

第四章 動態緩衝區調整方法

此論文研究方向即在發展出在 VOIP 架構下語音封包延遲時，如何讓語音播放更平順並減少語音品質的方法。並結合廣泛使用在網際網路閘道器的語音壓縮標準 G.729，並加入靜音抑制方法，針對通話雙方的談話內容做討論，只有在對話進行對話時才傳送封包，而對通話雙方沒有在通話下則是為靜音而不傳送封包，如此的方法則可以減少因網路延遲而造成語音的品質低落。如果在依據網路壅塞的情形，加入聲訊封包緩衝區緩衝排程演算法 [7]-[10]，則能讓 VOIP 語音封包更能順利傳送 [11]-[15]，減少網路延遲造成的語音品質傷害，提高語音播放品質。緩衝區的調整有分靜態和動態兩種，靜態是每次的緩衝個數都是維持一定的個數，並不會隨著網路狀況而變動，而動態調整的緩衝個數是會隨著網路狀況而變動，會根據網路的狀況給於不同的反應，並且動態調整的緩衝區可以縮短封包之間的時間差，讓語音能夠播放更平順，圖 4.1 可以看出靜態和動態緩衝區的差別，在圖中可以很明顯看出動態緩衝區能夠很逼近原來的封包延遲曲線，讓封包時間差變小。

本論文是選用 G.729 來作為語音的壓縮基礎，主要原因是 G.729 有較小的封包並且還有靜音封包的偵測方式，所以才採用語

音壓縮 G.729 做為 VoIP 在網路傳送的語音。雖然採用較小的封包可以減少網路延遲的效應，但是單單只用較小的封包傳送還是難保不會因網際網路的因素而讓語音品質下降，所以為了能讓語音品質仍能保持在一定水準下。在本章提出不同動態緩衝區調整辦法 (Dynamic buffer smoothing methods) 來改善網路延遲的方法，至於動態緩衝區調整辦法的詳細說明，下面的章節將會敘述，希望利用各種不同的緩衝區封包調整機制讓語音得到更好的語音品質。

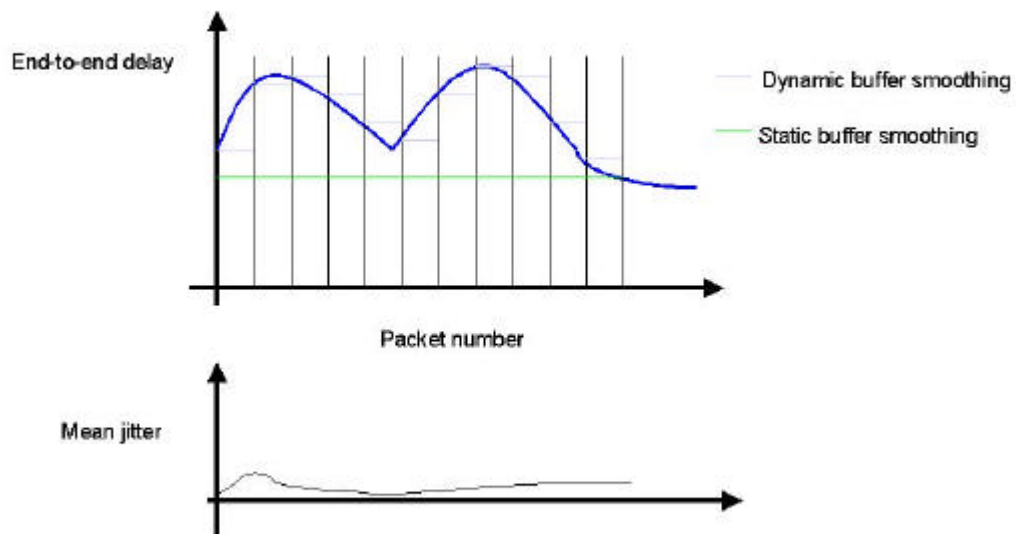


圖 4.1 靜態與動態緩衝區之比較

4.1 Per-packet 緩衝區動態調整

在參考論文中[15][16][17]中以這個方法最可以減少封包遺失

率，和即時反應出網路狀況，而這個調整方法是參考這篇論文[15]。

緩衝區的調整方式如果只針對每一個語音封包做調整，更可以讓語音封包遺失率大大降低，因為只是針對每個封包做調整，所以調整所影響的範圍也只限於下一個語音封包，因此降低封包遺失率。圖 4.2 是 Per-packet 緩衝區動態調整會利用到的參數，在下面定義出每個字母所代表的意思。

