國立交通大學

電機學院通訊與網路科技產業研發碩士班

碩 士 論 文

行動語音人機介面之研究

A study of mobile interactive voice response system

研 究 生：何依信

指導教授：張文輝　教授

中 華 民 國 九 十 六 年 六 月

行動語音人機介面之研究

學生：何依信　　指導教授：張文輝 博士

國立交通大學電機學院產業研發碩士班

中文摘要

人性化隨身資訊服務是未來的發展趨勢，其關鍵在於開發一聲控操作的語音人機介面。本論文在網際網路環境，建構一分散式語音辨認系統，再根據辨認結果回傳特定的有聲資訊給用戶。網路語音通訊最重要的課題是服務品質管理，特別是封包漏失、傳輸延遲及延遲顫動。為了補償封包漏失，我們採用多重敘述編碼架構，透過兩個獨立的網路通道傳送語音封包。至於延遲擾動的解決方案，一般是在接收端加入一播放緩衝器暫存語音封包，再彈性調整每個語音封包的播放時間。由於網路延遲在話務中間的變動，語音封包的晚到漏失率與其緩衝延遲及之間存在一個最佳化權衡的問題。我們將在多重敘述編碼架構下，根據客觀的音質預估模型，針對每個獨立封包的播放延遲進行音質最佳化調整。

A study of mobile interactive voice response system

Student:Yi-Hsin Ho Advisor:Dr. Wen-Whei Chang

Industrial Technology R & D Master Program of

Electrical and Computer Engineering College

National Chiao Tung University

Abstract

The purpose of this research is to develop an interactive voice response system that allows drivers to use voice-controlled commands to access the information server through the internet. We first implement a distributed speech recognition system, in which speech features extracted from a local front-end are transmitted through a data channel to a remote back-end recognition server. Another important issue to address is the playout buffer design, which is often used at the receiver to smooth out the jitter for timely reconstruction of the speech. We formulate the adaptive playout scheduling of multiple voice streams as a constrained optimization problem that leads to a better balance between end-to-end delay and packet loss. Also proposed is a perceptually motivated optimization criterion and a practically feasible algorithm for the playout buffer design.

誌謝

兩年的研究生涯，首先要感謝指導教授張文輝老師，讓我深刻體認到做研究所需具備的嚴謹，並在研究遇到瓶頸時從旁協助找出正確的方向。相信這對日後我在職場上的態度將有深遠的影響。另外也感謝實驗室的朋友在學業與研究上的諸多協助。特別要感謝這段期間怡敏給予我精神與生活上的支持鼓勵，使我碩士班的生活更加的多采多姿。最後要感謝父母及家人的支持讓我能順利完成碩士論文。

目錄

圖目錄

表目錄

第一章 緒論

1.1 研究動機與方向

網際網路的初期應用在於資料的傳送與接收，強調的是資料是否可以被正確地接收。近來由於行動通訊的頻寬增加，使得多媒體通訊在網路的應用增多。以智慧型運輸系統為例，整合無線通訊與網際網路，提供駕駛人更人性化的隨身資訊服務，以提升行動用戶的安全與便利，已成為智慧型運輸系統必備的功能。

針對整合型網路服務之研究，我們計畫在 Mobile Ad-hoc Network(MANET)網路架構下，提供一聲控操作的語音人機介面。由於 MANET 無線通訊平台有別於傳統有線網路，可靠度較低，封包延遲(delay)與漏失(loss)相對較高，因而嚴重影響網路的服務品質。目前網路語音通訊中面臨最主要的三項課題為封包漏失、傳輸延遲及延遲顫動(delay jitter)。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(playout scheduling)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]。

我們進一步採用多重敘述編碼(multiple description coding)架構，透過兩個獨立的網路通道傳送，以期在單一通道封包漏失時仍能還原可接受的輸出音質。在此架構下，我們將根據客觀的音質預估模型，針對每個獨立的封包做播放時間的音質最佳化調整，以適應網路在話務(talkspurt)中間變動的情形。由於每個經過緩衝器的封包播放的長度不是固定，必須配合使用音長比例調整(time-scale modification)，才能保持語音播放的連續性。最後，為了比較不同播放演算法的輸出音質，將參考國際電信聯盟 ITU 提供的 ITU-T P.862 以及G.107，製作一量測各項服務品質因素的測試平台。

1.2 章節概要

第二章介紹互動式語音人機介面與其在移動式自組網路(MANET)的軟體製作，第三章介紹播放排程演算法，第四章介紹在客觀的音質評量平台下，語音封包播放排程的最佳化設計，第五章先介紹網路延遲模型與 MANET 實地量測延遲，並藉此比較單一敘述與多重敘述串流傳輸的差異。

第二章 語音人機介面

近年來在軟硬體及網路的發展下，在任何地方都可進行計算與通訊的應用服務已經具體實現。此項需求導致手持式裝置的蓬勃發展，特別是嵌入式系統能提供低耗電、低成本、高移動性的優點。嵌入式系統原是用來執行特殊目的(special-purpose)的電腦,專門負責特定且少量的工作，因此系統工程師可以同時兼顧最佳化設計與降低成本。若從高移動性的應用角度來看，移動式自組網路(Mobile Ad-Hoc Network，MANET)架構更能滿足這方面的需求。

移動式自組網路開始於 1970 年代附近，由於組成簡單及生存能力強特點，其應用範圍越來越廣泛，例如車機系統以及軍事、地震等緊急通訊。有別於傳統的無線網路，自組式網路不需要固定的基礎網路架構(infrastrusture)，亦即不需要像手機一樣有固定的基地台或是路由器的網路架構。然而如果要成功達成通訊的目的，必須透過多重跳躍(multi-hop)的方式。再配合使用自組式網路閘道(ad-hoc gateway)，因為自組式網路的移動客戶端並無有效的 IP 位址來連上網際網路。

2.1 系統介面設計

移動式自組網路應用在車機系統上時，必須兼顧駕駛人的行車安全與操作方便，語音人機介面使用麥克風可避免駕駛人手動輸入。本章採用 linux 系統以符合上述需求，由圖 2.1 得知,代理者(Agent)代表車上架構較簡單的嵌入式系統，其軟體搭配 linux 作業系統。由於代理者的記憶體與運算能力有限的實際考量，因此額外增加一台電腦執行錄音以及語音特徵抽取參數的助理(Assistant)工作，其搭載的作業系統為 Windows。從整體架構來看圖 2.1，可以單純地把代理者以及助理看成是一個用戶端(Client)。伺服器(Server)端則利用特徵參數執行語音辨識，並依辨識結果回傳特定的有聲資訊給代理者。

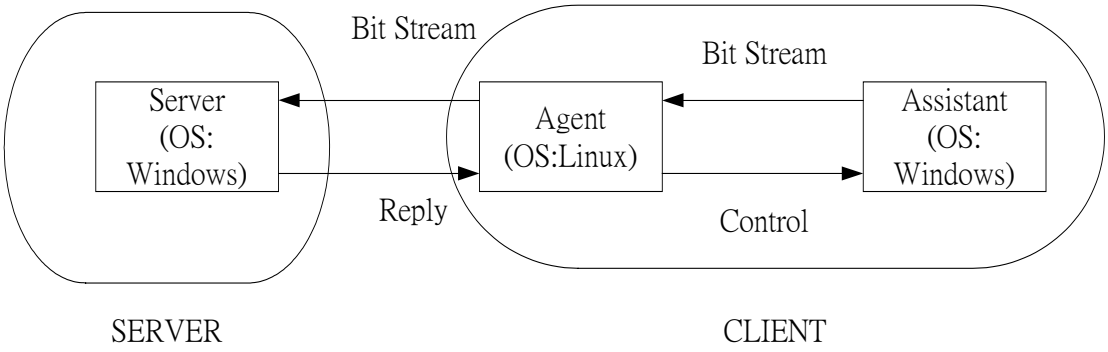


圖 2.1 語音人機介面

圖 2.2 描述整個系統運作的流程，由代理者控制錄音的開始與結束，等到助理完成錄音與參數抽取工作後，由助理將結果透過有線區

域網路回傳給代理者，這裡的區域網路無法對外連線。代理者再將結果透過無線網路傳給自組式網路的閘道器(ad-hoc gateway)，閘道器再透過網際網路(有線網路)傳給伺服器。伺服器的主要工作是執行分散式語音辨識(DSR)的後級辨識，之後再依辨識結果回傳有聲資訊給代理者，其流程為上述步驟的反向程序。

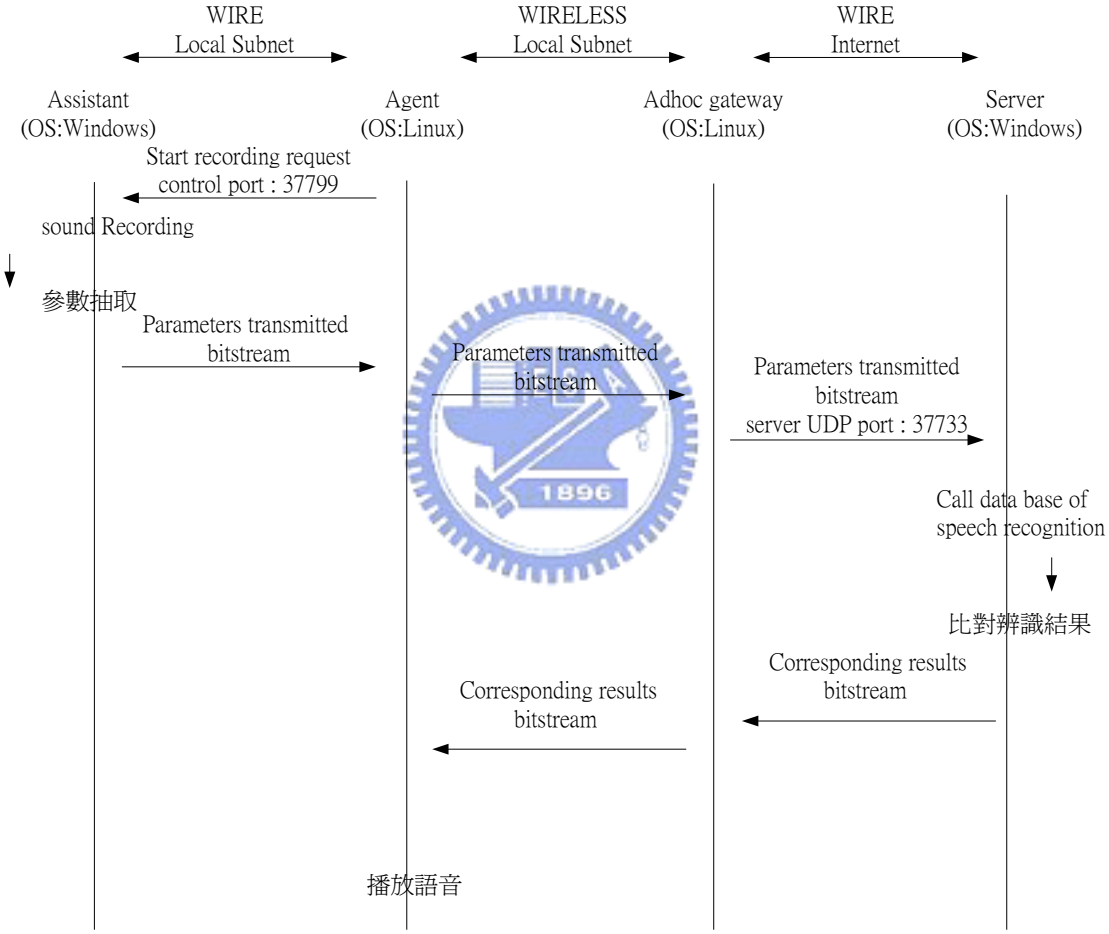


圖 2.2 語音人機介面的運作流程

2.1.1 分散式語音辨識

將自動語音辨識技術應用在行動或IP網路上的構想，透過DSR的製作可以具體實現。在ETSI的標準ETSI ES 202 212(v.1.1.1)裡，描

述了整個DSR系統中的語音處理、傳輸以及品質效能。標準中並定義了前端語音特徵參數的擷取，以及將其壓縮處理再傳送至伺服器端的編碼機制。DSR的基本概念就是在前端擷取語音參數，再經由資料通道(data channel)傳送至後端執行較複雜的語音辨識。主要是因為若直接將語音傳輸時，往往因為低位元語音編碼率以及通道傳輸錯誤使得系統效能嚴重地下降。所以DSR捨棄語音通道(voice channel)，而是代之以有錯誤保護的資料通道來傳輸語音特徵參數。DSR是將原本的辨識模組分成前端的參數抽取與後端的語音辨認，由於整個過程取決於參數的傳遞，因此可以有效對抗通訊環境中的干擾。圖2.3即為DSR的架構圖。

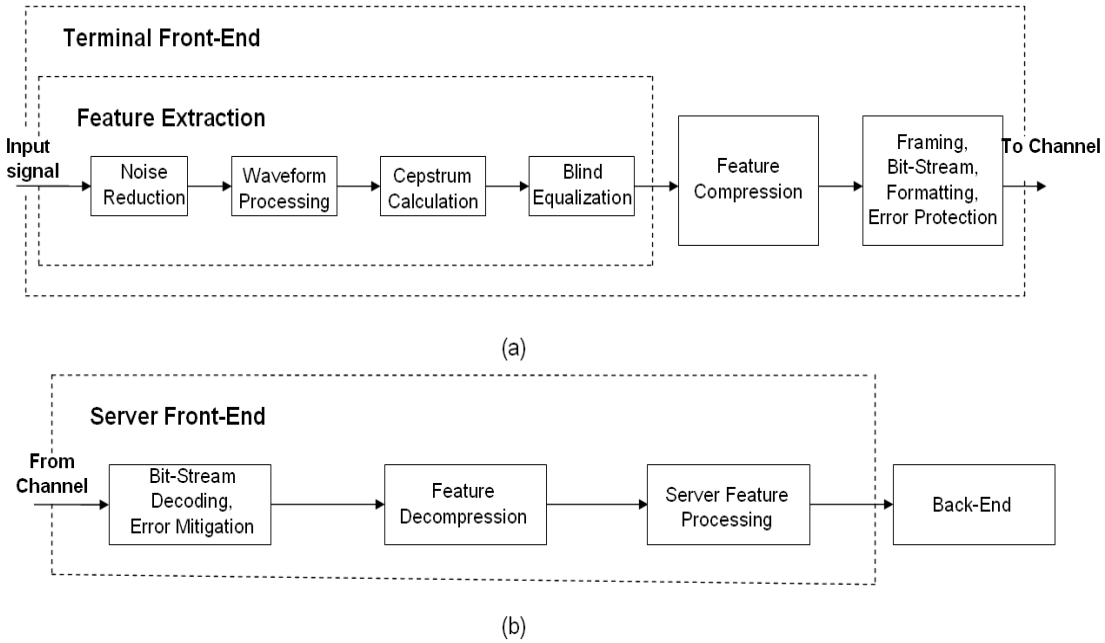


圖 2.3 分散式語音辨識系統(a)終端機(b)伺服器

2.1.2 軟體架構

由圖 2.4 得知，聲音先在用戶端錄製，經過參數抽取以及編碼後傳送至伺服器端作辨識。本論文將 DSR 的細部工作當成是黑盒子(black box)，只用來處理伺服器端接收到的語音參數，最後將辨識結果對應的語音檔案透過 udp 通訊協定回傳給客戶端。

考慮到網路延遲會影響語音通訊的即時性，我們採用 DSR 將聲音即時處理並將辨識參數由客戶端傳給伺服器端，而不是類似 ASR 架構要等到聲音從頭到尾在伺服器端收到後才開始作辨認。在傳回辨識結果應用的部分，可以將使用者需要的資訊以語音的型態播放出來，如圖 2.2 流程最後在用戶端播放。

2.1.3 網路協定封包製作

目前常用的網路通訊協定中，傳輸層(transport layer)主要分為 TCP(transmission control protocol)與 UDP(user datagram protocol)兩種，TCP 為連線導向(connection-oriented)，UDP 則為非連線(connectionless)導向，兩者都是以底層通訊協定 IP 為基礎。因為沒有做任何錯誤更正的動作，UDP 提供的是快速但不可靠的傳輸協定。TCP/IP 網路架構的設計目的是可以在異質網路下達到通訊的目的，基本上有 4 個層級，如圖 2.5 所示。

