

# 應用於即時影音通訊之適應性 UDP 頻寬控制

梁恩誌  
國立台北科技大學 電機  
工程系  
[s2318013@ntut.edu.tw](mailto:s2318013@ntut.edu.tw)

王永鐘  
國立台北科技大學 電機  
工程系  
[ycwang@ntut.edu.tw](mailto:ycwang@ntut.edu.tw)

王振興  
國立台北科技大學 電機  
工程系  
[jswang@ntut.edu.tw](mailto:jswang@ntut.edu.tw)

## 摘要

由於今日網路通訊技術的進步，促使即時影音通訊成為時尚興起的一種網路應用，然而實體網路設備並非理想，當網路發生壅塞，就會出現不可預測的封包遺失，造成接收端影音品質的失真。本論文利用一個以 UDP 為基礎的頻寬估測機制，配合 TCP 壅塞控制的概念，完成一個適用於即時影音通訊的 UDP 頻寬控制機制，降低資料封包的遺失機率。

## 1. 簡介

今日網路通訊技術的進步與成熟，促使即時影音通訊成為一個新興的應用，壓縮法的不斷演進，讓視訊與音訊訊號的即時傳送不再遙不可及，然而這些應用大都架構在 UDP 通訊協定上，UDP 的優點是能提供穩定位元率 (Bit Rate) 的資料流量以及沒有 TCP 等待重送機制造成的傳送延遲，缺點則是不保證傳輸資料到達接收端的完整性，所以當網路發生壅塞，出現不可預測的封包遺失時，會造成接收端影音品質的嚴重失真。我們知道網路設備實際上的有限資源如：傳送速度、處理時間、暫存器大小等限制會在傳送連續、大量的資訊時出現擁塞狀況，進而造成傳送資料發生錯誤、損毀或遺失等情形，使得接收端影音品質下降，因此，如何在傳輸即時影音資訊的過程中，配合當時的網路狀態調整傳輸的資料量，避免造成網路壅塞，已成為熱門的研究課題。

目前的解決方法主要是提高壓縮倍率或是在即時影音服務中加入頻寬偵測、控制機制，避免造成網路發生擁塞。Chung[1]等人提出的方法是利用一個最小均方值 (Least Mean Square, LMS) 控制器計算可用頻寬，優點是能貼近目前網路的可用頻寬，缺點是需要不斷回傳統計資訊給傳送

端，為網路帶來額外的負擔；Chung[2]等人利用一個單向的 UDP 頻寬估測機制，搭配適當的調整完成頻寬控制的目的，優點是不需要一直回傳資料給傳送端，缺點是頻寬估測的收斂速度太慢；Nam[3]等人利用一個探測 (Probing) 的機制來估測可用的網路頻寬，優點是能準確的估測出可用的網路頻寬，缺點則是會佔用資料傳輸路徑上的頻寬，增加網路發生壅塞的機率。

本論文的目的是提出一套以單向 UDP 頻寬偵測為基礎的適應性頻寬控制機制，這個機制主要是在接收端觀察資料封包的遺失狀況，利用一個低頻濾波器求出基本頻寬，加上一個回授控制器來加快估測頻寬的收斂速度，再將估測結果回傳給傳送端，傳送端可據此調整編碼模式、壓縮倍率等參數，調整傳輸流量。頻寬估測方法的回顧會在本論文的第二節中介紹，第三節是介紹本論文提出的頻寬估測機制，第四節說明本論文如何完成適應性的頻寬控制，第五節是模擬與驗證結果，最後是本論文的結論。

## 2. 頻寬估測

網際網路的 IP 協定是一種 Best Effort 的服務模式，它會盡全力將資料從傳送端 (Sender) 送到接收端 (Receiver)，然而，它並不保證傳輸的延遲程度或是遺失的情形。延遲形成的原因包括封包在主機 (Host) 與路由器 (Router) 中的處理延遲 (Processing Delay)、封包到達路由器後等待處理的延遲 (Queuing Delay)、封包在路由器中等待傳送的延遲 (Transmission Delay) 和連線傳遞的延遲 (Propagation Delay)，而造成封包遺失的情形有傳輸線錯誤與路由器的緩衝器溢滿 (Buffer Overflow)，其中以後者的影響最大，因為現今網路的主要線路大多已更換為光纖，因傳輸線錯誤造成封包遺失的原因只有光纖雷射暫時性故障，否則因輸線錯誤造成封包遺失