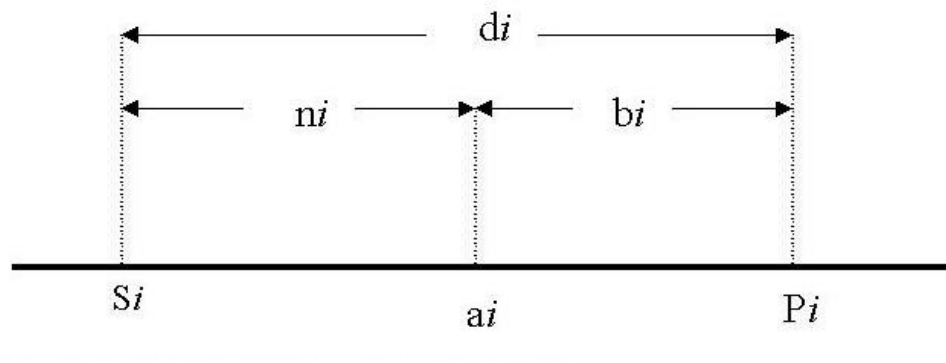


圖 4.2 Per-packet 參數

S_i ：表示第 i 個封包經由編碼後的時間

a_i ：表示第 i 個封包經過網路時間延遲之後到達解碼端之前的時間

P_i ：表示封包播放的時間

b_i ：表示第 i 個封包總共緩衝多少個封包

n_i ：表示第 i 個封包在網路所花費的延遲時間

d_i ：表示第 i 個封包從經過編碼端之後到語音封包順利播放出去，所花費的時間

Per-packet 緩衝區動態調整的方式是針對不同的網路狀況作不同的調整方式，而網路是否處於劇烈的狀態是由一個變動值（ Spike ）來判斷，這個變動值是可以任意定義的。

下面是 Per-packet 緩衝區調整的公式，因為 Per-packet 緩衝區調整是依據每一個封包作調整，所以當遇到網路變動劇烈的情況下，調整的方式便是以當時封包的延遲為緩衝區的調整個數，而網路狀況處於穩定變動時，就以機率的方式來調整緩衝區， α 代表一個調整係數，其值介於 1~0 之間，並藉由計算上一個的緩衝個數，和現在的時間延遲來決定下一個緩衝區的個數，因為是以機率來計算所以所計算出來的值會變得很小，這也表示緩衝區在穩定狀態下只需少數的緩衝區，在公式中，frame-time 是指一個語音封包所代表的時間，請參考表 3.1。

當網路時間延遲劇烈變化時

$$b_{i+1} = n_i / (\text{frame} - \text{time}) \dots\dots\dots (4-1)$$

當網路時間延遲穩定變化

$$b_{i+1} = (1 - \alpha) \times (n_i / (\text{frame} - \text{time})) + \alpha \times b_i \dots\dots\dots (4-2)$$

當網路時間延遲劇烈變化時，緩衝區的調整就以公式（4-1），公式（4-1）是根據每一個劇烈的網路時間延遲來調整緩衝區，這個方

式會大量增加緩衝區來減少晚到的語音封包遺失，而當網路時間延遲穩定變化時，就以公式（4-2）來調整緩衝區，以增加少量的緩衝區或者是不用增加，Per-packet 這種調整方式可以降低封包遺失率，相對地，常常根據每一個封包做調整會讓接受端的語音聽起來是斷斷續續的，語音忽快忽慢。

