第三章 VoIP 系統研究

在 VOIP 架構下，可以選用 G.729、G.723.1、G.711 等不同壓縮方式作為語音壓縮的方法，藉由這些壓縮方式在網路傳送較小的封包大小，減少封包遭受網路延遲的機會。表 3.1 是針對 G.729、G.723.1、G.711 等語音編碼的比較，表中分別介紹三種語音壓縮方式的壓縮後封包大小及每一個封包所代表的語音時間。

CODER	SPEECH TIME	SPEECH BYTES
G.711	10ms	80 bytes
G.723.1	30ms	24bytes
G.729	10ms	10bytes

表 3.1 不同語音壓縮比較

本章主要介紹在網路上可以得到的一些網路參數，進而利用這些參數和接收端前緩衝區（Buffer）的建置來改善語音的品質，並透過本論文提出的語音段落（Talk Spurt）和語音時間校等

方法來縮短因緩衝區存在而增長的語音長度，除此之外，還利用語音評估的方式如 SSSNR 來評估語音品質。

3.1 網路參數和封包處理方式

3.1.1 網路參考參數

資料在網路上傳送是無法避免網路的延遲，所以要利用網路的參數[15]來做緩衝區調整的依據，而緩衝區就是依照這些參數來做緩衝區（Buffer）的調整，藉由這些參數的調整來解決因為時間延遲而晚到的封包，這些參數包括網路封包的延遲（Delay）以及封包和封包之間的時間延遲差（Jitter），和封包開始的時間、封包到達接收端的時間、封包播放時間及在一個語音段落內（Talk Spurt）的平均時間延遲（Average delay）及變異量（Variation）等等，並藉由這些參數的調整來讓播放的語音能夠更平滑，不會因網路的時間延遲而讓語音品質下降。下面將對這些參數做一個定義。

Sender time 表示封包開始從傳送端傳送出來時，所記錄的時間，主要是要知道讓接受端做參考用的，讓接受端知道此封包在網路所延遲的時間。

Arrival time 表示封包在到達接受端時，所記錄的時間，主要是

要和傳送端做比較的，讓接受端知道此封包在網路所延遲的時間。

Network end-to-end delay 表示在網路上傳送資料總共究竟花費掉多少時間，藉此可以評估現在整個網路狀況是否處於壅塞的狀況。

Jitter 是考慮封包和封包之間相差的時間，從這個參數可以更清楚的知道封包和封包之間的特性，這個參數是為了補助 Network end-to-end delay 的缺點，因為在整段的時間延遲中，其時間延遲並不大，但是卻讓語音的品質下降，這是由於部份語音封包之間的時間差太大，才造成這種情況。

Loss rate 代表在一段時間後，計算在這段語音中其封包的遺失機率，藉以評估網路狀況。

Mean Delay 是針對 Network end-to-end delay 做一個平均值，希望能的到一個平均的時間延遲值，避免只考慮少數狀況。

Mean Jitter 是針對 Jitter 做一個平均值，希望能的到一個平均的時間延遲值，避免只考慮少數狀況。

3.1.2 VoIP 調整參數

VoIP 的應用多用在 IP 之網路上， IP 網路與 ATM 網路不同，無法完全提供應用程式網路服務品質保證（QoS）[14]，而現今所發展的 IP-QoS，如 RSVP 或 DiffServ [8][10][11]等等，也未能如 ATM

般提供完整的網路服務品質保證技術。如在網路無法提供適當的保證後，就必須了解網路的壅塞情況，適度的調節在每次的語音重新同步後下一語音段落開始時所需緩衝的個數，以期能提高語音品質。調節的方法考慮了網路延遲（Network end-to-end delay）[12]、封包遺失率（Loss rate）與網路變異（Network variation）等三項網路品質參數，一般而言，網路延遲大小會影響在連續封包的開始，如一開始緩衝不足，便會使緩衝區內常常呈現空的狀態，解碼器此時若無封包可解，會視為封包遺失；如緩衝數量足夠時，則緩衝區便可避免因網路延遲而呈現空的狀態，但又要避免延遲過長的情況。故從經驗得知，網路延遲大小所影響可決定一開始的緩衝個數，只要緩衝區有了足夠的個數則往後的播放皆能順利進行，故此網路參數會決定每次同步後語音段落所需的一開始緩衝個數，但此一參數一般採用平均值（Voice Mean Delay ; VMD）計算，對於網路變化無法立即反映。依照語音的特性，認為應考慮下列幾點。

(i) 封包遺失率

此一方法最為容易，利用接收到的封包依遺失率與連續遺失幾個封包時來調整緩衝大小，此一方法為在最近 K 個封包的遺失率大於 P 時，考慮將緩衝封包個數固定增加。此一方法可與語音段落相配合而

形成第二種方法，即使用上一語音段落發生的遺失率大於 P 時，在下一段落開始時便考慮加緩衝個數加大。其前述的方法較具彈性，可依照 k 的參數調整至適合的分析數據，但其 K 值分析複雜，不易得到最佳值，故採用了第二種方法，配合了語音段落的特性，來完成語音緩衝的調整。此一方法的優點為簡易計算，容易整合於產品之中，但其缺點為如當網路變化過快時，因調整緩衝大小的量為固定，無法有效反映於網路狀態的快速改變。此一方法其特性為利用過去的資料作為未來調整的決定，但是單單只是用一個語音段落的封包遺失率來調整下一個語音段落的緩衝個數時可能會不妥，因為語音段落有長有短，如果只是以一個語音段落就做調整話，在很短的語音段落情形下，只要遺失少數的封包就會造成巨大的封包遺失率，然後在根據此封包遺失率來調整緩衝區數目卻很不實際，所以可能在數個語音段落或一定封包數之後才計算封包遺失率可能更能正確計算出調整緩衝區的個數。

(ii) **語音段落內最大封包時間差延** (Jitter)