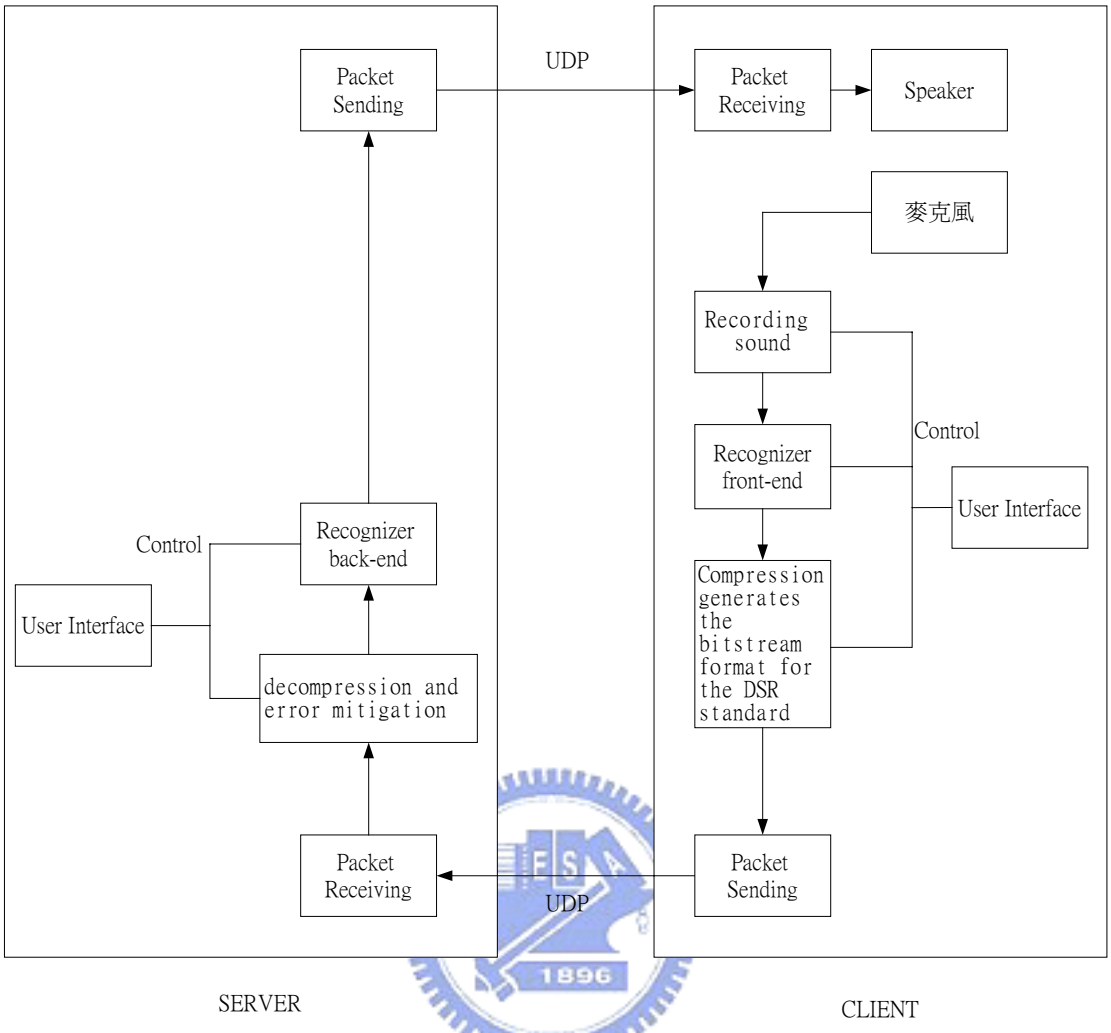


圖 2.4 軟體設計流程

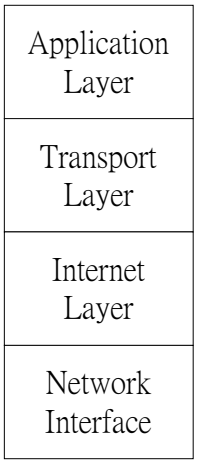


圖 2.5 TCP/IP 架構

舉例來說，目前用途廣泛的全球資訊網(world wide web)瀏覽功能，從圖 2.6 得知，此功能被歸類在應用層(application layer)，使用的通訊協定為 HTTP，在應用層下的一層為傳輸層(Transport Layer)，使用的網路層(Internet Layer)協定為 TCP。

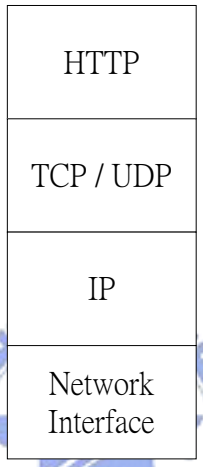


圖 2.6 TCP/IP 範例

選取 UDP 網路協定的原因是考慮到時間延遲會影響最後 MOS 值的評量，封包傳送延遲時間能夠越少，使用者的感受越好，MOS 值越高。UDP 屬於 best-effort transmission，其特色為盡可能利用所有的資源傳送，不需要預先建立連結，不論接收成敗皆不斷的傳送封包，符合即時性的需求，然而缺點是不可靠，無法保證封包的接收。圖 2.7 為本系統所採用的封包格式。

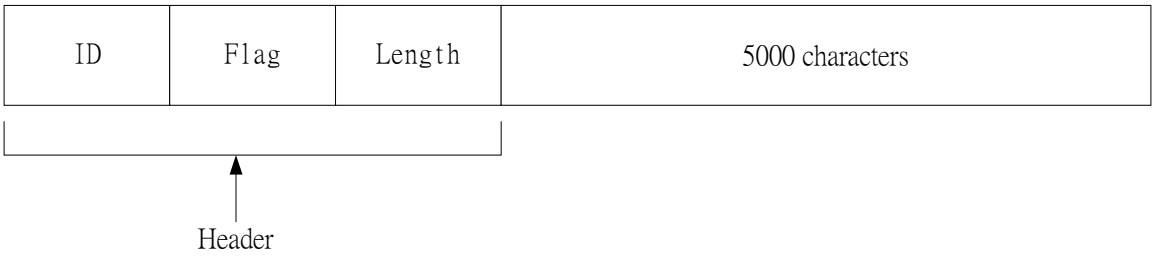


圖 2.7 封包格式

其中

ID: 辨識用的資訊，因為網路上有不同內容的封包，系統運作時需要有能夠辨識封包種類的資訊。

Flag:旗標(前3個byte是順序， 最後1個是type)。

Length:封包長度。

2.1.4 軟體運作的設定

用戶端:不用設定檔

伺服器端: 程式執行時會從執行檔的所在目錄下尋找設定檔。以下分別是這兩個目錄，其他的檔案是執行時會產生的暫存檔，可以移除。

- 目錄”htk”

  辨識函式庫用的設定檔，設定請參考 HTK 函式庫的說明，在這裡我們可以選擇適當的模組來辨識數字與中文字串。

- 目錄”back_data”

伺服器回傳資料的設定，back_data\config 是文字檔，依前述的格式設定，每一列第一項是 HTK 辨識的結果(一段文字，參考 HTK 函式庫)，第二項是相對應要回傳給用戶端的檔案(目前用戶端只支援副檔名為.wav 的檔案格式)

2.1.5 檔案架構

在安裝微軟的 Visual Studio.Net 2003 後,可以直接開啟副檔名為.sln 的檔案，可以開啟整個整理好的工作環境，且看到分類後的檔案。如果用其他的開發軟體版本必須自行建立編譯環境，基本上都是使用目錄分類。

以下為目錄下的各子目錄：

ETSI\

這個目錄下存放 ES 202 050 函式庫的程式碼，又分為 4 個目錄，分別代表原本的 4 個階段

AdvFrontEnd、coder_VAD、decoder_VAD、derivCalc

全部都改成 C++檔案(.cpp)，每個階段都以類別(class)封裝.

HTKLib\

這個目錄下存放辨識用的函式庫，程式的進入點是 HVite.c，目錄中其餘的部份會由函數 Hvite()呼叫.

Client\、Server\

Client 及 Server 端的程式部分，這邊的程式指的是 GUI 介面的主程式，主要是由開發軟體的精靈產生，目錄下還有資源檔、專案檔和編譯後的執行檔(在各自的 Debug 或 Release 目錄下)

Common\

原本的目的是放一些常用的程式碼，不過最後包含上述目錄以外的程式，以下是簡單的描述。

myDefine.h

通用的定義檔案，如 port 的指定，各種 buffer 的長度等。

myQueue.cpp(.h)

一個有同步機制寫的 queue。

SoundIn.cpp(.h) 、 SoundIn.cpp(.h)

在 win32 下錄音和播放的程式碼。

udp_client.cpp(.h) 、 udp_server.cpp(.h)

網路程式的部分。

MainClientThread.cpp(.h) 、 MainServerThread.cpp(.h)

主要作處理的部份，Client/Server 呼叫這部份程式(以執行緒(Thread)的方式)，這部份程式再呼叫下層的函式庫(如 ETSI 函式庫)做處理，處理的方式是利用 buffer 儲存每一

階段的資料，每階段間處理的函式庫讀前一階段的 buffer 做處理，然後寫入下一階段的 buffer。

test_exe8 區\

測試用的目錄，將編譯好的程式檔和設定檔一起執行。

linux_client\

將 client 部分搬到 linux 上的程式，使用 OSS 作為音效的介面(所以也有改到 common 下的檔案)，這目錄下所有的程式都是在 linux 下執行。說明如下：

linux_main.cpp

相當於沒有 linux 時的 client，編譯的動作為，在這個目錄下打 make 指令，在嵌入式系統上執行要把 CC = g++ -pthread 這一行的 g++用 arm-linux-g++取代，需要預先執行 config 指令。

soundout.cpp

另一個獨立的簡單小播放程式，用來播放 debug 時錄下來的 raw wave 檔，編譯時的指令為 make soundtest

rec_play_test\

裡面有 2 個獨立的錄(rectest.cpp)放(playtest.cpp)音程式，編譯方式是直接編譯這個檔案(如 g++ -o rectest

rectest.cpp -lpthread)，錄音程式執行方式./rectest [channel] [sample_rate]，會寫到 t1.wave 這個檔案上，放音./playtest [file] [channel] [sample_rate]，需要預先執行 config 指令。

exe\

測試區(嵌入式系統上使用)。

linux_agent\

屬於代理者的部分，主要是 linux_agent.cpp 這個檔案，編譯 windows 版本使用 VC(Visuial Studio C++)、linux 版本用 make 指令，需要預先執行 config 指令。

2.2 MANET 網路的延遲時間量測分析

本論文研究為國科會三年期整合型計畫「通訊/資訊聚合式車機系統之研發與應用」的一子計畫，目的在利用 MANET 整合平台開發一行動語音人機介面。至於車機系統軟硬體平台之規劃與開發，則是由交大電信系唐震寰教授率領的研究團隊負責。藉由他們所提供的 MANET 整合平台，我們嘗試量測並分析實驗環境設定對網路延遲的影響，以提供後續的語音播放緩衝設計之用。

[實驗 2-1]

目的:室內傳播環境對傳輸延遲的影響。

進行步驟:

雖然 MANET 主要探討室外的通訊應用環境,但為了進行初步效能量測與比較,本實驗同時於室內環境進行多接跳傳播實驗。本實驗利用具有鋼筋水泥牆壁遮蔽、以及傳播空間遭受限制的走廊環境進行,量測地點位於交通大學工程四館 9 樓,如圖 2.8 所示。固定節點設置在圖中標示之 1、2、3、4 位置,節點 1 負責發送量測用資料封包,量測設備分別於 2、3、4 位置量得 1-hop、2-hop、3-hop 傳輸模式之傳輸效能。由於水泥牆壁能夠有效遮蔽本實驗設備的無線電訊號,因此本實驗傳播場景能夠在有限的空間建立出多接跳傳輸環境,本實驗所設定的路徑跳接數與網路節點位置都經過實際測試與量測驗證,因此對於特定測試項目不會同時產生多條傳輸路徑提供傳輸服務。

本實驗主要的量測路徑與範圍包括 3 條夾角為 90 度之室內走廊,走廊兩側多為門窗緊閉之辦公室與實驗室,量測時網路節點間的傳播環境屬於視線內傳播。圖 2.9 呈現本實驗固定點與移動點之設備安裝方式。

結果與分析:

圖 2.10 表示本實驗封包傳輸延遲時間在不同接跳數下之效能變化。由圖中數值上的變化可知,1-hop 可能遭受室內傳播環境的空間

限制，其封包傳輸延遲與 2-hop 相當，而在跳接數增為 3 時，封包傳輸延遲發生顯著的增加。理論上此現象導因於中繼封包轉送節點數增加，而需要較多處理時間所造成。另一方面，封包傳輸延遲的標準差數值都相當高，甚至超過封包傳輸延遲之量測平均數值。由量測數據觀察可知，高標準差導因於若干延遲時間特別高的量測記錄，由此現象可知傳輸通道的不穩定性對封包傳輸延遲的效能影響嚴重。

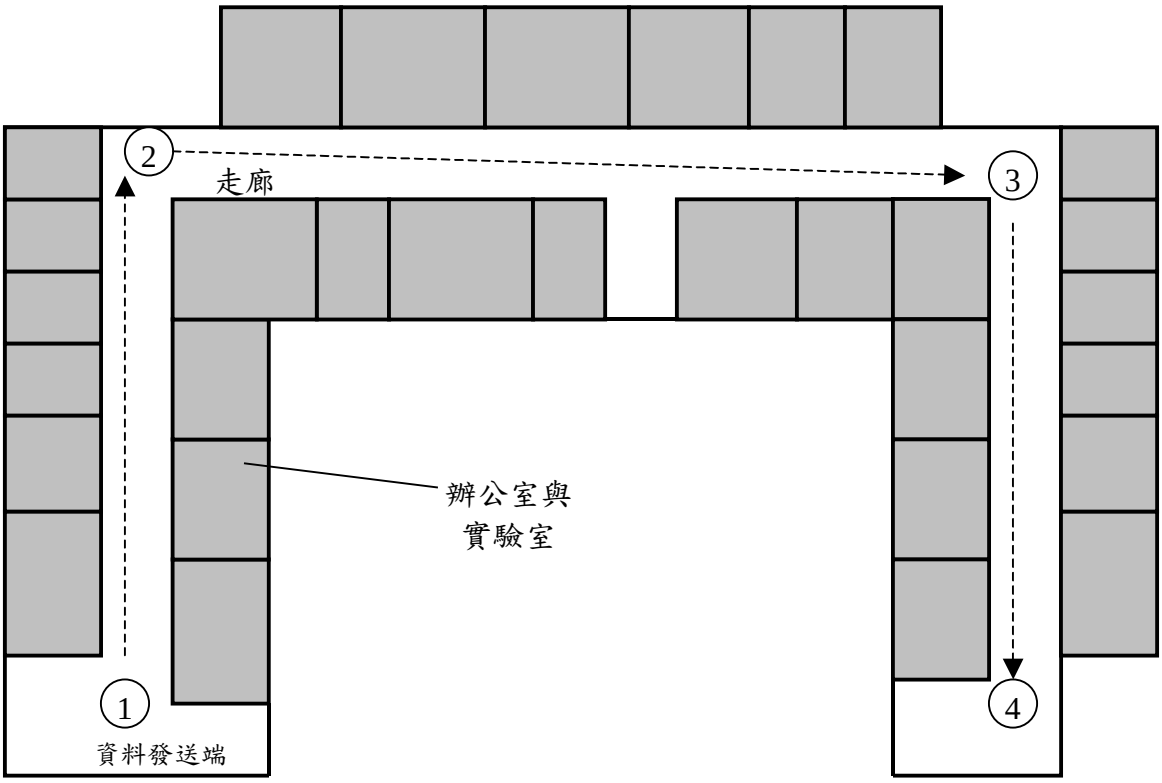


圖 2.8 室內多跳接傳播實驗環境

(a)倚靠欄杆支撐設置方式　　(b)於走廊中央設置方式

圖 2.9 室內量測之設備安裝

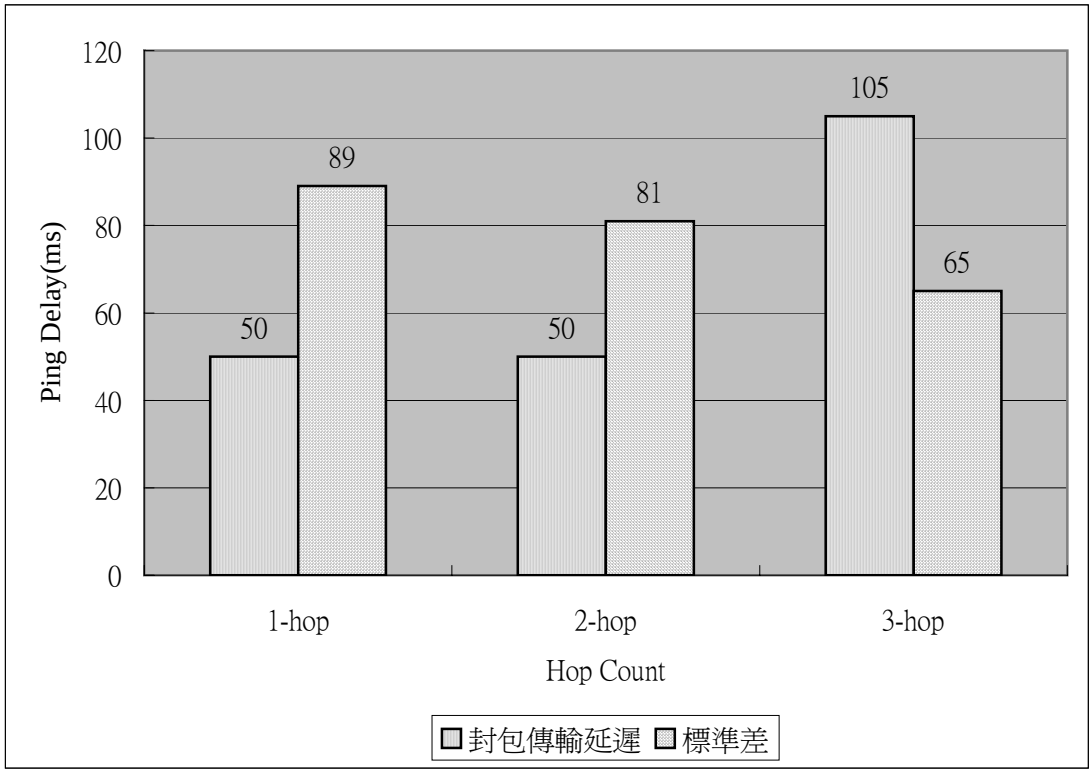


圖 2.10 室內環境實驗接跳數與封包傳輸延遲之關係

[實驗 2-2]

目的: 探討單一跳接的室外傳播環境對傳輸封包延遲的影響

進行步驟:

本實驗主要是在交通大學室外環境進行，利用臨時建置的固定節點與安裝於量測車輛之移動節點，進行在不同移動速率與傳播距離條件下，資料傳輸封包延遲之效能表現。圖 2.11 呈現本實驗固定點與移動點之設備安裝方式。

(a)利用腳架搭設臨時固定點 (b)可穩固於車外安裝MANET設備的移動點

圖 2.11 校園道路室外量測設備

本實驗主要的量測路徑與範圍包括 3 條主要路線，參照圖 2.12。路線 1 為具備視線內傳播特性之校園道路，道路周邊主要為操場、球場、以及低矮的建築物，雖然屬於視線內傳播，但由於該路段具有微幅高低起伏，因此對傳輸效能仍具有影響。路線 2 為稍微曲折之校園道路，道路周邊主要為約 4~5 層樓高之建築物，路線 2 在傳播環境上屬於輕度遮蔽。路線 3 為靠近路線 2 之反向路徑，由於該側具有較密集的樹木，在無線電傳播上產生較嚴重的遮蔽，道路周邊同為 4~5 層樓高之建築物，路線 3 在傳播環境上屬於重度遮蔽。圖 2.13 則標示了量測週邊環境以及資料發送端之位置。

圖 2.12 校園道路室外量測路徑

圖 2.13 室外量測周邊環境

結果與分析:

圖 2.14 顯示在路線 1 量測封包傳輸延遲之結果。一般而言，在 1-hop 傳播條件下封包延遲時間很短，除非遭受到傳播環境的遮蔽，大部分的封包傳輸延遲低於 100ms，總平均為 62ms。