的情形是非常少見的[4]。擁塞控制的目的便在避免傳送過量的封包造成路由器的緩衝器溢滿。

擁塞控制的機制有兩種[5]：流量控制（Rate Control）與流量成形（Rate Shaping）。前者是比對傳送資料的位元率（Bit Rate）與測得的可用網路頻寬，將封包的遺失數量最小化；後者是根據流量控制演算法計算出位元率的大小來送出資料。現在的頻寬偵測技術可分為兩大類：探測基礎法（Probe-based）、模型基礎法（Model-based）。

### 2.1. 探測基礎法

探測基礎法可以再細分為以傳送端為主（Sender-based）與以接收端為主（Receiver-based）的機制。Sender-based 的機制主要是由傳送端在某一個時間周期後，送出一串固定封包大小與時間間隔的封包串給接收端，接收端接收到測試封包後，直接將該封包回傳給傳送端，傳送端觀察回傳封包間的時間間隔變化與封包遺失情形來調整傳送資料的位元率；Receiver-based 的機制是傳送端直接增加傳送的資料量，若接收端可以成功接收又沒有造成擁塞，表示這次增加的量是正確的，反之，接收端可以直接丟掉資料新增的部份。探測基礎法的目的是藉由不斷的調整傳送位元率，使得封包的遺失機率低於一個期望的臨界值。

### 2.2. 模型基礎法

模型基礎法主要是依據 TCP 連線的流量模型（Throughput Model）來建構頻寬計算公式，因此，模型基礎法也被稱為 TCP 友好的速率控制方法。根據回傳訊息的有無，這個方法可以再細分為雙向（Two-way）與單向（One-way）兩個架構。雙向的模型基礎法是把測量得到的測試封包往返時間（Round Trip Time, RTT）、時間變化量與統計的封包遺失機率（Packet Loss Rate）等參數代入公式中計算，求得一個估測的頻寬值，估測的準確度取決於使用的公式、量測的時間點、量測的時間刻度等因素；單向的模型基礎法則是把測量得到的測試封包的傳送時間（Sending Time）、時間變化量與統計的封包遺失機率等參數代入公式中計算，求得一個估測的頻寬值，估測準確度取因素與雙向的模型基礎法接近，不過因為單向的時間計算是在不同的主機上，所以傳送端與接收端的时间同步準確度也會對結果造成

顯著的影響。表一是探測基礎法與模型基礎法的比較：

“表一 探測基礎法與模型基礎法的比較”

項目	探測基礎法	模型基礎法
封包格式	已知	未知
誤差範圍	較小	較大
估測時間點	由一個時間周期決定	有成功接受封包就能估測
估測時間	數秒	取決於模型的複雜度，通常較探測基礎法快
適合的應用	準確度要求高	延遲要求高

### 3. 動態的單向模型基礎頻寬偵測

單向頻寬偵測的優點是偵測路徑就是傳送路徑，缺點是需要做傳送端與接收端的时间同步，本論文利用簡單的網路時間協定（Simple Network Time Protocol, SNTP）完成時間的同步。

由上一節的介紹與表一的比較，本論文提出一套混合探測基礎法與模型基礎法的單向頻寬偵測機制，稱為動態的單向模型基礎頻寬偵測（Dynamic One-Way Model-Based Bandwidth Estimation, DOMBE），原理是將視訊與音訊的連續資料視為探測封包，然後代入一個頻寬估測模型中計算可用頻寬。

我們知道探測基礎法需要一個量測的時間點與週期，因此本論文利用  $Cnumber$  變數來定義這個週期的長短，其大小介於 1 到  $MaxNumber$  之間， $Nnumber$  代表目前已接收的封包數，若  $Nnumber$  小於  $Cnumber$  就將目前接收封包的大小與傳送時間加到  $TotalSize$  與  $TotalTime$  兩變數中， $Nnumber$  數量加 1，並且等待接收下一個封包；若  $Nnumber$  大於或等於  $Cnumber$  則進行頻寬的估測計算，估測流程如下：

