

國立交通大學

電信工程學系碩士班
碩士論文

網路語音傳輸系統規劃之研究

A Study of Transmission Planning for
Internet Voice Service



研究生：李世耀

指導教授：張文輝 博士

中華民國九十三年六月

網路語音傳輸系統規劃之研究
**A Study of Transmission Planning for
Internet Voice Service**

研究生：李世耀

Student: Shi-Yao Lee

指導教授：張文輝

Advisor: Wen-Whei Chang

國立交通大學
電信工程學系碩士班



Submitted to Institute of Communication Engineering
College of Electrical Engineering and Computer Science
National Chiao Tung University
In Partial Fulfillment of Requirements
for the Degree of
Master of Science
in Electrical Engineering

June 2004
Hsinchu, Taiwan, Republic of China

中華民國 九十三年 六月

網路語音傳輸系統規劃之研究

學生：李世耀

指導教授：張文輝 博士

國立交通大學電信工程學系碩士班

中文摘要

網路語音通訊的服務品質取決於諸多因素，主要是封包漏失、延遲與擾動。為了讓網際網路能提供更好的服務品質，傳送端利用前向錯誤控制來回復封包漏失，接收端則利用撥放暫存機制來補償延遲擾動。基於漏失與延遲之間的交互作用會影響系統建構，於是在硬體製作前應先作實地量測再據以規劃網路語音傳輸系統。本論文的首要部分著重發展一以軟體為基礎的量測系統，用以評估網際網路的各項服務品質因素。再根據聽覺評量並參考國際電信聯盟制定的標準建立不同因素所對應的音質損害，整合推導出單一能具體反應網路通話品質的音質評量指標。有別於隨機漏失模型，二狀態的吉伯特模型更能適切地描述封包漏失的時域相依特性，因此採用吉伯特模型來分析封包層級的前向錯誤控制效能。最後我們分別針對隨機與叢發性封包漏失模式，配合音質評估模型，完成前向錯誤控制的最佳化設計。

A Study of Transmission Planning for Internet Voice Service

Student : Shi-Yao Lee Advisor : Dr. Wen-Whei Chang

Institute of Communication Engineering
National Chiao Tung University

Abstract

The quality of voice over IP (VoIP) is mainly affected by network impairments such as delay, jitter and packet loss. In order for IP networks to provide quality of service, playout buffer algorithms at the receiver compensate for jitter and forward error control (FEC) is used to mitigate the impact of packet losses. Due to the dependence of network condition on loss and delay tradeoffs, it is a prerequisite to establish speech transmission planning prior to the system implementation. The first part of this thesis concentrates on developing a software-based measurement system that evaluates the characteristics of VoIP. We then combine individual loss and delay impairments using ITU standardized E-model to predict the subjective quality of VoIP. We also propose the use of voice quality prediction model for perceptual optimization of FEC design. Packet-level FEC performance is analyzed in the case of two-state loss model, and compared with the random loss model, in order to demonstrate the benefits of Gilbert model that more closely characterizes the temporal dependency in packet losses.

誌謝

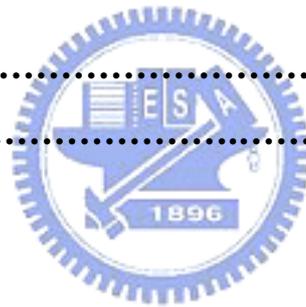
兩年的研究生涯，在張文輝老師悉心的指導下，讓我深刻體認到做研究所需具備的嚴謹，相信這將深遠影響日後我在職場上研究的態度。而每當研究遇到瓶頸時，老師更是不辭辛苦的從旁協助找出方向讓整個研究順利進行。另外特別感謝亨仰與承龍兩位學長在學業以及生活上給予的諸多協助。同時感謝這段期間父母與好友涵青在這兩年給予精神上的支持與鼓勵。最後將本論文獻給我的指導老師、學長、父母以及所有關心我的好友們。謝謝您們。



目錄

中文摘要.....	i
英文摘要.....	ii
致謝.....	iii
目錄.....	iv
圖目錄.....	vi
表目錄.....	viii
第一章 緒論.....	1
1.1 前言.....	1
1.2 研究動機.....	2
1.3 章節概要.....	3
第二章 網路品質量測平台.....	5
2.1 動機.....	5
2.2 量測擷取系統製作.....	6
2.3 服務品質參數的估算.....	9
2.4 系統模擬與討論.....	13
2.5 網路電話(VoIP)QoS 之應用.....	17
第三章 通話品質預測模型.....	20
3.1 主觀聽覺測試.....	20
3.2 音質評量指標.....	25
3.2.1 語音編碼損害因子- I_{ec}	27
3.2.2 封包漏失損害因子- I_{epi}	30
3.2.3 延遲損害因子- I_d	31

第四章 語音傳輸系統規劃.....	34
4.1 封包漏失回復機制.....	34
4.2 前向錯誤控制.....	36
4.3 吉伯特漏失模型.....	42
4.4 Reed-Solomon 碼回復效能.....	44
4.5 系統規劃與討論.....	48
第五章 結論與未來展望.....	60
5.1 結論.....	60
5.2 未來展望.....	61
參考文獻.....	63
附錄.....	66



圖目錄

圖 2.1	VoIP 量測系統測試平台.....	8
圖 2.2	品質參數估算流程圖.....	10
圖 2.3	演算法計算過程詳細說明實例.....	12
圖 2.4	狀況一之測量數據與延遲擾動曲線圖.....	14
圖 2.5	狀況二之測量數據與延遲擾動曲線圖.....	15
圖 2.6	狀況三之測量數據與延遲擾動曲線圖.....	16
圖 2.7	狀況一的延遲擾動補償與封包漏失關係.....	18
圖 2.8	狀況三的延遲擾動補償與封包漏失關係.....	19
圖 3.1	R 與 MOS 的轉換關係.....	26
圖 3.2	編碼位元率與對應的音質損害.....	29
圖 3.3	封包漏失與對應的音質損害.....	31
圖 3.4	單向通話迴音干擾.....	32
圖 3.5	單一路徑延遲與對應的音質損害.....	33
圖 4.1	與資料流格式有關的錯誤控制過程.....	39
圖 4.2	多封包錯誤控制模式.....	39
圖 4.3	檢查封包產生過程.....	41
圖 4.4	(3,2)區塊碼所衍生的額外延遲.....	41
圖 4.5	同位碼回復錯誤的過程.....	41
圖 4.6	吉伯特漏失模型.....	42
圖 4.7	k 值與 PLR_{FEC} 的關係.....	48
圖 4.8	G.729a 在隨機封包漏失不同(n,k)情形下與 R 值的關係....	52
圖 4.9	G.729a 在叢發漏失模型不同(n,k)情形下與 R 值的關係....	52

圖 4.10 G.711 在隨機封包漏失不同(n,k)與 R 值的關係.....55

圖 4.11 G.711 在叢發漏失模型不同(n,k)與 R 值的關係.....55

圖 4.12 吉伯特模型的 PLR_{FEC} 變化.....56

圖 4.13 叢發性封包漏失與 I_{epi} 對應關係.....56

圖 4.14 G.723.1 在隨機漏失封包模型不同(n,k)與 R 值的關係.....59



表目錄

表 3.1	絕對分類與分數的對應關係.....	24
表 3.2	衰減分類與分數的對應關係.....	24
表 3.3	比較分類與分數的對應關係.....	24
表 3.4	R 與音質對應關係.....	26
表 4.1	FEC 與 ARQ 的比較.....	36
表 4.2	G.729a 在隨機漏失模型下配合不同(n, k)的音質評估.....	51
表 4.3	G.729a 在叢發漏失模型下配合不同(n, k)的音質評估.....	51
表 4.4	G.711 在隨機漏失模型下配合不同(n, k)的音質評估.....	54
表 4.5	G.711 在叢發漏失模型下配合不同(n, k)的音質評估.....	54
表 4.6	G.723.1 在隨機漏失模型下配合不同(n, k)的音質評估....	58



第一章 緒論

1.1 前言

電話是人類傳達訊息最直接方便、自然、而有效的管道。而近年來由於網際網路的盛行以及日益增加的傳輸速度，於是進一步思考網路除了取得所需的數據文字資料外，是否可以作為彼此傳遞語音訊息的工具。因此新一代的網路環境便將多媒體的傳輸納入其中，語音、視訊及資料整合的架構，代表了未來訊息傳輸的潮流。網路電話(Voice over IP, VoIP)便成了網路整合後的衍生應用服務。它低廉的通話費用以及具彈性的附加應用，造成使用者人數逐年增加且有取代傳統電話的趨勢。對於相關技術的研發與創新，以及此應用平台所衍生出新問題的探討，更成為產學界研究發展的重要課題。其目標是提高語音品質且達到即時傳送的效果，並希望能夠將網路電話更進一步地應用於無線區域網路(Wireless LAN)上，讓網路電話的使用不受時空限制。

將語音結合入資料傳輸為主的網路之最大困難點，在於語音封包傳輸模式所衍生的通話延遲(delay)，延遲擾動(delay jitter)以及封包漏失率(packet loss rate)，構成了嚴重影響網際網路多媒體通訊品質的損害要素。舉例而言，延遲造成雙向通話過程中迴音的干擾，而延遲擾動與封包漏失皆產生斷斷續續的不流暢音質。因此解決這些問題，使其達到傳統電話穩定且高品質的通話水準，便成為值得探討的研究

課題。

系統規劃的首要之務是正確量測出影響網路通話品質的諸多因素，然而前人研究卻沒有考量兩通話終端內部設備存在的電子時脈誤差問題。在雙方長時間通話後，會存在某種程度誤差的不正確延遲擾動。而這將導致誤判網路狀態而加大或縮短撥放暫存器的長度，造成不必要的額外延遲或直接惡化通話品質。有鑑於此，本論文提出一以軟體為基礎的測量系統，在真實的網路環境中傳遞語音封包並擷取紀錄，藉著分析封包標頭(Header)的資訊來加以計算，以期準確估測出網路環境上 VoIP 應用的品質評估參數。所採用的技術是以與 PC NDIS 驅動程式互動的商業用封包監控工具為基礎。利用此一系統，封包延遲、漏失封包特性描述以及特定的 delay jitter 可以僅僅使用軟體就可以十分準確地量測出來。

1.2 研究動機

網路電話的服務品質取決於諸多因素，包括封包漏失、延遲時間、背景雜訊、及語音編碼失真，其中封包漏失率及延遲的容忍上限分別為 10% 及 150msec。目前相關技術都採用錯誤控制碼(forward error control)及播放緩衝機制(playout buffer algorithm)，前者運用保護位元執行封包漏失的回復處理[1,2,3]，後者則彈性調整封包播放時間以對抗延遲擾動(delay jitter)[4,5]。這兩者之間其實存在因果循環關

係，有效的錯誤控制碼所增加的延遲時間較長，進而影響播放緩衝時間的設定。除此之外，語音編碼處理有許多選擇[6]，如 G.711 PCM、G.729 CS-ACELP、G.723.1 MPC-MLQ，而不同模式所衍生的信號失真及延遲時間亦存在明顯差異[7]。較理想的系統規劃是整體考量不同關鍵元件的最佳組合設計，且因應隨時變化的網路傳輸特性作合理調整。針對此項需求，首要之務為建立一個能具體反應網路電話音質的聽覺評量指標[8,9]，以提供系統關鍵元件在錯誤控制與參數調整之用[10]。問題是音質不容易得到一致而客觀的認定，產業界通常使用昂貴的檢測儀器來衡量。主要是因為音質評量牽涉到通道特性及系統架構兩層面，而不同服務品質因素所對應的音質損害程度更存在明顯差異。有鑑於此，我們將針對 Voice over IP (VoIP) 的特定應用環境，設計並製作一正確量測各項服務品質因素的測試平台[11]。同時參考國際電信聯盟 ITU 針對網路系統規劃所制定的 E-model[12]，根據聽覺評量建立不同因素所對應的音質損害，再整合推導出單一能具體反應網路通話品質的音質評量指標。

1.3 章節概要

在本論文的架構上，第二章將會介紹擷取測量平台，並利用我們所提出優於傳統演算法來計算延遲擾動，並與修正後比較其差異性。第三章則會介紹音質評估模型，先個別介紹音質損害因子(impairment

factor)，並利用國際電信聯盟(ITU-T)規格來建立整體音質評估模型。

第四章則針對系統用以對抗封包漏失的錯誤控制來加以說明，並建立

隨機與叢發兩種不同模型的封包漏失機率推導，同時配合音質評估模

型進行語音傳輸的系統規劃。最後在第五章做總結與未來展望。



第二章 網路品質量測平台

網路電話利用 Voice over IP (VoIP)技術開發結合現代化通信系統與網路傳輸之新的應用服務。隨著使用者人數的逐漸增加，儼然有取代傳統電話的傾向。許多學者投入網路電話的研究，致力於改善傳輸過程中所遭遇到的損害使其聲音依舊保持順暢，並希望能夠將網路電話更進一步地應用於無線區域網路(Wireless LAN)上，讓使用網路電話時兼具行動力。本章的提出一以軟體為基礎的測量系統，以期準確估測出網路環境上 VoIP 應用的品質評估參數。

2.1 動機



網際網路多媒體通訊品質取決於傳輸資訊的時間延遲(delay)、延遲擾動(delay jitter)、封包漏失(packet loss)以及有限制的資料頻寬。針對即時多媒體通訊，提供穩定傳輸量以呈現高品質的應用是必須要求的，而利用播放緩衝器(Playout Buffer)之設計可以控制時間延遲與封包漏失的問題。考量封包太晚抵達或是未曾抵達所造成封包漏失的狀況。不論是因外部擾動造成的延遲或網路通道狀態所產生的封包漏失，直接的影響是在資訊撥放緩衝器(information playout)串中產生一個漏失的區間。而利用擾動補償緩衝器延遲所有已接收到的封包以允許所有封包較晚抵達，將會改善封包丟棄的狀況進而順暢音質，但相對地要付出延遲增加的代價。

網路電話通訊對於封包漏失與延遲都是非常敏感的。10 % 的封包漏失與 150 毫秒時間延遲是典型考量可接受的最大數值。因此調整擾動補償緩衝器大小，使得兩種品質量測在範圍之內，同時衡量彼此之間良好的折衷選擇。高品質的網路電話通訊要求對傳送端到接收端封包延遲擾動進行準確且即時的估測，這是為了適當且動態地設定擾動補償緩衝器長度使得通訊品質達到最佳。

量測系統的構想是希望能夠以封包時間統計資料來衡量網路電話的通訊品質。此系統的發展是以 PC NDIS 驅動程式互動的商業用封包監控工具為基礎，將封包延遲與漏失封包特性以及特定的延遲擾動現象以軟體運行準確地量測出來。



2.2 量測擷取系統製作

本節將描述一般封包計時量測的類型及其優缺點，並提出新量測技術其具有高準確性的混合式軟體基底量測系統。一般基於封包計時量測來評估 VoIP QoS 的方法主要有兩種：

- (1) 量測每個封包從傳送端到接收端的絕對時間延遲；
- (2) 量測已接收封包之間的 interarrival 時間。

第一種方法可以有效描述網路延遲與擾動的特性。但是由於它需要節點維持微秒(microsecond)位階的時脈同步與量測準確度，因此難以準確地實行。第二種方法較容易實行並產生非常接近關於應用位階效能

的資訊，但其遮蔽了基本網路時間延遲擾動統計資料(接收封包 interarrival 計時取決於傳送封包 interarrival 計時以及快速變化的網路延遲)。我們可以發現，前者需詳細說明需要的演算法與網路開發環境來分析出準確的資訊，而後者較適用於測定“所有狀況都考慮進去”的撥放緩衝器品質。

為克服上述方法的缺點，本論文提出一混合 VoIP 通道分析技術，其可以易於實行並產生足夠的詳細資訊使得傳送端應用延遲、網路擾動、以及接收應用統計資料全部可以識別出來。以一個紀錄傳送端與接收端 interarrival 計時以及封包序列識別碼的封包量測系統為基礎，所計算的端點對端點封包延遲擾動可以控制在 50 微秒 (microsecond) 的準確度之內。此外，所得到的資料對於計算在網路中漏失的封包數目與時間分布也十分有用。

接著，將描述如何使用此系統來產生、擷取與紀錄 VoIP 的封包。圖 2.1 顯示此量測系統的測試平台：兩份同時存在的網路封包擷取工具，一份置於接收終端機執行，而另一份放在傳送終端機執行。某些網路通訊協定與架構允許雜亂(promiscuous)監控，若考慮因作業系統忙碌所造成額外影響降至最低的狀況，使用“第三方”量測電腦是可行的。然而，節點內接收端與傳送端延遲擾動的問題也必須考慮成所有網路問題的一部份。

此系統是以穩定的作業系統-微軟視窗 2000(Microsoft Windows 2000)為基礎。由於有微秒(microsecond)位階計時、雜亂網路監控(可以手動濾掉外來無關的訊息傳輸)以及能紀錄多種參數，但仍在高於 MAC 層監控之能力，與 Network Device Interface Specification (NDIS) 層互動是必須的。NDIS 是在微軟視窗驅動程式供上層數種協定堆疊同時存取的網路介面協定，其驅動程式內建於網路卡中。幸運的是目前有許多商業監控軟體可以與 NDIS 層互動，並擷取與紀錄量測所需的封包資訊。本量測系統所選擇的軟體為 Sniffer，可以符合上述所提及的所有需求。

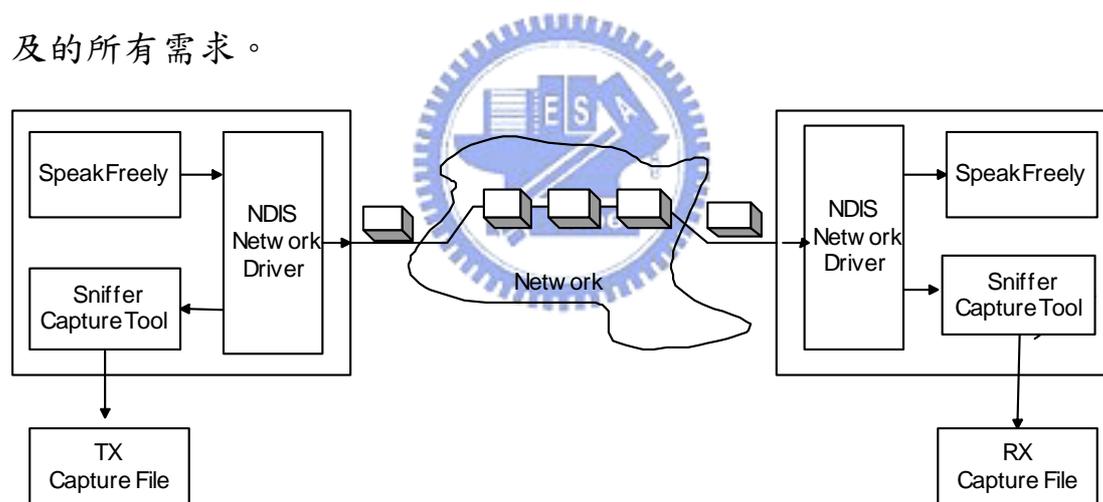


圖 2.1 VoIP 量測系統測試平台

最後，我們選擇應用於 VoIP 的開放共享軟體 Speak Freely 作為封包產生器。雖然此一應用軟體並不是 VoIP 服務最佳的實現(它不能維持一固定的緩衝器長度並缺乏良好的計時控制)，但是它作為一具有不同參數之 VoIP 測試封包產生器是非常有用的。雖然 Sniffer 軟體也可以提供封包產生器的應用，但是它並沒有能力產生具有序列識別之

封包，且其封包產生的 interarrival 時間也不準確，因此不予採用。

2.3 服務品質參數的估算

當 VoIP 封包經由此量測系統傳送，其相對應的 interarrival 時間會紀錄於傳送端與接收端終端機中。本節將描述如何利用在傳送接收兩端所蒐集之資料檔案來估測封包漏失與時間延遲擾動等品質評估參數。