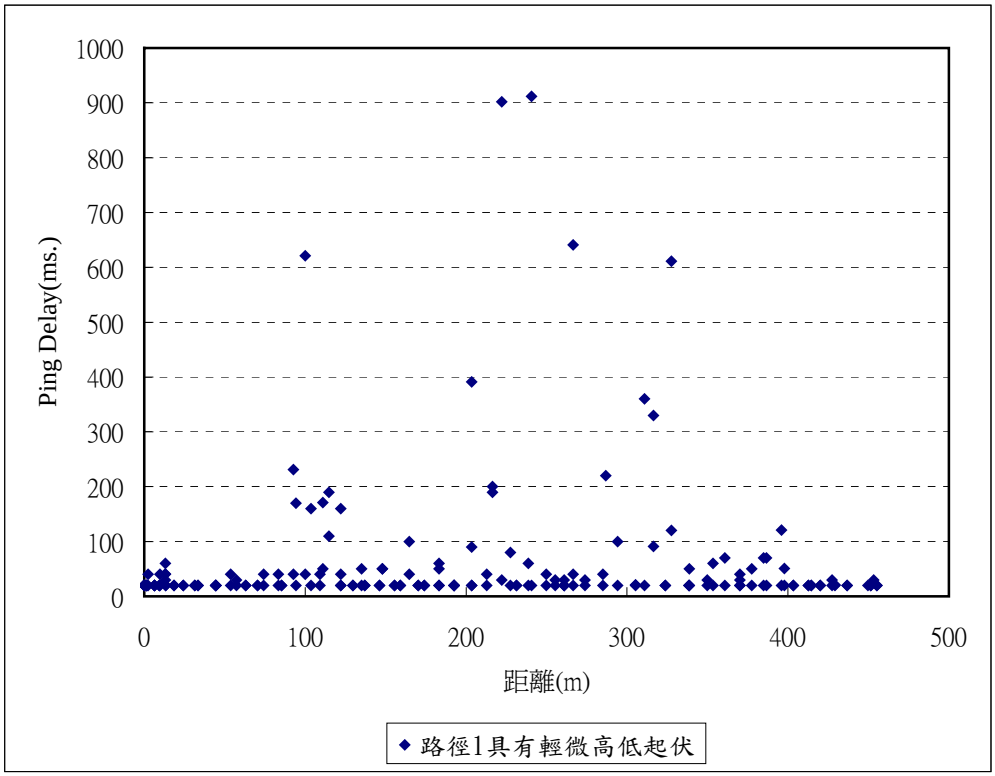


圖 2.14 路徑 1 封包傳輸延遲的量測結果

圖 2.15 顯示非視線內傳播在封包傳輸延遲之量測結果。從量測數據來觀察，傳播距離較長的區域會造成封包延遲時間較大的機率增加，然而大部分的延遲仍低於 100ms，路線 2 與路線 3 的封包傳輸延遲平均分別為 67ms 與 69ms。

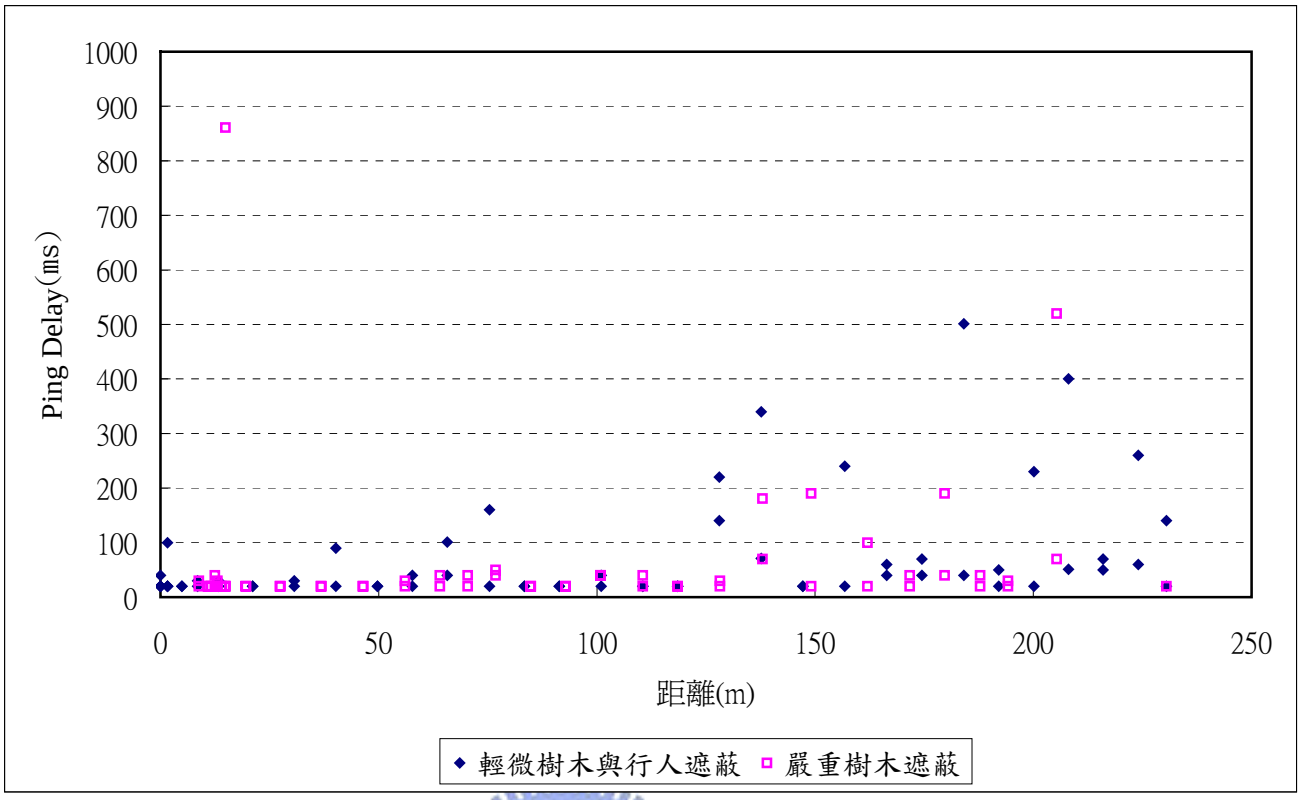


圖 2.15 有遮蔽路徑的封包傳輸延遲量測結果

根據本實驗在不同量測路徑下的測試結果可知，傳播環境的遮蔽程度對基於 WLAN IEEE 802.11b 之傳播通道有明顯的影響。尤其對資料傳輸速率的影響特別明顯，而對於封包延遲時間的影響較不顯著。此外，在如同校園環境之郊區道路進行行動通訊時，在網路節點相對速度不大的情形下，相對速度對傳輸效能的影響有限。表 2.1 列示本實驗各項傳播環境在節點間距 300m 內之資料傳輸率與封包延遲時間。

傳播環境	平均資料傳輸速率 (kBytes/sec)	平均封包延遲時間 (ms.)
路徑 1 (輕微高低起伏、相對速度高)	99.65	60
路徑 1 (輕微高低起伏、相對速度低)	103.31	62
路徑 2 (輕微遮蔽、相對速度高)	89.42	67
路徑 2 (輕微遮蔽、相對速度低)	88.16	67
路徑 3 (行道樹遮蔽、相對速度高)	51.46	68
路徑 3 (行道樹遮蔽、相對速度低)	42.24	69

表 2.1 室外環境進行單一跳接數傳播之傳輸效能比較

[實驗 2-3]

目的: 探討在多段跳接數(multi-hop)傳播條件下，室外傳播環境對傳輸封包延遲的影響。

進行步驟:

本實驗是在校園道路所進行之室外多跳接傳播，乃是利用具有輕度遮蔽、以及密度較低之人車移動體遮蔽之道路，並於道路上適當距離設置臨時固定之網路節點，以控制傳播時所經過之跳接數。如圖

2.16 所示，固定節點位置如圖所標示之 1、2、3 位置，節點 1 負責發送量測用資料封包，量測車輛行駛至適當位置以進行不同跳接數的資料接收並記錄量測結果。

圖 2.16 校園道路室外多跳接實驗之量測路徑

結果與分析:

圖 2.17 列示本實驗封包傳輸延遲的在不同接跳數下之效能變化。由圖可知，隨著跳接數的增加，封包傳輸延遲發生顯著的增加。另一方面，封包傳輸延遲的標準差數值都相當高，甚至超過封包傳輸延遲之量測平均數值，由量測數據觀察可知，高標準差導因於若干延遲時間特別高的量測記錄，由此現象可知傳輸通道的不穩定對封包傳輸延遲的效能影響嚴重。

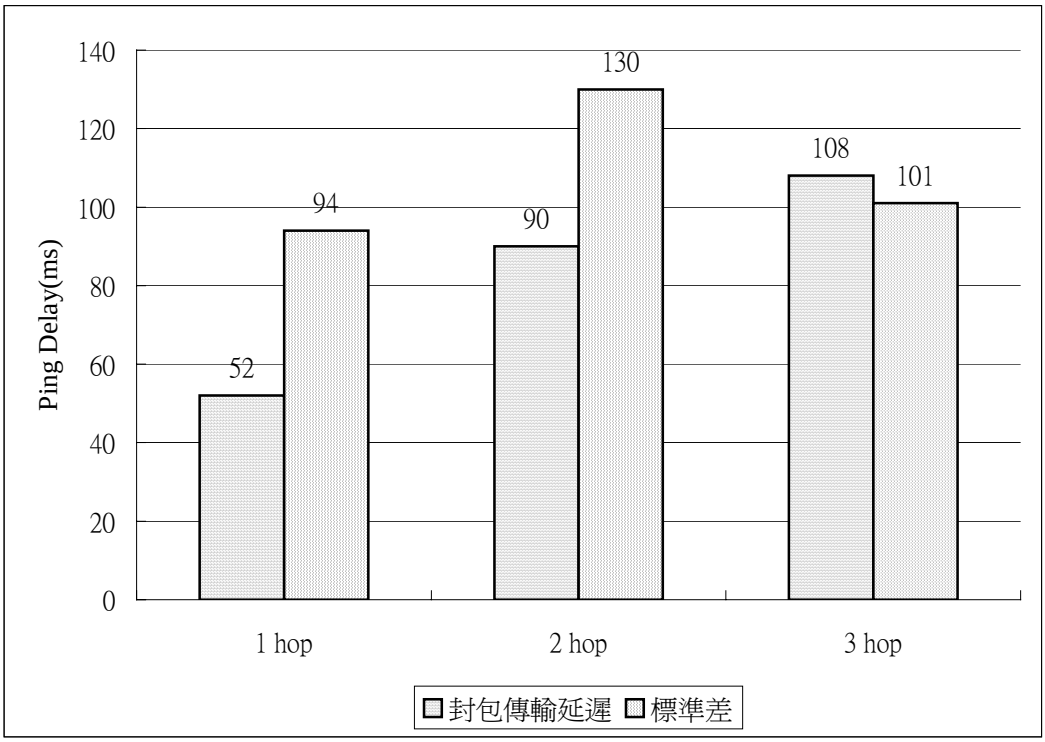


圖 2.17 室外環境跳接數目與封包傳輸延遲之關係

第三章 播放排程演算法

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。但網路電話仍存在許多問題，通常會遇到的問題為整體延遲(end-to-end delay)、延遲顫動(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一個封包，所以封包到達時的延遲和封包漏失，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間隔來產生封包並透過網際網路傳送到接收端。每個封包的網路延遲會取決於所走的路徑及該路徑上路由器的擁塞程度而有所不同，而這些網路延遲的差異即為延遲顫動。為降低延遲顫動在接收端的影響，接收封包在播放前會先被暫存在一緩衝器中一小段時間；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；然而封包的整體延遲將會增加。因此，本論文的研究重點是在不同的傳輸模式下,可能是單一路徑傳輸或多重串流傳輸，最終目標在封包的漏失率及平均整體延遲之間找一個權衡點。

本章將說明藉由播放緩衝器來降低延遲顫動的影響，首先介紹在

接收端播放緩衝器的角色;接下來探討目前相關研究中三種主要的播放排程演算法(playout scheduling algorithm)，同時針對網路延遲的特殊現象 Spike 作調整。最後結合多重串流傳輸技術，使演算法克服網路中種種負面因子，達到強健性的網路傳輸。

3.1 播放緩衝器簡介

如圖 3.1，語音信號以固定的時間間隔 L 產生封包並經由網路傳送。因為網路本身的特性，每個封包延遲並不會固定，導致有些封包會在語音預定的播放時間之後才到達。圖 3.1(a)說明了延遲顫動所造成的問題，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$(i-1)L$ 作為播放時間。然而，較大的網路延遲會造成晚到的封包無法順利播出，導致部分的封包漏失而降低通話品質。因此本論文使用的解決方案是加入播放緩衝器，如圖 3.1(b)，將封包暫存於緩衝器一小段時間再播放。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 3.2 說明三種基本播放緩衝器設計原理，其中語音封包的網路延遲以黑點表示，整體延遲以實線表示。當排定的播放時間越晚，整

體的延遲就越大。封包在播放時間之後到達，例如黑點在實線之上，即判定為漏失。這些播放排程設計的理想目標是盡可能使實線降低(降低整體延遲)，同時使黑點在實線以上的數目越少越好(晚到漏失最小化)。

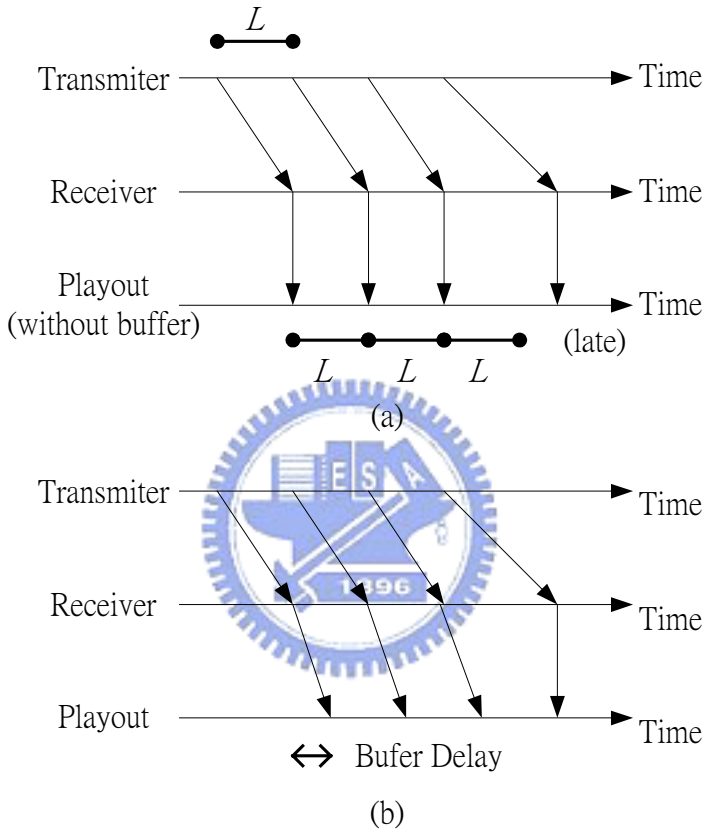


圖 3.1 播放緩衝器的影響

最簡單的方法，如圖 3.2(a)所示，對每一個語音封包皆使用固定的整體延遲，但這方法並不能有效的使延遲和漏失同時降低。主要是因為網路的特性隨時間而改變，而固定的整體延遲並不能反映這些變化。因此，較合理的播放時序演算法是參考網路延遲的變化，動態地調整不同區間的整體延遲時間。因為基於對話的特性，聲音會被靜

音(silence)區間區分為數個話務(talkspurt)區間。靠著延長或壓縮靜音區間的長度來調整個別話務的播放時間，圖 3.2(b)可看出比圖 3.2(a)的結果好。這種方法是以話務為單位作調整，會因為話務區間太長及在一個話務中網路延遲變化太大而使其效力受限。舉例來說，在第三個話務內的第 113、114、115 個封包有瞬間很大的網路延遲，也是一般常見的 Spike 發生。此種方法並不能適應這種情形，導致許多封包漏失造成聽覺上聆聽品質的降低。

最理想的播放排程演算法，應該是不只調整靜音區間也可針對話務內個別封包做調整。每個獨立的封包都可根據變動的延遲統計來決定其不同的排定播放時間，其結果可由圖 3.2(c)表示。這演算法可以透過動態性及反應較佳的方法來有效的調整播放時間同時降低漏失率及平均延遲。但特別強調的是，這種以封包為單位作調整的演算法，為了避免最後的聲音被不自然的延長或縮短，必須配合音長調整(time-scaling)以保持連續的播放效果。

長期觀察網路延遲的實地量測數據，可發現許多網路延遲皆存在 spike 現象。若一封包突然有網路延遲大幅度增加的現象，視為 spike 的開始點，雖然接下來的封包通常網路延遲都會遞減，但它們相對於

