

一個有效的遺失率估算法之設計與實驗

陳俊麟

屏東商業技術學院資訊管理所

李建用

屏東商業技術學院資訊管理所

摘要

本文假設在傳送端有一個可支援兩個相同內容但不同語音音質檔案（PCM 和 GSM）的雙模式編碼器。傳送端根據以變動框架長度為基礎之遺失率估算法，來估算目前網路上封包遺失的狀況，並以此作為切換高低音質檔案的依據。此演算法可過濾短期突發交通，並呈現長期交通之趨勢。除具有穩定與敏捷的特性外，此演算法亦能避免音質切換過度頻繁。

關鍵詞：遺失率、交通量測、RTP、RTCP



Design and Experiment of An Effective Loss Rate Estimation Method

Chin-Ling Chen

Department of Information Management, National Pingtung Institute of Commerce

Jian-Yong Lee

Department of Information Management, National Pingtung Institute of Commerce

Abstract

In this paper, we assume that the sender supports a two-mode codec with two audio sample files (PCM and GSM). The sender tracks the packet loss status according to the algorithm— loss rate estimation based on variable frame size and switches two audio sample files in accordance with packet loss status. The proposed algorithm smoothes short term variations in loss rates, while responds quickly to real changes in the loss rate of the traffic. The ability of the algorithm is qualified by the agility, stability and avoidance of frequent switching for two sample files.

Key words: loss rate 、 traffic estimation 、 RTP 、 RTCP



壹、緒論

本研究提出一個以變動框架長度為基礎之遺失率估算法，並實作一個 RTP (Real-Time Transport Protocol) [1] 語音傳送接收系統，來驗證此演算法之實用性。本研究假設傳送端有二個內容相同但具有不同高低語音品質之檔案，傳送端依據所量測到的遺失率，來傳送較適合網路狀況的檔案。傳送端根據以變動訊框長度為基礎之遺失率估算法，來估算目前網路上封包遺失的狀況，並以此做為切換高低音質檔案的依據。此演算法可過濾短期突發交通，並呈現長期交通之趨勢。除具有穩定與敏捷的特性外，此演算法亦能避免音質切換過度頻繁。

一、研究背景與動機

傳統 IP 網路架構中，由於僅提供盡力傳送 (best effort) [2] 的服務型態，在傳送即時語音或影像等多媒體串流資料時，會令使用者聽到聲音或看到的影像，有不清晰或斷斷續續等情況。因此，無法依封包載送的類型或依權重，來進行不同速率或等級的傳送，因此也無法提供任何 QoS (Quality of Services) 保證 [2]。在網路上進行語音傳輸，一般作法是依不同網路狀況，進行不同語音品質之調節。因此，實務上多以終端系統為基礎，由終端系統自行處理有關延遲、遺失或延遲差異等 QoS 參數。目前許多研究機構持續進行研究，試圖將 IP 網路之 QoS 訂定標準，如：ITU-T SG12 [3]。

二、研究目的

目前在網路中對語音進行傳送，一般使用 UDP 和 Real-Time Transport Protocol (RTP) 做為傳輸協定[1]。在網路中傳送語音資訊，除了會造成原本音樂品質的失真外，更須考慮播放時的延遲、遺失及延遲差異情形。P.VTQ[3]為目前 ITU-T 探討 IP 網路上有關即時多媒體品質的估計標準所暫時訂定的名稱，其主要探討 RTP 與 RTP Control Protocol (RTCP) 中的 QoS 參數，如：封包遺失率、延遲及延遲差異等。在語音品質量測方面，從使用者終端實際接收情形進行品質估算，較能實際評估使用者之傾聽品質。主要影響使用者傾聽品質之 QoS，有延遲、延遲差異及遺失率等。而遺失率具有直接決定語音品質的最大因素。因此，本研究將藉由一個以變動框架長度為遺失率估算基礎之演算法，來判斷網路上的狀況，並以此來進行即時語音品質的切換。此演算法能過濾短期突發交通，呈現長期交通趨勢，除具有穩定與敏捷的特性外，亦能避免音質切換過度頻繁。



貳、文獻探討

一、語音品質評量

語音品質的評量最早用於評量語音合成或語音編碼的水準。主要由於語音訊號在儲存、處理及傳輸過程中，會被混入一些噪音，致使原始品質因噪音干擾而變得不真實或品質變差。評量語音品質的方法，可分為主觀與客觀的評量方法 [4]。主觀品質評量 (Subjective Quality Measure) 是指透過一套標準的程序，由一些訓練過的受測者來聽，並且進行評分。以分數的好壞，來決定聲音品質。較常被使用的為平均意見分數 (Mean Opinion Score, MOS)。MOS 是一種讓受測者主觀認定聲音品質，依聽到的聲音分五個等級，進行 1 到 5 的評分。唯獨須先訓練受測者在什麼品質下，評定多少分數。一般電話的語音品質的 MOS，在 3.0 左右 [5-7]。客觀品質評量 (Objective Quality Measure) 是利用演算法計算噪音成分、訊號的頻譜失真程度，以做為語音品質的評量，如：訊噪比 (Signal-to-Noise Ratio, SNR)。

VoIP 語音品質評量的分類可分為主觀和客觀兩類。主觀品質評量是最可靠的方法，但是卻相當地費時且耗成本。因此，可以藉由測量終端和網路的實體特性，來估計出主觀品質，以作為客觀品質評量的基本依據。主觀品質評量可分為下列兩種：1) 傾聽的 MOS：語音接收者所聽到的品質；2) 會談的 MOS：多人會談間的收聽品質。有關 VoIP 客觀品質評量的方法，依不同的觀點分類，如表 1 所示 [4]。意見模式 (Opinion models) 指利用有關網路及終端品質參數 (噪音、迴音、延遲及編碼失真等) 所產生會談的 MOS 估計，例如：E-Model [8] 從終端對終端 (End-to-End) 之間兩端的對話中，評量有關噪音、迴音、延遲、編碼失真及網路延遲差異等等，來決定網路傳輸要素的等級。E-Model 主要使用 R 值變數作為語音評量等級，範圍從 0 到 100。口語階層客觀模式 (Speech-layer objective models) 必須以原始語音資料輸入，產生傾聽的 MOS 估計，例如：PESQ [5] 是由接收方依所接收到的語音訊號跟原來聲音做比較。封包階層客觀模式 (Packet-layer objective models) 指針對網路 IP 封包特性 (遺失率、延遲或延遲差異等) 來產生傾聽的 MOS 估計。使用封包資訊作為語音品質評量有下列的優點：1) 可以即時量測品質；2) 不需要與原始語音品質比較，即可得知語音的品質；3) 以使用者終端實際收到的封包為評量標準，較符合使用者實際傾聽的品質。但是，前提則必須制定網路語音封包 QoS 的標準。P.VTQ [3] 為目前 ITU-T SG12 探討 IP 網路上有關即時語音品質的估計所暫時訂定的名稱，主要探討 RTP 與 RTCP 中的 QoS 參數，如：封包遺失率、封包遺失模式、延遲及延遲差異等。口語階層客觀模式與封包階層客觀模式，均可使用傾聽的 MOS 進行估計評量。但是，口語階層模式必須要有原始語音資料與實際接收的語音資料相比較，所以較適於在傳送端不易得知接收端之接收封包資訊 (遺失率或延遲等) 的情況。而封包階層模式，則僅須即時地從接收的封包中來進行估計。

表 1：客觀品質評價方法的分類

種類	意見模式 (E-Model)	口語階層客觀模式 (PESQ)	封包階層客觀模式 (P.VTQ)
目標	網路規劃	基準點／管理	管理
量測程序	預定值	Active／Passive	Active／Passive
輸入資訊	品質參數	語音訊號	IP 封包
估計品質	會談的 MOS	傾聽的 MOS	傾聽的 MOS

二、網路交通量測與估計方法

網路交通量測有三個指標，分別為準確性 (Accuracy)、敏捷性 (Agility) 及穩定性 (Stability) [9] 等。準確性指能精確量測出網路上的實際交通；敏捷性指可以快速反應交通流量的改變；穩定性則指能過濾突發的交通，以顯示長期的交通趨勢。因此，敏捷性及穩定性兩者是相對的關係。網路之量測，依測量方法分主動式 (Active) 與被動式 (Passive) 兩種 [9]。主動式量測，藉由主動從網路收集到的即時資訊進行分析運算後，推測網路擁塞情形或狀態，再進行適時之傳輸調整。許多研究文章探討 non-TCP flow 傳輸率的控制，例如：TCP-friendly rate control (TFRC) [10] 和其他 Equation-based 擁塞控制 [11, 12] 等。這些機制，均透過一個與 TCP 相似的連線，提供接收端資訊的回饋。傳送端以發送探測 (Probe) 封包的方式，來得知封包遺失率和 Round Trip Time (RTT) 的資訊，進而調節網路傳輸。被動式的方法則針對依據網路節點本身，記錄從開始至結束期間所擷取的網路資訊加以分析，進而推測網路狀態與封包接收情形。[13-17] 探討透過 Real-Time Control Protocol (RTCP) 控制機制，來得知接收端的回饋資訊。就時間的考量，主動式的方法著重於某一個時間點，而被動式的方法則著重在整個過程之分析。以區域性而言，主動式的方法考量了封包在網路兩個端點間傳送時所有可能的狀況，如：封包繞送。而被動式的方法則考量了在單一節點上接收時的實際接收狀況，如：延遲、遺失率或延遲差異等。

進行網路交通即時量測時，多以時間區間為週期進行量測。當時間週期夠大時，雖可去除異常值的產生，但相對卻不容易反應交通情況；而當時間周期很小時，雖可即時反映即時交通狀況，但卻會嚴重消耗網路資源，造成網路擁塞。在使用網路交通流量量測上，最通用的方法是使用遞迴估計，可分為 Time Sliding Windows (TSW) [13-15] 及 Exponentially Weighted Moving Average (EWMA) [14][18-19] 等兩種方法。TSW 提供一個矩形式資料權重函式，利用一個移動式的固定視窗長度，來針對網路交通量做估計。EWMA 為一種時間參數的遞迴估計，使用一個固定權重因子，以時間性的變動（每週期）來做估計。我們將 EWMA 及 TSW 的概念，應用到遺失率的計算，其敘述如下：

1. Exponentially Weighted Moving Average (EWMA)

假設 $LossRate_{old}$ 為前一次計算的遺失率， $LossRate_{now}$ 表示為目前量測的遺失率，其中 α 代表權重因子， $0 < \alpha \leq 1$ 。在網路上對於語音封包遺失率的量測，多以 EWMA

進行平滑處理（如公式(1)），以降低異常值的影響。以 EWMA 應用於語音品質調節為例，當使用 α 值過高時，容易引起振盪（Oscillation）。因此，實務上以較低的 α 值進行平滑處理（smoothing），但其缺點為無法呈現實際封包遺失率。

$$LossRate_{new} = LossRate_{old} \times (1 - \alpha) + LossRate_{now} \times \alpha \quad (1)$$

2. Loss Rate estimation with Fixed frame size (LRF)

此法引用 TSW 概念，使用固定框架長度（Frame Size）對封包遺失率進行估計，以便能以較平滑的方式，呈現網路交通的封包遺失率。假設以 s 秒為一個量測單位，一個框架包含 n 個量測單位，其中 $n \geq 1$ ，我們以 $FS = n$ 來表示框架長度。假設 $LossRate_t$ 為第 t 時間估算的遺失率， X_t 為第 t 時間，所量測到的遺失率，我們可以得到公式(2)來估算封包遺失率：

$$LossRate_t = \frac{(LossRate_{t-1} \times FS + X_t)}{FS + 1} \quad (2)$$

在此公式中， X_t 的權重為 1，而第 $t-1$ 時間的遺失率（ $LossRate_{t-1}$ ）權重為 FS 。當 FS 愈大時，則表示所得到的曲線愈平滑。

參、以變動框架長度為基礎之遺失率估算法

本文提出一個改良 LRF 的演算法—以變動框架長度為基礎之遺失率估算法（Loss Rate estimation with Variable frame size, LRV）來估算目前網路上封包遺失的狀況。我們以框架長度為交通狀況量測單位，透過加大框架長度及遞減框架長度，來達到最佳的穩定及敏捷性。除此之外，我們也加入考量切換次數的因素，以作為獲得更佳切換的依據。

假設 FS 代表目前框架長度， Max_FS 及 Min_FS 為所允許的最大與最小框架長度。其中， Max_FS 及 Min_FS 可分別視為最高平滑度與最大敏捷度之指標。 $Upper$ 與 $Lower$ 分別代表高、低音質切換的遺失率上界及下界。我們將目前框架（ FS ）及最大框架（ Max_FS ）之間的差額，分成 K 等份， ADD_FS 為框架長度之增量，也就是 $ADD_FS = (Max_FS - FS) / K$ 。假設 $HighAudioQuality$ 為一個布林變數，代表目前所使用之音質是否為高音質。當 $HighAudioQuality$ 為 true 時，代表高音質；若為 false，則為低音質。 $LossRate(Min_FS)$ 與 $LossRate(FS)$ 分別代表使用 Min_FS 及 FS 的框架長度來計算封包遺失率的函式。我們利用 $count$ 值來計數超過門檻值的次數。當連續超過門檻值（ $Upper$ 或 $Lower$ ） C 次，即 $count \geq C$ 才會進行切換。在每次切換語音品質時，演算法會加大 FS 框架長度。每使用一次， FS 即遞減一次，直至等於 Min_FS 為止。若在遞減過程中，又出現切換動作，則會再加大框架長度，以防止音質切換過度頻繁。演算法如下：

Initialization :

$count = 0;$




```

    HighAudioQuality = true;
On each frame time :
    Calculate LossRate(Min_FS);
    Calculate LossRate(FS);
    if (FS > Min_FS)
        {FS--; }
        if (HighAudioQuality == false) //低音質檔案
            {LossRate = Max(LossRate(Min_FS), LossRate(FS));}
        else //高音質檔案
            {LossRate = Min(LossRate(Min_FS) , LossRate(FS));}
    if(((LossRate ≥ Upper)&&(HighAudioQuality==true))||
        ((LossRate ≤ Lower)&&( HighAudioQuality == false)))
        { coun ++;
          if (count ≥ C)
              {
                  ADD_FS=(Max_FS - FS)/ K;
                  FS=FS+ADD_FS;
                  count=0;
                  if (HighAudioQuality == true)
                      { HighAudioQuality = false;}
                  else {HighAudioQuality = true;}
                  Switch audio file;
              }
          }
    else{ count=0;}

```

透過演算法所計算的結果，遺失率會落在低音質、高音質及緩衝區等三個區間之一。當 $LossRate$ 高於 $Upper$ 且 $count \geq C$ 時，語音品質將切換為低語音品質；當低於 $Lower$ 時且 $count \geq C$ 時，則切回高語音品質。緩衝區為高、低音質切換之緩衝區域，可能為高音質或低音質。不同音質中的估算方式，依 FS 遞減的與否來區分。當 FS 處於最小值 ($FS = Min_FS$) 情況下，無論遺失率如何或使用那個函式，其計算結果均相等。然而，在 FS 遞