在上一節中，經由 Sniffer 軟體分別在傳送端與接收端所蒐集的記錄檔案皆含有每個封包的識別碼(ID number)以及 interarrival 時間等資訊。若整個實驗環境時脈一致，則時間延遲擾動可以非常容易利用上述資訊計算獲得。然而，實驗所使用的兩組量測用電腦有十分穩定卻不一致的時脈，特別是如果它們為一般個人電腦。實驗顯示這些電腦間的差別量測所發生的一點點改變，但是卻會在絕對計時上顯現出很大的差距。而由於此一測量系統是判定一個封包到達所需花費的時間，每次累加的 interarrival 時間必須反映正確的絕對計時。舉例而言，一測量顯示時脈差別為 0.138 毫秒/秒(ms/s)，這表示三十秒的語音之後，量測結果將會有高達 4.14 毫秒的誤差，因此如何克服此一問題是很重要的。幸運的是，此一類型的誤差十分線性而且穩定，因此不論是否為傳送序列的第一個或是最後一個封包，都可以去除此種誤差所造成的影響並使得最後每個封包量測的準確度在 50 微秒

(microsecond)之內。圖 2.2 表示所採用有效去除誤差的品質評估參數估測演算法流程圖。

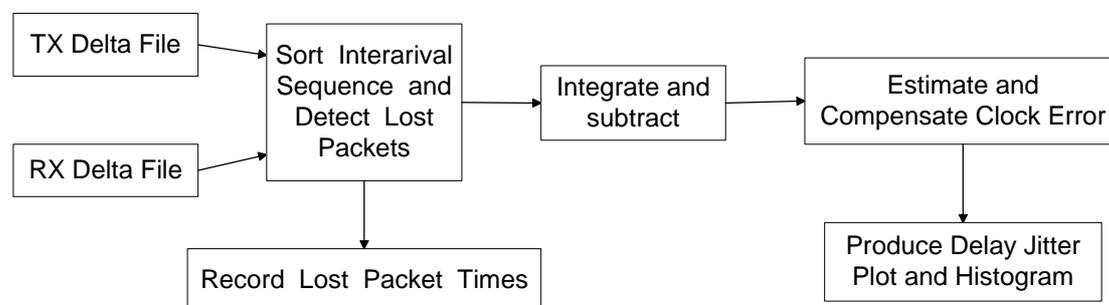


圖 2.2 品質參數估算流程圖

其詳細演算過程描述如下：

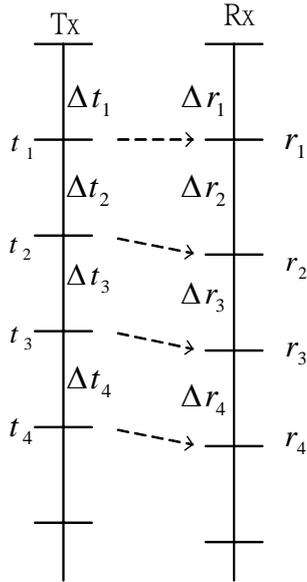
1. 首先此一演算法分別累加接收與傳送的 interarrival 表內的 interarrival 時間以獲得每個封包相對於接收/傳送第一個封包的累加時間。
2. 接著利用接收資訊中的識別碼(利用 16 位元片段識別碼計數器產生)，即可將接收封包與其相對應的累加時間依正確順序排列。然後在將每個傳送序列識別碼與排序完成的接收序列識別碼進行比對，依此方法找出的漏失封包及其對應的傳送時間資訊則轉移到另外一個獨立的檔案列表中。
3. 再將已排除漏失封包的傳送端與接收端之封包序列識別碼重新查

驗，以確保兩端資訊對應正確。若發生識別碼不匹配的狀況則表示有不正確的資料存在。

4. 將重新排序的傳送與接收序列之累加 interarrival 時間相減，獲得累加相差時間，再利用其與累加封包數目的近似線性關係進行線性逼近(linear regression)，所求得的斜率再除以平均接收封包 interarrival 時間即為平均估測延遲與誤差時間在接收封包 interarrival 時間所佔的比例。

5. 最後將每個接收封包的 interarrival 時間先扣除平均估測延遲與誤差時間所佔的部分，再減去對應傳送封包的 interarrival 時間即可獲得每個封包的延遲擾動。而由於整個封包延遲時間串是相對於第一個封包從傳送端到接收端傳送的時間長短，因此有些數值可能會出現負值。但是封包不可能在開始傳送之前到達—此延遲必須大於零。因此最後的時間延遲擾動數值串需至少往上平移到使其最小值為零。

此外，將於圖 2.3 舉一實例詳細說明此演算法計算過程，我們可以發現，若不針對時脈誤差進行修正，傳統演算法所估測的延遲擾動皆會受到時脈誤差干擾，無法獲得準確的品質評估參數。



$\Delta t_1, \Delta t_2, \Delta t_3, \Delta t_4, \dots$ 表示傳送端的 interarrival 時間
 $\Delta r_1, \Delta r_2, \Delta r_3, \Delta r_4, \dots$ 表示接收端的 interarrival 時間
 $t_1, t_2, t_3, t_4, \dots$ 表示每個封包相對於第一個封包的傳送時間
 $r_1, r_2, r_3, r_4, \dots$ 表示每個封包相對於第一個封包的接收時間
 $\Delta d_1, \Delta d_2, \Delta d_3, \Delta d_4, \dots$ 表示每個封包的延遲時間
 $\Delta j_1, \Delta j_2, \Delta j_3, \Delta j_4, \dots$ 表示每個封包的 delay jitter
 e 表示每秒接收端時脈相對於傳送端時脈的時脈誤差 (ms/s)

傳統演算法

計算延遲

$$\begin{aligned} \hat{\Delta d}_1 &= \Delta r_1 - \Delta t_1 = \Delta d_1 + e\Delta r_1 \\ \hat{\Delta d}_2 &= \Delta r_2 - \Delta t_2 = \Delta d_2 + e\Delta r_2 \\ \hat{\Delta d}_3 &= \Delta r_3 - \Delta t_3 = \Delta d_3 + e\Delta r_3 \\ \hat{\Delta d}_4 &= \Delta r_4 - \Delta t_4 = \Delta d_4 + e\Delta r_4 \end{aligned}$$

計算 delay jitter

$$\begin{aligned} \hat{\Delta j}_1 &= \hat{\Delta d}_2 - \hat{\Delta d}_1 = \Delta d_2 - \Delta d_1 + e(\Delta r_2 - \Delta r_1) = \Delta j_1 + e(\Delta r_2 - \Delta r_1) \\ \hat{\Delta j}_2 &= \hat{\Delta d}_3 - \hat{\Delta d}_2 = \Delta d_3 - \Delta d_2 + e(\Delta r_3 - \Delta r_2) = \Delta j_2 + e(\Delta r_3 - \Delta r_2) \\ \hat{\Delta j}_3 &= \hat{\Delta d}_4 - \hat{\Delta d}_3 = \Delta d_4 - \Delta d_3 + e(\Delta r_4 - \Delta r_3) = \Delta j_3 + e(\Delta r_4 - \Delta r_3) \end{aligned}$$

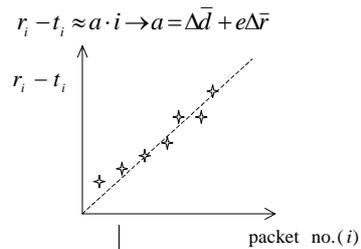
採用傳統演算法估算出的 delay jitter 會受到時脈誤差的影響

提出的演算法

計算累加相差時間

$$\begin{aligned} r_1 - t_1 &= \Delta d_1 + e\Delta r_1 \approx (\Delta \bar{d} + e\Delta \bar{r}) + \Delta j_1 \\ r_2 - t_2 &= \Delta d_1 + \Delta d_2 + e(\Delta r_1 + \Delta r_2) \approx 2(\Delta \bar{d} + e\Delta \bar{r}) + \Delta j_2 \\ r_3 - t_3 &= \Delta d_1 + \Delta d_2 + \Delta d_3 + e(\Delta r_1 + \Delta r_2 + \Delta r_3) \approx 3(\Delta \bar{d} + e\Delta \bar{r}) + \Delta j_3 \\ r_4 - t_4 &= \Delta d_1 + \Delta d_2 + \Delta d_3 + \Delta d_4 + e(\Delta r_1 + \Delta r_2 + \Delta r_3 + \Delta r_4) \approx 4(\Delta \bar{d} + e\Delta \bar{r}) + \Delta j_4 \end{aligned}$$

線性逼近求出代表平均估測延遲與誤差時間的斜率



計算 delay jitter

$$\hat{\Delta j}_i = \Delta r_i \left(1 - \frac{a}{\Delta \bar{r}}\right) - \Delta t_i$$

採用此演算法估算出的 delay jitter 已經減去時脈誤差的影響

圖 2.3 演算法計算過程詳細說明實例

2.4 系統模擬與討論

接著，以以太網路(Ethernet)環境進行量測，考慮三種不同的傳輸狀況，分別敘述如下：

狀況一：傳送端與接收端皆位於交通大學光復校區工程四館 812A 實驗室，使用配備相同的測量用電腦(Pentium IV 2.4GHz)。

狀況二：傳送端與接收端皆位於交通大學工程四館 812A 實驗室，但使用配備不同的測量用電腦(傳送端為 Pentium II 350MHz，接收端為 Pentium IV 2.4GHz)。



狀況三：傳送端位於交通大學博愛校區宿舍網路，接收端位於交通大學工程四館 812A 室，使用配備相同的測量用電腦(Pentium IV 2.4GHz)。

圖 2.4、2.5 以及 2.6 分別為狀況一、二以及三的測量數據與時間延遲擾動曲線圖。由圖 2.5 的累加相差時間曲線圖可以發現，狀況二的曲線斜率較大，雖然網路傳輸環境相同，但配備不同的電腦所造成的延遲與時脈誤差較大所造成的。而由圖 2.6 的累加相差時間曲線圖可以看出，其曲線斜率非常小，這是因為狀況三的封包傳輸是不同建築物

間的長距離傳輸，傳輸時間差所造成的影響會比延遲與時脈誤差影響為大。另外，在時間延遲擾動曲線時間圖中，也可以發現長距離的傳輸時間延遲擾動的變化較短距離為劇烈。狀況一與二的傳輸距離較短，封包傳送時間也較為穩定，因此累加相差時間主要是受延遲與時脈誤差影響所造成的。

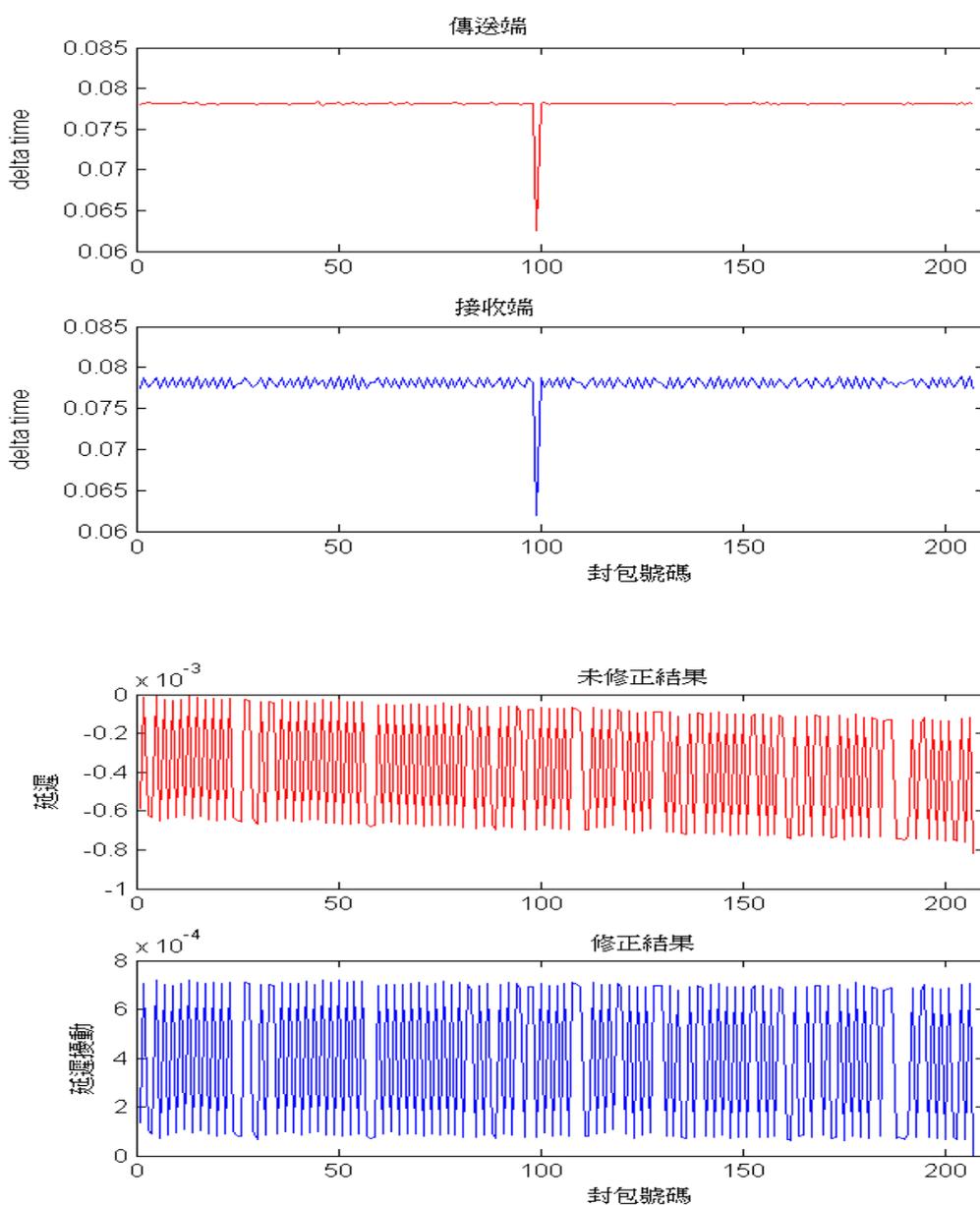


圖 2.4 狀況一之測量數據與延遲擾動曲線圖

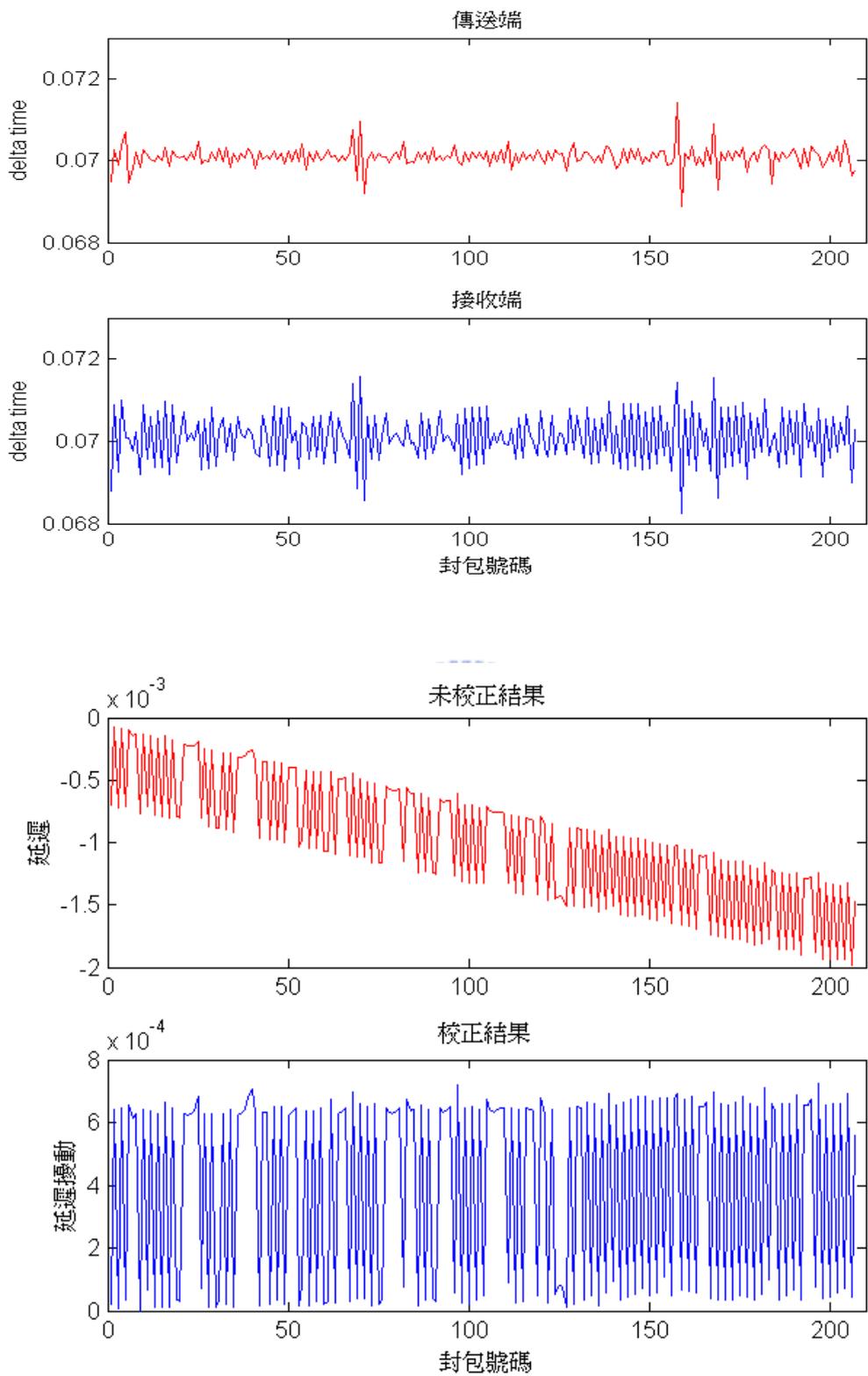


圖 2.5 狀況二之測量數據與延遲擾動曲線圖

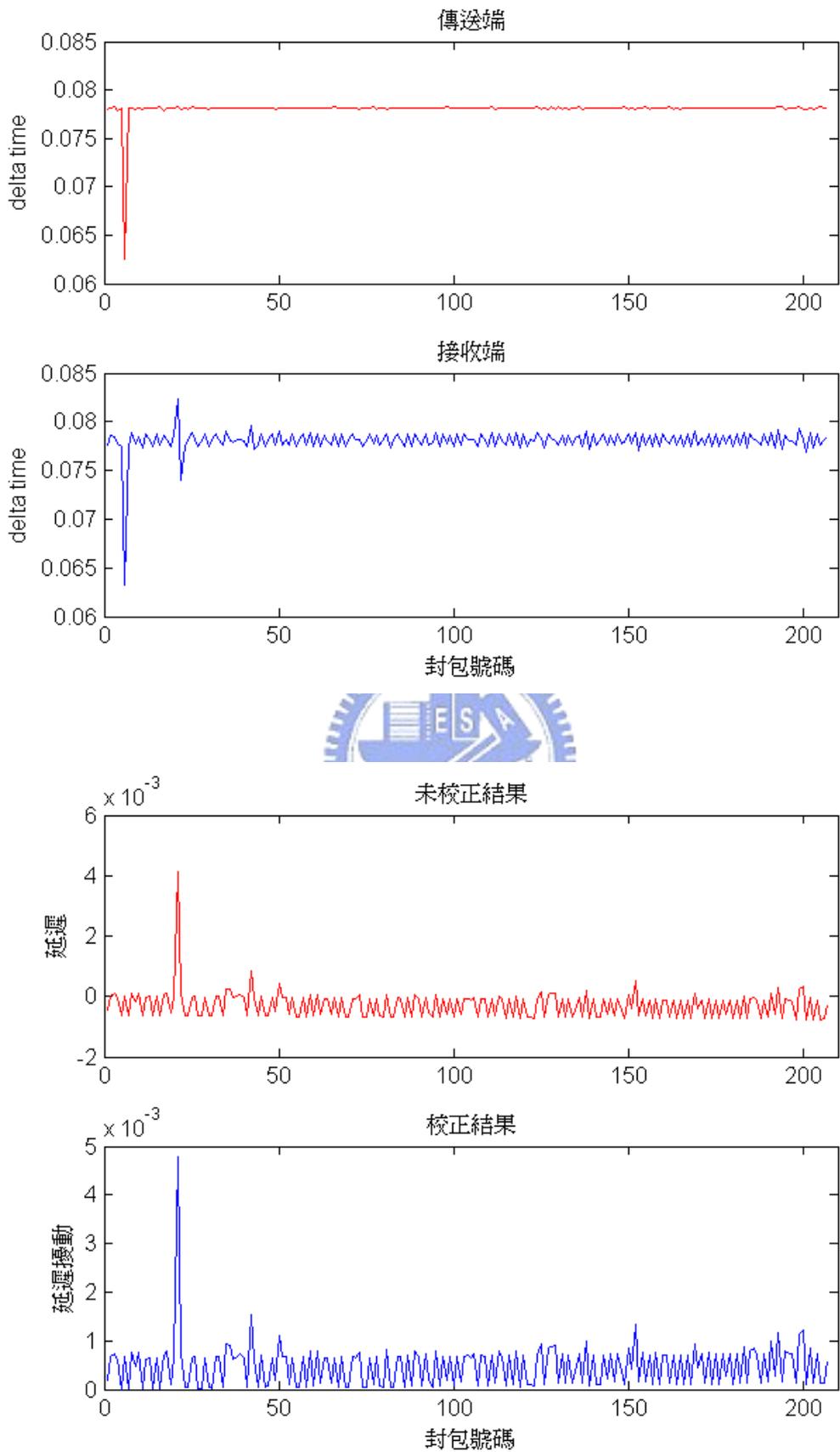


圖 2.6 狀況三之測量數據與延遲擾動曲線圖

2.5 網路電話(VoIP) QoS 之應用

在能夠準確地計算時間延遲擾動與封包漏失等重要的統計資訊之後，本節將描述如何利用此資訊有效應用於 VoIP 通訊系統的 QoS(quality of service)評估中。