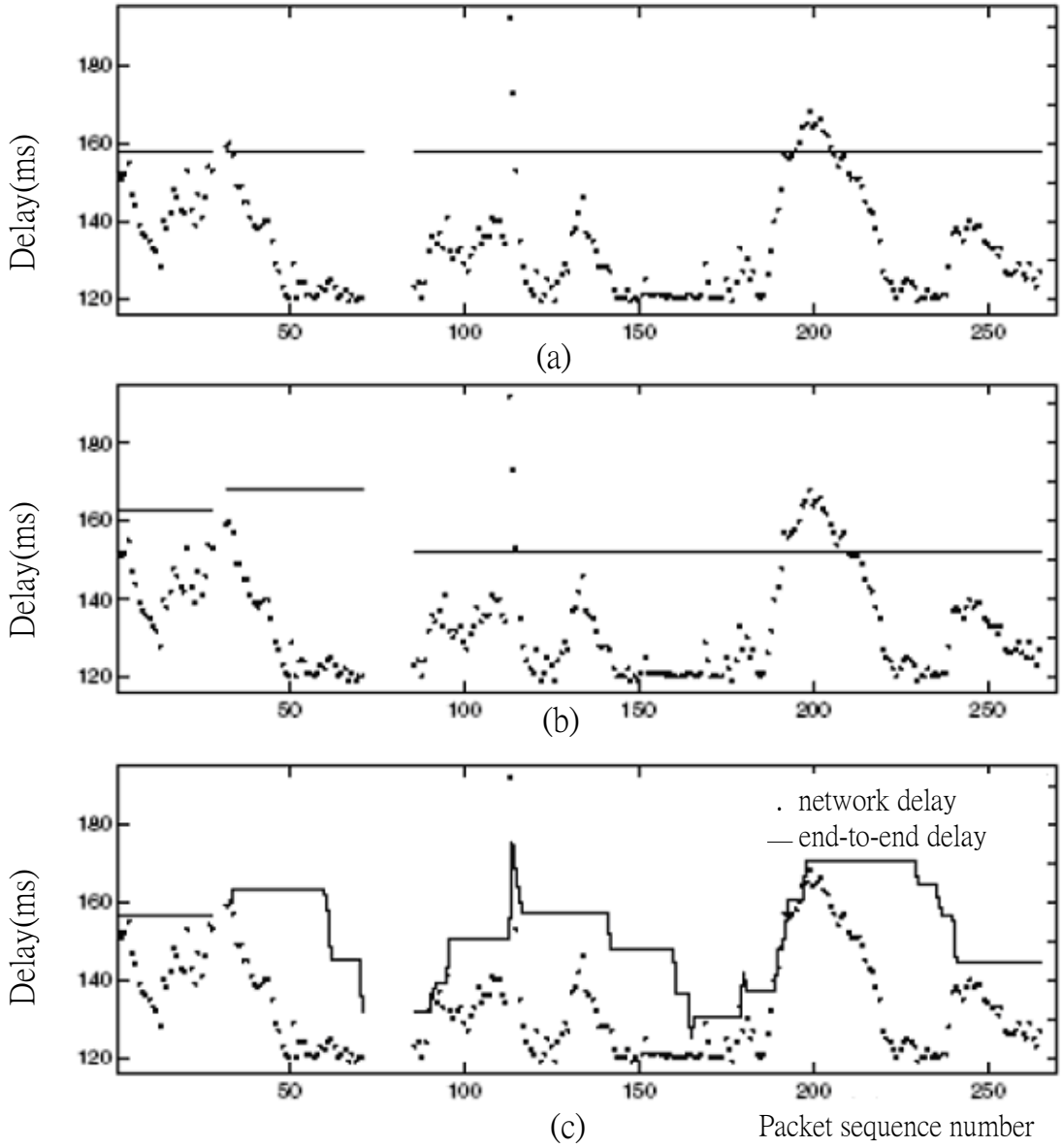


圖 3.2 三種播放排程演算法

其他封包的延遲仍然很大。當網路延遲回到一個穩定狀態值時，則視為 spike 的結束。網路延遲的 spike 是由於網路路由器突然發生大量壅塞的排隊現象造成，典型的延遲 spike 如圖 3.3 所示，發生在約第 5665 個封包。因此，近一步使播放排程演算法更有效，必須配合加入 spike 偵測機制作合理規劃。

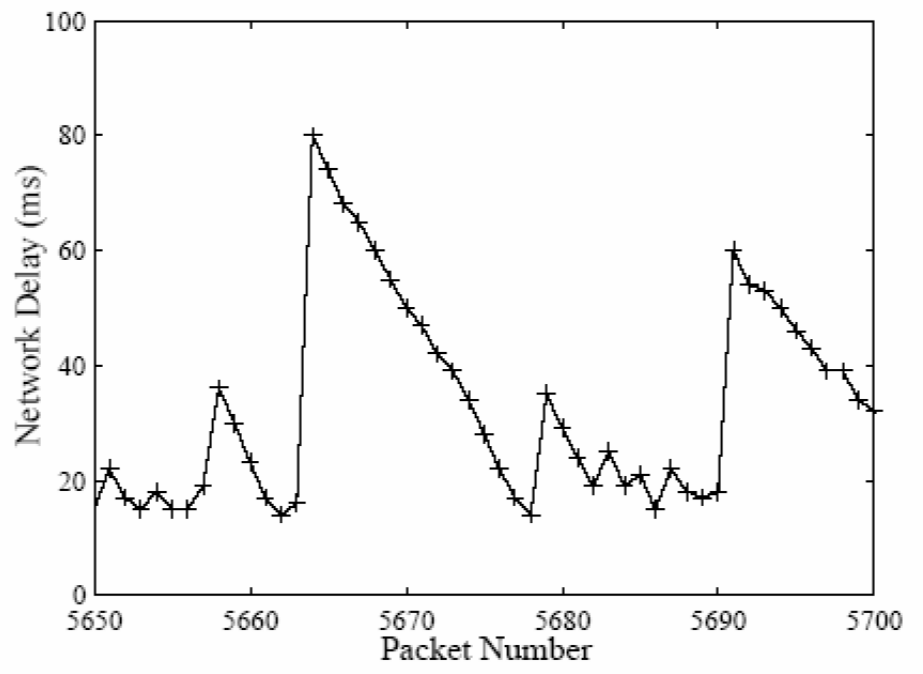


圖 3.3 典型網路 spike 現象

3.2 播放緩衝器效能分析

在前一節中，關於播放時序設計的目的及其適應性調整的好處已經有概念性的描述，接下來是定義一個可量化的效能分析方法。先介紹本論文會用到的一些基本標號，然後再定義兩項效能評估參數：”平均緩衝延遲(average buffering delay)”及”晚到漏失率(late loss rate)”。

語音以固定長度編碼後，封裝成固定大小的封包傳送出去，每個封包的間隔為 L。如圖 3.4 所示，下標 $i=1,2,\cdots,N$ 表示封包的次序號碼，假設總共有 N 個封包被傳送。接下來介紹一些基本標號：

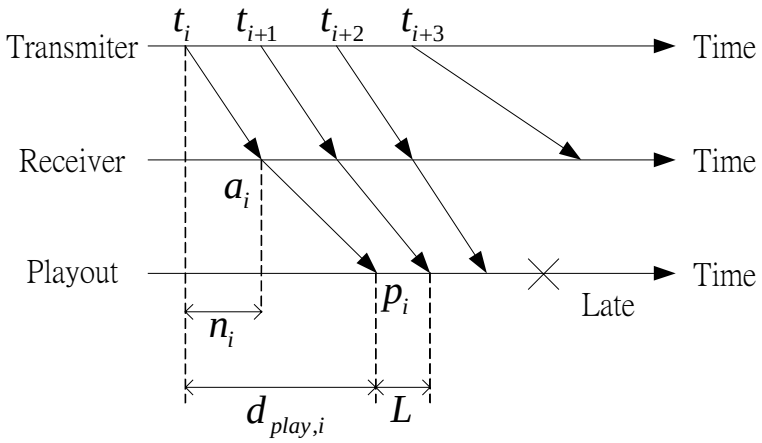


圖 3.4 第 i 個封包的相關時間參數

t_i：第 i 個封包的傳送時間。

a_i：第 i 個封包的接收時間。

p_i：第 i 個封包的播放時間。

$d_{play,i}$：第 i 個封包從傳送直到播放所經過的時間差，$p_i - t_i$，即為第 i 個封包的播放延遲(playout delay)。

n_i：第 i 個封包因傳輸所造成的網路延遲，$a_i - t_i$。網路延遲包含兩部分：一為傳送端到接收端的固定傳播延遲(propagation delay)，另一則為傳送過程的排隊延遲(queueing delay)。

當我們比較不同的播放排程演算法時，主要針對兩個數值作效能評估。描述如下

1. 平均播放延遲，如下所示：

$$d_{play,average} = \frac{1}{\left|N_{playout}\right|} \sum_{i \in N_{playout}} d_{play,i} \qquad (3.1)$$

其中 $N_{playout} = \{i \mid d_{play,i} \geq n_i\}$，為可被播放出來封包的集合，$|N_{playout}|$ 為此集合的個數。

2. 晚到漏失率，定義如下：

$$e_b = (N - |N_{playout}|) / N \qquad (3.2)$$

其中 N 為所有傳送封包的數量。這兩個數值可反映出上面提及的漏失及延遲的取捨衡量，也可用來比較不同播放排程演算法的效能分析。本論文的研究重點是考慮人耳聽覺效應的音質預測模型，針對這兩個參數作最佳化取捨，進而設計出最佳通話品質的播放排程演算法。

3.3 適應性播放演算法

適應性播放時間的調整方式主要分成兩大類，第一類為每一個話務作一次調整(per-talkspurt)，第二類則為每一個封包都做調整(per-packet)。第一類的調整方式主要著重在各個話務的第一個封包，描述如下：

- 如果第 i 個封包為第 k 個話務的第一個封包，它的播放時間 p_i 可以下式計算：

$$p_i = t_i + D^k \qquad (3.3)$$

其中 D^k 為第 k 個話務所屬個別封包的固定播放延遲。

- 在第 k 個話務接下來的封包，其播放延遲皆和第一個封包相同。

所以若第 j 個封包存在於第 k 個話務內，其播放時間可定義為：

$$p_j = t_j + D^k = t_j + p_i - t_i \qquad (3.4)$$

第二類的調整方式則是針對話務內每個封包都做調整，所以每個語音封包在播放前必須作音長調整的處理，使它們可以剛好在下一個封包的預測播放時間之前完整播出。動態的調整播放時間使封包漏失維持在其容忍範圍下，同時降低整體延遲進而有效的改善系統通話品質。接下來說明本論文主要用到的播放延遲估計方法，其中關鍵的適應性濾波器[3][4]屬於第二類(per-packet)的演算法。

適應性濾波器演算法通常是用在等化器及回音消除器上，主要目的為使實際的數據和估計值之間的均方差期望值最小化。過去的數據被暫存在一有限脈衝響應(FIR)濾波器中以用來計算現在的估計值，均方差準則用來調整適應性濾波器的係數向量。最近提出的一種演算法[4]，直接用適應性濾波器演算法來預測網路延遲，而播放延遲就以網路延遲的預測值及其變異數來決定。若能準確預測網路延遲，可以快速追蹤到網路流量的變化，因此能夠更有效率地調整延遲。

在本論文中，採用正規化最小均方(Normalized least mean square, NLMS)演算法來做適應性預測。第 i 個封包的網路延遲預測值可用下式表示：

$$\hat{n}_i = \overline{w}_i^T \overline{n}_i \qquad (3.5)$$

其中 $\overline{w}_i$ 為 *Mx1* 維度的適應性濾波器係數向量，$\overline{n}_i = \{n_{i-1}, n_{i-2}, ..., n_{i-M}\}$ 為包含過去 *M* 個網路延遲量測值的向量。

濾波器的係數向量根據 NLMS 演算法作更新的動作：

$$\overline{w}_{i+1} = \overline{w}_i + \frac{\mu}{\overline{n}_i^T \overline{n}_i + b} \overline{n}_i e_i \qquad (3.6)$$

其中 μ 為階層(step size)大小，b 為一個很小的常數。而估計誤差表示為：

$$e_i = n_i - \hat{n}_i \qquad (3.7)$$

其中 n_i 及 $\hat{n}_i$ 分別為第 i 個封包網路延遲的實際值與預測值。至於網路延遲的變異數 $\hat{v}_i$ 及播放延遲 $d_{play,i}$，則用自迴歸(autoregressive, AR)演算法計算，如下：

$$\hat{v}_i = \alpha \hat{v}_{i-1} + (1-\alpha)\left|\hat{n}_{i-1} - n_{i-1}\right| \qquad (3.8)$$

$$d_{play,i} = \hat{n}_i + \beta_i \hat{v}_i \qquad (3.9)$$

其中 α 為加權參數用以控制演算法之收斂速度。

圖 3.5 顯示了不考慮 spike 偵測功能的 NLMS 演算法的預測結果。在 spike 發生時，由於延遲突然變大，通常第一個封包都會漏失，而接下來的封包預測將迅速追上變大的值。但在 spike 區間，預估的延遲(3.9)內含安全緩衝項 $\beta \hat{v}_i$，導致在 spike 區間的整體延遲過大。

在正常模式下，用 NLMS 演算法配合安全緩衝係數 β 來預測延遲。在 spike 模式下，仍用 NLMS 演算法來預測延遲，但安全緩衝係數降低為 $\beta/4$，可簡單且有效的降低整體的延遲。此外又加入一個條件，為了不讓播放延遲下降的太快，以自迴歸演算法所得到的延遲(3.10)當作延遲估計的下限。

$$AR\_delay_i = \alpha AR\_delay_{i-1} + (1-\alpha)n_{i-1} \qquad (3.10)$$

而當 NLMS 演算法預測出來的延遲不再大於實際的網路延遲，即結束 spike 模式回到正常模式。圖 3.6 顯示加上 spike 偵測的 NLMS 演算法的結果，可以有效地降低整體延遲。

3.4 多重敘述編碼架構

網際網路的特性為經常變異的網路延遲以及封包漏失，後者的解決方法是向前錯誤更正(FEC, Forward Error Correction)，加入額外的保護位元在原來的封包上。雖然此法可在接收端還原有限的錯誤位元，但卻增加封包的整體延遲。此外，網路常發生叢發性的封包漏失(burst packet loss)，導致 FEC 的錯誤修正功效減低[5]。

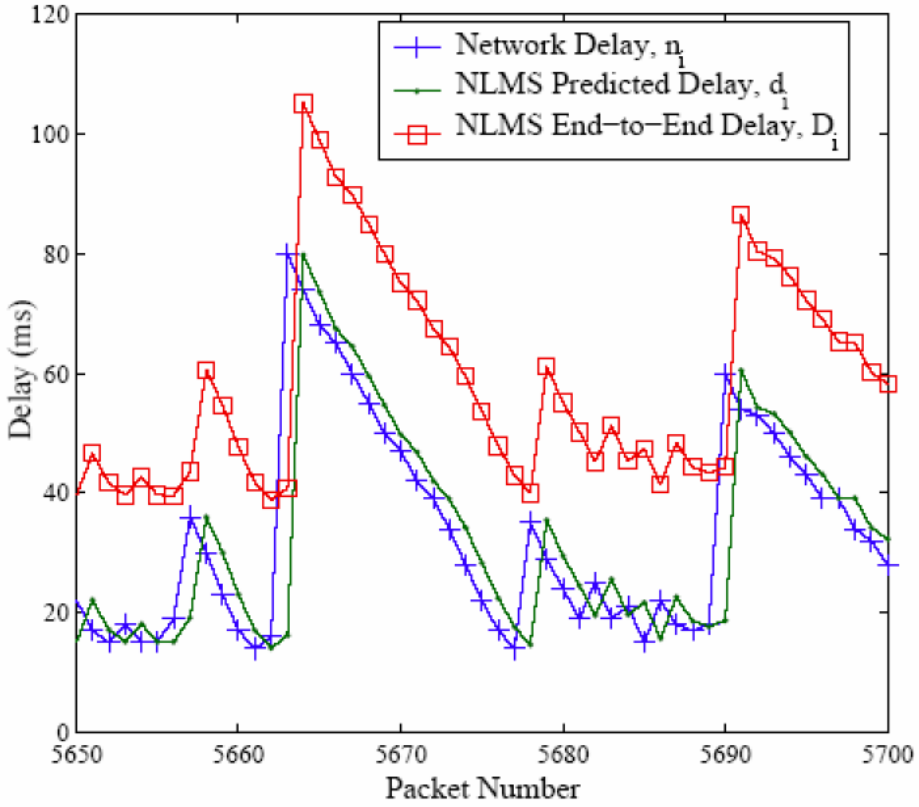


圖 3.5 NLMS 播放演算法

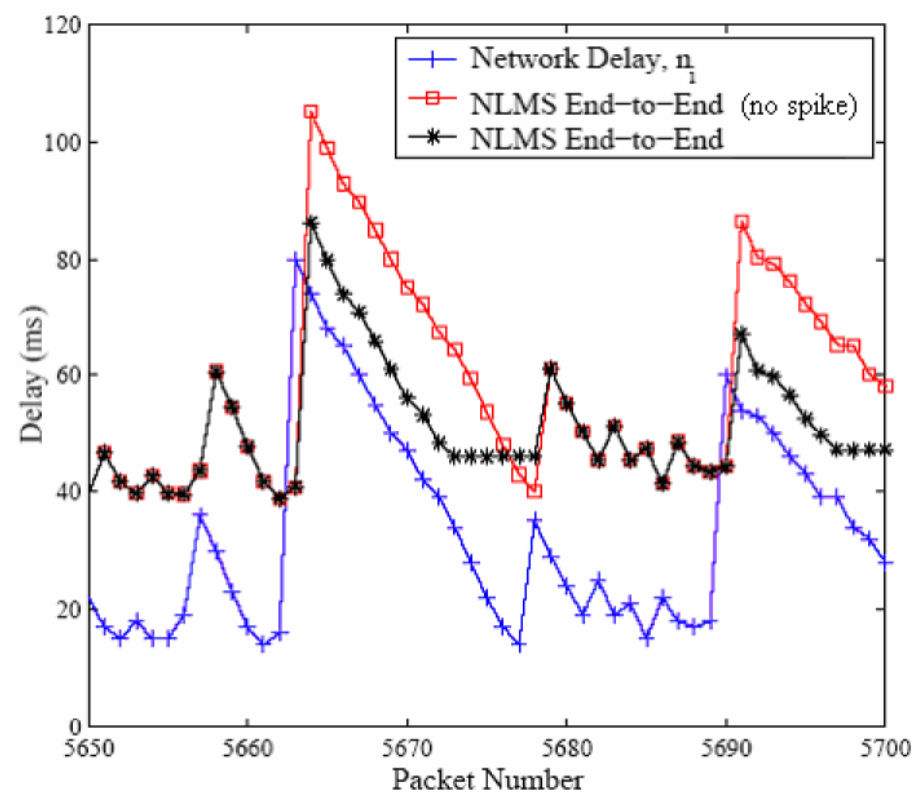


圖 3.6 spike 偵測對 NLMS 演算法的影響

為了提供更強健性的語音網路傳輸，有別於單一路徑的封包傳送模式，本論文引用多重串流(multi stream)傳輸系統作封包的傳遞。使用兩個路徑作傳輸，並假設在統計上兩條傳輸路徑沒有關聯，因此同時發生負面因素影響的機率遠低於單一路徑傳送時的影響。另外，由於路由(routing)的機制通常不能找到最佳傳輸路徑[6]，多重串流傳輸系統可降低此問題的影響。因為在兩條傳輸路徑中只要找到一條路徑，即可藉由封包重建的方法還原出可令人接受的聲音品質。[7]

