

第三章

研究方法

3.1 問題定義

在 WiMAX 的設計上，主要傳輸方式為 TDMA，根據時間的先後順序排程，根據標準所訂定的各項規則中，以 QoS 的要求為最優先，因此在設計 MAC Layer 功能時，就必須將 QoS 納入考量。

在第一章中我們已介紹在 WiMAX 上應該有的運作方式、機制與各項參數，但由於 WiMAX 標準並未在 connection admission control、bandwidth request、bandwidth allocation 及 scheduling 上規劃出基本的實際運作方式，全留待給設備商以及服務提供商實作，因此我們發現下面五個問題：

3.1.1 Bandwidth Request

WiMAX 在上行頻寬的請求，採用類似 802.11 競爭的作法，詳細過程如圖 1.3 所示。由於在 WiMAX 的環境中，上行是採用 TDMA 的方式，也就是照時間來排程相關資料傳送的順序，於是所有的頻寬決定權全在 BS 上，以致於低優先權連線即使成功傳送 BR 至 BS，也無法保證能夠獲得頻寬。然而，在 802.11 卻是使用 CTS、RTS 封包來宣告此時此刻是由喊話者來傳送資料而非經由 BS 來規劃，因此會發現在 WiMAX 上設計 contention 必須使用不同於 802.11 作法。最主要的原因是在 WiMAX 的環境上有 QoS 的設計，當系統處於高負載的情況下，即使低優先權的連線成功傳送 BR 封包至 BS 也無法保證一定能獲取相對應的 bandwidth。

3.1.2 rtPS 封包需在限定時間到達才算有效

rtPS 連線通常使用在像 video 這類的連線，因此會有時間限制的考量。當影片在播放時會有一段緩衝時間(buffer time)，當超過緩衝時間而封包卻未到時，會造成影片播放不流暢或影片中斷等現象，這段緩衝時間通常稱之為 Max_Latency。假設 rtPS 進入 queue 的時間為 T，則其限制時間(deadline) = T + Max_Latency，在限制時間內接收者收到的封包都算有效。假如超過限定時間，即使成功接收也應視為無效的封包。因此，必須針對 rtPS 的 queue 做 deadline time 的控管，以免頻寬被用來傳送無效的封包，而浪費了資源。

3.1.3 Bandwidth Allocation

在頻寬配置議題上，由於標準並未訂定實作方法，於是各家有各家的作法，至今最常見的方法為 strict priority queue。該方法最大的優點在於可以滿足高優先權連線的頻寬需求。但最大的缺點是可能會造成低優先權的連線 starvation，如何在高低優先權連線間取得平衡向來是研究者努力的方向，既要保證高優先權連線的 QoS 又能讓低優先權的連線得以傳送資料。

3.1.4 Scheduling

Scheduling 可分為 BS scheduling 與 SS scheduling。BS scheduling 通常是配合 bandwidth allocation 完成，而 SS scheduling 則是重新分配(redistribute)取得的頻寬，因此在 SS 重新分配的過程中，仍舊有可能遭遇如 bandwidth allocation 所遇到的問題而造成低優先權連線的 starvation。因此在設計 SS scheduling 時應考量如何在高低優先權連線間取得平衡，即既要保證高優先權連線的 QoS 等級，又能讓低優先權的連線得以傳送資料不至於 starvation。

3.1.5 CAC 放任 BE 連線無止盡進入是否恰當？

在 QoS 的設計上，BE 連線是屬於沒有任何服務要求的連線，因此在測試 CAC 公式時，會發現 BE 連線永遠被允許進入系統，但會發現由於 BE 連線取得寬頻的方式是採用 contention 的方式，也就是說當連線一多，BR 成功的機率會急速下降，此時如再考量系統負載，會發現當系統負載已經很高的情況下，根本無法分配頻寬給那些成功送出 BR 封包的 BE 連線。因此，在考量系統高負載的情況下，CAC 放任 BE 連線無止盡進入是否恰當，是個值得研究的議題。

