

# 在 WiMAX 網路中設計服務多使用者的排程方法

廖文華

大同大學資訊經營系

Email:whliao@ttu.edu.tw

劉蓁

大同大學資訊經營系

**摘要**—移動式使用者在 IEEE 802.16e 協定中被定義出來，用以支援移動式的網路。為了滿足不同使用者的需求，IEEE 802.16e 定義了五種 QoS 服務類別，其中只有 UGS (Unsolicited Grant Service) 服務類別能夠讓使用者不需要在每次獲得資源分配前，都向基地台提出頻寬要求，也因此減少了頻寬資源的浪費。然而目前的研究雖然能夠針對 UGS 服務類別作有效的排程，但是卻沒有考慮多個使用者的環境，其餘的研究則是考慮了多個使用者的環境，但是 UGS 服務類別的連線排程過後即不再是 UGS 的服務類別，當然也因此失去了它的優勢。本篇提出一個有效的 UGS 排程方法，讓 UGS 的連線在排程後仍然能保有 UGS 的運作方式，並且考慮了多個使用者的環境。

**關鍵詞**—排程、多使用者、WiMAX、IEEE 802.16e、UGS

## 一、簡介

無線網路被認為是比有線網路更加節省成本的，尤其在一些缺乏網路基礎建設的開發中國家，無線網路更是理想的選擇。為了滿足廣大無線網路的需求，WiMAX(worldwide interoperability for microwave access)被 IEEE 802.16 所提出來支援較高的網路效能輸出以及較長的傳輸距離(最遠到 30 哩)。WiMAX 提供了在單一頻道到達 75 Mb/s 以及多頻道到達 350 Mb/s 的傳輸率[2]。為了滿足移動的使用者，IEEE 802.16e 被提出來加強網路的移動性。

基本的 WiMAX 網路架構包含了基地台(Base Station, BS)及用戶端(Subscriber Station, SS)，通常 BS 會服務在它傳輸範圍內的 SS，在 IEEE 802.16e 中，移動用戶端(Mobile Subscriber Station, MSS)被加到網路之中來支援網路的移動

能力[1]，然而 MSS 通常為體積較小的行動裝置(例如：手機、PDA 等)，電力也有所限制，因此省電機制便成為重要的議題。

IEEE 802.16e 定義了五種服務類別，分別是：UGS、rtPS、ertPS、nrtPS、and BE。其中 UGS、rtPS、以及 nrtPS 使屬於即時的服務類別。UGS 的服務類別 BS 會在固定的時間內分配固定的頻寬資源給使用者，因此使用者不需要在獲得頻寬前向 BS 提出頻寬需求(Bandwidth Request)。相對的，rtPS 服務類別每個時間所需的頻寬資源不同，因此需要在獲得頻寬前向 BS 提出頻寬需求，才能獲得頻寬資源。nrtPS 服務類別則是分為 on 以及 off 兩個模式，在 on 模式 ertPS 的運作模式和 UGS 是相同的，但是當模式切換時，ertPS 仍然需要提出需求，並且在 off 模式之下，ertPS 必須保留固定的頻寬，以便要切換回 on 模式時可以利用這些保留的頻寬來提出需求。由此可知，三種即時服務中，只有 UGS 不需要耗費額外的資源來提出頻寬需求，也是 UGS 的優勢之一。

過去已有許多關於在 IEEE 802.16e 網路中排程及改善電力消耗的文獻[4] [5] [6] [7] [8] [9] [10]，我們可以發現有些文獻的方法雖然有考慮到多個 MSS 的環境，但是 MSS 卻沒辦法維持 UGS 規律的運作方式，另一些文獻的方法雖然雖然可以維持 UGS 規律的運作，但是卻沒有考慮到多個 MSS 的環境。

本篇文章的目的要為多個 UGS 服務類別的 MSS 來安排所使用的時槽(slot)，以避免 MSSs 所使用的時槽有重複的情形，並讓 MSS 的運作模式維持 UGS 規律的運作模式，以發揮 UGS 服

務類別的優勢。

本篇文章將在第二章介紹我們提出的方法，第三章是實驗的結果，第四章是本篇論文的結論。

## 二、多使用者之排程方法

在 IEEE 802.16e 網路架構中，基地台(BS)可能同時必須服務多個使用者(MSS)，然而在某一特定的時槽(slot)，BS 只能和某一特定的 MSS 做傳輸[3]，以圖 1 為例，MSS1 及 MSS2 都被同一個 BS 所服務，也就是說，MSS1 及 MSS2 不能夠在同一個 slot 做傳輸，否則就會產生干擾，MSS1 及 MSS2 都各自擁有 UGS 連線，並規律的做傳輸，但是如果我們不做任何處理，只是單純的把兩個 MSS 放在一起，它們所使用的 slot 就會像圖 1 一樣，有重疊的情形，重疊的 slot 在 BS 端會產生干擾以至於兩個 MSS 都無法正常傳輸，因此當我們考慮多個 MSS 的環境時，我們的方法必須避免這樣的情況。

在過去的研究中已經能夠解決如圖 1 所顯示的問題，然而過去的方法在調整之後，就不能維持 UGS 原本規律的運作，兩個 MSS 雖然沒有重複使用的 slot，但是 MSS2 的運作就不再是 UGS 了，因此本篇研究提出一個排程的方法來解決圖 1 的問題，並且維持每條連線 UGS 規律的運作方式。

