

# 平滑化快速廣播演算法

游象甫<sup>ab</sup> 楊宏昌<sup>a</sup> 陳奕明<sup>c</sup> 曾黎明<sup>a</sup> 郭鎮溢<sup>a</sup>

<sup>a</sup> 國立中央大學資訊工程研究所

<sup>b</sup> 國立中央大學電算中心

<sup>c</sup> 國立中央大學資管所

{yu, cyt, kuojy}@dslab.csie.ncu.edu.tw

## 摘要

隨著網路及電腦科技的進步，隨選視訊 (Video on Demand, VoD) 服務變得愈來愈普遍。為了降低頻寬與磁碟傳輸的需求，有學者提出一種稱為熱門影片廣播法的資料傳送方式，其伺服器播送影片所需頻寬與使用者人數無關，所以適合用於播送多人收看的熱門影片。過去的相關研究都是假設影片是固定位元率 (Constant Bit Rate, CBR) 編碼，所以無法支援變動位元率 (Variable Bit Rate, VBR) 編碼的影片。然而根據研究，相同品質的影片，CBR 的平均位元率比 VBR 的平均位元率多兩倍以上，所以採取 VBR 編碼可以改善系統效能。本文提出一個支援 VBR 影片的快速廣播法 (Fast Broadcasting)，稱為平滑化快速廣播法 (Smooth Fast Broadcasting)，可減緩所需頻寬及使用端 buffer 起伏，同時保證播放的連續性。對一部影片來說，假設  $B_{\max}^i$  和  $B_{\min}^i$  代表其在串流  $i$  ( $1 \leq i \leq K$ ， $K$  表示所有的串流數) 所需最大及最小的頻寬，則本方法保證其在播放過程所需最大及最小的頻寬差不大於  $\max(B_{\max}^i - B_{\min}^i)$ 。

**關鍵字：**熱門影片廣播法、隨選視訊、VBR

## 1. 緒論

隨選視訊 (Video on Demand, VoD) 是讓使用者透過網路來觀看的影片的一種服務。由於網路科技的進步，此服務變得愈來愈普遍。隨選視訊系統主要是採取主從式架構 (Client-Server Architecture)，在最早的系統中使用者獨自享有伺服器提供的影片頻寬，這種架構會隨使用者的增加而增加網路頻寬的需求，所以用於熱門影片的傳送很容易因為使用者的增加而造成頻寬不敷使用。

為了降低頻寬與磁碟傳輸的需求，許多方法被提出，有的方法是犧牲 VCR 的功能 (快轉或快速搜尋)，有的是建構近似 VOD 的系統。根據 [3] 的研究，80% 的使用者集中在少量 (10 部或 20 部) 較熱門的影片。於是有學者提出一種稱為熱門影片廣播法的播送方式，因為伺服器播送影片所用的頻寬與使用者人數無關，所以這個方法適合用於播送多人收看的熱門影片。其中的一種廣播法是將熱門影片分割成一段段的「區段」(segment)，然後使用若干個串流 (stream) 週期性的播送。這類方法 [1][5][6][7][8][13][20][21][23][24] 有著相似的準備

動作，就是影片伺服器會先將影片切割成數個區段，並且同時在不同的資料串流上播送它們，其中的一個串流週期的播送第一個區段，而其他的串流則根據不同廣播演算法所設定的排程，週期的播送剩餘區段。當使用者想要收看影片時，使用者必須等待最接近的第一個區段之起始點，因此最大的等待時間就等於第一個區段的時間長度。當使用者開始收看影片，使用者端的隨選視訊轉換器 (Set-Top Box, STB) 或電腦就會開始從其他的資料串流下載足夠的資料，而讓使用者端能夠依序的播放下一個區段。

上述的方法都是假設影片是固定位元率 (Constant Bit Rate, CBR) 編碼，所以無法支援變動位元率 (Variable Bit Rate, VBR) 編碼的影片。CBR 編碼為了固定編碼後的位元率，有時必須將影片品質降級；而 VBR 編碼會隨著影片畫面的複雜或簡單，使的壓縮編碼後位元率會高低起伏。而且據 [2][19] 的研究發現，對相同品質畫面的影片，CBR 的平均位元率比 VBR 的平均位元率多兩倍以上，所以採取 VBR 編碼可以改善系統效能，於是有學者提出了一些可支援 VBR 影片的方法。Pàris 提出了支援 VBR 影片的可變頻寬和諧廣播法 (Variable Bandwidth Harmonic Broadcasting, VBHB) [14]，這個方法是先將影片切割成處理過大小的區段，然後使用能保證第一個區段及第二個區段能即時傳送到使用者端的頻寬傳送第一個區段及第二個區段，其他的區段就依照謹慎和諧廣播法 (Cautious Harmonic Broadcasting, CHB) [12] 的規則再切割成若干個相同大小的子區段。D. Saporilla [18] 提出了一個整合 GoP smoothing、server buffering 及使用端先取技巧 [16][17] 的遞增區段廣播法來傳送 VBR 編碼的影片，稱為支援 VBR 影片的週期廣播法 (Periodic Broadcasting with VBR-encoded Video, VBR-B)。F. Li 提出一個探出適合的切割法的方法 (Trace Adaptive Fragmentation, TAF) [10]，這個方法參考 VBR-B 數學分析去推論對於不同網路環境及每一部影片可能的切割播送方式，並且依照最大等待時間以及播放的連續的限制找出可行的切割方式，最後從可行的切割方式找出封包遺失量最小一個的為最佳切割方式。

本文提出一個支援 VBR 影片的快速廣播法 (Fast Broadcasting)，稱為平滑化快速廣播法 (Smooth Fast Broadcasting)。這個方法先將 VBR 影片依據時間切割成數個等時間長的區段。為使所需要的頻寬變化較為平滑，首先把區段內的資料量平