如果利用語音段落遺失率來調整緩衝區大小可能會有些爭議，所以利用封包和封包之間的時間延遲差 (Jitter) 來作調整更可以反應出緩衝區的大小，為了不讓晚到的封包因為該播放而來不及播放而遺失的情形發生，於是就採用最大的時間延遲差，來做為調整的依據，

而這裡所指的封包之間的時間延遲差是在語音段落內所有封包延遲時間跟第一個封包延遲時間做相減，並取出最大的時間延遲值（Jitter value）當作下一個語音段落調整的依據，如此做可以確保在下一個語音段落內可以讓更多晚到的語音封包不會遺失，相對的更加深時間的延遲（Delay），再加上如果只有少數的語音封包時間延遲（Delay）發生，那就會因這些少數晚到的封包而讓下一個語音段落增加不必要的緩衝封包，也更加深時間延遲（Delay）。

(iii) 語音段落內封包和封包之間時間延遲變異性（Jitter variation）

為了避免上述這種情況發生，所以也要探討封包和封包之間時間延遲的變異性（Jitter variation），如果時間延遲的變異性太小的話則表示只有少數封包間的時間延遲是落在高點的，這就表示可以不用參考這些特別的晚到封包，而是取適當的時間延遲值（Jitter value）來當作下一個語音段落緩衝的封包數，相反的，時間延遲的變異性（Jitter variation）達到一定值時，在這個語音段落內最大的時間延遲值（Max Jitter value），是可以做為下一個語音段落的緩衝個數。

3.1.3 封包延遲（Delay）或遺失（Loss）

在探討網路延遲對封包有何影響之前，先考慮當接受端遇到經過網路這個方塊而沒有被網路丟棄（Drop）的語音封包時，接受端對

這個晚到的語音封包可以採用的應對方式將會有兩種情形，這兩種分別為語音封包延遲（Delay）、語音封包丟棄（Drop），參考圖 3.1 為 Delay 和 Drop 的說明圖。

封包延遲（Delay）：針對晚到的語音封包，不管如何接受端都要讓這個封包順利播放，所以就等待此封包到達讓其能夠順利播放，這種方式稱為延遲播放

封包丟棄（Drop）：在語音封包該播放時此封包卻晚到而來不及播放時，就把這個封包丟棄（Drop），這個方式稱為丟棄播放

這兩種方式各有優缺點，在封包延遲（Delay）中，把晚到的語音封包延遲一段時間讓其可以順利播放出來，這種方式可以保持語音的完整性，能夠完整聽到整段語音，但是在這種情形下會讓語音斷斷續續播放出來，更會因為等待晚到的語音封包，而增加了語音原本的播放長度，原本只有 1 分鐘的語音長度，可能因為等待晚到的封包而需要花費 1 分多鐘來聽完整段語音，在 VoIP 下雙方說話並不是只有一句話而已，所以對接受端而言時間延遲會一直累積下去，直到雙方溝通結束，這對即時語音的傷害是很大的。

而在封包丟棄（Drop）方面，當遇到本來該播放的語音封包，卻

因為沒即時到達的情形時，就把晚到的語音封包丟棄（Drop），如此做的話，會損失語音的完整性，卻可以在語音本來所需播放的時間內把語音播放完畢，但卻因晚到的封包被丟棄，而造成接收端無法聽到晚到的語音封包，這種情形也會降低語音的品質。

正所謂魚與熊掌不可兼得，也是這兩種情形的寫照，各有優缺點，所以在 VoIP 環境下採用哪一種方式來應付晚到的語音封包比較好，也是值得分析、探討的。

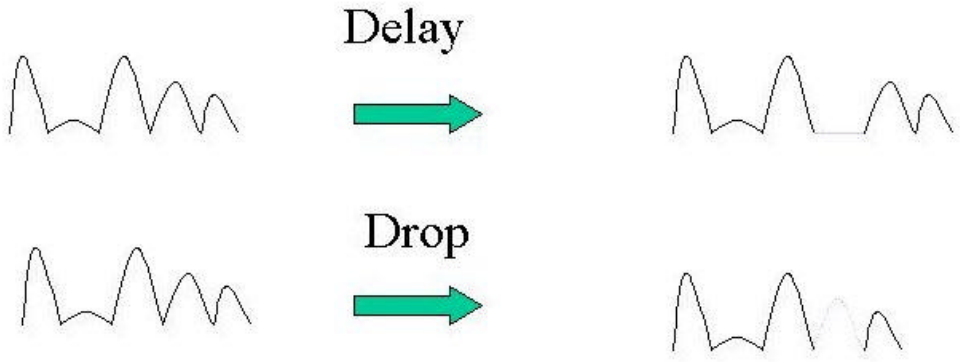

圖 3.1 Delay 和 Drop

3.2 緩衝區

3.2.1 緩衝區配置

在網路上傳送即時性之語音封包，會因封包切割與重組，及傳送介質速度所影響，而造成封包之延遲，此種效應對資料封包所造成之影響較小，但對具有即時性之語音封包就會造成很大影響。當語音資料開始解碼並播放時，因網路延遲而未到之語音封包，會讓解碼端無資料可解，即使在語音播放後到達，但已經失去時效性，也會被視為遺失之封包，而使得語音播放品質大量下滑，嚴重影響聽覺效果。針對解決方法，可在解碼器端前加一小量之緩衝區[17]，將收到之封包先緩衝於緩衝區內，等緩衝到一定程度時再開始播放語音資料，此種緩衝區方法常被人使用在即時性資料之傳送。但是語音資料有長短不同的時間延遲，較大之緩衝區雖然可以有效的緩衝網路延遲效應，使得語音封包播放較為穩定，但卻會對語音封包造成更大之延遲。例如較大的緩衝區會對使得播放品質提高，但相對地，會增加每一個語音封包的延遲。

如何在封包延遲時間與緩衝區大小之取得，也是需要去分析，圖 3.2 是封包在網路延遲的示意圖，而圖 3.3 是緩衝區配置來解決封包延遲的辦法。從圖 3.2 中可以看出第三個語音封包因為網路的延遲，而造成語音封包太晚到達接受端，讓接受端不能連續播放語音，形成

