

國立政治大學資訊科學系  
Department of Computer Science  
National Chengchi University

碩士論文

Master's Thesis

LTE下行鏈路中具調適服務品質及公平性考量

之排程研究

Adaptive QoS and Fairness Consideration for  
Downlink Scheduling in LTE

研究生：胡建彪

指導教授：張宏慶

中華民國 一 百 零 一 年 七 月

July 2012

LTE下行鏈路中具調適服務品質及公平性考量  
之排程研究

Adaptive QoS and Fairness Consideration for  
Downlink Scheduling in LTE

研究生：胡建彪

Student: Chien-Piao Hu

指導教授：張宏慶

Advisor: Hung-Chin Jang



國立政治大學

資訊科學系

碩士論文

A Thesis

Submitted to Department of Computer Science

National Chengchi University

in partial fulfillment of the requirements

for the degree of

Master

in

Computer Science

中華民國 一百零一年 七月

July 2012

## 致謝

時光飛逝，歲月如梭，碩士生涯轉眼間即將劃下句點，迎面而來的，是人生的另一個挑戰。在研究所的期間，首要感謝我的指導教授張宏慶博士，在學術上傳道授業，指導我完成這篇論文；生活上也充分地給予自由，發展額外的興趣，培養健全的人格發展。即使研究中遭遇到許多瓶頸，教授也從旁協助，給予學生正面的思考，讓學生謹記在心。

感謝實驗室中一起研究的夥伴，國淵、立吉、偉良、孜鈞給予專業上的協助，和生活上的娛樂；佩璇，鐘毅，偉迪，柏硯，華元，博為在實驗室中互相打氣，教學相長。此外，特別感謝昀峻，常常與我討論至深夜，遇到瓶頸時互相加油，一同找尋解決方案。

最後，感謝我的家人一直在旁支持，給予鼓勵，讓我無後顧之憂。感謝曾經幫助我的人，鼓勵我的人，讓我在面對困境時，愈挫愈勇。謝謝你們。



# LTE 下行鏈路中具調適服務品質及公平性考量之排程研究

## 中文摘要

隨著全球通訊技術的發展，第四代行動通訊系統(4G)已進入我們的生活之中。其中又以長期演進技術 (Long Term Evolution, LTE) 為代表。LTE 使用了正交分頻多工(Orthogonal Frequency-Division Multiplexing, OFDM)以及多輸入多輸出(Multi-Input Multi-Output, MIMO)技術，使得第四代行動網路在上傳以及下載時，能擁有更大的傳輸量，及更遠的傳輸距離。

當下載時，因為每位使用者所在的環境不同，造成傳輸量、延遲時間因而不同，所以基地台分配資源時，尚有許多改善的空間。目前，許多文獻在探討如何公平且有效地分配資源塊(Resource Block, RB)給使用者裝置(User Equipment, UE)，如 Proportional Fair (PF)與 Modified Largest Weighted Delay First (MLWDF)兩種排程機制。前者考量了優先權的公平性，但沒有考量服務品質(Quality of Service, QoS)；後者對於較即時的封包有較高的優先權，卻導致資源分配不均，將大量的資源給部份使用者。本篇論文著重於公平性以及服務品質的考量，使用不同方式的佇列存放各種不同的資源型態(resource type)，依照資源型態佔整體資源數量的比例，以及計算出 Adaptive Modified Largest Weighted Delay First (AMLWDF)的優先權值，針對各個 UE 與各個通道(channel)做全面性地配置，來提昇公平性及降低延遲以符合 QoS 要求。

# **Adaptive QoS and Fairness Consideration for Downlink**

## **Scheduling in LTE**

### **Abstract**

Owing to the development of global communication technology, the Long Term Evolution (LTE) is the latest technology for the fourth generation mobile communication systems (4G) that has entered into our lives. LTE uses Orthogonal Frequency Division Multiplexing (OFDM) and Multiple-Input Multiple-Output (MIMO) technology to provide high data rate transmission and long distance transmission when users doing download and upload.

When doing download, users may have different throughput, delay time, and jitter due to they are in different locations. To improve these performance indexes, the E-UTRAN Node B (eNodeB) has to allocate resource blocks efficiently.

In the literature, many works explore how to fairly allocate resource blocks (RB) to users. Proportional Fair (PF) and Modified Largest Weighted Delay First (MLWDF) are two example scheduling mechanisms. PF considers service priority and fairness, but doesn't consider the Quality of Service (QoS). MLWDF considers QoS but not service priority and fairness, and allows eNodeB giving more resources to particular users.

In this thesis, we focus on resource allocation problem of downlink scheduling in LTE. Considering fairness and QoS, we store various resource types into particular queues and calculate the priorities using Adaptive Modified Largest Weighted Delay First (AMLWDF). The relationships between users and channels are coordinated according to the priorities for the sake of enhancing the fairness and reducing the delay time and jitter.

# 目錄

第一章 簡介	1
1.1 背景	1
1.1.1 LTE 簡介	1
1.1.2 OFDM 基本架構	2
1.1.3 MIMO 基本架構	3
1.1.4 OFDMA 基本架構	3
1.1.5 LTE 分層架構	3
1.1.6 訊框格式與資源塊格式	5
1.2 研究動機	7
1.3 論文架構	8
第二章 相關研究	9
2.1 通道品質指標	9
2.1.1 CQI	9
2.2 排程演算法	10
2.2.1 Best CQI	11
2.2.2 Round Robin	13
2.2.3 Proportional Fair	14
2.2.4 Modified Largest Weighted Delay First (MLWDF)	15
2.2.5 Hungarian	15
2.3 服務品質 (QoS)	20
2.3.1 標準 QCI 屬性	20
2.4 總結	22
第三章 研究方法	23
3.1 問題分析	23
3.1.1 網路服務品質的考量	23
3.1.2 公平性及優先權的考量	23
3.1.3 避免佇列中的資料長期等待	24
3.1.4 時間複雜度的考量	24
3.2 研究方法	25
3.2.1 允入控制連線數之評估與調適	25
3.2.3 QoS 下 AMLWDF 之計算	25
3.2.4 AMLWDF 動態資源塊之分配	26
3.3 排程演算法之架構與流程	27

第四章 模擬實驗與結果分析 .....	39
4.1 實驗環境與假設 .....	39
4.1.1 評估指標 .....	39
4.1.2 模擬環境 .....	40
4.2 實驗數據與分析 .....	41
4.2.1 實驗一 .....	41
4.2.2 實驗二 .....	45
第五章 結論與未來研究 .....	49
5.1 結論 .....	49
5.2 未來研究 .....	50
參考文獻 .....	51



## 圖目錄

圖 1 - 1 LTE Layer2 架構圖.....	4
圖 1 - 2 Frame Structure Type 1 格式圖[4].....	6
圖 1 - 3 Frame Structure Type 2 格式圖[4].....	7
圖 2 - 1 SINR 與 CQI 之對應圖[5].....	10
圖 2 - 2 Best CQI 之示意圖.....	13
圖 2 - 3 Round Robin 之示意圖.....	14
圖 2 - 4 Hungarian 演算法步驟圖 I .....	16
圖 2 - 5 Hungarian 演算法步驟圖 II .....	16
圖 2 - 6 Hungarian 演算法步驟圖 III.....	17
圖 2 - 7 Hungarian 演算法步驟圖 IV.....	17
圖 2 - 8 Hungarian 演算法步驟圖 V.....	17
圖 2 - 9 Hungarian 演算法步驟圖 VI.....	18
圖 2 - 10 Hungarian 演算法步驟圖 VII.....	18
圖 2 - 11 Hungarian 演算法步驟圖 VIII-1.....	19
圖 2 - 12 Hungarian 演算法步驟圖 VIII-2.....	19
圖 2 - 13 Hungarian 演算法步驟圖 IX.....	19
圖 2 - 14 Hungarian 演算法步驟圖 X.....	20
圖 3 - 1 分類資源型態示意圖.....	29
圖 3 - 2 AMLWDF 矩陣圖.....	31
圖 3 - 3 保留區間圖.....	31
圖 3 - 4 UE 之 AMLWDF 優先權值計算圖.....	32
圖 3 - 5 GBR 與 non-GBR 之 Greedy 演算法步驟圖 I .....	33
圖 3 - 6 GBR 與 non-GBR 之 Greedy 演算法步驟圖 II .....	34
圖 3 - 7 GBR 與 non-GBR 之 Greedy 演算法步驟圖 III.....	35
圖 3 - 8 IMS 之 Greedy 演算法步驟圖.....	36
圖 3 - 9 AMLWDF 配置完成圖.....	37
圖 3 - 10 AMLWDF 之流程圖.....	38
圖 4 - 1 Jain' Fairness Index.....	42
圖 4 - 2 Average Throughput 與 UE Number 關係圖.....	43
圖 4 - 3 Average Throughput of GBR 與 UE Number 關係圖.....	43
圖 4 - 4 Average Throughput of Non-GBR 與 UE Number 關係圖...44	
圖 4 - 5 Average Delay Time 與 UE Number 關係圖.....	45
圖 4 - 6 Average Delay Time of GBR 與 UE Number 關係圖.....	45



圖 4 - 7 Average Delay Time of Non-GBR 與 UE Number 關係圖...46  
圖 4 - 8 Average Jitter 與 UE Number 關係圖.....47  
圖 4 - 9 Average Jitter of GBR 與 UE Number 關係圖.....47  
圖 4 - 10 Average Jitter of Non-GBR 與 UE Number 關係圖.....48