在網路傳輸過程中，發生封包時間延遲擾動會嚴重影響 VoIP 的通訊品質，因此其機率分布，特別是標準差對於 VoIP 網路基準統計資訊而言十分有用。由於接收端的 interarrival 時間是代表接收封包間的時間差，即為該封包傳送 interarrival 時間與時間延遲擾動時間的總和。當已知或是可以模擬出傳送端 interarrival 分布之後，有了描述延遲特性快速移動部分的 delay jitter 時間，就可以測定接收端 interarrival 分布。因此除了分析網路狀況之外，網路時間延遲擾動估算技術也能有效提昇 VoIP 系統效能評估的準確性，特別是非固定常數的 interarrival 時間。

VoIP 的通訊品質是取決於延遲與漏失，然而兩者卻無法兼顧。因此 VoIP 通訊系統通常會採用擾動補償緩衝器技術就此兩項品質量測取得折衷。在網路傳輸時，一旦任何封包到達接收端延遲的時間大於擾動補償緩衝器長度時，則被視為一丟棄封包。若此緩衝器長度較小，則延遲較短，但漏失封包的比率會增加；若此緩衝器長度較大，則延遲較長，但漏失封包的比率會降低。因此如何決定一適當的擾動

補償緩衝器長度以符合個別的網路環境需求是非常重要的。而我們可以利用上一小節所求得的延遲擾動時間串進行統計，即可畫出擾動補償曲線圖，如圖 2.7、2.8 所示。經由此曲線圖，即可以在已知系統可允許最高封包丟棄率(<10 %)的情況下決定出適當的擾動補償緩衝器長度，使 VoIP 通訊系統能夠有最佳的效能。有鑑於實驗進行的距離還不夠遙遠，所需的補償長度顯得微不足道，隨著距離的增加，延遲擾動的效應較明顯，因此需要較長的擾動補償。

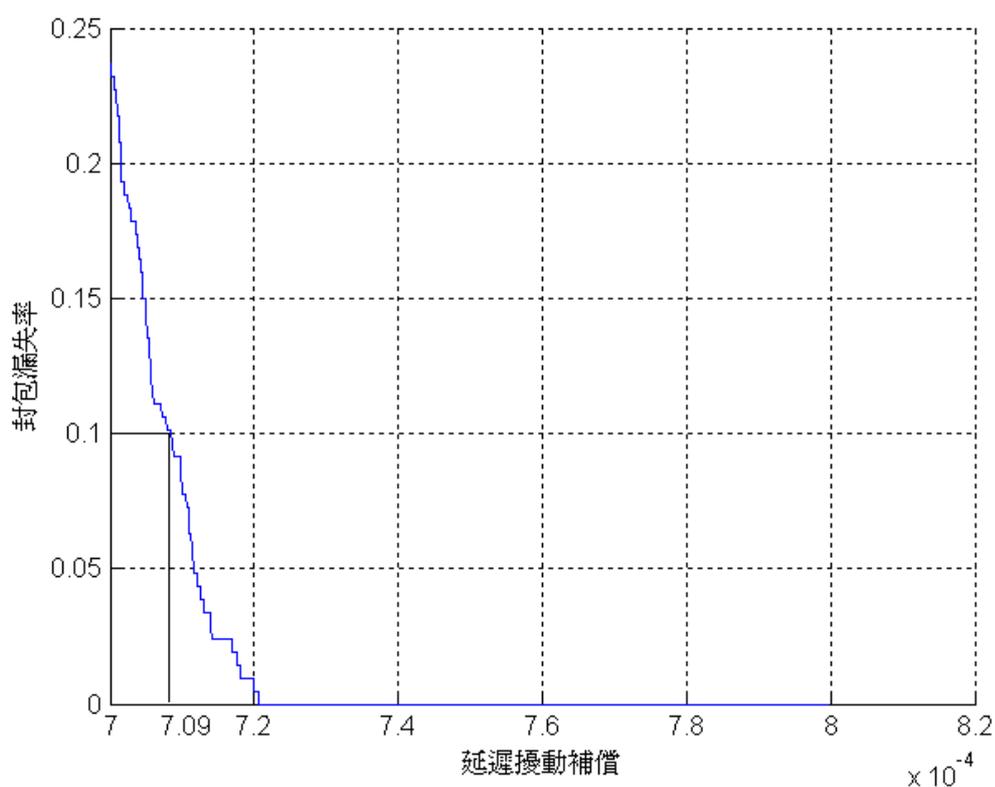


圖 2.7 狀況一的延遲擾動補償與封包漏失關係

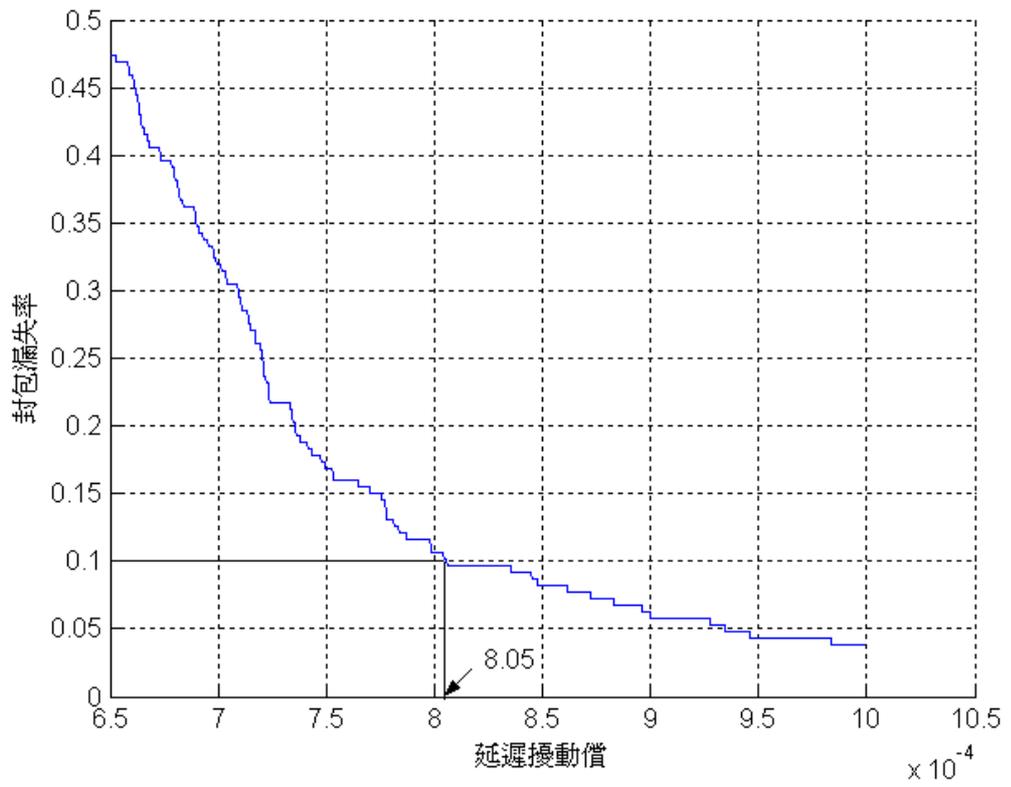


圖 2.8 狀況三的延遲擾動補償與封包漏失關係



第三章 通話品質預測模型

近年來由於網際網路電話(VoIP)低廉的通話費用以及更有效率的網路運用等種種優點，人們對於聲音能利用網路當傳輸媒介的接受度因而逐年增加。然而大眾已經習慣於傳統有線電話與行動電話優越的通話品質(toll quality)，因此在享受網路電話所帶來的好處之際勢必也會對通話品質做某種程度的要求，不過直至今日我們仍無法明確表示網際網路在語音品質這個部分可以達到何種程度。但是對於網路系統規劃者而言，就必須要有一個具體的音質評量指標供作參考，進而建構並調整系統關鍵元件參數之用，以確保使用者在通話中有較佳的語音品質且穩定的通話效能。所以我們必需去了解哪些因素會影響整體系統服務品質與效能，進而整合推導出一項能具體反應網際網路通話的音質評量指標模型。

3.1 主觀聽覺測試

傳統對於通話品質的界定，最直接的方式是以人類的主觀聽覺來判斷音質好壞，然而對於這種主觀音質的感受還是需要某種制定的量值用以區分程度差異。ITU 在標準規格[13, 14]中制定了平均評分分數(Mean Opinion Score，MOS)，評分的等級從感覺最好的 5 分到最差的 1 分。然而主觀感受的等級劃分並非只針對聲音的品質，另

外還有許多判斷方式。依國際電信聯盟(International Telecommunication Union, ITU)標準 P. 800 列出以下不同形式的評量基準：

[1] 絕對分類評分(Absolute Category Rating, ACR)：亦即不需要已

知熟悉系統的評比分數來做比較，由於沒有參考值可供作比較，因此這個方式純粹是測試者絕對的心裡感受而無法反應系統的好壞，例如當 MOS 為 4.3 時，我們可以把這樣的音質解釋成比一般品質還好一點，卻無法判斷是否極佳或是一般的系統，倘若有個不錯的系統的 MOS 為 4.1，就可以反應出這個系統音質為極佳，但若是有一般的系統的 MOS 為 4.7，那我們反而認為這樣的系統音質稍差。它可以是依照整體音質來分等級(Listening quality score)，也可以是了解語義需要集中多少注意力來分等級(Listening-effort score)，也就是對詞句的辨識度。前者比較針對實驗用的評估，而後者則傾向測試者的心理層面，因為當聽到一串語句時，直覺的反應就是去辨識語句的內容。

[2] 衰減分類評分(Degradation Category Rating, DCR)：當實驗因

子對於測試用的語句所造成的音質損害甚小，導致無法利用絕對分類來分等級時，此時就搭配尚未做語音處理的參考音源來做比較並依據失真程度來加以評分，這種評分通常是用來找出系統整合後的最佳化。

[3] 比較分類評分(Comparison Category Rating, CCR)：。這個方式基本上與 DCR 是很相似的，DCR 在聲音撥放程序上，首先聆聽的部分是尚未處理過的語音，接下來再去聆聽經過一些技術處理過的聲音再去分等級。而 CCR 的程序上則比 DCR 多出一種即先聆聽處理過後的語句再去聆聽未處理過的。聆聽者根據第二次聆聽的感覺再與第一次做比較。CCR 比 DCR 多一項好處也就是可以評斷經過語音處理過後的語音品質是否提升或是下降。原理在於語音編碼處理會抑制或是消除背景雜音，因此對於編碼過後的音質很有可能比原有的音質來的好，例如雜音消除系統，在這樣的情況，這種的評分就相當的有用。



表 3.1，3.2 與 3.3 分別說明了 MOS 與上述評比方式的對應關係。接下來描述主觀聽覺測試的過程，測試者是經由特定條件挑選出來的，處在特別設計過的房間，裡頭的噪音以及其他重要的環境因子皆被控制在某一種適合測試的程度來進行聆聽實驗。欲測試語句會預先在另外一間週遭噪音也是控制在相當低的層級下進行錄音，由於考量測試的準確度，每一段的語句大約會維持 2 到 3 秒，當然這些語句彼此沒有明顯的關聯性。經過語音編碼處理後再改變網路模擬用的參數因子，包括輸入不同語音能量層級(Speech input levels)、聆聽的能量層級(Listening levels)、隨機或叢發性錯誤、背景雜訊、編碼連結、不

同語音編碼方式的相容性等傳輸因子。所有測試者去聆聽撥放出來的聲音，並針對欲評量的方式打上分數，最後統計平均所有測試者的分數來當評比結果。從用戶角度看，通常認為 MOS 4.0~4.5 分為高品質，達到長途電話網的音質要求。MOS 3.5 分左右稱作普通音質，這時聽者能感覺到音質有所下降，但不影響正常的通話，可以滿足多數通信系統使用要求。MOS 3.0 分以下常稱為合成音質，這種語音一般只有足夠的可懂度，但是缺乏自然度，且不容易識別講話者。

由於所有的測試都是憑藉人耳的主觀聽覺來評分，往往會因為評分者當時對於環境的感受以及態度而直接影響到整個評分結果，因此難以達到一致且客觀的標準認定。更由於事前需詳盡準備各類測試用的環境設定，測試耗時且需花費相當龐大的人事經費，對於例行性的監控網路程序而言，這樣的評量方式就顯得沒有效率且不實際。另外就系統設計規劃而言，上述的測試方案都沒有考量到網路層服務品質的影響因素(延遲，擾動，漏失)，因此無法就網路傳輸所造成的音質損害問題加以處理改善。

了解語意所花的努力	音質	分數
完全不需要	極佳	5
開始需要專注聆聽	佳	4
適度的專注	普通	3
要花相當多的精神	差	2
即使專心聆聽也無法了解	極差	1

表 3.1 絕對分類與分數的對應關係

音質衰減	分數
完全聽不出來	5
可聽出來但不會造成困擾	4
些微困擾	3
造成困擾	2
非常大的困擾	1

表 3.2 衰減分類與分數的對應關係

針對第二種音質而與第一種的音質比較	分數
更好	3
很好	2
些微好	1
差不多	0
些微糟糕	-1
很糟糕	-2
更糟糕	-3

表 3.3 比較分類與分數的對應關係

3.2 音質評量指標

正因為主觀聽覺測試無法反應接收兩端點之間網路傳輸所造成的音質損害，因此 ITU 制定一個具體的音質評量模型(E-model)，採用主觀聽覺測試先建立不同因子所對應的音質損害，再加以整合計算得到最後的評分 R，提供系統規劃及調整系統關鍵元件參數之用。E 模型的式子如下：

$$R = R_0 - I_s - I_d - I_e + A \quad (3.1)$$

R_0 ：訊號雜音比，雜音部分包括背景噪音以及電路雜訊

I_s ：與語音信號同時產生的音質損害因子，包括量化、連接雜訊和側音(sidetone)帶來的干擾

I_d ：語音遲滯(包括通話迴聲)造成的音質損害因子

I_e ：低位元率語音編碼處理和封包漏失所造成的音質損害因子

A：補償損害因子(compensation impairment factor)，用以補償用戶基於接聽的方便而能忍受音質的影響，如行動電話

量測音質的 R 值範圍可以從最好的 100 到最差的 0，然而實際對於聲音品質可以接受的下限為 50。同時在 E 模型中，也定義了 R 值與平均評分分數(MOS)之間對應的關係，可以避免主觀評量繁瑣的人工測試過程，如圖 3.1 所示。

For $R < 0$ $MOS = 1$

For $0 < R < 100$ $MOS = 1 + 0.035R + 7 \times 10^{-6} R(R - 60)(100 - R)$

For $R > 100$ $MOS = 4.5$

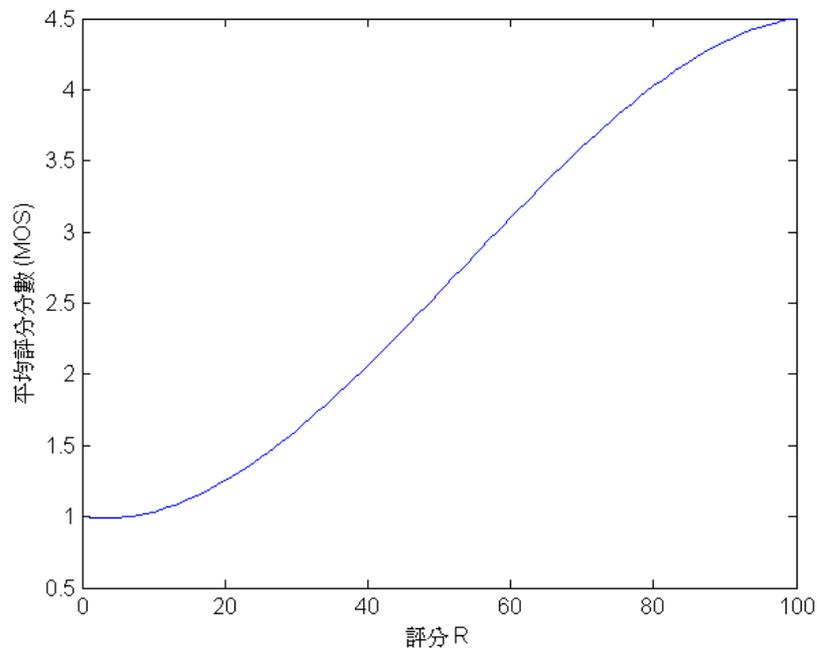


圖 3.1 R 與 MOS 的轉換關係

我們將 R 與音質的對應關係整理成表 3.4:

評分因子	分數	品質
$90 < R < 100$	4.34-4.5	極佳
$80 < R < 90$	4.03-4.34	佳
$70 < R < 80$	3.60-4.03	普通
$60 < R < 70$	3.10-3.60	差
$50 < R < 60$	2.58-3.10	極差

表 3.4 R 與音質對應關係

由於我們是針對網路傳送層來探討音質損害，因此對於 R_0 和 I_s 而言，它們與網路傳送過程並沒有直接的關係。因此我們可以採用 ITU 所設定的初始值，進而簡化 R 的計算方式，直接針對通道特性及系統架構兩層面來評估音質。如(3.2)所示[8]，

$$R(d, r, plr) = 94.2 - I_d(d) - I_e(r, plr) \quad (3.2)$$

其中 d 為單一路徑口對耳延遲(mouth-to-ear delay)， r 是編碼位元率， plr 則是封包漏失率。針對 I_e 進一步分析顯示，影響因素有低位元率語音編碼處理所造成的訊號失真，以及在傳輸過程中因網路擁擠或其他不可預知因素所導致的封包漏失。可分開討論如下，

$$I_e(r, plr) = I_{ec}(r) + I_{epl}(plr) \quad (3.3)$$

其中 I_{ec} 表示語音編碼造成的音質損害， I_{epl} 則表示封包漏失所造成的音質損害。

3.2.1 語音編碼損害因子- I_{ec}

使用語音壓縮技術可以減少資料傳輸量，有效節省頻寬的使用。其編碼處理有許多選擇，如 G.711 PCM、G.729a CS-ACELP、G.723.1 MPC-MLQ，依位元率區隔不同模式所衍生的信號失真亦存在明顯差異。簡述如下：

[1] G.711 PCM: 由 ITU-T 於 1972 年制定的 64Kbits/sec 語音壓縮標準，也是傳統電話所採用的編碼模式，它是一種採用取樣導向(sample oriented)的波形編碼模式。由於人耳所能聽到的頻率範圍最高不超過 4KHz，根據 Nysquist 取樣定理採 8KHz 的取樣頻率，再以 8bits 量化個別的取樣值。由於人通常在特殊情況下才會提高或是降低音量，所以一般講話的音量範圍給予線性的量化，其他區域給予非線性的量化，依量化方式不同而分成兩種規格，歐規是採用 $A-law$ ，美規則是採用 $\mu-law$ 。



