

A dynamic QoS-guarantee prefetching protocol for real-time multimedia SMIL presentation with ATM ABR service

張英超
朝陽科技大學資管所
台中縣霧峰鄉吉峰東路 168 號
icchang@mail.cyut.edu.tw

黃明鴻
朝陽科技大學資管所
台中縣霧峰鄉吉峰東路 168 號
s8854619@mail.cyut.edu.tw

摘要

對於 Player 來說，複合式即時 SMIL 多媒體簡報的播放，最重要的莫過於播放時媒體間同步關係的維持以及如何維持 QoS 的保證(如畫面的流暢度等)。舉例來說，當使用者(Client)按下播放鍵時，需要從伺服器(Server)端下載需要播放的媒體資料至 Client 端的 Buffer 內，以維持播放時足夠的流暢度，但這需要耗費使用者的等待時間以及 Buffer 的儲存空間。在本文中提出了一個在 ATM ABR 下確保 SMIL 的 QoS 需求下，依據 SMIL 簡報對各媒體物件播放順序的描述，計算當時的頻寬及網路延遲時間，動態預載接著需播出的連續型態的媒體資料至 Client 端的協定，同時減少使用者等待的時間與 Buffer 的使用量。

關鍵字：SMIL、ATM、Prefetch、多媒體、QoS

一、背景與動機

由於多媒體資料已不像以前只有單純的影片或語音，而是集合多種媒體資料共同播出的複合式多媒體文件。在複合式多媒體文件的應用環境中，夾雜著大量的數位影音 Video、Audio、Text、Image 等數種不同類型的媒體，使得播放器(Player)在播出這些資料的同時必須耗用相當的頻寬資源，像是 Video、Audio 這類連續型的媒體資料在播出時都必須要先預存一些資料在 Buffer 裡以保持播出時的順暢，但這些均是要付出使用者等待時間與 Buffer 空間的成本。這個問題，隨著影音資料越來越多而越行嚴重。因此假若在開始播出時，系統能自動的連續下載等一下欲播出的媒體資料

時，使用者就不需苦等這些時間，而達到順利觀賞的目的。同時要求系統能只下載需要的部分資料，以提高 buffer 的使用效率。

本篇研究的重點，就在 Client 端即時展示 SMIL 簡報的過程中，預先估計出每個媒體物件資料將要播出前所需要的預先下載(prefetch)時間，Client 依據此動態計算所得的預載時間向相對的媒體伺服器發出需求，預先下載需要的媒體片段至 Client 端的 Buffer 內，直至取得所需要的媒體資料，以維持連續不斷的播放服務品質(QoS)，並降低預載所需的 Buffer 大小，減少系統資源的浪費。這種動態計算預載時間的協定，對於系統資源稀少的 PDA 或移動性的主機更形重要。

SMIL(Synchronized Multimedia Integration Language:同步多媒體整合語言)[1- 2]是由 W3C(Wide Web Consortium) 組織所制訂的，其開放式架構可由使用者簡易的自訂多媒體展示模型的純文字 SMIL 檔來定義各個媒體間的動作及相互關係，其好處是可以內嵌在 HTML 的語法內，並在 SMIL 檔內根據各種媒體間的時間關係，描述其開始/結束時間與彼此間的同步，就能對於多媒體資料賦予同步展示的特性，並在能支援 SMIL 規格的瀏覽器播放，如 Microsoft Internet Explorer 6.0 Public Preview [3]。另外在 SMIL 2.0 [2]中定義了一預載(prefetch)的指令，可以讓使用者預先定義 prefetch 接著要播出的媒體資料(可指定百分比、大小或播出時間)至 Client 端，但是卻沒有提及如何讓 SMIL Player 應在何時向 Server

發出這個 prefetch 指令,以減少前置等待時間,並使用最少的 Buffer 空間,進而確保播出時媒體的 QoS,因此特別將重點放在如何支援此指令。SMIL 2.0 prefetch 範例如下：

```
<smil>
  <body>
    <seq>
      <par>
        <prefetch id="endimage"
          src="http://www.example.org/logo.gif"/>
        <text id="interlude"
          src="http://www.example.org/pleasewait.html" fill="freeze"/>
      </par>
      <video id="main-event" src="rtsp://www.example.org/video.mpg"/>
      <image src="http://www.example.org/logo.gif" dur="5s"/>
    </seq>
  </body>
</smil>
```