## 表目錄

表 2 - 1 4-Bit CQI Table[6].....	10
表 2 - 2 LTE QCI 各項需求參數[11].....	21
表 2 - 3 各排程演算法之比較表.....	22
表 3 - 1 頻帶與資源塊之對應表[12].....	27
表 4 - 1 模擬環境參數.....	41
表 4 - 2 AMLWDF 之公平性改進比例.....	42
表 4 - 3 AMLWDF 之產能改進比例.....	44
表 4 - 4 AMLWDF 之延遲時間改進比例.....	46
表 4 - 5 AMLWDF 之抖動率改進比例.....	48



# 第一章 簡介

## 1.1 背景

### 1.1.1 LTE 簡介

行動電話的發展，最早為美國警車上的單向廣播功能。後來二次世界大戰，許多行動電話技術被應用在軍事上。後來貝爾實驗室又提出了蜂巢觀念，每一個基地台以較低的功率提供較小範圍的服務。如此，其它較遠的基地台即可使用相同的頻段。回頭看行動電話的發展史，可以細分為一代、二代、三代行動電話網路。

第一代行動電話系統 (The First Generation Mobile System, 1G)，使用類比技術來傳遞訊息。而第二代行動電話系統 (The Second Generation Mobile System, 2G)，將類比訊息改用數位的方式傳遞，以保有較佳的傳送品質。後來隨著網際網路的蓬勃發展，為了也能利用無線電來傳遞封包資訊，所以發展出分封交換技術 (packet-switched) 來傳遞封包資料，稱為 GPRS (General Packet Radio Service)，也就是俗稱的 2.5 G。但為了滿足更大的需求，國際電信聯盟 (International Telecommunication Union, ITU) 希望各國能有統一的規格。以至於，第三代行動電話系統 (The Third Generation Mobile System, 3G) 應運而生。如今 3G 技術已趨成熟，4G 的定義也在 2003 年出爐：在高速行動下 (300km/h) 必須能達到 100Mbps 的傳輸速率，而靜止時能達到 1Gbps 的傳輸速率，且使用 All-IP 技術的核心網路。現今，LTE、WiMAX、UMB 等技術，常被泛稱為 4G。

此外，隨著全球通訊技術的快速發展，3GPP 組織為了能與 WiMAX 進行競爭，發起了通用移動通信系統 (UMTS) 的長期演進 (Long Term Evolution, LTE)。他與 3GPP2 所發起的超行動寬頻 (Ultra Mobile Broadband, UMB)，合稱「3G 演進技術」。而為了實現更高的傳輸速率，3GPP 的 LTE 和 3GPP2 的 UMB 都選擇了

使用 OFDM、以及多天線 MIMO 技術來當作空中接口的無線傳輸技術。目前 3GPP 的 LTE、WiMAX、IEEE 802.20 移動寬帶無線接入，都已經具備了 4G 的相關特性，所以可以將這些技術看成準 4G。

LTE (Long Term Evolution) 長期演進技術，成為近兩年 3GPP 最新的技術項目。其主要特色是在 20MHz 頻帶下，仍然能提供 100Mbps 下載、50Mbps 上傳的速率，改善小區邊緣用戶的性能，提高小區容量，降低系統延遲。而最大的特色就是 eNodeB 的演進，將傳統的 NodeB 作變更，使得網路能夠扁平化，以減少延遲。

在傳統的分頻多工 (Frequency-Division Duplexing, FDD) 下，一個使用者會佔據大部份頻寬，然而 LTE 使用了正交分頻技術 (OFDM)，將需要傳輸的資料分成許多子載波，可以傳遞更多資料。這些子載波，彼此是正交的，所以當接收者接收到資料時，並不會讀到別人的資料。再加上一大段的頻段，分給眾多的子載波，再利用反快速傅立葉轉換，以及分時多工 (Time-Division Duplex, TDD) 的方式，可以使眾多接收者在同一時間，接收到資料。這就是目前 LTE 所使用的下載技術 (OFDMA)，而上傳則是使用 SC-FDMA，利用反快速傅立葉以及快速傅立葉轉換將許多子載波又合成單一載波，來提供上傳。而多天線技術 MIMO 能使接收端同時收到更多筆資料，加大其傳遞的資料量。

### 1.1.2 OFDM 基本架構

傳統的無線行動網路，會將頻帶分成好幾個區域，每一個小區域給不同的人使用，其中利用輪流或跳躍的方式，使所有的使用者可以共享頻帶。但當頻帶的量不夠大時，使用人數便會受到限制。於是乎正交分頻多工 (Orthogonal Frequency-Division Multiplexing, OFDM) 之技術應運而生。

在 OFDM 中，採用反快速傅立葉轉換，將眾多可用的頻帶分成正交的子載波。而頻帶的重建是由快速傅立葉反轉換完成的。在頻寬方面，5、10 和 20MHz 是 LTE

支援的頻寬。該技術的一個優異特性是可以容易地改用其他頻寬。即便整個可用頻寬改變了，較小的頻寬單元也可維持不變。也就是大的頻帶可以再細分成更小的頻帶，通常為 12kHz，而這小頻帶則被稱為子載波。

### 1.1.3 MIMO 基本架構

多輸入多輸出 (Multi-input Multi-output, MIMO)，是 LTE 的另一項優勢。MIMO 採用了多天線的處理技術。多天線技術中，包含了「空間多工 (Spatial Multiplexing)」、「空間多樣 (Spatial Diversity)」、「波束成型 (Beamforming)」以及「預先編碼 (Precoding)」等領域。通常多天線技術會與正交分頻多工技術一起運作，以增加資料的產能及傳送距離，以及降低位元錯誤率。

### 1.1.4 OFDMA 基本架構

OFDMA (Orthogonal Frequency Division Multiple Access) 為 LTE 下行鏈路存取資料的方式。OFDMA 將頻帶劃分成許多子載波，且因為子載波的重疊佔用了頻譜，OFDMA 能提供較高的頻譜利用率，以及傳遞較多的資料量。使用者可以選擇較好的通道進行資料的傳輸，且與 CDMA 單一載波所承載的單一資料相比，OFDMA 更能對抗多路徑效應。

OFDMA 中的子載波可以被組合成更大通道，這些通道能被組合成可分配給使用者的大量單元。每一個大量單元的分配可在訊框間及該調變等級內改變，允許基地台根據目前的系統需求動態調整頻寬。

此外，每位使用者只佔用整體頻寬的一小部份，故可以調節每位使用者的功率。基於不同使用者的特殊需求，如即時語音，服務品質的考量也是一大重點。

### 1.1.5 LTE 分層架構

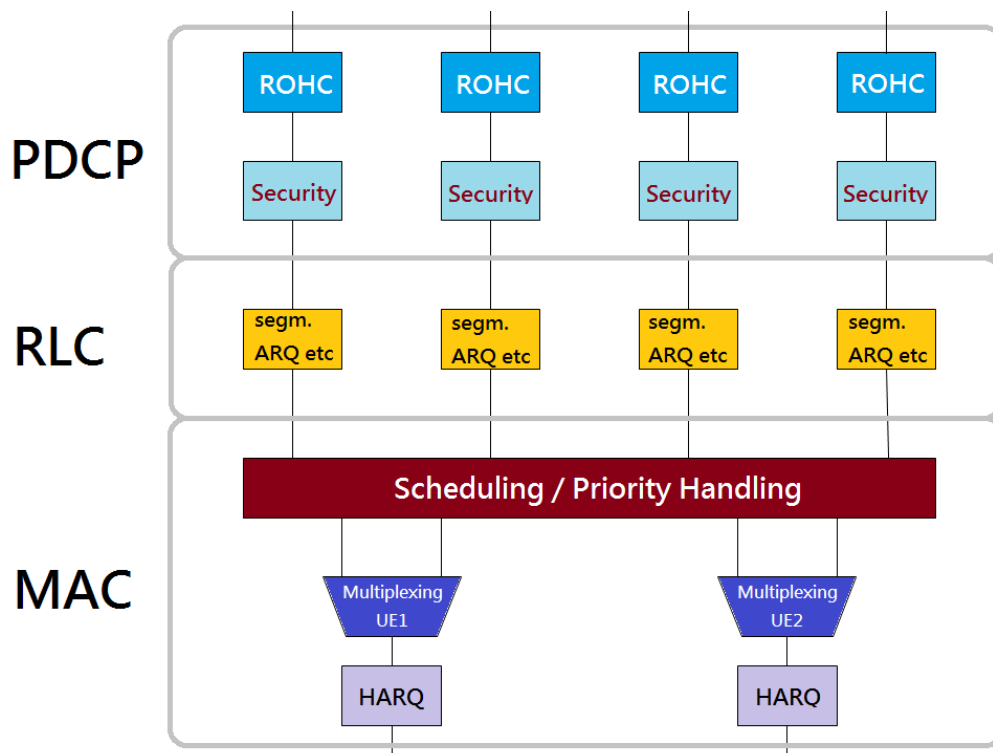


圖 1 - 1 LTE Layer2 架構圖

### 【MAC 層】

在[1]中，LTE 當中的 Layer 2，又分成三層，其中最底層負責與實體層聯繫的為媒體存取控制（Medium Access Control, MAC）層。它包含了邏輯通道與傳輸通道的對應、排程管理以及多工與解多工之任務。當資料傳送至 MAC 子層後，MAC 子層會執行排程以及多工等步驟來組成 MAC 子層的 PDU。但為了防止 PDU 在傳送資料時遺失，因此還提供了「混合式自動回覆請求重傳」（HARQ）功能。