3.2 研究進行步驟

本研究方法採用 MAC Layer co-Function 的設計方法，此方法不論在 connection admission control、bandwidth request、bandwidth allocation 以及 scheduling，都能反應其設計的適用性，尤其在因應 Fairness 的設計上，透過 co-Function 的設計將可使 MAC layer 最主要的 function 之間可以協同合作達到最大的效用。

圖 3.1 為研究進行的四步驟：首先針對 CAC 做設計，之後是 bandwidth request、bandwidth allocation 及 UL scheduling。

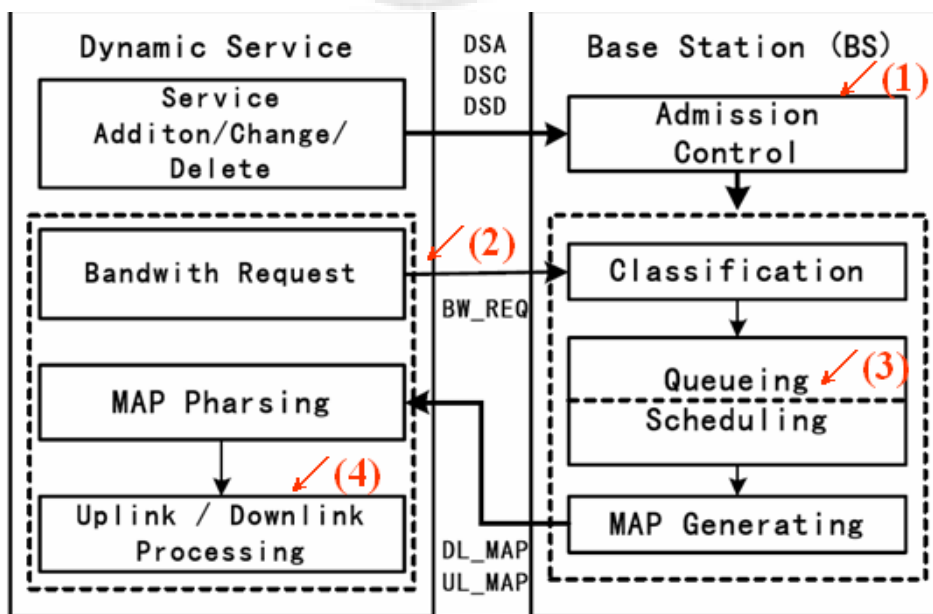


圖 3.1 Module diagram of BS and SS [1]

由於本研究著重於不同優先權連線之間的 Fairness，因此有必要先針對產生不公平的原因先做說明，由圖 3.2 我們可以發現這是在使用 strict priority bandwidth allocation 所造成的 throughput 現象，從圖可以發現低優先權的連線如 nrtPS、BE 在 throughput 有 starvation 的現象，但從圖 3.2 看來並非只是在 bandwidth allocation 上發生問題而已，因為 BE 的 throughput 一直是在 0 的位置，因此我們認為早在 bandwidth request 就已經出現效率不佳的問題，因此為了因應 Fairness 的全面考量，於是提出下列三點可能造成 Fairness 不佳的原因：

- (a) nrtPS、BE 的 bandwidth request 沒有成功送至 BS 所以沒有獲得 bandwidth(BW)。
- (b) 雖然 BR 有送到，但由於 bandwidth allocation algorithm 傾向分配 BW 給 high priority 的 connection(如 rtPS、ertPS、UGS)，以致於 nrtPS、BE 連線所獲得的 BW 不足，甚至沒有。
- (c) 雖然 UL-MAP 中 bandwidth allocation 已經有公平分配了，但由於 BS 只釋出相對應總量給 SSs，然而 SSs 的 redistributed scheduling algorithm 卻又將資源分給 high priority 連線，以致於 BE、nrtPS 連線分配不到 BW。