[2] G.729a CS-ACELP: 由 ITU-T 於 1995 年制定的 8Kbits/sec 語音壓縮標準，屬於一種音框導向(frame-oriented)的處理模式，將數位語音訊號切割成一連串固定長度為 10ms 的音框(80 個取樣值),並等分二個子音框(subframe)。G.729a 編碼器採用共軛架構代數碼激發線性預測技術(Conjugate-Structure Algebraic-Code-Excited Linear-Prediction, CS-ACELP)，兼具波形編碼以及參數編碼的優點。首先以十階線性預測濾波器係數來表示語音訊號的短時距相關特性，並利用最小誤差分析合成搜尋法找出語音訊號的長時距類周期激發源，最後利用多重脈衝訊號逼近其預測殘值。其中長時距類周期激發源及預測殘值的增益編碼部分，是使用共軛架構向量量化(conjugate-structure VQ)。

[3] G.723.1 MPC-MLQ: 為 ITU-T 於 1996 訂定的雙位元率 (6.3 Kbps 及 5.3 Kbps) 語音編碼壓縮標準，也是採用音框導向的處理模式。每個音框有 240 個數位化之取樣點。在編碼區塊的流程上，兩種速率編碼模式的參數計算過程大致相同，主要差異在於預測殘值編碼的不同。

每一種編碼標準均有其特定的聆聽 MOS(listening MOS) ，利用圖 3.1 即可求得其對應的 R 值。一般而言，聆聽 MOS 並未將延遲及封包漏失的音質損害納入考量，因此公式 (3.2) 可簡化為 $R = 94.2 - I_{ec}(r)$ ，而依此建立不同編碼標準與 I_{ec} 的對應關係如圖 3.2 所示

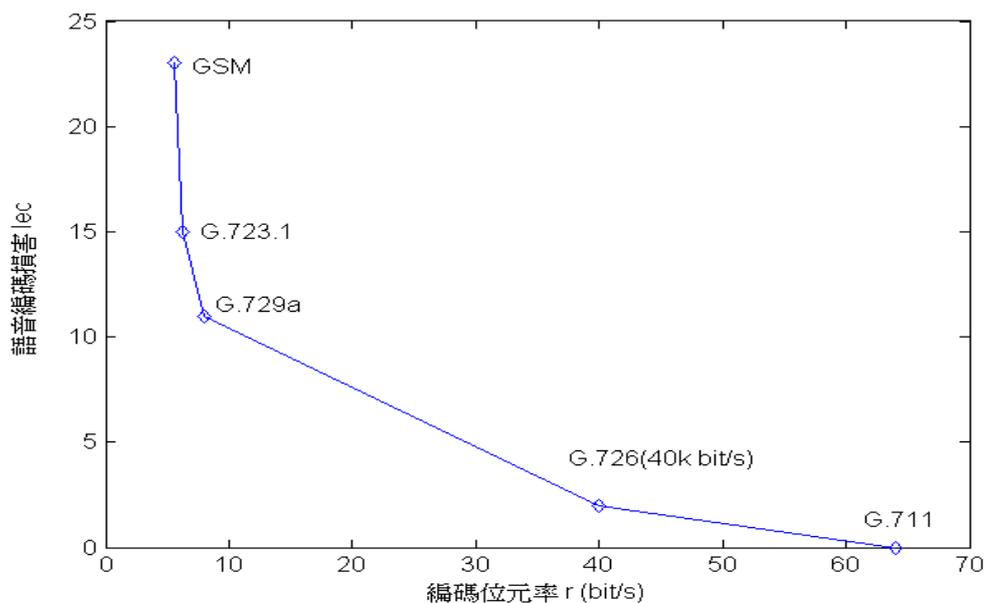


圖 3.2 編碼位元率與對應的音質損害

就圖 3.2 的關係可以看出，隨著編碼位元率的下降，音質的損害值明顯的增加，尤其是 G.729a，G.723.1 與 GSM，雖然三者位元率接近，造成的損害卻明顯的快速爬升。這是由於高壓縮率雖然能節省頻寬的使用，然而封包與封包之間的關聯性卻明顯的降低，在網路傳送語音封包時，若發生封包漏失的現象即有可能造成聲音斷斷續續有如被剪掉一樣。因此事先對封包做保護的動作有其必要性，此重要議題會在第四章探討。

3.2.2 封包漏失損害因子- I_{epi}

在上個小節提到聲音在一開始傳送時首先會經過語音編碼處理，由數據顯示造成的音質損害會隨著位元率的下降而提昇，而這小節主要是探討受到網路內部行為影響的音質損害。在網路中常常面臨到頻寬有限卻需傳送大量語音或資料封包，路由器需要更多時間消化而造成網路擁塞的現象，導致封包佇列時間過久而無法在預定時間內抵達終點，造成封包漏失的現象。若是資料封包可能會要求重送(ACK)的機制來改善，然而對有即時傳輸需求的語音封包而言，卻無法利用重送機制來做補強，使得原本整段語音會發生斷斷續續的現象。當然針對這樣的音質損害，接收端可採取適當的錯誤隱匿(error concealment)機制來處理。在附錄 A 中，我們將針對錯誤隱匿機制做一

簡單說明。前人研究[5]顯示不同語音編碼的封包漏失率所對應的損害因子 I_{epi} ，如圖 3.3 所示，可以近似為一條對數的軌跡。若以匹配曲線(curve fitting)的方式來逼近可得，

$$I_{epi}(plr) = a \ln(1 + b * plr) \quad (3.4)$$

其中參數 a 與 b 可以利用內插法得到 $a \approx 34.3$ ， $b \approx 12.8$ ，依此建立的匹配曲線畫在圖 3.3 供作比對。

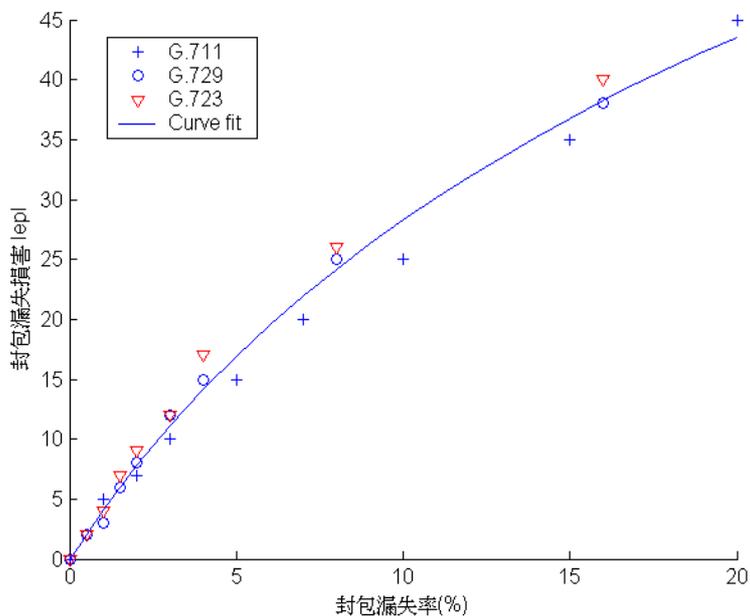


圖 3.3 封包漏失與對應的音質損害

3.2.3 延遲損害因子- I_d

就聆聽 MOS 而言，用戶往往對延遲比對封包漏失更能容忍，因為封包漏失會造成聽不清楚對方的話，而延遲並不會影響單向通話的音質。但就雙向的對話品質(Conversational MOS)而言，延遲增大

到一定程度以後，可能導致雙方同時講話或相互沈默，從而影響正常通話，減少雙方的互動。而造成延遲的因素有很多，諸如編碼與封裝處理造成延遲、傳送路徑延遲、撥放暫存器造成的延遲。

延遲所以造成雙向通話的音質損害，主要是因為網路電話所衍生的迴聲現象。語音經網路傳達而後撥放出來，此時接聽音的話筒會接收經由反射的聲音再傳遞給說話者，如圖 3.4 所示。通常要求兩端之間的延遲需小於 10ms，這樣的迴聲並不容易感覺出來，可當作適當的側音(sidetone)。

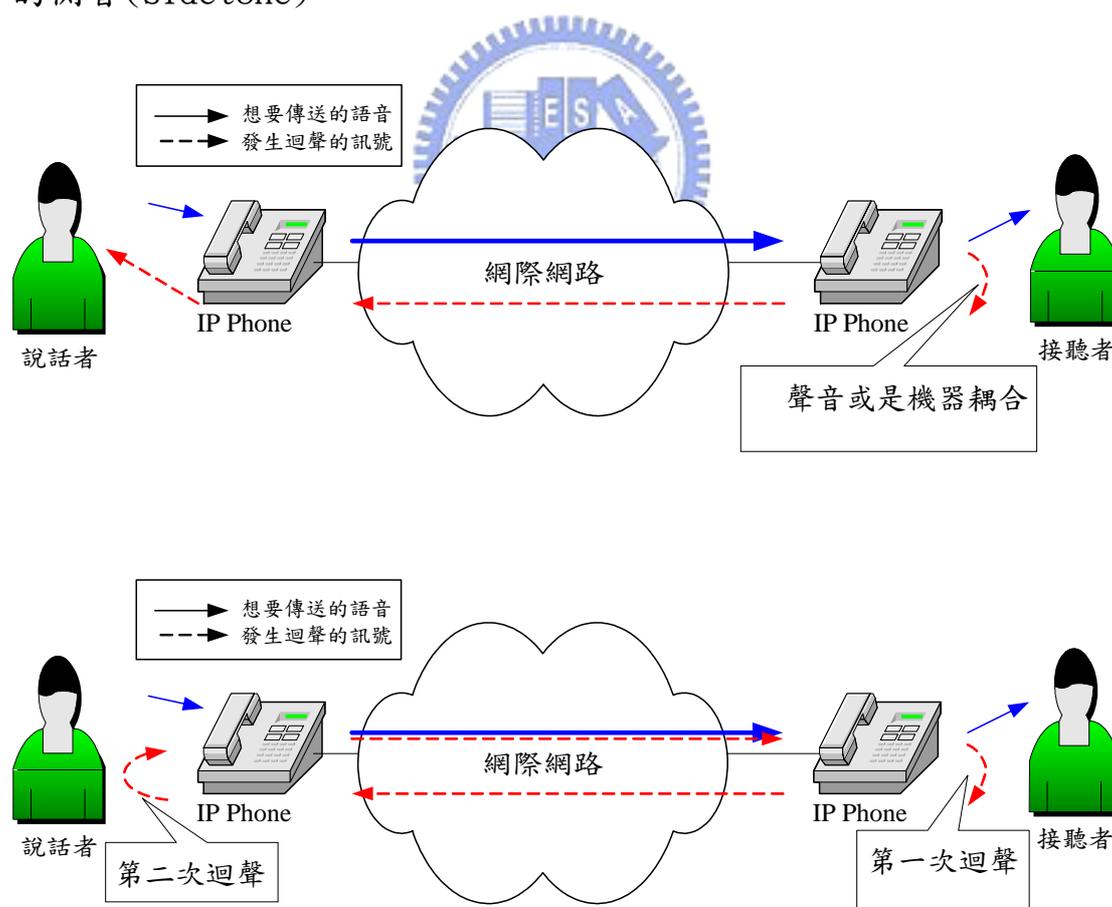


圖 3.4 單向通話迴音干擾

在前人研究[8]中，參考 ITU G.107 比對單一路徑口對耳的延遲與其

損害因子 I_d ，如圖 3.5 所示，再利用片段線性分析可推導得

$$I_d = 0.024d + 0.11(d - 177.3)H(d - 177.3) \quad (3.5)$$

其中 d 為單一路徑延遲，而 $H(x) = 0$ if $x < 0$ ， $H(x) = 1$ if $x \geq 0$

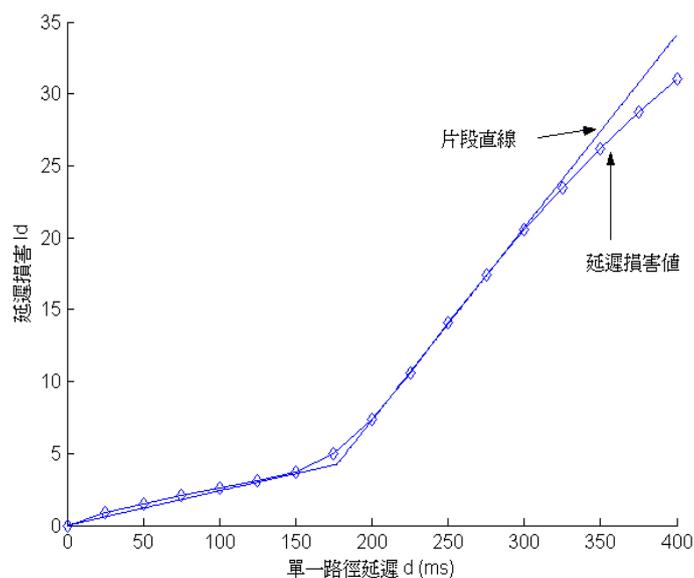


圖 3.5 單一路徑延遲與對應的音質損害

第四章 語音傳輸系統規劃

現今依賴網路來傳輸各種型態的資料流量日益增加，然而由於傳輸通道的雜訊存在以及網路內層的設備處理問題，造成封包漏失或是傳輸錯誤的發生機率增加。一可行的解決方案是在傳送端事先控制這些問題，以確保資料的完整性。本章的課題在於探討不同的錯誤控制方式，並建立錯誤回復處理後殘留錯誤的機率模型。

4.1 封包漏失回復機制



網路傳輸協定主要有兩種，TCP (Transport Control Protocol)和UDP (User Datagram Protocol)。TCP 會先建立兩端點的連線，並使用重傳機制來保證資料不會漏失且不會脫序的可靠性機制。而 UDP 則是屬於無連接模式，每個封包透過路由器(Router)傳送到目的地，無論接收端收到封包與否，UDP 皆無須重傳，因此適用於需即時服務的語音或影像資料，缺點則是不能保證資料傳遞的可靠度。

網路傳輸過程中不可避免會有封包漏失的情形發生，尤其當頻寬不足以傳送有傳輸速度限制的資料，例如聲音或者是影像訊號等。一般而言，封包漏失概分為兩種狀況，第一是傳輸層沒有收到封包，第二是收到封包的內容有錯誤，同樣會被丟棄，因此適當的漏失封包控

制或是回復機制有其必要性。最常見的兩種方式，一種是自動重傳要求(Automatic Repeat reQuest, ARQ)，主要是架構在 TCP 上，透過 ARQ 來傳輸可靠性要求較高的資料，如檔案傳輸或是電子信件等。顯然這個機制並不適合即時性的網路語音傳輸以及多重播送骨幹(Multicast backbone)，因為 ARQ 本身並不做任何錯誤更正，僅偵測錯誤並回覆內含漏失封包序號的 NACK 訊號給傳送端，要求重傳漏失的封包。而這樣一連串交互的控制訊息，將造成傳輸上的延遲，增加網路資源的耗損。

相較於資料傳輸，語音封包的傳輸對網路延遲是較敏感的，但是人耳的聽覺卻可以允許傳輸時些許的封包漏失(< 10%)，因為這樣的漏失所造成的音質損害是人類感觀所能容忍的上限，因此發展出另外一種前向錯誤控制碼(Forward Error Control, FEC)的機制，它是傳送端在傳遞一個資料封包前加上冗餘資訊(redundant information)，接收端可以利用這個資訊來做封包漏失的回復工作。另外必須注意的是，使用 FEC 的系統除了原始資料以外還要攜帶額外資料，接收端必須收到兩者再經過額外計算，才能回復可能漏失的原始資料。在考量冗餘資料會增加傳輸延遲以及接收端的計算複雜度之下，維持適當的冗餘資料量是必要的。此外在叢發性封包漏失(bursty packet lost)情形下，FEC 的效能會降低。表 4.1 整理了兩種方式的優缺點。有鑑於

兩者之間各有其優缺點，也有學者研究如何去合併這兩者的優點 (Hybrid ARQ)，有效減少封包漏失機率進而減少重傳機制也增加網路頻寬的使用效能，當應用在多重播送骨幹時的效能的確較單一使用 FEC 或 ARQ 來得好。

	優點	缺點
FEC	1. 不需重傳所以回復速度較快，增加傳輸效能 2. 主從架構的網路協定較簡單	需傳送冗餘資訊因而浪費頻寬
ARQ	無須傳輸額外資訊，因而不會佔去額外頻寬	延遲時間過長，造成網路擁塞

表 4.1 FEC 與 ARQ 的比較

4.2 前向錯誤控制

FEC 是一種消弭雜訊效應的技術，廣泛使用於各種數位資訊系統。舉凡家中的視聽設備，如 CD、數位錄音帶、數位電視等，以及辦公室中用來儲存資料的硬碟，甚至衛星通訊系統等均可見其相關應用。它的原理在於傳輸原始資料的同時附加額外保護資訊，接收端可以利用這些額外資訊來回復漏失的封包。傳輸媒介的不同，資料在傳輸的過程中發生錯誤的屬性就有所不同。如在無線網路或是窄頻通道的環境下，經常發生封包內部的位元錯誤，較強調針對錯誤的位元做更正；而在有線或是寬頻通道的環境下，位元錯誤就不是那麼的顯

著，主要是網路壅塞所衍生的封包漏失。依照不同層級的錯誤回復需求而劃分為三大類，依序為位元層級(Bit level)、位元組層級(Byte level)、封包或是訊框層級(Packet level or Frame level)

- 位元層級(Bit level)

在電路交換網路(Circuit switch network)中，較常發生位元錯誤，因此可以藉由數位訊號處理(DSP)晶片或是特殊應用的積體電路等一些典型的硬體設備在實體層(Physical layer)實現其錯誤控制。針對頻寬受限制的通道而言，通常使用格子碼(trellis code)加上 Viterbi 解碼器。而在寬頻通道的環境下，則使用區塊碼(Block code)或是迴旋碼(Convolutional code)，配合 Viterbi 解碼演算法。

- 位元組層級(Byte level)

對無線通訊而言，訊號由於受到反射、繞射或是都普勒效應的影響會產生多路徑衰落的現象，導致位元發生錯誤的情況相當顯著，倘若沒在鏈結層(data link layer, DLL)做更正的動作，在傳輸層(transport layer)將會是叢發性的封包漏失(bursty packet loss)。在鏈結層，每一個封包都會加上循環冗餘檢查(Cyclic Redundancy Checksums, CRCs)，用以檢查發生錯誤的真實資料位元。由於 FEC 解碼的計算相當複雜，因此在過去這些 CRCs 位元只當作檢測之用，然而近年來由於計

算能力提昇，可以利用這些資訊來修復錯誤的位元，常用的方式如 Reed-Solomon (RS) 編碼器。

- 封包層級(Packet level)

在網路傳送過程中錯誤發生可以區分為兩種，一是通道雜訊導致隨機的位元錯誤，另一種則是壅塞的連結導致封包漏失。在有線的網路環境裡，特別是光纖網路，位元發生錯誤的情況相當少，而當錯誤位元經由 CRCs 檢測到，這個封包在傳輸層就會被丟掉。因此不管是因隨機位元錯誤或是由於擁塞而丟棄封包，都會導致整個封包漏失的現象。換句話說，所要處理的問題不再針對位元錯誤，而是封包漏失，因而需在傳輸層針對封包漏失的問題做錯誤更正。在此研究中，我們將鎖定在封包層級的前向錯誤控制且採 Reed-Solomon (RS)編碼器。

在一連串的资料流裡，依回復用額外資訊的差異亦分為兩類模式，一個是完全與資料流的內容無關(Media independent)，另外一個則是與資料流的格式有關(Media specific)。

[1] 與資料流格式有關的前向錯誤更正-media specific FEC

通常應用在多人對講模式，主要是利用每一封包內攜帶其他封包的輔助資訊，萬一封包掉了，還可以利用成功接收封包的輔助資訊來做回復的動作，圖 4.1 說明其基本原理。原始封包會以較高位元率的

方式編碼，而冗餘的封包則是採用低位元率的模式編碼，冗餘的部分被後來的封包所背負著(piggy backed)，可以減少封包的數量。另外冗餘資訊可以多個封包產生，這樣的方式用來解決連續封包漏失的現象，如圖 4.2 所示。依此方式雖可以解決連續封包漏失的問題，若網路傳輸品質良好，錯誤發生機率很低的時候，發生叢發性的封包漏失可能性下降。這樣的機制顯然造成網路負擔，同時也因為要包裝這樣的封包勢必會增加處理時間而增加延遲。換句話說，並非無限制的增加額外的資訊，仍必須搭配網路的壅塞控制機制，否則網路的傳輸效能將因此而降低。

