

國立交通大學

電信工程學系

碩士論文

整合前向錯誤控制於

多重敘述語音播放排程之研究

Adaptive Joint Playout Buffer And FEC Adjustment  
For Multi-Stream Voice Over IP Networks

研究生：張永樂

指導教授：張文輝博士

中華民國九十八年六月

整合前向錯誤控制於  
多重敘述語音播放排程之研究  
Adaptive Joint Playout Buffer And FEC Adjustment  
For Multi-Stream Voice Over IP Networks

研究生：張永樂  
指導教授：張文輝

Student：Yung-Le Chang  
Advisor：Wen-Whei Chang

國立交通大學  
電信工程學系  
碩士論文



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

June 2009  
Hsinchu, Tainwan, Republic of China

中華民國 九十八 年 六 月

# 整合前向錯誤控制於 多重敘述語音播放排程之研究

學生：張永樂

指導教授：張文輝 博士

國立交通大學電信工程學系碩士班



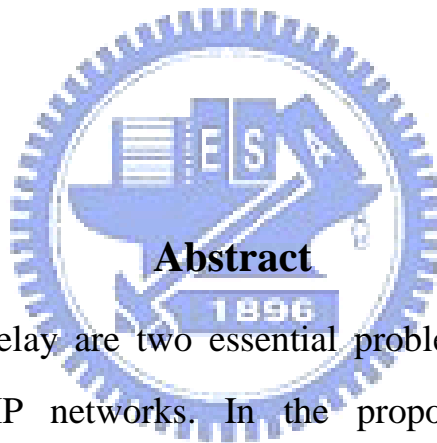
本論文所提出的多重敘述語音傳輸系統，是將音框編碼輸出的參數以兩個封包分別傳送，以期利用互相獨立的傳輸路徑提升其音框播放品質。為了準確反應多重敘述傳輸的音框回復品質，我們根據E模型提出一新的音質預測模型，其主要概念是區隔單一或雙重封包的接收狀態並加以適度整合。進一步以此音質預測模型作為多重敘述傳輸系統的發展平台，整合設計其播放排程與前向錯誤控制，以期依網路環境的時變特性適應性地調整其系統參數。系統模擬証實，新的音質預測模型更為接近主觀聽覺測試結果，我們提出的多重敘述傳輸系統能有效對抗封包漏失並輸出最佳音質。

# **Adaptive Joint Playout Buffer And FEC Adjustment For Multi-Stream Voice Over IP Networks**

Student: Yung-Le Chang      Advisor: Dr. Wen-Whei Chang

Department of Communications Engineering

National Chiao Tung University



## **Abstract**

Packet loss and delay are two essential problems to real-time voice transmission over IP networks. In the proposed system, multiple descriptions of the speech are transmitted to take advantage of largely uncorrelated delay and loss characteristics on different network paths. Adaptive joint playout buffer and FEC adjustment of multiple voice streams is formulated as an optimization problem leading to a better delay-loss tradeoff. The basic strategy is a perceptually motivated optimization criterion based on a modified ITU-T E-model for multiple-stream transmission . Experimental results show that the proposed multi-stream voice transmission system improves the delay-loss tradeoff as well as speech reconstruction quality.

## 誌謝

兩年的研究生涯當中，首先要感謝指導教授張文輝老師的細心指導，在研究與課業上引領我正確的研究態度及學習方向，並使我體認到做研究時所需具備的嚴謹，除此之外，當我在研究上遇到挫折及瓶頸時，老師也總是耐心地在旁協助並給予鼓勵支持，幫助我順利完成碩士論文。也要感謝實驗室朋友們在學業上與研究上的諸多協助，尤其是吳俊鋒學長不厭其煩地與我討論並提供研究上的建議。另外也要特別感謝室友張致遠同學在這段期間給予我精神上的鼓勵。最後要感謝母親及家人的支持，讓我能無後顧之憂地完成碩士班學業。



# 目錄

中文摘要 .....	i
英文摘要 .....	ii
誌謝 .....	iii
目錄 .....	iv
圖目錄 .....	vii
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.....	4
第二章 多重敘述的音質預測模型.....	5
2.1 通話品質預測模型 .....	5
2.1.1 主觀聽覺測試.....	6
2.1.2 音質評量指標.....	7
2.2 多重敘述語音編碼系統.....	13
2.2.1 MD-AMR編碼器 .....	13
2.2.2 MD-AMR解碼器 .....	15



2.2.3 MD-G. 729a編碼器 .....	<a href="#">18</a>
2.2.4 MD-G. 729a解碼器 .....	<a href="#">20</a>
2.3 多重敘述編碼的 $I_c$ 音質預測模型.....	<a href="#">23</a>
2.4 音質損害預測模型的整合.....	<a href="#">30</a>
<b>第三章 聽覺最佳化的適應性播放排程設計 .....</b>	<b><a href="#">32</a></b>
3.1 播放緩衝器.....	<a href="#">33</a>
3.2 晚到漏失機率模型 .....	<a href="#">37</a>
3.3 MD傳輸系統的播放排程設計.....	<a href="#">40</a>
3.3.1 以低延遲為考量的設計.....	<a href="#">40</a>
3.3.2 音質最佳化的設計.....	<a href="#">42</a>
<b>第四章 結合前向錯誤控制的MD傳輸系統 .....</b>	<b><a href="#">47</a></b>
4.1 封包漏失回復機制 .....	<a href="#">47</a>
4.2 前向錯誤控制 .....	<a href="#">49</a>
4.3 結合FEC的MD傳輸系統 .....	<a href="#">55</a>
4.3.1 Reed Solomon(RS)碼 .....	<a href="#">55</a>
4.3.2 叢發性封包漏失的回復機率推導 .....	<a href="#">57</a>
4.4 整合漏失回復後的MD音質損害模型 .....	<a href="#">60</a>
4.5 MD封包傳輸系統的最佳化設計.....	<a href="#">68</a>
<b>第五章 實驗結果 .....</b>	<b><a href="#">70</a></b>

5.1 $I_c$ 音質預測模型的驗證 .....	<a href="#">70</a>
實驗一:驗證新的 $I_c$ 模型.....	<a href="#">70</a>
實驗二:語音檔經MD和SD傳輸之下的音質比較 .....	<a href="#">73</a>
5.2 MD播放排程演算法.....	<a href="#">75</a>
實驗一:固定和動態設計的安全因子效能比較 .....	<a href="#">76</a>
實驗二:以不同考量(低延遲或音質)設計的MD播放排程比較 .....	<a href="#">78</a>
實驗三:相同編碼效率下,MD與SD傳輸的比較 .....	<a href="#">80</a>
實驗四:結合FEC的MD播放排程 .....	<a href="#">82</a>
實驗五:動態調整FEC與固定FEC的MD播放排程比較.....	<a href="#">84</a>
實驗六:動態調整FEC於MD和SD傳輸的比較 .....	<a href="#">90</a>
<b>第六章 結論與未來展望.....</b>	<b><a href="#">93</a></b>
<b>參考文獻 .....</b>	<b><a href="#">95</a></b>



# 圖目錄

圖 2.1 R與MOS的轉換關係 .....	<a href="#">9</a>
圖 2.2 MD解碼 .....	<a href="#">24</a>
圖 2.3 封包經MD解碼後分屬兩種不同的語音編碼輸出 .....	<a href="#">24</a>
圖 2.4 尋找新 $I_0$ 模型的實驗流程 .....	<a href="#">28</a>
圖 2.5 MD-G.729A編碼的音質損害因子 .....	<a href="#">29</a>
圖 2.6 MD-AMR編碼的音質損害因子 .....	<a href="#">29</a>
圖 3.1 播放緩衝器的影響 .....	<a href="#">34</a>
圖 3.2 平均延遲為 240MSEC 的累積分佈函數 .....	<a href="#">39</a>
圖 3.3 平均延遲為 55MSEC 的累積分佈函數圖形 .....	<a href="#">39</a>
圖 3.4 MD傳輸系統 .....	<a href="#">46</a>
圖 4.1 MEDIA SPECIFIC FEC .....	<a href="#">52</a>
圖 4.2 冗餘資訊以多個封包產生 .....	<a href="#">52</a>
圖 4.3 RS編碼過程 .....	<a href="#">53</a>
圖 4.4 FEC機制下的封包傳送與接收 .....	<a href="#">54</a>
圖 4.5 同位碼回復錯誤的過程 .....	<a href="#">54</a>
圖 4.6 吉伯特模型 .....	<a href="#">57</a>
圖 4.7 FEC回復情形 .....	<a href="#">62</a>
圖 4.8 結合FEC的MD傳輸系統 .....	<a href="#">69</a>
圖 5.1 固定與動態調整 $\beta$ 效能比較 .....	<a href="#">76</a>
圖 5.2 MD播放排程演算法效能比較 .....	<a href="#">78</a>
圖 5.3 MD與SD加上固定FEC RS(9,8)效能比較 .....	<a href="#">80</a>
圖 5.4 結合FEC的MD傳輸優勢( $P+Q=1$ ; TRACE1&2) .....	<a href="#">82</a>
圖 5.5 結合FEC的MD傳輸( $P+Q=0.8$ ; TRACE1&2) .....	<a href="#">84</a>
圖 5.6 結合FEC的MD傳輸( $P+Q=0.8$ ; TRACE3&4) .....	<a href="#">84</a>
圖 5.7 結合FEC的MD傳輸( $P+Q=0.3$ ; TRACE1&2) .....	<a href="#">85</a>
圖 5.8 結合FEC的MD傳輸( $P+Q=0.3$ ; TRACE3&4) .....	<a href="#">85</a>
圖 5.9 結合FEC的SD傳輸系統 .....	<a href="#">91</a>
圖 5.10 結合FEC的MD與SD傳輸效能比較( $P+Q=0.9$ ) .....	<a href="#">92</a>

# 表目錄

表 2.1 R與MOS的對應關係.....	9
表 2.2 AMR-WB編碼輸出及位元對照表.....	16
表 2.3 MD-AMR編碼輸出及位元對照表.....	17
表 2.4 G. 729A編碼輸出及位元對照表.....	21
表 2.5 MD-G. 729A編碼輸出及位元對照表.....	22
表 2.6 $I_e$ 模型參數.....	27
表 2.7 語音檔內容 .....	27
表 3.1 網路延遲的累積分佈函數 .....	37
表 4.1 ARQ VS. FEC.....	49
表 5.1 實驗語音檔內容 .....	70
表 5.2 音質預測模型的驗證 .....	71
表 5.3 音框經由MD與SD傳輸效能比較 .....	73
表 5.4 網路延遲(單位:MS).....	75
表 5.5 收到兩條串流比例( $\text{PROB}\{\Omega_1\}$ )的比較(固定與動態調整 $\beta$ ).....	76
表 5.6 整體延遲的比較(固定與動態調整 $\beta$ ).....	77
表 5.7 收到兩條串流比例( $\text{PROB}\{\Omega_1\}$ )的比較(低延遲或音質考量).....	78
表 5.8 整體延遲的比較(低延遲或音質考量).....	79
表 5.9 對照圖(5.4)，接收端音質相關數據的比較.....	82
表 5.10 對照圖(5.7)，整體延遲(MS)與回復後FER的比較.....	86

# 第一章 緒論

## 1.1 前言

近年來由於網際網路的盛行以及日益增加的傳輸頻寬，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設計考量，其中影音及資料的整合是未來通訊傳輸的主要潮流，而網路電話(Voice over IP, VoIP)即為其衍生的應用服務。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。針對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，更成為業界及學術界研究發展的重要課題，其主要目標是提昇語音品質且達到即時(real-time)傳送的效果。而將語音結合於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語音通訊品質的損害要素。為了補償延遲擾動，一具體可行方案是在接收端的應用層(application layer)中加入一播放緩衝器(playout buffer)，彈性調整每個語音封包的播放時間(playout time)。雖然這種方式會增加封包的整體延遲，但也相對降低了晚到封包漏失的機率。因此，在語音封包的緩衝延遲(buffering delay)與晚到漏失率(late loss rate)

之間存在一個最佳化權衡的問題，此即為語音封包播放排程研究的重要課題。若排定一個較晚的播放時間，將提高封包播放的機率而降低封包漏失率，但也相對衍生較高的緩衝延遲[1][2]。而為了抵抗封包漏失，主要的方法是在傳送端引入前向錯誤控制(Forward Error Control, FEC)[3][4][5]，而其原理是在傳輸原始封包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。問題是由於接收端必須收到原始及額外資訊，才能透過 FEC 解碼機制回復可能漏失的封包，所以不可避免地將為整個傳輸系統帶來額外的延遲。因此 FEC 雖然可以有效降低封包漏失率，但卻也無形中帶來更多的延遲損害。另外，一旦封包發生叢發性網路漏失，將很有可能接收端同時無法正確接收原始及額外資訊，使得 FEC 無法發揮其封包回復的能力。有鑑於此，近年來有學者提出多重敘述編碼技術(Multiple Descriptions Coding, MDC)[6][7]，其主要概念是將音框所屬的編碼參數分成兩個封包再傳送於互相獨立的傳輸路徑，透過接收到的單一封包來補償另一漏失封包的部分資訊，因此可以在不需增加整體延遲的情況下，有效地提昇其音框播放品質。

## 1.2 研究動機

在前人研究[6]，多重敘述播放排程設計是將整體延遲、單一及雙重封包漏失率透過特定參數加權整合成一成本函數，再依此函數隨

著網路環境變化適應性地調整其系統參數。然而其做法並未將延遲及封包漏失所造成的音質損害因子納入考量，而且也沒有考慮到不同封包接收情況所對應的音框回復音質差異。針對此問題，我們參考[7]的 MD 語音編碼方式，應用 G. 729a[8]和 AMR-WB[9]兩種編碼標準，基於聽覺最佳化的考量，將延遲及封包漏失音質損害因子納入成本函數來設計多重敘述傳輸系統。但這又衍伸出一項問題，由於 ITU-T 音質預測模型[10]原是針對單一敘述傳輸系統而設計，並無法精準預測多重敘述傳輸下的音框重建品質，所以為了設計出更有效率的 MD 傳輸系統，就必須另尋能精準預測音質的模型。因此，我們先分別探討單一及雙重封包接收狀態下的音框輸出音質，並建立可反應此兩種封包接收狀態的音質預測模型。透過此兩種接收狀態的相對比例對其加權整合，進而提出一個適用於 MD 傳輸系統的封包漏失及語音編碼音質損害的預測模型，接著再以此音質預測模型為發展平台來設計多重敘述傳輸系統。而為了能更有效地抵抗封包漏失損害，我們引入前向錯誤控制於 MD 傳輸系統中，並針對叢發性漏失的網路環境，推導封包於 MD 傳輸架構下的 FEC 回復機率及其回復後的音質損害。再基於推導結果，適應性地整合設計 MD 傳輸系統中播放排程與 FEC 的調整機制。其主要想法是結合 FEC 及 MD 的封包回復能力，透過 MD 的編碼方式可減少 FEC 所需的保護位元總量，並有效降低結合 FEC 所衍生的額

外延遲。至於 FEC 的適應性調整機制，同時考量當前的網路環境(封包漏失、網路延遲)，並在權衡整體延遲(包含 FEC 的額外延遲)、單一及封包接收狀態及封包雙重漏失之間的得失之後，選擇出最佳的系統傳輸參數，因此我們所提出結合 FEC 的 MD 傳輸系統，具備動態反應網路傳輸環境變化的音框回復能力。

### 1.3 章節概要

第二章介紹多重敘述語音傳輸系統並建立其對應的音質預測模型，第三章介紹在多重敘述音質評量平台上，語音封包播放排程的最佳化設計，第四章介紹基於聽覺最佳化考量，整合前向錯誤控制於多重敘述語音傳輸系統，第五章的系統模擬部分，先透過對語音檔的實作來驗證新的音質預測模型，接著利用網路模擬器(Network Simulator 2, NS2)[11]模擬網路傳輸環境，進一步驗證本論文所提出的強健性 MD 語音傳輸系統。

## 第二章

# 多重敘述的音質預測模型

網路即時通話系統的音質損害，主要是受到封包漏失及網路延遲的雙重影響。為了降低封包漏失所造成的音質損害，除了前向錯誤控制(FEC: Forward Error Control)以外，近年來也有學者提出多重敘述語音編碼(Multiple Description Speech Codec, MD-Speech Codec)模式。主要是結合語音編碼標準的前置處理[7]，在傳輸端將語音音框(frame)所屬的參數分成兩條串流，個別以封包的形式經由互相獨立的通道傳輸，當其中一條串流發生漏失時，其封包資訊藉由收到的另一條串流來部分補償，以期能夠降低封包漏失所造成的音質損害。本章節應用 G. 729a 和 AMR-WB 兩種語音編碼標準來實現其多重敘述編碼模式，並深入探討其傳輸特性，以建立適用於 MD 傳輸系統的音質預測模型。

### 2.1 通話品質預測模型

近年來由於網路電話(VoIP)低廉的通話費用以及更有效率的網路運用等種種優點，消費者利用網際網路當作影音多媒體的傳輸媒介之接受度逐年增加。然而消費者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)，因此在使用網路電話之際勢必也會對通話品質做一定程度的要求。對於網路系統規劃者而言，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。所以我們必須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路通話的音質評量指標模型。

### 2.1.1 主觀聽覺測試

對於通話品質的界定，最直接的方式是以人耳的主觀聽覺來判斷音質好壞，並透過某種制定的量值用以區分其程度差異。在ITU標準規格[12][13]中，制定了平均評比分數(Mean Opinion Score，MOS)，評分的等級從感覺音質極佳的5分到音質極差的1分。

所謂的主觀聽覺測試，測試者是經由特定條件挑選出來，並處在特別設計過的房間，房間裡的噪音以及其他重要的環境因素都被控制在某一種適合測試的程度來進行聽覺實驗。待測試用的語料庫會預先錄音，基於測試準確度的考量，每一段的語句大約會維持2到3秒，而且這些語句彼此之間沒有明顯的關聯性。經過語音編碼處理後再改變網路模擬系統的參數因子，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、不同語音碼碼方式的相容性等傳輸因子。所有測試者去聆聽播放出來的聲音，並針對不同的系統設定環境予以評分，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值4.0分~4.5分為高品質，達到長途電話網的音質要求。MOS值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值3.0分以下通常稱為合成音質，這種語音一般只有達到足夠聽的懂的程度，但是缺乏自然度，且不容易識別語者特徵。

由於所有的測試都是憑藉人耳的主觀聽覺來評分，往往會因為評分者對於當時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細部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並改善。

### 2.1.2 音質評量指標

主觀聽覺測試無法反應傳送與接收兩端之間經過網路傳輸所造成的音質損害，因此國際電信聯盟ITU制定一個具體的音質評量模型，簡稱E模型(E-model, ITU-T G.107)。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，可

提供系統規劃及調整系統關鍵元件參數之用。E模型的方程式表示如下

$$R = R_0 - I_s - I_d - I_e + A \quad (2.1)$$

其中

$R_0$ : 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。

$I_d$ : 語音延遲(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最低限度為50。同時在E模型中，也定義了R值與平均評比分數(MOS)之間對應的關係，可以避免主觀評量過程中繁瑣的人工測試過程。如圖(2.1)所示，其關係式為

$$MOS = \begin{cases} 1, R < 0 \\ 1 + 0.035R + 7 \times 10^{-6} R(R - 60)(100 - R), 0 < R < 100 \\ 4.5, R > 100 \end{cases} \quad (2.2)$$

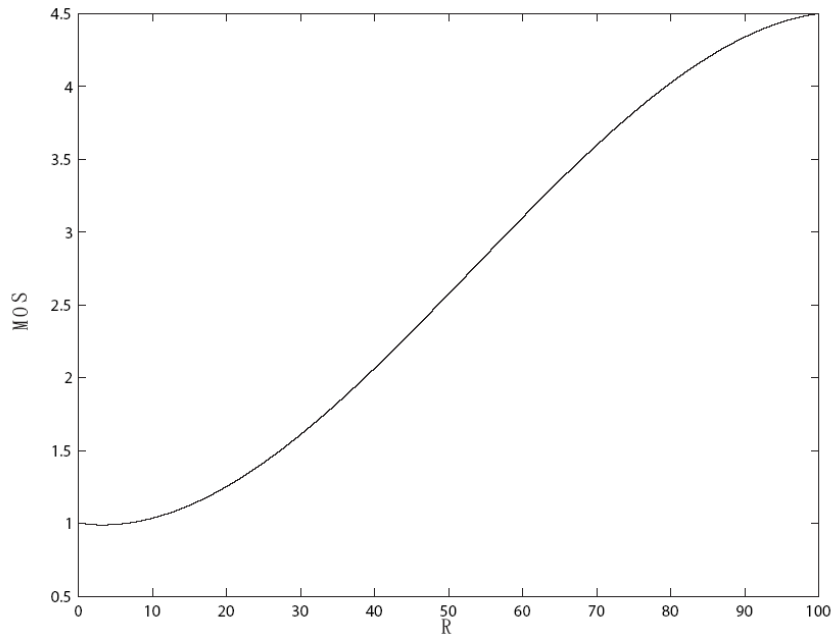


圖 2.1: R 與 MOS 的轉換關係

我們將R與MOS的對應關係整理成表2.1:

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 2.1 R 與 MOS 對應關係

由於我們僅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，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[14]。如式(2.3)所示

$$R(d, r, e) = 94.2 - I_d(d) - I_e(r, e) \quad (2.3)$$

其中d為單一路徑口對耳延遲(mouth-to-ear delay)，r是編碼位元率，e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

$$I_e(r, e) = I_{e, codec}(r) + I_{e, pl}(e) \quad (2.4)$$

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

#### [1] 語音編碼損害因子- $I_{e, codec}$

語音壓縮處理可以減少資料傳輸量而有效節省頻寬。其中編碼標準有多項選擇，如 G. 711 PCM、G. 729a CS-ACELP、G. 723. 1 MPC-MLQ、AMR-WB，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。每一種編碼標準均有其特定的聆聽 MOS(listening MOS)，利用圖(2.1)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(2.3)可簡化為

$$R(r) = 94.2 - I_{e, codec}(r) \quad (2.5)$$

由前人研究得知，隨著編碼位元率的下降，音質的損害值明顯增加。這是由於較高的壓縮率雖然能節省頻寬的使用，然而封包與封包

之間的關聯性卻相對的降低。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，因此有必要在封包傳送前對封包做保護的動作。

## [2] 封包漏失損害因子 - $I_{e,pl}$

在前一個小節提到，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傳輸影響的音質損害。在網路中常常面臨通道頻寬有限卻需要傳送大量語音封包或資料封包，路由器(router)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。若是資料封包的漏失，可以使用要求重送(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，使得整段語音經過網路後會發生斷斷續續的現象。根據研究指出，語音編碼與封包漏失的共同損害因子  $I_e$  可近似為：

$$I_e(r,e) = I_{e,codec}(r) + I_{e,pl}(e) = \gamma_1 + \gamma_2 \ln(1 + \gamma_3 e) \quad (2.6)$$

其中  $I_{e,codec}(r) = \gamma_1$ ， $I_{e,pl}(e) = \gamma_2 \ln(1 + \gamma_3 e)$ 。

## [3] 延遲損害因子 - $I_d$

就單向聆聽 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, MOSc)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，例如編碼與封裝處理造成的延遲、傳送路徑延遲、播放暫存器造成的延遲。

在前人研究[14]中，參考 E 模型(ITU G.107)比對單一路徑口對耳的延遲與其損害因子，利用片段線性分析可推導得

$$I_d = 0.024d + 0.11(d - 177.3)H(d - 177.3) \quad (2.7)$$

其中  $d$  為單一路徑延遲，而  $H$  是一個步階函數。



## 2.2 多重敘述語音編碼系統

### 2.2.1 MD-AMR 編碼器

前人所提出的 MD-AMR 編碼模式[7]，是依照 AMR-WB 編碼音框的參數特性來均分成兩條串流，其編碼處理詳如下述：

a) 導納頻譜頻率 (Impedance Spectrum Frequency, ISF):