4.2 Late-rate 緩衝區動態調整

語音段落緩衝區調整就是依照一定封包數目內幾個語音段落的封包遺失率來做緩衝區封包的增加或減少，因為要考慮語音段落有長有短，所以以一定數目的封包來作調整的依據較正確，在這裡以大於 200 個封包為依據，並且也要是語音段落的結束，在連續 200 個封包之後計算其語音段落封包遺失率，如果其封包遺失率大於調整的標準時就做一次調整，所以每當在數個語音段落之後（也就是 200 個封包之後），在下一個語音段落開始前，先作緩衝區配置，在這裡選擇 200 封包是依據調整界限的精準度來說，當調整的界限為 0.01 的封包遺失率時，所代表的意義是 100 個封包掉 1 個封包，但是語音段落有長有短，如果在短的語音段落中封包數不到 100 個，那只要遺失 1 個封包便超過調整界限，那永遠是處於挑整的狀態，所以在這才採用 200 個封包或是更多。除此之外，網路的延遲（Delay）情形是有

延續性的，當在接受端發現現在語音封包遺失率已經達到一定程度時，可能接下來的封包遺失率情形可能會更糟，所以就加大緩衝區的個數來減少封包的遺失，如果現在網路的封包遺失率已經漸漸減少，那就意味著現在網路狀況很好了，這也代表著現在已經不在需要大量的緩衝個數，接受端便可以漸漸減少緩衝的個數，增加緩衝區這個動作會讓語音聽起來較平滑但是卻會造成語音播放時間延遲時間更久，減少緩衝區這個動作卻會讓語音封包遺失率增加，降低語音的品質，語音播放品質和語音時間延遲正是魚與熊掌不可兼得的關係，而調整的方式可分為靜態和動態兩方面。

靜態調整方法中，在本論文中是依照前幾個語音段落（大於 200 個封包）的封包遺失率來作為下一個語音段落時所需緩衝的封包個數的依據，在前一節在探討封包遺失率與語音品質關係中，可以知道在語音封包遺失率在 0.1 之前語音的品質都是可以接受的，所以便有一個簡單的算式說明：

$$\begin{array}{l} \text{If (Loss rate) > 0.1} \\ \text{buffer}_{i+1} = \text{buffer}_i + 1 \end{array} \dots\dots\dots (4-3)$$

$$\begin{array}{l} \text{If (Loss rate) <= 0.1} \\ \text{buffer}_{i+1} = \text{buffer}_i \end{array} \dots\dots\dots (4-4)$$

靜態緩衝區調整辦法是不能有效的反應網路狀況，如果不同的封包遺失率都調整相同的緩衝封包數的話，是永遠追不上封包遺失率的，反而只會因增加緩衝區個數而讓語音延遲（Delay）越來越長，所以在語音遺失率很糟的情況下，需要把語音段落的緩衝區封包個數做大量增加的動作，如果語音封包遺失率不太糟的話，那下一個緩衝區增加的緩衝個數就較少，這個方法就是動態的調整方法。

動態緩衝區調整方面，更詳細的說明就是不再以一定的封包數為基準，因為遇到網路變動很劇烈的情形時，靜態緩衝區封包的調整已經不能滿足我們的需求，當網路封包遺失率不大時，我們以少量的緩衝區封包來調整是可以有效的解決語音延遲問題，但是當遇到網路劇烈變動時，仍以少量的緩衝區封包來調整那對語音品質是一點幫助都沒有；反過來說在網路劇烈變動下，大量的緩衝區封包調整是相當的有效，但是遇到網路變動不劇烈的情形下，還是做大量的緩衝區封包調整那對時間延遲來說無疑是雪上加霜，原本只需少量的緩衝區封包就可以達到想要的結果，現在卻多付出一些語音封包更增加了語音時間延遲。

為了不讓語音會因緩衝區封包大小的因素而影響語音的品質，就必須根據不同的網路狀況做一些不同的調整方式，而調整的方式仍是依據前一節所說利用前幾個語音段落(大於 200 個封包)封包遺失率來