減情況下 (FS 未等於 Min_FS)，當處於低音質時，透過 $Max(LossRate(Min_FS), LossRate(FS))$ 取得最大值。在遺失率上昇時，則所取的值為 $LossRate(Min_FS)$ ，因此能敏捷反應較高的遺失率；在遺失率下降時， FS 尚處於遞減階段，所取的值為 $LossRate(FS)$ 。亦即系統以較高平滑的方式來處理，以避免過快切換。當處於高音質時，則透過 $Min(LossRate(Min_FS), LossRate(FS))$ 取得最小值。在遺失率下降時，能敏捷反應較低的遺失率；在上昇時，則以較高平滑的方式，避免過快切換。也就是在低音質

緩衝區時，遺失率的估算為上昇快、下降慢；在高音質緩衝區時，則上昇慢、下降快。 $LossRate(Min_FS)$ 與 $LossRate(FS)$ 在 FS 框架長度加大（音質切換）後，兩者才會有所差異。在切換音質後， FS 呈遞減狀態， FS 值會愈來愈接近 Min_FS 值，最後 $LossRate(FS)$ 等於 $LossRate(Min_FS)$ 。因此，此演算法機制，在發生音質切換後，才會啟動高度平滑機制，否則僅以 Min_FS 進行一般性的平滑處理。主要目的除呈現實際的遺失率外，並期望在音質切換中，透過 FS 加大作用，減緩過快切回的情況避免振盪（Oscillation）產生，以及遞減作用能逐漸提高切換的敏捷度，以便適時切換音質。

肆、系統架構

本系統架構的功能模組，主要由傳送端與接收端所組成，如圖 1 及圖 2。分別說明如下：

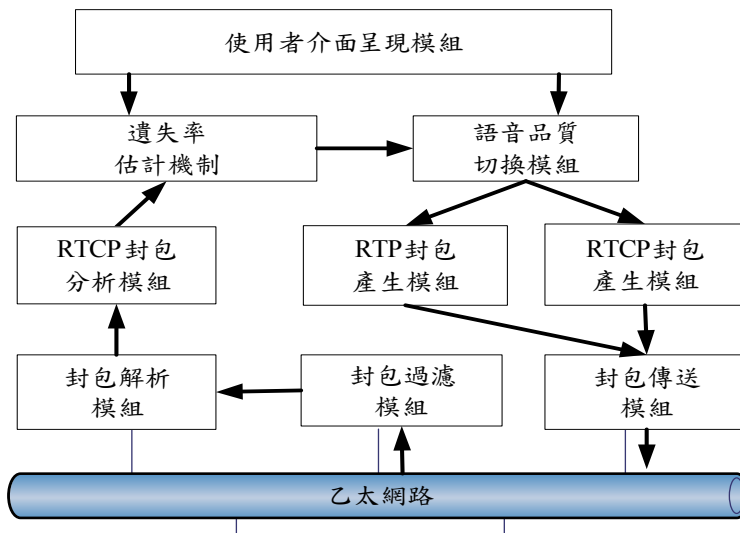


圖 1：傳送端架構

一、RTP 傳送端架構

• 封包過濾模組

當擷取到 Ethernet 封包，此模組便會記錄擷取時間。此模組的功能，僅在過濾擷取的封包是否正常，以便進行保留或丟棄。

• 封包解析模組

本功能模組主要針對封包過濾模組所擷取到的封包，進行逐層解析到傳輸層。依據實體層的原始資料，再將各層所屬的封包類別，一一做解析及給予實例化。此模組

將擷取到的資料，解析成 IP，接著再依據 IP 的表頭，辨別其上層是否為 UDP 封包。在逐層的解析中，由使用者介面中，可以決定是否要逐層丟棄不需要的封包。以 RTP 封包為例，即表示在網路層中，即僅留 UDP 及 TCP 協定的封包，到了傳輸層則僅留下 UDP 封包。因此，此功能模組，將封包解析到僅剩 UDP 封包。至於 RTP 的解析及判別，則屬另一功能模組。

• RTCP 封包分析模組

本模組主要將所獲得的 UDP 封包，在去掉表頭後，進一步解析出接收端所傳送的 RTCP 封包。在傳送端所須擷取之封包以 RTCP 中的 Receiver Report (RR) 封包為主。而 RTP 並沒有公認的埠號，本文為便於研究，以 9000 為 RTP 的埠號，而 RTCP 則為 9001 (RTP 的埠號+1)。所以，此模組首先辨別此 UDP 封包埠號是否為 9001 (RTCP)，接著再進行進一步辨別所載送之內容，是否符合 RTCP 的 RR 封包規定，接著再解析其內容 (遺失率)，以提供檔案切換機制模組使用。若不符合，則進行丟棄。

• 遺失率估計機制

從 RR 封包中所讀取出的接收端所計算的封包遺失率，判斷其封包遺失率是高或低於門檻值，並確認目前傳送之音質為高或低，再進行適當之音質切換。有關此機制請參閱第三節 LRV 估計演算法。

• 語音品質切換模組

此模組主要功能為依據遺失率估計機制的結果，再決定是否改變目前音質的轉換。因此，主要能針對語音編碼格式的轉換、播放點或取樣率等進行設定。在進行傳送端檔案語音品質切換時，必須先記錄目前媒體播放位置，並將其停止 (stop) 資源的佔有，以便順利將媒體的來源轉為欲切換的檔案。接著依公式(3)計算出目前媒體播放的時間點，再進行播放 (start)。假設有兩個不同媒體資料格式而相同內容的音樂檔 A 及 B，而要將目前播放之音樂檔 A 切換為 B 時，若希望播放的時間點是音樂檔 A 的停止點開始，而不是重頭播放，則必須考慮兩個不同媒體資料格式間的播放時間計算。時間點的計算跟時間戳記一樣，初始值為亂數且呈線性時間遞增。我們假設 MT (Media Time) 代表音樂檔 B 的播放時間點， $TBST$ (Time-Base Start-Time) 為音樂檔 A 的播放啟始時間基準點， TBT (Time-Base Time) 為音樂檔 A 的停止時之時間點， MST (Media Start-Time) 則為音樂檔 B 的播放啟始時間基準點， $rate$ 為媒體播放的比率值。 MT 的計算如下[1]：

$$MT = MST + (TBT - TBST) \times rate \quad (3)$$

• RTP 封包產生模組

此模組主要透過 Java Media Framework (JMF) 中的 Processor 類別讀取音樂檔案及轉換為 RTP 封包格式，再透過 DataSink 類別進行內容之讀取及傳送介面管理。

• RTCP 封包產生模組

依據 RTP 封包產生模組，所產生之 RTP 封包，進行傳送封包數與位元的統計、同步時間及時間戳記等，一併封入 Sender Report (SR) 封包。

- 封包傳送模組

此模組主要用來傳送 RTP 資料封包或 RTCP 的 SR 封包。

- 使用者介面呈現模組

此功能模組提供傳送端使用者之操作或是觀看的整個操作介面。除能控制檔案傳送格式外，也提供顯示接收到的 RR 報表的內容，以及統計圖表的呈現。

二、RTP 接收端架構

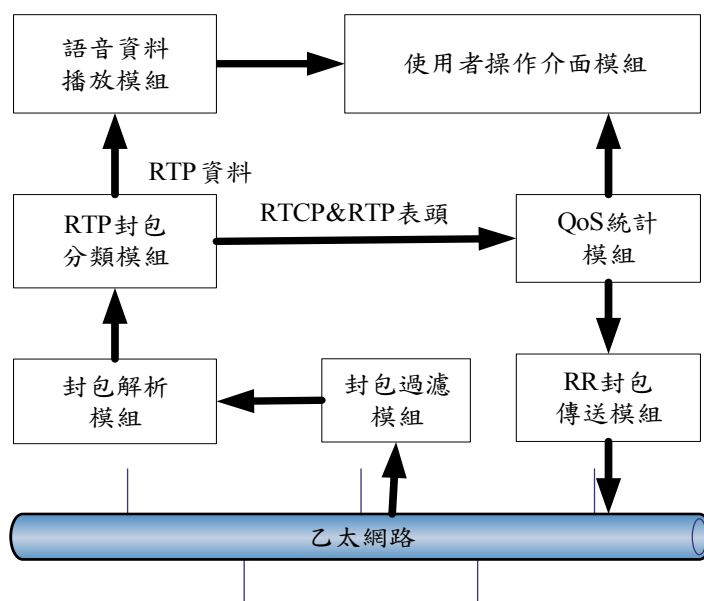


圖 2：接收端架構

- 封包過濾模組

與傳送端之封包過濾模組相同。不同之處，在於接收端會將過濾後的正常封包，放入由 Java 所實作的 vector 類別 packets 中，並記錄擷取時間，以提供後續 QoS 資訊統計。

- 封包解析模組

與傳送端之封包解析模組相同。

- RTP 封包分類模組

與傳送端之 RTP 封包分類模組類似。但是，在接收端則會將封包依埠號分類為 RTP (port number = 9000) 及 RTCP (port number = 9001) 封包，以便提供 QoS 統計模組使用。

- 封包 QoS 統計模組

本模組針對 RTP 封包進行相關的遺失、延遲及延遲差異的計算，並將結果儲存成檔案，以便提供數值給圖表使用。

- 使用者操作介面模組

此屬於接收端的使用者操作介面。如：系統操作面板環境、音樂播放、各種功能的選取及 QoS 的統計圖表等。

- 語音資料播放模組

本模組主要透過 Java 中的 JMF 套件撰寫而成。將擷取到的 RTP 封包依序置放到 bytes 串流緩衝區，再透過一個 Player 物件來播放。因此本模組非本文的重點。在設計上，僅使用基本的緩衝控制來播放 Wave 音樂。採用的播放方式為停滯同步控制（Blocking Synchronization Control）。此模組對於在應播的封包資料尚未到達前，實施暫停（Blocking）播放，直到接收到下一個應播的封包資料為止。只要封包晚到即進行丟棄，所以並不進行重播（Re-display）的動作。

- RR 封包傳送模組

依統計模組所計算的接收端 QoS 狀況產生一個 RR 封包，每隔約五秒對傳送端做回報。例如：在接收端，可由 SR 封包得知 RTP 封包的初始編號，再減去目前所收到的最大 RTP 封包編號，即可得到應該接收到的封包數，以此計算出遺失率，放入 RR 封包內。而傳送端即依 RR 封包的 QoS 狀況，來做為不同音質檔案切換的準則。

伍、模擬實驗

本實驗主要分為三個部份，分別為 EWMA 及 LRF 之比較、LRV 參數實驗、以及 EWMA、LRF 與 LRV 之比較。實驗電腦設備，如表 2 所示。實驗環境如圖 3，以一般 PC 作為資料傳送端，並使用 TCP 封包作為干擾因子；另一台 PC 則做為接收端，用來接收 RTP 和 TCP 封包。筆記型電腦作為 RTP 資料傳送端，能傳送不同音質的音樂檔。在區域網路中，各電腦以 10/100 Mbps Switch 相連接，乙太網路速度最高為 100 Mbps。

表 2：實驗電腦設備

	TCP 資料傳送端	TCP/RTP 接收端	RTP 傳送端
種類	一般 PC	一般 PC	筆記型
CPU	Intel 2.0G Hz	Intel 1.0G Hz	Intel 1.0G Hz
RAM	256MB	256MB	512MB
OS	Windows XP	Windows XP	Windows XP

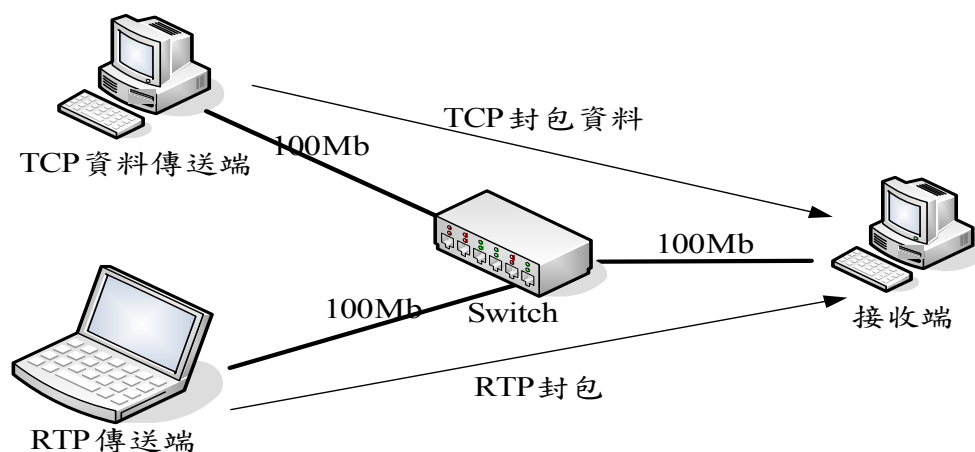


圖 3：實驗架構圖

實驗一：EWMA 及 LRF 之比較

此實驗主要針對 EWMA 及 LRF 等二個演算法來進行比較，其目的在於觀察各參數對於演算法的影響，以及比較 EWMA 與 LRF 間的差異。實驗之網路交通，總共使用四次 TCP 資料進行干擾，干擾時間區段如表 3。所使用的語音資料格式為 PCM，封包之基本資料如表 4。

表 3：網路交通干擾之時間區段

干擾順序	時間 (秒)		秒數
	開始	結束	
第一次	60	90	30
第二次	170	230	60
第三次	350	410	60
第四次	470	590	120

表 4：RTP 語音封包基本資料

項目	內容
資料類型	MPEG Audio/RTP
取樣率 (Hz)	44,100
封包數/秒	6
IP 封包大小 (Bytes)	1,402
封包平均傳輸率 (KB/s)	8.22
封包平均到達間隔時間 (ms)	166.7
Timestamp 的增值	15,120

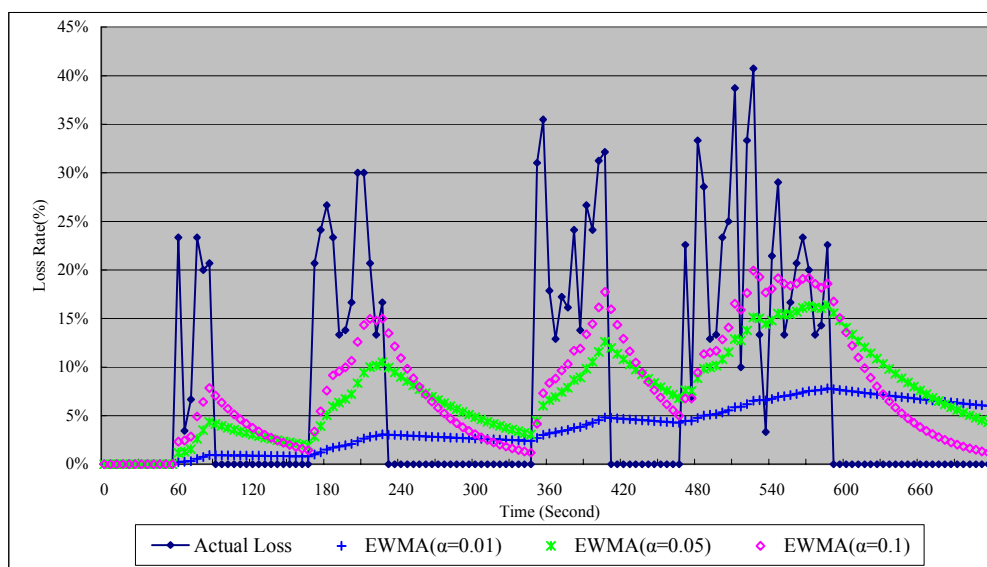


圖 4：EWMA

實驗過程中，EWMA 的 α 值，分別使用 0.01、0.05 及 0.1。LRF 的 FS 值，則分別為 10、20 及 100。實驗結果，如圖 4 及圖 5 所示。在兩圖中，EWMA ($\alpha=0.01$) 與 LRF ($FS=100$) 均為過度平滑穩定的曲線，難以呈現網路交通。EWMA ($\alpha=0.05$)、EWMA ($\alpha=0.1$)、LRF ($FS=10$) 與 LRF ($FS=20$) 等，則都能敏捷地反應實際交通。當 EWMA 之 α 值愈大或者 LRF 之 FS 值愈小時，估計的遺失率權重比率會愈高，敏捷性也愈高，穩定性則相對會愈低，亦即 EWMA 權重值 (α) 過大或 FS 過小，都會引起較大振盪 (Oscillation)。在圖 4 及圖 5 中，我們可以發現 EWMA ($\alpha=0.01$) 及 LRF ($FS=100$) 在估計值與實際值之間相差甚大，難以呈現網路交通之實際遺失率。實務上，我們會應用偏低 α 值或偏高 FS 值估算遺失率，以利用平滑的穩定特性，來避免振盪產生，但相對卻難以呈現實際封包遺失狀況。圖 6 中，顯示了 EWMA($\alpha=0.01$) 與 LRF ($FS=100$)、EWMA ($\alpha=0.05$) 與 LRF ($FS=20$)，以及 EWMA ($\alpha=0.1$) 與 LRF ($FS=10$) 等三組曲線。這是因為 EWMA 與 LRF 兩者是類似的估計法，其差別僅在於 EWMA 使用權重值，而 LRF 則使用框架長度來作估算。