AMR-WB 標準是將每一個音框分析所得到的線性預測係數 (Linear Prediction Coefficient, LPC)，轉換成量化失真靈敏度較低的 16 個 ISF 係數之後，再進行分次多階向量量化 (split-multistage vector quantization, S-MSVQ)，以因應無線電信網路的低位元率傳輸需求。比較特別的是，在第二個編碼模式裡，ISF 係數是以兩階分次量化器來量量化。在 MD 編碼端的兩條串流，都將傳送在第一個階段中碼向量索引 (index) 所屬的 16 個位元，這是由於假如沒有這部分的資訊，LPC 參數將完全無法重建回來。在第二個階段，向量將被分成 5 個次向量 (subvector)，而且這 5 個次向量將以  $6+7+7+5+5=30$  個位元來編碼。兩條串流都將傳送第一個次向量 (6 位元)，而第二個 (7 位元) 和第五個 (5 位元) 次向量由第一條串流傳送，其餘的第三個 (7 位元) 和第四個 (5 位元) 次向量則由第二條串流傳送。

b) 適應性碼簿的音高延遲 (Pitch delay for adaptive codebook):

在 AMR-WB 中，音高延遲是以次音框 (subframe) 來計算，但是第

二個(第四個)次音框的音高延遲是依據第一個(第三個)次音框的音高延遲進行編碼。因此，在分配串流傳送方式時，由第一條串流來傳送第一個和第二個次音框裡的位元，而其餘的第三個和第四個次音框位元則由第二條串流來傳送。

c)適應性及固定性碼簿增益(Adaptive and fixed codebook gains):

在 AMR-WB 語音編碼標準裡，適應性和固定性碼簿的增益(Gain)在每一個次音框裡是共同以 7 位元來量化。在多重敘述傳輸的部分，是將第一個和第三個次音框中代表增益的位元放在第一條串流，至於第二個和第四個次音框裡代表增益的位元則於第二條串流來傳輸。這樣的分配是為了當只成功接收到一條串流時，可以利用前一個次音框的資訊來補償漏失掉的部分。也就是說假如只接受到第一條串流，則於第二條串流傳輸的第二個次音框可以以第一個次音框來補償，第四個次音框內容可以以第三個次音框來補償。

d)固定性碼簿索引值(Fixed codebook indices):

每個次音框中，固定性碼簿向量是以 36 位元量化器來量化，關於四個次音框中固定性碼簿索引值的傳送分配是由第一條串流來傳送第一個和第三個次音框中的內容，第二條串流來傳送第二個和第四個次音框中的內容。

AMR-WB 編碼所得的四組語音參數依上述方式分配後，兩條串流額



外傳送音態偵測旗標(Voice Active Detection, VAD flag)。為了清楚解釋 MD-AMR 的編碼規則，我們以表(2.2)及(2.3)來呈現 AMR-WB 和 MD-AMR 的傳送參數及其對應的位元位置。經統計得知，MD-AMR 編碼後的每條串流位元速率為 6.9Kbps，合計為 13.8 Kbps，這比 AMR-WB 的 12.65Kbps(mode 2)多出了 1.15 Kbps 的冗息(redundancy)。這額外多出資訊的好處是，當我們只接收到其中單一串流時，仍能回復出令人接受的音質。值得注意的是，雖然 MD-AMR 單一串流位元率 6.9 Kbps 高於 AMR-WB mode 1 的 6.6kbps，但是因為後者是針對 6.6kbps 去做品質最佳化的編碼設計，所以當我們只收到一條串流時，經過 MD-AMR 回復後的音質將略遜於 AMR-WB 6.6 kbps。



### 2.2.2 MD-AMR 解碼器

當兩條串流都成功接收時，在接收端可以完全回復成 AMR-WB 編碼音框，再以 AMR-WB 的解碼器將語音完整呈現。若是兩條串流都漏失，則 AMR-WB 解碼器將啟動其漏失補償機制(loss- concealment)，盡可能的回覆漏失的音框資訊。

最後一種情況是當只收到其中一條串流時，其解碼過程如下(在此假設串流 2 漏失，串流 1 漏失的另一種情況可依此類推):

(1)串流 2 中，對應於導納頻譜對係數(ISP)索引值的第三個和第四個

次音框中的音高延遲(pitch-lag)值，將由串流 1 中第二個次音框的音高延遲值來取代。

(2) 第三個和第四個次音框中長期預測濾波參數(Long-Term Prediction-filtering flag, LTP-filtering flag)，以串流 1 中第二個次音框的對應值來取代。

(3) 第二個(第四個)音框的固定性碼簿向量由第一個(第三個)次音框來補償，至於增益(gain)部分，由衰減 3dB 的第二個(第四個)次音框來補償第一個(第三個)次音框中的內容。

AMR-WB codec output parameter (12.65Kbps) and corresponding bits	
Bit position	Content
1	VAD-flag
2~9	Index of 1st ISP vector
10~17	Index of 2nd ISP vector
18~23	Index of 3rd ISP vector
24~30	Index of 4th ISP vector
31~37	Index of 5th ISP vector
38~42	Index of 6th ISP vector
43~47	Index of 7th ISP vector
Subframe 1	
48~56	Adaptive codebook index
57	LTP-filtering-flag
58~66	Codebook index for track1
67~75	Codebook index for track2
76~84	Codebook index for track3
85~93	Codebook index for track4
94~100	VQ gain
Subframe 2	
101~106	Adaptive codebook index(relative)
107	LTP-filtering-flag

108~116	Codebook index for track1
117~125	Codebook index for track2
126~134	Codebook index for track3
135~143	Codebook index for track4
144~150	VQ gain
Subframe 3	
151~159	Adaptive codebook index
160	LTP-filtering-flag
161~169	Codebook index for track1
170~178	Codebook index for track2
179~187	Codebook index for track3
188~196	Codebook index for track4
197~203	VQ gain
Subframe 4	
204~209	Adaptive codebook index(relative)
210	LTP-filtering-flag
211~219	Codebook index for track1
220~228	Codebook index for track2
229~237	Codebook index for track3
238~246	Codebook index for track4
247~253	VQ gain

表 2.2 AMR-WB 編碼輸出及位元對照表

MD-AMR codec output parameter and corresponding bits			
parameter	Transmitted bits of Description I	# of bits	Transmitted bits of Description II
ISP	2~9	8	2~9
	10~7	8	10~7
	18~23	6	18~23
	31~37	7	31~37
	38~42	5	38~42
VAD	1	1	1
LTP-filtering	57, 107	2	160, 210

Pitch-delay	48~56	9	161~169
	101~106	6	204~209
Algebraic code	58~66	9	108~116
	67~75	9	117~125
	76~84	9	126~134
	85~93	9	135~143
	161~169	9	211~219
	170~178	9	220~228
	179~187	9	229~237
	188~196	9	238~246
Gains	94~100	7	144~150
	197~203	7	247~253
Total bits	138		

表 2.3 MD-AMR 編碼輸出及位元對照表

### 2.2.3 MD-G.729a 編碼器

MD-G.729a 編碼模式[7]，是依據 G.729a 編碼標準產生兩條平均位元速率相同的串流。為了使每條串流保持有效率的平均位元速率 (4.6 Kbps)，偶數音框和奇數音框分別以不同的位元數來編碼。

在每個音框中第二個次音框的音高延遲，是相對於其第一個次音框的音高延遲來編碼。假如沒有第一個次音框的音高延遲，則第二個次音框的音高延遲將無法解碼。因此，這兩個次音框的音高延遲資訊總是被分配在同一個串流來傳送。而 14 位元的適應性碼簿延遲部分，以串流 1 來說，是分配在奇數音框裡，在串流 2 這些位元則分配在偶數音框。

線頻譜對(LSPs)方面，G. 729 使用多階分次向量量化，在第一個階段，LSP 向量並沒有分開做量化，輸出的 8 位元將重複分配到二條串流來傳送。在第二個階段，維度 10 的殘餘向量分成兩個維度 5 的次向量(subvector)，且每個次向量個別以 5 位元來編碼。因此分配在每條串流中對應於 LSP 的位元總共有 13 個位元，包括第一個階段 8 位元和第二個階段 5 位元。至於碼簿索引值(codebook index)的分配，為了使兩條串流的品質更為平均，奇數音框(偶數音框)中對應碼簿索引的第一個(第二個)次向量由第一條串流來傳送，而第二條串流傳送的就是第二個(第一個)次向量。再來關於對應固定性碼簿向量及固定性碼簿符號(signs of fixed codebook)的位元分配，所有音框中的第一個次音框裡的資訊由第一條串流傳送，第二個次音框則由第二條串流傳送。適應性碼簿及固定性碼簿增益方面，由第一個(第二個)串流傳送第二個(第一個)次音框裡的資訊。

參照表(2.4)及(2.5)，可以清楚地將上述分配方式做些簡單的整理，由此可得知，奇數音框(偶數音框)的第一條(第二條)串流傳送 53 位元，偶數音框(奇數音框)的第一條(第二條)串流傳送了 39 位元，這些位元數是完成上述參數位元分配後，最後再加上兩個位元的音框指標(frame indicator)。它是用來描述此音框是屬於哪條串流，且是奇數還是偶數，如"00"則代表一個屬於第一條串流的奇數音

框，"01"則代表屬於第一條串流的偶數音框，"10"則代表屬於第二條串流的奇數音框，"11"則代表屬於第二條串流的偶數音框。

#### 2.2.4 MD-G.729a 解碼器

當兩條串流都成功接收時，接收端將可以利用所有資訊，完全回復成 G.729a 編碼過後的音框，再以 G.729a 的解碼器將語音完整呈現。若是兩條串流都漏失，則 G.729a 解碼器將啟動其漏失補償機制 (loss concealment)，盡可能的回覆所漏失掉的音框資訊。假如只收到一條串流，則解碼器會以最接近且正確收到的封包內容資訊來補償漏失掉的部分，如下述：

- (1) 漏失掉的第二階次向量設為 0。
- (2) 第一條串流中偶數(奇數)音框的音高延遲，則以前一個正確收到的音框的音高延遲加上 1 來取代。
- (3) 第一條串流中第二個次音框及第二條串流中第一個次音框的增益資訊由前一個次音框內容來取代。
- (4) 觀察分配時的方法可以理解，漏失掉的 LSP 向量可由收到的第一階向量和其中一個次向量來補償。

G. 729a codec output parameters and corresponding bits			
parameter	Bit position	# of bits	Content
Frame	---	1	SYNC_WORD
indicator	---	1	Frame size and fixed value 80
LSP stage1	1	1	MA predictor switch
	2~8	7	1st codebook
LSP stage2	9~13	5	2nd codebook low
	14~18	5	2nd codebook high
Pitch delay Subframe1	19~26	8	Pitch period
	27	1	Parity check on 1st period
Fixed codebook subframe1	28~40	13	Codebook pulse positions
Fixed codebook signs subframe1	41~44	4	Codebook pulse signs
Gains subframe1	45~47	3	Pitch and codebook gains stage 1
	48~51	4	Pitch and codebook gains stage 2
Pitch delay subframe2	52~56	5	Pitch period(relative)
Fixed codebook subframe2	57~69	13	Codebook pulse positions
Fixed codebook signs subframe2	70~73	4	Codebook pulse signs
Gains subframe2	74~76	3	Pitch and codebook gains stage 1
	77~80	4	Pitch and codebook gains stage 2

表 2.4 G. 729a 編碼輸出及位元對照表

MD-G. 729a codec output parameter and corresponding bits				
parameter	Description I		Description II	
	Odd packet Bit position (# of bits)	Even packet Bit position (# of bits)	Odd packet Bit position (# of bits)	Even packet Bit position (# of bits)
Frame indicator	--- (2)	--- (2)	--- (2)	--- (2)
LSP stage1	1 2~8 (8)	1 2~8 (8)	1 2~8 (8)	1 2~8 (8)
LSP stage2	9~13 (5)	14~18 (5)	14~18 (5)	9~13 (5)
Pitch delay subframe1	19~26 27 (9)			19~26 27 (9)
Pitch delay subframe2	52~56 (5)			52~56 (5)
Fixed codebook subframe1	28~40 (13)	28~40 (13)		
Fixed codebook subframe1			57~69 (13)	57~69 (13)
Fixed codebook signs subframe1	41~44 (4)	41~44 (4)		
Fixed codebook signs subframe2			70~73 (4)	70~73 (4)
Gains subframe1	45~47 48~51 (7)	45~47 48~51 (7)		
Gains subframe2			74~76 77~80 (7)	74~76 77~80 (7)
Total bits	53	39	39	53

表 2.5 MD-G. 729a 編碼輸出及位元對照表



## 2.3 多重敘述編碼的 $I_c$ 音質預測模型

在多重敘述傳輸過程中，接收端將會遇到三種情況，分別是兩條串流的封包皆成功接收( $\Omega_1$ )、只有其中一條串流的封包成功接收( $\Omega_2$ )、以及兩條串流封包皆發生漏失( $\Omega_3$ )。為了區隔 $\Omega_2$ 的單一封包漏失(packet loss)，我們將 $\Omega_3$ 的雙重封包漏失現象稱為音框刪除(frame erasure)。特別強調的是基於 MD 解碼器的功能， $\Omega_1$ 及 $\Omega_2$ 都被判定為"可播放"，不同的是 $\Omega_1$ 的音質較佳。問題是傳統的  $I_c$  音質預測模型，原是針對單一敘述傳輸系統而設計，只能反應封包漏失率(封包不能播放的比例)及不同語音編碼模式所造成的音質損害。而如上述 MD 解碼流程，我們可預期當接收端發生 $\Omega_2$ 情況時，其解碼輸出音質將不如 $\Omega_1$ 。因此 2.1.2 節所提到的傳統  $I_c$  音質預測模型已不敷使用，較理想的解決方案是必須將 $\Omega_1$ 及 $\Omega_2$ 這兩種接收情況做些區隔，且有效反應其所對應的輸出音質差異。

由於傳統的  $I_c$  音質預測模型無法正確反應 MD 語音封包的傳輸品質，所以我們必須尋找能夠反應出 $\Omega_1$ 及 $\Omega_2$ 音質差異的  $I_c$  音質預測模型。當發生 $\Omega_1$ 情況時，MD 解碼端將其兩條串流所含的封包資訊合併還原成完整的 G. 729a 或 AMR-WB 語音音框；而發生 $\Omega_2$ 情況時，由於只接收到其中一條串流，MD 解碼端會依據僅有的部份封包資訊透過補償將其重建成一個語音音框。而此語音音框的格式與 G. 729a 或

AMR-WB 相同，差別就在於其所含資訊不夠完整。上述 MD 解碼過程如圖(2.2)所示，其中虛線代表不能被用於播放的封包，而實線代表可用於被播放的封包，黑實心代表完整的語音音框，灰實心代表其所含資訊不夠完整的音框。

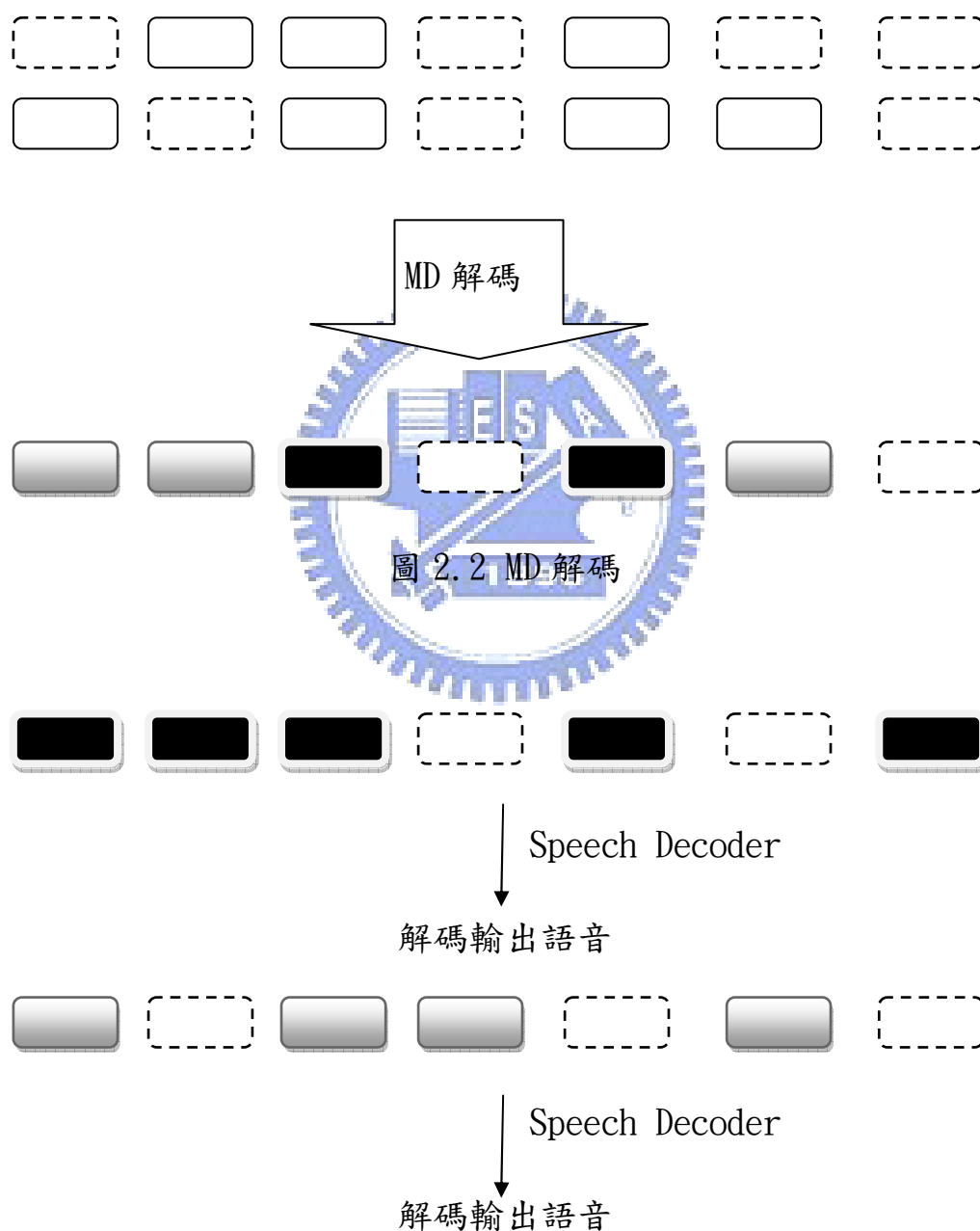


圖 2.3 封包經 MD 解碼後分屬兩種不同的語音編碼輸出

經由 MD 解碼後的語音音框，除了  $\Omega_3$  情況外，另兩種(黑實心與灰實心)可以視為分屬於兩種不同位元率的語音編碼輸出，因此 MD 解碼後的語音音框可以視為兩種不同的語音編碼輸出封包在網路上的傳輸漏失情形。

針對上述兩種不同語音編碼輸出的封包漏失狀況(如圖(2.3)所示)，我們的目標是尋找能夠精準預測其音質的  $I_e$  模型，進一步分析其編碼損害因子(黑實心及灰實心可視為其  $I_{e,codec}$  的不同)及封包漏失損害因子  $I_{e,pl}$ 。為了取得精確的  $I_e$  音質預測模型，我們參照前人研究的方法[15]使用 16 個英文語音檔案(包括 8 個男生、8 個女生，如表(2.7)所示)，來估測其封包漏失損害因子( $I_{e,pl}$ )及編碼損害因子( $I_{e,codec}$ )所對應的參數( $\gamma_{1,s}, \gamma_{2,s}, \gamma_{3,s}$ )，其中  $s=1$  對應於接收情況  $\Omega_1$ ， $s=2$  對應於接收情況  $\Omega_2$ 。

首先討論圖中的虛框情形，在一個特定的音框刪除比率(frame erasure rate, FER)= $\text{Prob}\{\Omega_3\}$  環境下，我們先將這些語音檔經過 MD 分成了兩個串流，再針對  $\Omega_1$  及  $\Omega_2$  這兩種情況分別來做 MD 解碼。由於封包漏失位置的不同，將造成輸出音質有所差異，所以經過多重敘述解碼處理的每個語音檔串流再分別選擇 30 組隨機指定封包漏失位置。最後再將解碼輸出的語音檔(此時已承受 MD 編解碼及封包漏失音質損害)，分別與其原始語音檔進入音質感知評估(Perceptual

Evaluation of Speech Quality, PESQ)演算法而得到 MOS 值，整個流程如圖(2.4)所示。特別強調的是，這些輸出語音檔並沒有考慮到延遲損害，所以將其經式(2.8)轉換得到的 R 值，將只有包括封包漏失損害及語音編碼損害。因此透過式(2.9)得到音質損害因子的量測值  $I_e = (I_{e,pl} + I_{e,codec})$ ，最後取其整體平均以求能夠補償不同的語者和漏失位置所造成的損害變異。依此工作流程，我們可得到在一特定的音框刪除率環境下， $\Omega_1$ 及 $\Omega_2$ 所對應的音質損害因子  $I_e$ ，如圖(2.5)及(2.6)所示。

$$R = 3.026MOS^3 - 25.314MOS^2 + 87.06MOS - 57.336 \quad (2.8)$$

$$I_e = 94.2 - R \quad (2.9)$$

圖中的  $FER = \text{Prob}\{\Omega_3\}$  是指一音框所屬的兩條串流同時發生漏失的發生比例。以  $FER = 0.05$  來說，代表假如傳送音框數有 200 個，其中 10 個音框會發生  $\Omega_3$ 。至於其他音框所屬兩串流的接收情況就是人工設定為  $\Omega_1$ (接收兩串流)或  $\Omega_2$ (接收單一串流)。圖中音質損害較高的曲線就是對應於  $\Omega_2$ ，音質損害較低的曲線就是對應於  $\Omega_1$ 。在測試過程當中，FER 在 0%到 30% 之範圍，每隔 3%做一次量測並得到相對應的損害因子，因而得到一組 11 個損害因子，分別對應 FER : 0%，3%，...，27%，30%。利用這組量測數值，根據最小平方差準則(least square error)，透過 NCSS 數值分析軟體[16]所提供的曲線匹配

(curve fitting)功能，執行非線性迴歸(nonlinear regression model)分析而取得其損害因子模型

( $I_{e,s}(e) = \gamma_{1,s} + \gamma_{2,s} \ln(1 + \gamma_{3,s}e)$ ,  $s=1,2$ )。如表(2.6)所示，其中  $I_{e,1}$  與  $I_{e,2}$  是分別對應於  $\Omega_1$  及  $\Omega_2$  漏失情況下的音質損害預測模型。

Codec (Both or only one stream available)	$(\gamma_1, \gamma_2, \gamma_3)$
MD-G. 729a (both streams available)	(21.962, 17.016, 16.088)
MD-G. 729a (only one stream available)	(52.6143, 191870, $2.08 \times 10^{-4}$ )
MD-AMR (both streams available)	(20.084, 22.958, 17.32)
MD-AMR (only one stream available)	(53.751, 111307, $6.06 \times 10^{-4}$ )

表 2.6  $I_e$  模型參數

語音檔 (wav)	內容
Male 1	Faces may be made into candles by filling with melted wax and a wick.
Male 2	Whoever cooperates in finding Nan's cameo will be rewarded.
Male 3	Brush fires are common in the dry underbrush of Nevada.
Male 4	To kayo him and maybe or maybe not kill.
Male 5	We got drenched from the uninterrupted rain.
Male 6	She had your dark suit in greasy wash water all year.

Male 7	He ripped down the cell-phone carefully and laid three dogs on the tin foil.
Male 8	No, they could kill him just as easy right now.
Female 1	These men were without capital or experience.
Female 2	The library has open shelves even in the unbound periodical stockroom.
Female 3	A site may also be attractive just through the beauty of its trees and shrubs.
Female 4	Masquerade parties tax one's imagination
Female 5	You do that or take you out a permit right now.
Female 6	They weren't as well paid as they should have been.
Female 7	Good services should be rewarded by big tips.
Female 8	Ran away on a black night with a lawful wedded man.