做緩衝區封包個數的變動，但是現在將不再是以單一種封包遺失率來做緩衝區封包數調整的依據，而是根據不同的封包遺失率做不同的調整，下面就作一個簡單的算式說明：

$$\begin{aligned} &\text{If (Loss rate) } > 0.1 \text{ and } < 0.2 \quad \dots\dots\dots (4-5) \\ &\text{buffer}_{i+1} = \text{buffer}_i + 1 \end{aligned}$$

$$\begin{aligned} &\text{If (Loss rate) } > 0.2 \quad \dots\dots\dots (4-6) \\ &\text{buffer}_{i+1} = \text{buffer}_i + 2 \end{aligned}$$

$$\begin{aligned} &\text{If (Loss rate) } \leq 0.01 \text{ 連續三次} \quad \dots\dots\dots (4-7) \\ &\text{buffer}_{i+1} = \text{buffer}_i - 1 \end{aligned}$$

假如前幾個語音段落遺失率(Loss Rate) 不是上面三種時

$$\text{buffer}_{i+1} = \text{buffer}_i \quad \dots\dots\dots (4-8)$$

Late-rate 動態調整方法請參考圖 4.3，在圖中可以清楚看到緩衝區的個數會隨著網路不同的遺失率而作變動，而這個調整方式是建立在語音解碼端前，Late-rate 調整方式是利用語音段落封包遺失率來評估緩衝區大小，這種調整方式可能會造成封包調整不夠快或是不能即時反應網路狀況而造成接下來的語音段落封包大量的遺失，特別是對

於網路狀況變動很大的時候，所以下一節將討論另外一種緩衝區調整辦法。

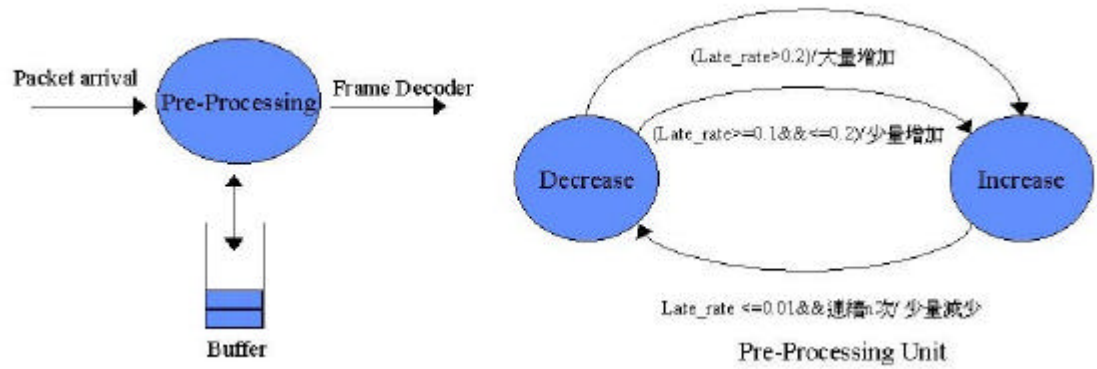


圖 4.3 Late-rate 緩衝區動態調整

4.3 Max Jitter 緩衝區動態調整

利用語音段落封包遺失率來調整緩衝區大小是不能完全保證晚到的封包不會因該播放卻沒有播放而被丟棄（Drop），追究晚到封包會被丟棄的原因是封包和封包之間的時間延遲（Jitter），如果兩者的距離差距太大時，當第一個封包播放之後，接下來的封包就要等一段相當長的時間才可以播放，而這段時間就是接受端所需要緩衝的封包數目，所以如果在語音段落內，以最大的封包時間差延（Max Jitter）來作調整的話如圖 4.4，其效果可能比利用語音段落的封包遺失率更好。