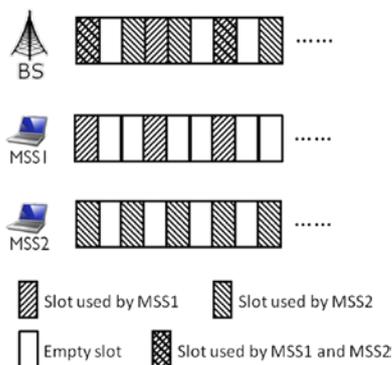


圖 1：多個 UGS 連線的 MSS 之運作情況。

本篇研究的方法流程如圖 2 所示。當我們開始演算法時，我們先將所有需要傳送的連線放入集合  $U$ ，演算法會一直執行直到集合  $U$  中沒有任何連線為止。首先我們會為每一條等待傳輸的連線設定一個權重值，並根據權重值，選擇一條權重值最大的連線做為目前要排程的對象，我們稱它為目標連線，權重值的計算方式在稍後會做詳細的說明。接著我們會檢查排程週期中剩餘的 slot 數是否還足夠安排目標連線，舉例來說，如果目前排程週期中剩餘的 slot 數是 10，但是目標連線在這段時間內需要 12 個 slot 來傳送資料，表示剩餘的 slot 數無法滿足目標連線，如果剩餘的 slot 數不足，我們會將目標連線從集合  $U$  中移除，因為這表示目標連線在這一回合沒有辦法被安排來做傳輸，然而被移除的連線在之後的回合仍然有機會被選來做傳輸，反之，如果剩餘的 slot 足夠，表示目標連線可以在這一回合作傳輸，因此我們會指定目標連線在其週期中被允許傳輸的時間( $G_t$ )、在其週期中間置的時間( $I_t$ )、以及開始時間( $ST_t$ )，但是當我們把目標連線加入排程之後，目標連線可能沒辦法維持 UGS 規律的運作方式，因此我們要檢查目標連線在加入排程後是否還是 UGS 的運作方式，如果不是，目標連線就需要先做一些調整，我們將會調整目標連線的週期以符合 UGS 連線的特性。然而無論目標連線是否經過調整，目標連線在其週期中被允許傳輸的時間( $G_t$ )之中都有可能穿插其他連線使用的 slot，以至於目標連線沒辦法維持 UGS 運作，當這樣的情況發生時，我們就會分割連線，以確保目標連線能維持 UGS 的運作方式。指定目標連線在其週期中被允許傳輸的時間( $G_t$ )、在其週期中間置的時間( $I_t$ )、開始時間( $ST_t$ )、以及分割連線的做法都會在後面的小節做詳細介紹。最後，我們會在實際將目標連線放入排程之前，檢察它的延遲限制是否被滿足，如果延遲限制可以被滿足，我們會將目標連線實際放入排程中，延遲限制檢查的方法也會在後面做討論。當連線排程完

畢之後，會從集合  $U$  之中被移除，並且演算法會在還有剩餘 slot 的情況下繼續執行。當下一回合開始時，集合  $U$  一樣會被指定為所有等待傳輸連線的集合，並再次開始我們的演算法。後面我們將針對演算法中的細節逐步做詳細的介紹。

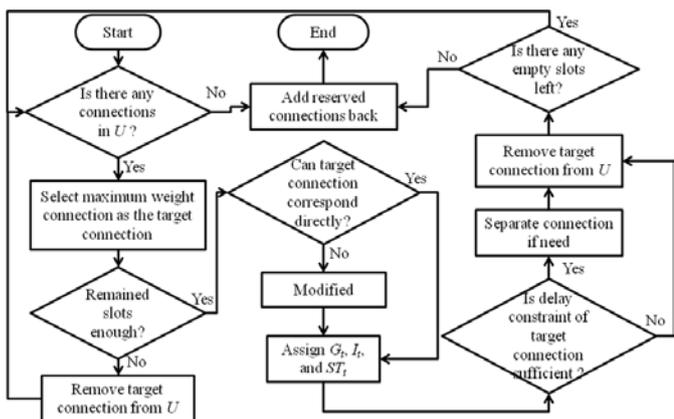


圖 2：本篇研究方法的運作流程。

### (一) 連線權重

當我們決定目標連線時，我們首先會為每一條等待傳輸的連線設定一個權重值，並且選擇權重值最大的連線做為目前要做排程的目標連線。權重值的計算方式如下。

$$W_i = \frac{\omega_i}{\omega_{max}} + \frac{I_i}{d_i} \quad (1)$$

$$i = \arg \max W_i$$

某連線  $i$  的權重值( $W_i$ )會和該連線的等待率( $\omega_i/\omega_{max}$ )、延遲率( $d_i/I_i$ )、以及該連線的循環週期( $C_i$ )相關。我們希望能夠選擇等待較久的連線做為目標連線，因此等待率計算方式中， $\omega_i$  為該連線目前等待的時間， $\omega_{max}$  則是目前網路中等待最久的連線所等待的時間，每當連線在某一回合沒辦法被成功排程時，它的等待時間就會增加，因此等待率越高，表示該條連線在網路中等待越久，也就表示該連線會有較高的權重值，因此會有較高的機率被優先選做為目標連線。延遲率( $I_i/d_i$ )則是該連線在其週期中的閒置時間( $I_i$ )除上