在第二封包和第三封包之間會有一段空白，而解決的辦法就像圖 3.3 一樣，在語音一開始時就先建立一個緩衝區（Buffer）來讓後面晚到的語音封包能夠正常播放。從圖中可以知道由於一開始有緩衝區的存在，而讓本來會晚到的第三個語音封包也能夠在接受端連續的播放出來，這就是利用緩衝區來解決晚到封包的辦法。

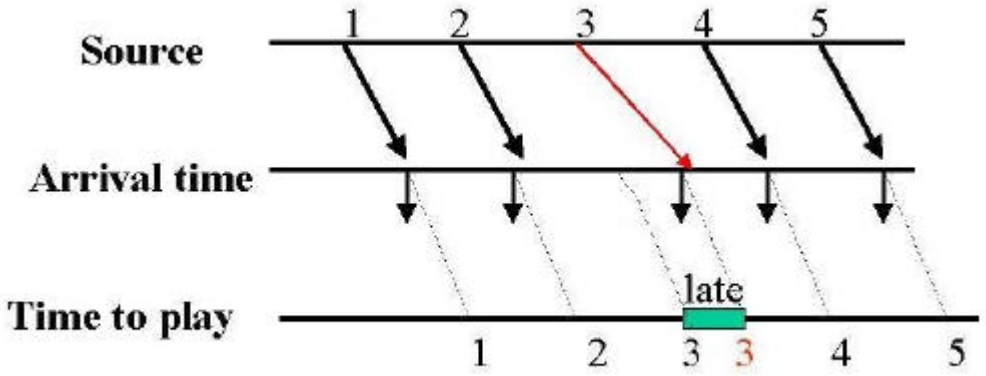

圖 3.2 封包網路延遲示意圖

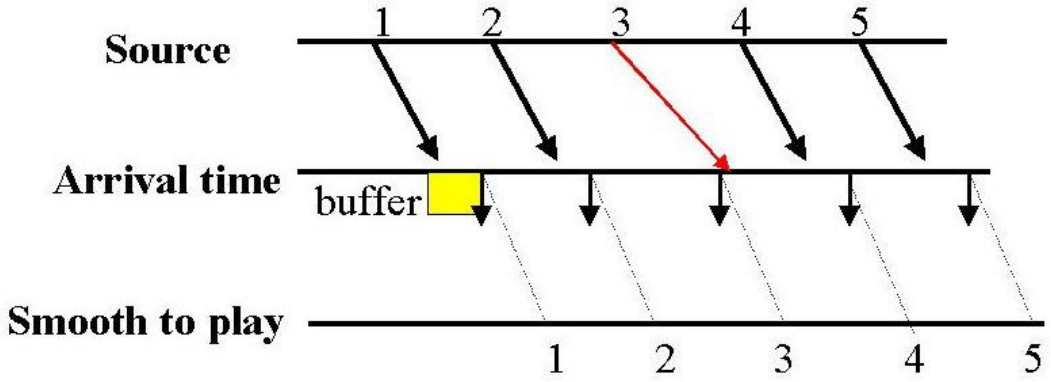

圖 3.3 緩衝區配置示意圖

在圖 3.3 中的 Buffer 是簡單的緩衝區配置，純粹只是為了讓後面晚到的封包能夠正常播放出，不會因為封包經過網路延遲之後封包來到的時間太晚，而讓語音不能夠連續正常播放，但是建立緩衝區，相對地也增加語音的等待時間。

3.2.2 緩衝區內處理過程

圖 3.4 說明經過網路時間延遲的封包在到達解碼端（Decoder）之前，先經過一個緩衝區（Smooth buffer），讓晚到的語音封包可以連續的到達解碼端，讓語音可以很平順的撥出。

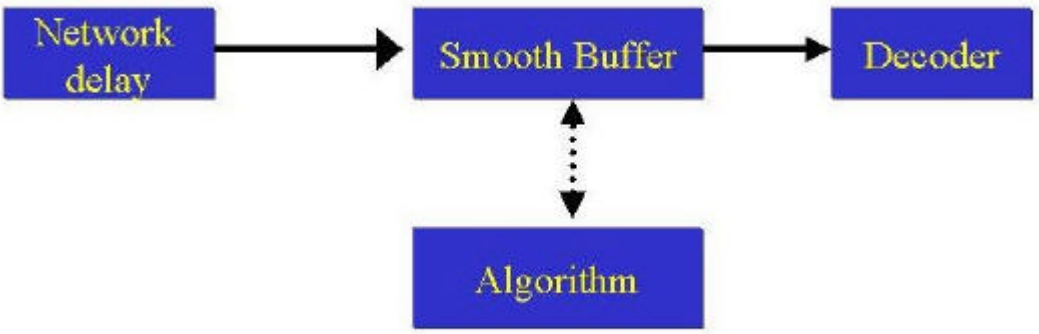

圖 3.4 Smooth buffer 表示圖

圖 3.4 中的演算法（Algorithm）是利用不同的調整方式，來調整

緩衝區在每一個時期所需要緩衝的個數，但是在探討緩衝區個數之前，首先定義何謂語音的段落，並藉由語音段落來作為調整緩衝區封包數的一個基準。

至於圖 3.4 中的 Smooth buffer 是採用兩種宣告，第一種宣告是在建立的緩衝區中記錄 0 或 1，0 表示這個緩衝空間是空的，沒有放任何資料；1 表示這個緩衝空間是有存放資料，另一種宣告是記錄進來資料封包的時間標的（Time stamp），記錄這個號碼是要確定封包播放的順序，而時間標的是利用 RTP 所得到的。Smooth buffer 運作並不是以先進先出（First In First Out）的方式來進行的，而是採用下列的方式進行。