表 2.7 語音檔內容

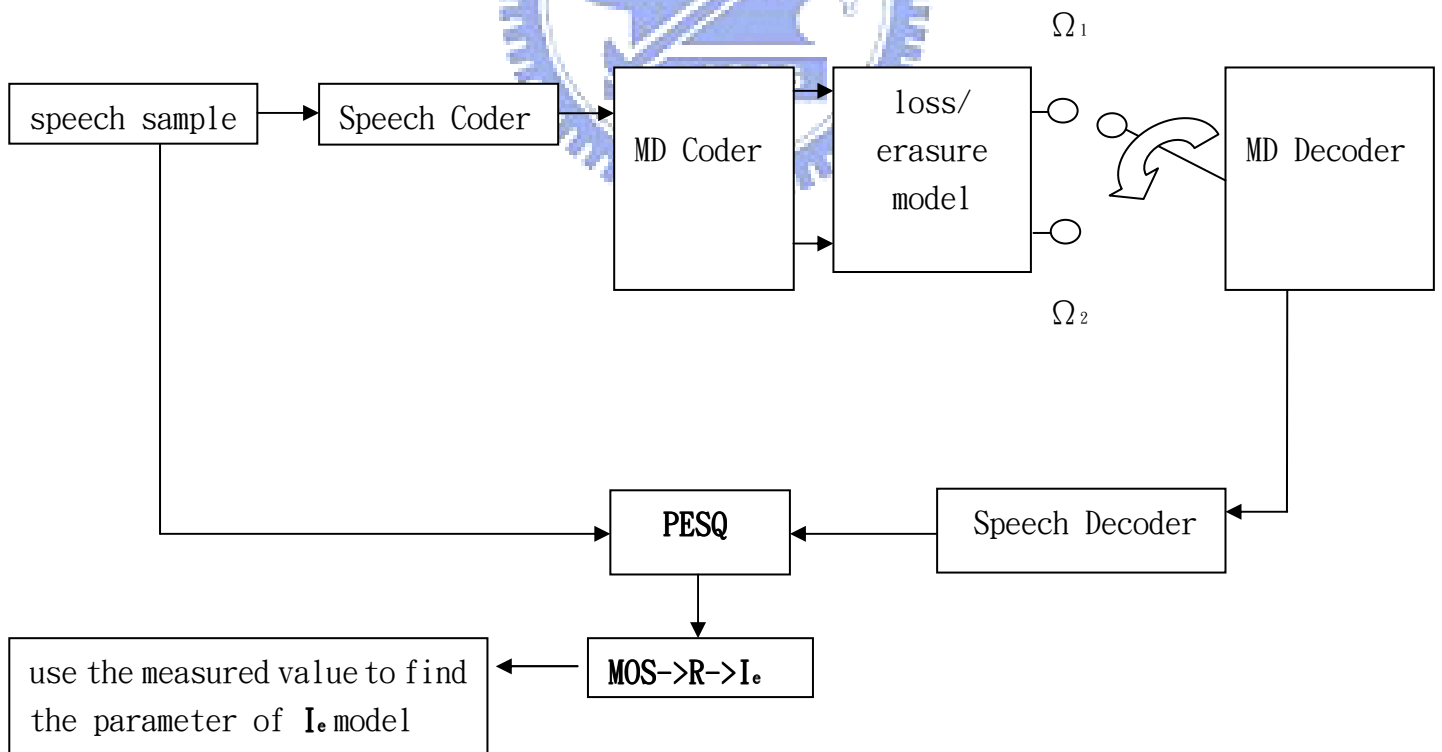


圖 2.4 尋找新  $I_e$  模型的實驗流程

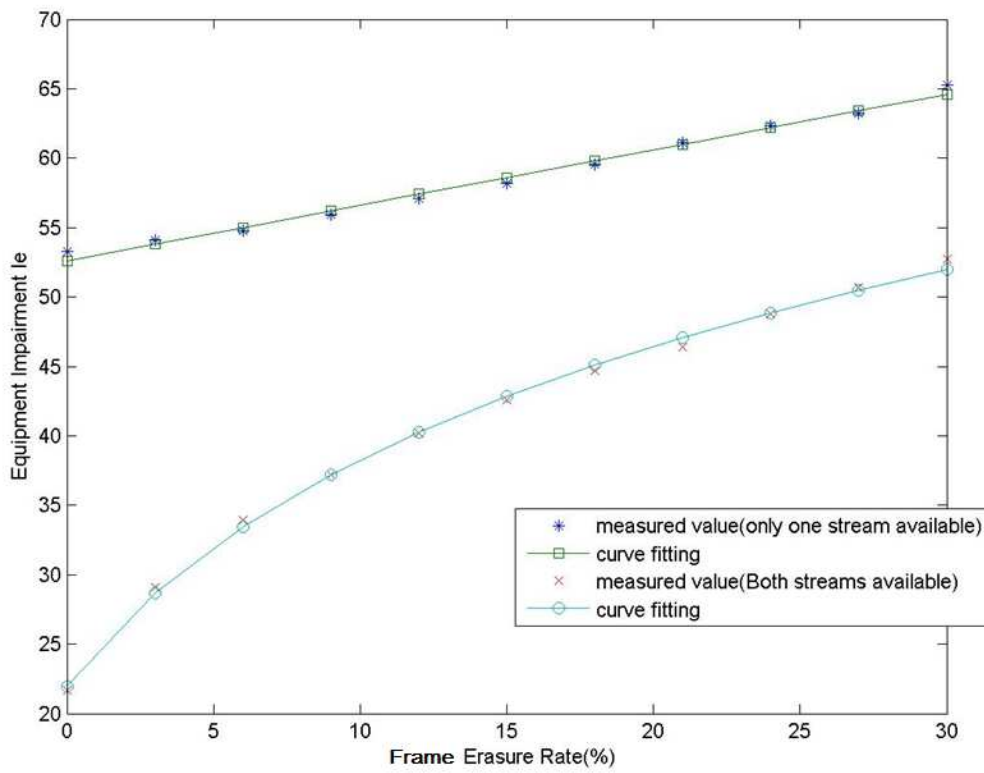


圖 2.5 MD-G.729a 編碼的音質損害因子

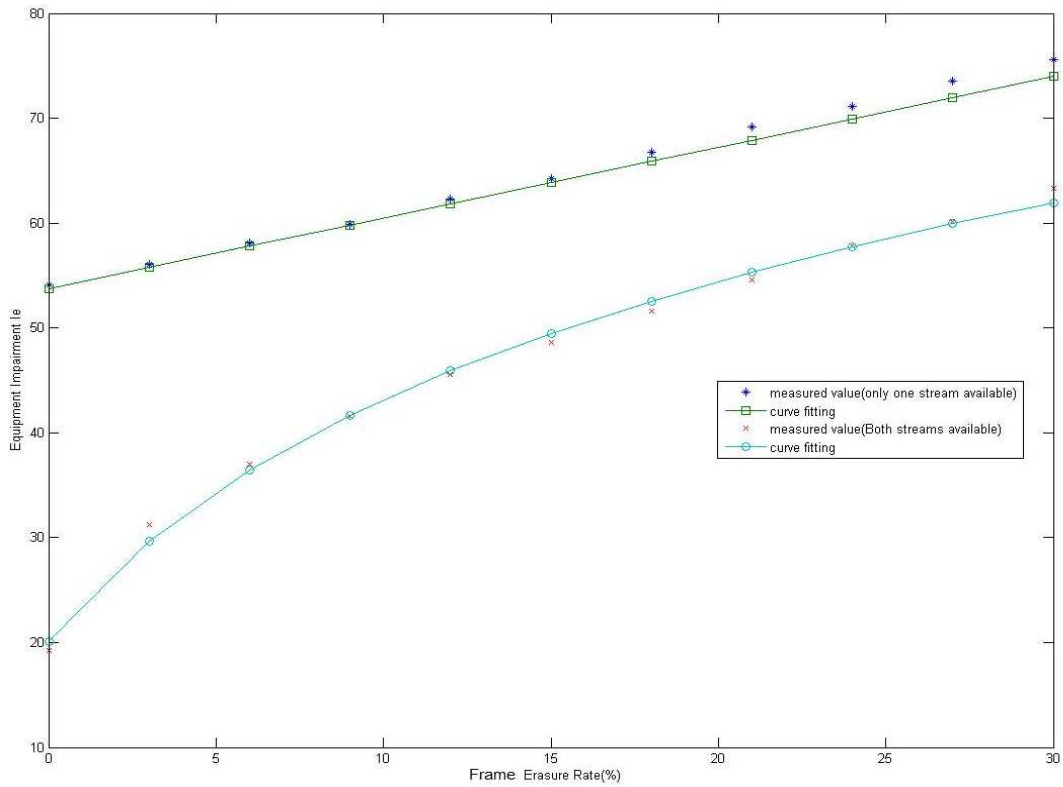


圖 2.6 MD-AMR 編碼的音質損害因子

## 2.4 音質損害預測模型的整合

在上一小節，我們基於 MD 編解碼的特性，透過 PESQ 找出能夠分別反應  $\Omega_1$  及  $\Omega_2$  接收情況的音質損害預測模型 ( $I_{e,1}$  及  $I_{e,2}$ )。在此，我們將進一步把這兩個預測模型做適度的整合，使其能夠準確預測實際環境中封包傳輸情況  $\Omega_1$  及  $\Omega_2$  混合發生時的輸出音質。令 FER  $\Pr\{\Omega_3\} = e$ ，而  $\Pr\{\Omega_1 \cup \Omega_2\} = 1 - \Pr\{\Omega_3\}$  是指有單一或兩個封包可用於播放其音框的機率。我們提出新的音質損害預測模型  $I_e(e)$ ，如下所示：

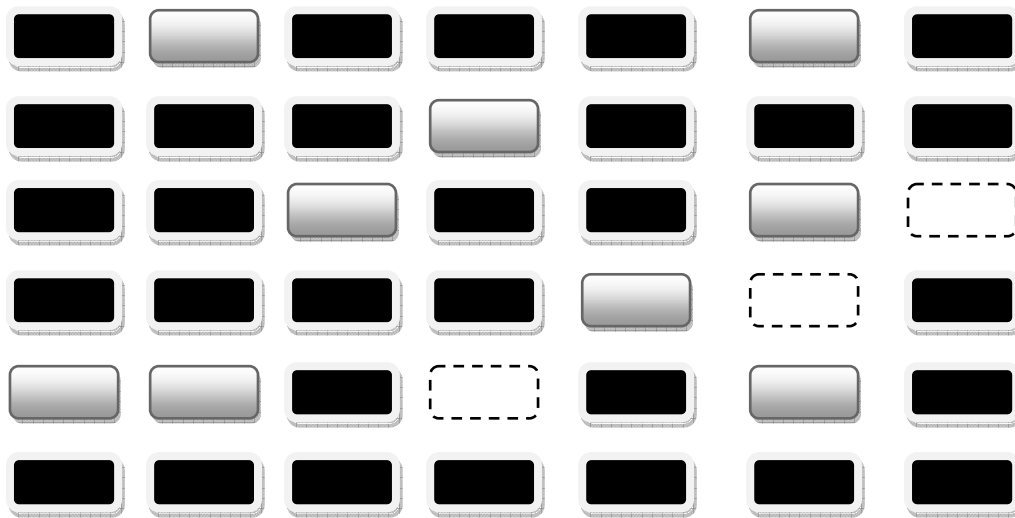
$$I_e(e) = \sum_{j=1}^2 \rho_j I_{e,j}(e)$$

$$\rho_1 = \Pr\{\Omega_1 | \Omega_1 \cup \Omega_2\} = \frac{\Pr\{\Omega_1, \Omega_1 \cup \Omega_2\}}{\Pr\{\Omega_1 \cup \Omega_2\}} = \frac{\Pr\{\Omega_1\}}{\Pr\{\Omega_1 \cup \Omega_2\}} \quad (2.10)$$

$$\rho_2 = \Pr\{\Omega_2 | \Omega_1 \cup \Omega_2\} = 1 - \rho_1$$

這個新  $I_e(e)$  模型簡單來說是，在音框可以播放部分或完整的條件下，先將兩串流封包分成兩類，分別對應  $\Omega_1$  及  $\Omega_2$  兩種情況。並依據這兩種情況的發生比例  $\rho_1$  及  $\rho_2$ ，對  $I_{e,1}$  及  $I_{e,2}$  做加權並整合所得。為了解新  $I_e(e)$  模型的運作模式，我們舉下例來表示說明：





圖中所示是 42 個 G. 729 音框在接收端經由 MD 解碼後的情形，如前面所敘述，黑實心框代表  $\Omega_1$  的兩串流經由 MD 解碼後的音框，灰實心框則代表  $\Omega_2$  的單一串流經由 MD 解碼後的音框， $\Omega_3$  的音框刪除則由虛線框來表示。由這些封包接收情況，可以算出  $\Pr\{\Omega_3\} = 3/42 = 0.07$

$$\Pr\{\Omega_1 \cup \Omega_2\} = 1 - \Pr\{\Omega_3\} = 0.93$$

$$\Pr\{\Omega_1, \Omega_1 \cup \Omega_2\} = \Pr\{\Omega_1\} = 26/39 = 2/3$$

因此，這一段音框區塊因其編碼及封包漏失所造成的損害因子，配合表(2.6)可得其新  $I_e(e)$  模型的預測值為：

$$I_e(0.07) = \frac{2}{3} I_{e,1}(0.07) + \frac{1}{3} I_{e,2}(0.07) = 41.6689$$

## 第三章

# 聽覺最佳化的適應性播放排程設計

近年來，廣受歡迎的網路電話(VoIP)發展迅速，因為能讓使用者節省可觀的長途或國際電話費。但網路電話仍存在諸多問題，常見的問題為整體延遲(end-to-end delay)、延遲擾動(delay jitter)、封包漏失(packet loss)、以及回音(echo)等。聲音在網路上傳送通常是被切割成一個個封包，所以封包到達接收端時的延遲和漏失，被視為評估網路電話品質好壞的準則。在傳送端，語音信號會以固定的音框間隔依序來產生封包，並透過網際網路傳送到接收端。其網路延遲會取決於行走的路徑及該路徑上路由器(Router)的擁塞程度而有所不同，而這些網路延遲的差異即為延遲擾動。為降低延遲擾動在接收端的影響，接收封包在播放前會先被暫存在緩衝器一小段時間。嚴重晚到的封包，即封包在排定的播放時間後才到達，則被視為晚到漏失(late loss)。藉由增加緩衝器延遲(buffer delay)，晚到漏失的封包將會減少，然而這將增加封包的整體延遲。針對這個議題，本章節的研究將結合第二章的音質預測模型，因應 MD 傳輸中的網路變動情形，來動態設計其適應性的播放排程機制。

### 3.1 播放緩衝器

在網路語音傳輸系統中，傳送端以固定的音框間隔  $T_p$  產生封包並經由網路傳送，而由於網路本身的特性，每個封包延遲並不會固定，導致有些封包會在接收端預定的播放時間之後才到達。圖 3.1(a)說明了延遲擾動所造成的問題，在缺乏播放緩衝器的情形下，封包會在被接收到的同時隨即被播放出去，第一個封包抵達時間即為其開始播放時間，接下來的第  $i$  個封包將以和第一個封包的播放時間間隔  $(i-1)T_p$  作為播放時間。然而，較大的網路延遲會造成晚到的封包(如圖中第四個封包)無法順利播出，導致部分的封包漏失而降低通話品質。加入播放緩衝器之後，封包抵達後將暫存於緩衝器一小段時間再播放，如圖 3.1(b)。此方法可大幅減少封包因晚到而漏失的機率，但整體延遲將從原本的網路延遲擴大為網路延遲與緩衝延遲的總合。

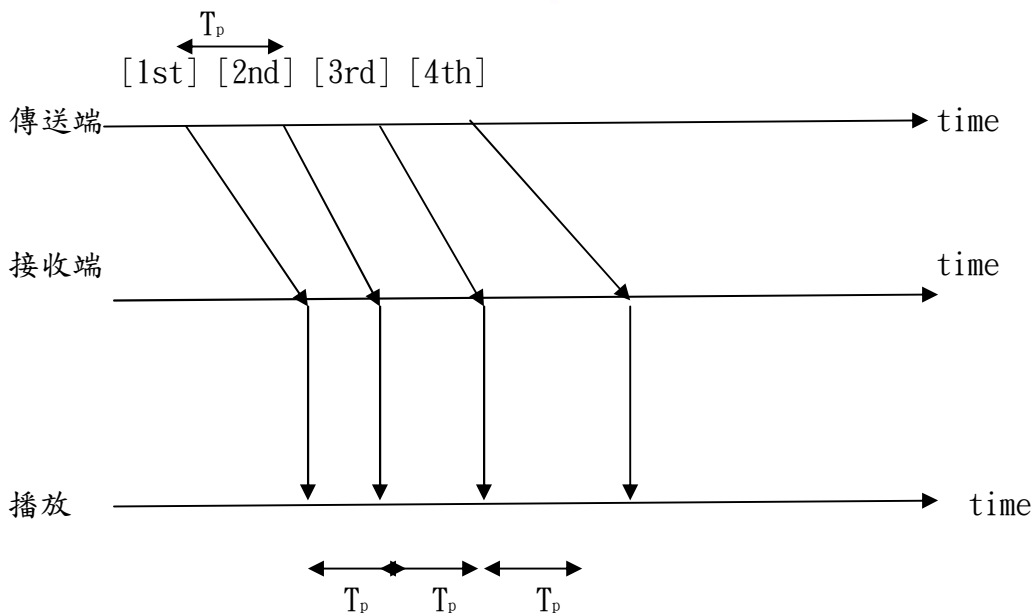


圖 3.1(a)未加入緩衝延遲

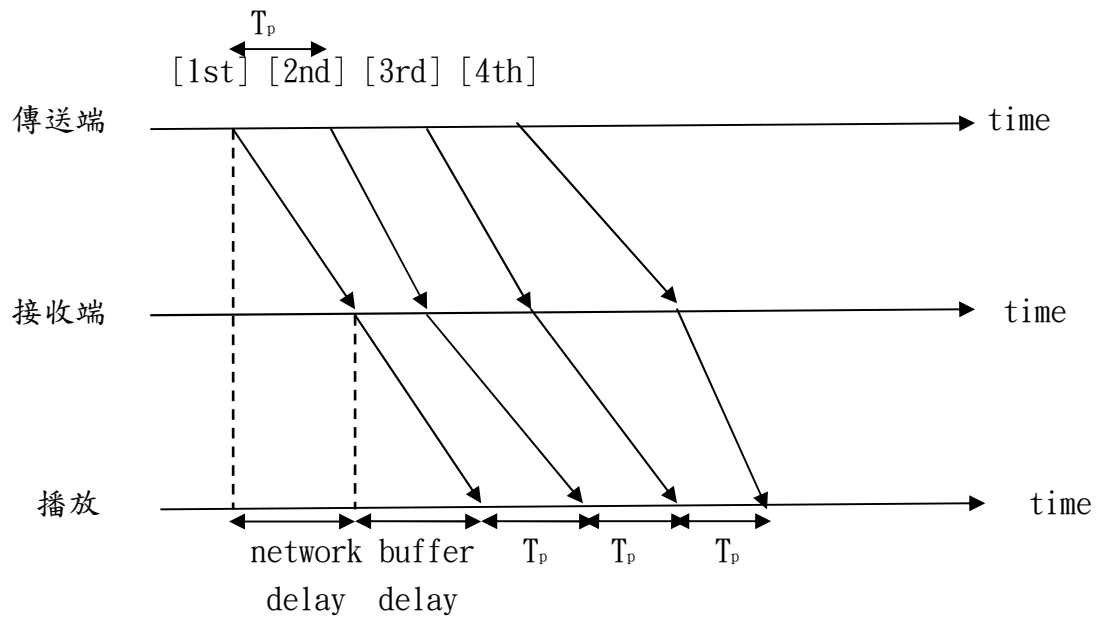


圖 3.1(b) 引入緩衝延遲(buffer delay)機制

圖 3.1 播放緩衝器的影響

因此如何在封包漏失及播放延遲之間取得平衡點，進而設計一個能因應網路時變特性的播放排程機制，是網路語音傳輸系統中一項重要的議題。針對播放排程的問題，近年來已有學者提出聽覺最佳化的原則，針對每個封包調整播放延遲(per-packet adjustment)[17]，亦即每個封包的播放延遲相異。問題是這種基於封包調整的播放排程機制雖可迅速因應網路延遲的動態擾動，卻衍生更棘手的不連續播放問題。假如當第  $i$  個封包的播放延遲大於或小於第  $i+1$  個封包播放延遲時，則將會造成聲音在播放時突然產生空白或扭曲。為了解決此封包播放不連續的問題，常見的做法是在接收端引入音長調整(Time-Scaling)[18]。而其機制主要是針對個別封包對應的播放延遲來調整其封包長度，透過拉長或縮小封包長度，來填補封包間的時間

空格或避免兩兩封包播放時間相互衝突(例如當要播放第  $i+1$  個封包時，而第  $i$  個封包卻還沒完成播放)。雖然這可解決播放不連續的問題，但是顯然地，封包長度的拉長或縮短將造成聲音突然加速或是減慢的情況。縱使如此靈敏的播放調整機制可大幅降低封包晚到的漏失比例，但是過度的音長調整仍有可能造成音質降低甚至無法辨認。

有別於上述基於封包的調整機制，另一種作法是在相鄰話務間調整其靜音(silence)區段時間的播放排程機制(per-talkspurt adjustment)[15]。在語音通話中，一段聲音的傳輸包括了話務(talkspurt)及靜音。以「No, they could kill him just as easy right now.」為例，這段話包含 10 個話務，每個話務本身是由數個甚至數十個封包所組成，其餘的則為 silence。調整播放延遲的做法就是在每個話務開始之前，一次決定其所屬封包的播放延遲。因此當設定完播放延遲後，每段話務裡的所有封包其播放延遲皆相同，也就是當接收端在播放此話務時，每個封包將依序播放出來。相較於每個封包長度的調整機制，基於話務的調整機制確實不能及時反應網路延遲情況，但是這種退而求次的方法，至少避免掉使用音長調整而造成額外的音質損害。

本論文探討主題是在每段話務之間調整其播放延遲，雖然降低封包晚到漏失的比例不如封包調整機制。但由於我們是著重於 MD 傳輸

系統之下，設計適應性播放排程演算法以調整話務之間靜音區段的長度，仍能透過 MD 傳輸的優勢有效的降低封包晚到漏失所造成的音質損害。MD 傳輸的播放排程機制設計，基本上來說，是透過估計兩條串流傳輸過程中的網路延遲，結合適應性播放排程演算法來設定其聽覺最佳化的播放延遲。至於網路延遲的估計過程如下：

其中  $\hat{d}_i^{(s)}$  和  $\hat{v}_i^{(s)}$  分別是代表串流 s 傳輸中，第 i 個封包網路延遲的平均值和變異數之估計值，分別是以自迴歸方法(Autoregressive , AR method)[17]來估計

$$d_{play,i} = \hat{d}_i + \beta \hat{v}_i \quad (3.1)$$

$$\hat{d}_i^{(s)} = \alpha \hat{d}_{i-1}^{(s)} + (1-\alpha)n_{i-1}^{(s)} \quad (3.2)$$

$$\hat{v}_i^{(s)} = \alpha \hat{v}_{i-1}^{(s)} + (1-\alpha)|n_{i-1}^{(s)} - \hat{d}_{i-1}^{(s)}| \quad (3.3)$$