均化用固定的頻寬傳播送，其次再依大小關係調換每個串流內播送區段的順序，以達到總需要頻寬的平滑化。我們可以保證播送一部VBR影片所需最大及最小的頻寬差不大於  $\max(B_{\max}^i - B_{\min}^i)$ ，其  $B_{\max}^i$  和  $B_{\min}^i$  代表其在串流  $i$  ( $1 \leq i \leq K$ ,  $K$  表示所有的串流數)所需最大及最小的頻寬。同時，本法也可用於多部VBR影片的播送，減小其頻寬起伏。

本文的其餘架構如下：第二節為相關研究；第三節討論平滑化快速廣播法；第四節分析提出的演算法並以實際影片評估其效能；第五節為結論。

## 2. 相關研究

本節回顧有關熱門影片廣播法的研究，主要是討論如何在一個固定頻寬下，讓使用者等待時間減少到最短，而討論的影片都是以 CBR 編碼的影片為主，根據 stream tapping 的作者 Steven W. Carter 的研究[4]，可以將這些方法為二類：批次廣播法 (Batching broadcasting) 跟區段式廣播法 (Segment based broadcasting)。但是在以區段為主的熱門影片播放演算法 (Segment Based Hot Video Broadcasting) 中卻鮮少深入討論有關播送 VBR 影片的研究。以下是我們對這類演算法的說明。

利用群播 (Multicast) 中有相同影片需求的使用者為一個群組 (Group) 的觀念，可以每隔一段時間分批將影片播出。但在有限的頻寬下，若是該影片為熱門影片且萬一頻寬不足，下一批次的使用者也只能夠等待一定時間後再重新加入群組觀看。

各種研究嘗試在頻寬和等待時間中取得一個平衡。這些區段式廣播式演算法具有下面特點：

1. 須把影片做片段的切割。
2. 當影片已經開始播放的時候，允許使用者動態加入熱門電影的群播頻寬。
3. 使用者端必須能夠同時接收超過一個以上的虛擬通道串流影音資料，可能為廣播或群播串流。
4. 使用者端必須使用硬碟儲存影音資料，並作適當的重整與播放。

相關方法有和諧廣播法 (Harmonic Broadcasting) [5]、快速廣播法 (Fast Broadcasting) [6]、新巴格達廣播法 (New Pagoda Broadcasting) [13]及遞迴頻率分割廣播策略 (RFS) [20]。

快速廣播演算法[6]是將一部影片切成  $N$  個相等的區段 (Segment)，分別為  $S_1, S_2, \dots$  和  $S_N$ 。我們假設區段  $2^{i-1}$  到  $2^i - 1$  為一個區段群，稱為  $G_i$ ，如圖 1 (a) 所示。最後我們把串流 (Stream)  $C_i$  播放區段群  $G_i$ 。串流  $C_i$  的頻寬等於影片的播放率  $b$ 。使用者端的最大等待時間等於區段的時間

長， $\frac{L}{N}$ 。所需要的總頻寬為  $\lceil \log_2(N+1) \rceil \times b$ 。

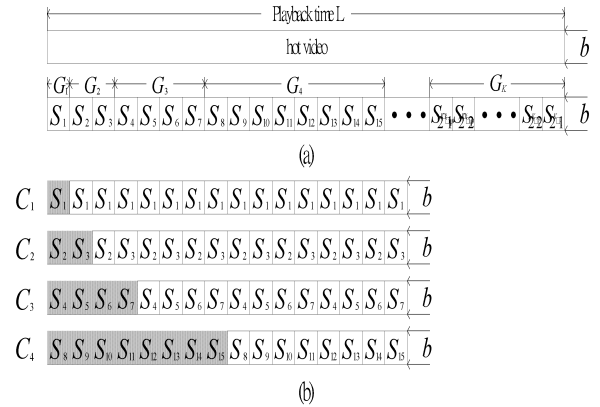


圖 1 (a)快速廣播法的切割方式(b)15 個區段的播送方式

如圖 1 (b) 所示將影片切成 15 個區段，則可分成 4 個區段群，所以伺服器端使用 4 個串流來播放。第一個串流週期性的重複播放  $S_1$ ，第二個串流週期性重複播放  $S_2$  和  $S_3$ ，第三個串流週期性重複播放  $S_4 \dots S_7$ ，第四個串流週期性重複播放  $S_8 \dots S_{15}$ 。因此可看出最長等待時間為影片長度的十五分之一。

## 3. 平滑化快速廣播演算法 (Smooth Fast Broadcasting, SFB)

### 3.1 背景

過去的廣播演算法研究大多針對 CBR 編碼的影片，如果將 VBR 影片切成數個區段，直接播送，在伺服器端可能因為磁碟傳輸率或頻寬需求不足而隨時中斷服務；在使用者端也可能因為網路不滿足所需頻寬的最高峰而無法取得所需要的影片資料。為了解決上述問題，我們提出支援 VBR 影片的平滑化快速廣播演算法 (Smooth Fast Broadcasting, SFB)，主要是根據 FB 做改進，可支援 VBR 編碼影片的播送，確保播放的連續性，並減少使用者端 Buffer 需求。更重要的是 SFB 可以減小所需頻寬及其起伏，避免頻寬浪費。

SFB 與 FB 主要有兩個不同點：

1. 下載與播放不同步。因為 VBR 編碼影片的資料消耗率會隨著時間改變，消耗的資料量可能大於已接收的資料量。當消耗的資料量大於已接收的資料量時，影片的播放就會不連續或中斷。為了使播放可以連續，SFB 要求使用者需將一個區段完全的收齊後才能播放。這個條件會使 SFB 的平均等待時間大於 FB 的平均等待時間，但兩個演算法的最大等待時間還是相等。

- 頻寬需求變動小。SFB 計算每個區段的平均位元率，而每個區段以此速率傳輸，因此對每個區段而言，所需的頻寬變異量為零。但 FB 以變動速率傳送區段，所以頻寬變異量必然大於零；此外，SFB 以每個串流為考量，掉換串流內傳送的區段順序，進一步減小所需頻寬的變異量，並可估計變異量的上限。