第一步:要進入緩衝區的封包先查看緩衝區內是否有標示 0 的空間，如果有的話，就放入標示為 0 的空間中，並把標示改為 1，如果全部標示都是 1 的話，表示現在緩衝區是處於溢滿的狀態，不能放資料到緩衝區內，如果此未進入緩衝區的封包超過它該播放的時間，這個封包就會被丟棄，如果此未進入緩衝區的封包未超過它該播放的時間，這個封包繼續等待緩衝區有標示 0 的空間，直到超過它該播放的時間，當緩衝區資料傳送到解碼端時，標示 1 就改成 0。

第二步:當資料封包進入到緩衝區時，並不會依序排列，而是依任意排列排列，但是取出封包時卻要依據時間標的(Time stamp) **依序取出播放。**

3.3 語音段落(Talk Spurt)與時間校正

3.3.1 語音段落 (Talk Spurt)

人在對話過程中，都有講完一段話停頓後再繼續說的特點，主要是要讓對方清楚知道說話者所要表達的意思，也就是人和人對話過程中並不是時時刻刻都是處於講話的狀態，可能只說了幾句話便停頓下來，然後再接著說下一句話，在所講的每句話中字與字之間也有停頓下來的時候，所以為了能正確區分出每一段話，在本論文中便定義出語音段落 (Talk Spurt) **，但是停頓的過程根據不同的說話者而言，語音長度可長可短，對網路來說如果長時間都是處於不說話的狀態下，網路所傳送的封包還跟一般有聲音的封包相同的話，對網路而言是很浪費頻寬，所以在** G.729[5] **和** G.723.1[4] **等不同語音壓縮標準下都有定義靜音封包**(Silence packet)**，這種封包所佔的資料量較少(表** 3.1 **有做說明) ，這在沒有對話的情形下較節省網路頻寬，而這種封包也是決定語音段落長短的依據，至於如何決定語音段落的依據，**

本論文是根據遇到有意義封包（Active packet）前，連續遇到一定個數的靜音封包(Silence)下定義這段語音一個語音段落。

電話兩端在通話過程中，任何一方真正說話的時間可能只佔全部通話時間的百分之四十左右，其他時間可能是講話中語氣停頓、或聆聽對方講話、或者是雙方都在沈默中，那網路對於這些靜音封包就沒有傳送的必要，所以當傳送的資料是靜音封包時，就先把語音段落找出來，並把多餘的靜音封包丟棄，這樣做不僅可以減少封包的傳送，更可以利用前一個語音段落所統計的一些參數來決定下一個語音段落前所需緩衝的個數，除此之外，更可以因為丟棄的靜音封包而減少在每一個語音段落之前而多增加的緩衝封包延遲時間，這也是所謂的靜音封包偵測（Silence Detection) [16]，上面提到的靜音封包丟棄方式將在下一節語音時間校正說明，圖 3.5 為利用 Voice Smooth 來改善 VOIP 的示意圖。在圖中可以看到兩種語音傳送情形，在沒有建立 smooth buffer 的情形下，語音經過網路的延遲之後，在接受端解碼前會發現所接收到的語音和原來的語音封包已經不同，所接收的語音封包，除了有原了的語音封包之外，其中也夾雜著一些因網路延遲而造成的時間間隔，這種狀況下，會讓接收端播放的語音聽起來斷斷續續。另一種語音傳送情形是為了避免第一種的情形發生，所以在接受解碼端前，建立一個 smooth buffer，而這個 smooth buffer 的功用，就

是要解決網路延遲而造成的時間間隔，方法就是語音在還沒解碼前，先在一個緩衝區內等待一段時間，讓因網路延遲而造成的時間間隔集中在原始語音之後，這種狀況下，會讓接收端播放的語音和原始語音相同，只是多了一段緩衝區等待的時間。

至於語音段落的判定規則為語音經過編碼後，當接受端遇到連續 3 個靜音封包（Silence packet）出現時，接受端就視為一個語音段落的結束，而後面接踵而來的靜音封包（Silence packet）對接收端來說就不是那麼重要，可以利用語音校正把這些不重要的靜音封包給丟棄，來減少因緩衝區的出現而增加的語音播放時間，圖 3.6 就說明如何判定語音段落（Talk Spurt）。在圖中，原始的語音封包包含了許多有意義的封包（Active packet）和靜音封包（Silence packet），而判定語音段落的方法就是，當語音封包連續遇到 3 個靜音封包時，就把這段語音視為語音段落的結束，直到遇到有意義的封包，另一個語音段落才開始。所以圖上可以很清楚看到原本的一段語音，已經被切成 3 個語音段落，而在每段語音段落後面接踵而來的靜音封包可以用來作語音時間校正的依據。