而在網路頻寬的支援方面,非同步傳輸(ATM)這種新一代高速網路技術其最大的特色是能夠有彈性地讓使用者依據本身的需求,選擇所需的寬頻服務,對於不同媒體特性的多媒體展示,提供了合適的網路支援。在 ATM Forum [4] [5]提供了四種不同類型的頻寬服務種類。其中 ABR 的服務型態允許使用者在每一次傳輸中,變動其網路傳送的頻寬,也就是可以透過流量控制(Flow Control)的技巧進行速率的調整。在 ABR 中會保證使用者可用頻寬能夠維持在最高傳輸速率(Peak Cell Rate, PCR)和最小傳輸速率之間(Minimum Cell Rate, MCR)之間,且根據 Network fusion[5]的價格表(表一),ABR 的服務型態的價格是四種服務型態中僅次於 UBR 最低的,因此非常適合一般使用者的需求。

本文在第二節探討過去的相關研究,第三節先介紹描述動態預載之協定,以及討論所遇到的問題。第四節說明模擬的結果與比較各種方法的優缺點,最後一節則是結論以及未來的研究方向。

表一、ATM 各種服務的價格表

Service type	QoS 保證	價格(US/MB)
CBR	High	2.0
VBR (real-time)	Middle	1.2
VBR (Non real-time)	Middle	0.75
ABR	Low	0.5
UBR	Very Low	0.4

二、相關研究

在過去有關討論複合式多媒體文件的文獻中,大多侷限於媒體間的同步關係或是網路傳輸時如何維持 QoS 的問題。關於多媒體動態預載方面的研究,在 Lin 等[6]針對 VOD 系統提出一個 VBR stream 排程的演算法架構,使媒體資料能即時傳輸到用戶端。

在 Hoossam 等[7]中則是提出一 "Scheduled-multicast" 的技術,利用 Tree 的觀念,在媒體資料從 Server 送至 Client 的過程中,分散在路徑上各個節點的 Buffer 內,以減少網路頻寬的需求並增加媒體的使用效率,使資料能即時廣播送達至 Client 端。

Khan 等[8]則是在 Internet 的環境下,提出一個可用在預載 Web Browser 上媒體物件的技術,藉由預測要播出的是哪一個媒體,而先將其片段資料先預載至 Client 端的 Buffer 上,剩餘的部分等媒體真正播出時再下載的方法,如此使用戶端在瀏覽網頁的過程當中感覺不到任何媒體下載時的延遲,但卻需要浪費較多的儲存空間。

這些論文都沒有考慮網路頻寬與延遲變化的因素,因此本文將重點擺在如何依當時的網路資源,在 Client 端動態預測媒體播出前的下載時間點,透過 ATM ABR 流量控制的機制,即時改變下一個片段的下載時間,使得 Buffer 的使用量可以達到最低,又能符合 QoS 的需求。

三、動態預載之協定

在媒體播出前,為了要讓使用者的前置等待時間降到最低,首先要能在其播出前估算其下載的前置時間點,才能準確的予以事先處理,因為網路的頻寬及延遲時間都是動態變動的,我們由底下的公式得出幾個重要的參數：

1. 傳輸延遲時間(D)的量測

在 Client 端向媒體資料所在的 Server 發出一測量延遲時間的 Request RM(resource management) Cell,並記錄這個時間為 T1。當 Server

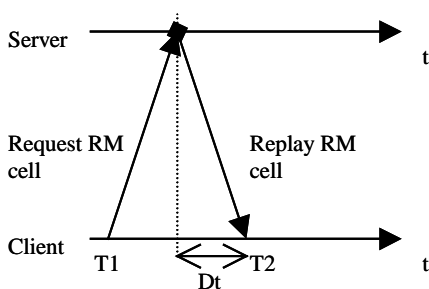
收到後再將此 Request RM Cell 以相反方向回傳 RM cell 給 Client 端，當 Client 端收到後記錄時間為 T2(其中時間的延遲包括了網路傳輸的延遲及 Server 端處理的延遲)。利用公式(1)將先後所得的時間差除以 2，就可測得在時間 t 由 Server 端發送 Cell 至 Client 端所需要經過的傳輸延遲時間 D_t。如圖一所示。