針對以上三個問題提出一個比較有效率的方式來解決 Fairness 在 WiMAX 上的問題。

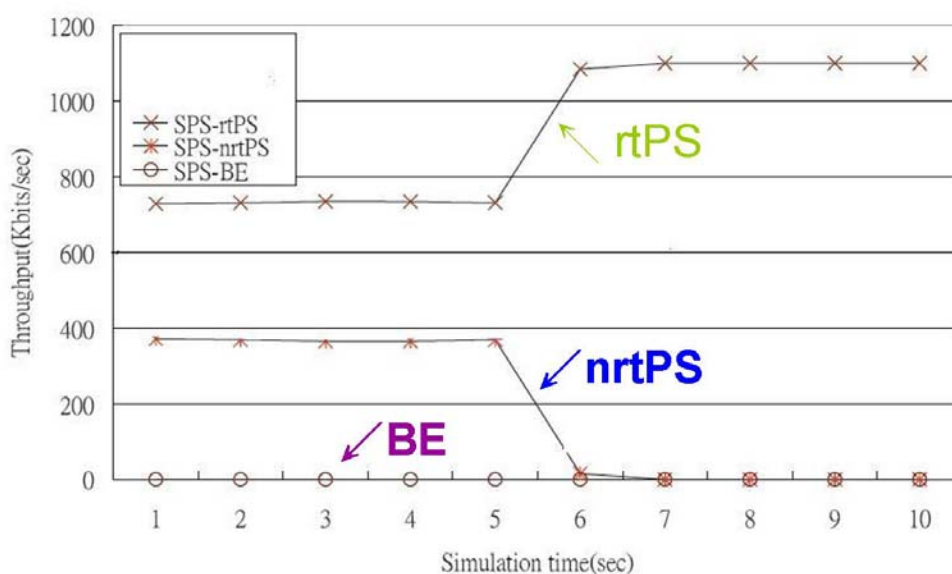


圖 3.2 Average throughput [5]

3.2.1 Connection Admission Control

一般在設計 CAC 都不會針對 BE 連線做控管，最主要是因為 BE 連線並沒有任何服務品質的要求，但由於 BE 會競爭 reserved BR slot 來送出 BR packet，我們發現當系統的 virtual loading 很高時，可供 nrtPS 以及 BE 連線使用的 BW 將會大大的降低，此時 reserved BR slot 將會相對的少，如果此時有超量的 active BE connections 將會造成所有 BE 連線都無法搶到頻寬。舉例如下：假設目前的 Reserved contention slot =10

而系統內的所有 BE connections = 10，則目前搶成功機率 33%。但由於系統幾近滿載，不足以應付 3 個成功的 BR 總和，假設可以滿足 2 個 BR packet，在此假設每次 BR 成功時配置的 BW 一樣。此時如果再允許 BE 連線進入系統，假設現在的 BE connection=20，而搶成功機率減為 10%，系統雖然可滿足 1 個成功的 BR，但事實上剩餘的一份 BW 卻浪費了。因此我們將採下列的作法

$$\text{定義 virtual loading} = \frac{\sum_{i=0}^I \sum_{j=0}^{J_i-1} r_{\min}(i, j)}{C_{\text{total}}} \quad (c)$$

當系統 virtual loading 已達 80%時，開始隨機拒絕新增的 BE 連線，而當系統 virtual loading 已達 90%，開始全面拒絕新增的 BE 連線。使用 virtual loading 的好處是可以用不用管系統實際的 loading，因為實際的 loading 變動太大，要很準確的計算出來有點難度。

3.2.2 Bandwidth Request with Dynamic Polling Interval Function (DPI Function)

由之前的介紹我們知道使用 contention 的作法，必須針對剩餘可傳輸資料空間做計算，以及 contention 會受到 active SSs 數量的影響，而且在配置時會有配置固定量的疑慮，

也由於之前的介紹發現在 WiMAX 的環境，似乎不適合使用 contention 的方式來傳送 BR，對於解決以上種種的疑慮，於是採用了 no contention 的方式，其設計理念如下

- (a)當 virtual load 低時(代表保留的 data section 小)，使用 DPI function 計算出 BE 跟 nrtPS 的 polling interval 較小