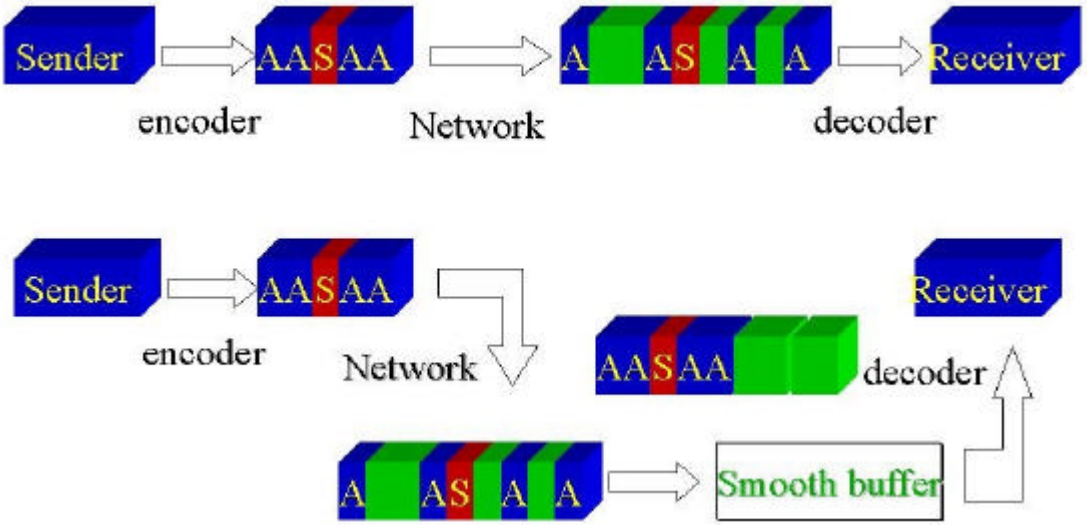

圖 3.5 Voice Smooth **改善** VOIP **示意圖**

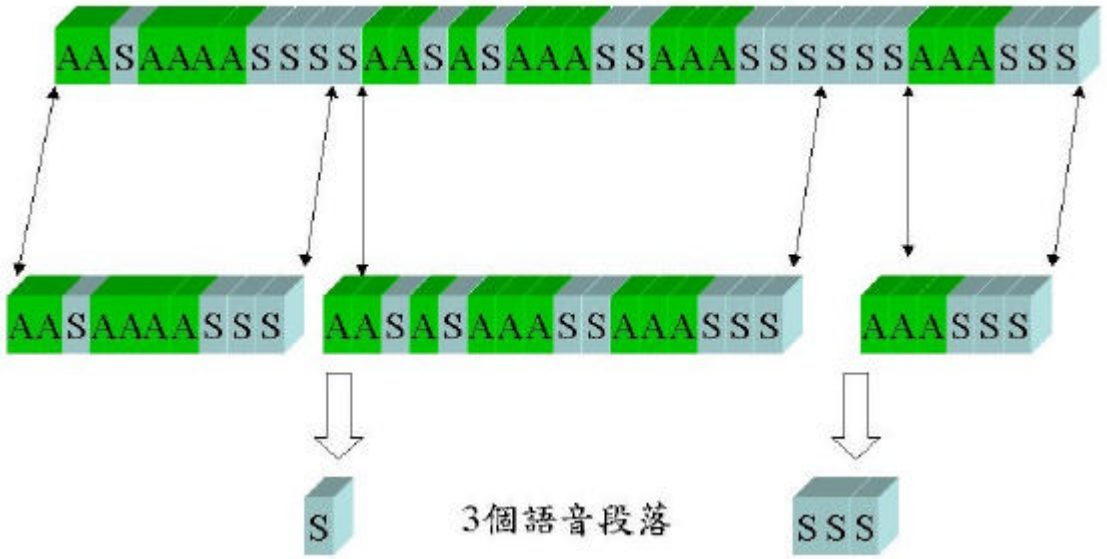

圖 3.6 **語音段落說明圖**

3.3.2 **語音時間校正**

網路的時間延遲會讓語音播放的時間愈來愈長，讓原本只需 10 **秒鐘便可以播放結束的語音，卻要花費** 11 **秒才可以播放結束。單單**

幾秒的語音長度是看不出這種現象對語音的影響，但在網路傳送的語音是連續不斷的、即時的，並不是只有幾秒鐘便結束，如果沒有適時的讓語音長度變短，會讓播放的語音愈積愈長，讓接受端花費更多的時間來聽對方的談話，所以在這一節將討論如何透過語音時間校正的動作，讓語音播放長度不會因時間延遲及緩衝區的存在而讓語音播放時間愈來愈長。

在上節中有提到如何決定語音段落的準則，在這裡語音時間校正的觀念也是經由語音段落延伸出來的，在前面介紹說利用連續 3 個靜音封包（Silence packet）來作為語音段落的區隔，而連接在後面的其他靜音封包（Silence）如果任意丟棄，這對語音來說是不正確的，就靜音封包而言它還是有存在的必要，如果任意的丟棄將會讓語音段落和語音段落之間聽起來是快速的，並且任意丟棄語音也是不合理的，真正的做法還是必須播放這些靜音封包，真正能丟棄這些靜音封包是要依照語音時間校正的原則。

語音校正是以靜音封包來完成的，其處理方法請參考圖 3.7，在圖中，A 為有意義封包（Active packet）、S 為靜音封包（Silence packet），在圖中可以看到在封包A1~A3中夾雜著許多的靜音封包 S1~S8，藉由語音段落的判斷法則，可以知道語音段落的結束是在 S3，每一個語音封包都有一個時間延遲值(d1~da)，每一個語音封包

除了考慮時間延遲之外還要考慮語音本身播放的時間，所以語音封包播放時間以S4為起始點來說，S5播放時間是d5+10，以此類推A3到達的時將是da+50。而語音校正方法是當A3到達的時間在S4~S8前，那S4~S8之語音封包將會被演算法加速取出並丟棄，省下播放靜音封包的時間用來校正系統常因網路延遲使的的語音撥放長度會大於原有說話者長度的問題，藉由這些丟棄的語音封包讓每一個語音段落時間不會愈來愈長。為了更進一步時間校正的動作，下面將以一個狀態圖做說明。