此外，當不同邏輯通道的資料，組合在同一個傳輸通道傳送時，MAC 子層會將得到的 SDU 封裝成同一個 PDU。反之，子層也會將 PDU 進行解多工，拆成各通道所需的 SDU。MAC 的 PDU 包含 header 與 payload 兩部份。Header 的功能為描述 payload 中的內容資訊。而 Payload 則包含了控制元素以及 MAC SDU。當有多個通道欲傳送資料時，就必須為 PDU 排程。在排程中，會加入一個 Bj 的參數作為輔助機制，其目的是避免較低優先權通道產生飢餓現象。

## 【RLC 層】

在[2]中，當資料進入無線電連結控制（Radio Link Control, RLC）層後，其 SDU 會被切成適當大小的 PDU，當封包如果遺失時，則 RLC 會發送自動重傳請求（Automatic Repeat Request, ARQ）。在 Layer 2 中，RLC 扮演協調的角色，將上層資料及下層資料互傳，且利用 buffer 暫存後再做進一步動作。

PDU 格式包含了 header 及 data 兩部份，且含有序號，可讓傳送端知道哪些 PDU 需要重傳、重新排序、重複接收等。ARQ 須視 RLC 子層的操作模式而定，可分為三種模式：透明模式（transparent mode）、承認模式（acknowledged mode）、非承認模式（unacknowledged mode）。ARQ 機制的功用是克服 PDU 遺失的問題，所以傳送端必需將 PDU 存在 buffer 中，以進行 PDU 的重傳。

## 【PDCP 層】

在[3]中，當資料從上層進入後，首當其衝的就是封包資料匯聚協定（Packet Data Convergence Protocol, PDCP）層，此層會將接收的資料封裝，再傳遞給下層。PDCP 層最大的功用，就是替 header 作壓縮以及解壓縮的動作。由於愈來愈多的無線連結技術採用以網際網路通訊協定為基礎的封包資料架構來傳送資料，因此 IP header 的 overhead 也愈來愈嚴重。通常 Header 的容量大約是 Data 的二到三倍，故需要針對 header 作壓縮，才能有效利用資源。此外，無線的連結也較容易遺失封包，且有較長的 round-trip time，因此需要強固的封包 header 壓縮方法，來提升使用率，在 PDCP 層常用的壓縮技術為強固檔頭壓縮（Robust Header Compression, ROHC）技術。

### 1.1.6 訊框格式與資源塊格式

目前 LTE 包含了兩種訊框格式[4]，因為不同的系統而有不同的結構。Frame Structure Type 1 稱為通用（Generic）訊框結構，而 Frame Structure Type 2



則稱為可選 (Alternative ) 訊框結構。第一型訊框格式應用在 FDD 與 TDD 模式下，而第二型訊框格式則只應用在 TDD 模式下。

### 【Frame Structure Type 1】

第一型的訊框格式，如下圖所示。每一個訊框 (frame)，佔用了 10 毫秒。一個訊框包含 10 個子訊框 (subframe)，各佔 1 毫秒。而每個子訊框包含兩個時間間隙 time slot (或稱時槽)。每一個 time slot 佔用 0.5 毫秒。所以在第一型訊框格式下，一個訊框共包含 20 個 time slot。

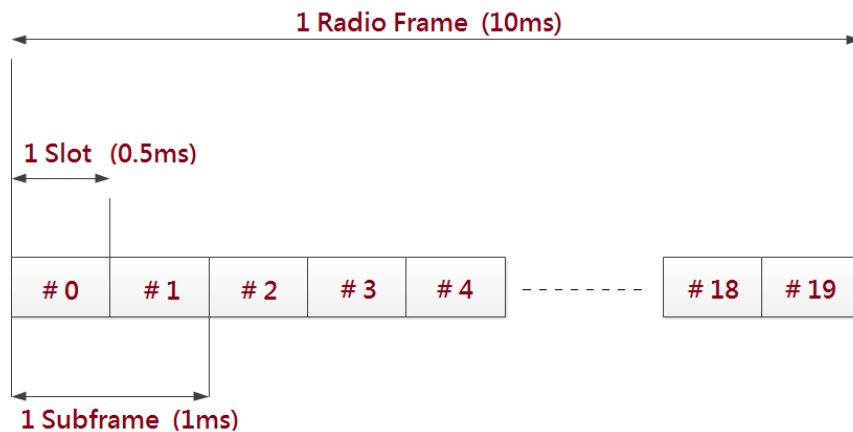


圖 1 - 2 Frame Structure Type 1 格式圖[4]

### 【Frame Structure Type 2】

第二型的訊框格式，如下圖所示。與第一型一樣，每一個訊框佔用 10 毫秒，然而一個訊框只包含兩個半訊框 (half-frame)，每一個半訊框佔用 5 毫秒。而每一個半訊框共有七個子訊框 (subframe)，從 #01 至 #06。每一個子訊框佔用 0.675 毫秒。同步與保護週期在 #0 與 #1 之間，下行鏈路同步時間間隙 (Downlink



Pilot Time Slot, DwPTS)、保護周期 (Guard Period, GP) 及上行鏈路同步時間間隙 (Uplink Pilot Time Slot, UpPTS)，三者共佔用 0.275 毫秒。其中子訊框#0 與 DwPTS 專供下行傳輸用，而子訊框#1 與 UpPTS 總是提供上行傳輸用。

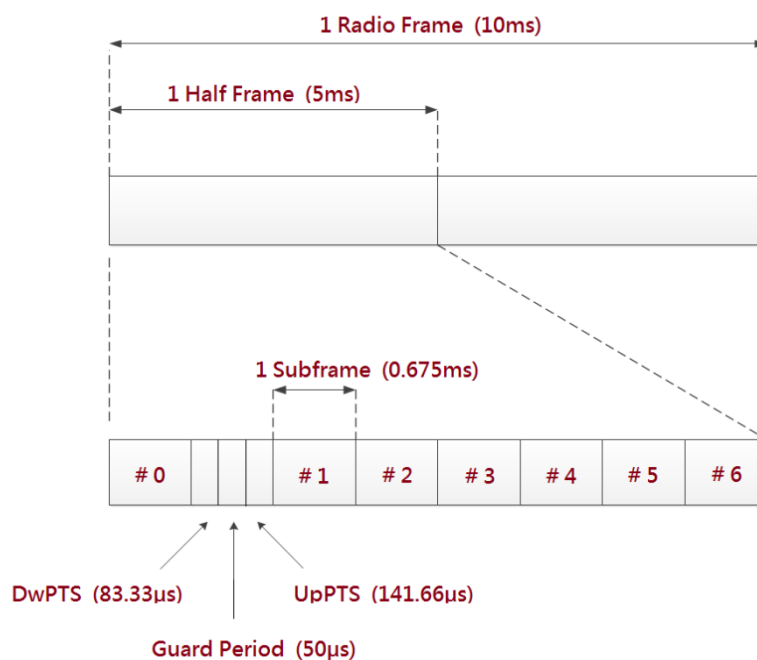


圖 1 - 3 Frame Structure Type 2 格式圖[4]

## 1.2 研究動機

LTE 是一個全數據的網路，傳輸速率下載為 100Mbps、上傳為 50Mbps。分別使用了正交分頻多工存取 (Orthogonal Frequency Division Multiple Access, OFDMA) 與單載波分頻多工存取 (Single-Carrier Frequency Division Multiple Access, SC-FDMA) 兩種不同技術。

這種新的無線傳輸技術，在資源利用上，是由子載波 (subcarrier) 及時槽 (time slot) 兩種維度所組成，單位分別為赫茲及毫秒。十二個子載波及一個時槽組成一個資源區塊 (Resource Block, RB)，此為分配使用者下載資源的基本單位。然而基地台分配資源時，因演算法的設計，可能使得資源分配不均，如

最佳通道品質法 (Best CQI)；或是雖然所有使用者皆能取得資源，但系統整體的吞吐量卻不顯著，如輪循法 (Round Robin)。因此有比例公平演算法 (Proportional Fairness) 的出現，在吞吐量以及公平性上取得平衡。

本篇論文考量使用者傳輸資料的優先權，依其即時性 (Real Time) 或非即時性 (Non Real Time) 而給定不同優先權。即時性的資料擁有較高的優先權，同時為了避免非即時性資料因長時間等待而飢餓，我們也給其回饋機制，讓其優先權能依時間而上升。在所有高優先權的資料中，給予較多的資源，剩下的資料則使用輪循的方式依序分配資源，以確保系統能擁有較佳的傳輸效益，並解決最佳通道品質法的不公平，及提升輪循法的低效益。因此如何有效地、公平地分配資源給多位使用者，為本論文研究的主题。

### 1.3 論文架構

本論文共分為五章節，其內容如下：

#### 第一章 簡介

簡介 LTE 背景、OFDMA 與 MAC 層架構、訊框格式與資源塊型式。

#### 第二章 相關研究

參考國內外的相關研究，討論在 OFDMA 中資源分配的相關議題。

#### 第三章 問題分析

詳細分析所發現的議題，針對各議題提出相應的解決方法，設計出適合的演算法及流程圖。

#### 第四章 模擬實驗與結果分析

透過模擬實驗的分析及觀察數據的結果，驗證所提出的方法得以維持服務品質與提升公平性。

#### 第五章 結論與未來研究

總結論文中前後流程所提出的問題，解決方法與未來的研究方向。

## 第二章 相關研究

### 2.1 通道品質指標

#### 2.1.1 CQI

通道品質指標 (Channel Quality Indicator, CQI) 是無線通道的通信品質量測標準，能代表一個給定通道的測量標準。一個較高的 CQI 表示一個通道擁有較高的品質，反之較低的 CQI，則擁有較低的品質。其它測量的標準，包括信號雜訊比 (Signal-to-Noise Ratio, SNR)、信號與干擾加雜訊比 (Signal to Interference plus Noise Ratio, SINR)、信號與雜訊失真比 (Signal-to-Noise and Distortion Ratio, SNDR) 等指標。