串流 s 中第 i 個封包網路延遲的平均與變異數估計值，是由該串流中前一個封包對應的估計值  $\{\hat{d}_{i-1}^{(s)}, \hat{v}_{i-1}^{(s)}\}$ ，配合其實際量測的網路延遲  $n_{i-1}^{(s)}$  分別加權所組成，可以預期  $\alpha$  的大小會影響估計的準確性。在此  $\alpha$  值設為 0.998002。式(3.1)中的  $\beta$  是個用來設定播放延遲的安全因子(safety factor)，讓設定的播放時間比封包抵達的估計時間更晚一點，讓播放排程有更足夠的時間來播放。當  $\beta$  值越高時，可想而知封包遲到的比例得以降低，但是同時也增加了整體延遲，因此  $\beta$  在播放排程演算法中扮演著關鍵的權衡(trade-off)角色。

### 3.2 晚到漏失機率模型

為了設定聽覺最佳化的播放緩衝延遲， $\beta$ 值的設定需要更精準的網路延遲分佈模型來幫助我們建立晚到漏失機率模型。首先，定義網路延遲的累積分佈函數 $F_D(d)$ 如下，

$$F_D(d) = Prob\{D \leq d\}$$

此式表示網路延遲不大於 $d$ 的機率。第 $i$ 個語音封包的晚到漏失，發生於網路延遲大於其所設定播放延遲 $d_{play,i}$ 的時候，因此晚到漏失機率 $e_{b,i}$ 可定義為：

$$e_{b,i} = Prob\{D > d_{play,i}\} = 1 - F_D(d_{play,i}) \quad (3.4)$$

上式建立了 $d_{play,i}$ 與 $e_{b,i}$ 之間的關係。觀察式(3.4)，由機率的基本公設得知 $F_D(d_{play,i})$ 是一個 $d_{play,i}$ 的遞增函數，確實 $d_{play,i}$ 越大則 $e_{b,i}$ 會越小。

前人提出了許多統計模型來描述封包傳輸的網路延遲特性，較常見的有Exponential模型及Pareto模型，其定義示於表(3.1)。

分佈	Exponential 分佈	Pareto 分佈
CDF: F(x)	$1 - e^{-(x-k_0)/\mu}, x - k_0 \geq 0$	$1 - (k/x)^g, x \geq k$

表 3.1 網路延遲的累積分佈函數

Exponential模型參數 $\{k_0, \mu\}$ 及Pareto模型參數 $\{k, g\}$ ，皆可由過去所量測到的一組L個網路延遲 $\{n_{i-1}, n_{i-2}, \dots, n_{i-L}\}$ ，依最大相似度估計理論(Maximum-Likelihood Estimation)計算而得。由前人研究，可知估

算公式如下[19]:

$$\hat{k}_0 = \min\{n_{i-1}, n_{i-2}, \dots, n_{i-L}\}$$

$$\hat{\mu} = \text{mean}\{n_{i-1}, n_{i-2}, \dots, n_{i-L}\} - \hat{k}_0$$

$$\hat{k} = \min\{n_{i-1}, n_{i-2}, \dots, n_{i-L}\}$$

$$\hat{g} = L \left( \sum_{l=i-1}^{i-L} \log\left(\frac{n_l}{\hat{k}}\right) \right)^{-1}$$

為了比較這兩種模型與網路延遲實際分佈的差異，首先利用網路延遲模型產生兩組資料，平均延遲分別為55msce以及240msec，其中平均延遲較大的一組具有較多的Spike現象。利用這些資料以及表(3.1)求出實驗的(Empirical)與模型化累積分佈函數，如圖(3.2)與圖(3.3)所示。



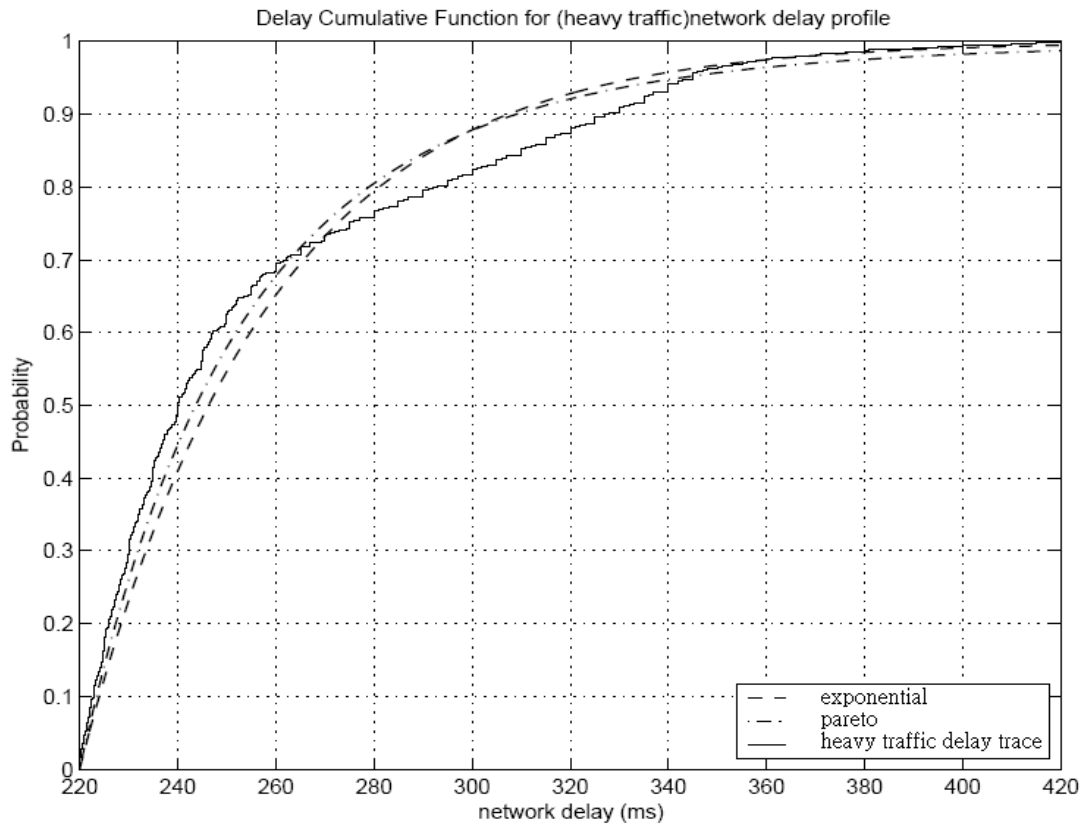


圖 3.2: 平均延遲為 240msec 的累積分佈函數

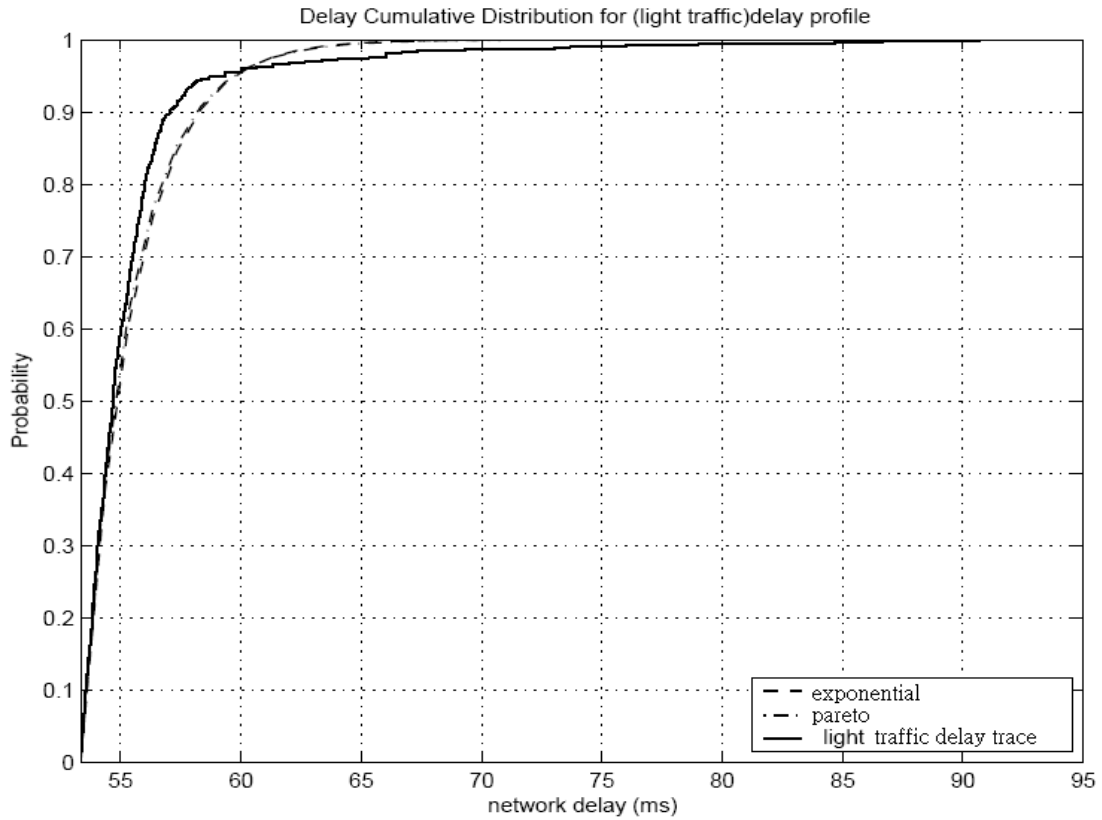


圖 3.3: 平均延遲為 55msec 的累積分佈函數圖形

當平均延遲較低時，Exponential 與 Pareto 延遲分佈模型的效果差異不大。然而在平均延遲較高的網路情況(且含有許多 Spike)，Pareto 分佈模型比 Exponential 分佈模型更適用於描述網路的延遲特性。因此本章節將選用 Pareto 模型來計算語音封包網路延遲機率，並藉此建立晚到漏失機率與播放延遲的關係。

### 3.3 MD 傳輸系統的播放排程設計

在這個章節我們將先介紹既有的 MD 播放排程演算法，之後再結合第二章所介紹的 MD 音質損害預測模型  $I_e$ ，考量封包於 MD 傳輸中的接收情況及其所對應的音質損害，進而設計聽覺最佳化的播放排程。

#### 3.3.1 以低延遲為考量的設計

前人研究中所提出的 MD 播放排程設計[6]，是將播放延遲  $d_{play,i}$  及音框不能完整播放的接收情況( $\Omega_2$  及  $\Omega_3$ )整合成一個成本函數：

$$\begin{aligned}
 C^i &= d_{play,i} + \lambda_1 \text{probability(both descriptions lost)} \\
 &\quad + \lambda_2 \text{probability(only one descriptions lost)} \\
 &= d_{play,i} + \lambda_1 \Pr\{\Omega_3\} + \lambda_2 \Pr\{\Omega_2\}
 \end{aligned} \tag{3.5}$$

式(3.5)是表示對應於第  $i$  個音框所建立的成本函數，並藉由調整  $\lambda_1$  及  $\lambda_2$  的大小來衡量取捨整體延遲及音框刪除的發生機率。此播放排程設計的概念，主要是尋找能夠使成本函數  $C^i$  達到最低值的最佳播放延遲  $d_{play,i}$ 。由 3.1 節可以得知延遲跟封包漏失是兩個無法同時降低的音質損害來源，透過事先設定的  $\lambda_1$  及  $\lambda_2$  尋找最佳播放延遲。而不一樣的參

數會尋找出不一樣的最佳播放延遲，由式(3.4)可以看出將 $\lambda_1$ 調整大一點，就是利用較長的播放延遲來換取更低的 frame erasure rate，希望至少能收到其中一條串流。而若將 $\lambda_2$ 調整大一點就是利用較長的播放延遲來換取更好的音框重建品質，藉著更長的播放延遲讓音框可由兩條串流來回復。另外，假若只單方面調整任何一個參數，將無法全面考量所有損害因子，例如當 $\lambda_2$ 被調整為極小值甚至是0時，代表此成本函數將只考量到延遲與 frame erasure rate 所造成的音質損害且並不是以音質預測模型為基礎來設計。另外，這樣的調整並沒有將音框播放品質的差異( $\Omega_1$ 及 $\Omega_2$ )納入考量，而是一味的降低播放延遲，雖然如此有效地降低延遲造成的音質損害，但因為播放延遲過短而造成在接收端有更多音框只收到其中一條串流。而在第二章的探討中，可以得知 $\Omega_1$ 及 $\Omega_2$ 兩種音框回復情形下的播放品質是無法相提並論的。因此很明顯可以看出，如此透過先調整參數( $\lambda_1$ 及 $\lambda_2$ )再依此參數來建立成本函數並設計播放排程的做法，並沒有將接收端音框播放品質納入考量。

### 3.3.2 音質最佳化的設計

相較於傳統的 SD 傳輸模式，MD 傳輸模式的重大優點，在於提供更具有彈性的 trade-off(整體延遲、frame erasure、對應於  $\Omega_1$  及  $\Omega_2$  的音框重建品質)。由第二章的結果可以得知最後兩項會反應出不一樣的語音品質，基於如此的發現，我們將其延伸引用到 MD 傳輸系統中，並據以設計基於聽覺最佳化的適應性播放排程設計。

基於聽覺最佳化的 MD 傳輸播放緩衝設計，我們必須將延遲、封包漏失、MD 解碼、語音編解碼、以及語音音框重建後的音質納入考量。以下整合上述這些影響因子，並依據第二章所提出的音質預測模型，建立一個能有效反應音質損害的成本函數：

$$I_{m,i}(d_i, e_i) = I_d(d_i) + I_e(e_i) = I_d(d_i) + \sum_{j=1}^2 \rho_j I_{e,j}(e_i) \quad (3.6)$$

其中整體延遲  $d_i$  是編碼延遲  $d_c$  加上播放延遲  $d_{play,i}$ ，也就是

$d_i = d_c + \hat{d}_i + \beta_i \hat{v}_i$ 。這裡所指的  $e_i$  是第  $i$  個音框不能播放的發生機率，也就是兩個串流同時漏失的  $FER = \text{Prob}\{\Omega_3\}$ ，可以表示為：

$$e_i = e_n^{(1)} e_n^{(2)} + e_n^{(1)} (1 - e_n^{(2)}) e_{b,i}^{(2)} + e_n^{(2)} (1 - e_n^{(1)}) e_{b,i}^{(1)} + (1 - e_n^{(1)}) (1 - e_n^{(2)}) e_{b,i}^{(1)} e_{b,i}^{(2)} \quad (3.5)$$

其中  $e_n^{(s)}$  分別代表串流  $s$  的網路漏失率， $e_{b,i}^{(s)}$  則代表串流  $s$  中的第  $i$  個封包的晚到漏失機率。而相應於第二章所提到的  $\rho_1$ ，在這裡則可以表示成：

$$\begin{aligned}\rho_1 &= \Pr\{\Omega_1 | \Omega_1 \cup \Omega_2\} = \frac{\Pr\{\Omega_1, \Omega_1 \cup \Omega_2\}}{\Pr\{\Omega_1 \cup \Omega_2\}} = \frac{\Pr\{\Omega_1\}}{\Pr\{\Omega_1 \cup \Omega_2\}} \\ &= \frac{1}{1-e_i} (1-e_n^{(1)})(1-e_n^{(2)})(1-e_{b,i}^{(1)})(1-e_{b,i}^{(2)})\end{aligned}\quad (3.7)$$

$\Pr(\Omega_1 \cup \Omega_2) = 1 - e_i$  代表音框可以被播放的機率，而  $\Pr\{\Omega_1\}$  指的是在播放時間前音框可以完整重建的機率，也就是說在這兩條串流中的封包皆不能發生網路漏失及晚到漏失，因此以  $(1-e_n^{(1)})(1-e_n^{(2)})(1-e_{b,i}^{(1)})(1-e_{b,i}^{(2)})$  來表示。而透過 Pareto 分佈來模擬網路延遲分布，晚到漏失機率可以表示成  $1 - F_{D,s}(x) = \left(\frac{k_s}{x}\right)^{s_s}$ ，可知  $\beta_i$  的大小會直接影響 FER。因此，我們可將

成本函數簡化成： $I_{m,i}(d_i, e_i) = I_{m,i}(\beta_i)$ ，進一步對  $\beta_i$  微分可得到：

$$I'_{m,i}(\beta_i) = c \hat{v}_i + \sum_{j=1}^2 \left\{ \rho_j \frac{\gamma_{2,j} \gamma_{3,j}}{1 + \gamma_{3,j} e_i} \frac{de_i}{d\beta_i} + \frac{d\rho_j}{d\beta_i} I_{e,j}(e_i) \right\} \quad (3.8)$$

其中

$$c = \begin{cases} 0.024, \beta_i < (177.3 - d_c - \hat{d}_i) / \hat{v}_i \\ 0.134, \beta_i > (177.3 - d_c - \hat{d}_i) / \hat{v}_i \end{cases} \quad (3.9)$$

$$\begin{aligned}\frac{d\rho_1}{d\beta_i} &= \frac{\hat{v}_i}{d_{play,i}(1-e_i)} (1-e_n^{(1)})(1-e_n^{(2)}) (\alpha_1 e_{b,i}^{(1)}(1-e_{b,i}^{(2)}) + \alpha_2 e_{b,i}^{(2)}(1-e_{b,i}^{(1)})) \\ &+ \frac{1}{(1-e_i^2)} \frac{de_i}{d\beta_i} (1-e_n^{(1)})(1-e_n^{(2)})(1-e_{b,i}^{(1)})(1-e_{b,i}^{(2)})\end{aligned}\quad (3.10)$$

至於  $e_i$  在  $\beta_i$  上的梯度 (gradient)，則可以寫成：

$$\frac{de_i}{d\beta_i} = \frac{-\hat{v}_i}{d_{play,i}} \left\{ (1-e_n^{(1)})(1-e_n^{(2)}) e_{b,i}^{(1)} e_{b,i}^{(2)} (g_1 + g_2) + e_n^{(1)}(1-e_n^{(2)}) e_{b,i}^{(2)} g_2 + e_n^{(2)}(1-e_n^{(1)}) e_{b,i}^{(1)} g_1 \right\} \quad (3.11)$$

接下來透過數值分析正割法(secant method)[20]，尋找成本函數最小值所對應的 $\beta_i$ ，從初始值 $\beta_i(-1)$ 和 $\beta_i(0)$ 開始，然後以式(3.11)開始做疊代計算。

$$\beta_i(j+1)=\beta_i(j)-\frac{\beta_i(j)-\beta_i(j-1)}{I'_{m,i}(\beta_i(j))-I'_{m,i}(\beta_i(j-1))}I'_{m,i}(\beta_i(j)) \quad (3.12)$$

新的 $\beta_i(j+1)$ 則用在下一次的疊代過程中，而這個疊代的動作將持續直到 $|\beta_i(j+1)-\beta_i(j)|$ 小於設定的臨界值(threshold)。因此當疊代的動作停止時，就是找到可使第 $i$ 個封包音質損害達到最低而聽覺最佳化的 $\beta_i$ 值。

最後，我們整理基於聽覺最佳化的MD傳輸系統，如圖(3.4)所示，其適應性播放排程演算法如下：

1. 首先在每個 talkspurt 裡，持續以 AR 方法估計平均網路延遲 $d_i^{(s)}$ 和網路延遲變異數 $v_i^{(s)}$ ，如式(3.2)及(3.3)，直到該 talkspurt 中最後一個封包。
2. 在串流 $s$ 中，每個 talkspurt 裡最後一個封包，更新其之前 $L$ 個封包的網路延遲與封包網路漏失記錄，並利用它們透過最大概似估計方法來計算 Pareto 分佈模型的參數 $(k_s, g_s)$ ，以及 $(e_n^{(1)}, e_n^{(2)})$ 。
3. 利用 secant method 來計算對應於成本函數最小值的 $\beta_i^{(s)}$ ：

$$I_{m,i}^{(s)}(\beta_i^{(s)})=I_d(d_c+d_i+\beta_i^{(s)}v_i^{(s)})+I_e(e_i(d_i+\beta_i^{(s)}v_i^{(s)})) \quad (3.12)$$

由於我們是在 talkspurt 之間調整播放延遲，且是以 talkspurt 中最後一個封包所記錄的網路傳輸資訊(網路延遲及漏失)來建立成本函數。所以在串流  $s$  中，每個 talkspurt 將只會尋找一次對應於成本函數最小值的  $\hat{\beta}_i^{(s)}$ ，且發生在 talkspurt 中最後一個封包，這樣比起針對每個封包來調整播放延遲的機制將會減少很多的計算量。

#### 4. 設定播放延遲：

由上述步驟，串流  $s$  裡任何一個 talkspurt 中最後一個封包將會找出一個成本函數最小值的  $\hat{\beta}_i^{(s)}$ 。由於這兩條串流裡的封包屬於同一個語音框資訊，因此兩條串流的播放延遲必須設定相同，且是以同時使得兩串流成本函數得到最低值的  $\hat{\beta}_i$ ，做為設定下一個 talkspurt 傳輸播放延遲的安全因子。如式(3.14)所示：

$$d_{play,i} = d_i^{(s^*)} + \hat{\beta}_i^{(s^*)} v_i^{(s^*)} \quad (3.14)$$

$$s^* = \arg \min \{I_{m,i}^{(s)}(\hat{\beta}_i^{(s)}), s = 1, 2\}$$

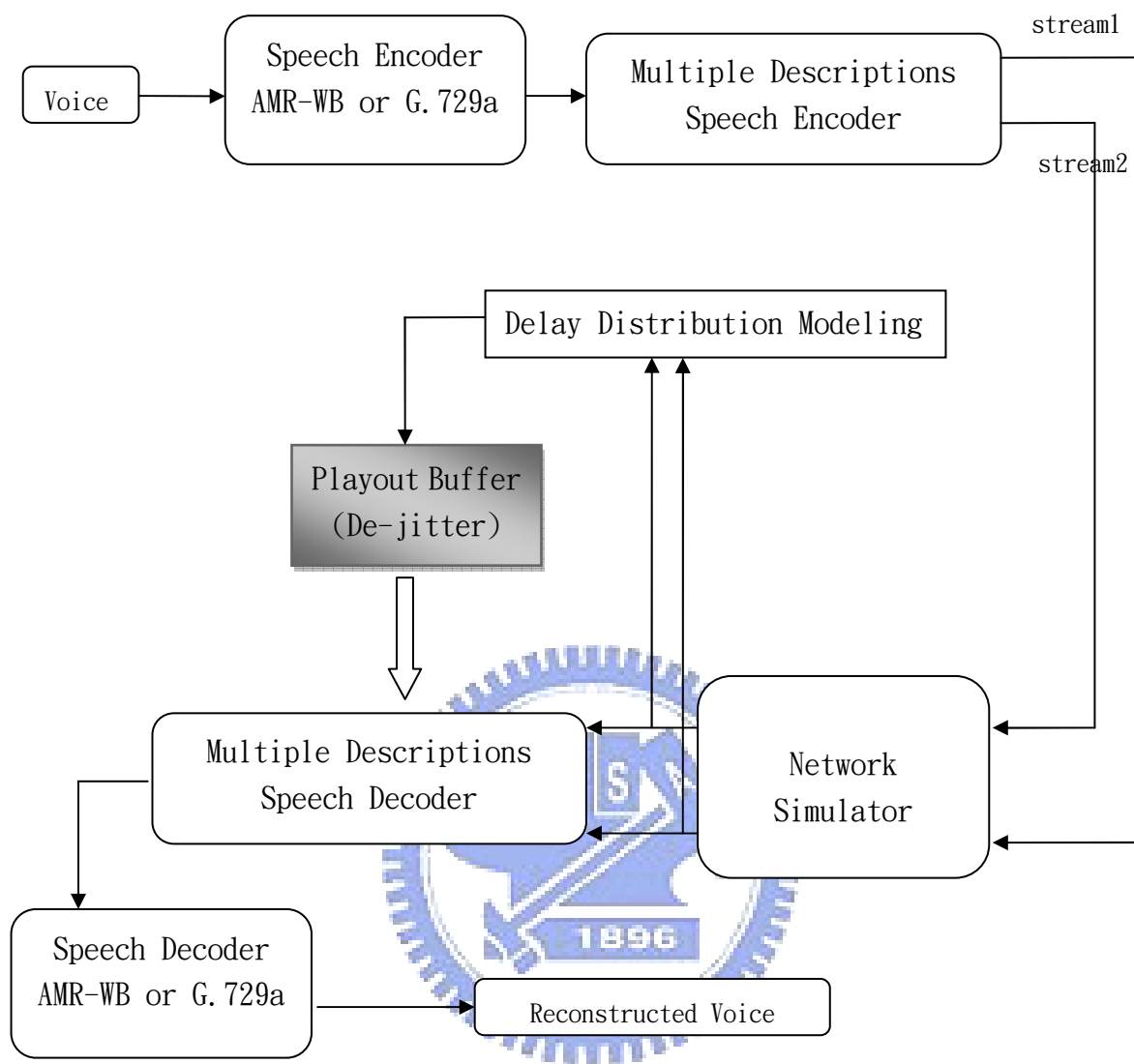


圖 3.4 MD 傳輸系統



## 第四章

### 結合前向錯誤控制的 MD 傳輸系統

在第三章我們提出 MD 傳輸播放排程演算法的設計，主要是針對晚到漏失的議題來做研究，以期依聽覺最佳化原則調整其安全因子 ( $\beta$ )。其結果是在合理的播放延遲設定下，有效地降低封包晚到漏失的機率，並利用 MD 編碼所提供的多重敘述傳輸，減輕了封包因網路漏失而造成的音質損害。而本章節將主要探討封包傳輸過程中遭遇的叢發性網路漏失(bursty network loss)問題，並將結合前向錯誤控制(Forward Error Control, FEC)與第三章的播放排程演算法，提出一種更強健性的 MD 語音傳輸系統。