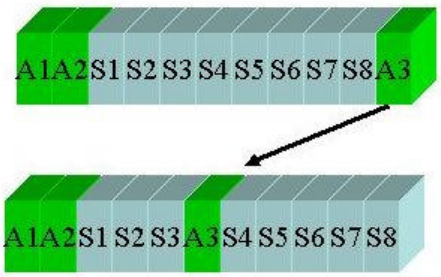

圖 3.7 時間校正說明圖

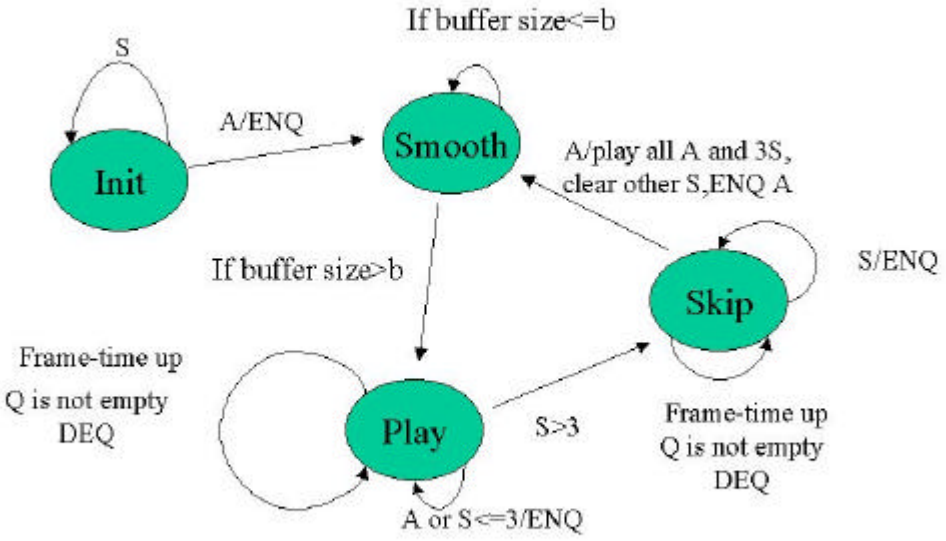

圖 3.8 時間校正狀態圖

在圖 3.8 時間校正狀態圖中，可以看見四個狀態圖，分別代表初始狀態(Init)**、緩衝狀態**(Smooth)**、播放狀態**(Play)**、節省狀態**(Skip)**，在圖中 A 表示有意義封包、S 表示靜音封包、b 表示緩衝區的緩衝個數，**ENQ **表示把封包放入** Queue **中，**DEQ **表示從** Queue **放出來，也就是把語音播放出來的意思。而時間校正的動作是採用語音封包一到達接受端時先利用初始狀態判斷語音封包形態，如果是** A **的話代表現在是語音段落的開始，接著便開始進入緩衝狀態，如果是** S **的話，就繼續停留在初始狀態；在緩衝狀態中，是根據緩衝區是否溢滿來決定是否跳到播放狀態，當緩衝區達到溢滿狀態時，為了能讓**

緩衝區能夠在放入新的語音封包，便進入播放狀態播放語音，直到連續播放到 3 個靜音封包之後才因語音段落結束而跳到節省狀態，在節省狀態中，語音段落之後的靜音封包並不是可以任意丟棄，而完全不播放。真正開始丟棄靜音封包是在節省狀態下接收到有意義封包時，才開始丟棄有意義封包之前和語音段落之後的靜音封包，在這裡要特別注意，在節省狀態下要完全播放所有有意義封包和接在有意義封包之後的連續 3 個靜音封包，但在節省狀態下遇到靜音封包仍然照常播放出去。

3.4 語音評估

3.4.1 SSSNR

一般單純的語音評估方式不外乎主觀方式和客觀方式，而客觀方式一般只需要算出語音的 SSSNR 值便可以判定語音的好壞，所以在這一小節將對 SSSNR 做一個完整的說明，並藉由這種評估方式來驗證語音品質的好與壞，至於如何運算在下面將做一個說明:

由於語音壓縮的方法都是將一長段語音切成一小段一小段的音框，如 G.729 和 G.723.1 亦是如此，因此最常被使用在語音品質評估方法就是 SSNR (Segmental Signal-to-Noise Ratio) ，這個方法是以音框 (Frame) 為單位，先計算出每一音框的 SNR (Signal-to-Noise Ratio) 值，最後再把各個音框的 SNR 值加總平均出來即為 SSNR

值。假設原始的語音訊號為 $s(k)$，經過運算處理後的語音訊號為 $\hat{s}(k)$，且一個音框有 N 個語音取樣訊號（Sample）、共有 M 個音框的話，則 SSNR 的計算如（3-1）。

i 為音框的計數器，k 為音框中的語音取樣訊號計數器

$$SSNR = \frac{1}{M}\sum_{i=1}^{M}\left(10\log_{10}\left(\sum_{k=1}^{N}\frac{s(k)s(k)}{(s(k)-\hat{s}(k))^2}\right)\right) \quad \text{.....(3-1)}$$

SSNR 方法的好處是簡單，並且可以明確判斷語音波形的相似度，SSNR 值越高語音品質越好。不過 G.729 與 G.723.1 之類以 CELP 為基礎的語音編碼方法，其基本的編碼架構主要還是以頻譜方面的線性預測濾波器參數為主，因此頻譜方面的評估方法是較適用的，有一種頻譜方面的評估方法稱為頻譜權重 SSNR（weighted spectral SSNR）[7]，其計算方法如式（3-2）所示。

$$Weighted\ Spectral\ SSNR = \frac{1}{M}\sum_{i=1}^{M}\left(10\log_{10}\left(\sum_{k=1}^{N}W(k)\bullet\frac{s^d(k)s^d(k)}{(s^d(k)-\hat{s}^d(k))^2}\right)\right) \quad \text{....(3-2)}$$

式（3-2）中的 M 及 N 同式（3-1）的定義，$W(k)$ 是可自行調整的權重係數，$s^d(k)$ 及 $\hat{s}^d(k)$ 是式（3-1）中的 $s(k)$ 及 $\hat{s}(k)$ 的頻譜能量（Power

spectra ），即$s^d(k)$是 $s(k)$經 N 點對 N 點離散傅立葉轉換（ Discrete Fourier transform ）後的 N 個係數取絕對值所得之結果，$s^d(k)$及$\hat{s}^d(k)$的運算如（ 3-3 ）所示。

$$s^d(k)=\left|DFT_N\{s(n)\}\right|=\left|\sum_{n=1}^{N}s(n)\exp(-j2\pi kn/N)\right| \quad ,1\le k\le N$$
$$\hat{s}^d(k)=\left|DFT_N\{\hat{s}(n)\}\right|=\left|\sum_{n=1}^{N}\hat{s}(n)\exp(-j2\pi kn/N)\right| \quad ,1\le k\le N \qquad \text{.....(3-3)}$$