在[5]中，通道品質的好壞，影響傳輸的資料量，通道品質好，則可以用較高的調變技術 (modulation) 來傳輸資料，使得一個 OFDM symbol 可以傳遞較多的位元數；反之，通道品質差，則一個 OFDM symbol 上只能傳遞較少的位元數。

從下列圖表，可以看出信號噪訊比與通道品質指標的對應關係。以及通道品質指標與調變的對應關係。信號雜訊比低，則對應的通道品質指標也較低，所以可選擇的調變等級就較低，如四位元相位偏移調變 (Quadrature Phase-Shift Keying, QPSK)；反之，信號雜訊比高，則對應的通道品質指標也較高，可選擇的調變等級就較高，如 16-QAM (Quadrature Amplitude Modulation)、64-QAM 等。

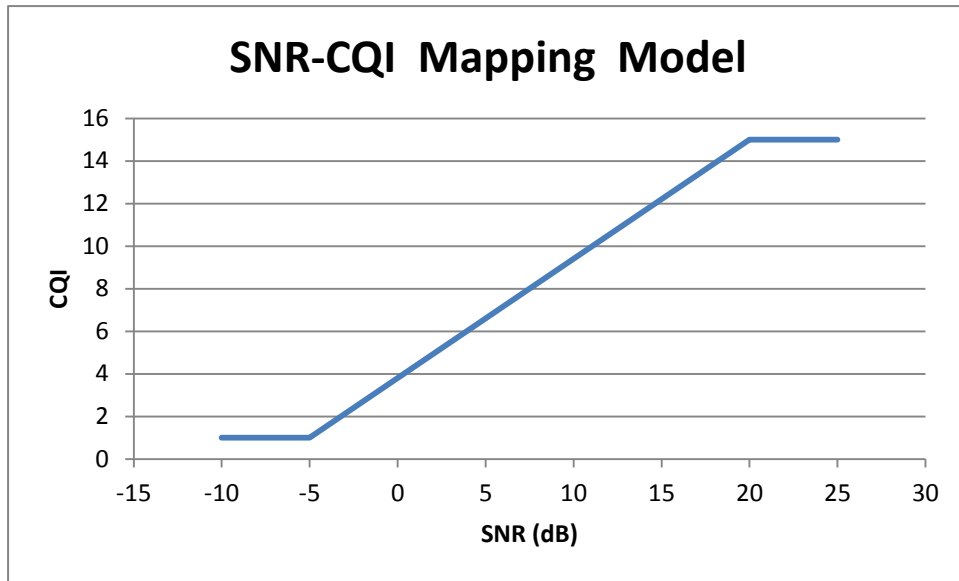


圖 2 - 1 SINR 與 CQI 之對應圖[5]

CQI Index	Modulation	Code Rate	Efficiency
0	Out of range		
1	QPSK	78/1024	0.1523
2	QPSK	120/1024	0.2344
3	QPSK	193/1024	0.3770
4	QPSK	308/1024	0.6016
5	QPSK	449/1024	0.8770
6	QPSK	602/1024	1.1758
7	16QAM	378/1024	1.4766
8	16QAM	490/1024	1.9141
9	16QAM	616/1024	2.4063
10	64QAM	466/1024	2.7305
11	64QAM	567/1024	3.3223
12	64QAM	666/1024	3.9023
13	64QAM	772/1024	4.5234
14	64QAM	873/1024	5.1152
15	64QAM	948/1024	5.5547

表 2 - 1 4-Bit CQI Table[6]

## 2.2 排程演算法

基地台在分配資源時，需要先與手機溝通，參考手機與基地台之間的通道品

質指標，判斷是否給與資源，並依據通道品質指標決定調變機制。如果使用較高的調變技術，如 QAM 等，來傳輸資料，則一個 OFDM symbol 可以傳遞較多的位元數；反之，使用較低的調變技術，如 QPSK，則一個 OFDM symbol 上只能傳遞較少的位元數。且基地台中的排程器是 MAC 層的一部分，用來控制上傳、下載資源的分配，且上傳、下載排程器是分開運作、互相獨立的。下載排程器會在每 1ms 時間間隔內動態地進行判斷，決定使用者以多少資源來接收 DL-SDH (Downlink Shared Channel) 傳輸。如果有多位使用者一同進行排程，每名使用者會有一個 DL-SCH，且都動態地對應到一組資源中。排程器不但會負責資源塊的大小、調變方式，也會控制資料傳輸速率。

一個基地台在分配資源塊 (Resource Block) 時，可能為了整體傳輸效益，例如將所有資源都分配給擁有較佳通道品質的使用者。雖然整體產能會因此提升，但擁有較低通道品質的使用者，則可能長時間分配不到資源，而產生資源分配不均的問題。另一方面，也可能為了使所有使用者都能取得資源，使得需要傳送即時、大量資料的使用者無法立即取得所需資源，而使得傳輸效益不彰。所以，如何有效地利用及分配資源，是一項非常重要的議題。

對於排程演算法，有兩個重要的參考指標：一個是吞吐量、另一個則是公平性。排程演算法的目的是充分利用通道品質隨著時間改變的特性，讓多個使用者依某種權重分配，提高系統的吞吐量。吞吐量是以在基地台涵蓋範圍中，單位時間內所傳輸的資料量來衡量。公平性是指在基地台涵蓋範圍內，所有使用者是否都能獲得一定的服務機會。好的排程演算法要能夠兼顧吞吐量以及公平性。在以下章節，我們將介紹一些常用的排程演算法。

### 2.2.1 Best CQI

Best CQI 演算法[7]在選擇傳輸的使用者時，會依據使用者的通道品質好壞來決定是否配給資源，亦即，會選擇當下擁有最高通道品質的使用者，分配其資

源。當其通道品質變差時，再選擇其它擁有最高通道品質的使用者。因此基地台永遠選擇當下擁有最好通道品質條件的使用者。

Best CQI 演算法中的吞吐量，會是吞吐量的極限值。因為在行動通訊網路中，每位使用者所在地點不相同、周遭環境不相同，接收的信號強度也不相同，因此 Best CQI 必定會服務與基地台較近、通道品質較佳的使用者，所以這種排程演算法是非常不公平的。

雖然 Best CQI 的優先權選擇是依據使用者的通道品質好壞來決定，整體來說可以擁有較好的產能，但它卻也捨棄了公平性。例如較偏遠的地區，因為地形的干擾或是距離過遠，導致訊號不易傳送、雜訊過強等問題，所以通常通道品質較差。所以在 Best CQI 之下，較低的 CQI 使用者可能就會常常無法被分配到資源，而產生所謂「飢餓」的問題。

以下用圖示來解釋 Best CQI 的運作方式。首先，基地台會根據每位使用者回傳的 CQI 高低配置資源，且優先將資源分配給當時 CQI 最好的通道。以圖為例，在圖中的第一區間，擁有較高的 CQI 為使用者 2，因為 Best CQI 會將資源分配給擁有最高的 CQI，所以此時資源會分配給使用者 2。接著，第二區間擁有較高的 CQI 為使用者 1，所以此時依照 Best CQI 演算法，會將資源分配給使用者 1。之後隨著時間的演進，擁有最高的 CQI 依序為使用者 1、使用者 3、使用者 2、使用者 2、使用者 2、使用者 2、使用者 1，所以資源也依序給與使用者 1、使用者 3、使用者 2、使用者 2、使用者 2、使用者 2、使用者 2、使用者 1。

從圖中可以看到，整個系統的吞吐量總是維持在高峰，因為系統將資源分配給擁有最高的 CQI 使用者。

至於 Best CQI 的演算法流程，首先會根據使用者回傳的通道品質指標，來判斷哪一位使用者是當下 CQI 最佳的，接著找出擁有最佳的 CQI 使用者之後，由排程器進行排程工作，將資源塊分配給此擁有最高 CQI 的使用者。



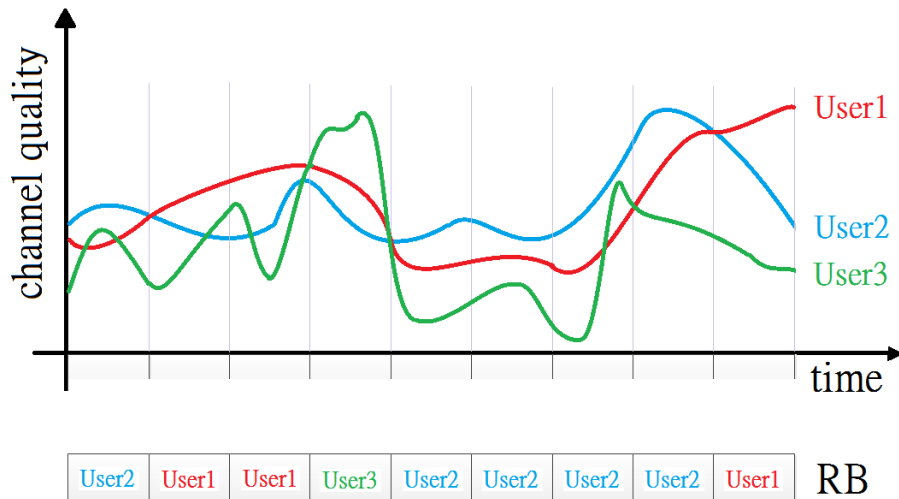


圖 2 - 2 Best CQI 之示意圖

## 2.2.2 Round Robin

因 Best CQI 演算法存在不公平的問題，無法服務通道品質較差的用戶。故為了考量到公平性，一般都會以 Round Robin 演算法[7]為衡量的準則。Round Robin 演算法是以輪流的方式，給定一個 time quantum 的時間區隔，在此時間區隔之內，只能給某一位使用者資源，該時間到了，就要把該資源給下一位使用者。因為是依序輪流，所以每位使用者被服務的機會是均等的。此外，Round Robin 演算法並不考慮使用者的 CQI，每位使用者皆能公平地被服務，資源不會被特定的使用者罷佔，所以 Round Robin 演算法是最公平的演算法。