$$D_t = \frac{T_2 - T_1}{2} \dots\dots\dots (1)$$

由 Client 端每發出一個 Request RM Cell 均需佔用少許的網路頻寬，且發出的次數越多，浪費的頻寬資源就越多，因此為了要減少無謂的頻寬浪費，就有必要需減少發出 Request RM Cell 的次數，採行的方式為將 Client 發出 Request RM Cell 的週期間隔時間分成四組，分別為 250ms、500ms、750ms、1000ms 四組週期，依據媒體類型(如動作片或文藝片，其編碼後的 bit rate 曲線斜率會有明顯的不同)、網路等情況分別採取最適合的時間週期，以達最佳的效果，將再第四節展示實驗的結果。

表二、測量週期/量測值/頻寬之關係

時間間隔	量測值	頻寬使用
250ms	最精確	最多
500ms	尚可	尚可
750ms	尚可	尚可
1000ms	最不準確	最少



圖一、網路延遲時間的測量

2. 頻寬(BW) 的量測

在媒體下載期間，為了避免浪費多餘的時間來進行頻寬的量測，此處採取一種方法，就是利用已知的媒體物件 i 下一個片段播出時所需要的資料量大小(Needed Bytes: NB_tⁱ)以及當 Client 從 Server 接收到此媒體片段資料的第一個 Bit 到最後一個 Bit 所經過的時間(Transfer

time: TT_tⁱ), 所測得當時物件 i 所使用的平均頻寬。

$$BW_t^i = NB_t^i / TT_t^i \dots\dots\dots(2)$$

3. 預測下載時間點(PT)

經由 SMIL 簡報中對各媒體內容播出開始及結束時間的描述，可得知各個媒體片段的播出時間(Display time: DT)。假設測得當時的頻寬為 BW，且各媒體片段播出時所需要的資料量大小為 NB，因此從圖二中可知藉由 DT、TT 與 D，可以反算得到 PT。但在實際上無法使用同一個片段所得到的 TT 與 D 去算出 PT，因為此時資料已經傳送下來了。因此在公式(3)中採用前一個片段(時間 t)量測所得的 D 與 BW，算出的下一個片段(t+1 時間)的預載時間點(Prefetch Time: PT) 為：

$$PT_{t+1}^i = DT_{t+1}^i - D_t^i - \frac{NB_{t+1}^i}{BW_t^i} \dots\dots\dots (3)$$

- i 為媒體物件編號, t 為時間點
- NB: 為此時間點物件片段所需要的資料位元數
- D: 為網路延遲的時間
- BW: 為此物件所分配到的平均頻寬
- DT: Display time 為物件片段已知的播放時間點
- PT: 為預測物件片段的下載時間點

因為網路頻寬 BW 無時無刻的在變動，且網路的延遲時間 D 也會變動，因此之前所求得的 PT 值在下一片段可能不再正確，使得 DT 會早於或是晚於正確應該播出的時間，而造成必須浪費過多的 Buffer 空間存放提早下載的資料或播放中斷，所以必須時常監視網路上頻寬與延遲的變化，以便能適時反應並加以調整 PT。以圖三來說明，當 Client 端開始要播放媒體資料時，會先向 Server 端發出一開始下載的命令(經由 ABR RM cell 中定義的 flag 來判別是何種命令)，假設網路延遲的時間為 D₀，這個命令在 Server 收到後再傳送第一個 Bit 到 Client 端時總共會延遲 2D₀ 的時間，而 Server 傳送最後一個 Bit (傳送完畢後立刻暫停傳送)到 Client 端時所經過的時間為 $TT_0^i = \frac{NB_0^i}{BW_0^i}$ ，系統設定的最初預載時間(Initial prefetch time: IPT)，必須滿足公式(4)的條件。

$$IPT > MaxRTT + \frac{NB_t}{MinBW} \dots\dots\dots(4)$$

法，並不像 MPEG Video 那般有明顯的 GOP 架構，最簡單的方式就是改傳送編碼速率較低的物件資料。

1.2.非連續媒體物件

像是 Image (如 GIF、JPG)、Text 等都是屬於此類，因此類媒體的特性必須將其資料完整下載下來才能播出，無法在中途臨時改變編碼方式，而且此種媒體物件可以在極短的時間內下載完畢，故此時無須對物件本身做任何的變動。