該連線的延遲限制( $d_i$ )，當延遲率越高時，表示該連線的延遲限制較緊迫，也就是當我們要對這條連線做調整時，成功的機會較低，所以我們優先選擇延遲率較高的連線，因為越先被選擇的連線需要做調整的機率越低。連線的週期( $C_i$ )和權重值成反比，這表示我們會優先選擇週期長度較小的連線做排程，這是由於在我們後續介紹的排程方法中，週期較小的連線有較大的機率使得後面的連線不需要做調整。當我們指定所有等待傳輸的連線權重值之後，我們會選擇權重值最大的連線做為目標連線。

### (二) 排程方法

在排程方法的一開始，我們會設定一個排程週期  $T$ ，代表整個排程重複運作週期的長度。在圖 3 中，我們假設連線  $i$  為一條已經被排程的連線，這時候網路中只有一條連線，所以我們的排程週期一定會等於該條連線的週期，因此在圖 3 的例子中， $T$  會被設定為 4，因為連線  $i$  每 4 個 slot 會重複一次週期。接著，我們觀察該條連線和其他連線之間的關係，來決定新的目標連線是否可以被直接排程到網路之中。

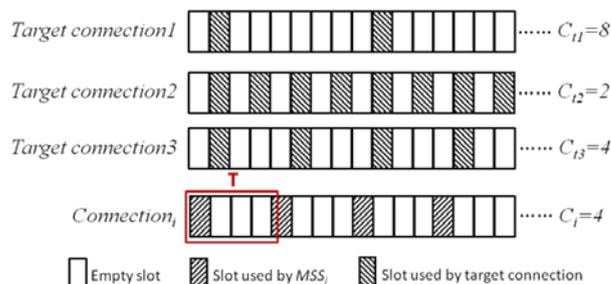


圖 3：多個 MSS 之間的相互關係。

我們觀察連線之間的關係，我們發現當目標連線的週期( $C_i$ )是排程週期  $T$  的倍數或是排程週期  $T$  是目標連線的週期( $C_i$ )的倍數時，目標連線可以直接被排程到網路中，並且不會和已經排好的連線使用重複的 slot。以圖 3 為例，假設連線  $i$  是目前網路中唯一排定的連線，因此排程週期  $T$  會被指定為連線  $i$  的週期，也就是 4，我們設計了三條目標連線的例子來觀察連線間的關係。目標連線 1 的週期為  $T$  的倍數 - 8，當我們把它排程到網路中時，我們發現它不會和已經排定的連

線使用重複的 slot，也就是說我們可以直接將它排進網路中，不需要做任何調整。目標連線 2 的週期為 2，是  $T$  的因數，我們發現當它被排到網路中時，也不需要做任何修改，並且不會和已排定的連線使用相同的 slot。而目標連線 3 擁有和  $T$  相同長度的週期，事實上也就是一倍的  $T$ ，我們發現它也和前兩者一樣，可以直接被排到網路中。因此我們可以得知，當目標連線的週期( $C_i$ )或排程週期  $T$  為對方的倍數時，目標連線就可以在不需要調整的情況下，直接被排程到網路中，同時，我們發現當目標連線排定之後，新的排程週期  $T$  會是目標連線的週期( $C_i$ )和排程週期  $T$  之中較大者，因此我們可以藉由目標連線的週期( $C_i$ )和排程週期  $T$  的最小公倍數來得到新的  $T$  值。所以我們可以藉由下列公式來獲得新的排程週期  $T$ 、目標連線在其週期中被允許傳送的時間( $G_t$ )、目標連線在其週期中的閒置時間( $I_t$ )、以及目標連線的開始時間( $ST_t$ )。

$$\begin{aligned}
 T &= LCM(T, C_i) \\
 G_t &= G_i \\
 I_t &= I_i \\
 ST_t &= find\_ST(t)
 \end{aligned}
 \tag{2}$$

目標連線在其週期中被允許傳送的時間( $G_t$ )及目標連線在其週期中的閒置時間( $I_t$ )都不需要做調整，因此維持本來的值，而連線的開始時間我們在稍後的章節介紹完整的方法來找尋每一條連線開始的時間。

我們進一步檢視當上述狀況不成立時連線間的關係，我們發現連線間可以被分為“Overlap”以及“No overlap”兩種情況。圖 4 顯示了兩個目標連線的週期( $C_i$ )和排程週期( $T$ )皆非對方倍數的例子，在圖 4 第一個例子中，目標連線的週期( $C_i$ )和排程週期( $T$ )雖然皆非對方的倍數，但是當兩條連線被排在一起時，仍然不會使用重複的 slot，也就是“no overlap”的情況，這個情況下我們一樣不需要調整目標連線就可以直接將它排近來，並且同樣可以利用目標連線的週期( $C_i$ )和排程週期  $T$  的最小公倍數來得到新的  $T$  值，因此“no overlap”的情況下我們可以直接

利用公式(2)來獲得新的  $T$ 、 $G_t$ 、 $I_t$ 、以及  $ST_t$ 。圖 4 第二個例子則是“Overlap”的情況，兩條連線被排在一起時，使用了重複的 slot，很顯然的，當這樣的情形發生，我們就必須做一些調整，才能夠避免連線間使用重複的 slot。