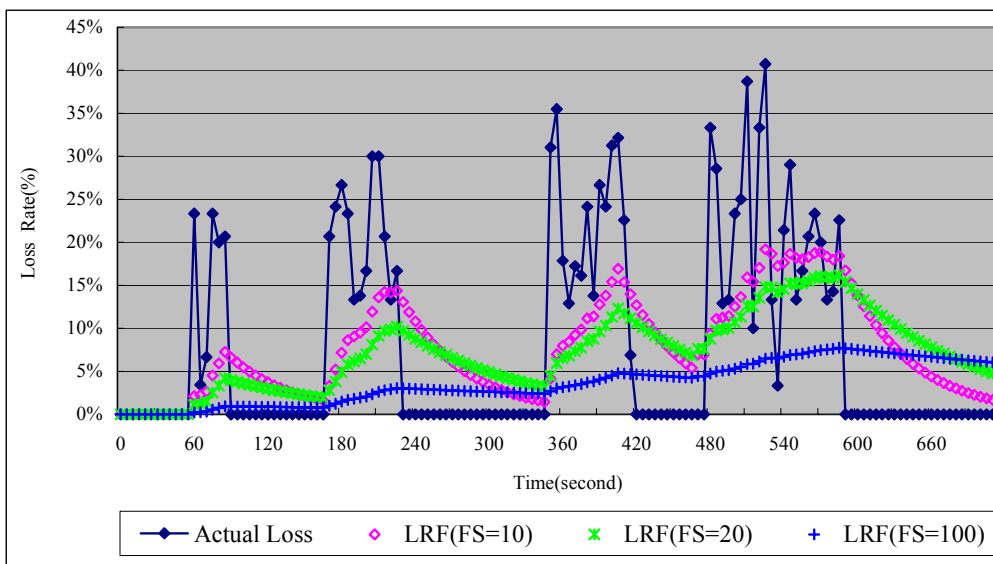


圖 5：LRF

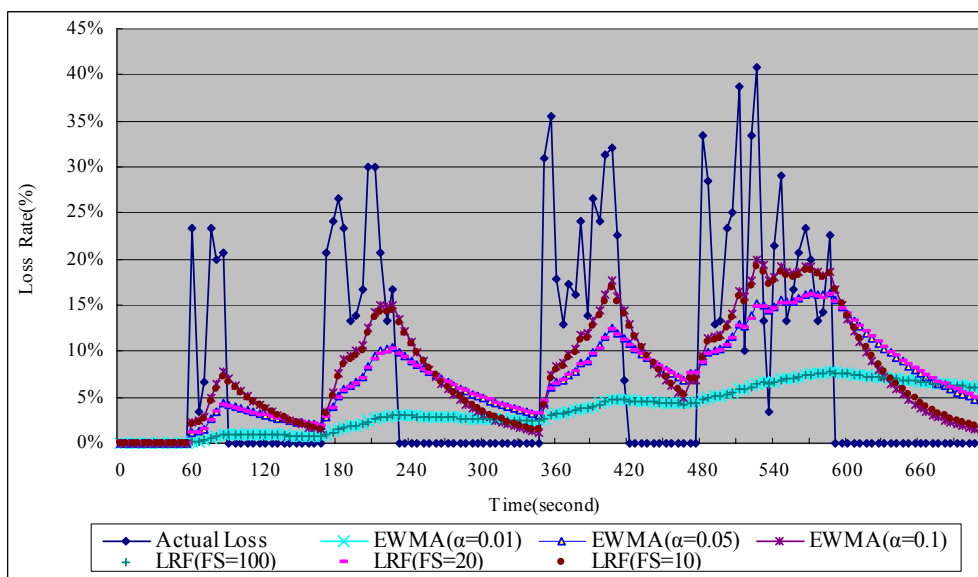


圖 6：LRF 與 EWMA 之比較

實驗二：LRV 參數實驗

此實驗主要針對 LRV 演算法的 Min_FS 、 Max_FS 、 K 及 C 值等四個參數設定來進行實驗，以便顯示各參數對於 LRV 遺失率估計演算法之影響，並且找出 LRV 演算法

中參數的適當值，以便在後續實驗中對各演算法進行比較。為便於實驗的進行，我們使用 TCP 交通 50% 網路負載作為網路之干擾。本實驗總共進行五次網路交通干擾，網路交通之時間區段如表 5。所使用的語音資料格式為 PCM。封包之基本資料，同表 4。

表 5：網路交通之時間區段

干擾順序	網路負載 (TCP 資料)	時間 (秒)		秒數
		開始	結束	
第一次	50%	60	120	60
第二次	50%	180	240	60
第三次	50%	300	360	60
第四次	50%	450	510	60
第五次	50%	600	660	60

在實驗中，音質切換之上、下限設定為 $Upper=10\%$ 、 $Lower=5\%$ [17]。實驗分為 Min_FS 與 Max_FS 、 Min_FS 與 K 、 Max_FS 與 K 、 C ，以及 $Upper$ 與 $Lower$ 等五組，分別探討其對 LRV 演算法之影響。

第一組： Min_FS 參數與 Max_FS 參數

此組主要實驗 Min_FS 與 Max_FS 參數之間的變動，對 LRV 演算法之影響。基本參數設定為 $K=6$ 、 $C=1$ 、 $Upper=10\%$ 及 $Lower=5\%$ 。首先將 Min_FS 視為變動參數及將 Max_FS 作為固定參數。實驗結果，如圖 7 至圖 9。從圖 7 中可以發現，當 $Min_FS=8$ 時，整個曲線很容易隨著網路交通干擾產生完全振盪。當 $Min_FS=32$ 時，則呈現相當平滑的曲線，也改善了振盪的幅度，但卻難以反應網路交通。主要是由於在 LRV 估計演算法中， Min_FS 為最小的框架長度，可視為反應網路交通的最佳值。所以， Min_FS 值愈小，愈容易反應網路交通，也愈容易造成振盪。圖 8 中，在加大 $Max_FS=80$ 後，振盪幅度則有稍微改善。從圖 9，可發現振盪幅度的改善更明顯。但是，過於加大 $Max_FS=160$ 的後果，則造成了 FS 遞減階段的時間過長，反而無法呈現出切換後的真實網路交通。在第一次網路干擾下， $Min_FS=8$ 曲線很快進行切換 (FS 從 8 增加為 33)，但由於抑制作用（框架增量加大）過大，因而難以立即反應後續無網路干擾的交通狀況。在第三次交通干擾下，也因抑制作用過大，無法估計出網路的干擾，直至抑制作用消除 (FS 已遞減至等於 Min_FS) 後，才能快速反應第四次的網路干擾，此為不完全振盪。 $Min_FS=16$ 及 $Min_FS=32$ 兩者均比 $Min_FS=8$ 曲線平滑，不易造成切換。因此，在 Max_FS 較大的情況下， Min_FS 的變動影響對 LRV 較為明顯。主要是由於 Max_FS 愈大，在不同 Min_FS 參數下，所造成的抑制作用差異性也愈大。所以在 LRV 估計演算法中， Max_FS 能決定切換時的抑制作用。

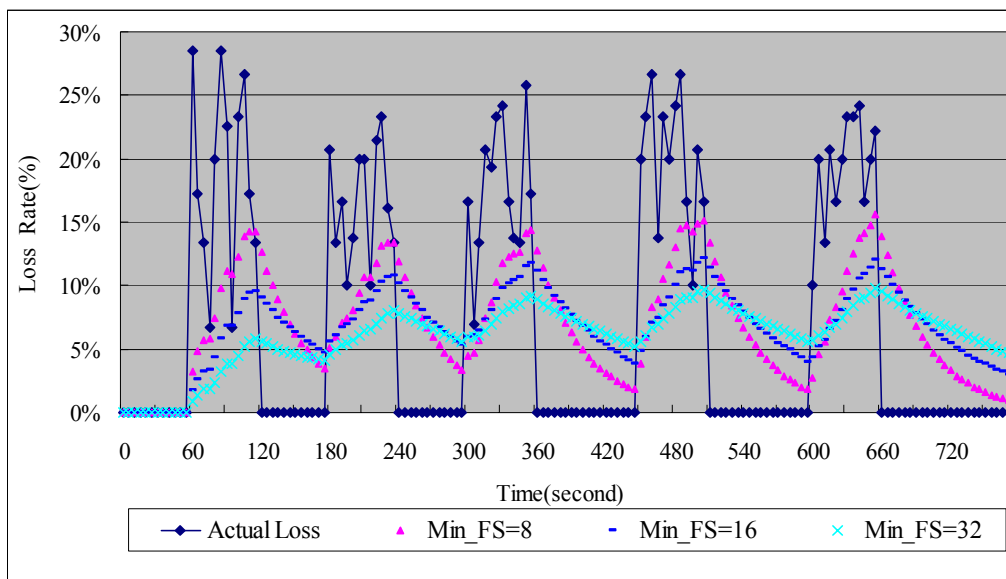


圖 7：LRV 演算法在 $Max_FS=40$ 下， Min_FS 變動之影響

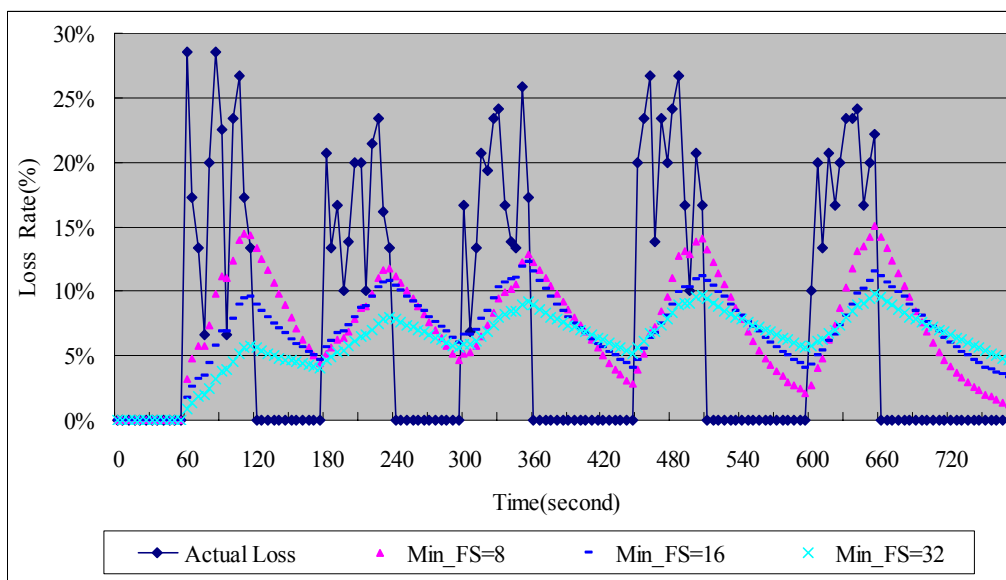


圖 8：LRV 在 $Max_FS=80$ 下， Min_FS 變動之影響

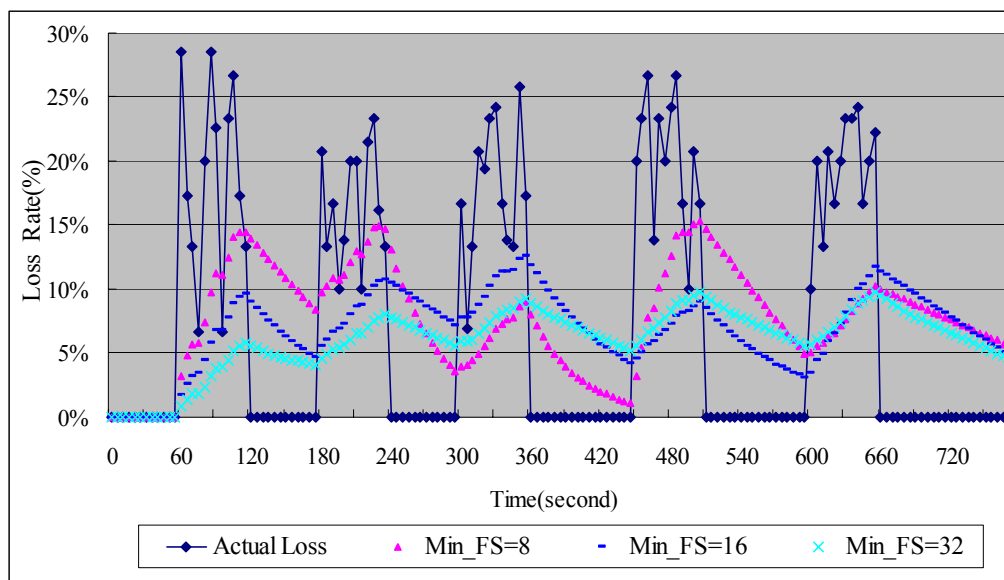


圖 9：LRV 在 $Max_FS=160$ 下， Min_FS 變動之影響

接著我們以 Min_FS 為固定參數， Max_FS 為變動參數進行實驗。實驗結果如圖 10 至圖 12。由於 Min_FS 為固定參數，圖 10 至圖 12 等各曲線，在未進行切換前，為無異之曲線。圖 10 中的 $Max_FS=80$ 及 $Max_FS=160$ 等曲線，在第一次網路干擾（第 60 秒至 90 間）到達切換標準後，即開始產生差異。而圖 11 中，則在第二次網路干擾中切換，才有所差異。在圖 12 中，則 $Min_FS=32$ 過大，形成三條曲線重疊，其原因在於沒有切換產生。所以， Max_FS 必須要在產生切換時，才能產生作用。