### 3.2 平滑化快速廣播演算法

```

#K is the number of streams.
#S[1..2^K-1] stores the size of the segments. If a segment is blank, its size equals 0.
#B[1..2^K-1] stores the bandwidth consumption of the time slots.
#V[1..2^K-1] stores the new arrangement of video segments
B[1..2^K-1] = 0
B[1] = S[1]
V[1] = 1
If K = 1 then return
For i = 2 to K do
  copy(B[], 1, 2^{i-2}, 2^{i-1} + 1)
  # copy B[1..2^{i-2}] to B[2^{i-2} + 1..2^{i-1}], for example, suppose B[1..4] = {8,3,0,0}.
  # After copy(B[], 1, 2, 3), B[1..4] = {8,3,8,3}
  sort_value_but_return_key(B[], 1, 2^{i-1}, C[])
  # sort B[1..2^{i-1}] and save the keys to C[1..2^{i-1}], for example, B[1..4] = {8,3,8,3}
  # then C[1..4] = {2,4,1,3}
  sort_value_but_return_key(S[], 2^{i-1}, 2^i - 1, T[])
  # sort S[2^{i-1}..2^i - 1] and save the keys to T[], for example, suppose S[4..7] = {8,5,4,9}.
  # Then, T[1..4] = {6,5,4,7}
  For j = 1 to 2^{i-1} do
    B[C[j]] = B[C[j]] + S[T[2^{i-1} + 1 - j]]
    V[2^{i-1} + j - 1] = T[2^{i-1} + 1 - j]
  End for
End for
End for

```

圖 2 SFB 的演算法

依據圖 2 的演算法，SFB 在伺服器端有下列幾個步驟：

- 將一部 VBR 影片依相等的時間長分割成  $N$  個區段，所需的串流個數為  $\lceil \log_2(N+1) \rceil$ 。
- 運用區段安排演算法，如圖 2，重新排列每個串流區段的順序。將一部 VBR 的影片分割成 15 個相等時間長但是不同大小的區段，如圖 3 (a) 所示。假設區段的大小分別為 3、4、9、5、3、2、7、5、4、7、2、5、7、5 和 3。如同快速演算法，這些區段可以分成四個區段群，並且使用四個串流來播送。首先考慮第一個串流，在第一個串流所需要的頻寬是固定的，因為第一個串流週期性的傳送  $S_1$ 。其次，我們考慮第二個串流，將第二個串流內的區段由大到小排入，即  $S_3$  和  $S_2$ ，如圖 3 (b) 所示。因為  $S_3$  和  $S_2$  的大小不同，我們得到前兩個串流所需要的頻寬為 12 和 7。再來考慮第三個串流，為了使頻寬起伏較平

順，先將第三個串流內最大的區段排入前兩個串流所需頻寬最小的位置；再將第三個串流內次大的區段排入前兩個串流所需頻寬次小的位置；重複動作直到第三個串流內的區段排完，如圖 3 (c) 所示。同理，我們可將第四個串流做適當的排序。最後我們可以得到播送順序如圖 3 (d) 所示。

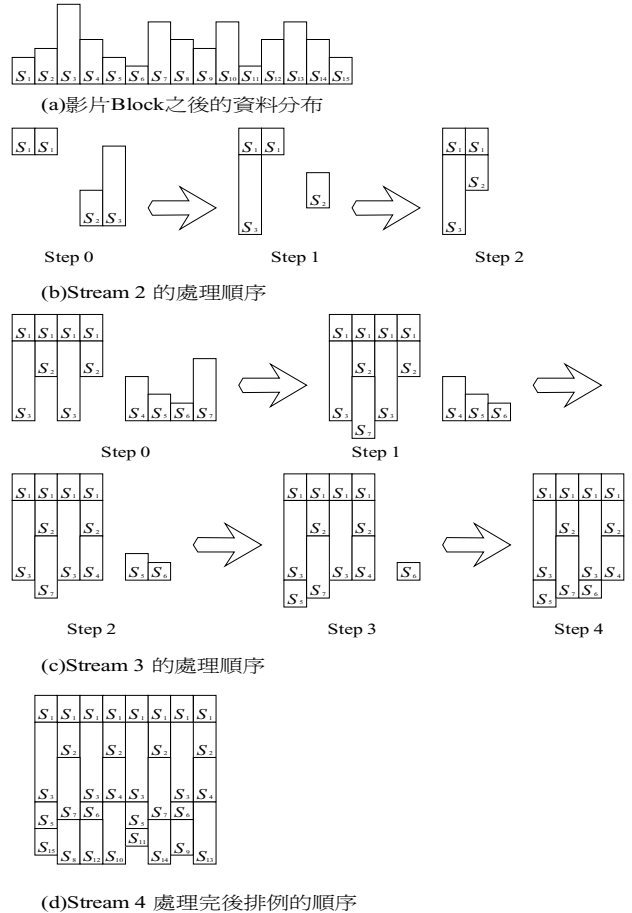


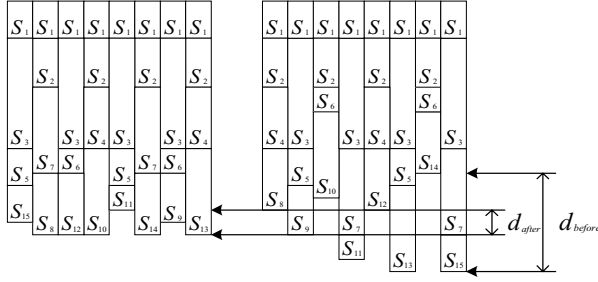
圖 3 SFB 中影片區段排列的步驟

- 伺服器依據排列後的順序週期性的傳送區段。

假設使用者端有足夠的空間儲存播放影片的資料。收看影片有下列幾個步驟：

- 在每個時間區間同時下載所需的區段。
- 為了確保區段在使用前能完整的存入 Buffer，我們延遲一段時間播放。如果使用者端在時間  $T_0$  開始下載，則影片可在時間為  $T_0 + \frac{L}{N}$  的時候開始播放，其中  $L$  為影片時間長， $N$  為影片切個的區段總數。
- 當使用者端取得全部的區段後就停止從網路下載資料。