2. 依據媒體重要性分配頻寬，以試著取得較大的頻寬：

過去的研究有透過標準的 SMIL 文件來延伸 SMIL 的定義而達到區別媒體物件重要性的方法，如 Yoshiki[9-10]中使用的自訂 Priority，以區別各媒體物件重要性，但這些方法並無法符合一般使用者的需求。在此我們提出一方法用來評估欲播出的媒體物件間的優先順序關係(重要性)，選擇優先權較高的媒體物件賦予其較多的頻寬而使其優先下載完成，而優先權較低的則降低其頻寬使其延後下載，如此使得媒體能更平順的播出。

物件重要性(Media Importance: MI)的產生方法，在此處是採用[11]所提出的方法，是經由解析 SMIL 檔而產生多媒體物件重要性表。雖然透過播放時間的長短、播放畫面的面積、播放畫面的深淺度、播放聲音的強弱的定義，可能仍不足以表示一個物件的重要性，但我們將嘗試透過此一方法，來判斷物件的重要性為何。

$$BW_i = \sum_j BW_j * MI_i \dots\dots\dots(5)$$

先以[11]中所提的方法得出每個物件的 MI 重要性(介於 0~1)程度值後，以當時總共可用的 ABR 頻寬來分配各個物件的頻寬值，再利用 ABR 之頻寬調整機制來調整各個物件所賦予的頻寬，也就是說若物件越重要，其所分配的頻寬就越大，完成下載的時間就越短，以期能符合物件之播放時間。詳細的演算法請參

考[12-13]。

在確定了 PT_{t+1} 後，就可以根據這個時間預載下一個片段。然而用到的是上次的延遲與頻寬，加上這二者都可能隨時間而變化，因此會進一步造成以下三種狀況(圖三)：

- (1.) 時間點 a 的情況表示 BW 變大或 D 變小，Client 接收資料的進度超前，Server 停止繼續傳送資料，以免繼續增加 Buffer 的使用量。
- (2.) 時間點 b 的情況表示完全符合媒體要播放的時間點 DT_{t+1} ，此為最理想的情況，可以使 Buffer 的使用量達到最低。
- (3.) 時間點 c 的情況表示 BW 變小或 D 變大，prefetch 完成的時間點會超過正常播放的時間 DT_{t+1} ，使得媒體播出時間延後而造成中斷，這時需要採取丟棄超過 DT_{t+1} 時間的資料，並採用與上面相同的補救措施。

到此階段已完成一個預載的循環，要繼續下一個媒體片段的下載預測，直到媒體全部下載完成為止。

整個預載的演算法流程如下：

```

Step 1:計算媒體剛開始播放時的 initial prefetch time

Step 2:
當到達下一個監測時間點或是播放時間點 begin
Client 向 Server 發出 request RM cell 及接收回傳的 reply RM cell
計算網路延遲時間  $D_t$ 
接收 Server 傳來的資料進 buffer
If 任一個 bit 收到的時間超過  $DT_t$  then
    Drop 之後所有 bits
Else
    播放這個片段的資料
    Case 1:
        通知 Server 改傳送較低速率之資料
    Case 2:
        依據媒體重要性分配頻寬
End if
計算網路頻寬  $BW_t$ 
計算下一個預載時間點  $PT_{t+1}$ 
{預載時間點正確的情況}
If  $PT_{t+1} >= DT_t + D_t$  then
    Exit
Else {預載時間點無法符合播出時間點的情況}
    Case 1:
        通知 Server 改傳送較低速率之資料
    Case 2:
        依據媒體重要性分配頻寬
End if
End
    
```

四、模擬結果

本模擬採用如圖四的網路架構,每個 link 頻寬為 155Mbps,使用圖五的 SMIL 文件,各物件的屬性如表三所示。

圖六說明了在監測網路延遲週期時間的安排上各種數據對效率的影響,分別是 250ms、500ms、750ms、1000ms 四組,因為 250ms 是監測週期最頻繁的,所以總體來說測得的值也最準確,但因為需要不時的發出詢問的 Request RM Cell,因此也最耗費頻寬資源。而 750ms 與 1000ms 因為間隔週期時間太長,故無法正確得出正確的網路延遲時間,至使得求出的 prefetch 時間變的不準確,只有 500ms 這一組所得出的值較接近原先之需求。