從圖 10 中可發現， $Max_FS=40$ 曲線由於切換後抑制作用的效用不大（框架增量小）及時間持續不久（框架長度遞減至 Min_FS 快）。所以仍可以敏捷地反應出網路交通。但卻容易產生振盪。而 $Max_FS=160$ 曲線，則由於切換後抑制作用的效用過大（框架增量大）及時間持續過久（框架遞減至 Min_FS 時間較久），則難以呈現出切換後的真實網路交通，直至抑制作用消除（第 210 秒 FS 遞減至 Min_FS ）後，才能快速反應網路交通。觀察圖 10 三條曲線，可得知當 Max_FS 愈大，抑制作用（框架增量）會愈大，曲線也愈平滑。尤其在連續切換（ FS 尚未遞減至 Min_FS 時，即再次切換）下，抑制作用的效果及時間性將會累加。在圖 11 中，三條曲線差異性並不大。主要是由於提高 Min_FS 後，三條曲線將更平滑，因此 Max_FS 影響效果變小。

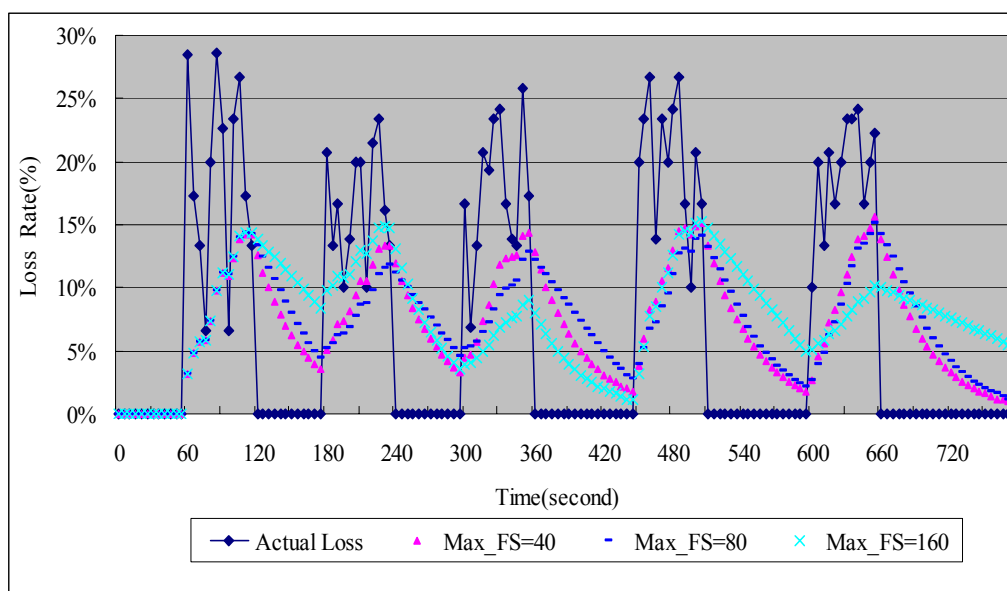


圖 10：LRV 在 $Min_FS=8$ 下， Max_FS 變動之影響

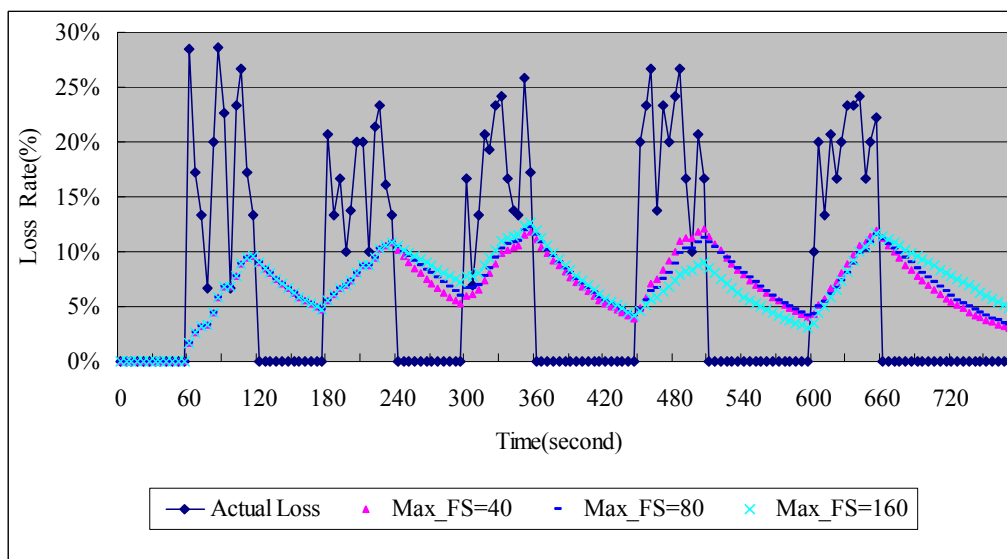
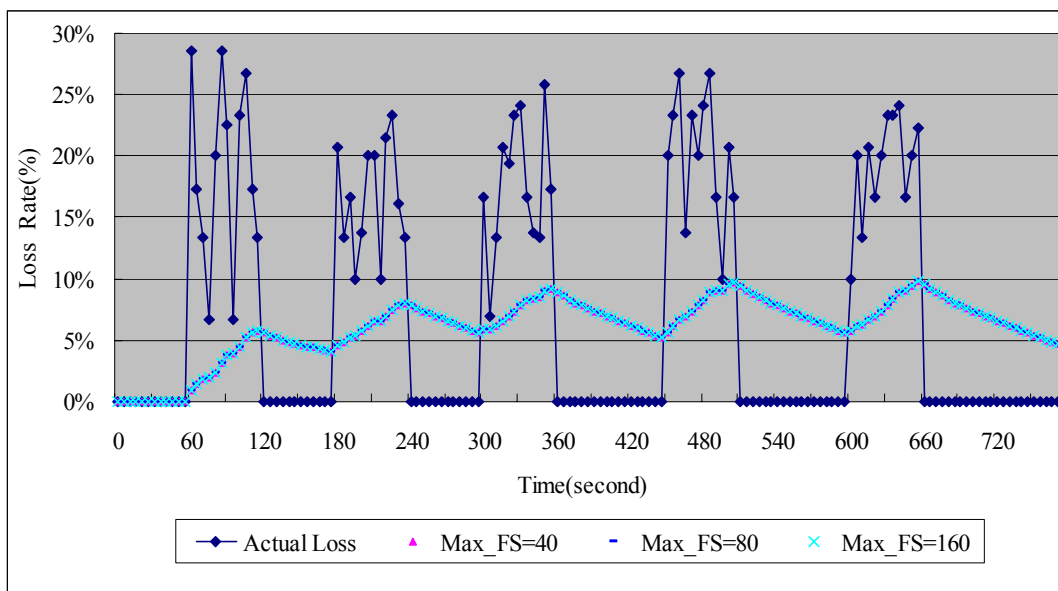
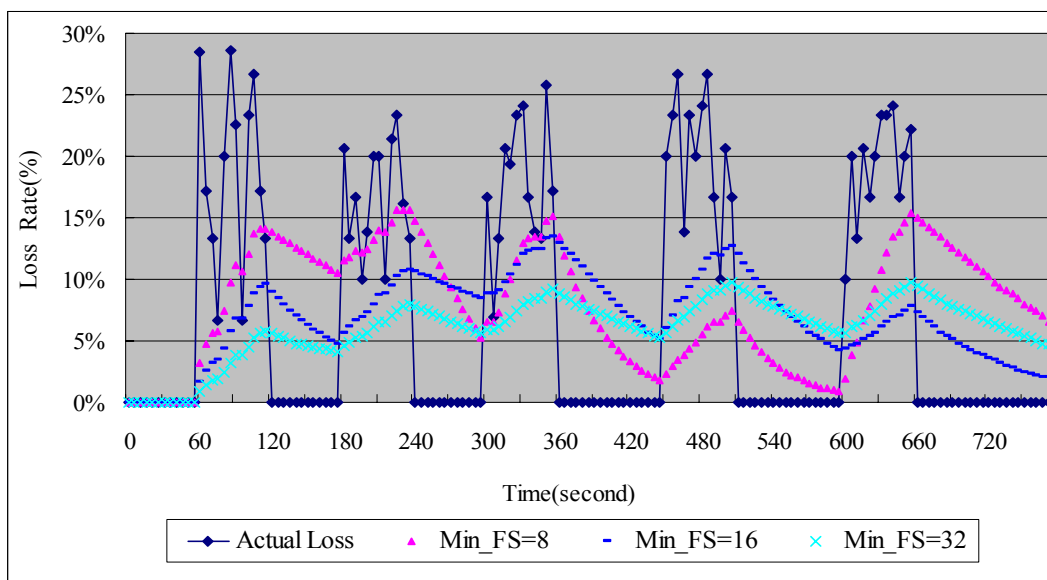


圖 11：LRV 在 $Min_FS=16$ 下， Max_FS 變動之影響

圖 12：LRV 在 $Min_FS=32$ 下， Max_FS 變動之影響圖 13：LRV 演算法在 $K=2$ 下， Min_FS 變動之影響

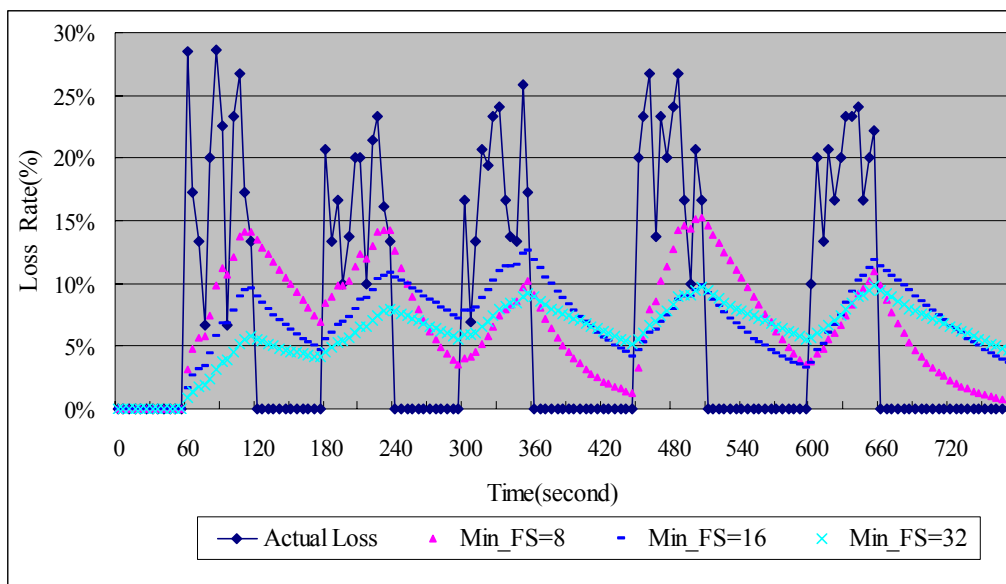


圖 14：LRV 在 $K=4$ 下， Min_FS 變動之影響

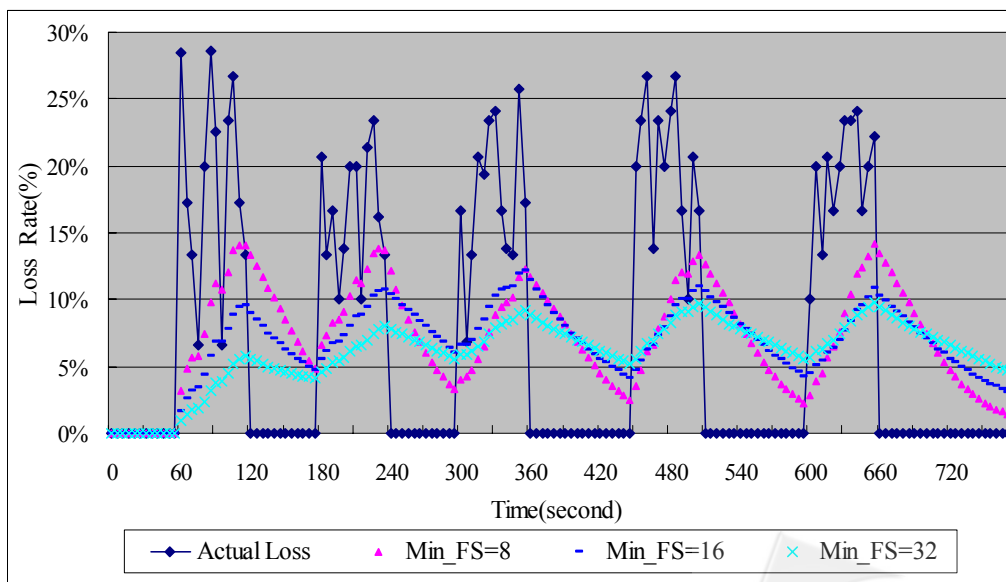


圖 15：LRV 在 $K=6$ 下， Min_FS 變動之影響

第二組：Min_FS 參數與 K 參數

此組主要實驗 Min_FS 與 K 之間的變動對 LRV 演算法之影響。基本參數設定為 $Max_FS=80$ 、 $C=1$ 、 $Upper=10\%$ 及 $Lower=5\%$ 。我們首先設定 Min_FS 為變動參數， K 為固定參數，實驗結果如圖 13 至圖 15。從圖 13 可知， $Min_FS=8$ 於第一次網路干擾（第 60 秒到第 120 秒）到達切換標準後，音質切換為低音質，一直到第 360 秒至 450 間（網路交通正常），才又切換為高音質。其原因主要是抑制作用過大。當 $K=2$ 時，框架增量為 Min_FS 與 Max_FS 差值的一半。 $Min_FS=16$ 整體曲線呈現比 $Min_FS=8$ 較為平滑，兩者均屬於不完全振盪類型。 $Min_FS=32$ 曲線因過於平滑而未發生切換。圖 14 與圖 15 也有類似情形。

接著以 Min_FS 為固定參數， K 為變動參數。實驗結果如圖 16 到圖 18。從圖 16 與圖 17 中看出，當 K 值愈大，抑制作用就愈小，所形成切換後的曲線即愈敏捷。實際上， Max_FS 與 K 參數兩者均會影響切換的框架增量。在圖 17 中，由於 $Min_FS=16$ 具有高平滑度，因此三條曲線的變化，並不如圖 16 的變化大。在圖 18 中，則是由於 $Min_FS=32$ 過度平滑而未切換，因此，所有曲線並無差異。