以下圖為例，雖然每位使用者的通道品質不相同，但一段時間間隔只會給某位使用者，之後每一個時間間隔，都會用輪流的方式，分配資源給下一位使用者。

Round Robin 演算法的流程，一開始會計算目前有多少使用者，接著挑選一位使用者，並分配資源給該使用者。接著排程器開始挑選下一位使用者，之後的資源分配即依照輪流的方式進行。

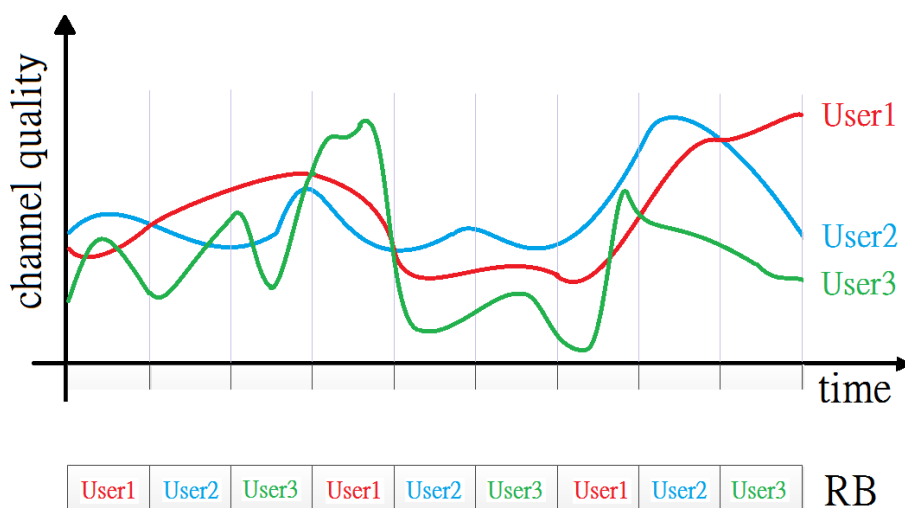


圖 2 - 3 Round Robin 之示意圖

### 2.2.3 Proportional Fair

比例式公平排程機制(Proportional Fair, PF)[8]，其產能及公平性介於 Best CQI 與 Round Robin 之間，可視為兩者的折衷方法，是由 Qualcomm 公司所提出的排程機制。在眾多的使用者中，希望在相互競爭的資源間取得一定程度的平衡，讓所有的使用者都能擁有一定程度的服務水平，不至於因搶不到資源而陷入長期等待。假設  $r_k(t, n)$  表示在時間  $t$  時，使用第  $n$  個子載波之瞬間可達到之速率， $R_k(t)$  為使用者  $k$  於時間  $t$  之平均速率，則在每個子載波，使用者  $k$  之最大配置量如下式：

$$\arg \max_k \frac{r_k(t, n)}{R_k(t)} \quad (1)$$

此公式是用來選擇擁有相對於之前速率比值最高的用戶。其中  $R_k(t)$  為第  $k$  個使用者的子載波之平均資料速率。其中，更新平均傳輸速率的公式如下：

$$R_k(t+1) = \left(1 - \frac{1}{t_c}\right) R_k(t) + \left(\frac{1}{t_c}\right) r_k(t, n) \quad (2)$$



其中 $R(t)$ 會不斷地根據過去的速率一直作修正，加總平均。 $t_c$ 是時間常數，表示滑動視窗 (sliding window) 的長度，實際反映一個用戶對於接收或是傳輸數據的容忍能力。在[8]中，較大的 $t_c$ 容忍能力較好，可以允許等待的時間較長，直到該用戶擁有較好的通道品質，也意味著系統會挑出擁有較佳通道品質的使用者，以提高系統產能，缺點是會產生一些延遲。

#### 2.2.4 Modified Largest Weighted Delay First (MLWDF)

MLWDF (Modified Largest Weighted Delay First)[9]，是針對 QoS 不同資料需求所提出的排程改良演算法。此演算法參考了 Proportional Fair 以平均傳輸速率為基礎，再加入封包延遲限制、封包遺失機率、封包延遲時間等考量。使用者的優先權不只與通道品質有關，且與延遲時間有關。

$$\arg \max_k \frac{r_k(t)}{R_k} * \frac{-\log(\delta_k)}{T_k} * D_k \quad (3)$$

此公式中， $\delta$  代表使用者的 QoS 延遲參數， $r_k(t)$  是使用者在第  $t$  個傳輸時間間隔 (Transmission Time Interval, TTI) 的瞬時傳輸速率， $R_k$  是使用者  $k$  的平均傳輸速率， $D_k$  是使用者  $k$  的封包延遲時間， $T_k$  是使用者  $k$  的封包傳輸最大延遲界限。MLWDF 排程演算法，在產能與延遲取得平衡，雖然考量延遲時間，保證 QoS，卻造成非即時性封包的等待時間過長，屬於一種不公平的排程演算法。MLWDF 排程演算法的產能較佳，且需要和允入控制一起合作，在[9]中也提到，綜合考量不同用戶間的封包延遲和當前通道品質，是針對 QoS 資料需求，提出的較佳產能排程演算法。

#### 2.2.5 Hungarian

匈牙利演算法[10]最初是用來尋找一個最佳二分圖匹配(optimum matching

of a bipartite graph)，它也可以用來解決圖論中的分配問題。它使用迭代法對行與列進行計算，從中找到一個擁有總和為最大（或最小）的完整配置。所欲解的問題必須為一方陣，若非方陣，則須假設虛擬行(列)，並令其元素為零。詳細解法步驟如下：

【Step 1】在矩陣左邊填入使用者（A 到 E），矩陣上方填入通道（W 到 Z），中間元素為對應值（如花費值、優先權值等）。

	Channel W	Channel X	Channel Y	Channel Z
UE A	10	19	8	15
UE B	10	18	7	17
UE C	13	16	9	14
UE D	12	19	8	18
UE E	14	17	10	19

圖 2 - 4 Hungarian 演算法步驟圖 I

【Step 2】確保矩陣為方陣，所以如有需要，必須補行或列。下圖因為不是方陣，故在最後增加一行，且該行的元素值需設為矩陣的最大值。接著找出各列之最小值。

10	19	<b>8</b>	15	<u>19</u>
10	18	<b>7</b>	17	<u>19</u>
13	16	<b>9</b>	14	<u>19</u>
12	19	<b>8</b>	18	<u>19</u>
14	17	<b>10</b>	19	<u>19</u>

圖 2 - 5 Hungarian 演算法步驟圖 II

【Step 3】在每列中，減去該列之最小值。接著從方陣中，找出每行的最小值。

2	11	0	7	11
3	11	0	10	12
4	7	0	5	10
4	11	0	10	11
4	7	0	9	9

圖 2 - 6 Hungarian 演算法步驟圖 III

【Step 4】在每行中，減去該行之最小值。接著從方陣中，找出值為 0 的元素。

0	4	0	2	2
1	4	0	5	3
2	0	0	0	1
2	4	0	5	2
2	0	0	4	0

圖 2 - 7 Hungarian 演算法步驟圖 IV

【Step 5】找出最少的線條（縱線或橫線）且包含最多的元素 0。如下圖，最少需使用四條直線（一條縱線及三條橫線）可以包含全部元素 0。如果最少直線數等於列數（或行數），則跳至步驟 9。

0	4	0	2	2
1	4	0	5	3
2	0	0	0	1
2	4	0	5	2
2	0	0	4	0

圖 2 - 8 Hungarian 演算法步驟圖 V

【Step 6】在未被直線包含的元素中，找出最小值。上圖最小值為 1。再將上圖被直線包圍的元素，加上此最小值。如果元素被直線包圍兩次，則需加上此最小值兩次。

1	5	2	3	3
1	4	1	5	3
3	1	2	1	2
2	4	1	5	2
3	1	2	5	1

圖 2 - 9 Hungarian 演算法步驟圖 VI

【Step 7】將方陣中所有元素，減去上步驟提及的最小值。

0	4	1	2	2
0	3	0	4	2
2	0	1	0	1
1	3	0	4	1
2	0	1	4	0

圖 2 - 10 Hungarian 演算法步驟圖 VII

【Step 8】如同步驟 5，再次找出最少的直線包圍最多的元素 0。如果直線數不等於列數（或行數），則跳至步驟 6 重覆之後步驟，直到找出包圍元素 0 的最少直線數等於列數（或行數）。

0	4	1	2	2
0	3	0	4	2
2	0	1	0	1
1	3	0	4	1
2	0	1	4	0

圖 2 - 11 Hungarian 演算法步驟圖 VIII-1

0	3	1	1	1
0	2	0	3	1
3	0	1	0	1
1	2	0	3	0
3	0	1	4	0

圖 2 - 12 Hungarian 演算法步驟圖 VIII-2

【Step 9】當找出最少直線數等於列數時，找出元素 0，且此元素 0 之該行與該列只有其值為 0。如下圖標記圓圈之元素 0，該行該列只有其為 0。

0	3	1	1	1
0	2	0	3	1
3	0	1	0	1
1	2	0	3	0
3	0	1	4	0

圖 2 - 13 Hungarian 演算法步驟圖 IX

【Step 10】將上步驟之標記圓圈之元素，對應到起始的矩陣，則該元素對

應之 UE 與通道即為所求。

	Channel W	Channel X	Channel Y	Channel Z
UE A	10	19	8	15
UE B	10	18	7	17
UE C	13	16	9	14
UE D	12	19	8	18
UE E	14	17	10	19