#### 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 UDP 較適用於要求即時服務的互動式影音通訊，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。問題是這個機制並不適合應用於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，也大幅降低傳輸效率。

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。為了降低封包漏失率，前人發展出封包層級的前向錯誤控制碼(Packet-level FEC)的機制，主要是傳送端在傳遞一個語音封包前加上冗餘資訊，接收端則利用此資訊來做封包漏失的回復工作。必須注意的是，使用 FEC 的系統須同時傳送原始資料以冗餘資料，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。基於傳輸延遲以及計算複雜度之考量，冗餘資料量的增加額度有其限制。

表(4.1)整理了兩種方式的優缺點：

	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	需使用更多的頻寬傳送冗餘資訊
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 ARQ vs. FEC

## 4.2 前向錯誤控制

FEC 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、位元組層級、封包層級。

- 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，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。

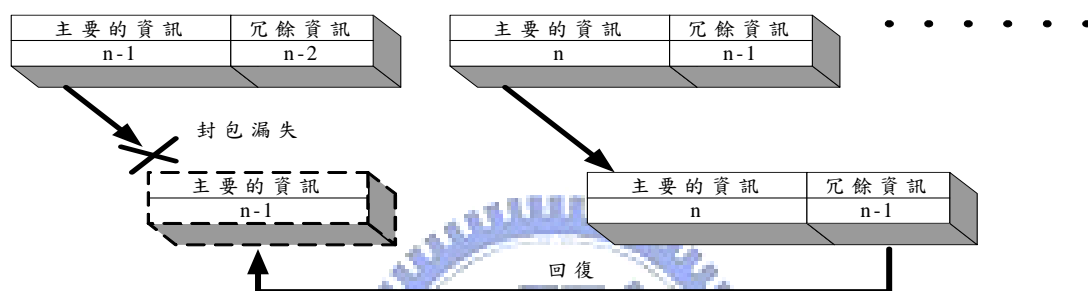


圖 4.1 media specific FEC

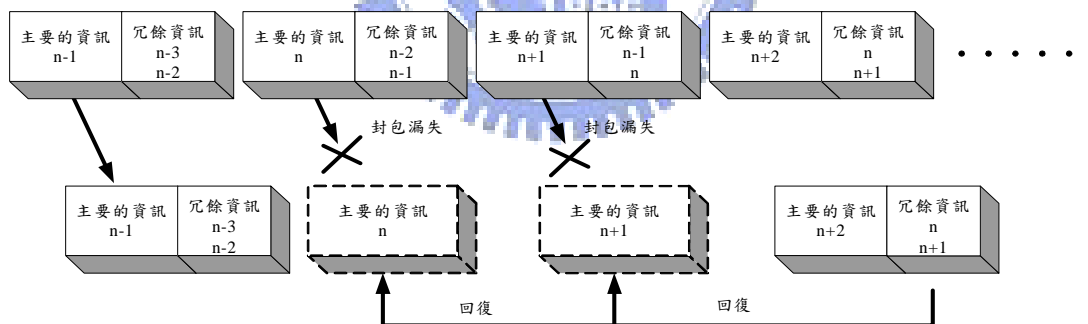


圖 4.2 冗餘資訊以多個封包產生

## [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，共同組合成 $N$ 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  $N$  個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。圖(4.4)以  $(5, 3)$  RS 碼為例子來說明封包經 FEC 機制後的傳送過程，其中  $T_p$  表示封包間隔的時間，則 FEC 區塊中第  $i$  個封包的額外延遲可表示為

$$D_{FEC,i} = (N - i) \cdot T_p \quad (4.1)$$

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。在此研究中，我們將鎖定在封包層級的前向錯誤控制，特別是採用 Reed-Solomon (RS) 編碼器。

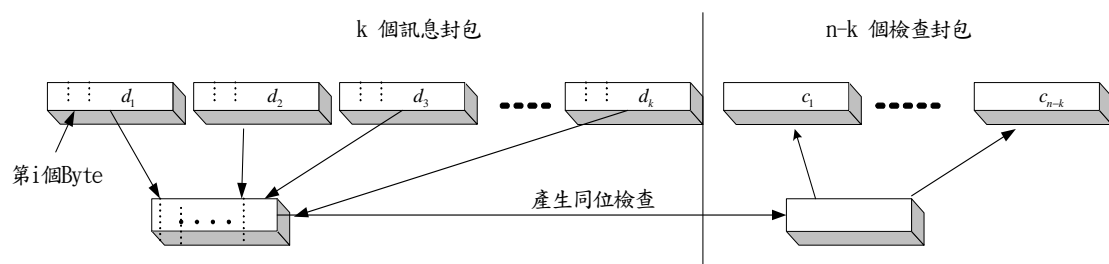


圖 4.3 RS 編碼過程

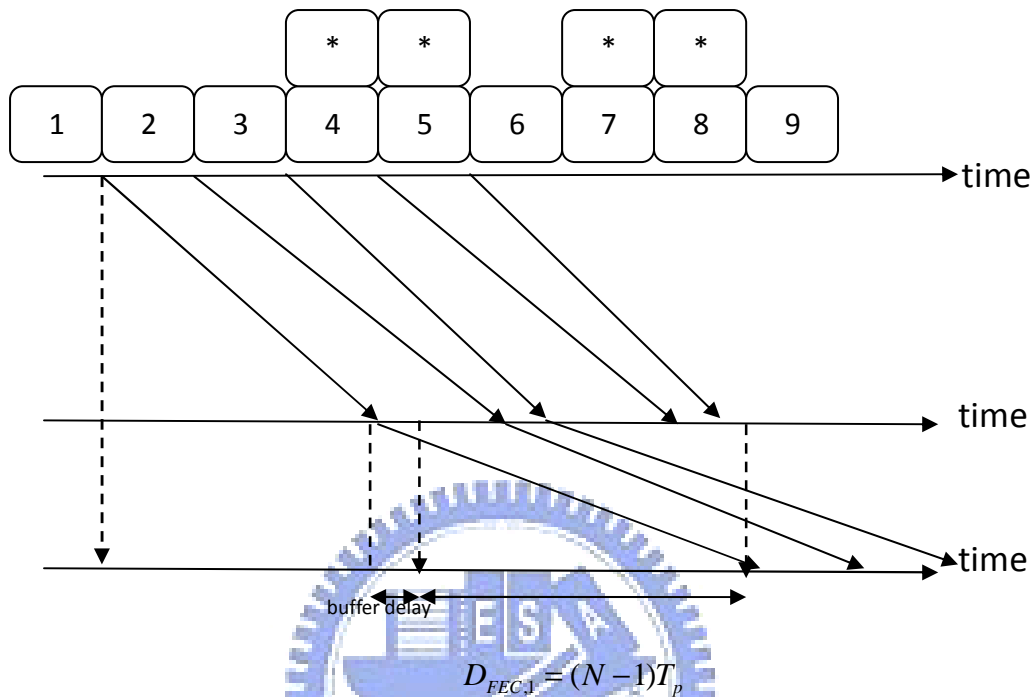


圖 4.4 (\*: redundant) FEC 機制下的封包傳送與接收

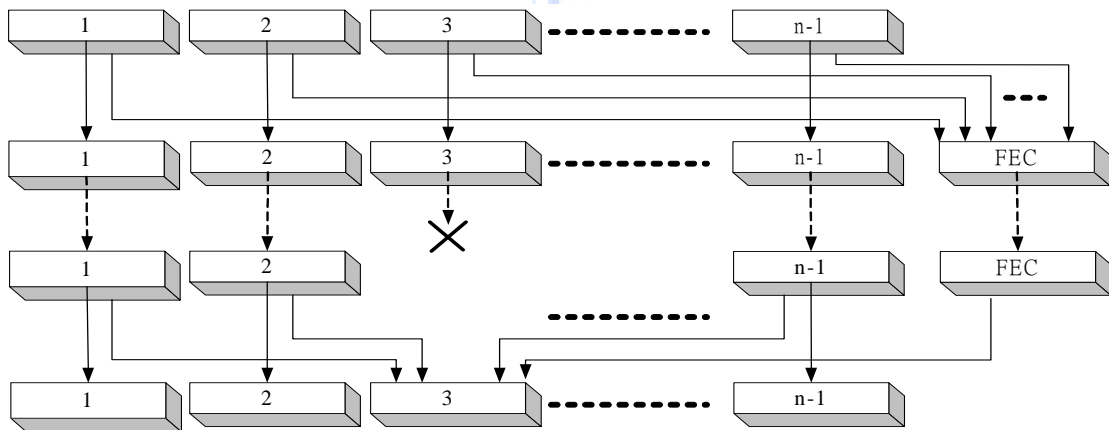


圖 4.5 同位碼回復錯誤的過程



### 4.3 結合 FEC 的 MD 傳輸系統

本節主要目標在於提出整合 FEC 和播放排程之適應性演算法，我們這裡選擇使用 Reed-Solomon(RS) Code 來當作封包層級的 FEC 機制。

#### 4.3.1 Reed-Solomon(RS)碼

RS(N, K)代表，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 個封包形成一個前向錯誤控制區塊(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  $\alpha$  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。當然最理想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[5]，假設  $p_u$  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

$$\text{少為 } PLR_{FEC} = p_u \left( 1 - \sum_{j=K}^{N-1} \binom{N-1}{j} (1-p_u)^j p_u^{N-1-j} \right) \quad (4.2)$$

上述式子可以解釋成在一個  $N$ -packet 的 FEC 區塊中，假如其中一個封包發生漏失機率 ( $p_u$ )，則此封包無法藉由 RS 碼回復的可能性就是在其餘  $N-1$  個封包之間，封包成功接收的數目少於  $K$  ( $\leq K-1$ )。換句話說，無法在掉一個封包之後，再正確接收到  $K$  個以上的封包。接下來的章節，將討論在叢發性漏失的網路情況下，RS 碼的回復情形。

### 4.3.2 叢發性封包漏失的回復機率推導

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。

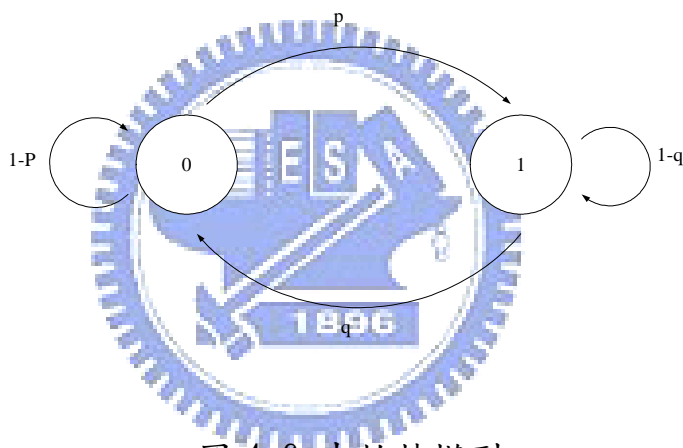


圖 4.6 吉伯特模型

狀態 1 與狀態 0 的穩態機率(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  $\pi_0 = \frac{q}{p+q}$ ， $\pi_1 = \frac{p}{p+q}$ 。特別強調地是，在

吉伯特模型假設下的原始封包漏失率為  $p_u = \pi_1 = \frac{p}{p+q}$ 。

接著依據吉伯特通道模型，我們引用[21]來定義一些機率分佈。首先

令  $p(i)$  表示掉了一個封包後，連續成功接收  $(i-1)$  個封包直到第  $i$  個封包發生漏失的機率：

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

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。接著定義  $R(m, n)$  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(圖(4.6))假設下， $p(i)$ 、 $P(i)$ 、 $q(i)$  與  $Q(i)$  可以推導得知

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

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

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

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



#### 4.4 整合漏失回復後的 MD 音質損害模型

本節將結合 4.2.2 的回復機率及第二章所提出的音質損害預測模型，透過參考文章[3]進而推導出 MD 傳輸中，封包漏失(包括網路漏失及晚到漏失)經 FEC 回復後的音質損害因子。為了有效區分不同串流的影響，我們將 4.2 節定義的機率附加下標以對應於串流  $s$ 。舉例而言，式子(4.8)至(4.11)定義的  $\{p_R(i), P_R(i), q_R(i), Q_R(i)\}$  將更改為

$$\{p_{R,s}(i), P_{R,s}(i), q_{R,s}(i), Q_{R,s}(i)\}$$

我們把焦點放在一個 N-packet FEC 區塊所屬  $K$  個語音封包(其餘  $N-K$  為 redundant)，其平均音質受損因子為：

$$I_{e,avg} = \frac{1}{K} \sum_{i=1}^K \sum_{j=1}^2 \rho_j(i) I_{e,j} \left( \prod_{s=1}^2 P_{FEC,s}(i) \right) \quad (4.12)$$

其中  $P_{FEC,s}(i)$  ( $s=1, 2$ ) 代表第  $s$  串流中，當封包發生晚到或網路漏失卻都無法由 FEC 回復的機率。若令  $p_s$  及  $q_s$  表示串流  $s$  所對應的吉伯特通道狀態轉移機率，則進一步可以寫成：

$$P_{FEC,s}(i) = \underbrace{\frac{p_s}{p_s + q_s}}_{\text{network loss}} (1 - P_{REC1,s}(i)) + \underbrace{\frac{q_s}{p_s + q_s}}_{\text{late loss}} (1 - F_{D,s}(D_{FEC,i})) (1 - P_{REC2,s}(i)) \quad (4.13)$$

在式(4.13)中，右邊的第一項代表封包  $i$  發生**網路漏失**卻無法經由 FEC 回復的機率，第二項則代表封包  $i$  發生**晚到漏失**卻無法經由 FEC 回復的機率。其中  $F_{D,s}(D_{FEC,i})$  代表封包  $i$  的網路延遲小於  $D_{FEC,i}$  的機率， $P_{REC1,s}(i)$  及  $P_{REC2,s}(i)$  分別代表串流  $s$  的第  $i$  個封包發生網路及晚到漏失後可經由 FEC 回復的機率。經由相關推導，我們證明這兩項機率

表示為：

$$P_{REC1,s}(i) = \sum_{L=1}^{N-K} \sum_{m=0}^{\min(L-1,i-1)} R_s'(m+1,i,D_{FEC,i}) R_s'(L-m,N-i+1,D_{FEC,i}) \quad (4.14)$$

$$P_{REC2,s}(i) = \sum_{L=1}^{N-K} \sum_{m=0}^{\min(L-1,i-1)} S_s'(i-m,i,D_{FEC,i}) S_s'(N-i-L+m+2,N-i+1,D_{FEC,i}) \quad (4.15)$$

其中  $R_s'(m,n,D_{FEC,i})$ 、 $\tilde{R}_s'(m,n,D_{FEC,i})$  是表示串流  $s$  中第  $i$  個封包發生網路漏失之後和之前的  $n-1$  個封包內有  $m-1$  個封包發生網路或晚到漏失的機率， $S_s'(m,n,D_{FEC,i})$ 、 $\tilde{S}_s'(m,n,D_{FEC,i})$  則代表接受到串流  $s$  中第  $i$  個封包其之後和之前的  $n-1$  個封包內接受到了  $m-1$  個封包的機率，進一步推導可得：

$$R_s'(m,n,D_{FEC,i}) = \begin{cases} P_{R,s}(n) \prod_{h=1}^{n-1} F_{D,s}(D_{FEC,i} - hT_p), & m=1, n \geq 1 \\ \sum_{j=1}^{n-m+1} \left\{ p_{R,s}(j) R_s'(m-1, n-j, D_{FEC,i} - jT_p) + P_{R,s}(j+1) (1 - F_{D,s}(D_{FEC,i} - jT_p)) \right. \\ \left. S_s'(n-j-(m-1)+1, n-j, D_{FEC,i} - jT_p) \right\} \prod_{h=1}^{j-1} F_{D,s}(D_{FEC,i} - hT_p) & \left. \right\}, 2 \leq m \leq n \end{cases} \quad (4.16)$$

$$S_s'(m,n,D_{FEC,i}) = \begin{cases} Q_{R,s}(n) + (1-p_s)^{n-1} \prod_{h=1}^{n-1} (1 - F_{D,s}(D_{FEC,i} - hT_p)), & m=1, n \geq 1 \\ \sum_{j=1}^{n-m+1} \left\{ q_{R,s}(j) + u(j-2)(1-p_s)^{j-1} \prod_{h=1}^{j-1} (1 - F_{D,s}(D_{FEC,i} - hT_p)) \right\} \\ F_{D,s}(D_{FEC,i} - hT_p) S_s'(m-1, n-j, D_{FEC,i}) & \left. \right\}, 2 \leq m \leq n \end{cases} \quad (4.17)$$

$$\tilde{R}_s'(m,n,D_{FEC,i}) = \begin{cases} P_{R,s}(n) \prod_{h=1}^{n-1} F_{D,s}(D_{FEC,i} + hT_p), & m=1, n \geq 1 \\ \sum_{j=1}^{n-m+1} \left\{ p_{R,s}(j) \tilde{R}_s'(m-1, n-j, D_{FEC,i} + jT_p) + P_{R,s}(j+1) (1 - F_{D,s}(D_{FEC,i} + jT_p)) \right. \\ \left. \tilde{S}_s'(n-j-(m-1)+1, n-j, D_{FEC,i} + jT_p) \right\} \prod_{h=1}^{j-1} F_{D,s}(D_{FEC,i} + hT_p) & \left. \right\}, 2 \leq m \leq n \end{cases}$$

(4.18)

$$\tilde{S}_s'(m, n, D_{FEC,i}) = \left\{ \begin{array}{l} Q_{R,s}(n) + (1-p_s)^{n-1} \prod_{h=1}^{n-1} (1 - F_{D,s}(D_{FEC,i} + hT_p)), \quad m=1, n \geq 1 \\ \sum_{j=1}^{n-m+1} \left\{ \begin{array}{l} \{q_{R,s}(j) + u(j-2)(1-p_s)^{j-1} \prod_{h=1}^{j-1} (1 - F_{D,s}(D_{FEC,i} + hT_p))\} \\ F_{D,s}(D_{FEC,i} + hT_p) \tilde{S}_s'(m-1, n-j, D_{FEC,i}) \end{array} \right\}, \quad 2 \leq m \leq n \end{array} \right.$$

(4.19)

where  $u(j-j_0) = \begin{cases} 1, & j \geq j_0 \\ 0, & j < j_0 \end{cases}$

式子(4.14)及(4.15)裡，其中第一個 summation 的上限(N-K)，是指在 FEC 區塊中，最多只能有 N-K 個封包不能被播放，換句話說最少要有 K 個封包能被播放，如此一來漏失(網路漏失或晚到漏失)的封包才能經由 FEC 回復，如圖(4.7)所示。

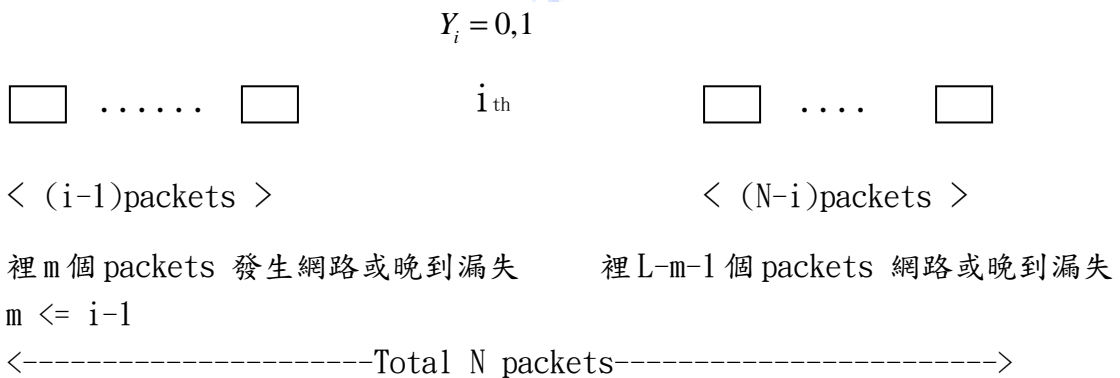


圖 4.7 FEC 回復情形



接下來我們將舉例說明式(4.14)及(4.15)的機率原理。

$D_{FEC,i}$  指的是 FEC 緩衝器中設定用來等待封包  $i$  的時間長度；由(4.1)式得知引入 FEC 保護封包傳輸時，將無形中增加  $(N-1)T_p$  的延遲時間，必須注意的是，由於封包是以間隔  $T_p$  依序傳送出去，所以這段  $(N-1)T_p$  時間就是用來等待 FEC 區塊中的第  $N$  個封包。換句話說，設定在 FEC 區塊中第  $i$  個封包的緩衝暫存器長度就是  $(N-i)T_p$ ，如圖(4.4)中，第二個封包及第一個 redundant 封包的抵達時間都超過了 FEC 區塊中所對應的等待時間。因此  $D_{FEC,i}$  可以表示成：

$$D_{FEC,i} = \hat{d}_i + \beta \hat{v}_i + (N-i)T_p, \quad i=1,2,\dots,N \quad (4.20)$$

$$\text{也就是 } D_{FEC,j} = D_{FEC,i} - (j-i)T_p, \quad j \geq i \quad (4.21)$$

為了詮釋  $P_{REC1,s}(i)$ 、 $P_{REC2,s}(i)$ 、 $R_s'(m,n,D_{FEC,i})$  及  $S_s'(m,n,D_{FEC,i})$  的物理意義，我們以例子來說明，從解釋  $P_{REC1,s}(i)$ 、 $P_{REC2,s}(i)$  開始，假設  $(N,K)=(7,4)$ ，並在串流  $s$  中第 3 個封包發生網路或晚到漏失的情況下 ( $i=3$ ) 進行推導其可以藉由 FEC 回復的機率(以下推導是以串流  $s$  中發生漏失的第 3 個封包為基準點，計算其前面及後面封包的接收情形)：

$$\begin{aligned}
P_{REC1,s}(3) &= \sum_{L=1}^{7-4} \sum_{m=0}^{\min(L-1,3-1)} \tilde{R}_s'(m+1,3,D_{FEC,3}) R_s'(L-m,7-3+1,D_{FEC,3}) \\
&= \sum_{L=1}^3 \sum_{m=0}^{\min(L-1,2)} \tilde{R}_s'(m+1,3,D_{FEC,3}) R_s'(L-m,5,D_{FEC,3})
\end{aligned}$$