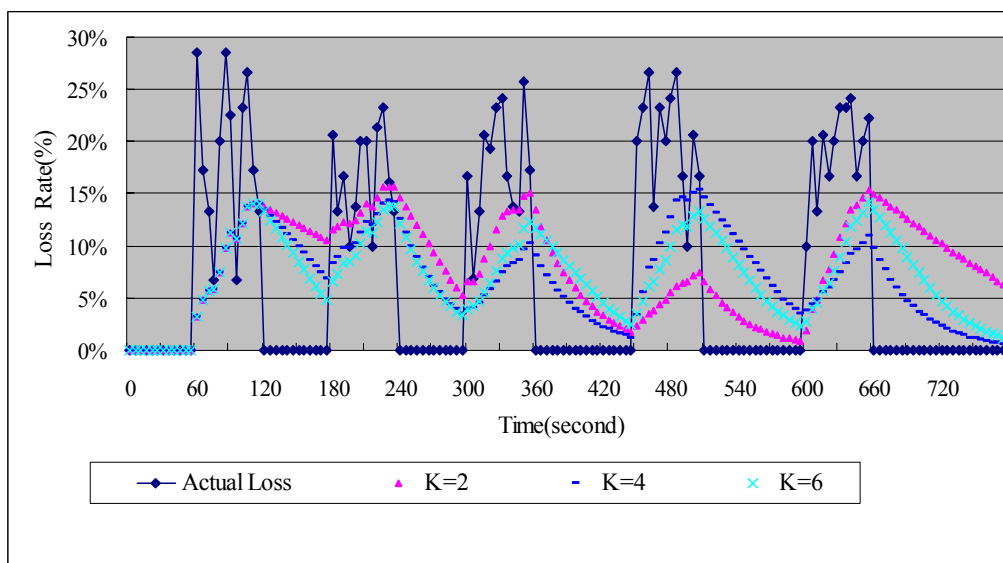
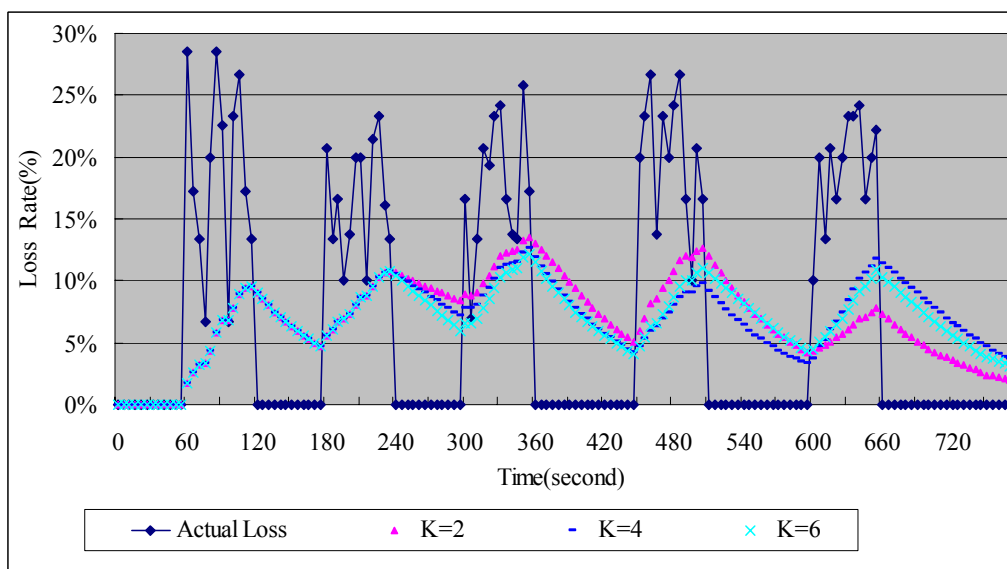
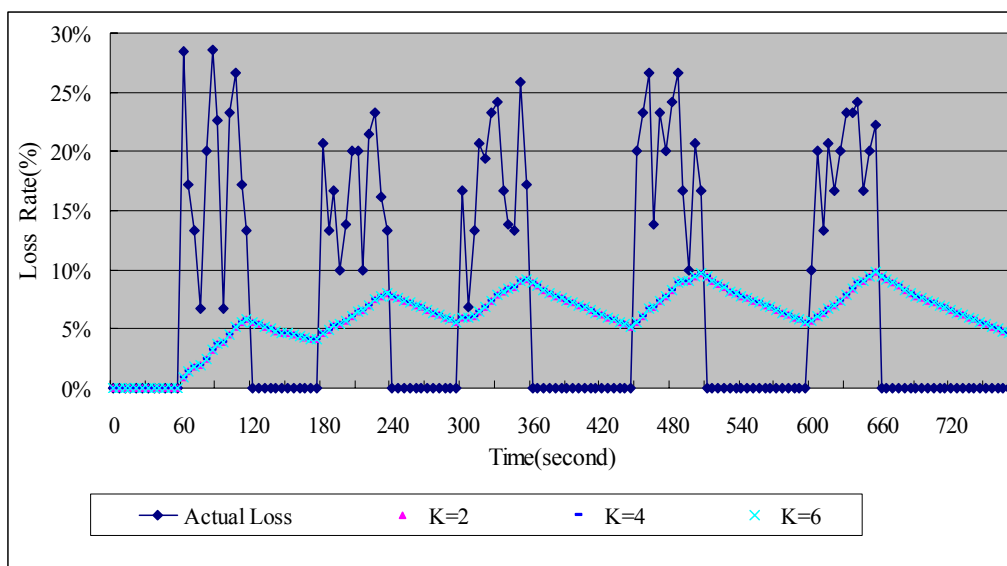


圖 16：LRV 在 $Min_FS=8$ 下， K 變動之影響

圖 17：LRV 在 $Min_FS=16$ 下， K 變動之影響圖 18：LRV 在 $Min_FS=32$ 下， K 變動之影響

第三組： Max_FS 參數與 K 參數

此組主要實驗 Max_FS 與 K 之間的變動對 LRV 演算法之影響。為較易引發切換，我們使用 $Min_FS=8$ 進行實驗，以便更容易觀察兩者之關係。其他基本參數設定為 $C=1$ 、 $Upper=10\%$ 及 $Lower=5\%$ 。我們首先將 Max_FS 視為固定參數， K 為變動參數。實驗結果如圖 19 至圖 21。從圖 19 中可發現， K 值的改變對於曲線的改變並不大，主

要由於各曲線的框架增量差異很小。當 $Max_FS=40$ 及 $Min_FS=8$ 時，在第一次切換後的框架增量，以 $K=2$ 、4 及 6 計算，則分別為 16、8 及 5。而在圖 20 及圖 21 中，隨著 Max_FS 加大後， K 參數對曲線的影響程度隨之變大。若使用 $Max_FS=80$ 計算，則框架增量分別為 36、18 及 12。若以 $Max_FS=160$ 計算，則分別為 76、38 及 25。同時我們發現當 K 值愈小，切換後的曲線也更平滑，直至抑制作用消除。因此，當 Max_FS 愈小， K 的變動對 LRV 的影響愈小；反之亦然。

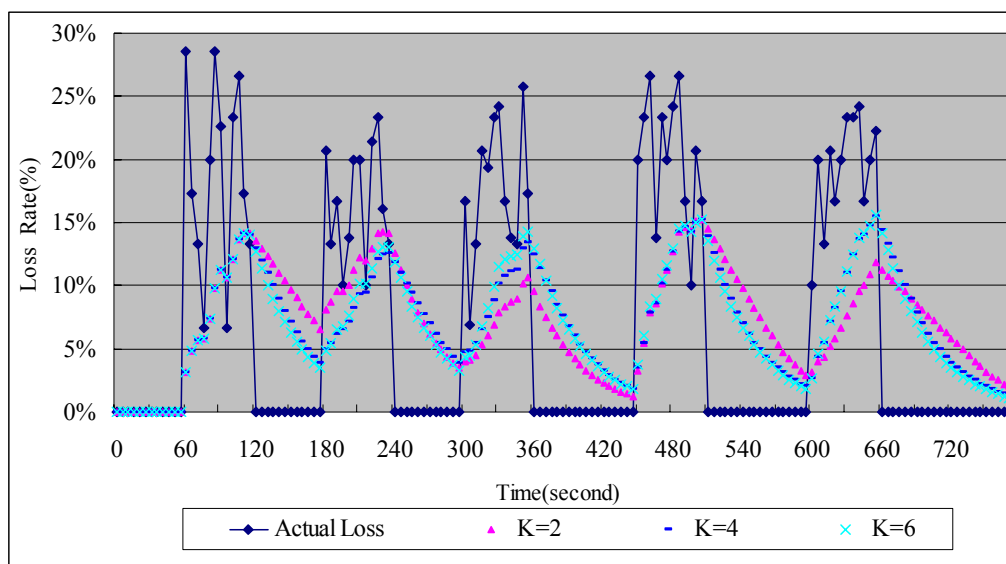


圖 19：LRV 演算法在 $Max_FS=40$ 下， K 變動之影響

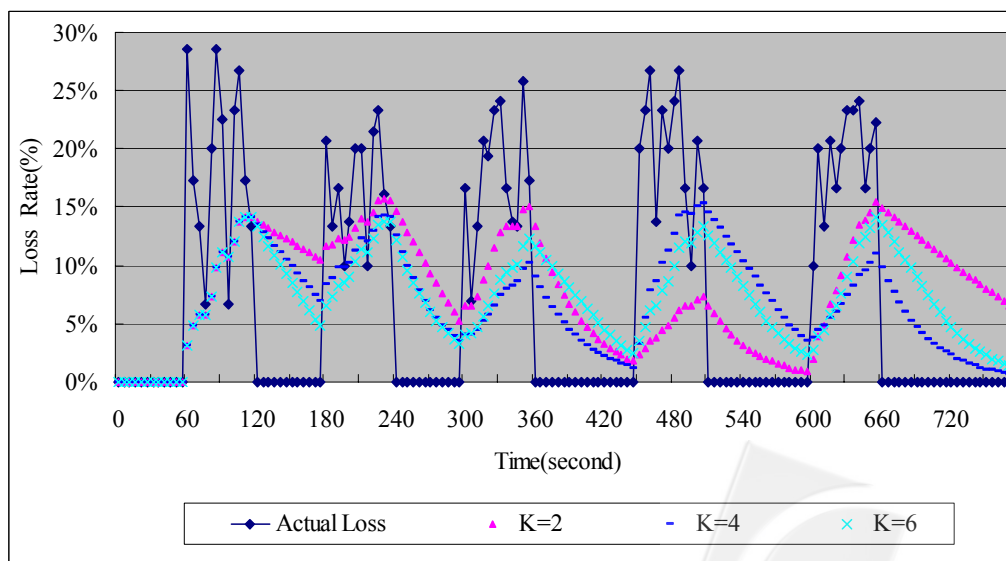


圖 20：LRV 在 $Max_FS=80$ 下， K 變動之影響

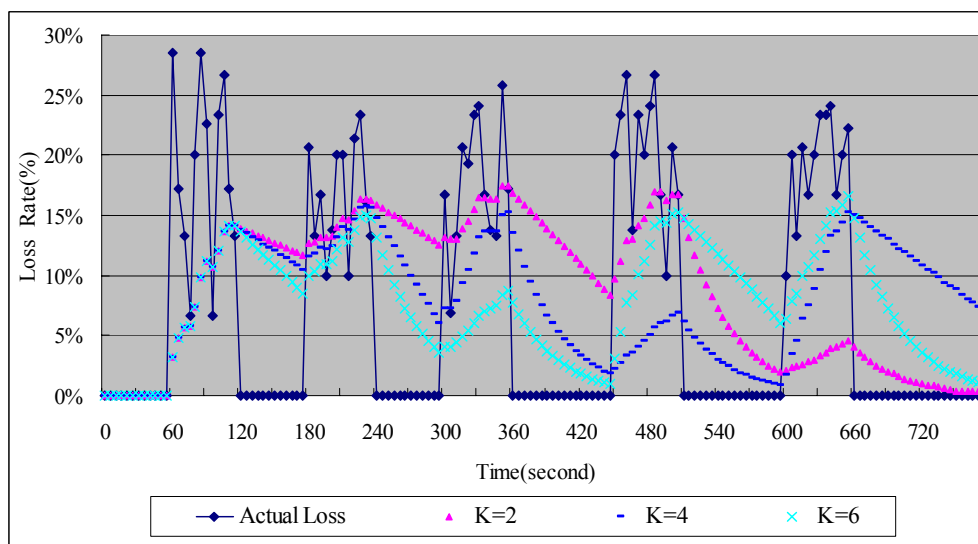


圖 21：LRV 在 $Max_FS=160$ 下， K 變動之影響

接著我們以 Max_FS 為變動參數， K 為固定參數。實驗結果，如圖 22 至圖 24。從圖 22 當中可以發現， Max_FS 的變動對 LRV 的影響很大。當 $Max_FS=40$ ，曲線則呈現完全振盪；而當 $Max_FS=80$ 與 $Max_FS=160$ ，曲線則呈現不完全振盪，但曲線之間差異仍很大。在圖 23 中， Max_FS 的變動對 LRV 的影響，跟圖 22 相比，明顯變小。同時也能看出在 $Max_FS=40$ 與 $Max_FS=80$ 時，曲線則呈現完全振盪類型，而僅當 $Max_FS=160$ 時，曲線才呈現不完全振盪類型。從圖 23 中，可發現 Max_FS 的變動對 LRV 的影響，跟圖 23 相比明顯又變小。因此，當 K 愈小時， Max_FS 的變動對 LRV 的影響愈大；反之亦然。

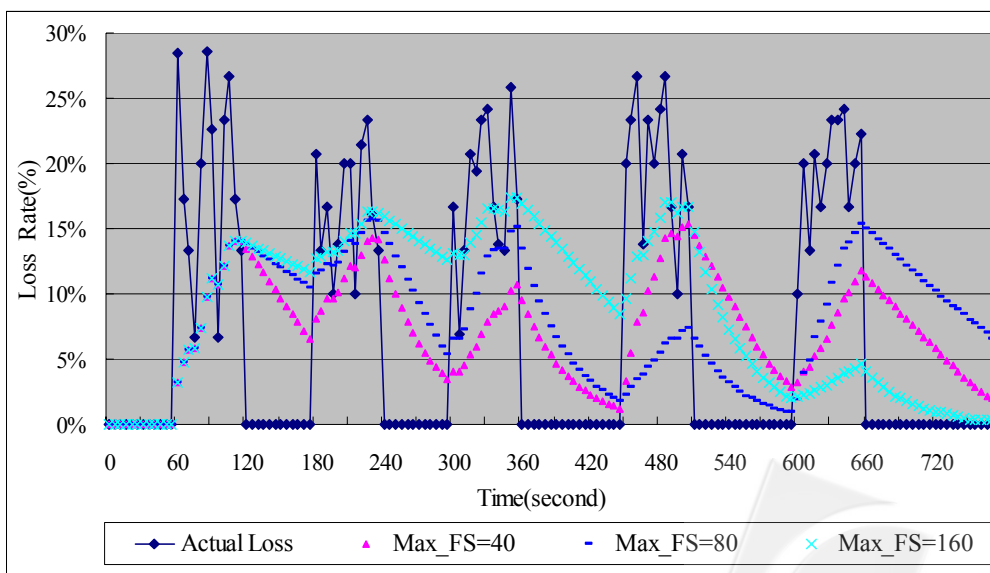
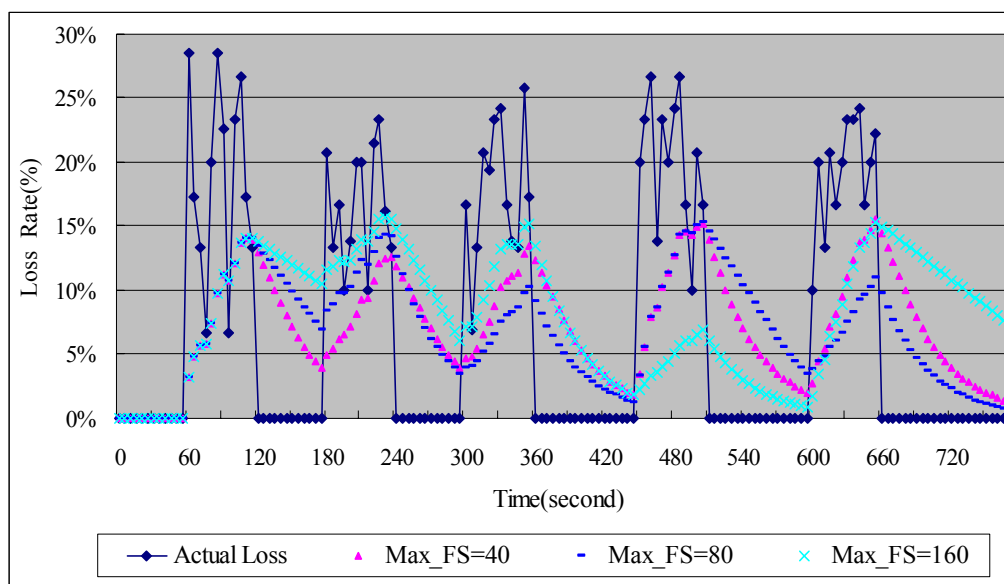
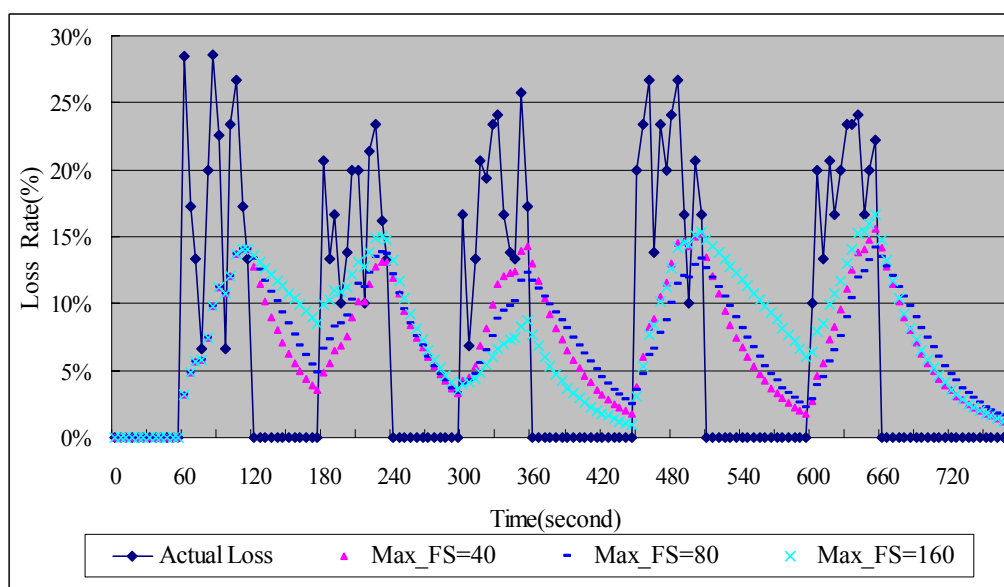