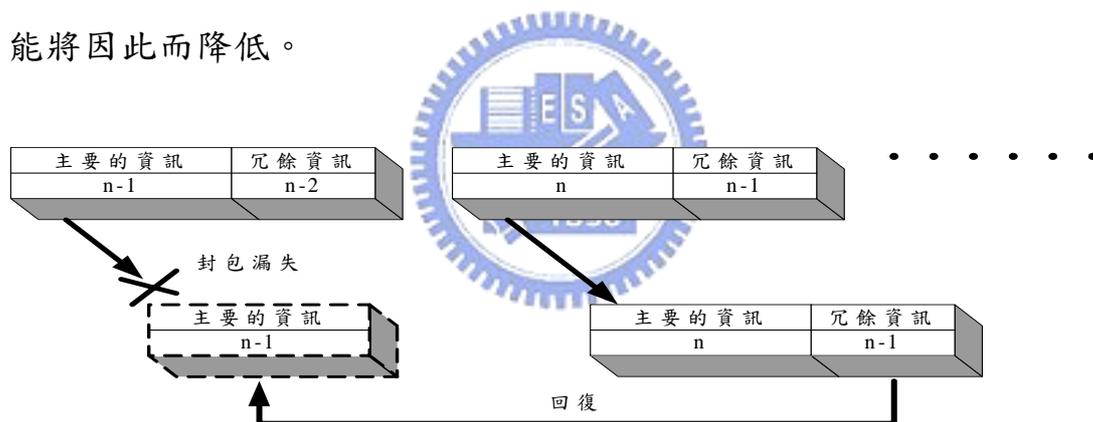


圖 4.1 與資料流格式有關的錯誤控制過程

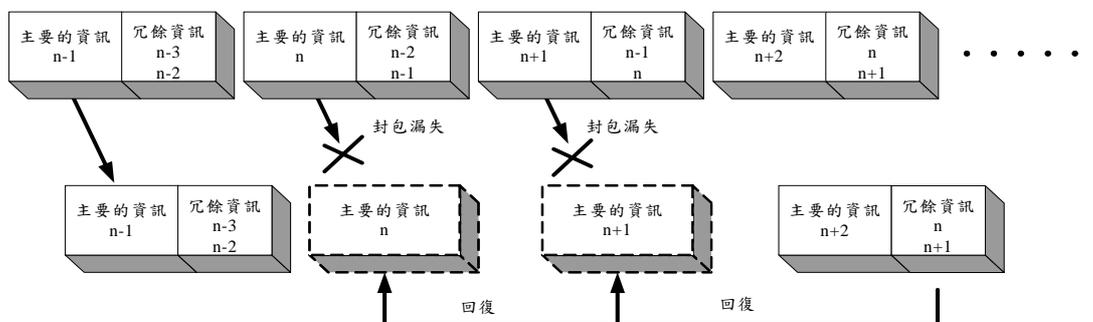


圖 4.2 多封包錯誤控制模式

[2] 與內容無關的前向錯誤更正-Media-Independent FEC

使用 (n, k) 區塊碼的編碼處理方式，是以 k 個訊息封包來產生 $n-k$ 個檢查封包，形成 n 個封包的區塊再傳遞出去，如圖 4.3 所示。訊息完整回覆的原則是 n 個封包中至少有 k 個被成功接收，則其他的漏失封包皆可被回復。針對任何一個特定封包而言，皆須等待所有封包被接收到才能進一步確認訊息的完整性，無形中衍生額外的延遲。如圖 4.4 所示，其中 T_p 表示封包間隔的時間，則額外的延遲可表示為

$$d_{FEC} = (k-1) \cdot T_p \quad (4.1)$$

有鑑於 d_{FEC} 在單一路徑的延遲佔了相當的比重，因此近年來有學者提出整合 FEC 與接收端的撥放暫存器[5]，在附錄 B 我們針對這個部分做一簡單說明。

在區塊碼的選擇方面，同位碼(Parity code)與 RS 碼是最常使用的錯誤更正方式。同位碼是每 n 個封包裡頭有 1 個是檢查封包，因此它僅只能回復單一封包漏失，圖 4.5 說明整個過程。相對而言，RS 碼的漏失回復能力較強，對於 RS 碼的過程會在 4.4 節做詳細描述。

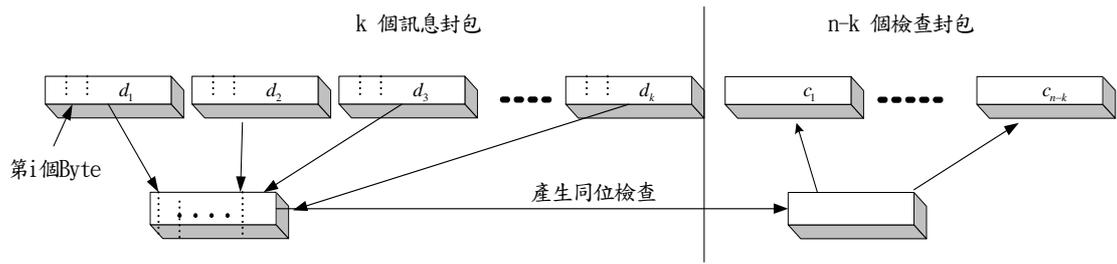


圖 4.3 檢查封包產生過程

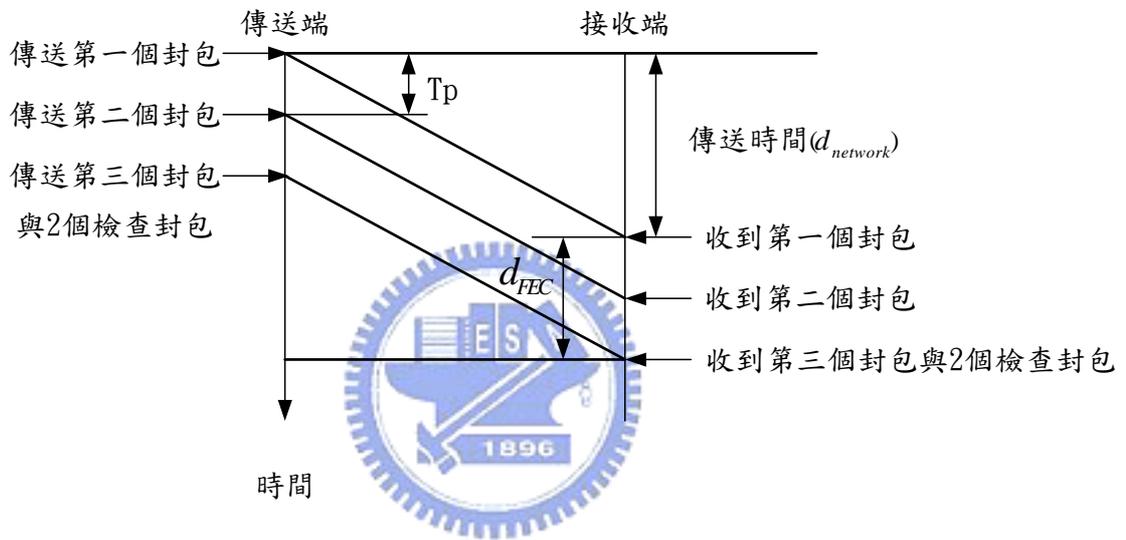


圖 4.4 (3,2)區塊碼所衍生的額外延遲

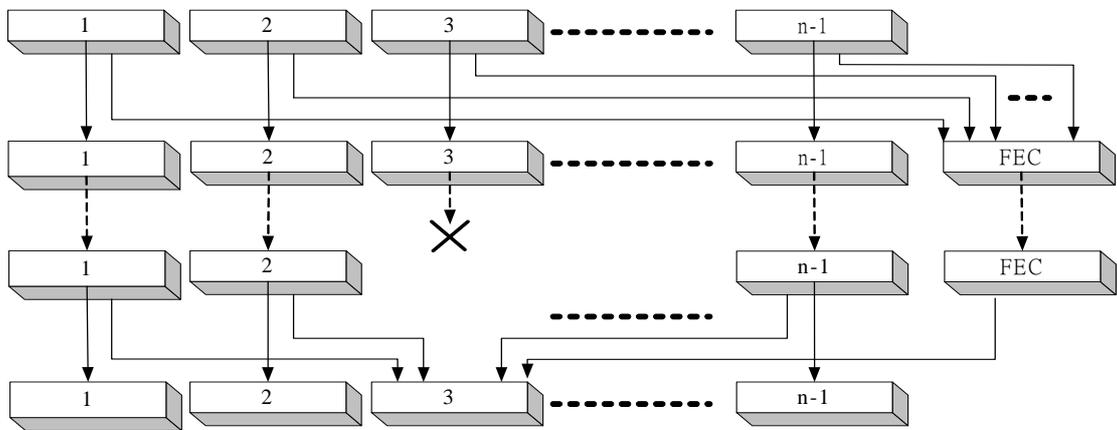


圖 4.5 同位碼回復錯誤的過程

4.3 吉伯特漏失模型

封包漏失的屬性可區分為隨機(random)以及叢發性(bursty)兩種，前向錯誤更正的效益會隨著漏失屬性的不同而存在明顯差異，因此漏失模型分析對 FEC 回復效能的掌握是很重要的一環。對於兩端點間的網路特性已經有許多學者提出相關模型，針對不同封包漏失屬性常採用隨機模型或有限狀態的馬可夫鏈模型(finite- state Markov chain model)，此處我們將運用吉伯特(Gilbert)模型進行有關叢發性錯誤之模擬。如圖 4.6，它是由兩個狀態來模擬網路漏失，其中 1 表示封包漏失而 0 表示成功接收，參數 p 與 q 分別表示 0 到 1 以及 1 到 0 的狀態移轉機率。更明確地說， $p = \Pr(Y_k = 1 | Y_{k-1} = 0)$ 且 $q = \Pr(Y_k = 0 | Y_{k-1} = 1)$ ， $P_c = 1 - q = \Pr[Y_k = 1 | Y_{k-1} = 1]$ ，其中 Y_k 表示第 k 個封包的接收狀態。

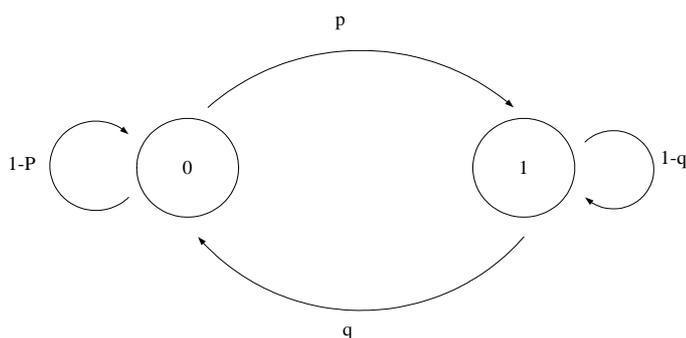


圖 4.6 吉伯特漏失模型

狀態 1 與狀態 0 的穩定機率(stationary probability)，分別為 $\pi_1 = \Pr(Y_k = 1)$ 與 $\pi_0 = \Pr(Y_k = 0)$ ，則可推導得

$$\begin{aligned}\pi_1 &\triangleq P(Y_k = 1) = P(Y_k = 1, Y_{k-1} = 0) + P(Y_k = 1, Y_{k-1} = 1) \\ &= P(Y_{k-1} = 0)p + P(Y_{k-1} = 1)(1-q) = \pi_0 \cdot p + \pi_1 \cdot (1-q)\end{aligned}$$

因為 $\pi_0 + \pi_1 = 1$ ，所以可求得 $\pi_0 = \frac{q}{p+q}$ ， $\pi_1 = \frac{p}{p+q}$ 。特別強調是在吉伯特模型假設下，原始封包漏失率 $p_u = \pi_1 = \frac{p}{p+q}$ ，叢發封包漏失的平均長度為 $\frac{1}{q}$ 。

此外定義一些機率分布， $p(i)$ 表示掉了一個封包後成功接收 $i-1$ 個封包直到第 i 個封包漏失的機率，

$$p(i) = \Pr(0^{i-1} | 1) = \Pr(\{Y_j = 0, \forall j \in \{1, \dots, i-1\}\} \cap \{Y_i = 1\} | \{Y_0 = 1\}) \quad (4.2)$$

而 $P(i)$ 則表示掉了一個封包後至少 $i-1$ 個封包成功接收的機率，

$$P(i) = \Pr(0^{i-1} | 1) = \Pr(\{Y_j = 0, \forall j \in \{1, \dots, i-1\}\} | \{Y_0 = 1\}) \quad (4.3)$$

其中 0^{i-1} 是指連續 $i-1$ 個封包被成功接收， Y_i 表示第 i 個封包的接收狀態，當 $Y_i = 0$ 表示成功接收， $Y_i = 1$ 則表示漏失。定義 $R(m, n)$ 表示發生一個封包掉落之後(或之前)的 $n-1$ 個封包內有 $m-1$ 個封包漏失的機率，可以利用疊代的方式[15]計算：

$$R(m, n) = \begin{cases} P(n) & , m=1, n \geq 1 \\ \sum_{i=1}^{n-m+1} p(i)R(m-1, n-i) & , 2 \leq m \leq n \end{cases} \quad (4.4)$$

同理可推， $S(m, n)$ 表示成功接收一個封包之後(或之前)的 $n-1$ 個封包內有 $m-1$ 個封包也是成功接收的機率，可以利用疊代的方式計算：

$$S(m, n) = \begin{cases} Q(n) & , m=1, n \geq 1 \\ \sum_{i=1}^{n-m+1} q(i)S(m-1, n-i) & , 2 \leq m \leq n \end{cases} \quad (4.5)$$

$$\text{其中 } q(i) = \Pr(1^{i-1}0 | 0) = \Pr(\{Y_j = 1, \forall j \in \{1, \dots, i-1\}\} \cap \{Y_i = 0\} | \{Y_0 = 0\}) \quad (4.6)$$

$$Q(i) = \Pr(1^{i-1} | 0) = \Pr(\{Y_j = 1, \forall j \in \{1, \dots, i-1\}\} | \{Y_0 = 0\}) \quad (4.7)$$

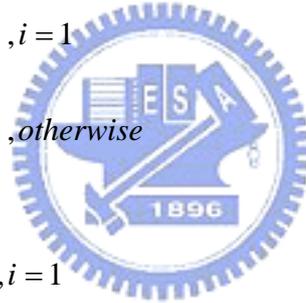
在吉伯特模型假設下， $p(i)$ 、 $P(i)$ 、 $q(i)$ 與 $Q(i)$ 可以推導得知

$$p(i) = \begin{cases} 1 - q & , i = 1 \\ q(1 - p)^{i-2} p & , otherwise \end{cases} \quad (4.8)$$

$$P(i) = \begin{cases} 1 & , i = 1 \\ q(1 - p)^{i-2} & , otherwise \end{cases} \quad (4.9)$$

$$q(i) = \begin{cases} 1 - p & , i = 1 \\ p(1 - q)^{i-2} q & , otherwise \end{cases} \quad (4.10)$$

$$Q(i) = \begin{cases} 1 & , i = 1 \\ p(1 - q)^{i-2} & , otherwise \end{cases} \quad (4.11)$$



4.4 Reed-Solomon 碼回復效能

Reed-Solomon(RS)碼可有效對抗封包漏失，對於 Bose Chaudhuri Hocquenghem (BCH)只能處理二進位位元錯誤的限制，RS 碼做了延伸。以 $RS(n, k)$ 來表示，利用 k 個訊息封包產生 $n-k$ 個檢查封包，一般來說可以更正 $t = \left\lfloor \frac{n-k}{2} \right\rfloor$ 個封包漏失，但若確知錯誤發生位置時，則可更正 $(n-k)$ 個封包漏失。接下來我們簡單描述 RS 碼如何產生檢查封

包，假設有一連串共 k 個訊息封包依序分別為 $\{d_1, d_2, \dots, d_k\}$ ；稱之為傳送群組(transmission group, TG)，利用 RS 碼產生 $n-k$ 個檢查封包 $\{h_1, h_2, \dots, h_{n-k}\}$ ，其中每個封包有 m 個位元，總共 n 個封包形成一個錯誤控制區塊(FEC block)。這 k 個原始封包皆為 Galois field $GF(2^m)$ 的元素。我們定義一個多項式 $F(X) = d_1 + d_2X + \dots + d_kX^{k-1}$ ，假設 α 為 $GF(2^m)$ 先前的元素，則 $p_j = F(\alpha^{j-1})$ 且 j 屬於 $\{1, \dots, n-k\}$ 。對於 RS 解碼器而言，只要這 n 個封包 $\{d_1, d_2, \dots, d_k, h_1, h_2, \dots, h_{n-k}\}$ 中成功接收至少 k 個封包，則在傳送群組 $\{d_1, d_2, \dots, d_k\}$ 內所有漏失封包都可以被回復。反之，若訊息與檢查封包漏失的總數超過 $n-k$ ，則其漏失無法回復。當然最好的情況就是不加任何冗餘資料，因此根據紀錄網路狀態的回訊(RTP, RTCP)給傳送端可以適時的更動冗餘資料的攜帶量，以增加網路內部頻寬的使用效益。

對於一個 $RS(n, k)$ 的 FEC 區塊而言，若超過 $n-k$ 個封包漏失則 RS 解碼器將無法回復漏失的封包，而在不同漏失屬性的情形下所表現出來的回復效能也不一樣。我們先探討隨機漏失的情況[16]， p_u 為原始的封包漏失率，使用 $RS(n, k)$ 回復處理後的封包漏失率將減少為

$$PLR_{FEC} = p_u \left(1 - \sum_{j=k}^{n-1} \binom{n-1}{j} (1-p_u)^j p_u^{n-1-j} \right) \quad (4.12)$$

接下來我們探討若發生叢發性封包漏失[17]，使用 $RS(n, k)$ 回復

處理後的封包漏失率為

$$PLR_{FEC} = \frac{1}{k} \sum_{i=1}^k i \cdot P_i(n, k) \quad (4.13)$$

其中 $P_i(n, k)$ 表示經由 FEC 的回復後還漏失 i 個訊息封包的機率，這是因為訊息封包與檢查封包的漏失總數目超過 $RS(n, k)$ 所能更正的上限 $(n-k)$ 。

$$P_i(n, k) \triangleq \Pr(i \text{ 個訊息封包漏失且不能回復})$$

$$= \Pr(i \text{ 個訊息封包漏失} \cap \text{至少} \lfloor n-k-i+1 \rfloor \text{ 個檢查封包漏失}) \quad (4.14)$$

$$\text{其中 } \lfloor x \rfloor = \begin{cases} 0 & , x < 0 \\ x & , x \geq 0 \end{cases}。$$

進一步分析回復後的封包漏失率，分別考慮在第 k 個訊息封包 d_k 是正確接收或是漏失的兩種情況，即

$$PLR_{FEC} = \frac{1}{k} \left\{ \sum_{i=1}^k i \cdot P_{1,i}(n, k) + \sum_{i=1}^{k-1} (k-i) \cdot P_{0,i}(n, k) \right\} \quad (4.15)$$

其中

$$P_{1,i}(n, k) = \Pr(\{(i-1) \text{ 個訊息封包漏失}\} \cap \{Y_k = 1\}) \cdot$$

$$\Pr(\{\text{至少} \lfloor n-k-i+1 \rfloor \text{ 個檢查封包漏失}\} | \{Y_k = 1\})$$

$$= \Pr(Y_k = 1) \cdot \Pr(\text{前}(k-1) \text{ 個訊息封包內有}(i-1) \text{ 個漏失} | Y_k = 1) \cdot$$

$$\left\{ \sum_{j=\lfloor n-k+1-i \rfloor}^{n-k} \Pr(\{\text{後}(n-k) \text{ 個檢查封包內有}j \text{ 個漏失}\} | \{Y_k = 1\}) \right\}$$

$$= \pi_1 \cdot R(i, k) \cdot \sum_{j=\lfloor n-k+1-i \rfloor}^{n-k} R(j+1, n-k+1) \quad (4.16)$$

$$P_{0,i}(n, k) = \Pr(\{(k-i) \text{ 個訊息封包漏失}\} \cap \{Y_k = 0\}) \cdot$$

$$\Pr(\{\text{至少}(n-2k+i+1) \text{ 個檢查封包漏失}\} | \{Y_k = 0\})$$

$$\begin{aligned}
&= \Pr(Y_k = 0) \cdot \Pr(\text{前}(k-1)\text{個訊息封包內有}(i-1)\text{個接收} | Y_k = 0) \cdot \\
&\quad \Pr(\{\text{最多}(k-1-i)\text{個檢查封包接收}\} | \{Y_k = 0\}) \\
&= \pi_0 \cdot S(i, k) \cdot \left\{ \sum_{j=0}^{k-1-i} \Pr(\{(n-k)\text{個檢查封包內有}j\text{個接收}\} | \{Y_k = 0\}) \right\} \\
&= \pi_0 \cdot S(i, k) \cdot \left\{ \sum_{j=0}^{k-1-i} S(j+1, n-k+1) \right\} \tag{4.17}
\end{aligned}$$