下面針對語音段落內最大封包時間差延（Max Jitter）動態緩衝

區調整作些定義。

$n_{i,0}$ ：表示在第 i 個語音段落第一個播放封包的時間延遲

(Delay) , 論文中以 $n_{i,0}$ 表示

n_{max} : 250msec , 代表語音單程最大容忍的時間延遲

利用上述的定義推導出下列的緩衝區調整公式:

$$J_{i,max} = n_{max} - n_{i,0} \dots\dots\dots (4-9)$$

$$J_{i,j} = (n_{i,j} - n_{i,0}, 0) \dots\dots\dots (4-10)$$

$$MJ_i = \text{Min}(\text{Max } J_{i,j}, J_{i,max}) \dots\dots\dots (4-11)$$

$$Snoot h_{i+1} = [MJ_i / \text{frame - time}] \dots\dots\dots (4-12)$$

$n_{i,0}$ 表示在第 i 個語音段落第一個播放封包的時間延遲

(Delay) , 並且定義封包和封包之間的時間延遲 (Jitter) 計算方式皆是參考同一個語音段落內的第一個播放封包(也就是 $n_{i,0}$) , 如公式 (4-9) , 在網路上只要太大的網路延遲皆會在網路上就被丟棄 , 根本不會傳送到接受端 , 所以在這裡才不考慮太大的網路延遲值 , 而只考慮一般可容忍的單程網路時間延遲 (Delay) 範圍 , 其範圍大都不超過 250msec , 所以才定義出最大的網路時延遲值 (Delay value) , 所定義的最大網路時間延遲 (Delay) , 可以知道最大的封包時間差延

(Max Jitter) 一定不會超過 250，並考慮所計算的封包時間差延

(Jitter) 會有負值的情況產生，所以最小值要定義為 0，所以(4-10)

是求在第 i 個語音段落內第 j 個封包的封包時間差延，當計算出在第 i 個語音段落內每個封包的封包時間差延時，就需要算出在這個語音段落內最大的時間差延 (Max Jitter)，作為下一個語音段落緩衝區的個數，如公式 (4-11)。

經過公式 (4-11) 計算出在第 i 個語音段落內最大的時間差延 (Max Jitter) 後，便可以決定第 $i+1$ 個語音段落時要緩衝多少個封包，因為論文所採用的是 G.729 語音壓縮，而且 G.729 每一個語音封包都是 10msec，所以第 $i+1$ 個語音段落緩衝封包個數就是 (4-11) 除於 10，所得的商就是第 $i+1$ 個語音段落緩衝封包個數，也就是公式(4-12) 所示，在這是以數學高斯來取其最小整數值。

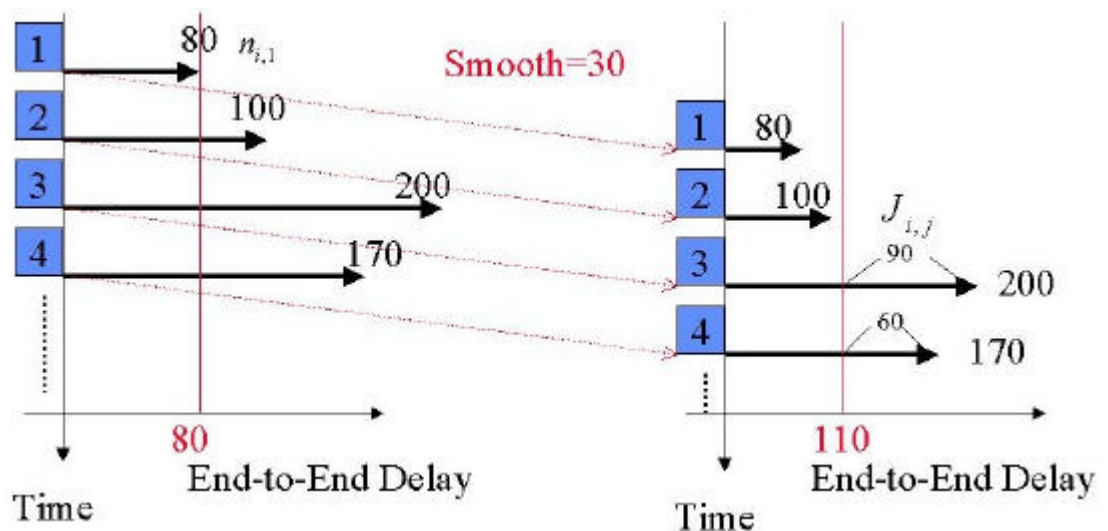


圖 4.4 Max Jitter 緩衝區動態調整