多重串流傳輸系統是先進行多重敘述編碼(multiple description coding, MDC)處理，再將語音封包分別經由不同的路徑傳送，以期提昇整體延遲與封包漏失之間的權衡。當每個串流封包的網路延遲都小於播放延遲時，可用較高的延遲成本換取一高品質的重建訊號；收到單一串流時，則只能還原部份的原始訊號。至於封包漏失的重建機制，本論文採用內插法。如圖 3.7 所示，當第 i 個偶數(或奇數)封包漏失時將，第 i 個奇數(或偶數)封包與其前一個封包結合後重建已經漏失的封包。多路徑串流傳輸既可改善單一路徑封包漏失的問題，又能避免向前錯誤更正碼所需傳送的額外保護位元。

其播放排程機制，目的在設定一播放延遲 $d_{play,i}$ 使其成本函數 $f(d_{play,i}) = d_{play,i}$ ，在一個限制集合下能達到最小值。就人耳的主觀聽覺效應而言，相較於一個完整的重健語音訊號，一個冗長的播放延遲更

讓人難以忍受。因此一個合理的限制集合可設定如下

$$\Omega_i = \{d_{play,i} : d_{play,i} \geq \hat{D}_i^{S_1} \cup d_{play,i} \geq \hat{D}_i^{S_2}\} \qquad (3.10)$$

其中 $\hat{D}_i^{S_k} = \hat{n}_i^{S_k} + \beta \hat{v}_i^{S_k}$ 是在第 k 個串流中第 i 個封包的延遲估測值，而 $\hat{n}_i^{S_k}$ 與 $\hat{v}_i^{S_k}$ 則分別是該封包網路延遲平均值與變異數之估側值。本論文將採用 NLMS 演算法估測值其平均值 $\hat{n}_i^{S_k}$。首先定義一組過去 N 個封包的網路延遲記錄 $\overline{n}_i^{S_k} = [n_{i-1}^{S_k}, n_{i-2}^{S_k}, ..., n_{i-N}^{S_k}]^T$，經由本論文先前提到的 FIR 濾波器估算其網路延遲平均值 $\hat{n}_i^{S_k} = (\overline{w}_i^{S_k})^T \overline{n}_i^{S_k}$。相關濾波器的係數調整，利用下列疊代(iterative)公式求得

$$\overline{w}_{i+1}^{S_k} = \overline{w}_i^{S_k} + \frac{\mu}{(\overline{n}_i^{S_k})^T \overline{n}_i^{S_k} + b} \overline{n}_i^{S_k} e_i^{S_k} \qquad (3.11)$$

其中μ是一個固定大小的步階，b 為一個很小的常數，而估測誤差值 $e_i^{S_k} = n_i^{S_k} - \hat{n}_i^{S_k}$。至於延遲變異數的相關估測，則採用自回歸演算法：

$$\hat{v}_i^{S_k} = \alpha \hat{v}_{i-1}^{S_k} + (1-\alpha) | \hat{n}_{i-1}^{S_k} - n_{i-1}^{S_k} | \qquad (3.12)$$

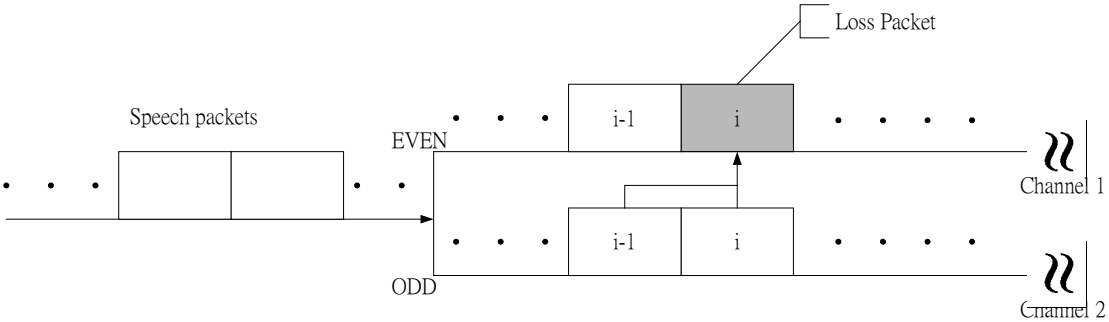


圖 3.7 多重串流的封包漏失補償

第四章 播放排程的聽覺最佳化設計

第三章已經介紹了基本的播放排程演算法，其目的都是個別考慮較低的漏失率或較短的平均播放延遲，然而漏失率與平均播放延遲兩者間存在著一個取捨(trade-off) 的關係。由後續本章所推導出來的通話品質預測模型可以發現，在單一路徑口對耳的延遲d超過177.3msec以上時，隨著d的增加，延遲損害因子增加的程度比在d小於177.3msec時來的嚴重。另一方面，封包漏失損害因子I_{epl}是一個封包漏失率(Packet Loss Rate)的函數，且其函數行為是一個對數函數。所以在封包漏失率低時，其損害因子的變化程度比起封包漏失率高時的變化來的大。先前介紹的演算法目的雖然是為了降低封包漏失率以及降低平均播放延遲，卻未針對網路變化的特性，考慮封包漏失損害因子與延遲損害因子之間的權衡，因而無法達到一個最佳的音質效果。

本章將適度修改先前使用固定β值的播放排程演算法，在設置播放延遲時考慮音質效果，目的是希望能夠因應時變的網路特性，適度的去權衡封包漏失率以及平均播放延遲之間的關係，進而動態地調整安全因子β值。

4.1 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，因此人們利用網路當作聲音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，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際網路通話的音質評量指標模型。

4.1.1 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[8][9][14]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。欲測試的語句會預先在另外一間週遭噪音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並改善。

4.1.2 音質評量指標

正因為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，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

$$R = R_0 - I_s - I_d - I_e + A \qquad (4.1)$$

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 4.1 所示。其關係式為

$$MOS = \begin{cases} 1, R < 0 \\ 1 + 0.035R + 7 \times 10^{-6} R(R-60)(100-R), 0 < R < 100 \\ 4.5, R > 100 \end{cases} \quad (4.2)$$

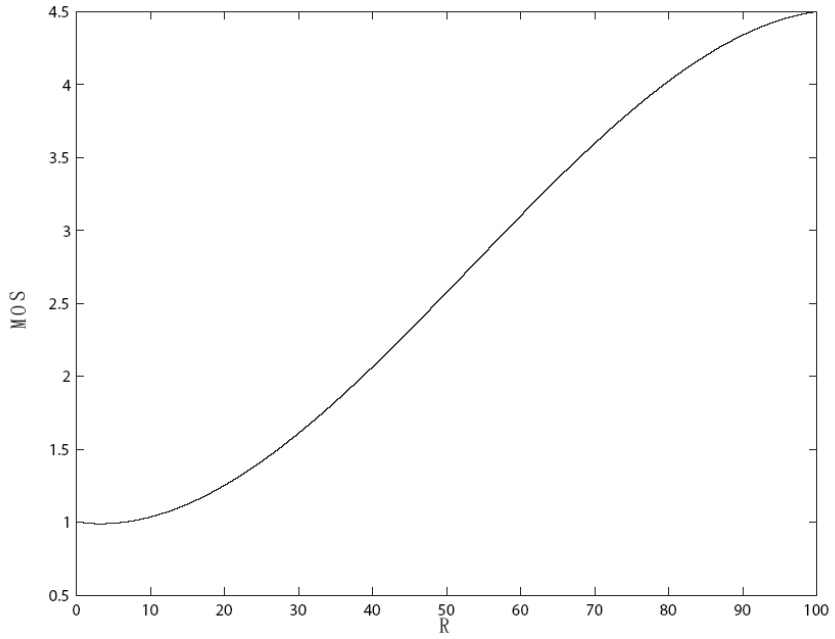


圖 4.1: R 與 MOS 的轉換關係

我們將*R*與平均評比分數的對應關係整理成表4.1:

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 4.1 *R*與平均評比分數對應關係

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，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[10]。如(4.3)所示

$$R(d,r,e) = 94.2 - I_d(d) - I_e(r,e) \qquad (4.3)$$

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

$$I_e(r,e) = I_{ec}(r) + I_{epl}(e) \qquad (4.4)$$

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。

[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。其中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。每一種編碼標準均有其特定的聆聽 MOS(listening MOS)，利用圖 4.1 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(4.3)可簡化為

$$R(r) = 94.2 - I_{ec}(r) \qquad (4.5)$$

由前人研究得知，隨著編碼位元率的下降，音質的損害值明顯的增加。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_{epl}

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通道行為影響的音質損害。在網路中常常面臨到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。若是資料封包的傳送可以使用要求重送(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損害因子I_e可近似為一個數學公式，

$$I_e(r,e) = I_{ec}(r) + I_{epl}(e) = \gamma_1 + \gamma_2 \ln(1+\gamma_3 e) \qquad (4.6)$$

其中$I_{ec}(r) = \gamma_1$，$I_{epl}(e) = \gamma_2 \ln(1+\gamma_3 e)$，而不同的語音編碼模式會對應一組$\gamma_1$，$\gamma_2$，$\gamma_3$[10]， 如表4.2所示。

Codec Type	γ_1	γ_2	γ_3
G.729a	11	40	10
G.711	0	30	15

表 4.2 不同語音編碼的γ_i

[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延遲、傳送路徑延遲、播放暫存器造成的延遲。

在前人研究[10]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

$$I_d = 0.024d + 0.11(d - 177.3)H(d - 177.3) \qquad (4.7)$$

其中d為單一路徑延遲，而$H(.)$是一個步階函數。

4.2 音質最佳化的播放排程機制

4.2.1 音質最佳化的設計

第二章介紹了兩個基本的播放排程演算法，可以發現第i個語音封包的播放延遲，$d_{play,i}$，是由網路延遲的兩個統計特性來估算，一個是該封包的平均延遲預估值$\hat{n}_i$，一個是該封包的變異數預估值$\hat{v}_i$。因此第i個語音封包的播放延遲可由下式表示:

$$d_{play,i} = \hat{n}_i + \beta_i \hat{v}_i \qquad (4.8)$$

在傳統的演算法中，β_i並不會隨著封包i而改變也就是$\beta_i = \beta$。對於

第i個語音封包而言，一個較大的β_i導致一個較長的播放延遲。從而決定較低的封包漏失機率。為了達到音質最佳化的播放延遲，β_i的調整必須取決於過去的網路延遲資訊與音質預測模型。

對於一個播放延遲的音質最佳化預估，目標就是達到一個最佳的MOS值，等同於考慮一個最小的音質損害。首先我們定義音質損害因子$I_{m,i}$，此因子是一個d_i和e_i的函數，由(4.4)(4.6)(4.7)可得知

$$I_{m,i}(d_i,r,e_i)=I_d(d_i)+I_e(r,e_i)=I_d(d_i)+I_{ec}(r)+I_{epl}(e_i) \quad (4.9)$$

$$=0.024d_i+0.11(d_i-177.3)H(d_i-177.3)$$

$$+\gamma_1+\gamma_2\ln(1+\gamma_3 e_i)$$

其中d_i代表第i個語音封包的整體延遲，$d_i=d_c+d_{play,i}$，d_c代表語音編碼所造成的延遲時間。另外，$e_i=e_n+(1-e_n)e_{b,i}$代表為第i個語音封包漏失機率，這裡的e_n是該封包的網路漏失機率(Network Loss Probability)，$e_{b,i}$則是封包的到達時間比排程播放時間來的晚所造成的緩衝漏失機率(Playout Buffer Loss Probability)。

由(4.3)及(4.10)的關係式，可把第i個封包所對應的R值改寫為下式，

$$R=94.2-I_d(d_i)-I_e(r,e_i)=94.2-I_{m,i}(d_i,r,e_i) \quad (4.10)$$

由圖4.1可以得知MOS是評分值R的遞增函數，求MOS值的最大值等於

求R的最大值，由(4.9)可得知一個最大的R_i值對應一個最小的$I_{m,i}$。當d_i增加對應於I_d增加，此時e_i下降對應於I_e下降;相反的，當d_i減少對應於I_d減少，此時e_i上升對應於I_e上升。針對一特定的語音編碼方式而言， 為了達到$I_{m,i}$的最小值就必須選取一個最佳的播放延遲$d^*_{play,i}$值。其首要之務是先建立緩衝漏失機率$e_{b,i}$的統計模型。

4.2.2 緩衝漏失機率模型

此模型主要是利用網路延遲的累積分布函數來建立緩衝漏失機率，而一個延遲的累積分佈函數，$F_X(X)$，定義如下

$$F(x) = Prob\{X \le x\}$$

此式表示網路延遲不大於x的機率。第i個語音封包的播放緩衝漏失機率網路延遲大於其播放延遲$d_{play,i}$的時候，因此緩衝漏失機率$e_{b,i}$可定義為:

$$e_{b,i} = Prob\{X > d_{play,i}\} = 1 - F_x(d_{play,i}) \qquad (4.11)$$

上式建立了$d_{play,i}$與$e_{b,i}$之間的關係。觀察(4.11)，由機率的基本公設得知$F_x(d_{play,i})$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。

對於網路傳輸的封包延遲特性，前人研究了許多統計模型來描述，例如Exponetial模型及Pareto模型，其定義示於表4.3。

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

表 4.3 網路延遲的累積分佈函數

Exponetial模型參數$\{k_0, \mu\}$及Pareto模型參數$\{k, \alpha\}$，皆可由過去所量測到的N_L個網路延遲$\{n_{i-1}, n_{i-2}, ..., n_{i-N_L}\}$，依最大相似度預估理論(Maximum-Likelihood Estimation)計算而得。由前人研究可知估算公式如下[11]:

$$\hat{k}_0 = \min\{n_{i-1}, n_{i-2}, ..., n_{i-N_L}\}$$

$$\hat{\mu} = mean\{n_{i-1}, n_{i-2}, ..., n_{i-N_L}\} - \hat{k}_0$$

$$\hat{k} = \min\{n_{i-1}, n_{i-2}, ..., n_{i-N_L}\}$$

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

為了比較這兩種模型與網路延遲實際分佈的差異，首先利用前人研究的網路延遲模型產生兩組資料，平均延遲分別為55msce以及240msec，其中平均延遲較大這組具有較多的Spike現象。利用這些資料以及表4.3求出實驗的(Empirical)與模型化累積分佈函數，如圖4.2與圖4.3所示。

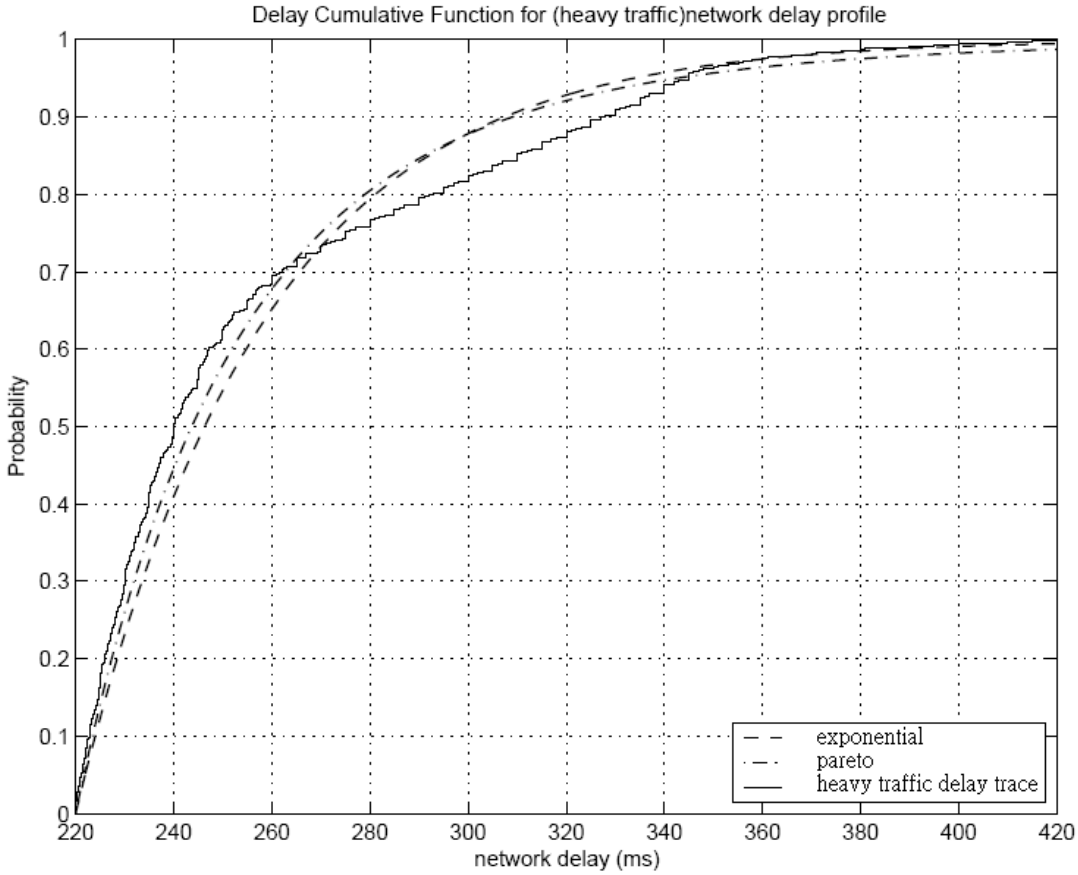


圖 4.2: 平均延遲為 240msec 的累積分佈函數

當平均延遲較低時，Exponetial 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 模型比 Exponential 模型更適用於描述網路的延遲特性。因此本論文將選用 Pareto 模型來計算語音封包網路延遲機率，藉此建立緩衝漏失機率與播放延遲的關係。