Step1：

以 1 式計算目前時間週期的傳輸流量（Bandwidth）與資料單位的傳送時間（TimeInterval）。

$$Bandwidth = \frac{TotalSize}{TotalTime} = \frac{1}{TimeInterval} \quad (1)$$

Step2 :

將 1 式的結果代入 2 式中，利用 1 式與 2 式的結果來模擬探測基礎法中的已知傳送封包串時間的特點，再使用 3 式評估目前的網路狀態，其中  $a$  是指數加權平均係數； $g = 1$  代表目前的網路可用頻寬趨近無限大； $g = 0$  表示目前的網路已達到負載的極限； $g < 0$  則是網路進入壅塞狀態，需要調降傳送的 Bit Rate。

$$\text{MeanTimeInterval} = (1-a) * \text{MeanTimeInterval} + a * \text{TimeInterval} \quad (2)$$

$$g = \frac{\text{MeanTimeInterval} - \text{TimeInterval}}{\text{MeanTimeInterval}} \quad (3)$$

Step3 :

透過 4 式評估前後時間間隔的傳輸流量變異程度。 $\text{LastTotalSize}$  與  $\text{LastTotalTime}$  分別代表上一個時間週期統計的接收資料量與傳送時間。

$$\text{VarBandwidth} = \frac{\text{TotalSize} - \text{LastTotalSize}}{\text{TotalTime} - \text{LastTotalTime}} \quad (4)$$

Step4 :

如果沒有發生爆發性 (Bursty) 的封包遺失，就將 1、3、4 三式的計算結果代入 5 式中，求出目前的可用頻寬，反之則以 6 式計算可用的網路頻寬， $b$  是低頻濾波器係數。

$$\text{AvailBandwidth}_{k+1} = (1-b) * \text{AvailBandwidth}_k + b * \text{Bandwidth} + g * |\text{VarBandwidth}| \quad (5)$$

$$\text{AvailBandwidth}_{k+1} = \frac{1}{2} * \text{AvailBandwidth}_k \quad (6)$$

#### 4. 適應性 UDP 頻寬控制

為了讓 DOMBE 機制的反應速度能隨網路狀態變化，同時兼具公平性 (Fairness) 和 TCP 親和性 (TCP Friendliness) 的能力，因此本論文借鏡 TCP 壅塞控制 (Congestion Control) 的概念，規劃出起動狀態、快速恢復狀態與穩定狀態三個不同的處理狀態，適當的調整  $Cnumber$  與  $\text{AvailBandwidth}_{k+1}$  的統計方式。

起動狀態：

設初始的  $Cnumber = 1$ ，為了更快偵測出目前網路的極限，所以修改 5 式如下：

$$\text{AvailBandwidth}_{k+1} = \text{AvailBandwidth}_k + |\text{VarBandwidth}| \quad (7)$$

限制  $\text{AvailBandwidth}_{k+1} \leq 2 * \text{AvailBandwidth}_k$ ，避免 7 式的模型增加速度過快，超過實際網路頻寬值太多，造成收斂速度變慢，另外，若  $Cnumber <$  每秒的傳送張數 (Frame Per Second,

FPS)，則設  $Cnumber$  為原先的兩倍。如果發生封包遺失或 Bursty 的封包遺失時，就使用 6 式並進入快速恢復狀態。

快速恢復狀態：

這個狀態是期望能快速恢復到穩定的頻寬範圍，因此採用的模型與限制和起動狀態一樣，不同的是，當估測的頻寬值大於進入快速恢復狀態時之頻寬值的 1.5 倍時，DOMBE 機制就進入穩定狀態。