最後再將(4.16)及(4.17)代入(4.15)得到

$$\begin{aligned}
PLR_{FEC} &= \frac{\pi_1}{k} \sum_{i=1}^k i \cdot R(i, k) \cdot \sum_{j=[n-k+1-i]}^{n-k} R(j+1, n-k+1) \\
&\quad + \frac{\pi_0}{k} \sum_{i=1}^{k-1} (k-i) S(i, k) \cdot \sum_{j=0}^{k-1-i} S(j+1, n-k+1) \tag{4.18}
\end{aligned}$$

圖 4.7 顯示在吉伯特模型假設下的 PLR_{FEC} 與 k 之關係，其中叢發封包漏失的平均長度 $\frac{1}{q}$ 分別為 3 與 5，原始封包漏失率 $p_u = \pi_1$ 分別為 0.1 與 0.01，區塊長度 n 為 20。觀察得知，使用 FEC 後的封包漏失率的确明顯的降低，而且在叢發封包漏失長度較短時有較好的漏失回復效能。當然 k 值愈接近 n 值時，表示檢查封包減少，則 PLR_{FEC} 愈接近原來的 $PLR = P_u$ 。

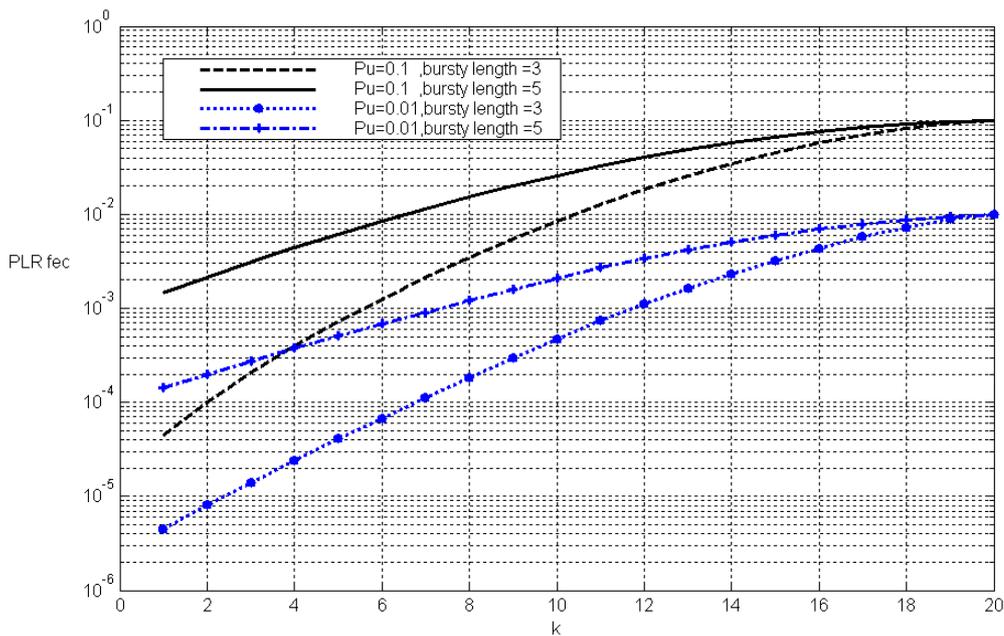


圖 4.7 k 值與 PLR_{FEC} 的關係

4.5 系統規劃與討論

上節提到使用較大的 n 值與較小的 k 值的確可以讓封包漏失率明顯的下降，同時也讓封包漏失所對應的損害值 I_e 跟著降低，問題是 (n, k) 可以一直增加使得封包漏失率降到最低嗎？在本章第二小節提及隨著 (n, k) 值的增加，所衍生的額外延遲 d_{FEC} 也會使得整體效能降低，因此 (n, k) 最佳值的設定必須同時考量封包漏失與網路延遲的交互影響。

我們先定義單一路徑網路延遲是有哪些因子所組合成的。當封包將進入網路時造成傳輸延遲，在網路傳輸也會造成傳撥延遲以及變動的佇列延遲，統稱為網路延遲 $d_{network}$ 。在接收端為了順暢音質所加上的撥放暫存器，會造成額外的暫存延遲 $d_{de-jitter_buffer}$ 。再加上編碼延遲

d_{codec} 及 FEC 延遲 d_{FEC} ，而產生口對耳延遲(mouth-to ear delay)：

$$D_{m2e} = d_{\text{codec}} + d_{\text{network}} + d_{\text{FEC}} + d_{\text{de-jitter_buffer}} \quad (4.19)$$

接下來我們設定一些條件，看看在這樣的條件下(n, k)變化所表現出來的回復效能。首先固定 $\frac{n}{k} = \frac{3}{2}$ ，這個動作在於雖然(n, k)呈線性增加，但整體傳輸位元率依舊維持固定。舉例而言，G.711、G.729 與 G.723 的位元率分別從原本 64Kb/s、8 Kb/s 與 6.3 Kb/s 擴充 $\frac{3}{2}$ 倍而為 96Kb/s、12 Kb/s 與 9.45 Kb/s。另外依據美國 AT&T 公司在網際網路延遲的實際量測而設定 d_{network} 為 35ms，並採用 Cisco 內定 $d_{\text{de-jitter_buffer}}$ 的初始值 40ms。



[實驗一]

目的：在固定 FEC 編碼增益比 $\frac{n}{k}$ 的條件下，針對語音編碼技術 G.729a 為核心的語音傳輸系統，探討其衍生延遲與封包漏失的交互影響，並分析選擇(n, k)值的最佳化問題。

架構：以語音編碼 G.729a 技術為核心，其編碼器以 $T_p = 20\text{ms}$ 音框和 5ms look-ahead 延遲來進行線性預估分析的處理，所以 $d_{\text{codec}} = 25\text{ms}$ 。另外 $d_{\text{de-jitter_buffer}} = 40\text{ms}$ ， $d_{\text{network}} = 35\text{ms}$ ， $d_{\text{FEC}} = 20(k-1)\text{ms}$ 。所以 $D_{m2e} = 100 + 20(k-1)\text{ms}$ 。至於封包漏失，則分別以隨機漏失模型(P_u)與吉伯特漏失模型(P_u, P_c)來進行以下的模擬。

步驟：完整的音質評量公式(3.2)其影響因素包括漏失、延遲大小以及語音編碼技術。首先，在 3.2.1 小節的語音編碼損害關係圖 3.2，我們可以找到 G.729a 所對應的 I_{ec} 值為 11。接著分別考慮兩種漏失模型在不同的 $FEC(n,k)$ 作用下原始封包漏失 P_u 的影響，隨機漏失模型用(4.12)式可得修正後封包漏失率 PLR_{FEC} 。而吉伯特漏失模型(P_u, P_c)代入(4.18)式可得 PLR_{FEC} 。再利用(3.4)式計算出修正後漏失機率影響音質損害的大小 I_{epi} ，最後利用(3.3)整合得到 I_e 。另外考量延遲的效應，將 $D_{m2e}=100+20(k-1)ms$ 代入(3.5)式計算延遲損害值 I_d 。最後整合 I_e 與 I_d 代入(3.2)式可獲得音質評估 R 值，並進一步利用圖 3.1 轉換成 MOS 值。



討論：

從表 4.2 與 4.3 可以看出在不管是高或低封包漏失率的情形下，使用 FEC 可有效控制封包漏失而得到較高的 R 值，而 FEC 在叢發性漏失模型下操作所得到的 R 值都會略小於隨機漏失。另外隨著 (n, k) 增加時，低漏失率幾乎完全被回復使得 $I_{epi} \approx 0$ 而 $I_e \approx I_{ec} \approx 11$ ，因此對於 R 值並沒有太大的補償效用。反倒是延遲損害的情形愈加明顯，成為導致惡化 R 值的主因。從 MOS 的角度來看，不管 (n, k) 為何值，在 $P_u = 20\%$ 的條件下所得到的音質幾乎不被接受。由圖 4.8 可看出在隨機漏失模型中，無論 P_u 為 5% 或是 20%，皆在 $(n, k) = (6, 4)$ 時

的音質評估最佳。不過可以看到在 P_u 為 5% 時，在 $k=4$ 以後下降的速度很快，這是因為之後會影響 R 值就幾乎只剩下延遲因子。由圖 4.9 可以知道吉伯特漏失模型中，無論 P_u 為 5% 或是 20%，(6, 4) 亦得到最佳的音質評估。值得注意的是，圖 4.9 在 k 超過 4 所對應的 R 值下降速度較圖 4.8 和緩。

n	k	I_d	I_e		R		MOS	
			$P_u = 5\%$	$P_u = 20\%$	$P_u = 5\%$	$P_u = 20\%$	$P_u = 5\%$	$P_u = 20\%$
3	2	2.88	12.9	32.69	78.41	58.62	3.96	3.03
6	4	3.84	11.44	27.89	78.91	62.46	3.98	3.23
9	6	7.29	11.11	24.63	75.78	62.26	3.86	3.22
12	8	12.65	11.03	22.17	70.51	59.36	3.62	3.07
15	10	18.01	11.01	20.23	65.17	55.94	3.39	2.89
w/o FEC	2.4	27.21	54.94	64.58	36.85	3.33	1.91	

表 4.2 G.729a 在隨機漏失模型下配合不同(n, k)的音質評估

n	k	I_d	I_e		R		MOS	
			$P_u = 5\%$	$P_u = 20\%$	$P_u = 5\%$	$P_u = 20\%$	$P_u = 5\%$	$P_u = 20\%$
3	2	2.88	19.08	38.64	72.24	52.67	3.70	2.72
6	4	3.84	15.26	33.62	75.10	56.74	3.83	2.93
9	6	7.29	13.21	30.23	73.69	56.67	3.76	2.93
12	8	12.65	12.16	27.60	69.38	53.94	3.57	2.78
15	10	18.01	11.62	25.43	64.57	50.75	3.33	2.61
w/o FEC	2.4	28.96	54.06	62.84	37.74	3.25	1.95	

表 4.3 G.729a 在叢發漏失模型下配合不同(n, k)的音質評估

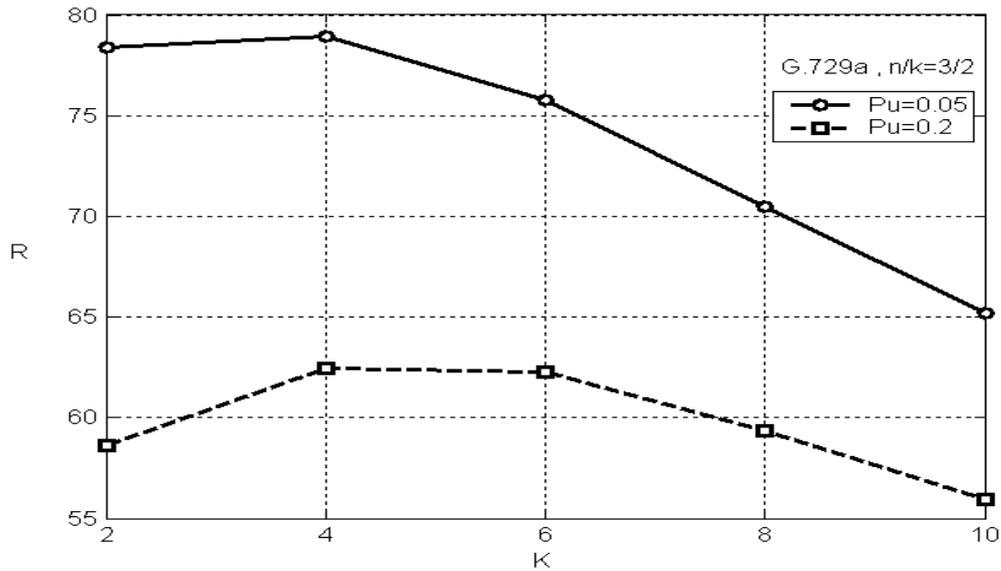


圖 4.8 G.729a 在隨機封包漏失不同(n,k)情形下與 R 值的關係

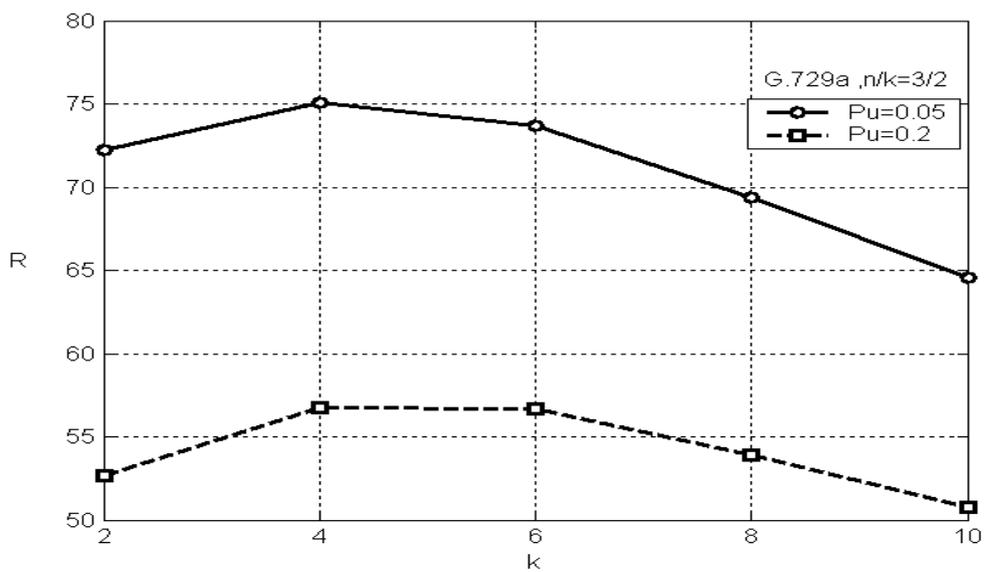


圖 4.9 G.729a 在叢發漏失模型不同(n,k)情形下與 R 值的關係

[實驗二]

目的：在固定 FEC 編碼增益比 $\frac{n}{k}$ 的條件下，針對語音編碼技術

G.711 為核心的語音傳輸系統，探討其衍生延遲與封包漏失的

交互影響，並分析選擇(n,k)值的最佳化問題。

架構：以語音編碼 G.711 技術為核心，其編碼器是以個別取樣點作處理再以 $T_p=10\text{ms}$ 做封包化，所以 $d_{codec}=10\text{ms}$ 。另外 $d_{de-jitter_buffer}=40\text{ms}$ ， $d_{network}=35\text{ms}$ ， $d_{FEC}=10(k-1)\text{ms}$ ，總和為 $D_{m2e}=85+10(k-1)\text{ms}$ ，至於封包漏失，則分別以隨機漏失模型(P_u)與吉伯特漏失模型(P_u, P_c)來進行以下的模擬。

步驟：首先，在 3.2.1 小節的語音編碼損害關係圖 3.2，我們可以找到 G.711 所對應的 I_{ec} 值為 0。接著分別考慮兩種漏失模型在不同的 FEC(n, k)作用下原始封包漏失 P_u 的影響，隨機漏失模型用(4.12)式可得修正後封包漏失率 PLR_{FEC} 。而吉伯特漏失模型(P_u, P_c)代入(4.18)式可得 PLR_{FEC} 。再利用(3.4)式計算出修正後漏失機率影響音質損害的大小的 I_{ept} ，最後利用(3.3)整合得到 I_e 。另外考量延遲的效應，將 $D_{m2e}=85+10(k-1)\text{ms}$ 代入(3.5)式計算延遲損害值 I_d 。最後整合 I_e 與 I_d 代入(3.2)式可獲得音質評估 R 值，進一步利用圖 3.1 轉換成 MOS 值。

討論：

由於 G.711 的封包設定為 10 ms，因此 I_d 的效應不那麼的顯著，甚至在(n, k)為(15, 10)所造成的延遲剛好達到人類能忍受的上限，所以對於音質的損毀不那麼顯著。因此惡化 R 值的主因落在 I_e 上。由圖 4.10 可看出在隨機漏失模型中， P_u 為 5% 時(n, k)為(6, 4)時的音質

評估最佳，在20%的情形下，(15, 10)可以表現最佳的音質。由圖 4.11 可看出在吉伯特漏失模型中，不管 P_u 為5% 或是20%，(15, 10)對應的音質評估最佳。在 P_u 為20% 且 k 由 8 增加為 10 時的 R 值突然提昇約 13 左右，這可以由圖 4.12 與圖 4.13 來說明。從圖 4.12 我們可以看出在 k 為 8 與 10 時，所得的 PLR_{FEC} 分別為 0.039 與 0.045，再從圖 4.13 這個範圍所對應的音質損害 I_{epi} ，可看到損害的落差將近 15，這就解釋了為何會有突然 R 值大幅的增加。

n	k	I_d	I_e		R		MOS	
			$P_u = 5\%$	$P_u = 20\%$	$P_u = 5\%$	$P_u = 20\%$	$P_u = 5\%$	$P_u = 20\%$
3	2	2.28	2.07	22.4	89.84	69.51	4.34	3.57
6	4	2.76	0.49	17.64	90.94	73.79	4.36	3.77
9	6	3.24	0.12	14.35	90.83	76.6	4.36	3.89
12	8	3.72	0.03	11.84	90.44	78.63	4.35	3.97
15	10	4.2	0.009	9.84	89.99	80.15	4.34	4.03
18	12	6.62	0.003	8.21	87.57	79.35	4.28	4.00
w/o FEC		2.04	16.96	43.55	75.19	48.61	3.83	2.50

表 4.4 G.711 在隨機漏失模型下配合不同(n, k)的音質評估

n	k	I_d	I_e		R		MOS	
			$P_u = 5\%$	$P_u = 20\%$	$P_u = 5\%$	$P_u = 20\%$	$P_u = 5\%$	$P_u = 20\%$
3	2	2.28	8.18	38.35	83.74	53.56	4.16	2.76
6	4	2.76	4.30	33.50	87.14	57.93	4.26	2.99
9	6	3.24	2.23	29.96	88.73	60.99	4.31	3.15
12	8	3.72	1.17	27.01	89.31	63.44	4.32	3.28
15	10	4.2	0.62	13.39	89.38	76.60	4.32	3.89
18	12	6.62	0.33	11.65	87.24	75.91	4.27	3.86
w/o FEC		2.04	28.58	51.45	63.58	40.70	3.28	2.10