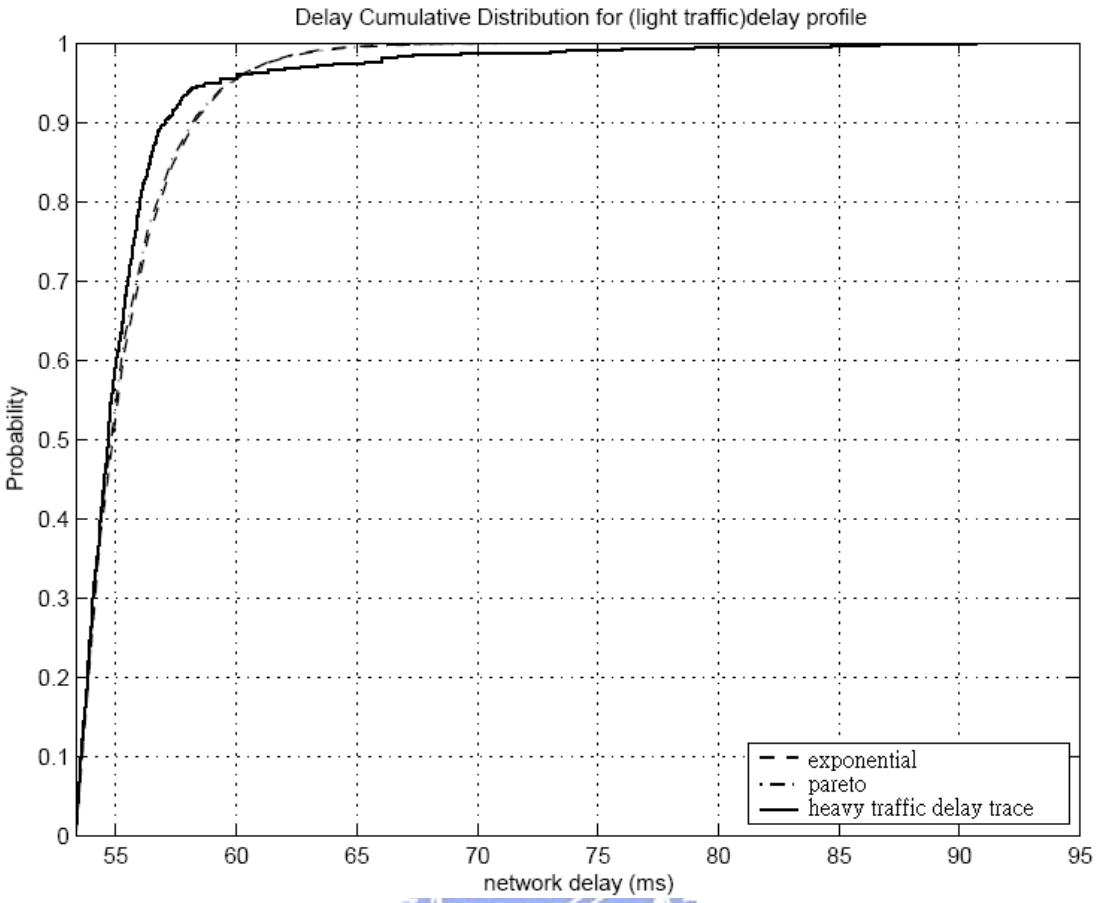


圖 4.3: 平均延遲為 55msec 的累積分佈函數圖形

4.3 安全因子的動態調整機制

由 4.2.1 節可知，播放延遲取決於$\hat{n}_i$以及$\hat{v}_i$，然而這兩個估計量是由過去的網路延遲資訊所決定。因此對於系統設計者而言，安全因子β_i扮演著權衡整體延遲與封包漏失率的重要角色。有別於過去的演算法中β_i設定為一個常數，本節將結合 4.2.1 節音質最佳化準則與 4.2.2 節的緩衝漏失機率建立一個新的演算法，可因應網路的延遲變化，動態調整β_i取得最佳的音質效果(即最低的損害因子)。

由(4.6)(4.7)式，可得知$I_{m,i}$可表示成一個β_i的函數。為了使音質損害最小，需要先計算$I_{m,i}(\beta_i)$相對於β_i的一階導函數(1^{st} order

derivative)如下式

$$\frac{dI_{m,i}}{d\beta_i} = 0.024\hat{v}_i + 0.11\hat{v}_i H(d_i - 177.3) \qquad (4.12)$$

$$+ \frac{\gamma_2\gamma_3(1-e_n)}{1+\gamma_3 e_i}\frac{de_{b,i}}{d\beta_i}$$

其中利用Pareto延遲模型可得

$$\frac{de_{b,i}}{d\beta_i} = -\alpha k^{\alpha}\hat{v}_i d_{play,i}^{-(\alpha+1)} \qquad (4.13)$$

4.4 最佳化的割線演算法

在一維空間搜尋的最佳化問題中，其挑戰在於尋找最小值的過程是否會收斂，以及搜尋時間是否過長。本論文採用一種迭代(Iterative)執行的割線演算法(Secant Method)，目標為找到最佳化的安全因子β_i^*使損害因子(Impairment Factor)的值最小。割線法起源自牛頓(Newton)演算法，其方程式如下

$$\beta_i^{(j+1)} = \beta_i^{(j)} - \frac{I'_{m,i}(\beta_i^{(j)})}{I''_{m,i}(\beta_i^{(j)})} \qquad (4.14)$$

相較於陡峭逼近方法(Steepest Descent Method)只使用一階導函數，牛頓法同時使用了1階與2階導函數的結果，使搜尋的過程更有效率；問題是當處理的函數並非正定義(positive-definite)時，牛

頓法不保證可以找到最小值。更明確地說，牛頓法(相同於割線法的要求)的理想工作環境為 2 階導函數(2nd Order Derivative)大於 0。另外，2 階導函數可能過於複雜甚至於無法計算。為了克服上述牛頓法遭遇的問題，割線法將牛頓法中的 2 階導函數取代 1 階導函數運算的結果，表示如下

$$I''_{m,i}(\beta_i^{(j)}) = \frac{I'_{m,i}(\beta_i^{(j)}) - I'_{m,i}(\beta_i^{(j-1)})}{\beta_i^{(j)} - \beta_i^{(j-1)}} \qquad (4.15)$$

因此割線法的式子可以表示成

$$\beta_i^{(j+1)} = \beta_i^{(j)} - \frac{\beta_i^{(j)} - \beta_i^{(j-1)}}{I'_{m,i}(\beta_i^{(j)}) - I'_{m,i}(\beta_i^{(j-1)})} I'_{m,i}(\beta_i^{(j)}) \qquad (4.16)$$

有別於牛頓演算法需要 2 階導函數，割線法可以避免複雜的 2 階導函數的計算，或是當 2 階導函數不可得的時候可以使用。這個問題將在下一節中詳細的描述。

從上述的方程式(4.16)來看，割線法的使用需要兩個起始點(initial value): $\beta_i^{(0)}$ 以及 $\beta_i^{(-1)}$。進一步結合 E 模型的方程式，利用最小音質損傷因子(Impairment Factor)找出對應的 β_i^* 值，進而得到最佳化的播放延遲時間。以下分成兩種情形討論割線法結合 E 模型的應用。

4.4.1 單一描述編碼的應用

根據 E 模型所定義的損傷因子為

$$I_{m,i} = 0.024d_i + 0.11(d_i - 177.3)H(d_i - 177.3) \qquad (4.17)$$

$$+ \gamma_1 + \gamma_2 \ln(1 + \gamma_3 e_i)$$

其中 $e_i = e_n + (1 - e_n)e_{b,i}$ 為封包的網路漏失機率 e_n 加上播放延遲時間過低導致的晚到遺失機率 $e_{b,i}$。

我們假設網路延遲的特性符合 Pareto 分佈，$e_{b,i}$ 可以寫成

$$e_{b,i} = 1 - F(d_{play,i}) = \begin{cases} (\frac{k}{d_{play,i}})^{\alpha}, d_{play,i} > k \\ 0, d_{play,i} \le k \end{cases} \qquad (4.18)$$

其中 $d_{play,i} = \hat{n}_i + \beta_i \hat{v}_i$。

接著利用割線演算法可以求得最佳化後的 $\beta_i^* = \arg\min_{\beta_i} I_{m,i}$。

4.4.2 多重描述編碼的應用

在式(4.17)E 模型所定義的損傷因子中，整體封包漏失機率 e_i 為

$$e_i = e_n^{s_1} e_n^{s_2} + e_n^{s_1}(1 - e_n^{s_2})e_{b,i}^{s_2} + e_n^{s_2}(1 - e_n^{s_1})e_{b,i}^{s_1} + \qquad (4.19)$$

$$(1 - e_n^{s_1})(1 - e_n^{s_2})e_{b,i}^{s_1} e_{b,i}^{s_2}$$

其中

$e_n^{s_k}$ ：封包在路徑 k 的網路漏失機率。

$e_{b,i}^{s_k}$ ：封包在路徑 k 的晚到遺失機率。

進一步可求得到 E 模型損傷因子的 1 階導函數

$$I'_{m,i}(\beta_i) = \frac{dI_{m,i}}{d\beta_i} = c\hat{v}_i + \frac{\gamma_2 \gamma_3}{1 + \gamma_3 e_i} \frac{de_i}{d\beta_i} \qquad (4.20)$$

其中

$$c=\begin{cases}0.024, d_i<177.3\\ 0.134, d_i\geq 177.3\end{cases}$$

最後利用割線法求得最佳化後的 $\beta_i^* = \arg\min_{\beta_i} I_{m,i}$。

至於微分 $\frac{de_i}{d\beta_i}$ 的計算，可分成以下三種情形討論。

$$\frac{de_i}{d\beta_i}=\begin{cases}\frac{-\hat{v}_i}{d_{play,i}}\{(1-e_n^{s_1})(1-e_n^{s_2})e_{b,i}^{s_1}e_{b,i}^{s_2}(\alpha_1+\alpha_2)+\\ e_n^{s_1}(1-e_n^{s_2})e_{b,i}^{s_2}\alpha_2+e_n^{s_2}(1-e_n^{s_1})e_{b,i}^{s_1}\alpha_1\}, d_{play,i}>n_i^{s_1}\text{且}d_{play,i}>\overline{n}_i^{s_2}\\ \frac{-\hat{v}_i}{d_{play,i}}\{(1-e_n^{s_1})(1-e_n^{s_2})e_{b,i}^{s_1}(\alpha_1)+\\ e_n^{s_2}(1-e_n^{s_1})e_{b,i}^{s_1}\alpha_1\}, d_{play,i}>\overline{n}_i^{s_1}\text{且}d_{play,i}<\overline{n}_i^{s_2}\\ \frac{-\hat{v}_i}{d_{play,i}}\{(1-e_n^{s_1})(1-e_n^{s_2})e_{b,i}^{s_2}(\alpha_2)+\\ e_n^{s_1}(1-e_n^{s_2})e_{b,i}^{s_2}\alpha_2\}, d_{play,i}<\overline{n}_i^{s_1}\text{且}d_{play,i}>\overline{n}_i^{s_2}\end{cases}$$

第五章 實驗結果

本章將應用第 4 章音質評量平台,利用音質損害因子取得播放延遲與封包漏失的最佳化權衡,同時比較單一串流與多重串流傳輸之間的差異。為了比較播放排程演算法的優劣，需要事先蒐集網路延遲的記錄檔案(delay traces)。但由於傳送與接收兩個節點間的同步問題，單向網路延遲的量測有其困難度。因此本章實驗使用的單向網路延遲檔案將從以下兩種環境取得：

- 網路單向延遲模。
- 在自組式網路測試平台下，利用 ping 程式得到封包的網路雙向延遲(round-trip delay)，將其除以 2 之後視為單向網路延遲。

5.1 網路單向延遲模型

本節將介紹一種數學模型，用來模擬不同環境的單向網路延遲。

5.1.1 模型簡介

基於前人研究，一連串定期傳送的語音封包受到網際網路串流(internet traffic)的影響，可被一批伯努利隨機程序(batched Bernoulli process)近似[12][13]。因此，在本實驗中，用單一伺服

器排隊模型來模擬。此伺服器具一 FIFO 緩衝器，有兩個輸入端，一個代表語音串流(audio traffic)，而另一個則為網際網路串流。模型如圖 5.1，其中 D 表示語音封包整體延遲中的固定部分，例如傳輸延遲與編碼延遲。而伺服器的服務速度(service rate)以 μ 表示，單位為 bits/s。聲音訊號依固定大小的音框(frame)且週期性地傳送，以 P 表示其長度，單位為 bits；δ 表示兩個封包傳送間的時間差。而網路串流會以許多串流依序疊加而成，而和聲音訊號競爭分享一般的網路資源。

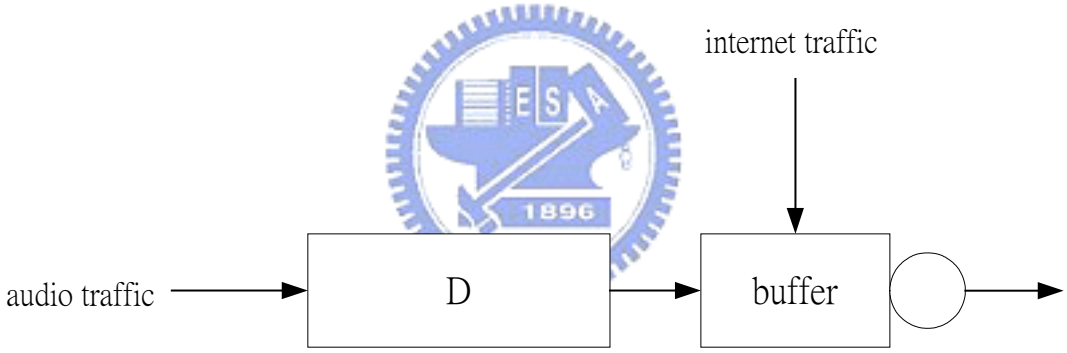
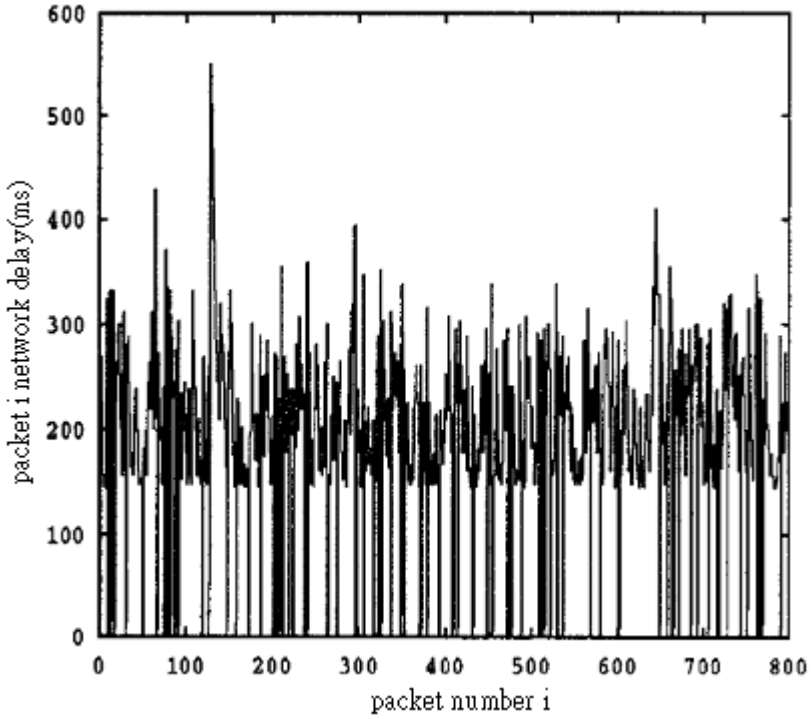
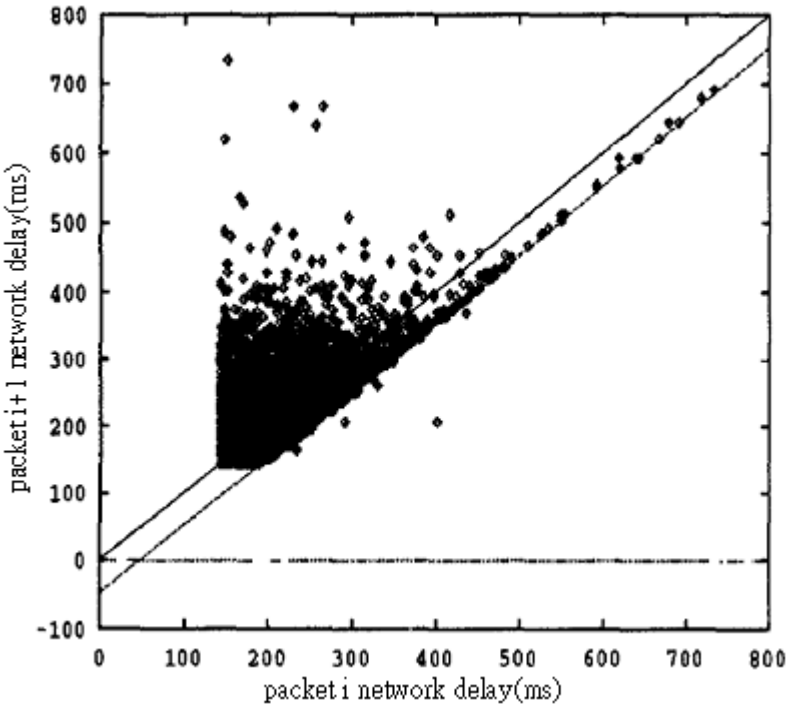


圖 5.1 網路延遲的模型

第 i 個語音封包的整體延遲以 d_i 表示，圖 5.2(a)表示其量測值序列，圖 5.2(b)則表示其相位圖(phase plot)。相位圖座標以 $(x = d_i, y = d_{i+1})$ 標示每個封包的整體延遲。圖 5.2 中，i 的範圍為 0 至 800，且兩個封包傳送間的時間差 δ ＝50ms。



(a)



(b)

圖 5.2 整體延遲的時間(a)連續圖及(b)相位圖

5.1.2 網路延遲分析

首先考慮網路串流很小的情形，如 Telnet 封包，其特色為緩衝器內的網路封包很少且很小。在此情況下，連續語音封包的等待時間(waiting time)接近一常數。第 i 個語音封包的等待時間(不包括服務時間)可表示為

$$w_{i+1} = w_i + \varepsilon_i \qquad (5.1)$$

其中 ε_i 為一平均值為 0 且變異數很小的隨機程序。所以封包的整體延遲 $d_{i+1} = D + w_{i+1} + P/\mu$ 及 $d_i = D + w_i + P/\mu$，其差距為

$$d_{i+1} - d_i = w_{i+1} - w_i = \varepsilon_i \qquad (5.2)$$

因此，相位圖上的點鄰近對角線 $d_{i+1} = d_i$(如圖 5.2(b)所示)，且靠近最小延遲點(D, D)，在圖 5.2(b)中，$D \approx 140ms$。