圖 2 - 14 Hungarian 演算法步驟圖 X

在限制條件的前提下，每位用戶最終只能配置一個通道，且不允許重複配置，但其缺點是，擁有較佳通道品質的使用者，可能會因長期佔有通道，反而使擁有較差通道品質的使用者，長期等待資源而餓死 (starvation)。

關於匈牙利演算法的複雜度計算，因為該演算法為啟發式演算法 (heuristic algorithm)，隨著矩陣不同，無法每次都將其複雜度完全精確計算出來，其計算次數約為  $(11K^3 + 12K^2 + 31K) / 6$ ，其中  $K$  為方陣的大小。基於上述，可以推算出匈牙利演算法時間複雜度的上界為  $O(K^3)$ ，空間複雜度的上界為  $O(K^2)$ 。

## 2.3 服務品質 (QoS)

服務品質 (Quality of Service, QoS)，是一種控制機制，針對不同使用者或是不同資源型態的需求，採取不同的優先順序，並保證資料流能達到一定品質的需求。舉例來說，即時影音就需要較高的服務品質，也就是對資料流量有較嚴苛的限制。如果流量過小，使用者觀看即時影音便感到不流暢。所以 QoS 的保證，對於容量有限的網路是非常重要的。

### 2.3.1 標準 QCI 屬性

QCI (QoS Class Identifier) 是進化封包系統 (Evolved Packet System, EPS) 承載最重要的 QoS 參數，是一個數量等級，其值介於 1 到 9，代表不同的 QoS 特性。下表定義了 EPS 系統的 QCI 屬性，共包含了資源型態 (resource type)、優先權 (priority)、封包延遲限制 (packet delay budget)、封包遺失率 (packet error loss rate)、服務種類 (example service)，其中資源型態又依據不同需求的服務分為保障位元速率 (Guaranteed Bit Rate, GBR) 與非保障位元速率 (Non-guaranteed Bit Rate, non-GBR) 兩類。這些參數決定了如何分配無線電資源。

QCI	Resource Type	Priority	Packet delay budget	Packet error loss rate	Example services
1	GBR	2	100 ms	$10^{-2}$	Conversational voice
2		4	150 ms	$10^{-3}$	Conversational video
3		3	50 ms	$10^{-3}$	Real time gaming
4		5	300 ms	$10^{-6}$	Non-conversational video
5	Non-GBR	1	100 ms	$10^{-3}$	IMS signaling
6		6	300 ms	$10^{-6}$	Video, TCP-based
7		7	100 ms	$10^{-6}$	Voice, Video, Interactive gaming
8		8	300 ms	$10^{-3}$	Video, TCP-based
9		9	300 ms	$10^{-6}$	

表 2 - 2 LTE QCI 各項需求參數[11]

## 2.4 總結

根據相關文獻的排程機制，選出與本研究較為相關的論文，整理成如下表格，並針對其優缺點進行比較：

排程機制	公平性	系統產能	QoS 保證	複雜度
Best CQI	最低	最高	無	較低
Round Robin	較佳	最低	無	較低
Proportional Fair	最高	次佳	較差	高
MLWDF	次佳	次佳	最高	高
Hungarian	次佳	較佳	無	最高

表 2 - 3 各排程演算法之比較表



## 第三章 研究方法

本章節將介紹允入控制 (Call Admission Control, CAC) 與排程法則。允入控制說明如何挑選適合的 UE，達到有效的頻寬利用。排程法則考量服務品質，讓即時性的資料能擁有較高的優先權，有效地安排資料傳輸的先後，讓系統的產能提高，傳輸延遲降底，滿足 QoS 需求。

### 3.1 問題分析

#### 3.1.1 網路服務品質的考量

隨著多媒體時代的來臨，網路上的資料已經不侷限在數據資料及語音。多媒體資料，像是串流或是即時影像、即時語音等，充斥著整個網路，在 QCI 參數屬性表中，它們依用途被區分為不同的資源型態。不同的資料型態會有不同的 QoS 需求等級，每個 QoS 等級都有不同的參數規定，如高畫質視頻、對延遲時間要求嚴苛的即時語音服務、或是網頁、檔案等延遲時間要求不高的傳輸類型。其中，即時性的資料通常會有延遲限定，必須在 delay constraint 時間內傳輸完成，否則即使送達目的，也將視為無效封包。以 Best CQI 為例，Best CQI 排程機制只以通道品質為分配依據，如果有些即時性的資料擁有較差的通道品質，這些傳輸的資料可能會因通道品質不佳而較慢傳輸，造成資料的延遲。

以 Proportional Fair 演算法為例，雖然傳送的資料不容易因在佇列中長期等待而餓死，但因為此排程法中並未考量傳輸資料的型態，容易導致較即時的資料無法在限制時間內傳達，而成為無效封包，造成使用者受話或是聽即時語音時有延遲的現象發生。

#### 3.1.2 公平性及優先權的考量

在現有的排程機制中，MLWDF 的優先權是依照當下傳輸速度與過去平均傳輸

速度的比值、封包的遺失機率與封包的延遲時間，而求出優先權高低。此機制雖能讓即時性封包立刻獲得大量的資源塊，卻也排擠了非即時性封包的資源塊，使非即時性封包的延遲時間大幅提高、產能大幅下降而失去公平性。

此外，資料型態為 IMS 的封包雖然延遲要求並沒有其它類型的封包來的急迫，但仍然擁有較高的傳送優先權，然而 PF 與 MLWDF 兩種機制並未能就此有效區別。根據 QCI 參數屬性表中可以得知不同的服務類型會有不同的 QCI 優先權值，此優先權值也必須是考量的因素之一。

### 3.1.3 避免佇列中的資料長期等待

常見的資源分配演算法，大部份是針對單一頻道中的各個使用者，當其擁有較好的通道品質時，可以優先分配資源給該使用者。考慮的因素可能包含平均傳輸速率、當下傳輸速率、滑動視窗時間、封包遺失機率等。但因為考慮因素不同，造成特定類型封包的使用者會長期罷佔資源塊，造成其它使用者短時間之內無法取得資源。且特定類型使用者所傳輸之資料不見得有急迫性，使得其它有急迫性之封包可能枯等造成「餓死」，因為此種特定需求而造成不公平，使得傳輸效益大打折扣。

### 3.1.4 時間複雜度的考量

在動態的資源分配演算法中，匈牙利演算法能有效的分配資源給各個使用者，使得分配最佳化。匈牙利演算法需經幾個步驟，如一開始的初始化，再從每位使用者中找尋該使用者對應的各通道中的最小比較值（假設以最佳 CQI 為考量，則此比較值即為 CQI 的值），再從每個通道中找尋該通道對應的各使用者中的最小值。經由演算法步驟而增減數值並使之為零，再從眾多零當中，找出最少的線，來包含全部的零，經由不斷的驗證，找出適合每位使用者的通道。由於此演算法，必須不斷地驗證是否由最少的線，來包含最多的零，導致時間複雜度為 $O(K^3)$ ，

空間複雜度為 $O(K^2)$ ，極度不利於排程演算法的即時性且大大增加基地台的負擔。

## 3.2 研究方法

綜合上一小節提出的問題，我們結合允入控制、公平性、以及封包優先權的考量，設計出一個稱為 AMLWDF (Adaptive Modified Largest Weighted Delay First) 的演算法。

### 3.2.1 允入控制連線數之評估與調適

在 LTE 中，當使用者對基地台發出頻寬請求時，基地台先會判斷使用者的通道品質以及 QCI 等可供查詢服務品質的資訊。但對於眾多的使用者，要如何有效挑選使用者也是一個課題。

使用者欲傳遞的資料，依 QCI 可分為兩類，一類為 GBR，另一類為 non-GBR，GBR 有時效性，必須在規定的時間內完成傳送，否則視為無效封包。而另一類 non-GBR 則有較寬的限制。所以一般來說，GBR 的資料大多是語音封包、即時影像、即時語音，需要即時互動的類型。而 non-GBR 則是一般檔案的傳輸、網頁、電子郵件、peer-to-peer 等。此外，在 non-GBR 中，又有 IMS 的資料類型，屬於控制訊號，其優先權為 1，高於 QCI 其它 GBR 與 non-GBR，故需要優先處理。因此，在挑選使用者時，首先會對此三種型態 IMS、GBR、non-GBR 作分類。

接下來，為了達到公平性，每一種型態 (IMS、GBR、non-GBR) 至少都會有一位使用者進入排程演算法。所以當可以使用的通道數過少時，在公平性的原則下，會輪流分配；當可使用的通道數足夠時，會先分配給每一種資源型態至少都能擁有一個通道。

### 3.2.3 QoS 下 AMLWDF 之計算

MLWDF 演算法根據過去的平均傳輸速度，以及當下的瞬時速度，求出一個比較值(公式 4)。藉由這個比較值，來判斷當下哪一位使用者較適合優先配給資源塊。但 MLWDF 演算法並不能有效地區分 QCI 的九個優先等級，只能粗略地符合 QoS 需求。所以在此，我們必須在 MLWDF 演算法中，再加上一個優先權的因子。此因子呈指數分佈，用來將 QCI 優先權的九個級距放大，且能有效提高較高優先權的權重，使 MLWDF 演算法更有效益。公式如下：

$$\arg \max_k \frac{r_k(t)}{R_k} * \frac{-\log(\delta_k)}{T_k} * D_k * \text{Exp}\left(\frac{1}{\text{Priority}_k}\right) \quad (4)$$