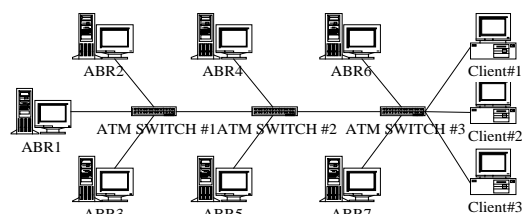
接下來模擬的部分,將分成沒有 prefetch、固定前置時間 prefetch (間隔時間為 250 ms) 及動態 prefetch 三個方法來測量。

Video1 及 Video 2 是分別來自動作片電影尖峰時刻的 DVD MPEG2 部分片段,其 bit rate 曲線圖分別為圖七及圖八。而 Video3 及 Video 4 則是分別來自文藝片電影落跑新娘的 DVD MPEG2 部分片段,其 bit rate 曲線圖分別為圖九及圖十。在模擬的過程當中,Video3 及 Video4 會取代 SMIL 檔中的 Video1 及 Video2,這兩種不同類型的影片將用來比較物件不同 bit rate 曲線對 prefetch 結果的影響,圖十至圖十七為模擬的結果。

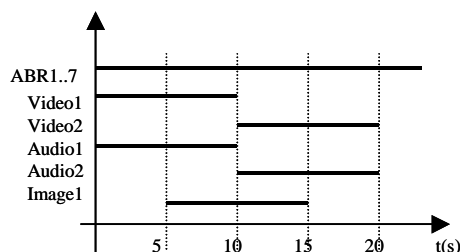
橫軸表示 Client 端的时间軸,縱軸為接收到的總資料量。實驗的曲線越接近真實 Video 的播放需求,代表此種方法使用越少的 Buffer,滿足 QoS 需求。而在背景設計一個 VBR 的流量經由 Switch1 到 Switch3,其頻寬值是由 Poisson ($\lambda=200$ /ms)所求得的,平均流量為 84.8Mbps,另外安排兩個 ABR 流量從 Server ABR3 至 Client#1 及 Server ABR7 至 Client#3,其頻寬值是由 Poisson ($\lambda=4$ /ms)所求得的,平均流量為 10Mbps。

在一開始時因為各媒體需要一段 initial pref-

etch time,所以整個系統播出的時間會往後移,而後,沒有做 prefetch 的一方,就會隨著網路頻寬的起伏而有劇烈的變動而有忽停忽放的現象。而固定時間 prefetch 的情況則會隨著媒體大小不同及頻寬變動而變化,可能會因為媒體較小而提前下載完成或是因太大而無法準確預估前置時間而延後播出,且會使得 Buffer 的使用量較大。而在動態 prefetch 的結果方面,每經過一段固定的監測點都會重新預估下一個下載的 PT 值,若不符合時,則會採取進一步的措施(如改變媒體的編碼方式或依媒體的 MI 值以增加或降低所分配的頻寬),使得媒體播出的時間不會因為網路頻寬的變動而影響甚劇,合乎使用者 QoS 的需求。



圖四、模擬之 ATM 網路系統



圖五、SMIL 檔內各媒體播放時間之關係

圖五的 SMIL example 如下:

```
<smil>
...
<body>
<seq>
  <par>
    <video src="rtp://ABR1/Video1.mpg" begin="0s"
dur="10s" region="video1"/>
    <audio src="rtp://ABR5/Audio1.mp3" begin="0s"
dur="10s"/>
    
  </par>
  <par><video src="rtp://ABR2/Video2.mpg"
begin="10s" dur="10s" region="video2" />
  <audio src="rtp://ABR4/Audio.mp3" begin="10s"
dur="10s"/>
  </par>
</seq>
</body>
</smil>
```

</seq>
</body>
</smil>

表三、各媒體物件屬性

媒體物件	媒體類型	檔案大小	播放時間
Video1	Video	7.7Mb	10s
Video2	Video	7.8Mb	10s
Video3	Video	7Mb	10s
Video4	Video	7.2Mb	10s
Audio1	Audio	100Kb	10s
Audio2	Audio	90Kb	10s
Image1	Image	20Kb	10s