這個方法主要是評估語音在頻率方面係數的相似度，並且依人耳對頻率感覺的靈敏度加上權重係數，因此是一個不錯的語音評估方法。不過由於權重係數必須選擇恰當，否則所得到的結果會不客觀，因此設定權重係數 $W(k)=1,\ k=1,..,N$，如此便使每一個係數重要性均相等，因此式（ 3-2 ）可簡化為式（ 3-4 ），並將結果稱為 Spectral SSNR (簡稱 SSSNR)。

$$SSSNR=\frac{1}{M}\sum_{i=1}^{M}\left(10\log_{10}\left(\sum_{k=1}^{N}\frac{s^d(k)s^d(k)}{\left(s^d(k)-\hat{s}^d(k)\right)^2}\right)\right) \qquad \text{.....(3-4)}$$

SSSNR 語音評估方式在一般評估語音是常常使用的評估方式，藉由這種評估方式可以斷定語音品質是好或是壞。

在 VoIP 環境下，語音會因為網路延遲的關係而造成封包的遺

失，這對語音來說是一大傷害，所以對於遺失的封包來說，就把遺失的封包當成零來計算，並求得其 SSSNR 值來判斷語音的好壞，把遺失的封包當成零來計算其 SSSNR 值需要注意一點，那就是如果遺失的封包是擁有很高能量的話，把這個封包當成零對於計算其 SSSNR 是不太適當的，所以對於經過網路延遲的語音將透過主觀評估方式播放出來，來讓聽到的人來評估語音的好與壞。

3.4.2 封包遺失率與語音品質之比較

利用隨機的封包遺失率針對一段語音做一個量測的計算，從不同的封包遺失率中計算出不同的 SSSNR 值，藉由這些數值的計算，能夠定出一個封包遺失率（Loss rate）評量的依據，圖 3.8 是根據不同封包遺失率所產生的不同 SSSNR 值的分布圖，並依據這個數據分布圖，來作為緩衝區動態調整演算法（Algorithm）的調整界限，這些數據是以隨機的封包遺失率連續作 100 次之後，所得到的平均 SSSNR 值，表 3.2 是圖 3.9 不同封包遺失率對不同 SSSNR 的數值分布。

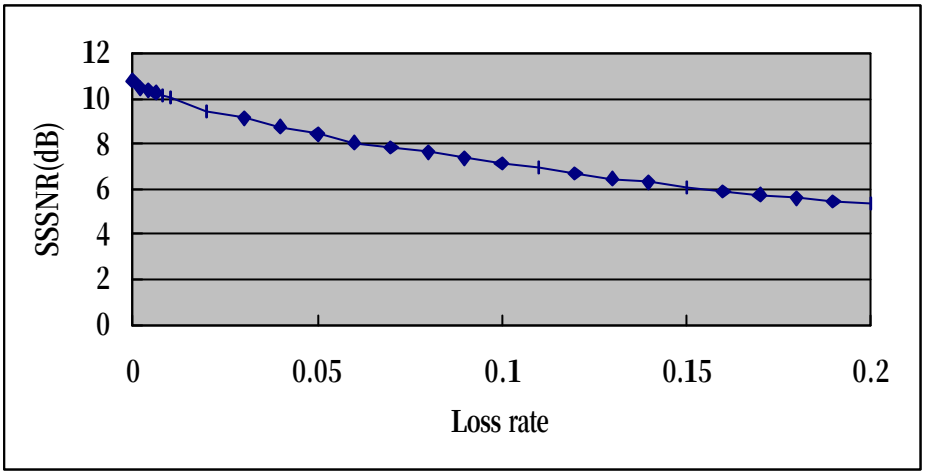

圖 3.9 不同封包遺失的 SSSNR 值分布

Loss Rate	SSSNR	Loss Rate	SSSNR
0	10.77597	0.09	7.375709
0.002	10.43704	0.1	7.152715
0.004	10.35496	0.11	6.934948
0.006	10.24884	0.12	6.705656
0.008	10.13402	0.13	6.461011
0.01	10.06384	0.14	6.308104
0.02	9.437043	0.15	6.096691
0.03	9.120939	0.16	5.903226
0.04	8.740501	0.17	5.763178
0.05	8.417483	0.18	5.598251
0.06	8.037152	0.19	5.484069
0.07	7.821378	0.2	5.373203
0.08	7.613389		

表 3.2 不同封包遺失率的 SSSNR 值分布

由圖 3.9 和表 3.2 的圖形分布及數據下，可以知道在多少 SSSNR 值範圍之內語音品質是可以接受的，如此便可以推斷出當封包遺失率在何種範圍內是可以忍受的，當發現封包遺失率已經超過可以接受的範圍時，便需要在語音解碼端前增加緩衝區的個數，好讓封包遺失率減少並提升語音播放品質。

從數據方面看來，在沒有任何封包遺失的情形之下，經過 SSSNR 的計算原始語音的 SSSNR 值是 10.77597 dB，經過不同的封包遺失率之後可以看到其 SSSNR 值正快速的遞減，在遺失率 0.2 時 SSSNR 值已經掉到本來的一半了，這對語音而言，已經是處於不能忍受的地步，所以就客觀方面的數值方面而言，在遺失率低於 0.1 之下，算是在可以接受範圍之內，因為其 SSSNR 值仍然還有 7dB 以上，所以本論文定出如果語音封包遺失率超過 0.1 以上的話，語音壓縮解碼端前的緩衝區就要調高緩衝個數。

上面的數據是比較客觀的，如果再搭配上真正撥出來讓大家聽聽看那對別人就更有說服力，圖 3.10~圖 3.13 是不同的語音封包遺失率的語音波形圖，希望藉由語音波形讓大家看看不同封包遺失率下所造成的影響，圖 3.10 是語音的原始波形。

在圖 3.10~ 圖 3.14 中，可以看出語音波形有重複的情形產生，

這是因為語音長度不夠長的話，在作封包遺失率時，遇到少量的封包遺失很容易就會讓封包遺失率上升，為了避免這種情況發生，就把原始語音的長度作重複的動作，讓語音長度能變長，讓封包遺失率能更準確，這就是圖 3.10~圖 3.14 中，語音波形重複的原因。