$\Rightarrow L=1:$

$$\begin{aligned}
&\sum_{m=0}^{\min(0,2)} \tilde{R}_s'(m+1,3,D_{FEC,3}) R_s'(1-m,5,D_{FEC,3}) \\
&= \tilde{R}_s'(1,3,D_{FEC,3}) R_s'(1,5,D_{FEC,3}) // \Pr(\text{前面2個封包漏失0個}) \Pr(\text{後面4個封包漏失0個})
\end{aligned}$$

$L=2:$

$$\begin{aligned}
&\sum_{m=0}^{\min(1,2)} \tilde{R}_s'(m+1,3,D_{FEC,3}) R_s'(2-m,5,D_{FEC,3}) \\
&= \tilde{R}_s'(1,3,D_{FEC,3}) R_s'(2,5,D_{FEC,3}) // \Pr(\text{前面2個封包漏失0個}) \Pr(\text{後面4個封包漏失1個}) \\
&+ \tilde{R}_s'(2,3,D_{FEC,3}) R_s'(1,5,D_{FEC,3}) // \Pr(\text{前面2個封包漏失1個}) \Pr(\text{後面4個封包漏失0個})
\end{aligned}$$

$L=3:$

$$\begin{aligned}
&\sum_{m=0}^{\min(2,2)} \tilde{R}_s'(m+1,3,D_{FEC,3}) R_s'(3-m,5,D_{FEC,3}) \\
&= \tilde{R}_s'(1,3,D_{FEC,3}) R_s'(3,5,D_{FEC,3}) // \Pr(\text{前面2個封包漏失0個}) \Pr(\text{後面4個封包漏失2個}) \\
&+ \tilde{R}_s'(2,3,D_{FEC,3}) R_s'(2,5,D_{FEC,3}) // \Pr(\text{前面2個封包漏失1個}) \Pr(\text{後面4個封包漏失1個}) \\
&+ \tilde{R}_s'(3,3,D_{FEC,3}) R_s'(1,5,D_{FEC,3}) // \Pr(\text{前面2個封包漏失2個}) \Pr(\text{後面4個封包漏失0個})
\end{aligned}$$

在這7 - packet的FEC區塊中，全部最多只能有3個封包發生漏失，  
這樣一來FEC才有辦法回復已經發生網路漏失的第3個封包。

$$\begin{aligned}
P_{REC2,s}(3) &= \sum_{L=1}^{7-4} \sum_{m=0}^{\min(L-1,3-1)} \tilde{S}_s'(3-m,3,D_{FEC,3}) S_s'(7-3-L+m+2,7-3+1,D_{FEC,3}) \\
&= \sum_{L=1}^3 \sum_{m=0}^{\min(L-1,2)} \tilde{S}_s'(3-m,3,D_{FEC,3}) S_s'(6-L+m,5,D_{FEC,3})
\end{aligned}$$

$\Rightarrow L=1$ :

$$\begin{aligned}
&\sum_{m=0}^{\min(0,2)} \tilde{S}_s'(3-m,3,D_{FEC,3}) S_s'(5+m,5,D_{FEC,3}) \\
&= \tilde{S}_s'(3,3,D_{FEC,3}) S_s'(5,5,D_{FEC,3})
\end{aligned}$$

// Pr(前面2個封包有2個成功接收) Pr(後面4個封包4個成功接收)

= Pr(前面2個封包漏失0個) Pr(後面4個封包漏失0個)

$L=2$ :

$$\begin{aligned}
&\sum_{m=0}^{\min(1,2)} \tilde{S}_s'(3-m,3,D_{FEC,3}) S_s'(4+m,5,D_{FEC,3}) \\
&= \tilde{S}_s'(3,3,D_{FEC,3}) S_s'(4,5,D_{FEC,3})
\end{aligned}$$

// Pr(前面2個封包有2個成功接收) Pr(後面4個封包3個成功接收)

= Pr(前面2個封包漏失0個) Pr(後面4個封包漏失1個)

$$+ \tilde{S}_s'(2,3,D_{FEC,3}) S_s'(5,5,D_{FEC,3})$$

// Pr(前面2個封包有1個成功接收) Pr(後面4個封包4個成功接收)

= Pr(前面2個封包漏失1個) Pr(後面4個封包漏失0個)

$L=3$ :

$$\begin{aligned}
&\sum_{m=0}^{\min(2,2)} \tilde{S}_s'(3-m,3,D_{FEC,3}) S_s'(3+m,5,D_{FEC,3}) \\
&= \tilde{S}_s'(3,3,D_{FEC,3}) S_s'(3,5,D_{FEC,3})
\end{aligned}$$

// Pr(前面2個封包有2個成功接收) Pr(後面4個封包2個成功接收)

= Pr(前面2個封包漏失0個) Pr(後面4個封包漏失2個)

$$+ \tilde{S}_s'(2,3,D_{FEC,3}) S_s'(4,5,D_{FEC,3})$$

// Pr(前面2個封包有1個成功接收) Pr(後面4個封包3個成功接收)

= Pr(前面2個封包漏失1個) Pr(後面4個封包漏失1個)

$$+ \tilde{S}_s'(1,3,D_{FEC,3}) S_s'(5,5,D_{FEC,3})$$

// Pr(前面2個封包有0個成功接收) Pr(後面4個封包4個成功接收)

= Pr(前面2個封包漏失1個) Pr(後面4個封包漏失0個)

在這7-*packet*的FEC區塊中，全部最多只能有3個封包發生漏失，這樣一來FEC才有辦法回復已經發生晚到漏失的第3個封包。

接下來推導  $R_s'(m, n, D_{FEC,i})$  及  $S_s'(m, n, D_{FEC,i})$ ，假設  $n=5, m=3$ ：

$$R_s'(3, 5, D_{FEC,i}) = \sum_{j=1}^{5-3+1} \left\{ \begin{aligned} & \{ p_{R,s}(j) R_s'(3-1, 5-j, D_{FEC,i} - jT_p) + P_s(j+1)(1 - F_{D,s}(D_{FEC,i} - jT_p)) \} \\ & \left\{ S_s'(5-j-(3-1)+1, 5-j, D_{FEC,i} - jT_p) \right\} \prod_{h=1}^{j-1} F_{D,s}(D_{FEC,i} - hT_p) \end{aligned} \right\}$$

$$= p_{R,s}(1) R_s'(2, 4, D_{FEC,i} - T_p) + P_s(2)(1 - F_{D,s}(D_{FEC,i} - T_p)) S_s'(3, 4, D_{FEC,i} - T_p)$$

//Pr(1|1) Pr(接下來 3個封包中有1個發生"1"或"Δ")+Pr(Δ|1)Pr(接下來 3封包中有1個發生"1"或"Δ")

$$+ p_{R,s}(2) F_{D,s}(D_{FEC,i} - T_p) R_s'(2, 3, D_{FEC,i} - 2T_p)$$

//Pr(01|1) Pr(接下來 2個封包中有1個發生"1"或"Δ")

$$+ P_s(3) F_{D,s}(D_{FEC,i} - T_p) (1 - F_{D,s}(D_{FEC,i} - 2T_p)) S_s'(2, 3, D_{FEC,i} - 2T_p)$$

//Pr(0Δ|1)Pr(接下來 2個封包中有1個發生"1"或"Δ")

$$+ p_{R,s}(3) F_{D,s}(D_{FEC,i} - T_p) F_{D,s}(D_{FEC,i} - 2T_p) R_s'(2, 2, D_{FEC,i} - 3T_p)$$

//Pr(001|1)Pr(接下來 1個封包中有1個發生"1"或"Δ")

$$+ P_s(4) F_{D,s}(D_{FEC,i} - T_p) F_{D,s}(D_{FEC,i} - 2T_p) (1 - F_{D,s}(D_{FEC,i} - 3T_p)) S_s'(1, 2, D_{FEC,i} - 3T_p)$$

//Pr(00Δ|1)Pr(接下來 1個封包中有1個發生"1"或"Δ")

註:其中Δ代表對應的封包晚到漏失,1代表網路漏失,0代表成功到達

$$S_s'(3, 5, D_{FEC,i}) = \sum_{j=1}^{5-3+1} \left\{ \begin{aligned} & \{ q_{R,s}(j) + u(j-2)(1-p_s)^{j-1} \prod_{h=1}^{j-1} (1 - F_{D,s}(D_{FEC,i} - hT_p)) \} \\ & F_{D,s}(D_{FEC,i} - hT_p) S_s'(3-1, 5-j, D_{FEC,i}) \end{aligned} \right\}$$

$$= q_{R,s}(1) F_{D,s}(D_{FEC,i} - T_p) S_s'(2, 4, D_{FEC,i} - T_p)$$

//Pr(1|0) Pr(接下來 3個封包中有1個非"1"且非"Δ")

$$+ q_{R,s}(2) F_{D,s}(D_{FEC,i} - 2T_p) S_s'(2, 3, D_{FEC,i} - 2T_p)$$

//Pr(10|0) Pr(接下來 2個封包中有1個非"1"且非"Δ")

$$+ (1-p_s)^2 (1 - F_{D,s}(D_{FEC,i} - T_p)) (1 - F_{D,s}(D_{FEC,i} - 2T_p)) S_s'(2, 3, D_{FEC,i} - 2T_p)$$

//Pr(Δ0|0) Pr(接下來 2個封包中有1個非"1"且非"Δ")

$$+ q_{R,s}(3) F_{D,s}(D_{FEC,i} - 3T_p) S_s'(2, 2, D_{FEC,i} - 3T_p)$$

//Pr(110|0)Pr(接下來1個封包中有1個非"1"且非"Δ")

$$+ (1-p_s)^3 (1 - F_{D,s}(D_{FEC,i} - T_p)) (1 - F_{D,s}(D_{FEC,i} - 2T_p)) F_{D,s}(D_{FEC,i} - 3T_p) S_s'(2, 2, D_{FEC,i} - 3T_p)$$

//Pr(ΔΔ0|0)Pr(接下來1個封包中有1個非"1"且非"Δ")

以上分別推導當特定封包發生網路及晚到漏失時，其可藉由 FEC 回復的機率 ( $P_{REC1,s}(i)$  及  $P_{REC2,s}(i)$ )，以及當特定封包發生漏失或晚到漏失時，其之後 4 個封包裡分別漏失掉及接受到 2 個封包的機率 ( $R_s'(m,n,D_{FEC,i})$  及  $S_s'(m,n,D_{FEC,i})$ )。從上述數學式子可以看出後者是以疊代的形式代表著其所有可能的情形，並將其全部所對應的機率相加。此外  $\tilde{R}_s'(m,n,D_{FEC,i})$ 、 $\tilde{S}_s'(m,n,D_{FEC,i})$  可由式子(4.16)至(4.19)發現其與  $R_s'(m,n,D_{FEC,i})$ 、 $S_s'(m,n,D_{FEC,i})$  之間的相似對應關係，兩者推導方式相似，因此不再贅述。

最後反應 MD 解碼後音質的  $\rho_j(i)$  其對應的式子如(4.21):

$$\begin{aligned} \rho_1(i) &= \Pr(\Omega_1 | \Omega_1 \cup \Omega_2) \\ &= \frac{\text{封包於兩條串流成功接收的機率}(\Pr(\Omega_1, \Omega_1 \cup \Omega_2))}{\text{音框可以播放的機率}(\Pr(\Omega_1 \cup \Omega_2))} = \frac{\prod_{s=1}^2 (1 - P_{FEC,s}(i))}{1 - \prod_{s=1}^2 P_{FEC,s}(i)} \\ \rho_2(i) &= 1 - \rho_1(i) \end{aligned} \tag{4.21}$$

## 4.5 MD 封包傳輸系統的最佳化設計

在 4.3 節，我們結合 MD 的音質損害模型及封包漏失回復機率而得出式(4.12)所示的整體音質損害。接著，我們據以提出適應性整合 FEC 及 MD 播放排程的目標函數(criterion)。整體系統的最佳化設計問題描述如下：

$$\begin{aligned}
 &\text{Given:} && p_s, q_s, F_{D,s}(d) \\
 &\text{maximize:} && R = 94.2 - I_{e,avg} - I_d(\hat{d}_{i,s} + \beta \hat{v}_{i,s} + (N-1)T_p + dc) \\
 &\text{over all:} && (N, K), \beta, s = \{1, 2\} \\
 &\text{subject to:} && \begin{cases} \frac{N}{K} \times MD \text{ coding gain} < 2 \\ K \geq \text{The number of packets of next talkspurt} \end{cases}
 \end{aligned} \tag{4.22}$$

其中 FEC 碼率的限制， $(\frac{N}{K} \times MD \text{ coding gain} < 2)$  是為了讓傳輸頻寬更有效率。此最佳化演算法是從接收到每個 talkspurt 的最後一個封包接收之後開始進行，事先記錄該封包之前 L 個封包實際量測所得的網路延遲與封包網路漏失狀態。網路延遲記錄可用以建立其 Pareto 分佈模型，封包網路漏失狀態也可被用來估測其吉伯特通道模型參數  $(p, q)$ 。依據此 MD 傳輸過程的動態網路變動情形，在 talkspurt 之間尋找能夠使其音質達到最佳狀態的系統參數  $(N, K, \beta)$ 。 $(N, K)$  將用在下一個 talkspurt 傳送端的 FEC 機制，而同時等待接受下一個 talkspurt 的接收端，也依  $\beta$  值將第 i 個封包的 FEC 緩衝延遲調整為  $D_{FEC,i} = \hat{d}_i + \beta \hat{v}_i + (N-i)T_p, i=1 \sim N$ ，播放延遲(playout delay)設定為

$D = \hat{d}_i + \beta \hat{v}_i + (N-1)T_p$ ，以及整體延遲  $D_{m2e} = D + dc$ 。最後，適應性整合

FEC 並調整播放延遲的 MD 傳輸系統如下(圖(4.8))所示：

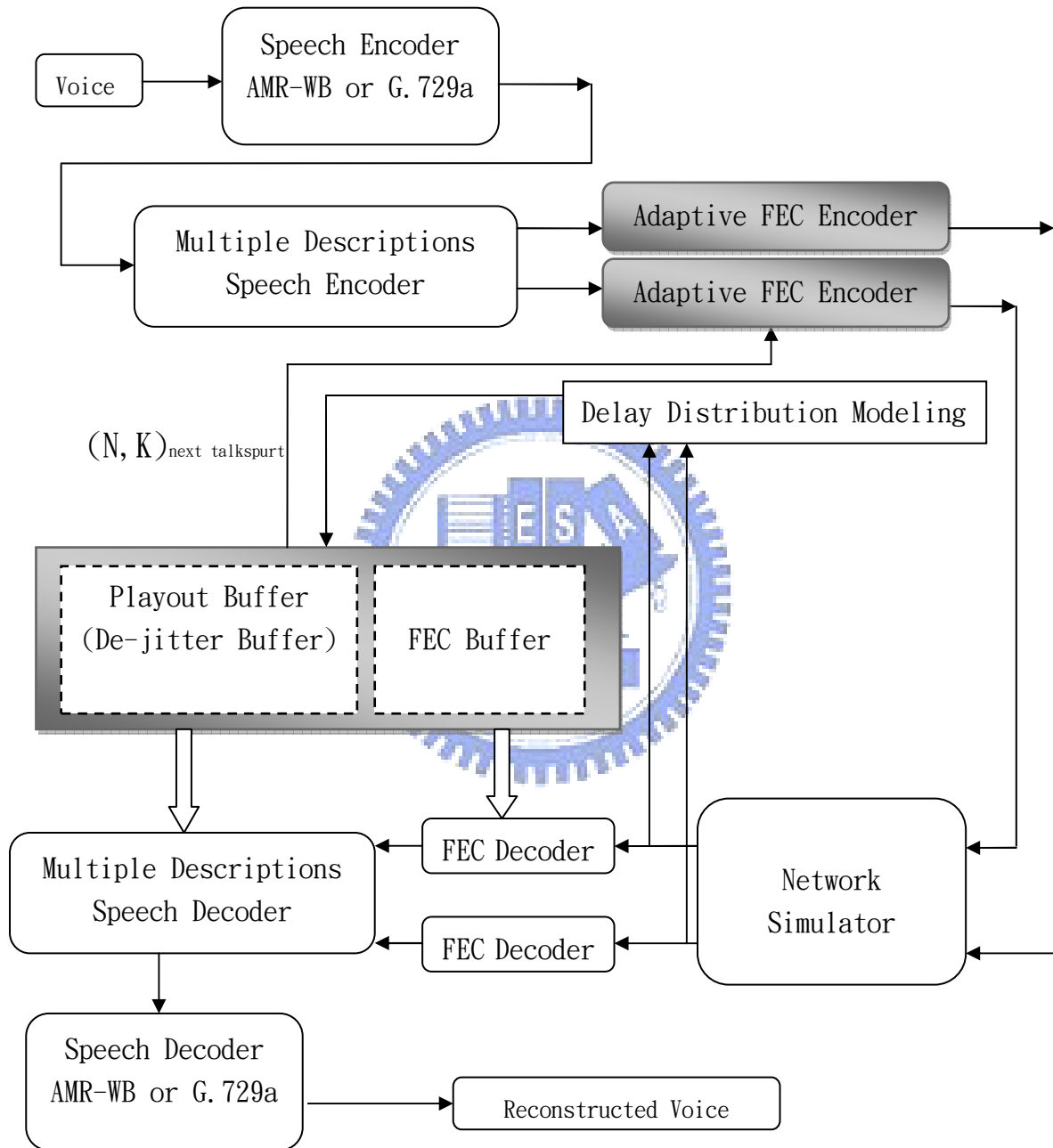


圖 4.8 結合 FEC 的 MD 傳輸系統

# 第五章

## 實驗結果

本章節將以系統模擬驗證前面章節所提出的演算法，首先以語音檔實作方式驗證第二章所提出的  $I_e$  音質預測模型，接著透過網路模擬器(Network Simulator 2, NS2)模擬封包網路延遲，及應用 G.729a 語音編碼標準來實作本論文所提出的 MD 傳輸系統，並探討前面章節所提到的 MD 播放排程演算法之間的優劣。

### 5.1 $I_e$ 音質預測模型的驗證

實驗參數

語音檔 (wav)	內容
M10_8k	A moth zig-zagged along the path through Otto's garden.
Female_9	The big dog loved to chew on the old rag doll.
Male_9	Contrast trim provides other touches of color.
Female_10	The overweight charmer could slip poison into anyone's tea.

表 5.1 語音檔內容

#### ●實驗一：驗證新的 $I_e$ 模型

為了驗證新的音質損害因子模型( $I_e(e)$ )能較準確反映實際音質，我們使用 4 個語音檔(如表(5.1)所示)，並人工給定通道環境及接收情況，透過 PESQ 音質量測與提出的音質損害因子預測模型做比較。

其實驗結果如下表：



語音檔 (經 G. 729a 或 AMR-WB 編碼)	實驗設定	實驗結果( $I_e$ )		
		傳統模型預測出的損害因子	實際量測出的損害因子	
M10_8k.wav 經由 G. 729a 編碼後產生 330 個音框	不能播放:40 個音框 (FER $\sim$ 0.12)	40.3689	46.2651	
	收到兩條串流:200 個音框 只收到其中一條串流:90 個音框	45.6725		
	不能播放:16 個音框 (FER $\sim$ 0.05)	31.7762	36.5710	
	收到兩條串流:270 個音框 只收到其中一條串流:44 個音框	34.9678		
	Female_9.wav 經由 G. 729a 編碼後產生 338 個音框	不能播放:34 個音框 (FER $\sim$ 0.12)	39.4735	44.1399
		收到兩條串流:260 個音框 只收到其中一條串流:44 個音框	42.0227	
不能播放:50 個音框 (FER $\sim$ 0.15)		42.8513	47.7964	
收到兩條串流:260 個音框 只收到其中一條串流:28 個音框		44.3834		
Male_9.wav 經由 AMR-WB 編碼後產生 162 個音框		不能播放:8 個音框 (FER $\sim$ 0.05)	34.4057	40.8426
		收到兩條串流:120 個音框 只收到其中一條串流:34 個音框	39.4766	
	不能播放:16 個音框 (FER $\sim$ 0.10)	43.1581	44.5381	
	收到兩條串流:136 個音框 只收到其中一條串流:10 個音框	44.3455		

Female_10.wav 經由 AMR-WB 編碼後產生 215 個音框	不能播放:17 個音框 (FER $\approx$ 0.08) 收到兩條串流:100 個音框 只收到其中一條串流:98 個音框	40.0454	49.1639
		49.5018	
	不能播放:10 個音框 (FER $\approx$ 0.05) 收到兩條串流:180 音框 只收到其中一條串流:25 個音框	34.4057	39.3969
		37.1761	

表 5.2 音質預測模型的驗證

### 實驗結果分析:

實驗中每個語音檔都設定不同的封包漏失及接收情形，並將量測其音質損害因子，然後與傳統的音質損害預測模型和我們所提出的模型做比較。如同第二章尋找新  $I_e$  音質損害預測模型過程中的步驟一樣，其中設定不能播放的封包及只收到其中一條串流的封包皆各有 30 組不一樣的指定位置，最後再對其所量測出來的  $I_e$  取平均，這是為了抵銷掉封包位置不同所帶來音質上的差異，以求更精準的實驗結果。至於  $\Omega_2$  中音框收到哪一條串流也是隨機指定的。

由表(5.2)可看出相較於傳統  $I_e$  音質損害預測模型而言，新的  $I_e$  音質損害預測模型能夠較精準地預測實際的音質情況，那是因為新  $I_e$  模型考量了  $\Omega_1$  及  $\Omega_2$  兩種情況的音質也反應出其之間的不同。而假如我們使用傳統  $I_e$  預測模型來預測的話，會因為  $\Omega_1$  及  $\Omega_2$  皆被判定為"可播放"也就是沒有"漏失掉"，所以當在計算最後接收端 FER 時，很容

易將  $\Omega_1$  及  $\Omega_2$  兩種情況混為一談。而由實驗數據可清楚得知，如此的預測是過於樂觀、不切實際的。

### ●實驗二: 語音檔經 MD 和 SD 傳輸之下的音質比較

為了更加凸顯 MD 語音傳輸系統在降低封包漏失所造成的音質損上，比單一串流(Single Description, SD)傳輸更具優勢，我們循著實驗一的實驗模式，以實際對語音檔編碼並設定其封包漏失情況來呈現單一與多重串流傳輸音質上的差異。簡單來說，實驗目的是為了說明使用 SD 編碼傳輸時，發生封包漏失的音質損害與應用 MD 編碼的傳輸系統中，收到其中一條串流並以其來重建封包後的音質損害比較。

語音檔 (經 G. 729a 或 AMR-WB 編碼)	傳輸模式	實驗設定	實驗結果 ( $I_e$ )
M10_8k. wav 經由 G. 729a 編碼後產生 330 個音框	SD	不能播放:10 個音框 收到 320 個音框	26. 5029
		不能播放:50 個音框 收到 290 個音框	41. 3721
	MD	只收到其中一條串流: 10 個音框 兩條串流接收到:320 個音框	23. 7652
		只收到其中一條串流: 50 個音框 兩條串流接收到:290 個音框	32. 4399
Female_9. wav 經由 G. 729a 編碼後產生 338 個音框	SD	不能播放:20 個音框 收到 318 個音框	36. 7851
		不能播放:60 個音框 收到 278 個音框	47. 2325
	MD	只收到其中一條串流: 20 個音框 兩條串流接收到:318 個音框	28. 8520
		只收到其中一條串流: 60 個音框 兩條串流接收到:278 個音框	34. 9023