針對不同 bit rate 曲線所作的模擬(圖十一、圖十三與圖十六、圖十七), 從圖中可看出並不會因為媒體資料量不同(曲線變化較劇烈)而使 prefetch 動作失去效果, 因此可證明此動態調整 prefetch 時間的方法確實可符合需求。

五、結論

在本文中提出了一個能動態預測媒體下載時間點的協定, 使得 Client 端的 Player 在播出的同時, 能在背景不斷的估算媒體從 Server 端下載的情況, 進而減少媒體下載的前置時間, 使得媒體播出時能夠保證一定的 QoS 需求, 減少 Buffer 的使用, 並維持 SMIL 中各媒體的同步關係, 這對於未來應用在 ATM 無線網路中如何在移動式主機(Mobile Host)以有限的頻寬及 Buffer 容量來播放多媒體的資料將是下一步的研究重點。

參考文獻

[1] W3C, "W3C Issues Synchronized Multimedia Integration Language(SMIL 1.0) Specification", <http://www.w3.org/TR/REC-smil>

[2] W3C, "W3C Issues Synchronized Multimedia Integration Language(SMIL 2.0) Specification", <http://www.w3.org/TR/SMIL20.html>, 2000.

[3] Microsoft Internet Explorer 6.0 Public Preview, <http://microsoft.com/windows/ie/preview/default.asp>, 2001.

[4] The ATM Forum Technical committee, "Traffic Management Specification Version 4.0", April. 1996.

[5] Networkfusion, "Qwest revealed to Network World last week a new line of enterprise services in the basic categories favored by the Big 3 carriers: frame relay, ATM, private lines and Internet access ", <http://www.nwfusion.com/news/1214qwest.html>, 1998.

[6] Chow-Sing Lin, Min-You Wu and Wei Shu, "Transmitting Variable-Bit-Rate Videos on clustered VoD systems", IEEE International Conference on Multimedia and Expo, vol.3, pp1461-1464, 2000.

[7] Hossam El-Gindy, Chi Nguyen, Antonios Symvonis, "'Scheduled-Multicast' with Application in Multimedia Networks", IEEE International Conference on Networks, pp191-198, 2000.

[8] Javed I. Khan and Qingping Tao "Partial Prefetch for Faster Surfing in Composite Hypertext", Proc. of the 3rd USENIX Symposium on Internet Technologies USITS'01, pp13-24, March 2001.

[9] Yoshiki Terashima, Keiichi Yasumoto, Teruo Higashino, Kota Abe, Toshio Matsuura and Kenichi Taniguchi, "Extension of SMIL with QoS Control and its Implementation", IEEE International Conference on Multimedia and Expo, Vol3 , pp 1683-1686, 2000.

[10] Yoshiki Terashima, Keiichi Yasumoto, Teruo Higashino, Kota Abe, Toshio Matsuura and Kenichi Taniguchi, "Integration of QoS Guarantees into SMIL and its Flexible Implementation", Proc. of IEEE IWQoS, pp 164-166, 2000.

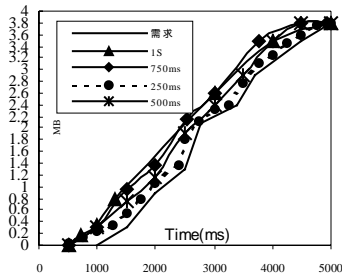
[11] 張英超, 謝盛文, "透過 ABR 服務動態確保 SMIL 多媒體簡報品質的架構", 朝陽科技大學資訊管理系碩士論文, 2001.

[12] 張英超, 李麗華, 謝盛文, "在 ATM 上透過 SOM 建構一個公平又有效的流量控制機

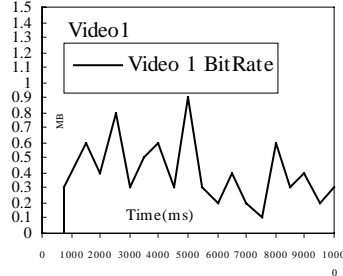
制",第五屆電腦與通訊研討會, pp.1A-13-1A-18, 大葉大學,彰化, Oct. 2000.

the 2000 International Computer Symposium (ICS2000), Workshop on Computer Networks, Internet, and Multimedia, pp.1-6, Chiayi, Taiwan, R.O.C., Dec. 2000.