Available BW 較大	reserved slots for virtual loading
-----------------	------------------------------------

- (b)當 virtual load 高時(代表保留的 data section 大)，使用 DPI function 計算出的 BE 跟 nrtPS 的 polling interval 較大

Available BW 較小	reserved slots for virtual loading
-----------------	------------------------------------

我們設計了一個稱為 DPI function 的 MAC co-function，DPI function 的設計中會將 system virtual loading 以及優先權考量進來，因此我們取了二個參數，分別是 virtual loading 以及 rtPS 的 Max Latency (=50ms)，因此 DPI function 表示式設計如下

$$\text{DPI Function} = 5 + 10^{\text{virtual_loading}} * \text{rtPS_MAX_latency}(=50\text{ms}) \quad (\text{d})$$

- (a)會使用 rtPS MAX Latency (50ms)以及 frame length (5ms)是因為考量到優先權的設計，即使系統的負載再怎麼小(=0)，也不希望 BE 跟 nrtPS 的 polling interval 小於 rtPS_latency，如果小於的話，會造成所謂的優先權反轉現象，使得 BE 的優先權反而優於有即時頻寬需求的 rtPS 連線的 polling interval (50ms)。
- (b)會採用 10 的指數以及 virtual loading 是因為考量到 virtual loading 很大時，等量增加的 polling interval 無法即時反應 high loading 的環境，因而造成的低落效能，因此考量當 virtual loading 愈來愈大時，為了因應龐大的頻寬需求，決定採用指數的方式來快速遞增 polling interval 的長度。

3.2.3 Bandwidth Allocation Work Coordination with Polling Interval Function

對於 UL bandwidth allocation 我們知道 BS 並沒有辦法知道 SSs traffic queue 的實際狀況，BS 只能知道 SSs 送上來的 bandwidth request 封包的值，因此爲了要滿足 QoS

的要求，BS 唯一的辦法就是永遠先滿足那些 high priority connection 的 BR，而忽略 low priority connection 的 BR 封包，[1][2][3]的作法雖然可以滿足 Fairness 的要求，但卻有可能造成 high priority connection 無法滿足 QoS 的要求，因為其將頻寬讓給 low priority 的連線使用了。

因此，為了避免這樣的情形再度發生，我們必須把 virtual loading 考量進來，當 virtual loading 高的情況下，大部分的 BW 應該讓給 high priority connection 使用，而當 virtual loading 低的情況下則可以把多一點的 BW 分配給 low priority connection 使用，因為 DPI function 可以反應出 virtual loading 以及優先權的觀念，因此在 UL bandwidth allocation 的設計中，我們將使用 DPI function 來設計 BW allocation。BW allocation 執行的流程有二個步驟，分別如下

首先 Round 1 會先滿足基本的 Rmin 需求，由於 UGS、rtPS、以及 nrtPS 都有 Rmin 的要求，因此針對系統所有的連線會依序分配如下：

$$UGS = \sum_{ss=1}^n UGS_n * UGS_min$$

$$rtPS = \sum_{ss=h}^n nrtPS_n * nrtPS_min$$

$$nrtPS = \sum_{ss=1}^n rtPS_n * rtPS_min$$

在 Round 2 階段則會將剩餘的 BW 依比重分配，直到 bandwidth 用完為止，其分配順序以及數量如下所示：

$$rtPS = \min \left\{ BR - R_{min}, BW * \frac{DPI\ function()}{rtPS\ latency + DPI\ function()} * \frac{1}{rtPS_n} \right\}$$

$$nrtPS = \min \left\{ BR - R_{min}, BW * \frac{rtPS_latency}{rtPS_latency + DPI\ function()} * \frac{1}{nrtPS_n + BE_n} \right\}$$

$$BE = \min \left\{ BR, BW * \frac{rtPS_latency}{rtPS_latency + DPI\ function()} * \frac{1}{nrtPS_n + BE_n} \right\}$$