Male_9. wav 經由 AMR-WB 編碼後產生 162 個音框	SD	不能播放:5 個音框 收到 157 個音框	31.4887
		不能播放:10 個音框 收到 152 個音框	36.4953
	MD	只收到其中一條串流: 5 個音框 兩條串流接收到:157 個音框	20.3953
		只收到其中一條串流: 10 個音框 兩條串流接收到:152 個音框	23.2604
Female_10. wav 經由 AMR-WB 編碼後產生 215 個音框	SD	不能播放:15 個音框 收到 200 個音框	38.5671
		不能播放:20 個音框 收到 195 個音框	42.3488
	MD	只收到其中一條串流: 15 個音框 兩條串流接收到:200 個音框	27.9583
		只收到其中一條串流: 20 個音框 兩條串流接收到:195 個音框	29.8418

表 5.3 音框經由 MD 與 SD 傳輸效能比較

### 實驗結果分析:

實驗二循著實驗一的做法，利用 30 組不一樣的位置來指定音框漏失其中一條串流。由表(5.3)可以看出，相對 SD 傳輸來說，MD 編碼傳輸的好處在於當封包漏失之後，接收端還能以僅有的部份資訊來重建音框。換句話說，只有當兩條串流都發生漏失時，MD 解碼才會啟動錯誤隱匿機制。以上兩項實驗一及實驗二皆著重於編碼音質損害及封包漏失音質損害並未考量延遲所造成的音質損害( $I_d$ )。

## 5.2 MD 播放排程演算法

### 1. 網路環境:

模擬 MD 傳輸的網路延遲使用表(5.4)中所有路徑，SD 傳輸則使用 Trace1。packet loss rate 從 0%到 15%，每遞增改變 3%做一次實驗。

路徑	平均	變異數(variance)
Trace 1	101.5431	14.4806
Trace 2	121.4828	19.0436
Trace 3	201.5431	14.4806
Trace 4	221.4828	19.0436

表 5.4 網路延遲(單位:ms)

### 2. 語音檔:

使用 Female\_8.wav("Ran away on a black night with a lawful wedded man")，經由 G.729a 編碼後產生 272 個音框，其中有 8 段 talkspurt 共含有 196 個音框，其餘 76 個音框為 silence。

## ● 實驗一：固定和動態設計的安全因子效能比較

本實驗主要呈現播放排程演算法中，隨著網路傳輸環境變化來動態設計安全因子的優點。

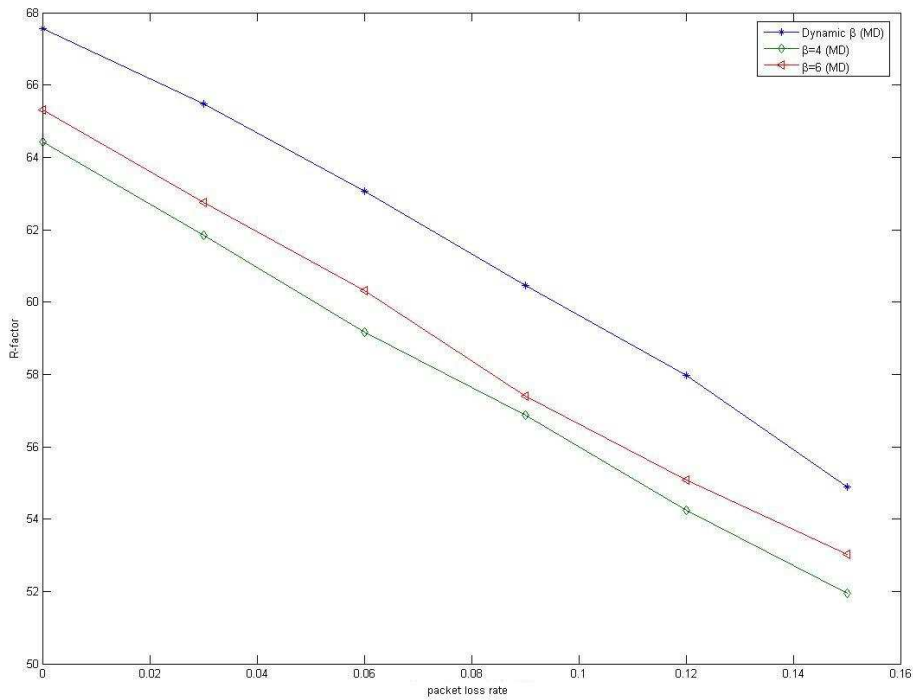


圖 5.1 固定與動態調整  $\beta$  效能比較

	0%	3%	6%	9%	12%	15%
Dynamic $\beta$ (MD)	0.9551	0.9020	0.8516	0.8010	0.7542	0.7089
$\beta=4$ (MD)	0.8580	0.8111	0.7639	0.7231	0.6801	0.6418
$\beta=6$ (MD)	0.8806	0.8311	0.7880	0.7401	0.7005	0.6628

表 5.5 收到兩條串流比例(Prob{ $\Omega_1$ })的比較(固定與動態調整  $\beta$ )

	0%	3%	6%	9%	12%	15%
Dynamic $\beta$ (MD)	137.568	137.572	137.580	137.583	137.595	137.589
$\beta=4$ (MD)	134.159	134.180	134.202	134.220	134.241	134.253
$\beta=6$ (MD)	136.174	136.196	136.217	136.241	136.271	136.282

表 5.6 整體延遲的比較(固定與動態調整  $\beta$ )

### 實驗結果分析：

由圖(5.1)可以明顯看出，固定  $\beta$  值的設計方法表現比較差，這是由於一旦傳輸環境發生變化，每個 talkspurt 所需的播放延遲將有所不同，因此設定固定的  $\beta$  值，會造成額外或過短的播放延遲。而隨著網路傳輸環境動態設計安全因子的優勢，主要在於動態設計  $\beta$  值是隨著環境變化來調整每個 talkspurt 播放延遲，因此能夠有效降低封包晚到漏失的機率也能避免過長的緩衝時間而造成的音質損害。表(5.5)及(5.6)也可顯示出動態設計  $\beta$  值不僅避免接收端設定額外的整體延遲也可以兼顧音框回復品質，使得最後在接收端音框收到兩條串流的比例比固定  $\beta$  值的設計還多。總而言之，有效率地調整播放延遲並兼顧音框重建品質，是動態設計安全因子的傳輸系統最大的優勢。

● 實驗二: 以不同考量(低延遲或音質)設計的 MD 播放排程比較

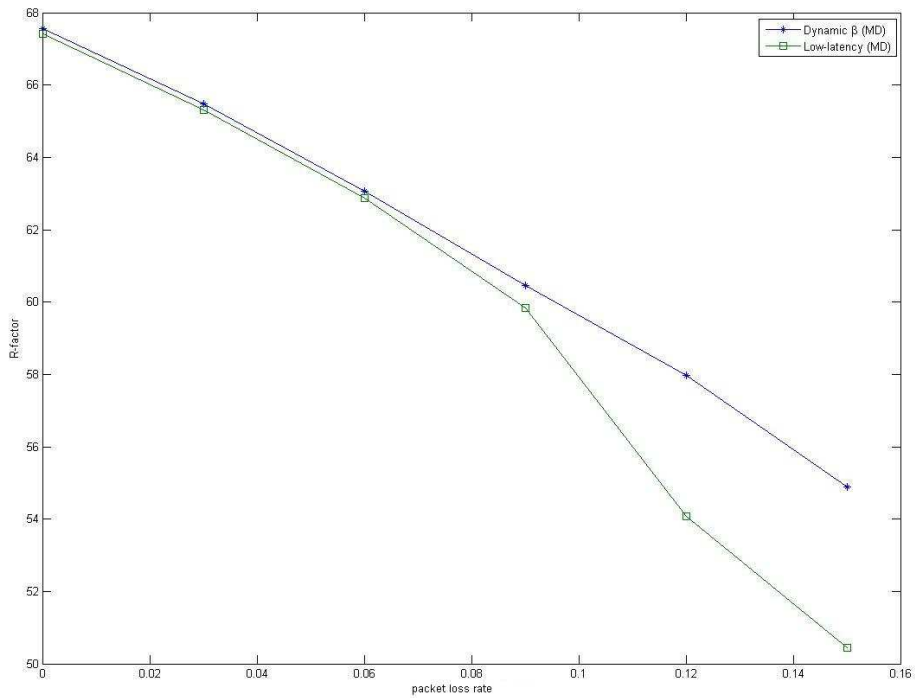


圖 5.2 MD 播放排程演算法效能比較



	0%	3%	6%	9%	12%	15%
音質最佳化	0.9551	0.9020	0.8516	0.8010	0.7542	0.7089
低延遲 ( $\lambda_1, \lambda_2$ ) = $(10^3, 10^7)$	0.9510	0.8968	0.8476	0.7960	0.7416	0.6930

表 5.7 收到兩條串流比例(Prob{ $\Omega_1$ })的比較(低延遲或音質考量)



	0%	3%	6%	9%	12%	15%
音質最佳化	137.568	137.572	137.580	137.583	137.595	137.589
低延遲( $\lambda_1, \lambda_2$ ) = $(10^3, 10^7)$	138.654	138.628	138.613	138.590	138.530	138.475

表 5.8 整體延遲的比較(低延遲或音質考量)

### 實驗結果分析:

本實驗主要探討分別以低延遲與音質最佳化考量的兩種 MD 播放延遲演算法的比較，由實驗數據可以看出以低延遲為考量的演算法表現較不理想，這是由於此演算法較著重於延遲造成的音質損害，但卻未將音框重建品質納入考量。所以設計出的播放延遲可能僅足夠收到其中一條串流，因此收到兩條串流的比例較低(如表(5.6)所示)，這使得接收端過度依賴 MD 回復，因此音框重建品質較差。另外，必須先透過調整參數再來設計播放延遲的做法，也缺乏適應性調整系統傳輸的能力。反觀以音質最佳化為考量的設計將音框重建品質做區隔( $\Omega_1, \Omega_2$ )，並將其納入整體延遲及 frame erasure rate 的 trade-off 中，不僅能使接收端播放較佳的音框品質，也能隨著網路傳輸變化適應性調整每個 talkspurt 的播放延遲。此外，表(5.7)顯示在( $\lambda_1, \lambda_2$ )= $(10^3, 10^7)$ 設定下，在接收端低延遲演算法所設計的整體延遲不僅沒有比較低，反而比音質最佳化考量的演算法來得長，但是設定較長的播放延遲卻也沒反應出較高的  $\text{Prob}\{\Omega_1\}$ 。

### ●實驗三:相同編碼效率下，MD 與 SD 傳輸的比較

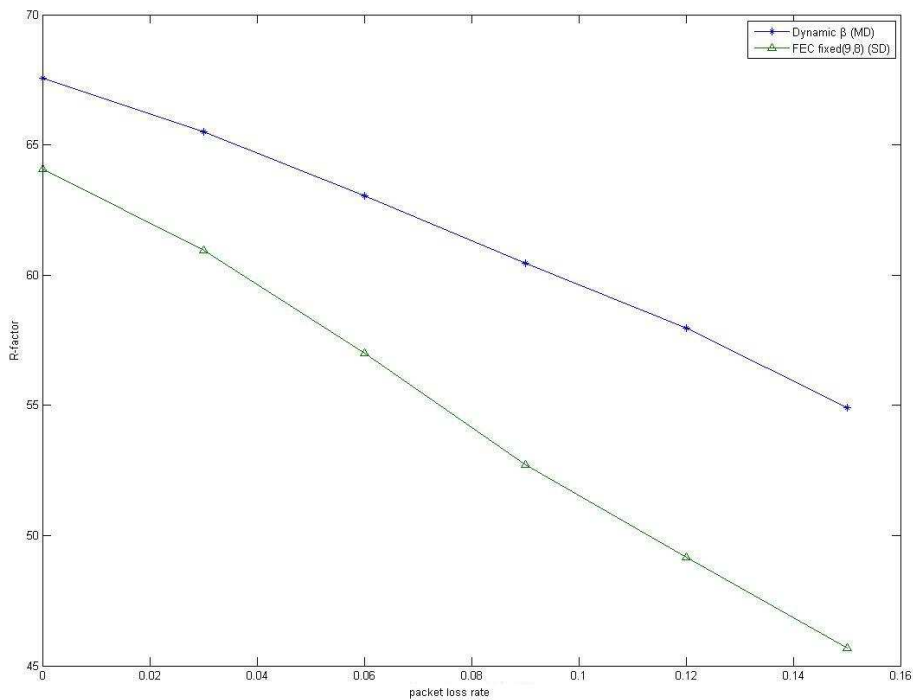


圖 5.3 MD 與 SD 加上固定 FEC RS(9, 8)效能比較

#### 實驗結果分析:

本實驗探討 MD 傳輸與加上 FEC 保護的 SD 傳輸效能比較，FEC 的選擇以接近 MD 傳輸編碼效益為主，由於 G. 729a 的位元率為 8Kbps 經由 MD 編碼後位元率增加到 9.2Kbps，整體編碼效益為  $9.2/8=1.15$ 。因此為了公平起見，SD 傳輸下的 FEC 選擇 RS (9, 8)作為固定的封包傳輸保護機制，其中 SD 傳輸中的安全因子設計方式是實行動態設計的做法[22]。由圖(5.3)可以看出，在相近的編碼效益下，MD 傳輸表現較佳，尤其是在網路環境較糟糕時，更能凸顯 MD 的優勢。這是由於相對 RS FEC(9, 8)來說，MD 也可視為一種前向錯誤控制，只要收到

其中一條串流就可以挽救不能播放的音框，因此 frame erasure rate 能夠有效降低。不過由於音框接收情況不同造成的音框品質相異，已在第二章所提出的音質預測模型上納入考慮，所以本實驗中 MD 傳輸下，音框回復情況是接近實際情況並非是過於樂觀的判斷。另外，圖中 SD 加上 FEC 的保護之所以在網路情況糟糕時表現越差，是由於 RS FEC(9, 8)的保護效果本就有限，在接收端必須等待 9 個封包再來檢視有無成功接收到 8 個以上的封包，才能回復漏失掉的封包，因此在惡劣的網路情況下，RS FEC(9, 8)不僅無法發揮其對抗封包漏失的功能，反而在接收端增加播放延遲，所以才會造成如圖(5.3)中的情況。



●實驗四:結合 FEC 的 MD 播放排程

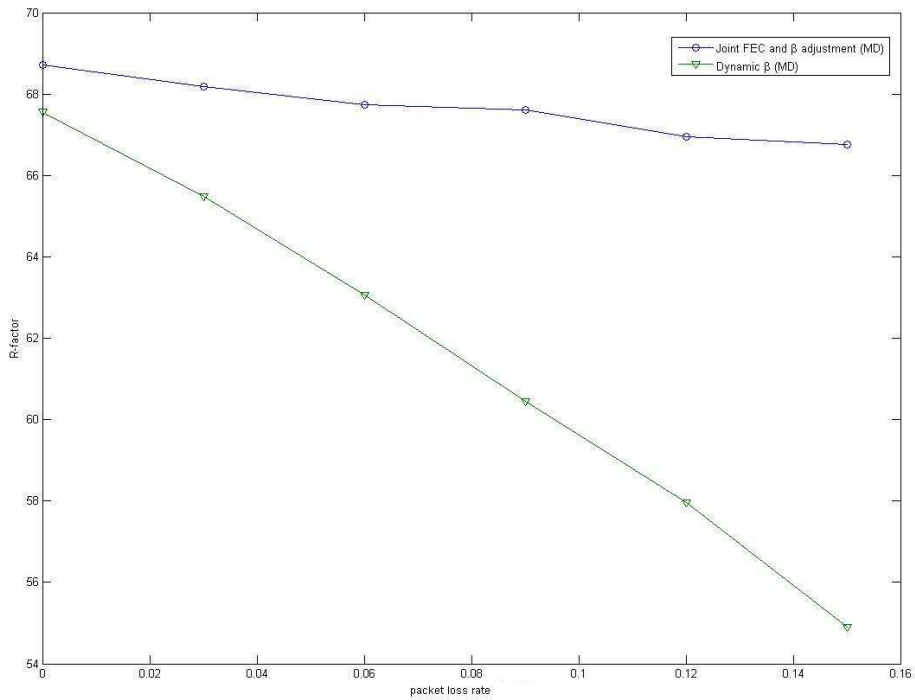


圖 5.4 結合 FEC 的 MD 傳輸優勢( $p+q=1$ ; Trace 1&2)

		0%	3%	6%	9%	12%	15%
Joint FEC and $\beta$ adjustment (MD)	Delay (ms)	147.115	156.137	172.835	173.069	173.051	173.214
	Frame erasure	0	0	0	0	0	0
	Prob{ $\Omega_1$ }	1	0.990	0.989	0.985	0.963	0.957
dynamic $\beta$ (MD)	Delay (ms)	137.568	137.572	137.580	137.583	137.595	137.589
	Frame erasure	0	0.0017	0.0059	0.0114	0.0180	0.0303
	Prob{ $\Omega_1$ }	0.955	0.902	0.852	0.801	0.754	0.709

表 5.8 對照圖(5.4)，接收端音質相關數據的比較

## 實驗結果分析：

首先觀察圖(5.4)可以清楚看出，適應性地整合調整 FEC 及  $\beta$  的 MD 傳輸機制表現比未在傳送端加入 FEC 保護的 MD 傳輸更好，且在網路環境變得糟糕時，更能凸顯其優勢。而透過表(5.8)我們可以推斷出適應性地整合調整 FEC 及  $\beta$  的 MD 傳輸機制的表現能夠如此強健，來自於其以增加整體延遲的代價換來 frame erasure rate 降低的好處，雖然引入 FEC 會使得接收端必須延長其播放延遲，但是透過 FEC 回復封包漏失而所降低的音質損害是比其所帶來的額外延遲音質損害來得多，因此實驗結果也告訴我們如此的 trade-off 在音質上的考量下是值得的。由圖中可清楚觀察到，一旦封包傳輸開始發生網路漏失時，FEC 就開始發揮其回復的功效，反觀只單純動態性調整  $\beta$  的 MD 傳輸系統，在網路環境變差時，就無法抵抗封包漏失所造成的音質損害，同時也更依賴 MD 回復( $\text{Prob}\{\Omega_1\}$ 較低)，因此其效能在網路環境更差時逐漸被拉開。

## ● 實驗五: 動態調整 FEC 與固定 FEC 的 MD 播放排程比較

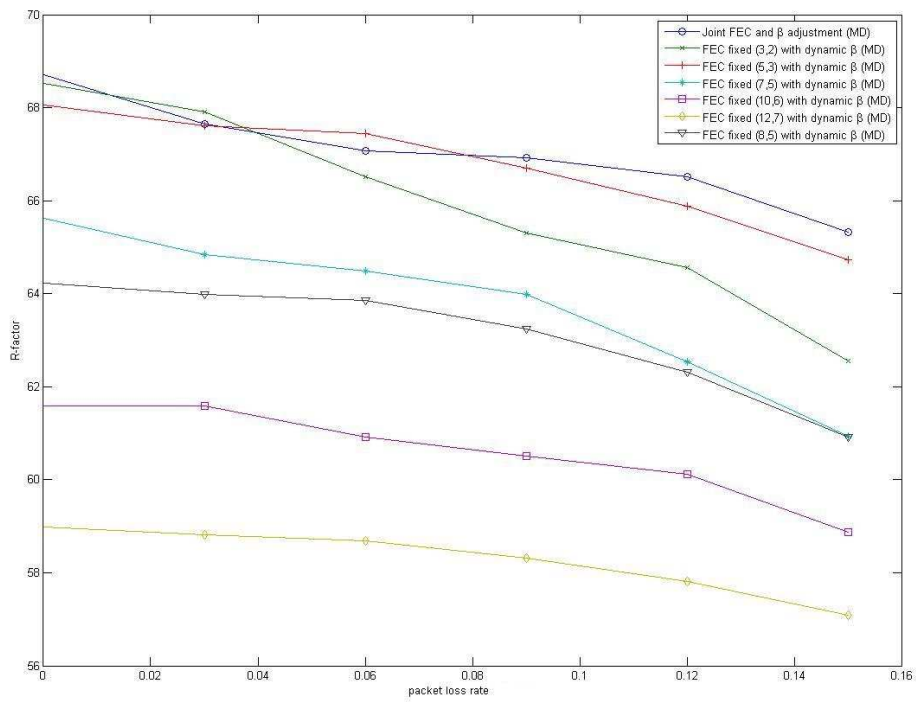


圖 5.5 結合 FEC 的 MD 傳輸( $p+q=0.8$ ; Trace 1&2)

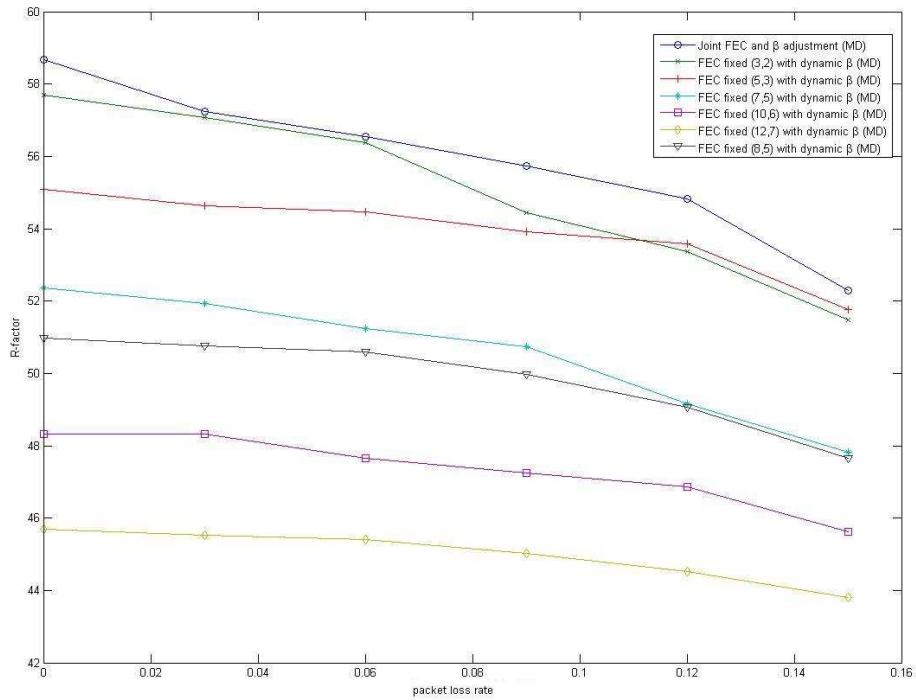


圖 5.6 結合 FEC 的 MD 傳輸( $p+q=0.8$ ; Trace 3&4)

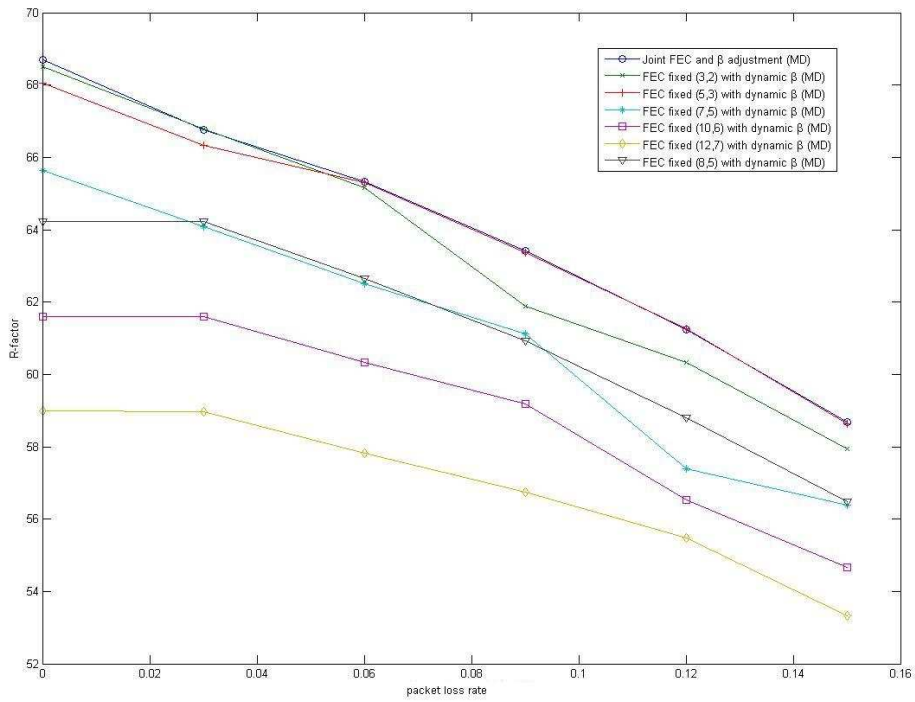


圖 5.7 結合 FEC 的 MD 傳輸( $p+q=0.3$ ; Trace 1&2)

