ming Tseng, and Yi-Ming Chen, "Fast and Lossless Broadcasting Schemes for VBR-encoded Hot Videos," The 2004 International Conference on Software, Telecommunications and Computer Networks (SOFTCOM2004), Venice, Italy, Oct. 2004.
- [10] K.H. Ho, W.F. Poon and K.T. Lo, "Peer-to-Peer Broadcasting Scheme for Video-on-Demand System" International Symposium on Communications and Information Technologies Oct. 2006
- [11] Minjung Kim and John A. Copeland, "Bandwidth sensitive caching for video streaming application" IEEE International Conference on Communications 2003
- [12] Kui Gao; Yuan Zhang; Wen Gao; Simin He, "Real-Time Scheduling Supporting VCR Functionality For Scalable Video Streaming" IEEE International Symposium on Personal, indoor and Mobile Radio Communication Proceedings (PIMRC) 2003
- [13] Jiangchuan Liu, Xiaowen Chu, Jianliang Xu, "Proxy cache management for fine-grained scalable video streaming" IEEE INFOCOM 2004
- [14] Susie J. Wee, Bhaskaran Vasudev, "Compressed-Domain Reverse Play of MPEG Video Streams" SPIE International Symposium on Voice, Video, and Data Communications, Boston, MA, Nov, 1998.
- [15] Bo Shen, Sung-Ju Lee, and Sujoy Basu "Caching strategies in transcoding-enabled proxy systems for streaming media distribution networks" IEEE TRANSACTIONS ON MULTIMEDIA, VOL. 6, NO. 2, APRIL 2004
- [16] Lei Guo, Songqing Chen, Zhen Xiao, Xiaodong Zhang, "Disc: Dynamic interleaved segment caching for interactive streaming" IEEE International Conference on Distributed Computing Systems(ICDCS) 2005
- [17] James Z. Wang, Philip S. Yu, "Fragmental Proxy Caching for Streaming Multimedia Objects" IEEE TRANSACTIONS ON MULTIMEDIA, JAN 2007
- [18] E. Zegura, K. Calvert, and S. Bhattacharjee, "How to model an internetwork," in INFOCOM96, San Francisco, CA, 1996.