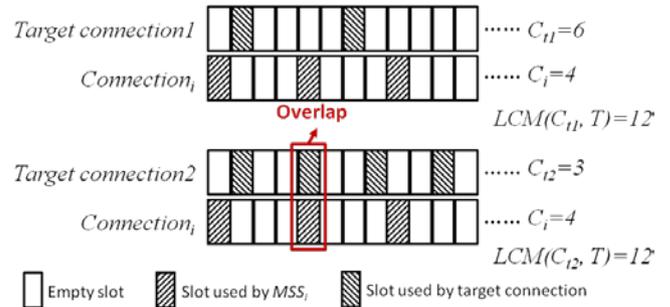


圖 4：多個 MSS 之間重複使用時槽的情況。

當我們檢察連線間是否有重疊的情況時，我們假設  $Z_j$  為目標連線第  $j^{th}$  個使用的 slot 的編號，以圖 4 為例，目標連線 1 的  $Z_1$  為 2 且  $Z_2$  為 7。接著我們針對所有  $j$  以及所有已排定的連線  $i$  計算  $Z_j \bmod C_i$  的餘數是否為零，如果有任何一組  $Z_j \bmod C_i$  為零，代表連線間會有重疊的情形，反之，若所有  $Z_j \bmod C_i$  皆不為零，則表示連線間不會有重疊的情形。

接下來我們就要探討當連線發生重疊的情形時，要如何做調整。在前面的分析中，我們已經得知如果目標連線的週期( $C_i$ )或排程週期( $T$ )為對方的倍數，目標連線就可以在不重疊的情況下排進網路中，因此我們考慮把目標連線的週期調整成和排程週期  $T$  相同，若是兩者相同，我們就可以依照前面分析的結果，把目標連線安排近來，因此我們可以藉由下列公式達到目的。

$$\begin{aligned}
 G_t &= G_i \times \lfloor T / C_i \rfloor \\
 I_t &= T - G_t
 \end{aligned}
 \tag{3}$$

我們利用  $G_i \times \lfloor T / C_i \rfloor$  計算目標連線在排程週期  $T$  的時間內，平均會有多長的傳送時間，並且取下整數，來獲得目標連線新的傳送時間，而既然我們要把目標連線的週期調整成  $T$ ，我們就

可以利用  $T - G_t$  來獲得目標連線新的閒置時間。圖 5 顯示了一個調整目標連線的例子，在圖 5 (a) 中，目標連線以及已排定的連線  $i$  彼此有重複的情形，經由公式(3)的調整，我們可以獲得  $G_t = 1$  以及  $I_t = 3$ ，我們將結果繪於圖 5 (b)，我們發現圖 5 (b) 中目標連線的週期已經被調整為  $T$  了，當然也就可以直接被排進網路中。

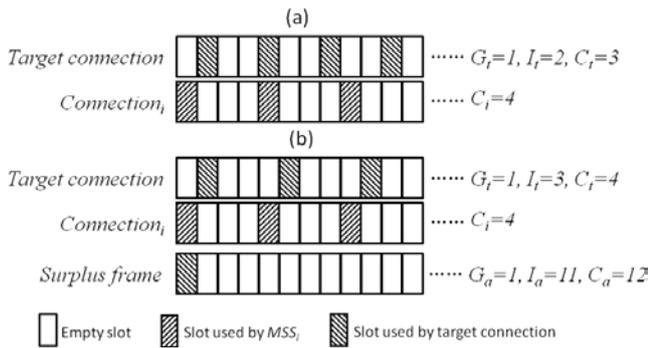


圖 5：調整 MSS 的例子。

但是當我們比較圖 5 (a) 以及圖 5 (b) 的目標連線，可以發現圖 5 (b) 的目標連線比圖 5 (a) 少了一個 slot，因此我們必須要把少的 slot 補回來，才能滿足該使用者原本的需求，因此我們為該使用者新增一條連線，把不足的 slot 利用這條新的連線補回來，我們稱這條連線為 additional connection。我們可以發現，不足的 slot 一定是每  $\text{LCM}(T, C_t)$  個 slot 才會需要一次，以圖 5 (b) 的例子而言，不足的 slot 每 12 個 slot 才會需要補一次，因此我們可以得知 additional connection 的週期一定是  $\text{LCM}(T, C_t)$ ，我們可以藉由下列公式或得 additional connection 的傳輸時間( $G_a$ )、閒置時間( $I_a$ )、以及週期( $C_a$ )。

$$\begin{aligned}
 C_a &= \text{lcm}(C_t, T) \\
 N_n &= G_t \times (C_a / C_t) \\
 N_o &= G_t \times \lceil T / C_t \rceil \times (C_a / T) \\
 G_a &= N_n - N_o \\
 I_a &= C_a - G_a
 \end{aligned} \tag{4}$$

當我們計算出 additional connection 的傳輸時間( $G_a$ )大於 1 時，我們會把 additional connection 分割成  $G_a$  條連線，並且讓每一條連線的傳輸時間( $G_a$ )都等於 1，舉例來說，假設我們得到一條  $G_a = 2$  且  $I_a = 4$  的 additional connection，我們會把這條 additional connection 分成 2 ( $G_a = 2$ ) 條  $G_a = 1$  且  $I_a = 4$  的連線，這樣做的原因會在後面加以說明。