圖 22：LRV 在 $K=2$ 下， Max_FS 變動之影響

圖 23：LRV 在 $K=4$ 下， Max_FS 變動之影響圖 24：LRV 在 $K=6$ 下， Max_FS 變動之影響

第四組：C 參數

此組主要實驗 C 參數對 LRV 演算法之影響。其他參數分別為 $Upper=10\%$ 、 $Lower=5\%$ 、 $Min_FS=8$ 、 $Max_FS=80$ 及 $K=6$ 。我們以 $C=1$ 、 3 、 5 及 7 進行實驗，結果如圖 25

所示。 C 參數對 LRV 演算法而言，主要作用在延遲切換及避免振盪產生。然而演算法中所使用的 $Upper=10\%$ 及 $Lower=5\%$ ，已具有避免振盪的功能。因此， C 參數的效用明顯不高。

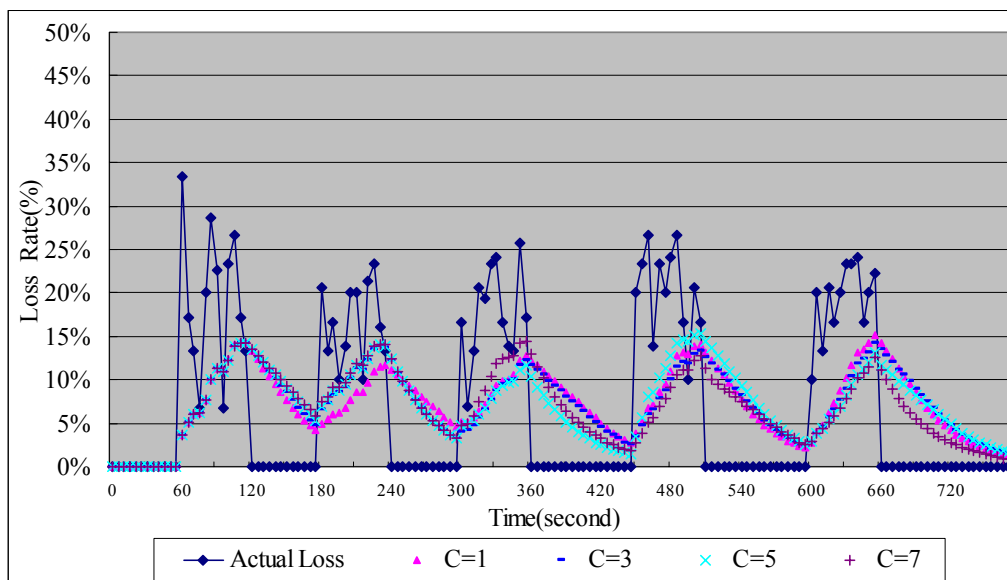


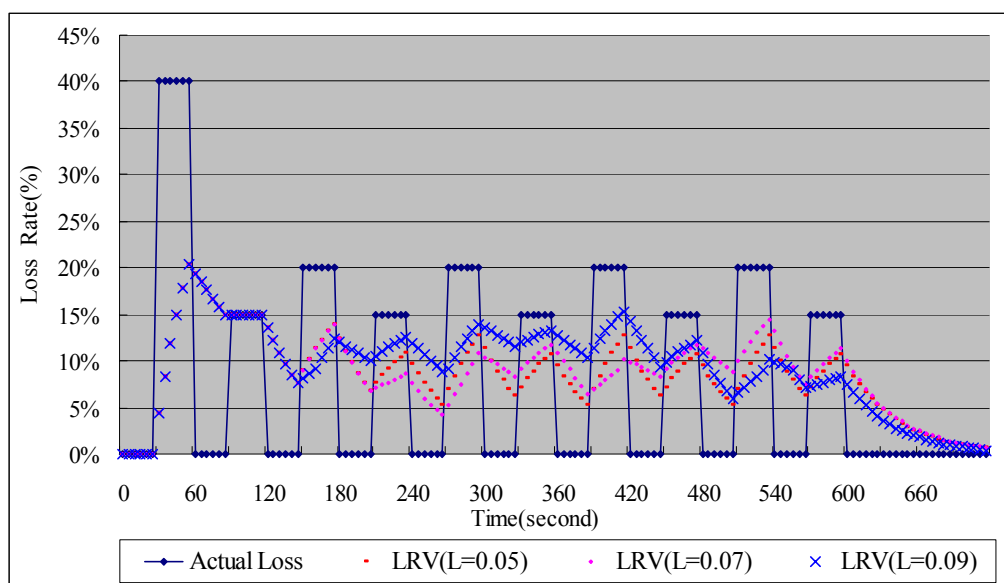
圖 25：C 參數對於 LRV 之影響

第五組：Upper 參數與 Lower 參數

此組主要觀察 $Upper$ 與 $Lower$ 之間的變動對 LRV 演算法之影響。網路交通干擾情況，則使用 30 秒的遺失率作為短期干擾。除在第一次封包遺失率為 40%，其他依序以 15% 及 20% 的封包遺失率交錯實施（如表 6），以作為模擬極度不穩定的網路交通，藉以觀察 $Upper$ 與 $Lower$ 的關係。同時為觀察 Min_FS 與切換上、下界的關係，我們也以 $Min_FS=8$ 、16 分別進行實驗。其他基本參數設定為 $C=1$ 、 $K=16$ 及 $Max_FS=110$ 。模擬結果以 U 代表 $Upper$ ， L 則代表 $Lower$ 。我們首先使用 $Min_FS=8$ ，並將 $Upper$ 視為固定參數， $Lower$ 為變動參數。實驗結果如圖 26 至圖 28。從圖 26 中，可發現當 $Lower$ 愈高（愈接近 $Upper$ ），則曲線會顯得更平滑。主要由於當切換緩衝區變小後，則將會易於產生音質切換。當音質切換次數愈頻繁的情況下，經由 LRV 抑制作用的累積，會令曲線愈來愈平滑。在圖 27 及圖 28 中，隨著 $Upper$ 的加大後，切換緩衝區也隨之變大，音質切換次數也將減少。但是，曲線仍然是相當平滑。其原因主要是由於無論遺失率是處於高音質或低音質，只要發生音質切換隨即產生抑制作用。因此，即使在 $Min_FS=8$ 的高敏捷度下，也能讓曲線顯得更平滑。

表 6：Upper 與 Lower 實驗，封包遺失率之時間區段

順序	封包遺失率	時間 (秒)		秒數
		開始	結束	
第一次	40%	30	60	30
第二次	15%	90	120	30
第三次	20%	150	180	30
第四次	15%	210	240	30
第五次	20%	270	300	30
第六次	15%	330	360	30
第七次	20%	390	420	30
第八次	15%	450	480	30
第九次	20%	510	540	30
第十次	15%	570	600	30

圖 26：LRV 在 $Min_FS=8$ 及 $Upper=0.9$ 下， $Lower$ 變動之影響

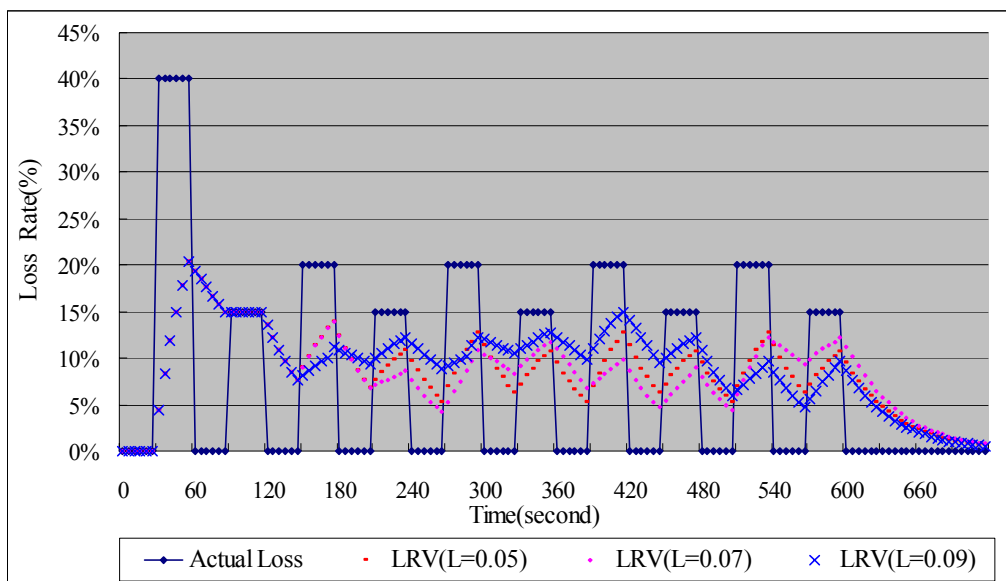


圖 27：LRV 在 $Min_FS=8$ 及 $Upper=0.1$ 下， $Lower$ 變動之影響

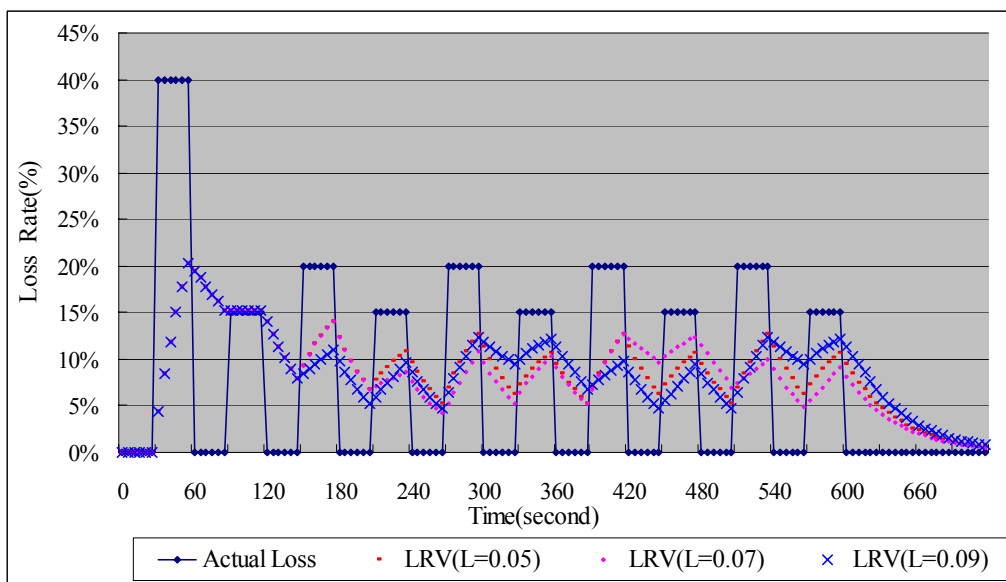


圖 28：LRV 在 $Min_FS=8$ 及 $Upper=0.12$ 下， $Lower$ 變動之影響

接著我們以 $Min_FS=8$ ，並將 $Lower$ 為固定參數， $Upper$ 為變動參數，實驗結果如圖 29 至圖 31。從圖 29 中發現，三條曲線互為重疊。其原因在於 $Lower$ 參數值設定過

低，當切換為低音質後，在極不穩定的網路交通下，無法切換回高音質。在圖 30 中，第 420 秒至第 510 秒間造成曲線間差異變大的原因，主要由於處於高音質或低音質的抑制作用影響。圖 31 也有相同情形。