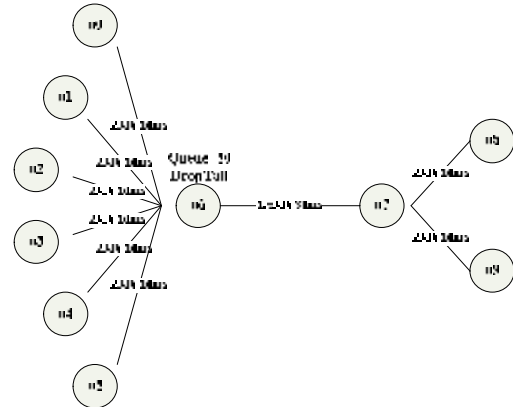
穩定狀態：

在穩定狀態下，本論文使用原本的 DOMBE 機制來估測網路可用頻寬，前後時間間隔估測頻寬值的變化量會被限制在 10% 以內，這是因為穩定狀態下的頻寬本就不應該會劇烈變化，同時， $Cnumber$  的大小是以  $60 * FPS$  為單位，根據有無封包遺失、估測的頻寬變化量與  $g$  的正負做不同的調整，如果發生爆發性封包遺失，穩定狀態會進入快速恢復狀態。

#### 5. 實驗與模擬

由於實際網路是多變、無法控制的，為求網路模擬環境的一致，本論文選擇以 NS2 (Network Simulator-Version 2) [6][7] 來建構模擬平台，這套網路模擬軟體廣泛的應用在各種系統的模擬上，同時也是許多研究採用的驗證工具，具有相當的公信力，以下將先介紹模擬時的假設條件，最後以幾種不同的例子證明本論文的頻寬控制機制確實兼具 Fairness 和 TCP Friendliness。

圖一是所有模擬例子的網路拓模，環境參數如圖中所示：



“圖 1 模擬的網路拓模”

網路拓模採用星狀圖連接，主要是為了模擬傳送端與接收端間的瓶頸連線 (Bottle Link)，圖 1 中左邊的六個節點為傳送端，右邊的兩個節點為接收端。視訊資料一般是以 UDP 傳送，在沒有調



整壓縮參數的前提下，視訊的 Bit Rate 可以視為是一個穩定的流量，所以在模擬中設定 UDP 節點的訊務流量產生方式為 CBR (Content Bit Rate)，詳細的系統模擬設定如下：

UDP1：n0-n8 連線，使用本論文的流量調整機制，初始值為 300Kbps。

UDP2：n1-n9 連線，使用本論文的流量調整機制，初始值為 300Kbps。

TCP1：n2-n8 連線，版本為 Reno。

TCP2：n3-n9 連線，版本為 Reno。

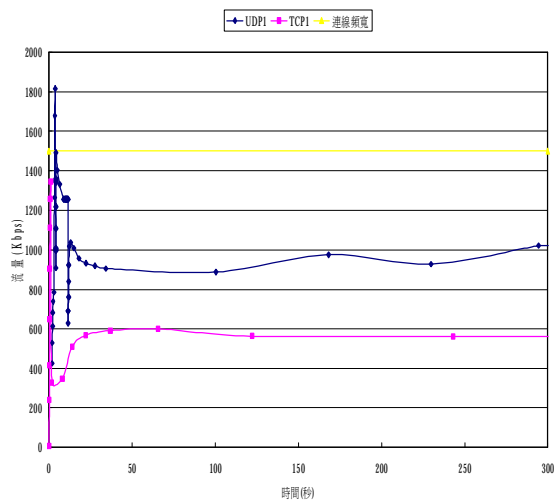
TCP3：n4-n8 連線，版本為 Reno。

TCP4：n5-n9 連線，版本為 Reno。

DOMBE 的參數設定： $FPS = 25$ ； $MaxNumber = 300 * FPS$ ； $a = 0.125$ ； $b = 0.0975$ 。

Case1：

TCP1 啟動時間：0.1 秒，結束時間：300.0 秒；UDP1 啟動時間：2.0 秒，結束時間：300.0 秒。模擬目的是證明本論文的頻寬機制能與既存的 TCP 連線達成共享頻寬。模擬結果如圖 2 所示：