由以上的式子我們可以得知，當 virtual loading 大時，DPI function() 的值會比較大，因此

$\frac{DPI\ function()}{rtPS_latency + DPI\ function()}$ 計算出來的值會比較大，所以 rtPS 連線可以取得大部分系

統的頻寬，而當 virtual loading 較小時， $\frac{\text{DPI function}()}{\text{rtPS_latency} + \text{DPI function}()}$ 計算出來的值會比較小，因此多出來的頻寬可以讓給 nrtPS 以及 BE 連線使用。

3.2.4 SS UL Redistributed Scheduling

WiMAX 的標準中，BS BW allocation 的方式如果採用 GPSS(Grant Per SSs)可以有效降低 BS granting loading，因此當 BS 根據各個連線的 BR 封包配置完 BW 之後，會將屬於同一個 SS 的 BW 集合起來，在 UL-MAP 中一次配置總量的 BW 給同一個 SSs，因此 SSs 就有必要根據其底下所有的 traffic queue 的狀況，做一次重新再分配的動作以符合需求，由於 BS BW allocation 已經有考慮 Fairness，因此 SS 在重新再分配的動作上可以採用 BS BW allocation 的方法，但需要做一些修正，我們都知道因為 BS 無法得知 SSs 實際 traffic queue 的狀況，所以只能根據當時的 virtual loading 以及 QoS 的考量來配置 BW，但 SSs 卻很清楚底下 traffic queue 的狀況，因此在 SS 重新再分配的動作上，絕對有必要針對有 deadline 的封包加以處理，以免送出無效的封包，因此 SS redistributed scheduling 變成了三回合，除了第一回合是新增的，其餘二回合和 BS BW Allocation 原理一樣，如下說明：

在 Round 1 階段中會檢查有沒有即將到期的封包，由於只有 rtSP 連線才會有 delay 的考量，因此我們必須檢查 rtPS Queue 是否有 $\text{deadline} - \text{now}() < \text{frame-time-length}$ ，如果有的話先滿足。

在 Round 2 階段中則必須先滿足基本的 Rmin 需求，其順序及數量如下，

$$\begin{aligned} \text{UGS} &= \sum_{ss=1}^n \text{UGS}_n * \text{UGS}_{\min} \\ \text{rtPS} &= \sum_{ss=1}^n \text{rtPS}_n * \text{rtPS}_{\min} \\ \text{nrtPS} &= \sum_{ss=1}^n \text{nrtPS}_n * \text{nrtPS}_{\min} \end{aligned}$$

在 Round 3 階段中則將剩餘的 BW 依比重分配直到 Bandwidth 用完或者所有請求都

已滿足，假設 $rtPS_ratio = \frac{DPI\ function()}{rtPS_latency + DPI\ function()}$ ，

$$nrtPSBE_ratio = \frac{rtPS_latency}{rtPS_latency + DPI\ function()}$$

則分配順序及數量如下規則所示

$$BW_rtPS = \min\left\{ BR - R_{min}, BW * \frac{rtPS_n * rtPS_ratio}{rtPS_n * rtPS_ratio + (nrtPS_n + BE_n) * nrtPSBE_ratio} \right\}$$

$$BW_nrtPS = \min\left\{ BR - R_{min}, BW * \frac{nrtPS_n * nrtPSBE_ratio}{rtPS_n * rtPS_ratio + (nrtPS_n + BE_n) * nrtPSBE_ratio} \right\}$$

$$BW_BE = \min\left\{ BR, BW * \frac{BE_n * nrtPSBE_ratio}{rtPS_n * rtPS_ratio + (nrtPS_n + BE_n) * nrtPSBE_ratio} \right\}$$

3.2.5 Fairness Definition

$$Fairness = \left| \frac{rtPS_throughput}{rtPS_total} - \frac{\left(\frac{nrtPS_throughput}{nrtPS_total} + \frac{BE_throughput}{BE_total} \right)}{2} \right| \quad (f)$$

上式(f)所計算出來的值介於 0~1 之間，以 0 為最公平，1 則為最不公平。