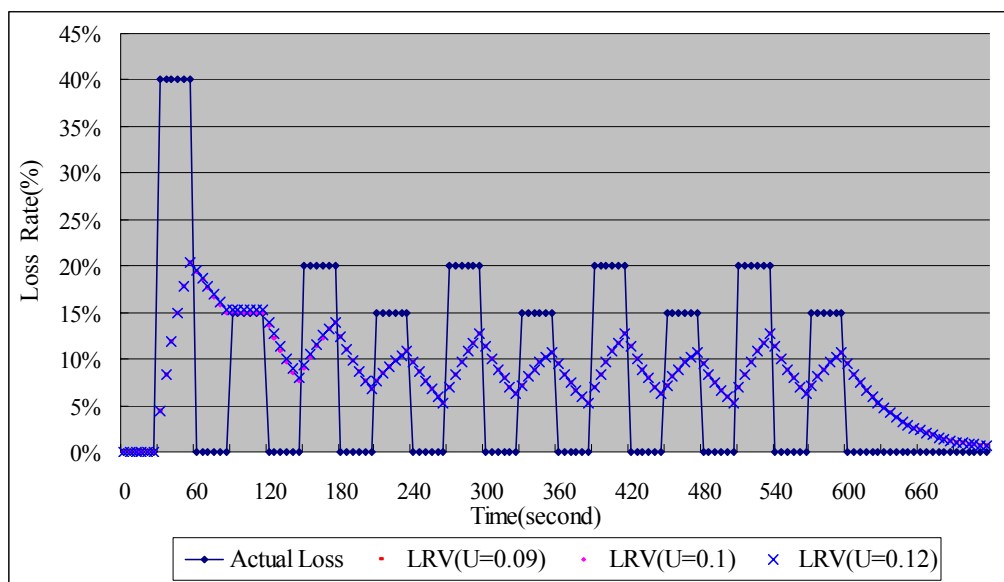


圖 29：LRV 在 $Min_FS=8$ 及 $Lower=0.05$ 下， $Upper$ 變動之影響

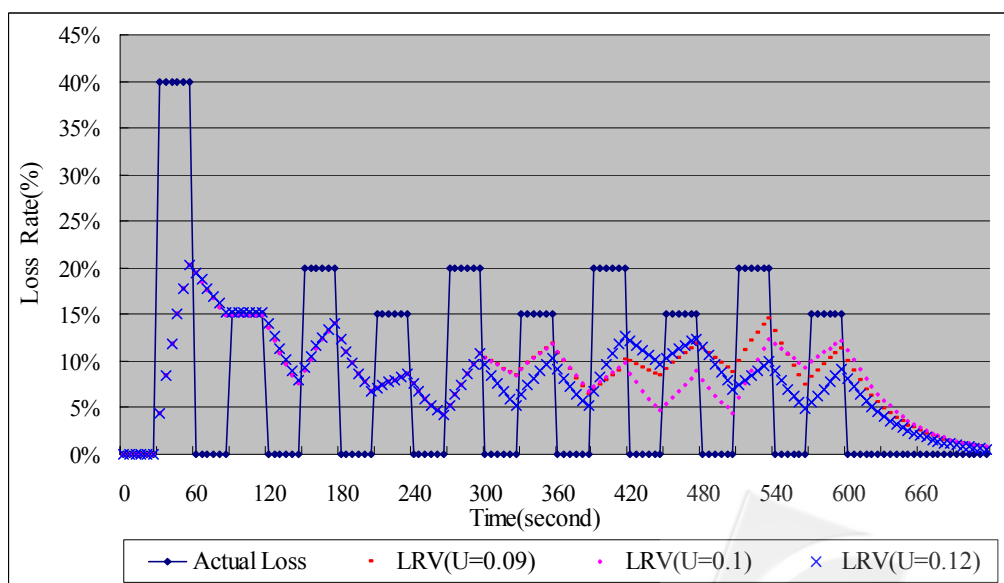


圖 30：LRV 在 $Min_FS=8$ 及 $Lower=0.07$ 下， $Upper$ 變動之影響

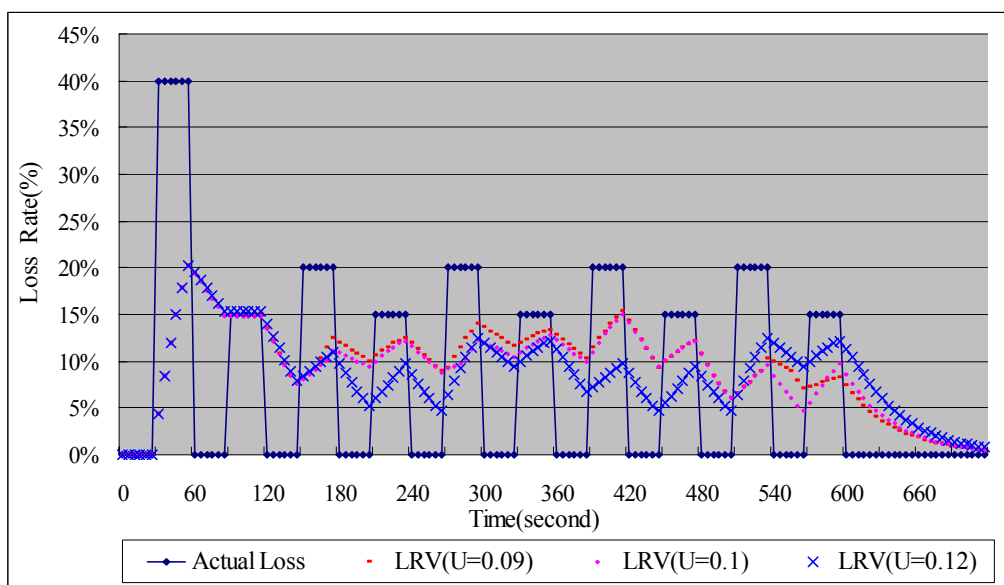


圖 31：LRV 在 $Min_FS=8$ 及 $Lower=0.09$ 下， $Upper$ 變動之影響

我們接著將調整 $Min_FS=16$ ，並將 $Upper$ 視為固定參數， $Lower$ 為變動參數。其實驗結果如圖 32 至圖 34。我們另外將 $Lower$ 視為固定參數， $Upper$ 為變動參數，其實驗結果如圖 35 至圖 37。從圖 32 到圖 37 中，我們發現當 $Min_FS=16$ 時，各曲線間的差異性很小。當 $Min_FS=8$ 時（從圖 26 到圖 31）時，各曲線差異則比較大。因此，當 Min_FS 愈高時，除平常曲線會愈平滑外， $Upper$ 與 $Lower$ 參數變動的影響也相對變小。其原因主要是 Min_FS 限制了曲線的最高敏捷度或最低平滑度。

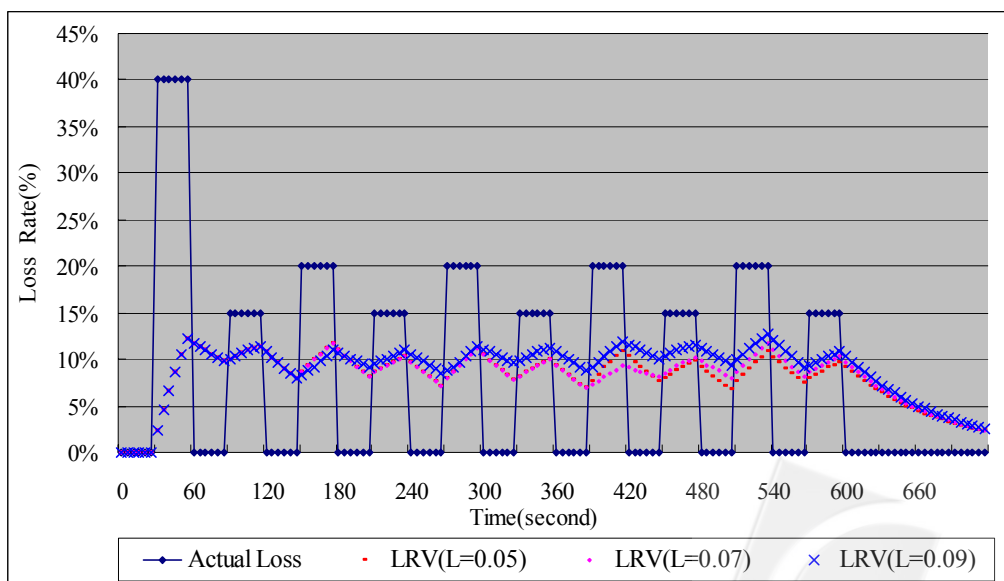


圖 32：LRV 在 $Min_FS=16$ 及 $Upper=0.9$ 下， $Lower$ 變動之影響

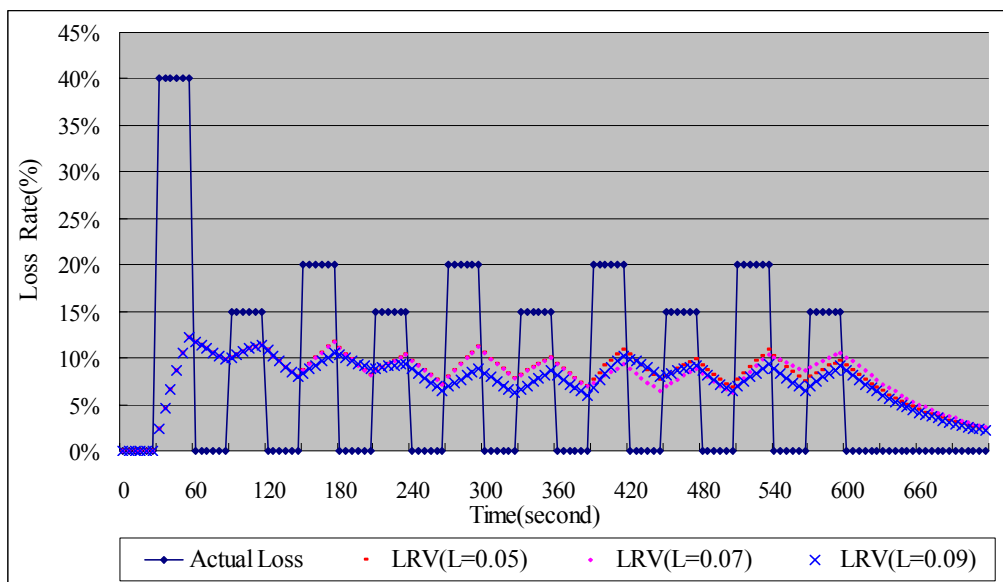


圖 33：LRV 在 $Min_FS=16$ 及 $Upper=0.1$ 下， $Lower$ 變動之影響

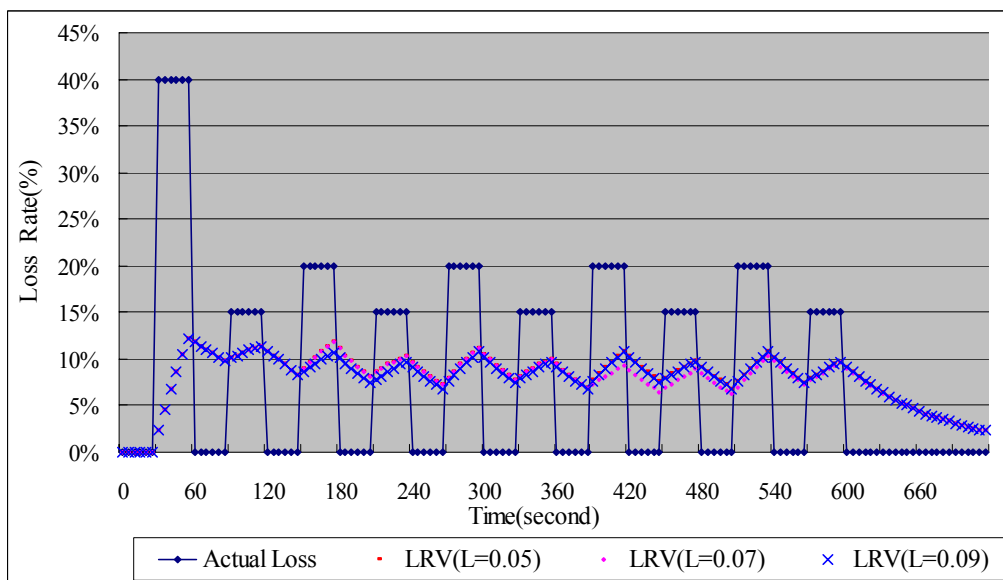


圖 34：LRV 在 $Min_FS=16$ 及 $Upper=0.12$ 下， $Lower$ 變動之影響

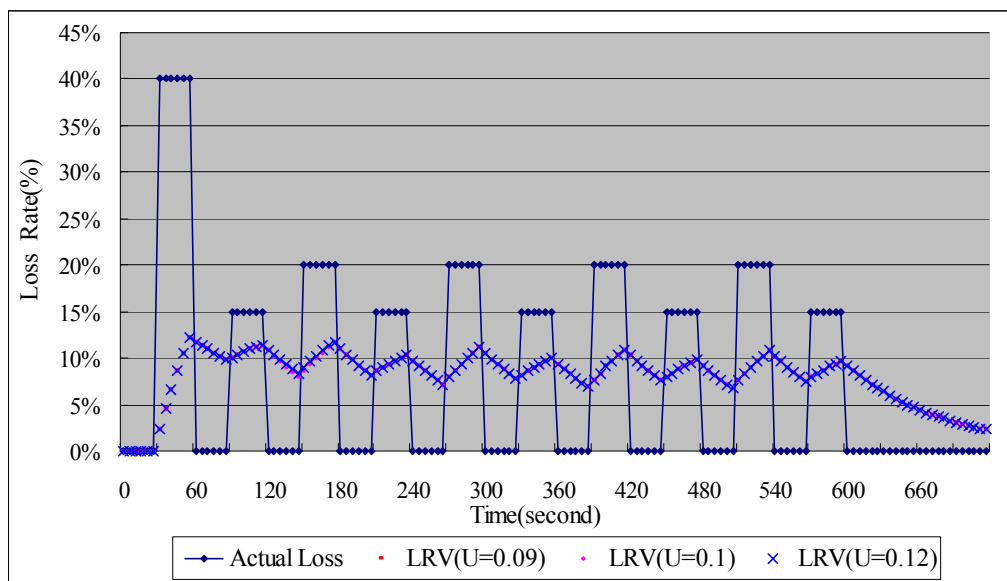


圖 35：LRV 在 $Min_FS=16$ 及 $Lower=0.05$ 下， $Upper$ 變動之影響

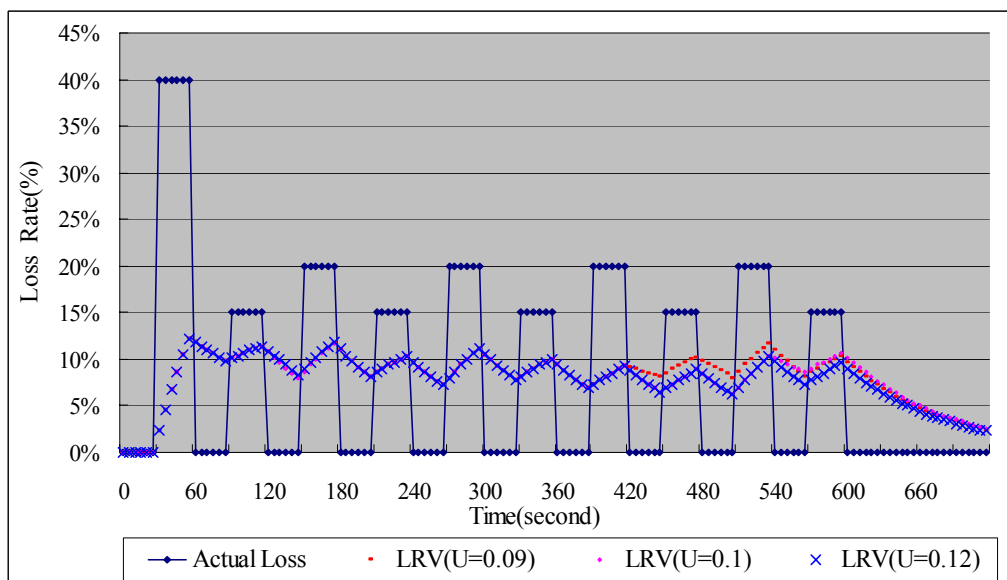


圖 36：LRV 在 $Min_FS=16$ 及 $Lower=0.07$ 下， $Upper$ 變動之影響

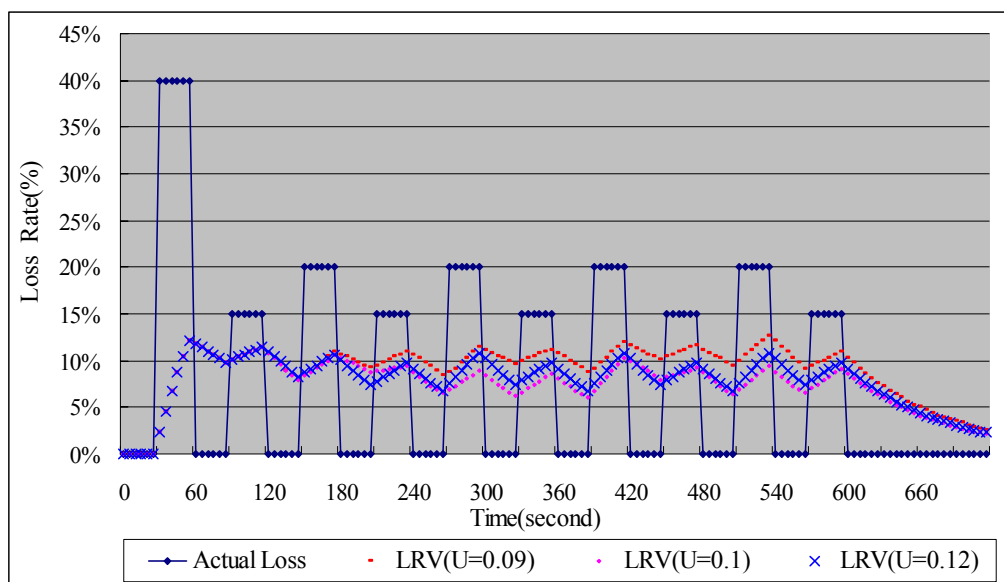


圖 37：LRV 在 $Min_FS=16$ 及 $Lower=0.09$ 下， $Upper$ 變動之影響

由五組實驗之結果可知， Min_FS 為反應網路交通最重要的影響參數， Max_FS 與 K 則決定切換後抑制作用的效用及時間。而 $Upper$ 及 $Lower$ 兩者的差異即為切換緩衝，當切換緩衝愈小，則愈容易產生切回原音質，反之則愈難切回原音質。從實驗中也可得知， Min_FS 對 Max_FS 、 K 或音質切換緩衝的影響很大，主要是由於 Min_FS 為決定反應網路交通並進行切換的最大因素。但是，四者間仍須相互搭配取適當值。從表 7 到表 9 中，列出彼此間影響 LRV 的可能結果。若曲線不存在抑制作用稱為平常曲線；反之，則稱為抑制曲線。因此， Min_FS 能決定平常曲線的平滑或敏捷，而 Max_FS 及 K 則則決定了抑制曲線的平滑或敏捷度。