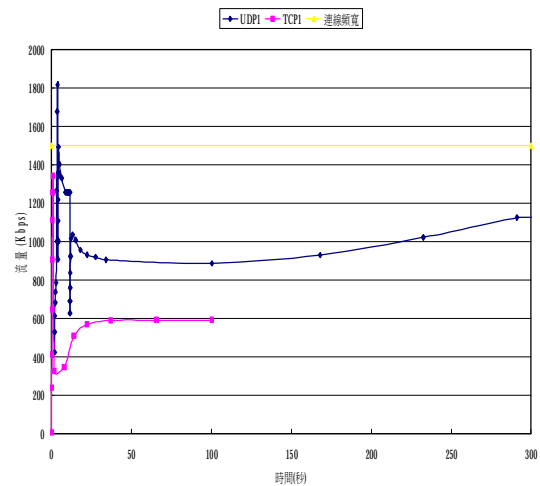
“圖 2 TCP Friendliness 之模擬結果”

從圖 2 中可以看出，本論文提出的機制與 TCP1 連線在系統時間 12 秒後就進入穩定狀態，取得平衡的時間大約花費了 10 秒，這段時間中，本論文提出的頻寬控制機制在 4 秒左右超越了設定的 Bottle Link 頻寬，因此遭遇封包遺失，促使本論文的頻寬控制機制從起動狀態進入快速恢復狀態，接著在 5 秒時進入穩定狀態，這時候估測的頻寬大約是 1400Kbps，十分接近設定的 Bottle Link 頻寬 1500Kbps，但是 TCP1 連線仍然需要傳送資料，本論文的頻寬控制機制在發現網路處於壅塞狀態後，逐步調降估測的頻寬大小，而這時

候 TCP1 連線則試圖從前面的封包遺失狀態中取得恢復，所以其傳輸流量增加速度相當的快，在 11 秒時造成一個爆發性的封包遺失，使本論文的系統再次進入快速恢復狀態，不過這時候的估測頻寬已十分接近平衡點，最後，整個模擬在 12 秒的時候進入平衡狀態。本論文提出的機制在 12 秒後估測的頻寬值會有略為高低起伏的情形，這是為了能隨時偵測到網路狀態的變化所做的設計。

Case2：

TCP1 啟動時間：0.1 秒，結束時間：100.0 秒；UDP1 啟動時間：2.0 秒，結束時間：300.0 秒。這個模擬是用來證明本論文的頻寬機制在與既存的 TCP 連線達成共享頻寬之餘，也能偵測出網路是否存在多餘的可用頻寬，並緩慢的增加傳輸流量。模擬結果如圖 3 所示：

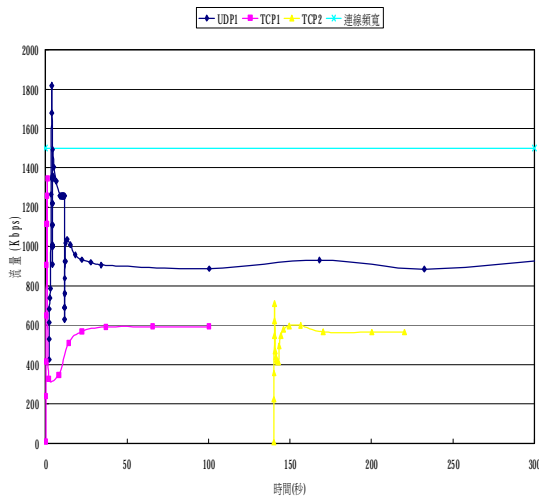


“圖 3 偵測網路多餘頻寬之模擬結果”

這次模擬中，TCP1 連線與 UDP1 連線取得平衡的過程與劇情一樣，所以不再贅述，從圖 3 中可以清楚看到，在系統時間 100 秒的時候，TCP1 連線停止傳送資料，UDP1 發覺網路壅塞狀態降低，因此從 100 秒後緩慢調升估測頻寬的大小，最後靠近 300 秒時出現略微的頻寬降低是因為累計的資料封包數不足的關係。

Case3：

TCP1 啟動時間：0.1 秒，結束時間：100.0 秒；TCP2 啟動時間：140.0 秒，結束時間：220.0 秒；UDP1 啟動時間：2.0 秒，結束時間：300.0 秒。這個模擬是用來與 Case2 的結果做對照，證明本論文的頻寬機制能準確的偵測出網路是否存在多餘的可用頻寬，並適當的調整傳輸流量。模擬結果如圖 4 所示：