表 4.5 G.711 在叢發漏失模型下配合不同(n, k)的音質評估

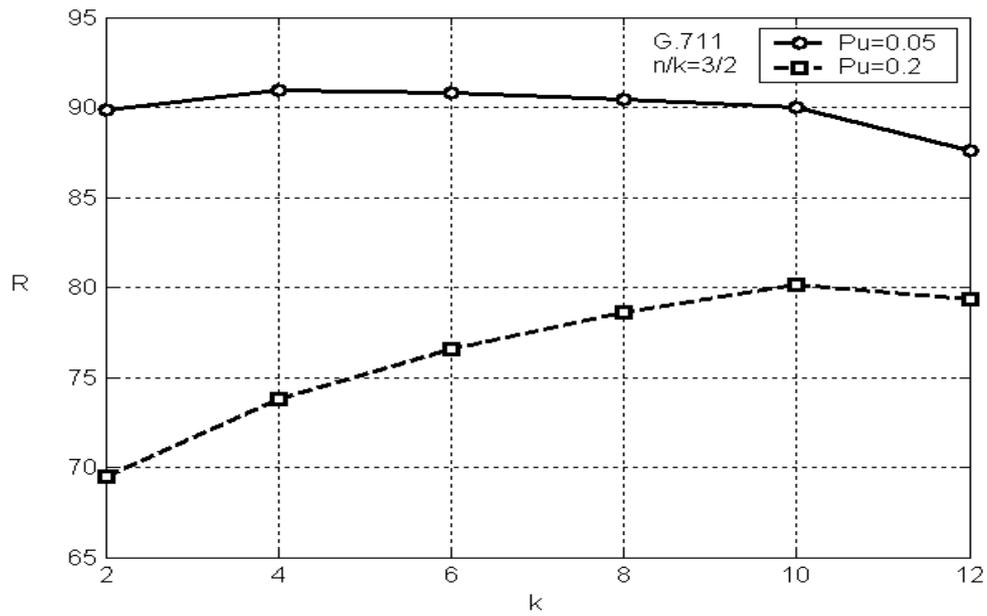


圖 4.10 G.711 在隨機封包漏失不同(n,k)與 R 值的關係

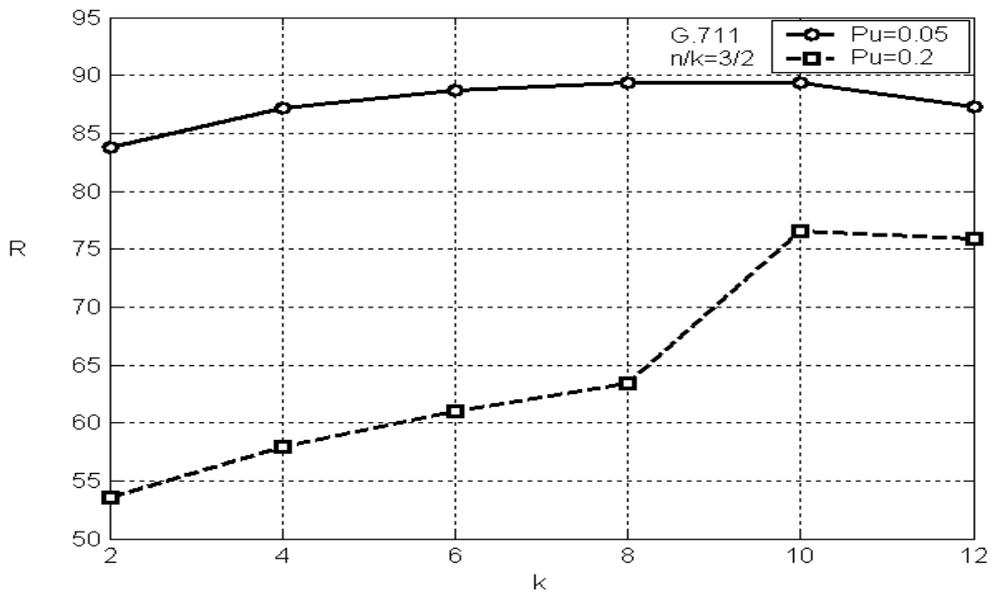


圖 4.11 G.711 在叢發漏失模型不同(n,k)與 R 值的關係

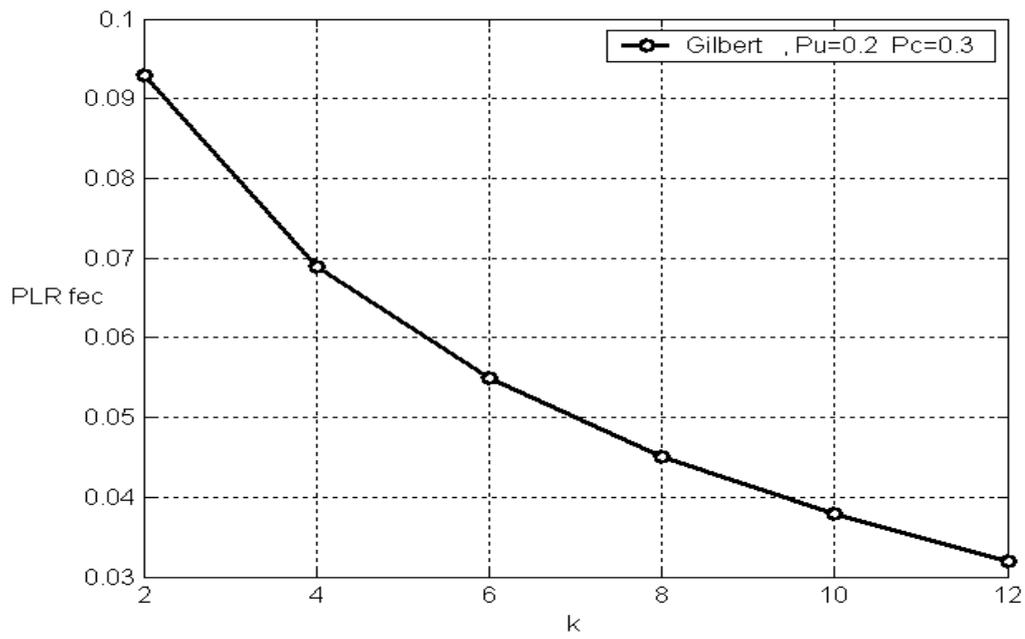


圖 4.12 吉伯特模型的 PLR_{FEC} 變化

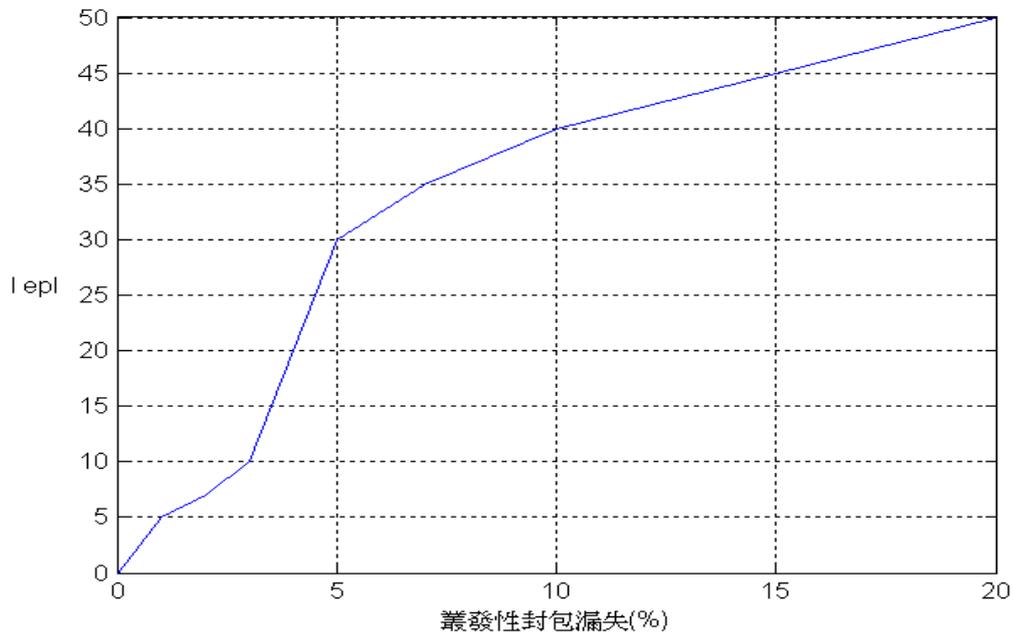


圖 4.13 叢發性封包漏失與 I_{epi} 對應關係

[實驗三]

目的：在固定 FEC 編碼增益比 $\frac{n}{k}$ 的條件下，針對語音編碼技術

G.723.1 為核心的語音傳輸系統，探討其衍生延遲與封包漏失的交互影響，並分析選擇 (n,k) 值的最佳化問題。

架構：以語音編碼 G.723.1 技術為核心，其編碼器以 $T_p=30\text{ms}$ 音框和

7.5ms look-ahead 延遲來進行線性預估分析的處理，所以 $d_{\text{codec}}=37.5\text{ms}$ 。另外 $d_{\text{de-jitter_buffer}}=40\text{ms}$ ， $d_{\text{network}}=35\text{ms}$ ， $d_{\text{FEC}}=30(k-1)\text{ms}$ 。

總合而得 $D_{m2e}=112.5+30(k-1)\text{ms}$ 。由於沒足夠的資料產生叢發性封包漏失所對應的音質損害關係，因此只單純以隨機漏失模型(P_u)來進行以下的模擬。

步驟：首先，在 3.2.1 小節的語音編碼損害關係圖 3.2，我們可以找到

G.723.1 所對應的 I_{ec} 值為 15。接著分別考慮兩種漏失模型在不

同的 $\text{FEC}(n,k)$ 作用下原始封包漏失 P_u 的影響，隨機漏失模型

用(4.12)式可得修正後封包漏失率 PLR_{FEC} 。再利用(3.4)式計算出

修正後漏失機率影響音質損害的大小的 I_{ep} ，最後利用(3.3)整

合得到 I_e 。另外考量延遲的效應，將 $D_{m2e}=112.5+30(k-1)\text{ms}$ 代

入(3.5)式計算延遲損害值 I_d 。最後整合 I_e 與 I_d 代入(3.2)式可獲

得音質評估 R 值，進一步利用圖 3.1 轉換成 MOS 值。

結論：

由表 4.6 可以看出在低漏失率情況下，當 $(n, k)=(3, 2)$ ， $(6, 4)$ ， $(9, 6)$ 時 FEC 確定可以改善 R 值，但 $(n, k)=(12, 8)$ 及 $(15, 10)$ 所對應的 R 值都低於未使用 FEC 的 R 值。換言之，即使封包漏失已完全回復($I_{epi} \approx 0$)，但因為延遲因子 I_d 的嚴重惡化影響了整體的 R 值。由圖 4.14 可看出， P_u 為 5% 時， (n, k) 為 $(3, 2)$ 時的音質最佳，而在 20% 的情形下， $(6, 4)$ 可以表現最佳的音質。從 MOS 的角度來看，不管 (n, k) 為何值，在 $P_u = 20\%$ 的條件下所得到的音質幾乎不被接受。因為 G.723.1 屬於較低的位元率編碼模式，無法容忍高封包漏失率。

n	k	I_d	I_e		R		MOS	
			$P_u = 5\%$	$P_u = 20\%$	$P_u = 5\%$	$P_u = 20\%$	$P_u = 5\%$	$P_u = 20\%$
3	2	3.42	17.08	37.40	73.70	53.38	3.77	2.75
6	4	7.63	15.49	32.64	71.08	53.93	3.65	2.78
9	6	15.67	15.13	29.36	63.40	49.17	3.27	2.53
12	8	23.71	15.03	26.85	55.45	43.64	2.86	2.25
15	10	31.75	15.01	24.84	47.44	37.61	2.44	1.95
w/o FEC		2.7	31.97	58.55	59.53	32.95	3.08	1.73

表 4.6 G.723.1 在隨機漏失模型下配合不同 (n, k) 的音質評估

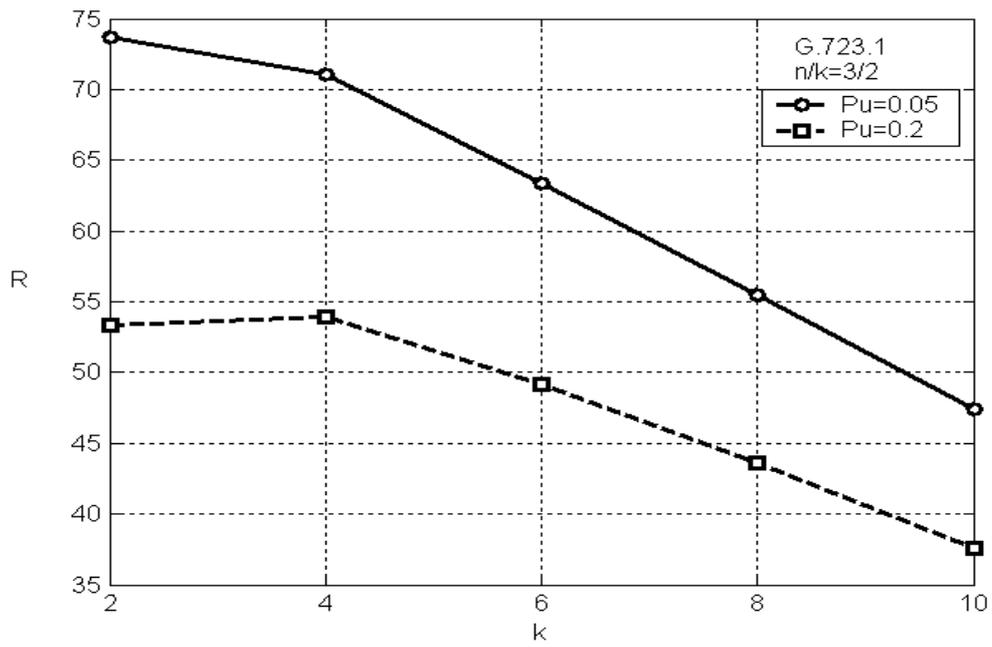


圖 4.14 G.723.1 在隨機漏失封包模型不同(n,k)與 R 值的關係



第五章 結論及未來展望

5.1 結論

針對本論文的各章節作回顧與總結。第二章著重在兩終端設備內部衍生出電子時脈誤差而影響系統的量測準確性。當沒有任何網路內部資訊可提供給終端系統業者用以規劃時，直接針對兩端點傳輸語音訊息並進行量測的工作是評估網路效能最好的依據。因此正確的量測數據將有效的輔助系統規劃者對網路狀態採取有用的控制與更動，其中封包延遲與漏失這兩項參數為其主要指標。由於量測時包含了兩終端設備的時脈，為了獲得正確的延遲就必須考量時脈的同步，防止此誤差納入網路狀態的判斷。本章實驗結果顯示，即便兩台一樣配備的電腦，內部仍存在些微的時脈誤差問題，值得注意的是在乙太網路的環境，這樣的誤差在兩終端的距離加大時，相較網路內部造成的延遲，就沒那麼具影響力。換句話說，隨著距離的縮減，就必須考慮這樣的誤差所造成的不正確延遲。

第三章在介紹 ITU-T 所制定與 E-model 相關的標準，將用以設計一音質評估模型指標。不難發現觀察封包延遲與漏失不僅能評估網路效能，更可以進一步反應人類對於聲音傳輸遭到損害的層度。將真實環境中所量測的延遲與封包漏失的數據，經由音質評估模型即可有效預測在此狀態的網路所表現出來的音質高低。

第四章則是探討如何處理封包漏失現象。藉由分析不同網路通道所造成的漏失機率模型，探討錯誤控制碼(FEC)的作用與其影響多媒體傳輸所產生的效應。在假設網路傳輸環境的實驗中，利用特定語音編碼標準設計傳輸系統。藉由第三章的音質評量模型我們可以有效預估此系統的音質高低。當網路傳輸漏失率增加或叢發性封包錯誤的情形發生時，語音傳輸的品質將令人無法忍受。因此研究中考量以 FEC 處理增加系統對於封包漏失的強健性。由實驗結果可知，在網路傳輸環境較差的情況下，設計合適的 FEC(n,k)碼，允許會造成系統效能下降之延遲效應、傳輸資料量與計算複雜度等問題，可大幅改善封包漏失，並有效提升傳輸音質。而當傳輸環境較佳，即封包錯誤發生機率很小時，以 FEC 改善網路傳輸的空間變的很有限，此時延遲效應等問題將明顯限制系統於 FEC 上的規劃。在實驗中，我們假設特定的網路模型、不同的封包漏失模型以及各類語音編碼系統搭配固定編碼增益比之條件設計 FEC(n,k)碼，用意隱含對於實際應用中系統規劃之價值，簡單的說，透過網路品質量測瞭解網路傳輸延遲、傳輸頻寬、漏失現象的分佈範圍，作為傳輸系統中語音編碼、撥放暫存器與 FEC 相關處理設計與規劃的考量。

5.2 未來展望

未來將可發展一適用於網路環境的行動裝置，提供使用者進行多

媒體傳輸服務。系統運作之流程可建立在無線的環境下提供語音的傳輸，故對於網路狀態的監控用以調整服務品質參數是必要的工作。因此會利用本論文提出的理論，架構在無線環境下量測封包延遲特性，並校正儀器所造成的誤差。另外根據統計的擾動特性，發展動態調整緩衝器長度的機制，讓系統發揮更佳的效能。



參考文獻

- [1] C. Perkins, O. Hodson, and V. Hardman, “A survey of packet loss recovery techniques for streaming audio,” *IEEE Network*, vol. 12, pp.40-48, Sept./Oct. 1998.
- [2] J.-C. Bolot, S. F. Parisis, and D. Towsley, “Adaptive FEC-based error control for Internet Telephony,” in *Inforcom'99*, Mar. 1999, pp. 1453-1460.
- [3] T.-W. Angus Lee, S.-H. Gary Chan, Q. Zhang, W.-W. Zhu, and Y.-Q. Zhang, “Optimal Allocation of Packet-Level and Byte-Level FEC in Video Multicasting over Wired and Wireless Networks,” in *Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE*, Volume: 3, 25-29 Nov. 2001, pp. 1994-1998.
- [4] S. B. Moon, J. Kurose, and D. Towsley, “Packet Audio Playout Delay Adjustment: Performance Bounds and Algorithms,” *ACM/Springer Multimedia Systems*, vol. 5, pp-17-28, Jan. 1998.
- [5] C. Boutremans and J.-Y. Le Boudec, “Adaptive Joint Playout Buffer and FEC Adjustment for Internet Telephony,” in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communication Societies*. IEEE, Volume: 1, 30 March-3 April 2003 pp. 652-662
- [6] L. Hanzo, F.C. A. Somervilie, and J.P. Woodard, *Voice Compression and Communication*. Wiley-Interscience, 2001.
- [7] “Transmission impairment due to speech processing,” ITU-T Recommendation G.113, Feb. 2001.

- [8] Cole, R. G. and Rosenbluth, J. H. "Voice over IP performance monitoring, " *ACM Computer Communication Magazine* 34, 12 (Dec.1996), pp. 9-24.
- [9] M.E. Perkins, C.A. Dvorak, B.H. Lerich, and J.A. Zebarth, "Speech Transmission Performance Planning in Hybrid IP/SCN Networks," *Communications Magazine, IEEE*, Volume: 37, Issue: 7 , July 1999, pp.126-131.
- [10] Wenyu Jiang, and Henning Schulzrinne, "Comparison and Optimization of Packet Loss Repair Method on VoIP Perceived Quality under Bursty Loss," *In Proc.ACM International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, May 2002, pp. 73-81.
- [11] J.Feigin and K.Pahlavan, "Measurement of characteristics of voice over IP in a wireless LAN environment," in *Proc. IEEE International Workshop on Mobile Multimedia Communications*, Nov. 1999, pp. 236-240.
- [12] "The E-model , a computational model for use in transmission planning," ITU-T Recommendation G.107, May. 2000.
- [13] "Method for subjective determination of transmission quality," ITU-T Recommendation P.800, Aug. 1996.
- [14] "Subjective performance assessment of telephone-band and wide-band digital codecs," ITU-T Recommendation P.830, Feb. 1996.
- [15] E.O.Elliott, "A model of the switched telephone network for data communications," *Bell Syst. Tech. J.*, vol. 44, no. 1,pp. 89-109, Jan. 1965.
- [16] Jonathan Rosenberg, Lili Qiu, and Henning Schulzrin, " Integrating Packet FEC into Adaptive Voice Playout Buffer Algorithm on

the Internet,” In *Proceedings of the Conference on Computer Communications (IEEE Infocom)* (Tel Aviv, Israel, Mar. 2000).

- [17] Pascal Frossard, “FEC Performance in Multimedia Streaming,”
IEEE COMMUNICATION LETTERS, vol. 5, NO. 3, pp. 122-124
MAR. 2001.