### 3.3 頻寬起伏上限分析



排列前後頻寬起伏之比較

圖 4 使用 SFB 重新列前後的比較圖

如圖 4 所示，我們可發現重新排列前與排列後的起伏明顯改善很多。SFB 可以保證重新排列後的頻寬起伏會小於或等於  $\max(B_{\max}^i - B_{\min}^i)$ ，其中

$B_{\max}^i$  與  $B_{\min}^i$  分別表示第  $i$  個串流所需要的最大頻寬與最小頻寬，且  $1 \leq i \leq K$ ， $K$  表示所有的串流數。以下是證明，先證明一個輔助定理：

*Lemma 1*：假設有兩個已排序過的數列

$X = \{x_1, x_2, \dots, x_n\}$  和  $Y = \{y_1, y_2, \dots, y_n\}$ ，其中  $0 \leq x_1 \leq x_2 \leq \dots \leq x_n$  和  $0 \leq y_1 \leq y_2 \leq \dots \leq y_n$ 。

令一個新的數列  $Z = \{z_1, z_2, \dots, z_n\}$ ，其中

$z_i = x_i + y_{n-i+1}$ ，而且  $1 \leq i \leq n$ 。則我們要證明

$z_{\max} - z_{\min} \leq \max(x_n - x_1, y_n - y_1)$ ，其中  $z_{\max}$  與  $z_{\min}$  為數列  $Z$  中的最大值與最小值。

證明：從數列  $Z$  中任取兩個數  $z_j$  與  $z_k$ 。由假設得知  $z_j = x_j + y_{n-j+1}$  跟  $z_k = x_k + y_{n-k+1}$ 。所以可以推導得到  $z_j - z_k = (x_j - x_k) + (y_{n-j+1} - y_{n-k+1})$ 。

當  $j \geq k$  時，因為  $0 \leq x_1 \leq x_2 \leq \dots \leq x_n$ ，所以我們可以推論得到

$$x_n - x_1 \geq x_j - x_k \geq 0 \quad (1)$$

因為  $j \geq k$ ，所以兩邊變號可得  $-j \leq -k$ ，在再兩邊同時加上  $n+1$ ，則可以得到  $n-k+1 \geq n-j+1$ ，

又因為  $0 \leq y_1 \leq y_2 \leq \dots \leq y_n$ ，所以我們可以推論得到

$$0 \geq y_{n-k+1} - y_{n-j+1} \geq y_1 - y_n \quad (2)$$

則由式子 (1) 和 (2) 可以推論  $x_n - x_1 \geq x_j - x_k + y_{n-j+1} - y_{n-k+1} \geq y_1 - y_n$

也就是說  $x_n - x_1 \geq z_j - z_k \geq y_1 - y_n$

即證明  $\max(x_n - x_1, y_n - y_1) \geq |z_j - z_k|$ ；

當  $j < k$  時，因為  $0 \leq x_1 \leq x_2 \leq \dots \leq x_n$ ，所以我們可以推論得到

$$x_n - x_1 \geq x_k - x_j \geq 0 \quad (3)$$

因為  $j < k$ ，所以兩邊變號可得  $-j > -k$ ，在再兩邊同時加上  $n+1$ ，則可以得到  $n-k+1 < n-j+1$ ，

又因為  $0 \leq y_1 \leq y_2 \leq \dots \leq y_n$ ，所以可以得到

$$0 \geq y_{n-k+1} - y_{n-j+1} \geq y_1 - y_n \quad (4)$$

則由式子 (3) 和 (4) 可以推論， $x_n - x_1 \geq x_k - x_j + y_{n-k+1} - y_{n-j+1} \geq y_1 - y_n$

也就是  $x_n - x_1 \geq z_k - z_j \geq y_1 - y_n$

即證明  $\max(x_n - x_1, y_n - y_1) \geq |z_k - z_j|$ 。□

由上述的輔助定理證明，我們可以得到下面定理：

*Theorem 1*：使用 SFB 播送 VBR 影片所需頻寬的最大起伏會小於或等於  $\max(B_{\max}^i - B_{\min}^i)$ ，其中

$B_{\max}^i$  與  $B_{\min}^i$  分別表示第  $i$  個串流所需要的最大頻寬與最小頻寬， $1 \leq i \leq K$ ， $K$  表示所有的串流數。

證明：假設  $d_i$  是 SFB 播放第  $i$  個串流所需頻寬的最大起伏，其中  $1 \leq i \leq K$ 。

則由輔助定理可推論：

$$d_1 = 0$$

$$d_2 = \max(d_1, B_{\max}^2 - B_{\min}^2)$$

$$d_3 = \max(d_2, B_{\max}^3 - B_{\min}^3)$$

⋮

$$d_{K-1} = \max(d_{K-2}, B_{\max}^{K-1} - B_{\min}^{K-1})$$

$$d_K = \max(d_{K-1}, B_{\max}^K - B_{\min}^K)$$

綜合上述方程式，可以推導出

$d_K = \max(B_{\max}^i - B_{\min}^i)$ ，其中  $1 \leq i \leq K$ ，故得證。□

#### 4. 分析與數據模擬

我們以侏儸紀公園三及星際大戰二部曲作為模擬分析的影片，影片時間總長分別為 4800 秒與 8400 秒，影片大小分別為 2.66GB 與 5.77GB，其 VBR 編碼的位元率分布圖如圖 5 及圖 6 所示，很明顯的起伏非常大。利用上述影片本文分析使用者等待時間、所需的 Buffer 大小及所需磁碟傳輸率。