接下來討論另一種在封包時間差 δ 很小的情形下，且兩個連續語音封包中間收到一個很大的網路封包(一個或多個 FTP 封包)。假設 B 為此網路封包的大小(單位為 bits)，在第 $i+1$ 個語音封包前收到。因此第 $i+1$ 個語音封包的排隊延遲為：

$$w_{i+1} = w_i + B/\mu \qquad (5.3)$$

因此，

$$d_{i+1} - d_i = B/\mu \qquad (5.4)$$

如果 $w_{i+1} > \delta$，因為要等待伺服器處理此網路封包，會有一個以上的

語音封包累積在第$i+1$個語音封包之後。假設累積有 k 個語音封包，且在第$i+1$個及第$i+k$個語音封包間沒有其他網路封包到達此伺服器。則第$i+1$個至第$i+k$個語音封包會以固定間隔離開此排隊伺服器，此間隔為P/μ。因此，可得到：

$$d_{i+2} - d_{i+1} = P/\mu - \delta \qquad (5.5)$$

同理：

$$d_{i+3} - d_{i+2} = ... = d_{i+k} - d_{i+k-1} = P/\mu - \delta \qquad (5.6)$$

在相位圖上，滿足上式的點會位於一直線上，在圖 5.2(b)中以虛線表示。若$\delta < P/\mu$，語音封包會導致排隊伺服器飽和，因此必須保持在$\delta > P/\mu$。

接下來根據 Lindley 遞迴方程式的兩個連續應用，令第 i 個封包的等待時間為w_i，第 i 個封包的服務時間為y_i，而第 i 個封包與第 $i+1$ 個封包的抵達時間差為x_i，因此：

$$w_{i+1} = max(w_i + y_i - x_i, 0) \qquad (5.7)$$

我們假設第一個語音封包到達此排隊緩衝器的時間為δ，因此第 i 個語音封包到達的時間為 $i\delta$。我們假設網路串流b_i，單位為 bit，在時間 $i\delta$及$(i+1)\delta$中間到達，b_i為一個表示網路串流的隨機變數，再假設所有的b_i皆在相同的時間t_i到達排隊緩衝器。讓wb_i表示此網路封包的等待時間，根據(5.7)可得：

$$wb_i = max(w_i + P/\mu - t_i, 0) \qquad (5.8)$$

將w_{i+1}及wb_i再套用在Lindley遞迴方程式上，可得：

$$w_{i+1} = max(wb_i + b_i/\mu - (\delta - t_i), 0) \qquad (5.9)$$

將(5.8)代入(5.9)，可得：

$$w_{i+1} = max(max(w_i + P/\mu - t_i, 0) + b_i/\mu - (\delta - t_i), 0) \qquad (5.10)$$

只要在區間$[i\delta, t_i + i\delta]$內緩衝器不是空的，$w_i + P/\mu - t_i$項是正值。因此上式可簡化為

$$w_{i+1} = max(w_i + P/\mu - t_i + b_i/\mu - (\delta - t_i), 0) \qquad (5.11)$$

$$= max(w_i + (P + b_i)/\mu - \delta, 0)$$

如果區間$[i\delta, t_i + i\delta]$內緩衝器不是空的，$w_i + (P + b_i)/\mu - \delta$項是正值。因此，(5.11)可簡化為

$$w_{i+1} = w_i + (P + b_i)/\mu - \delta \qquad (5.12)$$

因此

$$b_i = \mu(w_{i+1} - w_i + \delta) - P \qquad (5.13)$$

b_i的機率分佈可依$w_{i+1} - w_i$的分佈來估計。然而(5.13)成立的條件，區間$[i\delta, t_i + i\delta]$內緩衝器不是空的。基於上述的分析，可依使用者設定的網路狀態隨機產生M個網路封包b_i。

5.2 移動式自組網路下的封包傳輸延遲

如圖 5.3，本實驗所建置之移動式自組網路平台，包括 F1~F5 以及 Rx 共 6 個固定點，利用多組臨時建置的固定節點與安裝於量測車輛之移動節點。在交通大學室外校園環境進行，4 個移動發送點分別移動於 R1,R2 等區域並同時發送資料至 Rx，其佈設地點如圖所示。分別在不同時間進行 1~4 個使用者數目之通訊服務，發送端為移動狀態，分別標示為 MH1,MH2,MH3,MH4，接收端則為 Rx。傳輸平均跳接數與發送端當時所在位置有關。移動車輛行進路線為圖中虛線箭頭所示之路段，以平均約 20km/hr 速率在這些路徑上反覆移動。

以下為實驗的結果與現象：

1. 傳輸路徑之跳接數可能為 1-hop 到 3-hop。
2. 實驗數據為多個點一起傳送資料。
3. 傳輸延遲與同時間進行通訊節點的數目關係不大。
4. 雙向延遲小且無封包漏失。

觀察圖 5.4，為突顯無線網路環境的不穩定傳輸，有別於模擬所得的網路延遲，移動式自組網路的延遲量測值具有較大的變異。舉例而言，模擬網路延遲的相位圖 5.4(b)中，鄰近對角線的一直線符合本論文第三章 spike 特性。而觀察移動式自組網路的網路延遲相位圖 5.4(f)及(h)，在有限的封包內並沒有發生類似 spike 的現象。

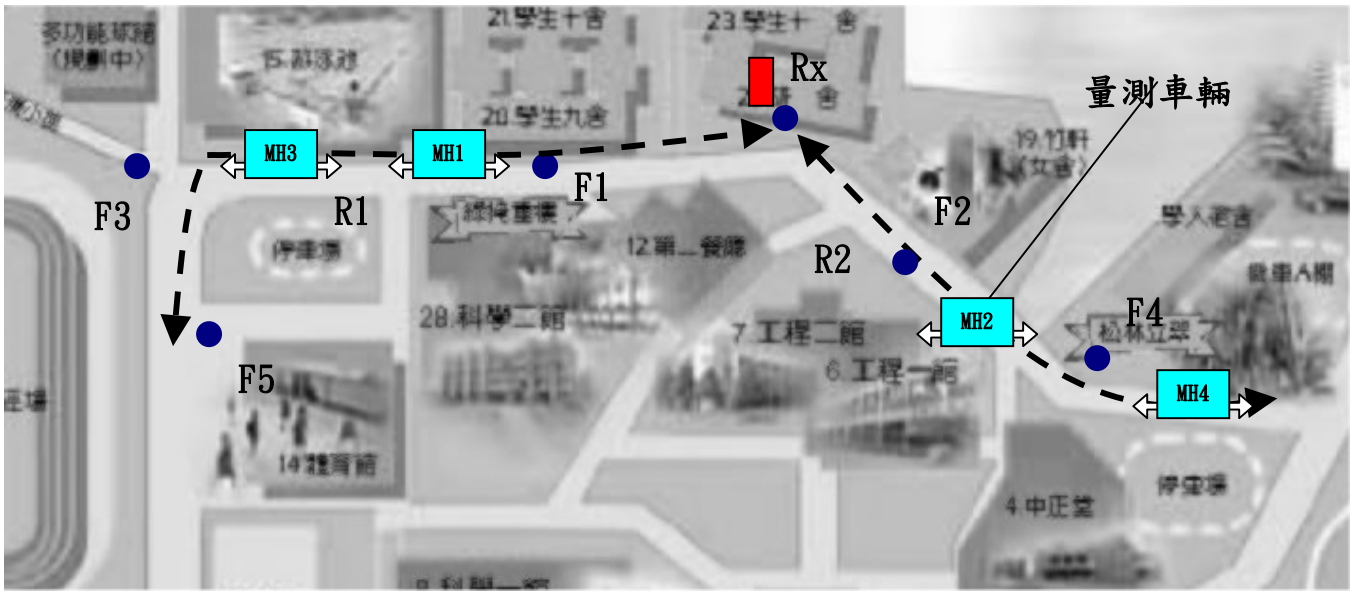
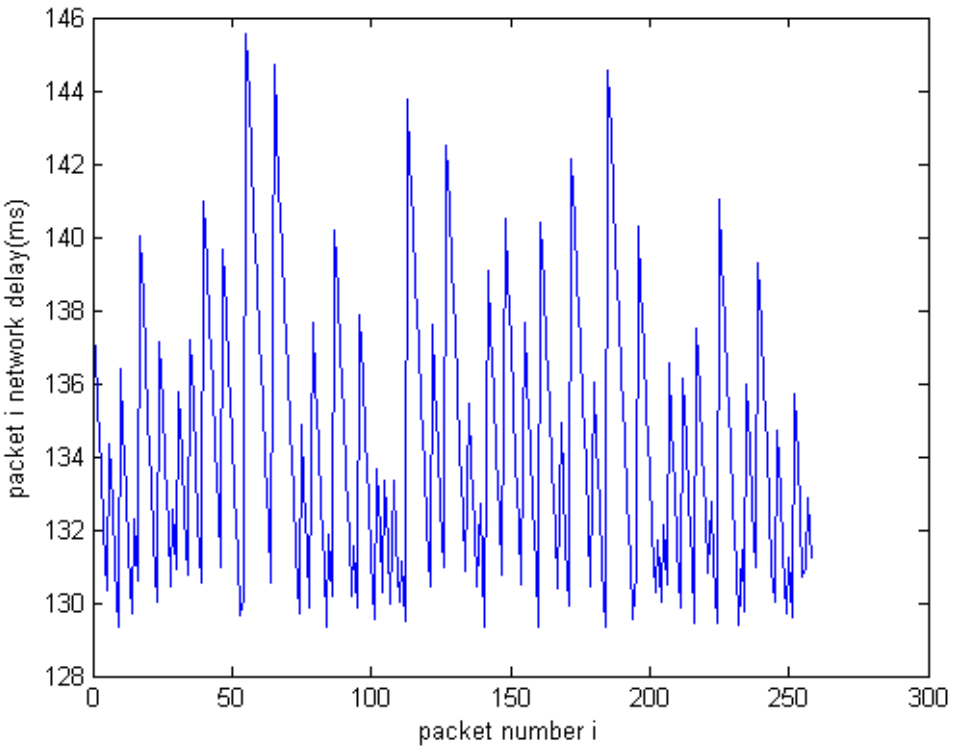
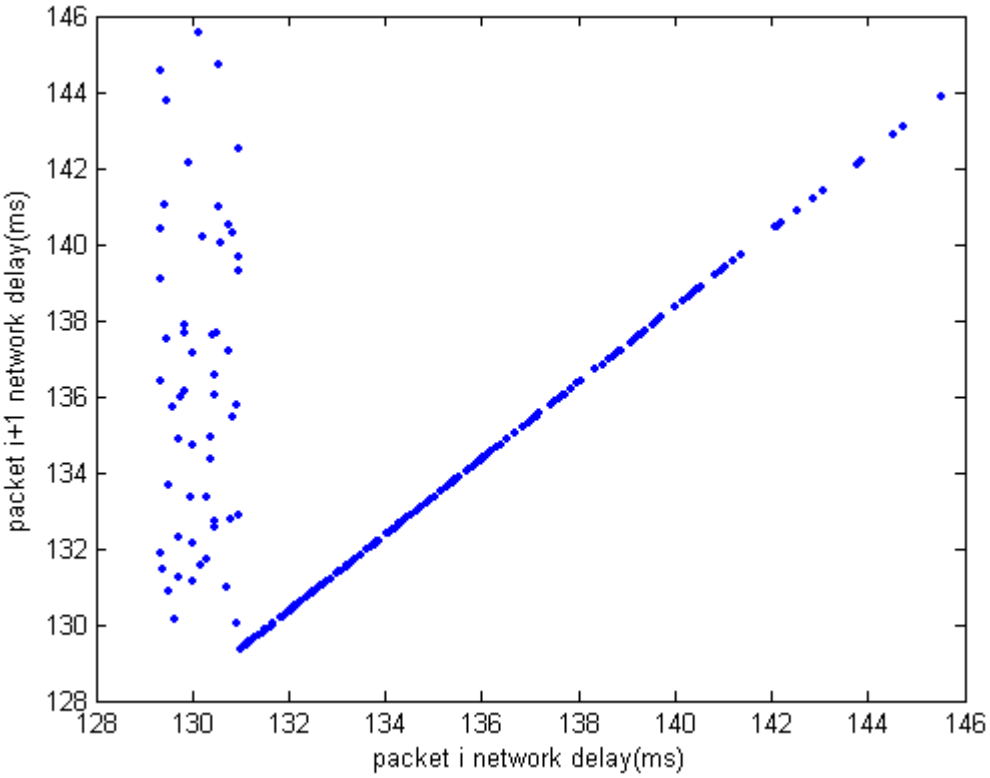