附錄 A 接收端的錯誤隱匿

雖然傳送端對於欲傳送的封包有一定的控管方式，來抵抗傳輸時發生的錯誤或漏失。然而有時候網路傳輸通道的不穩定性，無可避免有較高的位元錯誤率及封包遺失率，極有可能讓這些方式無法更正所有的錯誤。造成接收端在合成語音訊號時仍有斷續的現象，影響語音品質，更甚無法辨識，因此接收端必須利用一些機制來隱蔽這些錯誤，進而順暢整體音質。



錯誤隱蔽的基本原理在於產生類似原始的語音用以隱蔽漏失的部分來順暢整體音質。分為三種，嵌入(insertion)、內插(interpolation)以及重生(regeneration)。其中重生基本原理是利用音框經由壓縮處理過的特徵參數來合成漏失語音封包，在此不予討論。

A.1 嵌入

利用替代(fill-in)來彌補漏失的部分，其中最簡單的方式是僅把漏失的區間前後兩端結合起來(splicing)，這樣的技術僅適用於低漏失率以及約 4 到 16ms 的叢發性漏失，由於會干擾適應性的撥放暫存器的效能，所以並不適合作為修復。取代方法有三個，一個是在區段內置入靜音用以維持封包間的時序，適用於 2 %為上限的漏失率以及小於

4ms 的封包長度，另外一種方式則以較舒適的背景雜訊取代靜音，第三種則是複製漏失封包前的第一個成功接收的封包來替代，若發生較長的叢發性漏失，採用逐步衰減的方式來替代較後面的漏失封包，如圖 A.1 所示。由於這種方式採用很少計算量卻得到不錯的音質，因此常被使用，如 GSM 系統。

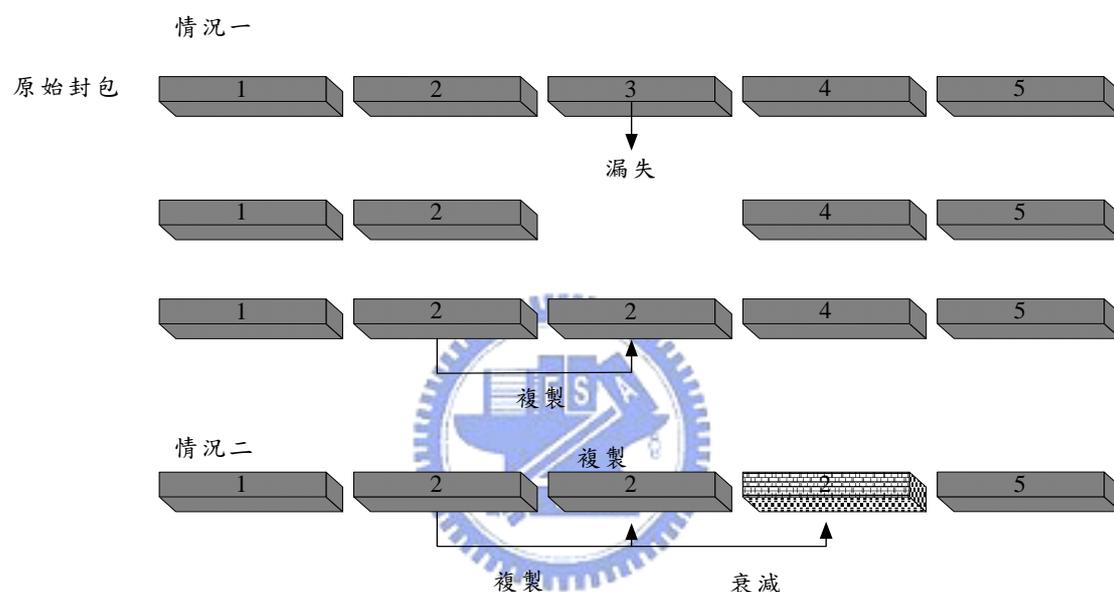


圖 A.1 複製封包代替漏失封包過程

A.2 內插

計算量普遍比嵌入的方式複雜，它卻考慮到訊號特性的改變，使得回復的訊號更具真實性。一般分為二種模式，

- 波形替代(Waveform substitution)：分為單邊與雙邊兩種計算模式，單邊則是利用檢索漏失前的封包，找出最類似漏失的部分而嵌入，雙邊的計算方式類似單邊模式。缺點是極有可能產生切割雜訊(clipping noise)。圖 A.2 描述單邊的操作模式，在漏失前的

第一個封包裡取 L 個取樣點當作樣版(template)，接著在包含 N 個取樣點的視窗裡搜尋最匹配樣版的取樣點。兩者之間存在最小正規絕對差(minimal normaled absolute difference)為匹配條件。當完成搜尋的動作後，緊接在這匹配區段後的 M 個取樣點，以樣版與匹配區段個別的平均振幅比值予以加權，最後在嵌入至漏失的區段。

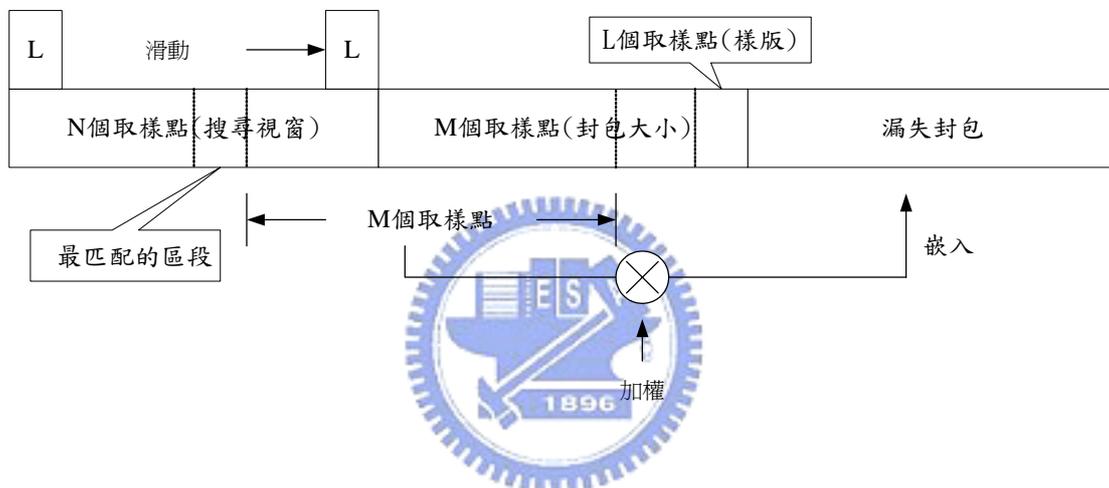


圖 A.2 單邊波形替代

- 基頻週期波形複製(Pitch Waveform Replication)：由於上述的方法並沒有考慮波形的類周期性，縱然有加權的動作依然造成切割雜訊的產生。因此這個利用聲音的類周期特性來隱匿錯誤可以有不錯的效果，後來修改成為 ITU-T 制定 G.711、G.723 與 G.729 封包錯誤隱蔽(Packet Error Concealment, PLC)的標準規格。它藉由檢測漏失區段前後波形的類周期性用以判斷漏失訊號屬於有聲或是無聲的區段，前者就計算出類週期並使用波形替代的方式掩蔽漏

失的區段；後者則直接複製的方式。

附錄 B 整合暫存撥放器與前向錯誤控制

第四章討論到 FEC 處理後的封包漏失機率統計模型，但若封包延遲過長超過撥放暫存器所允許的時間，也可能造成某種程度的損害。前人的研究在兩端點分開討論，後來卻發現兩者之間其實存在某種關聯性，因此須建立兩者共同的封包漏失機率模型[5]。首先做一些假設：

[1] 延遲程序 D_i ：第 i 個封包的網路延遲，其機率為 $F_D(d) = P[D_i \leq d]$

[2] 封包依序抵達： $a_1 \leq a_2 \leq a_3 \cdots \leq a_i \leq a_{i+1}$ ，也就是 $\Pr[D_{n+1} \leq D_n - T] = 0$

與 $\Pr[D_{n+i} \leq D_n - iT] = 0$ ，其中 T 為兩個封包之前的時間間隔

定義 t_i ：第 i 個封包的傳送時間

a_i ：第 i 個封包的抵達時間，為 $t_i + D_i$

P_i ：第 i 個封包的撥放時間，為 $t_i + D$ ，其中 D 為撥放延遲，

是網路加上暫存器造成的延遲，且在一段 talkspurt

裡對所有的封包都為固定值。

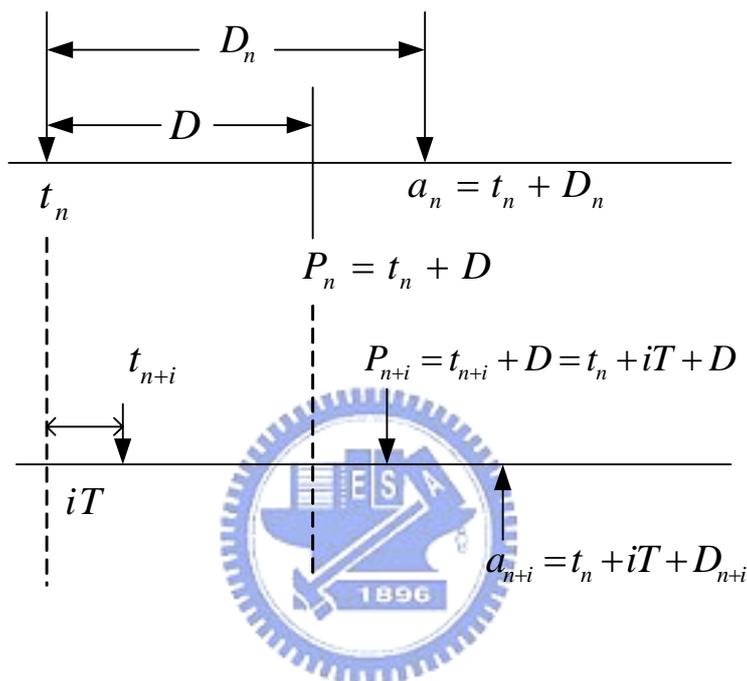
[3] 通道狀態與延遲是互相獨立的

接著我們描述一些特性：

(1) 假設第 n 個封包在抵達時超過撥放時間($P_n = t_n + D$)，則之後所有的封包同樣會超過 P_n

$$P[D_{n+i} \leq D - iT | D_n > D] = 0 \quad , \quad i \geq 0 \quad (\text{B.1})$$

[說明]:

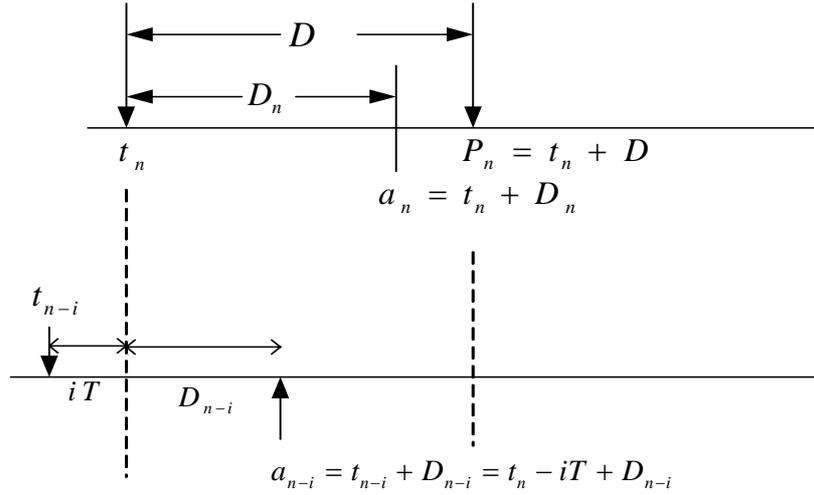


根據第 2 個假設， $a_{n+i} \geq a_n$ 即 $(iT + D_{n+i}) \geq D_n$ ，加上 $D_n > D$ 這個條件，所以 $P[D_{n+i} + iT \leq D | D_n > D] = 0$

(2) 假設第 n 個封包在撥放時間($P_n = t_n + D$)前就已抵達，則之前所有封包也會在 P_n 前抵達

$$P[D_{n-i} > D + iT | D_n \leq D] = 0 \quad , \quad i \geq 0 \quad (\text{B.2})$$

[說明]:



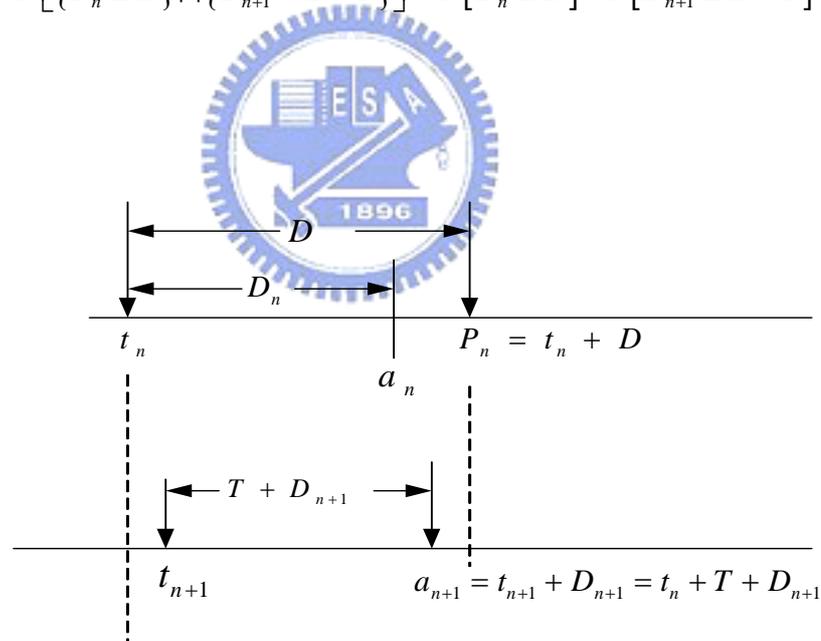
根據第 2 個假設， $a_{n-i} \leq a_n$ 即 $D_{n-i} \leq (iT + D_n)$ ，加上 $D_n \leq D$ 這

個條件，所以 $P[D_{n-i} > D + iT \mid D_n \leq D] = 0$

(3) 聯合機率 $P[\{D_n \leq D\} \cap \{D_{n+1} > D - T\}] = P[D_n \leq D] - P[D_{n+1} \leq D - T]$ (B.3)

[說明]:

1.



給予 $D_{n+1} \leq D - T$ 的條件，加上第 2 個假設得到

$$\begin{aligned} a_n &= t_n + D_n < a_{n+1} \leq t_n + D \\ \Rightarrow D_n &\leq D \\ \Rightarrow p[D_n \leq D \mid D_{n+1} \leq D - T] &= 1 \end{aligned}$$

2. $P[\{D_n \leq D\} \cap \{D_{n+1} > D - T\}]$

$$\begin{aligned}
&= P[D_n \leq D] \cdot P[D_{n+1} > D-T | D_n \leq D] \\
&= P[D_n \leq D] \cdot (1 - P[D_{n+1} \leq D-T | D_n \leq D]) \\
&= P[D_n \leq D] \cdot \left(1 - \frac{P[D_n \leq D | D_{n+1} \leq D-T] \cdot P[D_{n+1} \leq D-T]}{P[D_n \leq D]} \right) \\
&= P[D_n \leq D] \cdot \left(1 - \frac{P[D_{n+1} \leq D-T]}{P[D_n \leq D]} \right) \\
&= P[D_n \leq D] - P[D_{n+1} \leq D-T]
\end{aligned}$$

接下來計算封包漏失模型，在(4.13)強調的是 FEC 區塊的特性，但由於這小節是加上撥放時間延遲的考量，因此將針對每個封包的行為來探討。將(4.13)改寫成

$$PLR_{FEC} = \frac{1}{k} \sum_{i=1}^k \bar{P}(i) \quad (B.4)$$

其中 $\bar{P}(i)$ 表示第 i 個封包漏失機率，在接收端封包漏失的情形可分為兩種：一是封包未收到或內容錯誤而被丟棄；另一種是即使封包內容沒有錯誤然而超過撥放時間依舊被丟棄。

$$\begin{aligned}
\bar{P}(i) = & \{1 - P[(\text{第}i\text{個封包可被重建}) | (Y_i = 1)]\} \cdot P[Y_i = 1] + \\
& \{1 - P[(\text{第}i\text{個封包可被重建}) | (Y_i = 0 \cap D_i > D)]\} \cdot P[Y_i = 0 \cap D_i > D] \quad (B.5)
\end{aligned}$$

根據 RS(n, k) 的說法，超過 n-k 個封包漏失，則漏失的部份將無法被完整重建。換句話說，根據(B.2)，必須在撥放時間 $P_i (= t_i + D)$ 前成功接收至少 k 個封包，才能重建第 i 個漏失的封包。所以第 i 個封包能被重建可定義為一個 (n, k) FEC 區塊內存在 k' ($k' \geq k$) 個封包使得

$\{Y_j = 0\}$ 且 $\{D_j \leq D + (i - j)T\}$, $j \in \{1, 2, \dots, n\}$ 。藉由這樣的定義也可以得知

$P[(\text{第}i\text{個封包可被重建}) | (Y_i = 0 \cap D_i > D)] = 0$, 說明如下 :

根據(B.1) , 可以知道假如第 i ($i < k$) 個封包在撥放時間

$P_i (= t_i + D)$ 後抵達 , 則隨後的封包 j ($i < j \leq n$) 也都會超過 P_i 。

$$P[D_j \leq D - (j - i)T | D_i > D] = 0 \quad , \quad i < j \leq n$$

換句話說 , 一但發生這種情形 , 絕不可能成功接收超過 k 個

封包 , 所以漏失的封包也就不能被重建。

加上第 3 個假設 , 因此(B.5)可以化簡成

$$\bar{P}(i) = \frac{p}{p+q} [1 - P_{REC}(i)] + \frac{q}{p+q} [1 - F_D(D)] \quad (\text{B.6})$$

其中 $P_{REC}(i) = P[(\text{第}i\text{個封包可被重建}) | (Y_i = 1)]$

$$= P\left[\left\{\text{在 } (n, k) \text{ FEC 區塊裡, } \exists k' (k' \geq k) \text{ 個封包使得 } \{Y_j = 0\} \cap \{D_j \leq D + (i - j)T\}, j \in \{1, \dots, n\}\right\} | \{Y_i = 1\}\right]$$

進一步觀察可知 $P_{REC}(i | A_g) = 0$ for $g \leq k$, 因為 $Y_i = 1$ 而少於 k 個封包在

$P_i = t_i + D$ 之前抵達。所以

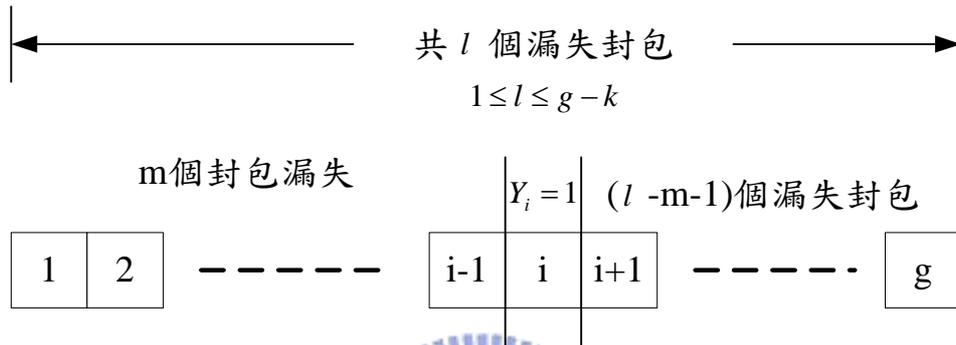
$$P_{REC}(i) = \sum_{g=1}^n P_{REC}(i | A_g) \cdot P(A_g) = \sum_{g=k+1}^{n-1} P_{REC}(i | A_g) \cdot P(A_g) + P_{REC}(i | A_n) \cdot P(A_n)$$

其中 $A_g = \{D_g \leq D - (g - i)T\} \cap \{D_{g+1} > D - (g + 1 - i)T\}$,

$$A_n = \{D_n \leq D - (n - i)T\} \quad ,$$

由(B.3) , $P(A_g) = P\{D_g \leq D - (g - i)T\} - P\{D_{g+1} \leq D - (g + 1 - i)T\}$ 。

$$\begin{aligned}
P_{REC}(i | A_g) &\triangleq P[\text{在 } g \text{ (其範圍為 } k+1 \leq g \leq n) \text{ 個封包裡接收至少 } k \text{ 個封包} | \{Y_i = 1\}] \\
&= P[\exists k' (k' \geq k) \text{ 個封包在 } \{1, 2, \dots, g\} \text{ 中, 使得 } Y_j = 0, j \in \{1, 2, \dots, g\} | Y_i = 1] \\
&= P[\{\text{在 } g \text{ 個封包裡的漏失封包數目 } l \text{ 介於 } (1, g-k)\} | \{Y_i = 1\}] \\
&= P_{PAR}(k, g, i)
\end{aligned}$$



其中 $0 \leq m \leq i-1$, $(l-m-1) \geq 0 \Rightarrow l-1 \geq m$, 則 $\min(l-1, i-1) \geq m \geq 0$

$$P_{PAR}(k, g, i) = \sum_{l=1}^{g-k} \sum_{m=0}^{\min(l-1, i-1)} R(m+1, i) \cdot R(l-m, g-i+1) \quad (\text{B.7})$$

其中 $R(m+1, i)$ 及 $R(l-m, g-i+1)$ 可由式(4.4)疊代計算。