表 7： Min_FS 與 Max_FS 對 LRV 之影響

項目	Max_FS 過高 (抑制作用高)	Max_FS 過低 (抑制作用低)
Min_FS 過高 (平常曲線過於平滑)	過於平滑	不完全振盪
Min_FS 過低 (平常曲線過於敏捷)	不完全振盪	完全振盪

表 8： Min_FS 與 K 值對 LRV 之影響

項目	K 過低 (抑制作用高)	K 過高 (抑制作用低)
Min_FS 過高 (平常曲線過於平滑)	過於平滑	不完全振盪
Min_FS 過低 (平常曲線過於敏捷)	不完全振盪	完全振盪

表 9：Max_FS 與 K 設定過大或過小之影響

項目	K 過低	K 過高
Max_FS 過高	抑制作用極高	—
Max_FS 過低	—	抑制作用極低

實驗三：遺失率估計演算法之比較

此實驗主要針對 EWMA、LRF 及 LRV 等遺失率估計演算法進行比較。實驗中，總共進行四次網路交通干擾，30 秒屬突發交通，60 秒以上屬長期交通，網路交通之時間區段如表 10。切換時所使用的語音資料格式為 PCM 與 GSM。RTP 語音封包基本資料，如表 11。

在本實驗中，我們使用 EWMA ($\alpha = 0.05$) 及 LRF ($FS=8$)。有關 LRV 的實驗，我們選擇了較適當參數值進行實驗。如： $Max_FS=110$ 、 $Min_FS=8$ 與 12 及 $K=6$ 。音質切換之參數為 $Upper=10\%$ 、 $Lower=5\%$ 及 $C=1$ 。在圖 38 中，LRF ($FS=8$) 為最敏捷的曲線，然而相對地卻容易引起振盪。LRV ($Min_FS=8$) 在第 200 秒音質切換為低音質後，由於音質剛切換不久，框架長度增大形成抑制作用，減緩曲線下降，造成在第 240 秒至 360 秒間（無網路干擾下）能比 LRF ($FS=8$) 更平滑往下降，此抑制作用可以避免振盪的發生。接著 LRV ($Min_FS=8$) 在第 285 秒中 (FS 遞減至 Min_FS) 恢復至原本的敏捷性 (Min_FS)，此為框架長度遞減所形成的抑制作用遞減。LRV ($Min_FS=8$) 隨即在 305 秒中切回高音質後，於第 360 秒至第 480 秒間（第三次的網路干擾下）依然能比 LRF ($FS=8$) 更平滑往上升（抑制作用影響），直至第 390 秒再切換為低音質。由於此時為低音質，因此在第 480 秒至第 600 秒間（第三次的網路干擾下）能呈現真實的網路負載。從圖 38 中觀察到，一般常用的 EWMA ($\alpha=0.05$) 為所有曲線中最平滑的曲線，但是相較其他曲線並無法真實呈現網路交通。反而 LRV ($Min_FS=12$) 在四次的網路交通干擾下，均能比 EWMA ($\alpha=0.05$) 快速的反應交通。因此，LRV 除了能很敏捷反應網路交通之外，更具備穩定的特性。

表 10：網路交通干擾之時間區段

干擾順序	網路負載 (TCP 資料)	時間 (秒)		秒數
		開始	結束	
第一次	40%	60	90	30
第二次	60%	180	240	60
第三次	60%	360	420	60
第四次	60%	480	600	120



表 11：RTP 語音封包基本資料

項目	高語音品質	低語音品質
資料類型	PCM	GSM
取樣率 (Hz)	44,100	8,000
封包數/秒	6	16.6
IP 封包大小 (Bytes)	1,402	139
封包平均傳輸率 (KB/s)	8.22	1.1
封包平均到達間隔時間 (ms)	166.7	12.05
Timestamp 的增值	15,120	479

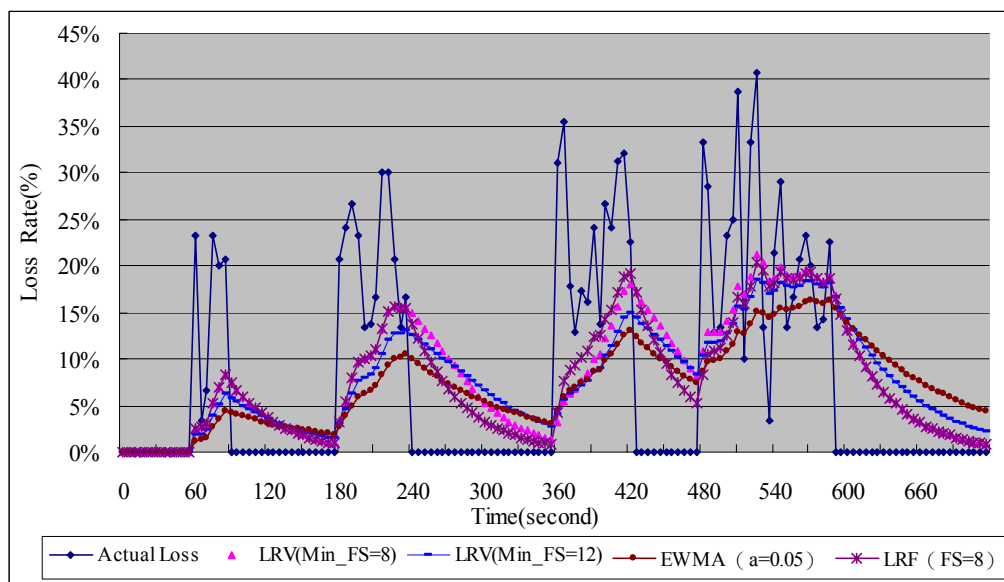


圖 38：EWMA、LRF 與 LRV 之比較

●實驗四：LRV 與 EWMA 的切換次數比較

此實驗主要針對 LRV 及 EWMA 等遺失率估計演算法進行切換次數的比較。本實驗的操作過程與實驗二的第五組實驗相同，總共進行十次。實驗區段如表 6。RTP 語音封包基本資料，如表 11。在本實驗中，我們使用 EWMA ($\alpha=0.05$)、LRF ($Min_FS=8$) 及 LRF ($Min_FS=16$) 等進行實驗。LRV 的參數，如表 12 所示。為方便觀察 EWMA ($\alpha=0.05$)、LRF ($Min_FS=8$) 及 LRF ($Min_FS=16$) 等切換次數的比較，我們使用各種不同 Upper 及 Lower 組合，彼此進行比較。實驗過程與實驗二的第五組實驗相同，實驗結果切換次數之比較如表 12。

表 12：LRV 與 EWMA 切換次數比較

No	Upper	Lower	切換緩衝 (Upper-Lower)	LRV (Min_FS=8)	LRV (Min_FS=16)	EWMA ($\alpha=0.05$)
1		0.05	0.0004	2	2	2
2	0.09	0.07	0.02	6	4	8
3		0.9	0	8	8	20
4		0.05	0.05	2	2	2
5	0.1	0.07	0.03	6	4	8
6		0.9	0.01	6	6	10
7		0.05	0.07	2	2	0
8	0.12	0.07	0.05	4	2	0
9		0.9	0.03	6	2	0

從表 12 中可以發現，隨著切換緩衝的降低，各演算法下的切換次數均往上升。尤其當切換緩衝為 0(No.3)時，各演算法下的切換次數明顯偏高，而以 EWMA($\alpha=0.05$) 最高，達 20 次。同時也發現除了當 Upper 設為 0.12 時，EWMA($\alpha=0.05$) 無造成切換外，LRV 演算法的切換次數均比 EWMA($\alpha=0.05$) 要低。而且隨著切換緩衝變小，所造成切換次數的增加，LRV 也明顯比 EWMA($\alpha=0.05$) 少。

陸、結論與未來展望

本文提出一個以變動框架長度為基礎之遺失率估計法，此法可支援不同音質的語音傳輸。依據實驗結果顯示，本文所提出之演算法，針對封包遺失率的估計，不僅具備穩定性，更能敏捷地反應網路交通情況。在 LRV 的演算法中，Min_FS 為反應網路交通最重要的影響參數，而 Max_FS 與 K 則決定切換後抑制作用的效用大小及時間長短。在 Min_FS 愈小的情況下，Max_FS 或 K 參數之變動對各 LRV 曲線間所產生的差異性會愈大。相對地，在 Min_FS 愈大的情況下，Max_FS 或 K 之變動對各 LRV 曲線間所產生的差異也會愈小。其原因主要由於 Min_FS 為決定反應網路交通並進行切換的最大因素。但是，三者間仍須相互搭配，以取得適當值。LRV 在跟 EWMA 及 LRF 比較後，也較能呈現更敏捷及平滑的曲線。

此演算法同時具備簡單性及方便性，也適合用於即時語音品質切換機制。其優點主要如下：

- 一、避免音質切換過度振盪：在網路極度不穩定下，所估計到遺失率變異會很大。因此，高低音質會有多次切換。每次音質切換後，當網路狀況處於低音質區間且遺失率曲線下降時，或者在高音質下且遺失率曲線上昇時，此法使用較高之平滑方式進行抑制作用，以避免切換過度頻繁。同時，此法亦考慮到其切換的次數問題，當進行連續切換下，其抑制作用持續累積，以便過濾極度不穩定的網路交通。

二、可估計出真實的網路交通：當進行音質切換後，框架長度會呈直線遞減。當經過一段時間後，即會恢復原本的敏捷性，以便快速偵測網路狀況。當估計到的遺失率，處於低音質區間且遺失率上昇時，或者處於低音質區間且遺失率下降時，此法能以最小框架之敏捷性顯示所估計到封包遺失率。

此外，此法亦可採用 EWMA 權重方式來對遺失率做估算。未來研究，將針對 EWMA 之參數進行研究，或者考量加入 Multicast、分散式代理機制的測量或搭配其他 QoS，做更深入之探討。

參考文獻

1. H. Schulzrinne, S. Casner, R. Frederick and V. Jacobson, "A Transport Protocol for Real-Time Applications", RFC 3550, Internet Engineering Task Force, July 2003.
2. G. Huston, "Next Steps for the IP QoS architecture", RFC 2990, Internet Engineering Task Force, November 2000.
3. ITU-T draft, P.VTQ, "The assessment of voice transmission quality from protocol analysis information in IP networks".
4. A. Takahashi, H. Yoshino and N. Kitawaki, "Perceptual QoS assessment technologies for VoIP", IEEE communications magazine, vol. 42, no. 7, July 2004, pp. 28-34.
5. ITU-T Recommendation P.862, "Perceptual evaluation of speech quality (PESQ), an objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codes", February 2001.
6. J. Light and A. Bhuvaneshwari, "Performance analysis of audio codecs over Real-Time Transmission Protocol (RTP) for voice services over internet protocol", Second Annual Conference on Communication Networks and Services Research (CNSR'04), May 2004, pp. 351-356.
7. L. Roychoudhuri, E. Al-Shaer, H. Hamed and G. B. Brewster, "Audio transmission over the internet: Experiments and observations", ICC 2003 – IEEE International Conference on Communications, vol. 26, no. 1, May 2003, pp. 552-556.
8. ITU-T Recommendation G.107, "The E-Model, a computational model for use in transmission planning", March 2003.
9. D. Awduche, A. Chiu, A. Elwalid, I. Widjaja and X. Xiao, "Overview and principles of internet traffic engineering", RFC 3272, Internet Engineering Task Force, May 2002.
10. M. Handley, J. Padhye and J. Widmer. "TCP Friendly Rate Control (TFRC) : Protocol Specification", Internet-Draft Draft-ietf-tsvwg-tfrc-01.txt, Internet Engineering Task Force, March 2001.

11. J. Padhye, J. Kurose, D. Towsley and R. Koodli. "A model based TCP-friendly rate control protocol", In Proceedings of NOSSDAV, 1999.
12. S. Floyd, M. Handley, J. Padhye and J. Widmer. "Equation-based congestion control for unicast application", In Proceedings of SIGCOMM, pp. 57-66, 2000.
13. D. D. Clark and W. Fang, "Explicit allocation of best-effort packet delivery service", IEEE/ACM Trans. Networking, vol.6(4), August 1998, pp 362-373.
14. F. Agharebparast and V. C. M. Leung, "A new traffic rate estimation and monitoring algorithm for the QoS-enabled internet", GLOBECOM 2002-IEEE Global Telecommunications Conference, vol. 22, no. 1, December 2003 pp. 3883-3887.
15. N. Cranley and L. Murphy, "Adaptive quality of service for streamed MPEG-4 over the internet", ICC 2001 – IEEE International conference on communications, no. 1, June 2001, pp. 1206-1210.
16. T. Yoshimura, T. Ohya, T. Kawahara and M. Etoh, "Rate and Robustness Control with RTP monitoring agent for mobile multimedia streaming", ICC 2002 – IEEE International Conference on Communications, April 2002.
17. I. Busse, B. Deffner and H. Schulzrinne, "Dynamic QoS control of multimedia applications based on RTP", Computer Communications, January 1996.
18. P. Young, "Recursive estimation and time-series analysis", Berlin, Springer- Verlag, 1984.
19. S. Palacharla, A. Karmouch and S. A. Mahmoud, "Design and Implementation of a Real-time Multimedia Presentation System using RTP", COMPSAC'97-21st International Computer Software and Applications Conference, August 1997, pp. 376.