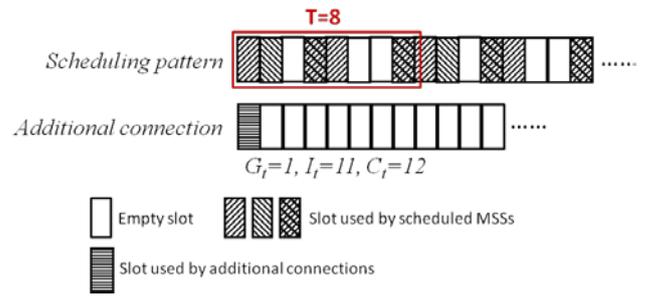


圖 6：Additional connections 運作的例子。

當所有能夠被排程連線都被排進網路之後，我們會把 additional connections 加回網路之中。圖 6 為一 additional connection 的例子，當我們把 additional connection 加回網路中時，我們發現排程週期  $T$  和 additional connection 的週期 ( $C_a$ ) 未必相同，因此當我們選擇某一個閒置的 slot 來讓 additional connection 做傳輸時，該 additional connection 下一次週期所使用的 slot 不一定是  $T$  中相同的位置，以圖 6 的例子而言，如果我們選擇第 3 個 slot 讓 additional connection 做傳輸，則 additional connection 會在 11 個 slot 之後再次需要一個 slot 做傳輸(因為週期  $C_a$  為 12)，因此我們會找到 additional connection 下一次做傳輸的 slot 為第 15 個 slot ( $T$  週期中的第 7 個 slot)，再下一次週期所需要的 slot 則是第 27 個 ( $T$  週期中的第 3 個 slot)，因此我們發現，additional connection 在  $T$  週期中，每  $\text{GCD}(T, C_a)$  個 slot 會需要一次傳輸時間，以圖 6 而言，additional connection 每  $\text{GCD}(8, 12) = 4$  個 slot 就會需要一個 slot，因此當我們選擇第 3 個 slot 讓 additional connection 做傳輸時，我們就必須保留  $T$  週期的第 3 以及第 7 ( $=3+4$ ) 個 slot。我們可以利用  $T / \text{GCD}(T, C_a)$  決定需要保留的 slot 數量。

此外，當有多條週期相同的 additional connection 時，它們可以共享保留的 slot，以達到更有效的頻寬利用。以圖 6 為例，假設我們保留了  $T$  週期的第 3 及第 7 個 slot，當 additional connection 使用第 3 個 slot 做傳輸時，事實上接下來被保留的 2 個 slot – 第 7 及第 11 個 slot，雖然被保留了，卻不會傳送資料(因為 additional connection 會在第 15 個 slot 才傳送第二次週期的資料)，也就是說我們可以安排另外 2 條週期相同的連線來使用這 2 個閒置的 slot，而不需要另外花費其他 slot 來讓這 2 條週期相同的連線做傳輸。因此我們可以利用  $C_a/\text{GCD}(T, C_a)$  得到可以被安排在一起的連線數量，以上述例子而言，我們就可以得到有  $12/\text{GCD}(8, 12)=3$  條週期相同的連線可以被安排在一起。由於這樣做可以善加利用頻寬資源，避免頻寬資源的浪費，因此我們在前面曾經提到若是 additional connection 的傳輸時間( $G_a$ )大於 1 時，我們會把 additional connection 分割成  $G_a$  條連線，並且讓每一條連線的傳輸時間( $G_a$ )都等於 1，就是這個原因。

### (三) 指定連線開始時間

當我們為未調整的連線指定開始時間( $ST_i$ )時，我們找尋第一個閒置的 slot 來做為連線開始的時間，以圖 5 (a) 為例，我們會指定第二個 slot 做為目標連線的開始時間。

首先，第一個閒置的 slot 勢必會大於所有已排定連線傳輸時間( $G_i$ )的總和，因為所有連線都是從第一個閒置的 slot 開始，因此我們可以先將

$$ST_i \leftarrow \left( \sum_{i=1}^n G_i \right) + 1$$

目標連線的開始時間指定為。接下來，我們將會逐一檢查每一個 slot 是否有其他連線使用，當我們找到第一個沒有任何連線使用的 slot，也就是第一個閒置的 slot，演算法就會停止，並指定該 slot 為目標連線開始的時間。為了檢查 slot 是否有其他連線使用，我們利用目標連線目前的開始時間( $ST_i$ )減去目前檢查的連線之開始時間( $ST_j$ )，如果減去後的數值是目前檢查連線週期( $C_j$ )的倍數，表示該 slot 已經被使用了，我們會跳往下一個 slot 繼續做檢查，反之若是某一個 slot 所有已排定的連線都無法構成上述條件，表示該 slot 沒有連線使用，因此我們就可以指定目標連線開始時間為該 slot。完整的演算法

顯示如下，其中  $n$  指的是所有已排定且未被調整連線的數量。

然而當我們為調整後的連線指定開始時間時，為了避免在實際分配頻寬給使用者之後使用者的資料才產生，我們會選擇  $T$  週期的最後一個閒置 slot 來做為開始時間。以圖 5 (b) 為例，目標連線的開始時間將會被指定為第四個 slot，當我們利用  $T$  週期的最後一個閒置 slot 做為開始時間時，我們只要檢查連線的閒置時間是否滿足延遲限制，就可以確保資料在獲得頻寬資源前就已經產生。