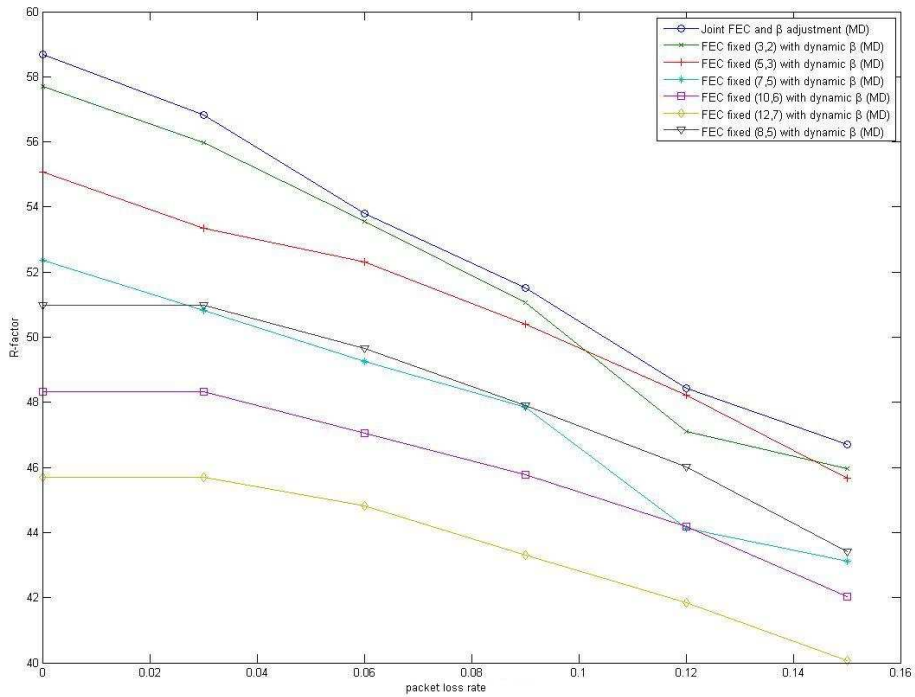


圖 5.8 結合 FEC 的 MD 傳輸( $p+q=0.3$ ; Trace 3&4)

		0%	3%	6%	9%	12%	15%
Joint FEC and $\beta$ adjustment (MD)	Delay (ms)	147.115	156.108	172.879	172.932	173.066	173.271
	Frame erasure	0	0	0	0.0041	0.0082	0.0180
FEC=(3, 2) with dynamic $\beta$ (MD)	Delay (ms)	155.045	155.089	155.155	154.975	155.026	155.040
	Frame erasure	0	0	0.0010	0.0102	0.0102	0.0184
FEC=(5, 3) with dynamic $\beta$ (MD)	Delay (ms)	174.425	174.524	174.510	174.432	174.428	174.376
	Frame erasure	0	0	0.0007	0.0021	0.0050	0.0110
FEC=(7, 5) with dynamic $\beta$ (MD)	Delay (ms)	194.858	194.906	195.016	194.683	194.811	194.788
	Frame erasure	0	0	0	0.0031	0.0112	0.0184
FEC=(8, 5) with dynamic $\beta$ (MD)	Delay (ms)	205.356	205.356	205.402	205.449	205.467	205.392
	Frame erasure	0	0	0.0006	0.0028	0.0050	0.0114
FEC=(10, 6) with dynamic $\beta$ (MD)	Delay (ms)	224.961	224.910	225.020	225.063	225.042	225.027
	Frame erasure	0	0	0.0004	0.0013	0.0050	0.0106
FEC=(12, 7) with dynamic $\beta$ (MD)	Delay (ms)	244.464	244.462	244.535	244.557	244.554	244.536
	Frame erasure	0	0	0.0005	0.0010	0.0027	0.0069

表 5.9 對照圖(5.7)，整體延遲(ms)與回復後 FER 的比較



## 實驗結果分析：

本實驗主要呈現在同樣動態調整 $\beta$ 的 MD 傳輸下，適應性調整 FEC 的好處。由圖(5.5)至(5.8)可以看出在不一樣的漏失環境( $p+q=0.8$  &  $0.3$ )及網路延遲(Trace1&2 及 Trace3&4)下，適應性調整 FEC 的 MD 傳輸機制皆表現較為優越。仔細觀察實驗結果，可以發現固定 FEC(5, 3) 及(3, 2)在不一樣的 frame erasure rate 環境下，其效能接近適應性調整 FEC 的機制，在同樣網路延遲較短的情況下(Trace1&2)，FEC(5, 3) 及(3, 2)在較低的 packet loss rate 環境下皆可以與適應性調整 FEC 的機制並駕齊驅，但是一旦環境(packet loss rate)開始變差時，FEC(3, 2)就逐漸落居下風，原因在於封包漏失的情況也超過其封包回復能力的範圍，而在此時適應性調整 FEC 的機制，就會隨著網路環境的變化選擇更為適當的 FEC 來保護封包的傳送，如圖(5.5)及(5.7)所示。而在同樣網路延遲變長的情況下(Trace3&4)，此時相對於封包漏失，網路延遲所造成的音質損害影響更為顯著，因此從圖(5.6)及(5.8)可以看出，原本在 packet loss rate 較低的情況下與適應性調整 FEC 機制表現差不多的 FEC(5, 3)，在較長的網路延遲環境中就表現得較差了，原因是在如此的傳輸環境中(高延遲)，FEC(5, 3)比起(3, 2)雖然可以回復更多的封包漏失，但是其所額外帶來的延遲所造

成的音質損害反而更加顯著，所以才會造成如此的實驗結果；而當 packet loss rate 開始變高時，封包漏失造成的音質損害逐漸提高，因此 FEC(5, 3) 逐漸發揮其比 FEC(3, 2) 更加優越的封包回復能力，因而表現更好的效果。而適應性調整 FEC 的機制之所以可以在不一樣的 frame erasure rate 下皆表現強健，在於其適應性調整的機制不僅考量了 packet loss rate 的變化，也同時考慮了網路延遲大小所對應的音質損害，因此它可以隨著網路環境的變化，適應性地選擇較能反應網路情況的 FEC。另外，在其他固定 FEC 的實驗結果，我們發現雖然 FEC(12, 7) 擁有最高的編碼效益(1.7143)，但是其所帶來的額外延遲卻更為巨大( $11T_p$ )，所以雖然相較於其他 FEC 而言，其對抗封包漏失的能力較佳(曲線較平緩)，但由於其額外的延遲而造成更糟糕的延遲音質損害，所以其最後整體表現(R 值)較差。而 FEC(10, 6) 雖然與 FEC(5, 3) 有相同的編碼效益，甚至在更糟糕的叢發性漏失環境下，其回復能力能夠比 FEC(5, 3) 來得好，但是在我們的實驗環境下，相對於 FEC(5, 3) 而言，其所帶來兩倍的額外延遲所造成的音質損害是其最後表現較差的最大原因。而在圖(5, 5)及(5, 6)中，FEC(7, 5) 表現大致上比 FEC(8, 5) 好，當 packet loss rate 變高時，FEC(8, 5) 才表現更為接近 FEC(7, 5)，這是由於此時叢發性的封包漏失情況較不為嚴重( $p+q=0.8$ )，所以 FEC(8, 5) 比起 FEC(7, 5) 更佳的回復能力沒有辦

法被突顯出，反而因為在接收端，比 FEC(7, 5)多帶來  $T_p$  的延遲，所以效果較差。而一旦叢發性漏失變得更為嚴重時( $p+q=0.3$ )，FEC(8, 5)就發揮其較佳的回復能力而使得其最後效能比 FEC(7, 5)還好。從這些實驗數據，也可以觀察出並不是使用最高編碼效益的 FEC 就能夠表現最佳，選擇 FEC 時也必須考慮其所帶來的額外延遲，因此才會有 FEC(12, 7)表現較差的結果，而我們所提出的適應性調整 FEC 機制正是將所有損害音質的因素(網路延遲、frame erasure rate)納入考量，因此能夠隨著不一樣的網路環境來適應性地選擇一組最佳 FEC 來保護封包的傳輸，從表(5.9)中的整體延遲更可以看出其隨著 packet loss rate 的提高而動態調整其 FEC 的選擇，並做出最佳的 trade-off。



## 實驗六:動態調整 FEC 於 MD 和 SD 傳輸的比較

在本實驗我們將利用 4.3 節中所提出的整合漏失回復後的 MD 音質損害模型裡的 FEC 回復機率來設計動態調整 FEC 的單一串流傳輸系統，並限制其 FEC 碼率，以其於公平的條件下與 MD 傳輸系統做比較。

在第四章 4.3 小節裡式(4.13)即是代表在串流 s 中第 i 個封包發生網路或晚到漏失時，沒辦法藉由 FEC 回復的機率。為了設計出適用於 SD 傳輸系統的 FEC 機率模型，我們將應用此式並簡化 4.4 節的 MD 系統架構而整理出 SD 封包傳輸系統中的最佳化設計：

$$PLR_{FEC} = \frac{1}{K} \sum_{i=1}^K \left\{ \frac{p}{p+q} (1 - P_{REC1}(i)) + \frac{q}{p+q} (1 - F_D(D_{FEC,i})) (1 - P_{REC2}(i)) \right\} \quad (5.1)$$

$$I_e(e) = \gamma_{1,1} + \gamma_{2,1} \ln(1 + \gamma_{3,1}e) \quad (5.2)$$

Given:  $p, q, F_D(d)$

maximize:  $R = 94.2 - I_e(PLR_{FEC}) - I_d(\hat{d}_i + \beta \hat{v}_i + (N-1)T_p + dc)$

over all:  $(N, K), \beta \quad (5.3)$

subject to:  $\begin{cases} \frac{N}{K} < 2 \\ K \geq \text{The number of packets of next talkspurt} \end{cases}$

經由 MD 系統簡化的適應性整合 FEC 並調整播放延遲的 SD 傳輸系統如圖(5.9)所示。在接下來的實驗我們將讓音框經由語音編碼處理後，分成三種方式來處理，第一種方式為將其作 MD 傳輸，讓音框分成兩個封包並獨立於 Trace1 及 Trace2 傳送，第二種 SD 傳輸(Trace1)，最後一種為 SD 傳輸(Trace2)。

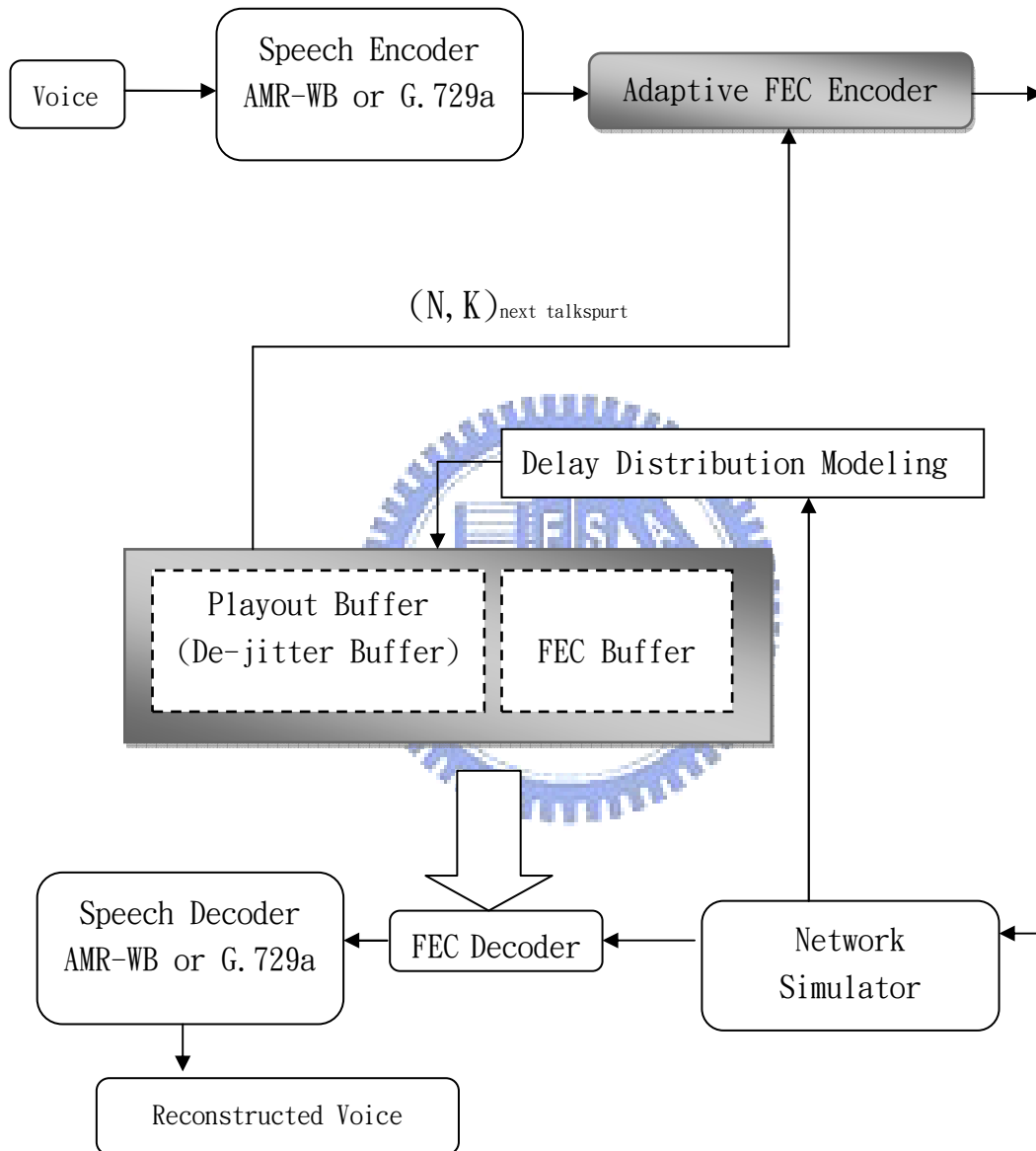


圖 5.9 結合 FEC 的 SD 傳輸系統

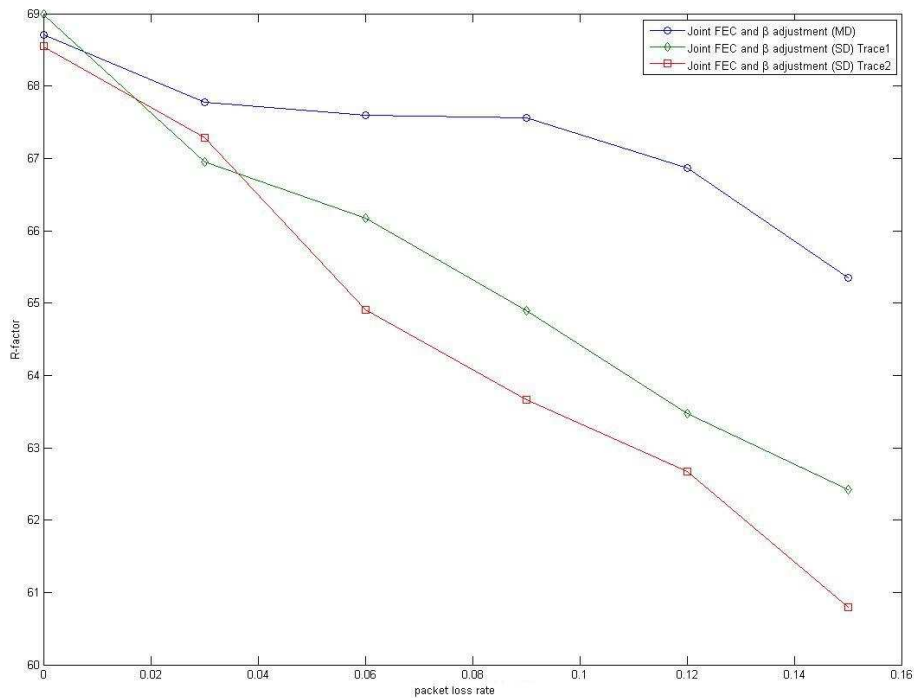


圖 5.10 結合 FEC 的 SD 與 MD 傳輸效能比較( $p+q=0.9$ )

### 實驗結果分析：

本實驗主要的目的為驗證同樣是整合 FEC 並動態調整安全因子的情況下，MD 與 SD 傳輸系統的效能比較。由圖(5.10)可以觀察出，在相同傳輸效率限制之下，MD 傳輸系統表現強健許多，尤其是在網路漏失環境趨於惡化時更為顯著，這是由於當網路環境不理想時，路徑多樣性(path diversity)的優勢逐漸顯現出來，藉由收到其中一條串流來回復音框。MD 傳輸於圖中的平緩曲線也正說明了 MD 編解碼能充分利用 path diversity 來有效抵抗網路漏失。

## 第六章 結論與未來展望

本論文旨在探討網路語音通訊應用中，如何有效克服封包傳輸在網路環境所遭遇的諸多問題，例如：網路延遲、延遲顫動及封包漏失。只要作法是結合前向錯誤控制與多重敘述編碼來處理封包漏失的問題，並同時設計封包播放排程以消除接收端的延遲顫動。有別於傳統的音質預測模型，我們先設計出一適用於多重敘述傳輸的音質預測模型，並透過系統模擬證實其預測結果更接近主觀聽覺測試。再依此為發展平台設計以聽覺最佳化為目標的傳輸系統，而系統模擬結果也顯示我們所提出的多重敘述系統能有效降低緩衝延遲及音框刪除比率(FER)。系統設計同時考量了不同封包接收狀態下所對應的語音輸出品質，在單一及雙重封包漏失、整體延遲之間做出理想的取捨，因而能展現較佳的效能。為了抵抗更惡劣的網路漏失情況，我們再進一步結合前向錯誤控制於多重敘述傳輸系統中。主要概念是利用其封包回復機制提供傳輸系統更好的封包漏失抵抗能力，雖然其大幅降低的 FER 背後所付出的代價是額外帶來的 FEC 緩衝延遲。然而在最後系統模擬結果中，可以觀察到我們所提出的多重敘述傳輸系統，能夠在權衡整體延遲及 FER 兩者之間的得失後，選擇出最佳的系統參數，使其於最後模擬結果表現出相對強健的輸出效能。此外，結合前向錯誤

控制不僅降低了 FER，也同時增加了雙重封包接收狀態的比例，使得接收端減少了依賴多重敘述解碼的音框重建機制的機會，因而能輸出優良的音質。在系統模擬結果中也可以觀察到，比起固定前向錯誤控制的設計，適應性調整前向錯誤控制的機制能在不同的網路條件下皆能保持其強健的效能。而其優勢主要來自於其成本函數不僅能反應網路傳輸環境的變化，也同時考量不同的前向錯誤控制參數其封包回復能力及額外延遲，因而能選擇出最適於當時網路傳輸環境的系統參數  $(N, K)$ 。

系統模擬驗證了本論文提出的傳輸系統能有效改善語音品質，但是在尋找系統最佳參數的演算法設計中含有大量的計算複雜度。因此在實作過程中，將耗費不少時間，未來將可朝此方向做改進。另外，因應無線網路與網際網路整合應用的崛起，未來將可應用本論文提出的多重敘述傳輸架構，將無線傳輸中更為嚴重的延遲顫動納入考量，以期能夠發展適於無線網路語音應用的多重敘述傳輸系統。



## 參考文獻

- [1] R. Ramjee, J. Kurose, D. Towsley, and H. Schulzrinne, “Adaptive playout mechanisms for packetized audio applications in wide area networks,” in *Proc. IEEE Infocom Conf. Comp. Commun.*, vol. 2, (Toronto, Canada), pp. 680-688, June 1994.
- [2] S. B. Moon, J. Kurose, and D. Towsley, “Packet audio playout delay adjustment: Performance bounds and algorithm,” *ACM/Springer Multimedia Systems*, vol. 5, pp. 17-28, Jan. 1998.
- [3] 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
- [4] J.-C. Bolot, S. F. Parisi, and D. Towsley, “Adaptive FEC-based error control for Internet Telephony,” in *Inforcom’99*, Mar. 1999, pp. 1453-1460.
- [5] J. Rosenberg, L. Qiu, and H. Schulzrinne, “Integrating packet FEC into adaptive voice playout buffer algorithms on the internet,” in *Proc. IEEE INFOCOM 2000*, vol. 3, Tel Aviv, Israel, Mar. 2000, pp. 1705–1714.
- [6] Yi-Liang, “Robust Low-Latency Voice and Video Communication Over Best-Effort Networks” Ph.D. dissertation , Dept. Elect. Eng., Stanford Univ. ,CA,USA, 2003.

- [7] J. Balam and J. D. Gibson, "Multiple Descriptions and Path Diversity for Voice Communications Over Wireless Mesh Networks," *IEEE TRANSACTIONS ON MULTIMEDIA*, vol. 9, no. 5, p.1073, 2007.
- [8] *Adaptive Multi-rate Wideband Speech Transcoding*, 3GPP TS 26.190 Std.
- [9] *Coding of Speech at 8 kbit/s Using Conjugate-Structure Algebraic-Code-Excited Linear-Prediction (CS-ACELP)*, ITU-T G.729 Recommendation, Nov. 2000.
- [10] The E-model , a computational model for use in transmission planning, "ITU-T Recommendation G.107, May. 2000.
- [11] The Network Simulator 2. Available: [http://www.isi.edu/nsnam-  
/ns/](http://www.isi.edu/nsnam/ns/)
- [12] "Method for subjective determination of transmission quality, "ITU-T Recommendation P.800, Aug. 1996.
- [13] "Subjective performance assessment of telephone-band and wideband digital codecs, "ITU-T Recommendation P.830, Feb. 1996.
- [14] Cole, R. G. and Rosenbluth, J. H. "Voice over IP performance monitoring," *ACM Computer Communication Magazine* 34, 12 (Dec.1996), pp. 9-24.
- [15] L. Sun and E. Ifeakor. New models for perceived voice quality prediction and their applications in playout buffer optimization for VoIP networks. In *Proc. IEEE Communication*, volume 3, pages 1478–1483, 2004.

- [16] NCSS (Statistical & Power Analysis Software).  
Available: [http://www.ncss.com/ncss\\_upgrade.html](http://www.ncss.com/ncss_upgrade.html)
- [17] 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.
- [18] W. Verhelst and M. Roelands, "An overlap-add technique based on waveform similarity (WSOLA) for high quality time-scale modification of speech," in *Proc. ICASSP*, Apr. 1993, pp. 554–557.
- [19] K Fujimoto, S Ata, and M Murata "Statistical analysis of packet delays in the internet and its application to playout control for streaming applications," *IEICE Transactions on Communications*, vol. E84-B, pp. 1504-1512, June 2001.
- [20] E.KP. Chong and S.H. Zak, *An Introduction to Optimization*, John Wiley & Sons, Inc., 2001
- [21] Pascal Frossard, "FEC Performance in Multimedia Streaming," *IEEE COMMUNICATION LETTERS*, vol. 5, NO. 3, pp. 122-124 MAR. 2001.
- [22] Chun-Feng Wu, and Wen-Whei Chang, "*PERCEPTUAL OPTIMIZATION OF PLAYOUT BUFFER IN VOIP APPLICATIONS*" ,First International Conference on Communications and Networking in China (CHINACOM 2006), Oct,2006