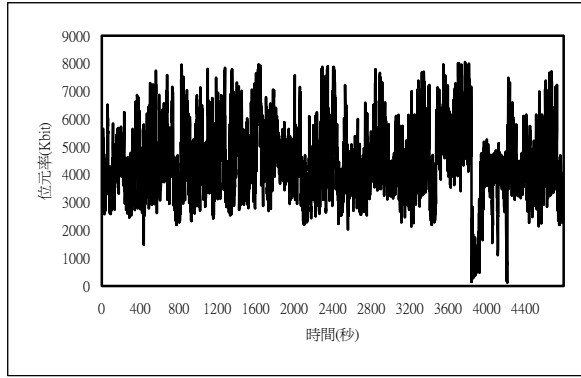


圖 5 侏羅紀公園三用 VBR 編碼的位元分布圖

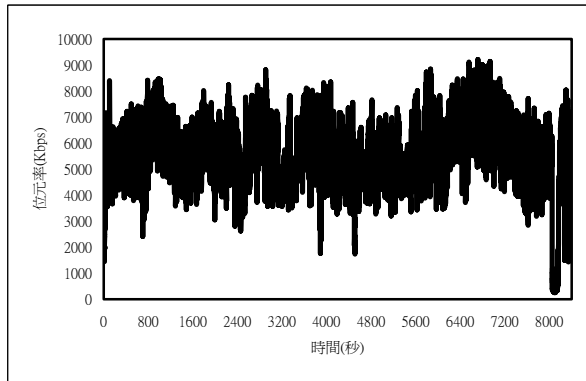


圖 6 星際大戰二部曲用 VBR 編碼的位元分布圖

#### 4.1 使用者等待時間分析

假設使用者端有足夠的空間存放要播放的影片，為了確保影片可以連續播放，影片區段從網路上存入 Buffer 所需要的時間不可以大於區段的時間長，所以使用者最大的等待時間就是一個區段的時間長。

假設將一部影片切割成  $N$  個區段，則每個區段時間長（最大的等待時間）為

$$\delta = \frac{L}{N} \quad (5)$$

還可以推算所需要的串流個數為

$$K = \lceil \log_2(N+1) \rceil \quad (6)$$

假設用 SFB 轉換後的區段播放順序為  $X = \{X_j\}$ ， $1 \leq j \leq 2^K - 1$ （其中當  $j \geq N$  時， $X_j$  的大小為零），則第  $i$  個串流週期性的播送  $X_{2^{i-1}}$  到  $X_{2^i-1}$ ，其中  $1 \leq i \leq K$ 。在時間  $T_0 + (p-1)\delta$  到  $T_0 + p\delta$  之間，伺服器端會傳送區段  $X_{2^{q-1} + ((p-1) \bmod 2^{q-1})}$ ，其中  $1 \leq p \leq 2^{K-1}$  和  $1 \leq q \leq K$ 。假設  $X_j$  的大小為  $x_j$ ，則一段時間所要傳送的總資料量為

$$D_p = \sum_{q=1}^K x_{2^{q-1} + ((p-1) \bmod 2^{q-1})} \quad (7)$$

由式子(5)和(7)可推導所需要的頻寬為

$$B_p = \frac{1}{\delta} \sum_{q=1}^K x_{2^{q-1} + ((p-1) \bmod 2^{q-1})} = \frac{2^K - 1}{L} \sum_{q=1}^K x_{2^{q-1} + ((p-1) \bmod 2^{q-1})} \quad (8)$$

若給定一頻寬大小為  $B$ ，假設時間  $T_0 + (p-1)\delta$  到  $T_0 + p\delta$  間下載影片區段真正所需時間為  $\delta_{p,B}$ ，則由(8)可知，

$\delta_{p,B} = \frac{1}{B} \sum_{q=1}^K x_{2^{q-1} + ((p-1) \bmod 2^{q-1})}$ 。因為 VBR 影片切割成的區段大小不一，所以存取時間  $\delta_{p,B}$  在不同的時間區間會不一樣。所以最大的等待時間  $\delta_{\max,B}$  就是  $\delta_{p,B}$  的最大值，其中  $1 \leq p \leq 2^{K-1}$ ，而且  $\delta_{\max,B} \leq \delta$ ，否則播放會不連續。

$$\delta_{\max,B} = \max(\delta_{p,B})$$

其中  $\delta_{p,B} = \frac{1}{B} \sum_{q=1}^K x_{2^{q-1} + ((p-1) \bmod 2^{q-1})}$  和

$$1 \leq p \leq 2^{K-1}。$$

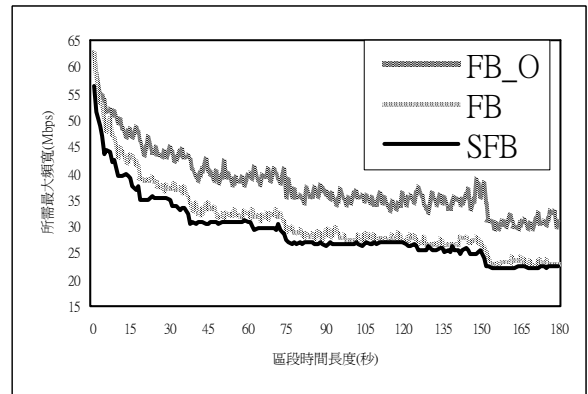


圖 7 侏羅紀公園三使用 SFB 排列前後所需要的最大頻寬

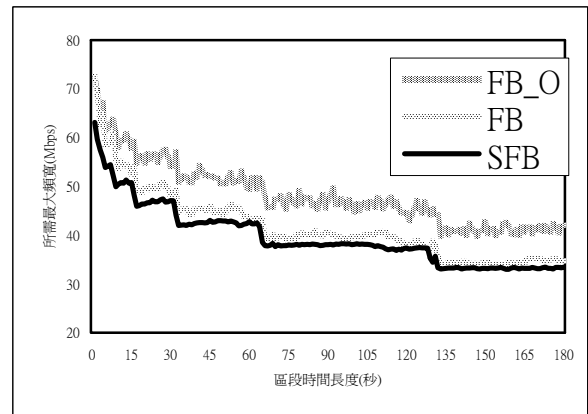


圖 8 星際大戰二部曲使用 SFB 排列前後所需要的最大頻寬

如圖 7、圖 8 所示，SFB 所需最大頻寬在不同的區段時間長度下都比 FB 所需最大頻寬小(FB\_O 曲線代表 FB 完全沒修改時所需最大頻寬，FB 曲線代表 FB 播放時其影片區段已作平均化)。