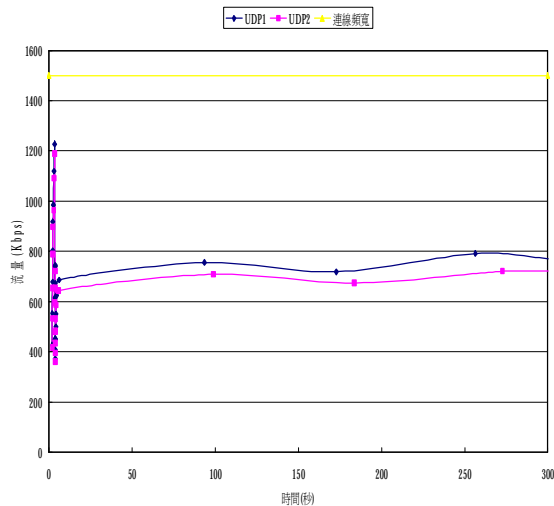


“圖 4 適應性頻寬估測之模擬結果”

這次的模擬與上一個類似，在 100 秒的時候 TCP1 停止傳送資料，UDP1 開始調升傳輸流量，在 170 秒時，UDP1 發覺傳輸路徑中出現新的傳輸流量，因此降低自己的傳輸流量，接著在 230 秒這個點上，因為新出現的 TCP2 已經在 220 秒停止傳送資料，網路負荷再度降低，所以 UDP1 又開始緩慢調升傳輸流量，從這個模擬結果可以看出，本論文提出的頻寬控制機制確實能根據網路狀態做適當的調整。

Case4 :

UDP1 啟動時間：2.0 秒，結束時間：300.0 秒；UDP2 啟動時間：2.0 秒，結束時間：300.0 秒。這個模擬的目的是用來證明本論文的頻寬控制機制除了 TCP Friendliness 外，還具有 Fairness 的特性。模擬結果如圖 5 所示：

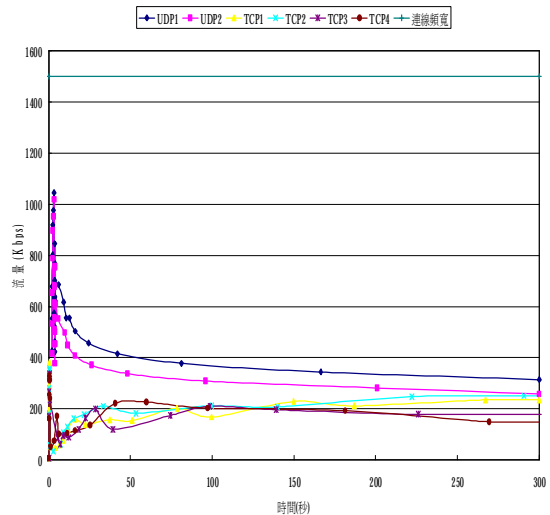


“圖 5 Fairness 之模擬結果”

前面幾個模擬主要都是在證明本論文的頻寬控制機制具有 TCP Friendliness 的特性，這個模擬則是要證明本論文的頻寬控制機制也具有 Fairness 的特性，因為一般的即時影音通訊應用都會有兩個以上的使用者參與，如果一套頻寬控制機制無法在使用者之間取得平衡，就算它能提供再好的 TCP Friendliness 特性也是沒有辦法實際應用的，從圖 5 可以清楚的看出，本論文提出的頻寬控制機制具有優異的 Fairness 特性。

Case5 :

TCP1~TCP4 啟動時間：0.1 秒，結束時間：300.0 秒；UDP1、UDP2 啟動時間：2.0 秒，結束時間：300.0 秒。這個模擬的目的是證明本論文的頻寬控制機制是能同時兼具 TCP Friendliness 與 Fairness 兩種特性。模擬結果如圖 6 所示：