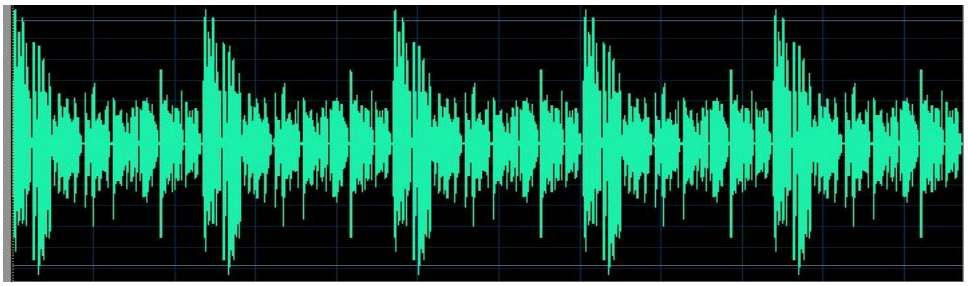

圖 3.10 原始語音波形

圖 3.11 Loss Rate 0.02 語音波形

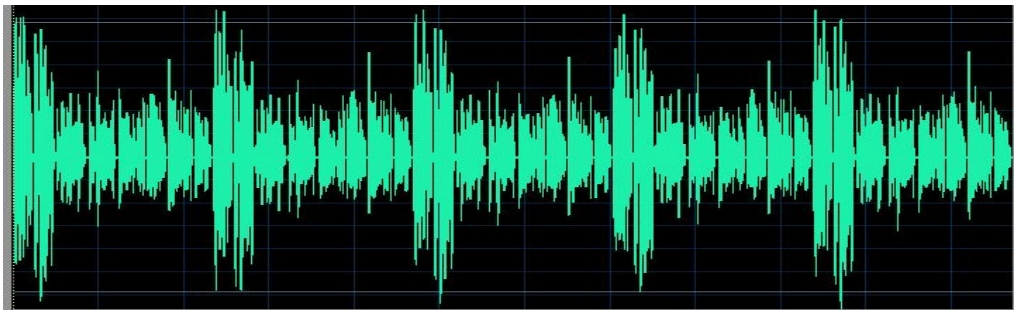

圖 3.12 Loss Rate 0.1 語音波形

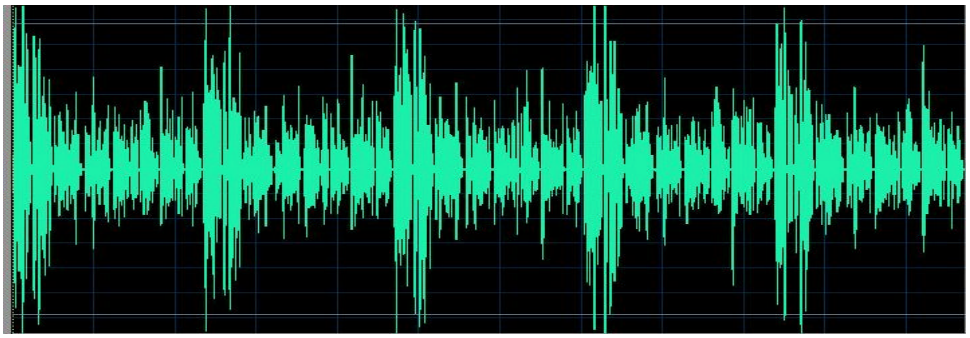

圖 3.13 Loss Rate 0.15 語音波形

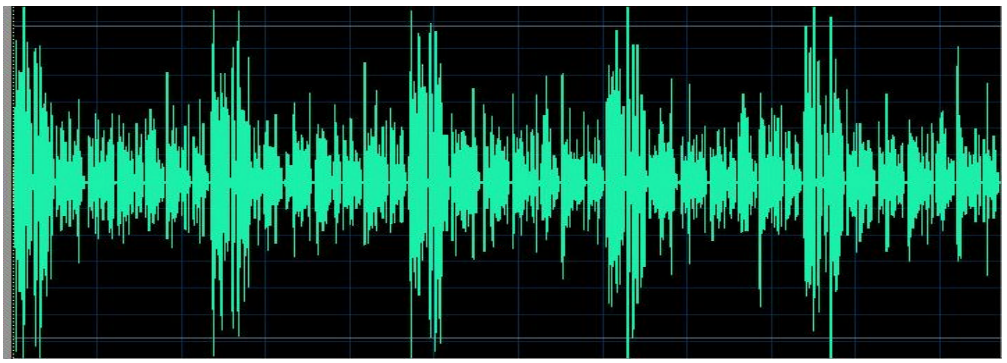

圖 3.14 Loss Rate 0.2 語音波形

因為在封包遺失率 0.2 時其 SSSNR 值已經驟減成原來的一半，所以所作的主觀分析就只考慮到封包遺失率 0.2 以下，在經過播放軟體 CoolEdit 播放之後，在封包遺失率 0.2 時只可以勉強聽出語音所要表

達的意思，所以在考慮主觀及客觀下，本論文就以語音封包遺失率 0.1 和 0.2 作為緩衝區調整的基準。

3.4.3 加強語音評估方法

那如何判定網路傳送語音的好壞呢?如果是單純的一般語音，那只需要算出語音的 SSSNR 便可以判定語音的好壞，但是透過網際網路傳送的語音會因為網路的關係而發生封包的遺失、延遲，所以在研究即時傳送語音之前要先定一個語音品質評估的方法。

如果以 SSSNR 來計算語音品質的好壞，勢必要讓原始語音和解碼後的語音長度相同，才可以做計算，但是經過網路的時間延遲及緩衝區的配置後，會讓解碼後的語音長度和原始語音長度不盡相同，所以為了能讓解碼後的語音長度能夠和原本語音長度能相同，在這裡並不會探討額外增加的緩衝封包及時間延遲封包，而只針對解碼後語音封包是否有遺失，當解碼後的語音封包遺失時，就以 0 來補這個遺失封包的位置，如此便可以讓解碼後的語音長度和原始語音長度相同。