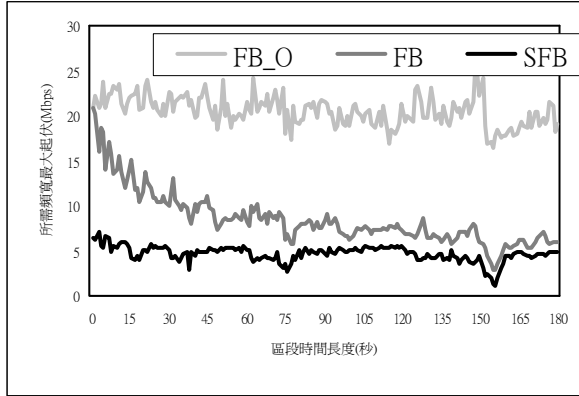


圖 9 侏羅紀公園三使用 SFB 排列前後所需要頻寬的最大起伏

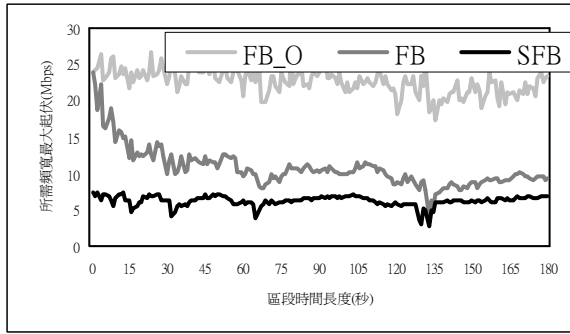


圖 10 星際大戰二部曲使用 SFB 排列前後所需要頻寬的最大起伏

如圖 9、圖 10 所示，SFB 頻寬的最大起伏都較 SB 有所改善，如區段時間長度越短則改善的情況越好。

#### 4.2 所需 Buffer 分析

因為影片區段資料的消耗率小於資料的接收率，所以使用者端必須將部份影片區段先暫時存入磁碟中。假設使用者到達時間為  $T_0 + u\delta$ ，則在  $T_0 + u\delta + (p-1)\delta$  到  $T_0 + u\delta + p\delta$  間，使用者會從串流  $C_{\lceil \log_2 p \rceil + 1}, C_{\lceil \log_2 p \rceil + 2}, \dots, C_K$  下載所需要的區段資料並且存入磁碟，所存入的資料量為

$$I_p^u = \sum_{q=\lceil \log_2 p \rceil + 1}^K x_{2^{q-1} + (u+p-1) \bmod 2^{q-1}}, \quad \text{其中}$$

$$0 \leq u \leq 2^{K-1}, \quad 1 \leq p \leq 2^{K-1}$$

$$I_p^u = 0, \quad \text{其中 } 0 \leq u \leq 2^{K-1}, \quad 2^{K-1} + 1 \leq p$$

(9)

因為使用者端不能邊收邊播放，所以在同一段時間裡，使用者端會把先前存入磁碟的資料取出來

播放，而從磁碟取出的資料量為

$$O_1^u = 0$$

$$O_p^u = s_{p-1}, \quad \text{其中 } 0 \leq u \leq 2^{K-1} - 1,$$

$$2 \leq p \leq 2^K \quad (10)$$

假設  $Z_p^u$  是在時間  $T_0 + u\delta + (p-1)\delta$  到  $T_0 + u\delta + p\delta$  間所需的 Buffer 大小，則由式子 (9)、(10) 可以推導出在時間  $T_0 + u\delta$  到  $T_0 + u\delta + \delta$  間， $Z_1^u = I_1^u$ ；其他時間所需要的 Buffer 大小為

$$Z_p^u = Z_{p-1}^u + I_p^u - O_p^u, \quad \text{其中 } 0 \leq u \leq 2^{K-1} - 1,$$

$$2 \leq p \leq 2^K - 1 \quad (11)$$

在時間  $T_0 + u\delta + (2^K - 1)\delta$  到  $T_0 + u\delta + 2^K\delta$  間，使用者已經停止下載影片區段的資料，而且磁碟中的資料也會在這一段時間內消耗完，所以  $Z_{2^K}^u = 0$ 。

根據(11)，在已知的串流個數  $K$  下我們可以計算出所需要的 Buffer 大小  $\{Z_1^0, Z_2^0, \dots, Z_{2^{K-1}}^0, \dots, Z_{2^K}^0\}$ 。根據式子(5)及

(6)， $K = \lceil \log_2(\frac{L}{\delta} + 1) \rceil$ ，所以我們可以推論出所需最大 Buffer，

$$\max \{Z_p^u \mid u = 0, 1, \dots, 2^{\lceil \log_2(\frac{L}{\delta} + 1) \rceil} - 1;$$