我們可以利用公式(5)來為調整過後的連線指定開始時間，其中  $m$  代表所有已排定且調整過後的連線數量。

$$ST_i = T - G_i + 1 - \sum_{i=1}^m G_i \quad (5)$$

### (四) 分割連線

當我們把目標連線排回網路中時，目標連線的傳輸時間內可能會穿插了別的連線使用的 slot，以圖 7 為例，目標連線的傳輸時間為 4 個 slot，然而已經排定的排程所形成的空白區域 (blank area) 為 2，比目標連線的傳輸時間短，因此當我們把目標連線對應回去，目標連線的傳輸時間內就穿插了其他連線使用的 slot，當這樣的情況發生時，也就代表目標連線對應回去之後，將不再是 UGS 規律的運作，當然也就不符合本篇研究的目的。

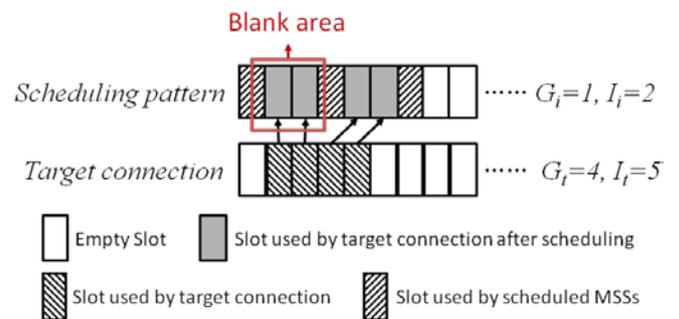


圖 7：連線被允許傳輸的時間超過 blank area 的例子。

事實上無論連線是否被調整，這樣的狀況都有可能發生，即使一開始我們判定目標連線可以在沒有重疊的情況下排進網路之中，也有可能因為傳輸時間過長而發生這樣的情況，因此無論目標連線先前是否做過調整，都必須要針對這個問題做出處理。我們處理的方式為分割連線，即是把目標連先分割成多條連線，以確保每一條連線都可以維持 UGS 規律的運作。

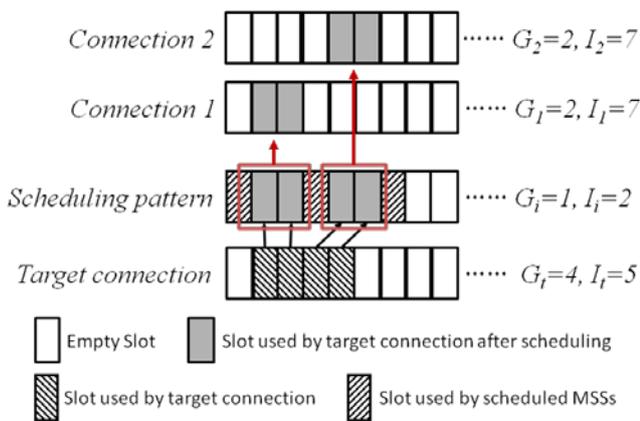


圖 8：分割連線。

圖 8 為一分割連線的例子，目標連線的傳輸時間比排程所形成的空白區域長，因此當目標連線對應回排程時，目標連線使用的第 2 個及第 3 個 slot 間就穿插了其他連線使用的 slot。首先我們要檢查是否會有穿插的情形出現，我們逐一檢查目標連線週期內所有使用的 slot，檢查是否有其他連線使用同一個 slot，若有任何已經排定的連線使用該 slot，表示目標連線在該 slot 會被其他連線穿插，檢查的方式類似前面尋找開始時間時使用的方法，因此在這邊就不再贅述。當我們找到穿插的 slot 之後，我們會把穿插 slot 左右分為兩條連線，並指定左邊的連線新的  $G_i, I_i$  及  $ST_i$ ，然後把右邊的連線設定為目標連線，繼續針對目標連線做檢查，直到所有目標連線的傳輸時間都檢查完為止。以圖 8 為例，目標連線會在第 2 及第 3 個使用 slot 間被分為 2 條連線，當我們分別檢視被分出來的連線一及連線二，我們發現新的兩條連線都可以維持 UGS 規律的運作。

穿插 slot 左邊連線的傳輸時間( $G_o$ )會是目前

檢查的 slot 數  $j$  減去 1，閒置時間( $I_o$ )則是目標連線的週期減去剛剛計算出來的新傳輸時間( $G_o$ )，開始時間( $ST_o$ )會維持本來目標連線的開始時間( $ST_t$ )，以圖 8 為例，連線一為穿插 slot 左邊的連線，當我們檢察到第 3 個 slot 時發現了穿插，因此連線一的傳輸時間( $G_o$ )為  $3-1=2$ ，閒置時間( $I_o$ )則是目標連線的週期 9 減去剛剛計算出的傳輸時間 2 得到 7。而穿插 slot 右邊的連線會被指定成新的目標連線，傳輸時間( $G_t$ )會是原本目標連線的傳輸時間減去剛剛計算出左邊連線的傳輸時間( $G_o$ )，閒置時間( $I_t$ )則是原本目標連線的閒置時間加上新計算出的傳輸時間( $G_o$ )，以圖 8 為例，連線二為新的目標連線，它的傳輸時間為原本目標連線的傳輸時間 4 減去連線一的傳輸時間 2 得到 2，閒置時間則是原本目標連線的閒置時間 5 加上連線一的傳輸時間 2 得到 7，接著我們使用前面尋找開始時間的方法，為新的目標連線找到適當的開始時間。本方法會執行到所有目標連線使用的 slot 都檢查完畢，沒有任何穿插 slot 為止。