上式為 AMLWDF 之公式，在眾多的使用者中，選出使用者 k。 $\delta$  代表使用者 k 的 QoS 延遲參數， $r_k(t)$  是使用者在第 t 個 TTI 的瞬時傳輸速率， $R_k$  是用戶 k 的平均傳輸速率， $D_k$  是使用者 k 的封包延遲時間， $T_k$  是使用者 k 的封包傳輸最大延遲界限。 $\text{Priority}_k$  為使用者 k 的 QCI 優先等級，取倒數再取指數的目的是讓 QCI 優先等級較高的使用者，擁有較大的 AMLWDF 值。而 QCI 優先等級分佈較中間或是較低的使用者，AMLWDF 值會較相近，但仍能區分優先權而不致於優先權反轉。此目的是為了讓 IMS 封包能擁有最高優先權可能；GBR 之封包能擁有次高優先權可能；non-GBR 之封包則擁有較低之優先權可能。

### 3.2.4 AMLWDF 動態資源塊之分配

一般常見的排程演算法，通常只針對一個通道當中的所有使用者，依照其指標的高低，例如比較 CQI (或 SINR) 的高低，或是比較在佇列裡的排隊順序，以及比較過去平均傳輸速率與當下瞬時傳輸速率的比值。如果沒額外的限制，有可能會發生同一位使用者，同時擁有大量的通道數，即在某一時刻之下，他同時可以獲得大量資源塊。雖然就演算法的設計而言，可能當下對於此使用者非常有利，但客觀來說，資源過度集中在某些使用者，對整體而言並不公平。如果該使用者一直擁有很高的指標數值，則會造成對其它使用者的不利。

使用動態資源演算法的好處，就是它會針對所有使用者及所有通道做有效率的分配，在使用者數與通道數相同的情況下，讓每位使用者必能取得一個通道，以「公平性」來看每個人都有機會取得資源，而不會造成所有資源都過分集中在同一位使用者身上。

在一些動態資源的演算法中，擁有效益排程最大化的莫過於匈牙利演算法。匈牙利演算法針對通道與使用者配對的所有可能性，反復運算，以取得最佳解，在所有使用者之間取得平衡，以提昇系統的最大效益。但因為不斷反復運算，造成時間複雜度高達 $O(K^3)$ 。因此在本論文中，採取折衷的做法，使用近似最佳解的貪婪演算法，既可達到逼近匈牙利的效率，而時間複雜度也不致於過大，也可有效分配使用者資源，以符合公平及效能的折衷辦法。

### 3.3 排程演算法之架構與流程

透過允入控制機制可決定允許進入系統的 UE。將這些 UE 存入佇列中，經過運算決定每一個 UE 該配置多少個 RB，計算完之後依照 AMLWDF 的優先權值挑選適合的 UE 配給 RB，以完成此次排程。排程詳細步驟如下：

#### 【STEP 1】計算 Resource Block 數

首先，必須知道整個系統的頻寬，且 OFDM symbol 需要 10%的保護頻帶，所以假設系統分配到的頻帶寬為 5MHz，則實際可用的頻帶寬為 4.5MHz。一個子載波的頻寬佔 15kHz，一個 Resource Block (RB)有十二個子載波，共佔頻寬 180kHz，經由 3GPP 規定的標準可以得知可用的 RB 數量。

Channel bandwidth (MHz)	1.25	3	5	10	15	20
Number of RBs in frequency band	6	15	25	50	75	100

表 3 - 1 頻帶與資源塊之對應表[12]



## 【STEP 2】計算 User Equipment 數

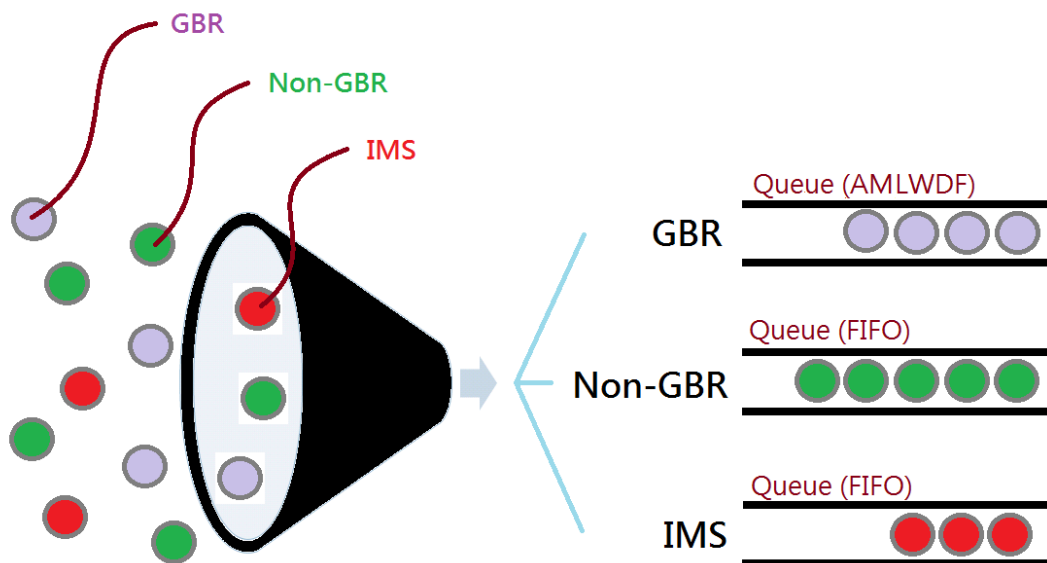
計算使用者裝置( User Equipment, UE )的傳輸需求。系統一開始會統計各個使用者裝置所需要的服務，將之存在佇列中以等待資源，等待基地台提供服務。我們可以藉由佇列，來得知當前共有多少個 UE 需要被服務。

## 【STEP 3】分類資源型態

根據 QCI 中的資源型態，可分為 GBR 與 non-GBR 兩類，其中 non-GBR 包含 IMS signaling，為控制信號。GBR 對於時間的延遲有較嚴格的限制，而 non-GBR 則較為寬鬆。QCI 的值與優先權大致吻合，但仍有少數不同。例如，QCI 表中 non-GBR 類別裡的 IMS signaling，雖然 QCI 的值為 5，但卻有最高的優先權 ( Priority 為 1)，為了避免之後的演算法造成優先權反轉，故此處先將 IMS signaling 特別挑出。因此，我們將資源型態分為三類：優先權最高的 IMS 類、對時間延遲較嚴格的 GBR 類、以及對時間延遲較不嚴格的 non-GBR 類。

## 【STEP 4】存入佇列

如下圖，根據不同的資源型態，存入不同的佇列。首先將 IMS Signal 資源型態挑出後，依照先後順序存入 IMS 佇列中。Non-GBR 資源型態挑出後，依先後順序，存入 non-GBR 佇列中。GBR 的資源型態最後挑出。將所有挑出的 UE 按照 AMLWDF 演算法求出新的優先權值，再按照優先權高低，依序存入 GBR 佇列中。



Classify Resource Type

Store in Queue

圖 3 - 1 分類資源型態示意圖

【STEP 5】挑選 UE

為了確保每種資源型態，在每一次分配中，都能分配到資源塊，所以在分配資源時先預留保留空間給每一種資源型態。令保留空間如下：

$$N_{Guard-IMS} = \left\lceil \frac{\sqrt{N_{IMS}}}{N_{GBR} + N_{Non-GBR} + N_{IMS}} \right\rceil * N_{RB} \quad (5)$$

$$N_{Guard-NonGBR} = \left\lceil \frac{\sqrt{N_{Non-GBR}}}{N_{GBR} + N_{Non-GBR} + N_{IMS}} \right\rceil * N_{RB} \quad (6)$$

$$N_{GBR} = N_{RB} - (N_{Guard-IMS} + N_{Guard-NonGBR}) \quad (7)$$

令此次 slot 對於  $UE_i$  配置數為：

$$RB\_Allocated_i$$

首先計算 IMS signal 資源型態所需的資源塊數量 $N_{Guard-IMS}$ ，再從 IMS 佇列中挑選出前 $N_{Guard-IMS}$ 個 UE，被挑選出後的 UE 如果還有剩餘的資料要傳送，則再重新存入佇列中排隊。被挑選中的 $UE_i$ ，其 $RB\_Allocated_i$ 數要累加。接著計算 non-GBR 資源型態所需的資源塊數量 $N_{Guard-NonGBR}$ ，再從 non-GBR 佇列中挑選出前 $N_{Guard-NonGBR}$ 個 UE，被挑選出後的 UE 如果還有剩餘的資料要傳送，則再重新存入佇列中排隊。被挑選中的 $UE_i$ ，其 $RB\_Allocated_i$ 數要累加。最後計算剩餘的 RB 數，求出 $N_{GBR}$ 後，再從 GBR 佇列中挑選出前 $N_{GBR}$ 個 UE。被挑選出後的 UE 如果還有剩餘的資料要傳送，則再重新存入佇列中排隊。被挑選中的 $UE_i$ ，其 $RB\_Allocated_i$ 數要累加。