$$p = 1, 2, \dots, 2^{\lceil \log_2(\frac{L}{\delta} + 1) \rceil} - 1\}$$

與區段時間長  $\delta$  的關係。

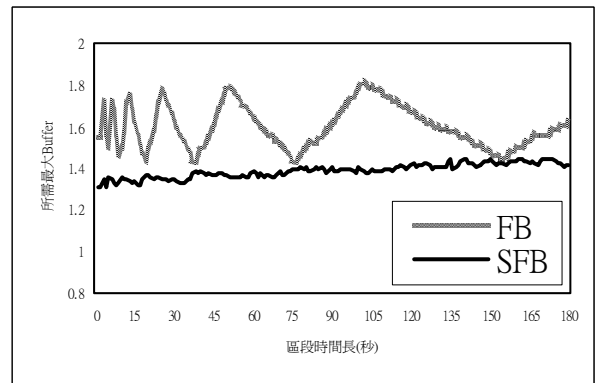


圖 11 侏羅紀公園三使用 SFB 排列前後所需要的最大 Buffer

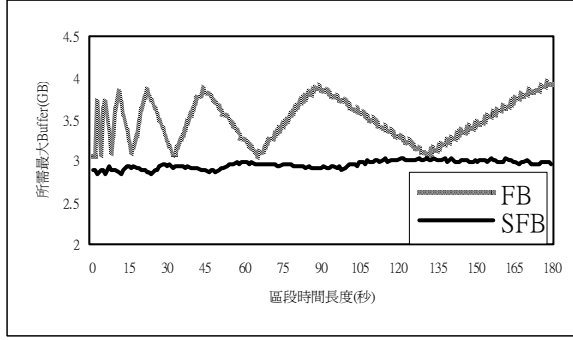


圖 12 星際大戰二部曲使用 SFB 排列前後所需要的最大 Buffer

如圖 11、圖 12 所示，SFB 所需最大 Buffer 在不同的區段時間長度下都比 FB 所需最大 Buffer 小，且在不同區段時間長度下 Buffer 的需求量較穩定。

#### 4.3 所需磁碟傳輸率分析

使用者端的磁碟傳輸率包含寫入資料及讀出資料的頻寬。由式子(9)，我們知道在時間  $T_0 + u\delta + (p-1)\delta$  到  $T_0 + u\delta + p\delta$  之間所要存入的影片區段資料量，所以可以得知要寫入的頻寬為

$$W_p^u = \frac{1}{\delta} \sum_{q=\lceil \log_2 p \rceil + 1}^K x_{2^{q-1} + (u+p-1) \bmod 2^{q-1}}, \quad \text{其中}$$

$$0 \leq u \leq 2^{K-1} - 1, \quad 1 \leq p \leq 2^{K-1}$$

$$W_p^u = 0, \quad \text{其中 } 0 \leq u \leq 2^{K-1} - 1, \quad 2^{K-1} + 1 \leq p \quad (12)$$

同一時間，讀出的頻寬等於影片資料的消耗率。因為 VBR 編碼的影片的消耗率會隨時間不同而不同，為了簡化情況我們只考慮每個區段的最大影片消耗率，令  $b_{S_i}$  為區段  $S_i$  的最大影片消耗率。因為使用者端要下載完整的區段後才開始播放影片，所以在時間  $T_0 + u\delta$  到  $T_0 + u\delta + \delta$  的讀出頻寬為零，即  $R_1^u = 0$ ；其他的讀出頻寬為

$$R_p^u = b_{S_{i-1}}, \quad \text{其中 } 0 \leq u \leq 2^{K-1} - 1, \quad 2 \leq p \leq 2^K \quad (13)$$

假設  $\Phi_p^u$  是在時間  $T_0 + u\delta + (p-1)\delta$  到  $T_0 + u\delta + p\delta$  間所需的最大磁碟傳輸率，由式子(12)、(13)我們可以把最大磁碟傳輸率寫成

$$\Phi_p^u = W_p^u + R_p^u, \quad \text{其中 } 0 \leq u \leq 2^{K-1} - 1, \quad 1 \leq p \leq 2^K$$

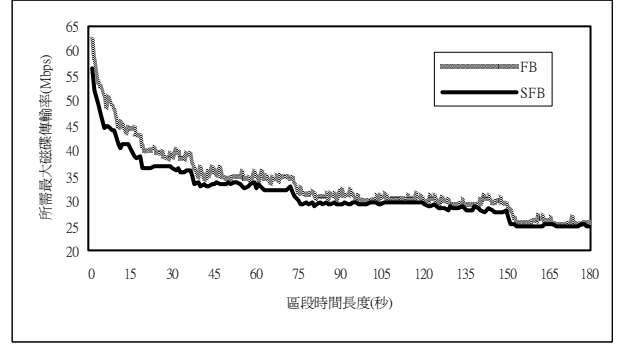


圖 13 侏羅紀公園三使用 SFB 排列前後所需要的最大磁碟傳輸率

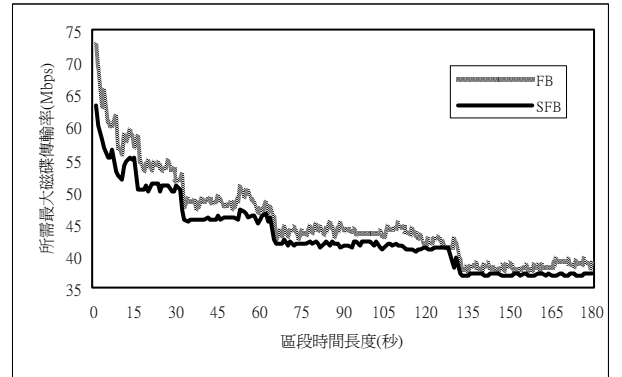


圖 14 星際大戰二部曲使用 SFB 排列前後所需要的最大磁碟傳輸率

圖 13、圖 14 表示使用者端所需最大磁碟傳輸率與區段長度的關係，SFB 所需最大磁碟傳輸率在不同的區段時間長度下都比 FB 所需最大磁碟傳輸率小。

## 5. 結論

視訊廣播服務在網路上已經是相當普遍，本文提出一個支援 VBR 編碼影片的廣播法，稱為平滑化快速廣播法(SFB)。這方法改良快速廣播法，除可以確保 VBR 影片播放的連續性，還可以減小頻寬起伏。我們進一步導出其頻寬起伏上限，並以數學分析使用者等待時間、Buffer 使用量及所需磁碟傳輸速率。