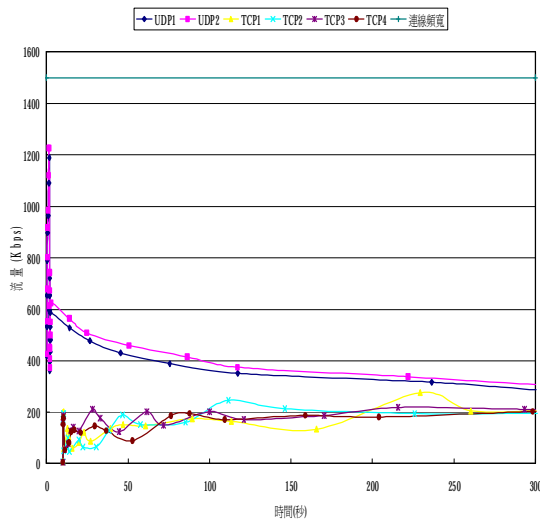


“圖 6 TCP Friendliness 與 Fairness 之模擬結果一”

從圖 6 可以看出，4 個 TCP 連結開始傳輸資料後便很快的達到 Fairness 的平衡點—375Kbps，在 UDP1 和 UDP2 也開始傳送資料後，4 個 TCP 連結的傳輸資料量迅速下降，而 2 個 UDP 連結也偵測到網路的負荷相當大，因此主動調降傳輸的資料量，在 2 個 UDP 連結啟動一分鐘後，整個網路的傳輸流量便進入 Fairness 的平衡狀態。

Case6 :

TCP1~TCP4 啟動時間：10.0 秒，結束時間：300.0 秒；UDP1、UDP2 啟動時間：0.1 秒，結束時間：300.0 秒。這個模擬的目的是證明本論文的頻寬控制機制不會因為起動時間的不同，而影響到 TCP Friendliness 與 Fairness 兩種特性。模擬結果如圖 7 所示：



“圖 7 TCP Friendliness 與 Fairness 之模擬結果二”

這個模擬的啟動順序與 Case5 相反，2 個 UDP 連結都在 0.1 秒的時候啟動，4 個 TCP 連結則在 10 秒才啟動，圖 7 證明在 UDP 佔用大部份的網路頻寬後，本論文的頻寬控制機制仍然能在偵測到新的傳輸流量出現後，自動調降傳輸流量，以達到 TCP Friendliness 與 Fairness 的目標。

## 6. 結論

提供即時影音服務更穩定的品質，不僅是研究者追求的目標，也是使用者的期待。本論文提出的適應性 UDP 頻寬控制機制，利用一個以 UDP 為基礎的單向頻寬估測方法，配合 TCP 壅塞控制的概念，完成一個適用於即時影音通訊的 UDP 頻寬控制機制，透過實作與模擬證明，這個方法確實能同時兼顧 Fairness 和 TCP Friendliness 等特性。未來的研究工作將朝向更快速的頻寬估測收斂速度、更高精準度的時間同步機制與搭配錯誤隱藏的完整品質維護架構進行。

## 7. 參考文獻

- [1] Y. J. Chung, J. W. Kim, and C. C. Kuo, “Real-Time Streaming Video with Adaptive Bandwidth Control and DCT-Based Error Concealment,” *Analog and Digital Signal Proc. of IEEE Trans. on Circuits and Systems*, vol. 46, no. 7, July 1999, 951-956.
- [2] Y. J. Chung, Y. G. Kim, J. W. Kim, and C. C. Kuo, “Receiver-Based Congestion Control Mechanism for Internet Video Transmission,” *Proc. of IEEE*

*International Symposium on Circuits and Systems*, vol. 4, June. 1999, pp. 239-234.

[3] S. Y. Nam, S. Kim, J. Kim, and D. K. Sung, “Probing-Based Estimation of End-to-End Available Bandwidth,” *IEEE Communications Letters*, vol. 8, no. 6, June 2004, 400-402.

[4] 蔡明志譯，**電腦網路**，台北：東華書局，1997，第 6-69 頁。

[5] 戴顯權，**多媒體通訊**，高雄：紳藍出版社，2003，第 11-1 至 11-83 頁。

[6] UCB/LBNL/VINT, “Network Simulation - NS(Version 2.0),” October 1998.

[7] Fall, K., K. Varadhan, *The ns Manual*, <http://www.isi.edu/nsnam/ns>, April 2002.

[8] J. F. Kurose, K. W. Ross, *Computer Networking: A Top-Down Approach Featuring the Internet – second edition*, Boston: Addison Wesley, p.p. 184-291.