### (五) 延遲限制

在我們前面的方法中，連線的傳輸時間及閒置時間都可能會被調整，因此我們必須在實際將連線排進排程之前，檢察該連線的延遲限制是否被滿足。

在 UGS 連線中，兩次傳輸時間之間形成的間隔時間必須小於連線的延遲限制，而在我們的方法中，閒置時間即是兩次傳輸時間之間所形成的間隔時間，因此我們只要檢查連線的閒置時間是否小於延遲限制即可。

若是閒置時間小於延遲限制，表示連線的延遲限制能夠被滿足，該連線就可以順利被排進排程中；若是閒置時間大於延遲限制，表示連線的延遲限制沒辦法被滿足，當延遲限制不能被滿足，我們會在這一回合先放棄這條連線，不過如同前面所提到的被放棄的連線在後面的回合會有較大的機率被優先選擇。

## 三、實驗結果

在本章中，我們將利用模擬實驗的結果，來

證實本研究方法的成果。

模擬實驗中我們計算頻寬使用率、連線選擇率、以及為了調整連線所犧牲的頻寬浪費率。頻寬使用率指的是網路中有多少比例的頻寬資源是被有效使用的，連線使用率是指所有等待傳輸的連線有多少比例被選擇做傳輸，頻寬浪費率則是指我們為了讓連線可以維持 UGS 規律的運作而調整連線，產生 additional connection 時，浪費了多少比例的頻寬資源，舉例來說，若是我們保留了 2 個 slot 給 additional connection，但是實際上這 2 個被保留的 slot 可以讓 3 條週期相同的連線做傳輸，這時候如果我們只有一條連線，就一定會因此而造成頻寬的浪費，我們可以藉由： $1 - (\text{保留 slot 數}) * (\text{已排定的連線數量}) / (\text{能夠同時使用保留 slot 的連線數量})$  來計算頻寬浪費率。實驗相關之參數設定列於表 1。

表 1：實驗參數。

參數名稱	參數值
傳輸時間	1-5
閒置時間	1-25
連線數量	1-15
延遲限制	2 * 閒置時間

圖 9、圖 10、及圖 11 顯示了實驗的結果，在實驗裡，連線數量從 1 至 15 做調整，傳輸時間同樣從 1 至 5 單位隨機選取，閒置時間則是分別從 1 至 5 單位、6 至 10 單位、11 至 15 單位、16 至 20 單位、21 至 25 單位，五個區間內隨機選取，可以從閒置時間的區間長短看出傳輸時間跟閒置時間相互的關係。

在圖 9 中，顯示頻寬使用率隨機實驗的結果，頻寬使用率隨著連線增加而增加，當連線數量超過 10 之後，頻寬使用率就已經非常接近 90%，甚至在連線數量更多時超過 90%，而當閒置時間的區間越長時，頻寬使用率會在連線數量較多的情況下越高，這是由於當閒置時間的區間越長時，我們的方法調節能力較佳，因而頻寬的浪費較少的緣故。

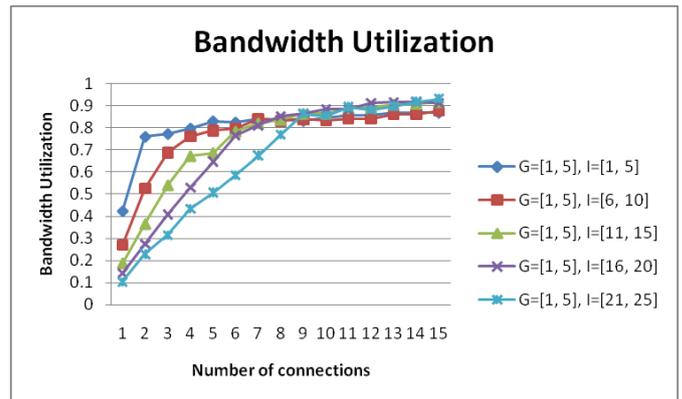


圖 9：頻寬使用率。

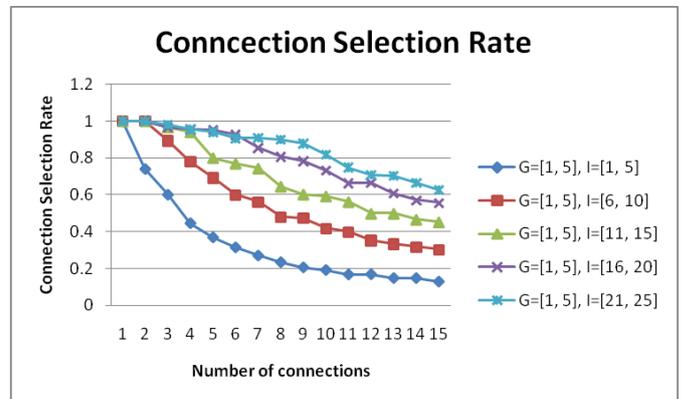


圖 10：連線選擇率。

圖 10 顯示連線選擇率隨機實驗的結果，連線選擇率在連線數量 1 時為 100%，因為只有一條連線可供選擇，此時該條連線一定會被選中，當連線數量增加，連線選擇率就下降了，而閒置時間的區間越長，表示資料量越少，所以連線選擇率就會比較高，下降的速度也比較慢。