我們以實際 VBR 編碼影片做評估，結果顯示 SFB 在頻寬需求量、Buffer 需求量及磁碟傳輸速率都小於 FB。尤其是 Buffer 的需求量在不同時間長的區段切割下僅會平穩增加，不會像 FB 起伏過大。未來我們計畫研究支援即時性 VBR 影片的廣播法，同時繼續改良 SFB，進一步減小頻寬的起伏。

## 誌謝

作者特別感謝國家科學委員會經費贊助此一研究，合約編號為 NSC 92-2213-E-008-004。

## 參考文獻

- [1] K. C. Almeroth and M. H. Ammar, "The use of multicast delivery to provide a scalable and interactive video-on-demand service," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 5, pp. 1110-1122, Aug 1996.
- [2] I. Dalgic and F. A. Tobagi, "Characterization of quality and traffic for various video encoding schemes and various encoder control schemes," TechRep. CSL-TR-96-701, Departments of Electrical Engineering and Computer Science, Stanford University, August 1996.
- [3] Asit Dan, Dinkar Sitaram, Perwez Shahabuddin, "Dynamic batching policies for an on-demand video server," *Multimedia Systems*, vol. 4, no. 3, pp. 112-121, June 1996.
- [4] Darrell D. E. and Steven W. Carter, "Stream tapping : a system for improving efficiency on a video on demand server," Technical Report UCSC-CRL-97-11, University of California, 1997
- [5] L.-S. Juhn and L.-M. Tseng, "Harmonic broadcasting for video-on-demand service," *IEEE Transactions on Broadcasting*, vol. 43, no. 3, pp. 268-271, September 1997.
- [6] L.-S. Juhn and L.-M. Tseng, "Fast data broadcasting and receiving scheme for popular video services," *IEEE Transactions on Broadcasting*, vol. 44, no. 1, pp. 100-105, March 1998.
- [7] L.-S. Juhn, and L.-M. Tseng, "Enhanced harmonic data broadcasting and receiving scheme for popular video service," *IEEE Transactions on Computer Electronics*, vol. 44, no. 2, pp. 343-346, May 1998.
- [8] L.-S. Juhn, and L.-M. Tseng, "Adaptive fast data broadcasting scheme for video-on-demand services," *IEEE Transactions on Broadcasting*, vol. 44, no. 2, pp. 182-185, June 1998.
- [9] T. L. Kunii et al., "Issues in storage and retrieval of multimedia data," *Multimedia Systems*, vol. 3, no. 5, pp. 298-304, 1995.
- [10] F. Li, and I. Nikolaidis, "Trace-adaptive fragmentation for periodic broadcasting of VBR video," in *Proceedings of 9<sup>th</sup> International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV'99)*, June 1999.
- [11] B. Ozden, R. Rastogi, and A. Silberschatz, "On the design of a low cost video-on-demand storage system," *Multimedia Systems*, vol. 4, no. 1, pp. 40-54, 1996.
- [12] J.-F. Paris, S. W. Carter, and D.D. E. Long, "Efficient broadcasting protocols for video on demand," in *Proceedings of the 6<sup>th</sup> International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, Montreal, Canada, pp. 127-132, July 1998.
- [13] J.-F. Paris, "A simple low-bandwidth broadcasting protocol for video-on-demand," in *Proceedings of International Conference on Computer Communications and Networks*, pp. 118-123, 1999.
- [14] J. F. Paris, "A broadcasting protocol for compressed video," in *Proceedings of Euromedia '99 Conference*, Munich, Germany, pp 78-84, Apr 1999.
- [15] M. Reisslein and K. W. Ross, "A join-the-shortest-queue perfecting protocol for VBR video on demand," In *IEEE International Conference on Network Protocols*, Atlanta, GA, October 1997.
- [16] M. Reisslein, K. W. Ross, and V. Verillotte, "A decentralized perfecting protocol for VBR video on demand," in *Multimedia Applications, Services and Techniques- ECMAST'98*(Lecture Notes in Computer Science Vol. 1425), pages 388-401, Berlin, Germany, May 1998.
- [17] M. Reisslein and K. W. Ross, "High-Performance Perfecting Protocols for VBR Pre-recorded Video," *IEEE Network*, vol. 12, no. 6, Nov/Dec 1998.
- [18] D. Saporilla, K. Ross, and M. Reisslein, "Periodic broadcasting with VBR-encoded video," *IEEE INFOCOM 1999*, pp 464-471, 1999.
- [19] W. S. Tan, N. Duong and J. Princen, "A comparison study of variable bit rate versus fixed bit rate video transmission," in *Australian Broadband Switching and Services Symposium*.
- [20] Yu-Chee Tseng, Ming-Hour Yang, and Chi-He Chang, "A recursive frequency-splitting scheme for broadcasting hot videos in VOD service," *IEEE Transactions on Communications*, vol. 50, no. 8, pp. 1348-1355, August 2002.
- [21] Yu-Chee Tseng, Ming-Hour Yang, Chi-Ming Hsieh, Wen-Hwa Liao, and Jang-Ping Sheu, "Data broadcasting and seamless channel transition for highly demanded videos," *IEEE Transactions on Communications*, vol. 49, no. 5, pp. 863-874, May 2001.
- [22] S. Viswanathan and T. Imielinski, "Pyramid Broadcasting for video on demand service," in *Proceedings of IEEE Multimedia Computing and Networking Conference*, vol. 2417, pp. 66-77, San Jose, California, 1995.
- [23] H.-C. Yang, H.-F. Yu, and L.-M. Tseng, "Adaptive Live Broadcasting for Highly-Demanded Videos," *Journal of Information Science and Engineering*, vol. 19, no3, May 2003.
- [24] Z.-Y. Yang, L.-S. Juhn, and L.-M. Tseng, "On Optimal Broadcasting Scheme for Popular Video Service," *IEEE Transactions on Broadcasting*, vol. 45, no. 3, pp. 318-322, September 1999.