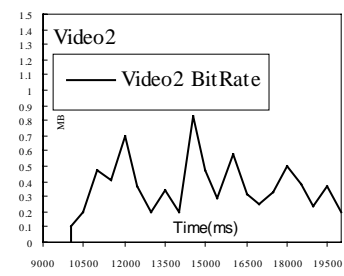
[13] Ing-Chau Chang and Sheng-Wen Hsieh, "A Fair and Efficient SOM-based Feedback Congestion Control in ATM Networks", Proc. of



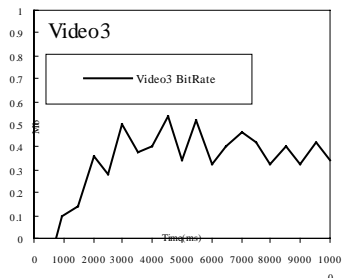
圖六、監測時間週期之比較



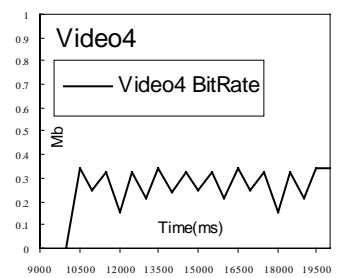
圖七、Video1 的 Bit Rate 曲線圖



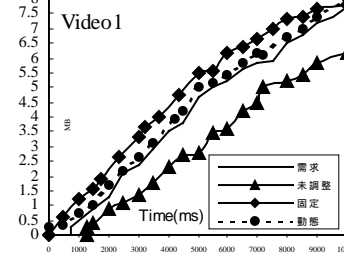
圖八、Video2 的 Bit Rate 曲線圖



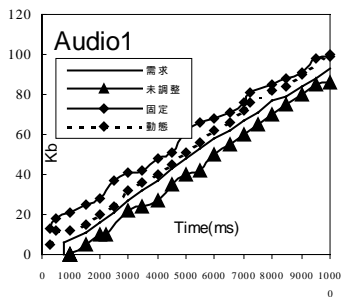
圖九、Video3 的 Bit Rate 曲線圖



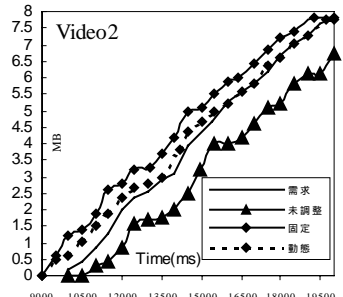
圖十、Video4 的 Bit Rate 曲線圖



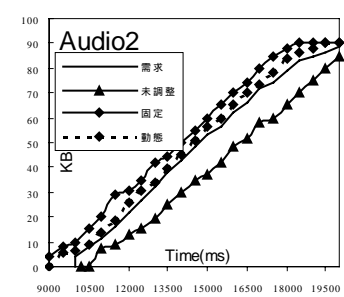
圖十一、Video1 的各種控制方法曲線圖



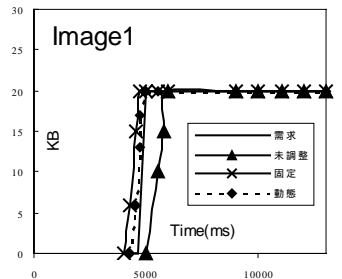
圖十二、Audio1 的各種控制方法曲線圖



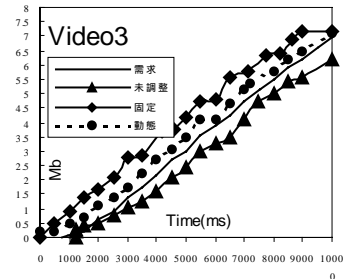
圖十三、Video 2 的各種控制方法曲線圖



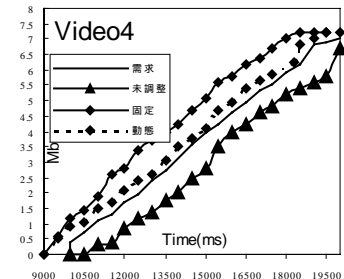
圖十四、Audio2 的各種控制方法曲線圖



圖十五、Image1 的各種控制方法曲線圖



圖十六、Video3 的各種控制方法曲線圖



圖十七、Video4 的各種控制方法曲線圖