頻寬浪費率隨機實驗的結果顯示於圖 11，頻寬浪費率為調整連線時，為了滿足 additional connection，而造成的頻寬資源浪費。當連線數量增加時，頻寬浪費率會上升，但是當連線數量更多時，頻寬浪費反而會下降，這是因為一開始連線數量增加時，additional connection 遇到週期相同的連線機率還不大，但是出現 additional connection 的機率卻上升，所以頻寬浪費率也就上升了，但是隨著連線數量變得更多時，雖然出現 additional connection 的機率也會上升，但是 additional connection 遇到相同週期連線的機率卻

更高了，所以也就有更多機會將週期相同的連線排在一起，所以浪費的頻寬反而減少了，這個結果也證明了我們的方法在連線數量很多的環境中，頻寬的使用非但不會下降，反而上升。從圖 11 中還可以看出當閒置週期的區間越長，浪費的頻寬會越少，這可以呼應圖 11 的結果，在圖 11 中，閒置週期的區間越長，在連線數量較多時，頻寬利用率越高，這就是因為頻寬浪費率較少的緣故。

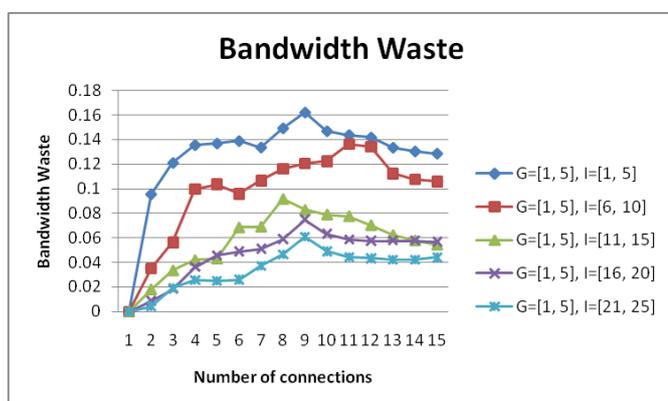


圖 11：調整連線造成的頻寬浪費率。

#### 四、結論

在本研究中，我們針對 UGS 服務類別提出一個排程的方法，並能夠有效的運用在多個 MSS 的環境，避免同一基地台服務的 MSS 同時做傳輸，同時能夠維持 UGS 規律的運作。實驗結果顯示我們的方法可以在連線較多時達到超過 90% 的頻寬使用率，並且盡量選擇較多的連線來運作，以達到較高的連線選擇率，同時隨著連線數量增加，減少頻寬的浪費，這顯示了我們的方法極具彈性，能夠應用在很多連線的環境下而不影響排程效能。未來可以針對如何把本研究之方法和實際狀況結合，以期能更有效的被運用在現實環境中。

#### 致謝

本研究承國科會與大同大學補助，計畫編號：NSC98-2221-E-036-032 及大同大學 B98-N01-033，特此致謝。

#### 參考文獻

- [1] IEEE Standard 802.16e-2005, "IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access System," *Proceedings of IEEE Std.*, Feb. 2006.
- [2] Bo Li, Yang Qin, Chor Ping Low, and Choon Lim Gwee, "A Survey on Mobile WiMAX," *IEEE Communications Magazine*, Vol. 45, No. 12, pp. 70-75, December 2007.
- [3] Chakchai So-In, Raj Jain, and Abdel-Karim Tamimi, "Scheduling in IEEE 802.16e Mobile WiMAX Networks : Key Issues and a Survey," *IEEE Journal on Selected Areas in Communications*, Vol. 27, No. 2, pp. 156-171, February 2009.
- [4] Tuan-Che Chen, Ying-Yu Chen, and Jyh-Chen Chen, "An Efficient Energy Saving Mechanism for IEEE 802.16e Wireless MANs," *IEEE Transactions on Wireless Communications*, Vol. 7, No. 10, pp. 3708-3712, October 2008.
- [5] Tuan-Che Chen, Jyh-Chen Chen, and Ying-Yu Chen, "Maximizing Unavailability Interval for Energy Saving in IEEE 802.16e Wireless MANs," *IEEE Transactions on Mobile Computing*, Vol. 8, No. 4, pp. 475-487, April 2009.
- [6] Jenhui Chen, Yu-Lin Li, and Hua-Chiang Yin, "Adaptive Traffic Indication Algorithm for Energy Efficiency in IEEE 802.16e Systems," *Proceedings of Communications of the ACM*, February 2008.
- [7] Chia-Yen Lin, His-Lu Chao, Yang-Jang Liao, and Tzu-Jane Tsai, "Least-Awake-Slot Scheduling with Delay Guarantee for IEEE 802.16e Broadband Wireless Access Networks," *Proceedings of IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications*, 2008.
- [8] G. Fang, E. Dutkiewicz, Y. Sun, J. Zhou, J. Shi, and Z. Li, "Improving mobile station energy efficiency in IEEE 802.16e WMAN by burst scheduling," *In proceeding of IEEE GLOBECOM 2006, WLC 17-5*, December 2006.
- [9] S. C. Huang, R. H. Jan, and C. Chen, "Energy efficient scheduling with QoS guarantee for IEEE 802.16e broadband wireless access networks," *In proceeding of International Wireless Communications and Mobile Computing Conference (IWCMC 2007)*, pp. 547-552, August 2007.
- [10] Shih-Chang Huang, Chien Chen, Rong-Hong Jan, and Cheng-Chung Hsieh, "An Energy-Efficient Scheduling for Multiple MSSs in IEEE 802.16e Broadband Wireless," *Proceedings of IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications*, 2008.