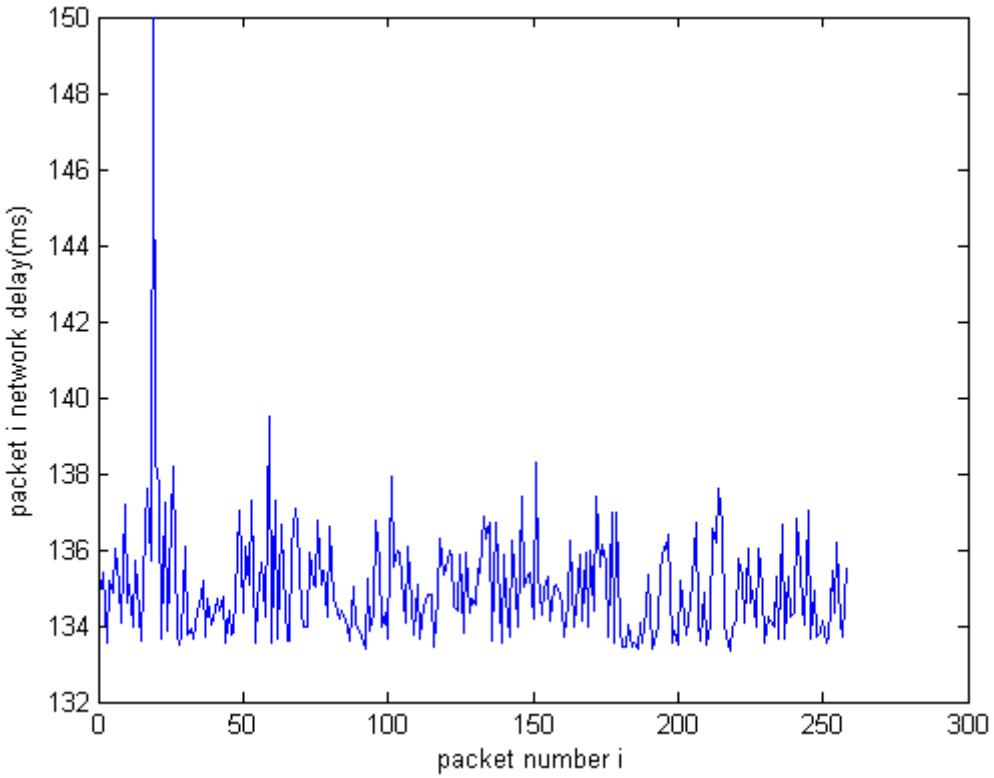
圖 5.3 固定點對移動點之多使用者實驗場景



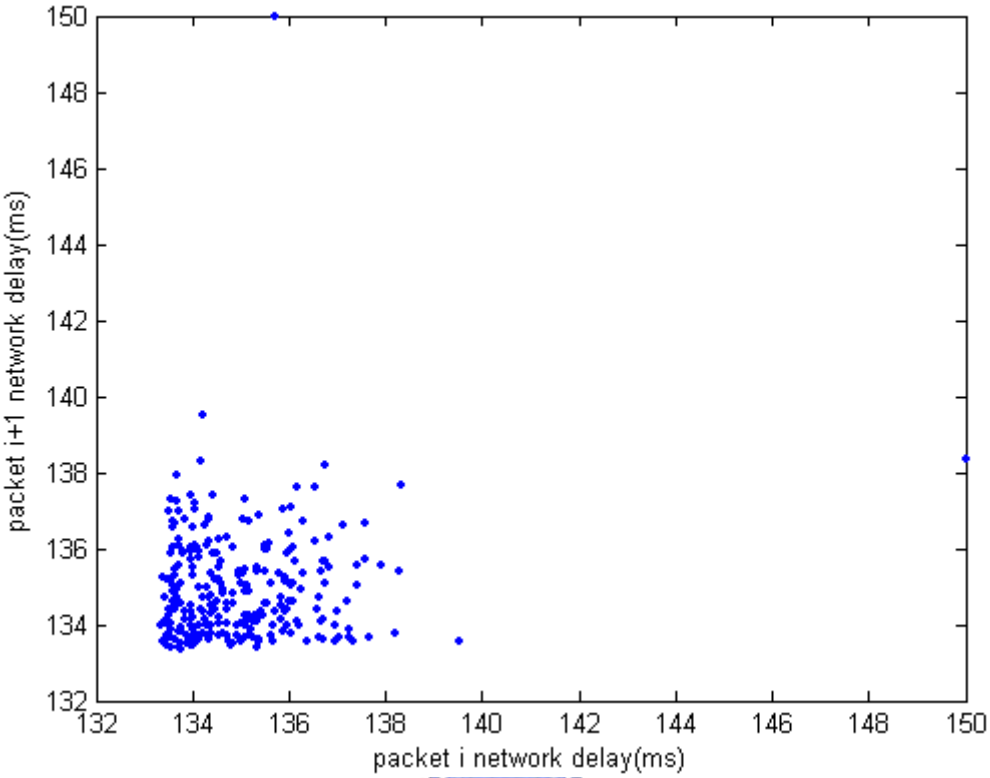
(a) 路徑 1 之模擬網路延遲



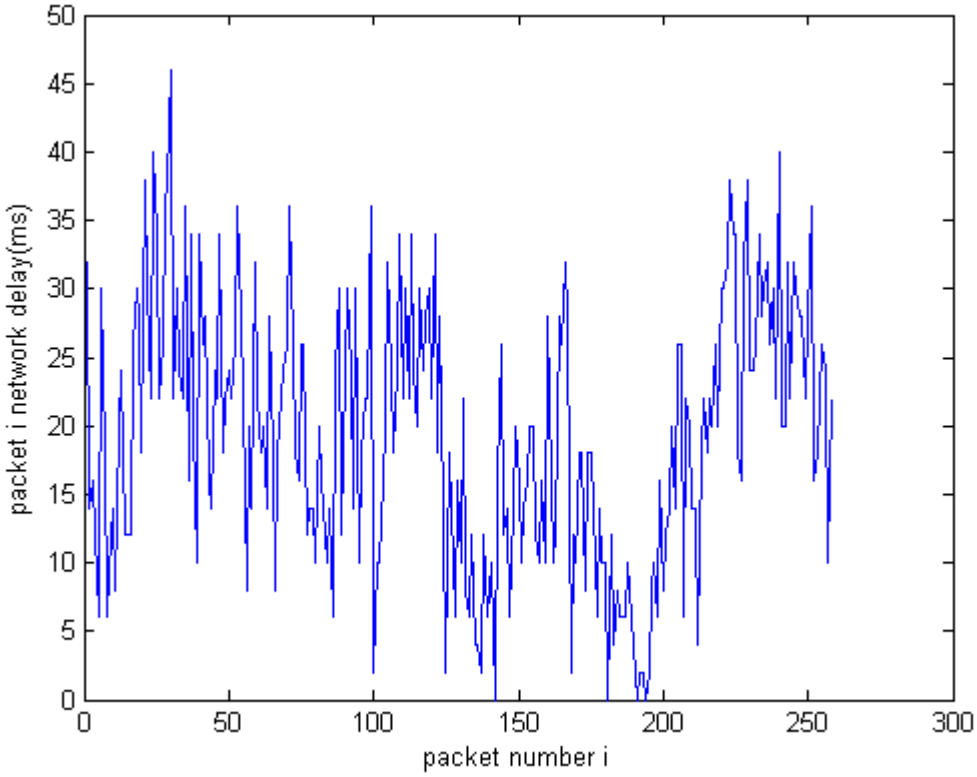
（b）路徑1之模擬網路延遲相位圖



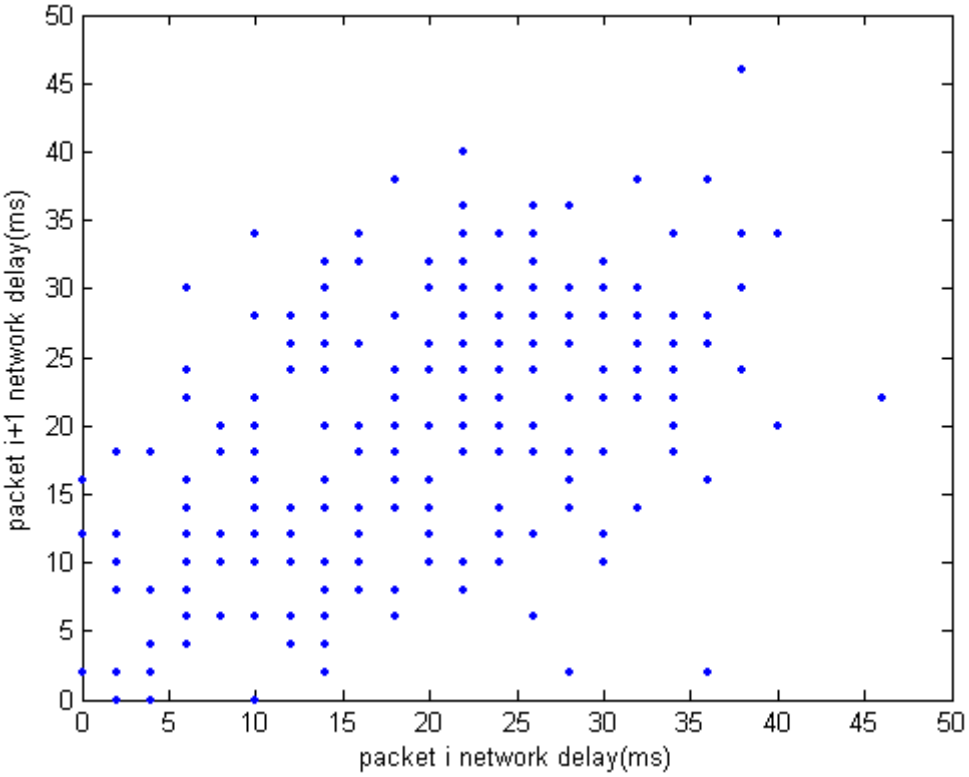
（c）路徑2之模擬網路延遲



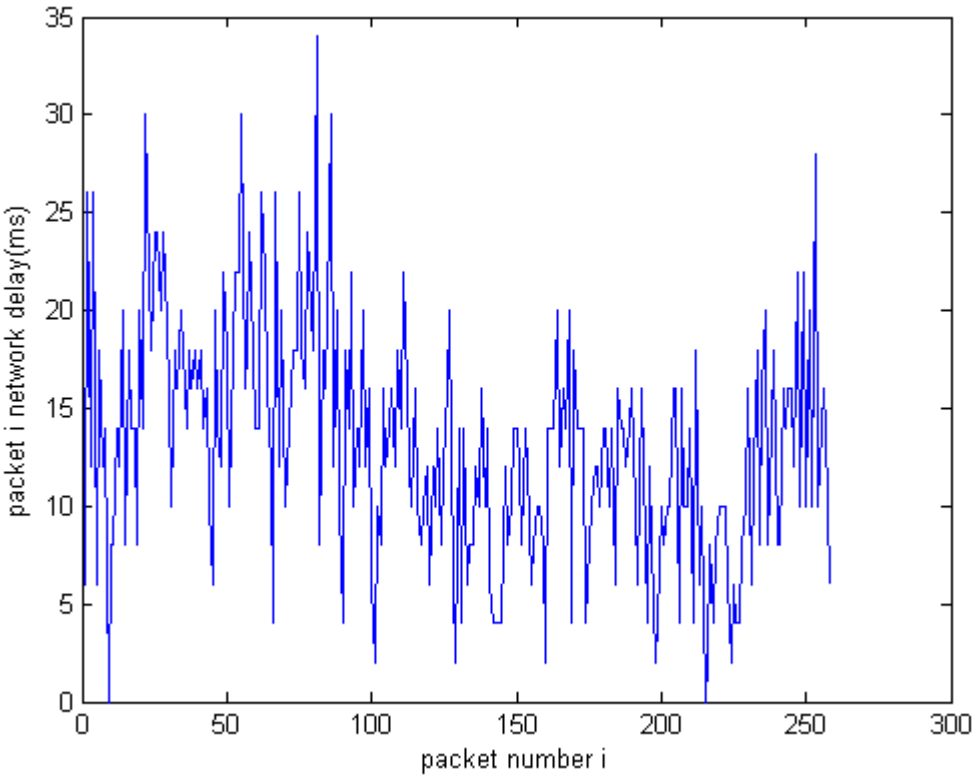
（d）路徑2之模擬網路延遲相位圖



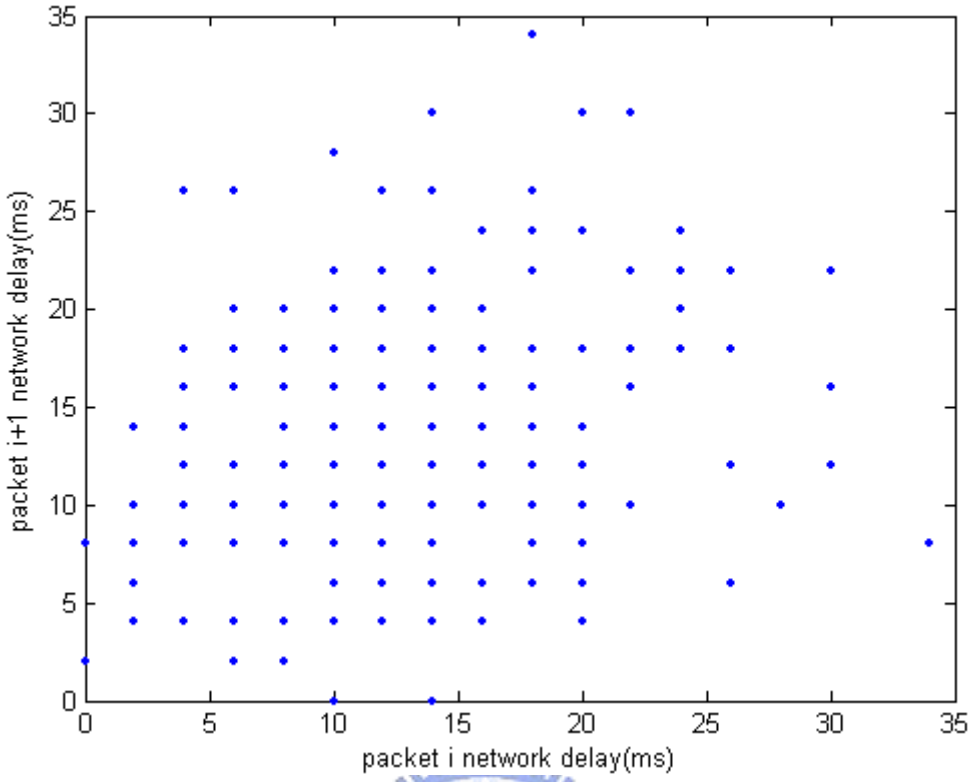
（e）路徑1之移動式自組網路延遲



（f）路徑1之移動式自組網路延遲相位圖



（g）路徑2之移動式自組網路延遲



(h) 路徑2之移動式自組網路延遲相位圖

圖 5.4 網路延遲的連續圖及相位圖

5.3 播放排程演算法的效能比較

如圖 5.5，將上述由模擬以及 MANET 量測所得的網路延遲檔案代入本論文第三章與第四章描述的多重串流傳輸、內插法與播放緩衝音質最佳化權衡的演算法中，結合音長比例調整(time-scaling)參數、封包漏失、話務靜音區間資訊代入音長調整機制作封包長度的調整。由(4.5)得知，PESQ 量測平台無法反映出延遲對聲音品質的影響，因此需將演算法求得的播放延遲的影響代入(4.7)求出延遲損害因子 I_d，再藉由(4.3)得到 E 模型定義的 R 值，最後根據 R 與 MOS 值的轉換式(5.16)(5.17)求得對應的 MOS 值[15]。

$$R = 3.026MOS^3 - 25.314MOS^2 + 87.060MOS - 57.336 \quad (5.16)$$

$$MOS = 1 + 0.035R + R(R-60)(100-R) \times 7 \times 10^{-6} \quad (5.17)$$

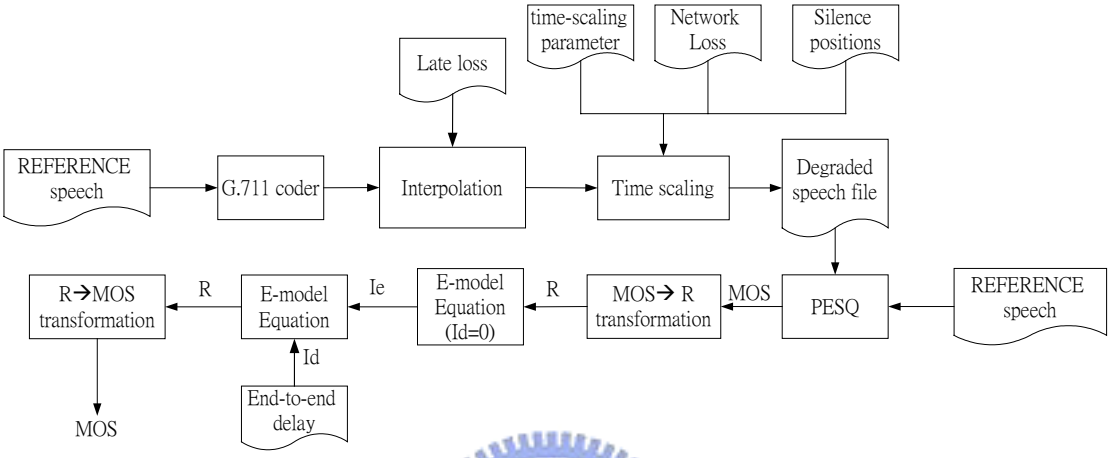


圖 5.5 音質評比流程圖

實驗的參考語句如圖 5.6，語音長度約為 9.5 秒，依 8000 赫茲取樣且每個取樣的量化位元數目為 8。兩組網路延遲檔案分別來自模擬與移動式自組網路環境，在模擬環境中路徑 1 及路徑 2 的封包傳輸露失率分別設定為 2%及 0.5%，在 MANET 環境量測所得的封包傳輸漏失率則皆為 0%，由表 5.1 與 5.2 中得知，我們提出的多重串流傳輸架構，若同時考慮漏失與延遲的音質最佳化權衡，不只能降低整體延遲，其封包漏失率也有顯著的下降。在模擬環境中 MOS 能改善約 0.07~0.18，而在 MANET 環境中 MOS 則改善約 0.08~0.14。

有　一　個　老　爸　教　兒　子　識　字

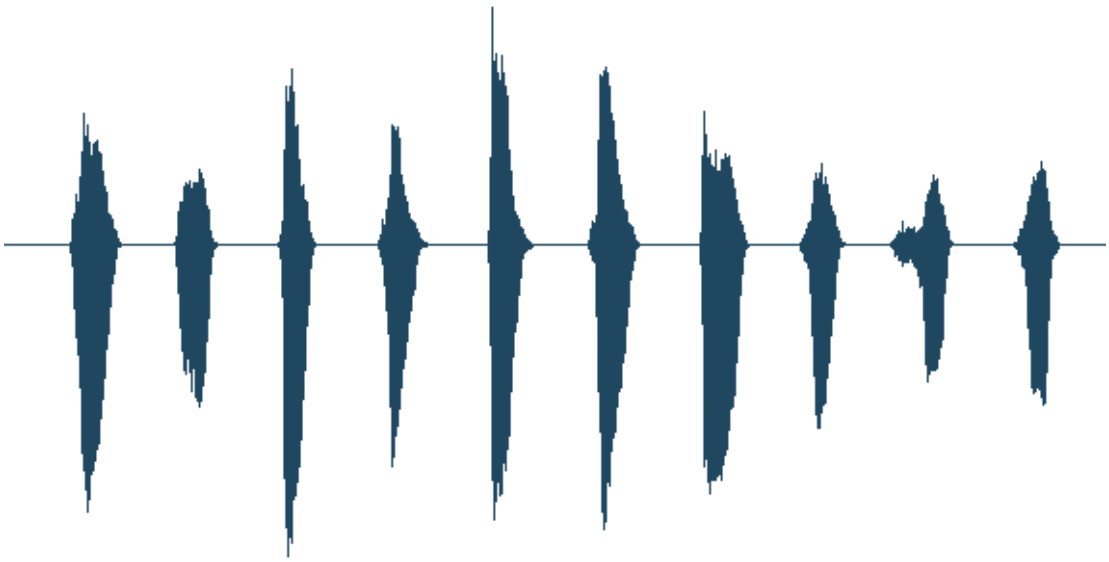


圖 5.6 受測語句及波形

Simulation	mean(ms)	variance	delay(ms)	loss(%)	MOSc
path1	134.618	16.644	183.653	2	3.3338
path2	135.039	2.4531	171.5892	1.4387	3.2265
MDC	--	--	166.2532	0.01	3.4086

表 5.1 在模擬網路狀態下語音傳輸的音質評分

MANET	mean(ms)	variance	delay(ms)	loss(%)	MOSc
path1	19.1846	93.418	153.3071	0.9434	2.8843
path2	13.3308	36.446	166.9883	1.4151	2.944
MDC	--	--	153.7735	0	3.0215

表 5.2 在移動式自組網路狀態下語音傳輸的音質評分

第六章 結論與未來展望

在整合無線網路與網際網路的服務應用中，本論文探討語音人機介面在 MANET 環境中面臨的問題：網路延遲、延遲顫動及封包漏失。在系統製作部份，我們在網際網路建構一分散式語音辨認系統，再根據辨認結果回傳特定的有聲資訊給用戶。至於適應性播放排程機制，主要是針對每個封包依據 NLMS 演算法估測其網路延遲，再配合音質預測模型，彈性調整而得聽覺評量上最理想的播放延遲時間設定。此外，結合多重串流傳輸的方式，將減輕路由機制無法找到最佳的傳輸路徑以及叢發性漏失的影響。實驗結果顯示，基於多重敘述編碼的封包播放排程機制能同時降低緩衝延遲及漏失率，因而得到最佳的聲音品質。

雖然多重敘述編碼可以改善語音品質，但如何透過兩個不同的路徑傳送封包是一個值得深入研究的課題。本論文所使用的兩種網路延遲產生模式，分別為以 MANET 實際傳送封包而得到的延遲數據，另一個是以數學模型而取得的延遲數據。前者數據的取得較實際但需要花費大量的時間，而後者則可任意改變模型參數的設定而得到多樣的數據。但兩者都屬於 off-line 的實驗結果，因此未來方向應設法克服傳送與接收端同步的問題。

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] P. DeLeon and C. Sreenan, “An adaptive predictor for media playout buffering,” in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, vol. 6, (Phoenix, AZ), pp. 3097-3100, Mar. 1999.

[4] A. Shallwani and P. Kabal, “An adaptive playout algorithm with delay spike detection for real-time VoIP,” in *Proc. IEEE Canadian Conf. Elec. Comp. Eng.*, (Montreal, Canada), May 2003.

[5] J.-C. Bolot, “End-to-end packet delay and loss behavior in the Internet,” *Computer Comm. Review*, vol. 23, no. 4, pp. 289–298, Sept. 1993.

[6] S. Savage, A. Collins, E. Hoffman, J. Snell, and T. Anderson, “The end-to-end effects of Internet path selection,” *Computer Comm. Review*, vol. 29, no. 4, pp.289–99, Oct. 1999.

[7] Y. J. Liang, N. Färber, and B. Girod, ”Multi-stream voice over IP using packet path diversity,” in

[8] “Method for subjective determination of transmission quality,” ITU-T Recommendation P.800, Aug. 1996

[9] “Subjective performance assessment of telephone-band and wideband digital codecs,”ITU-T Recommendation P.830, Feb. 1996.

[10] Cole, R. G. and Rosenbluth, J. H.“Voice over IP performance monitoring,”.ACM Computer Communication Magazine34, 12 (Dec.1996), pp. 9-24.

[11] 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.

[12] J.-C. Bolot, "Characterizing end-to-end packet delay and loss in the Internet," J. High-Speed Networks, vol. 2, no. 3, pp. 289-298, Dec. 1993.

[13] J.-C. Bolot and A. Vega-Garcia,” The case for FEC-based error control for packet in the internet,” ACM Multimedia Systems, 1997.

[14] International Telecommunication Union, “Perceptual evaluation of speech quality (PESQ), an objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs,” ITU-T Recommendation P.862, Feb 2001.

[15] ITU, “The E-Model, a computational model for use in transmission planning,” ITU, Geneva, Switzerland, ITU-T Rec. G.107, 2003.