公式中，取開根號的目的，是因為當 UE 數量快速上升時，保留區間的數量不致於因此快速成長。UE 少的時候，GBR、non-GBR、IMS 的數量皆能維持一定的比例，以達到公平性。但當一個基地台中的 UE 數上升時，如果不取開根號，會造成 GBR 的 UE 空間被限縮，會無法保證 GBR 都能在限制的時間之內達到一定的傳輸量，QoS 也就無法被保證。對此，取開根號能減少 non-GBR 與 IMS 的數量，再加上 GBR 的數量是用總通道數量減去 non-GBR 及 IMS 數量，而不是取開根號，以此保證 GBR 能維持一定的數量，符合 QoS 的要求。

#### 【STEP 6】UE 與通道置入矩陣

將被挑選的 UE 存入方陣中，某些 UE 經前面計算可能會被分配到兩個以上的資源塊，也就是 $RB\_Allocated_i$ 數可能大於 1。則該 UE 要填入方陣中的次數為 $RB\_Allocated_i$ 。每填入一次方陣， $RB\_Allocated_i$ 減 1，直到 $RB\_Allocated_i$ 為零為止。依序查看每種資源型態的所有佇列中的每位 UE 是否都分配完畢。圖 3-2 說明不同資源型態從佇列取出的方式，而圖 3-3 說明不同資源型態需從佇列中應取出的個數計算方式。



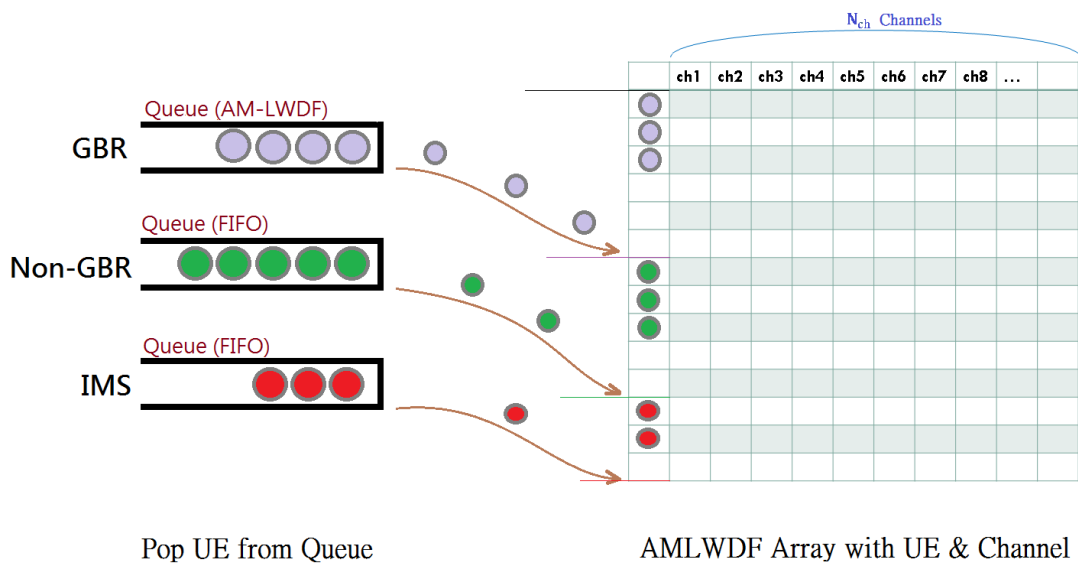


圖 3 - 2 AMLWDF 短陣圖

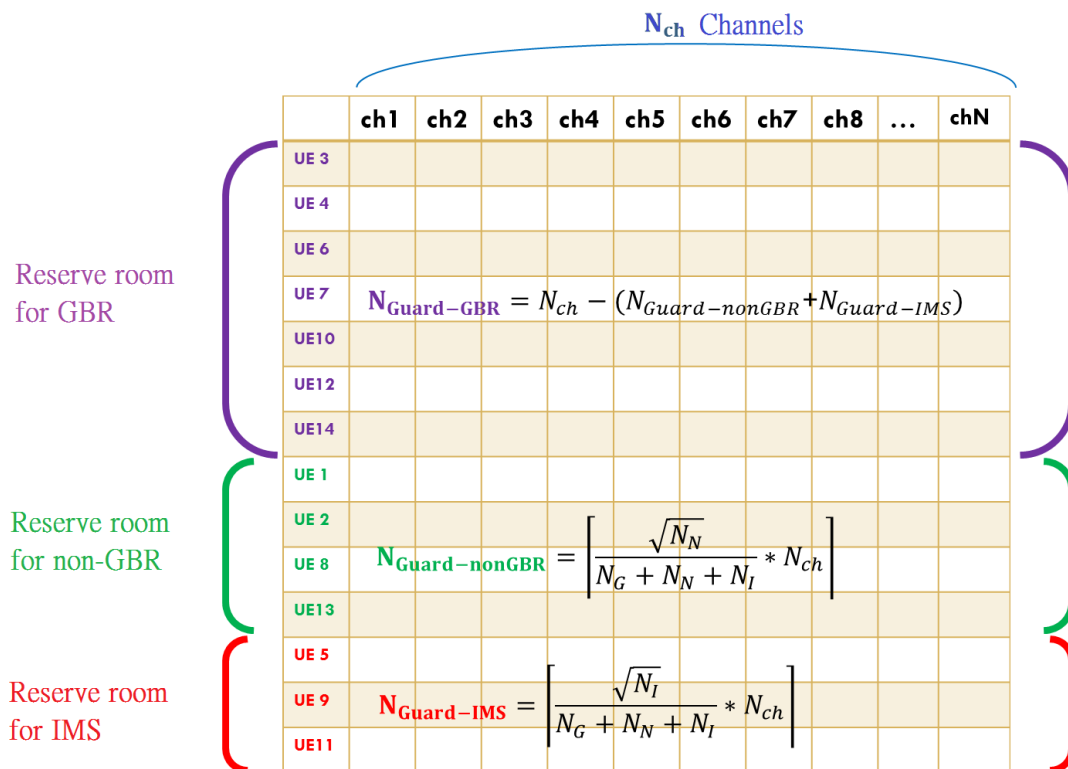


圖 3 - 3 保留區間圖

【STEP 7】計算 AMLWDF 優先權值

在配置資源塊前，必需先計算優先權值 AMLWDF，其計算方式如下：

$$AMLWDF_{k, c} = \frac{r_k(t)}{R_k} * \frac{-\log(\delta_k)}{T_k} * D_k * \text{Exp}\left(\frac{1}{\text{Priority}_k}\right) \quad (8)$$

以此計算出每位使用者 k 對於不同通道 c 的 AMLWDF 值，如下圖所示填入矩陣中。

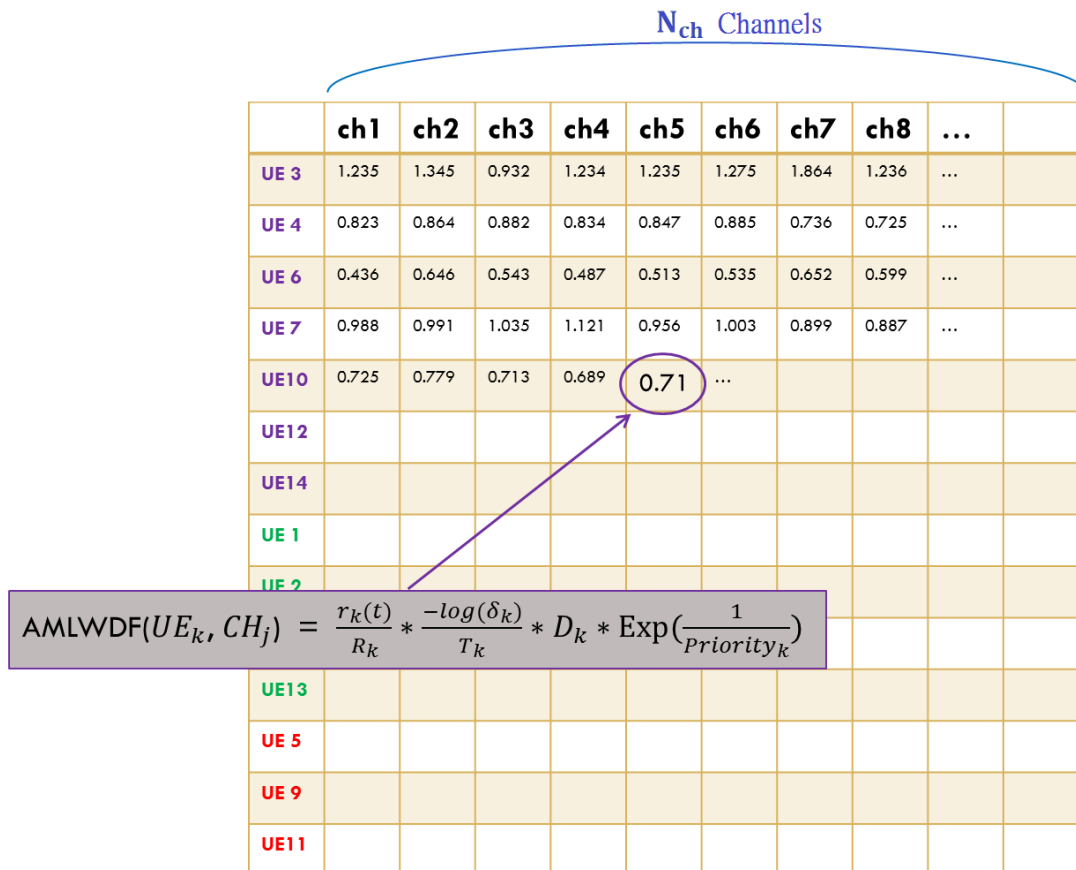


圖 3 - 4 UE 之 AMLWDF 優先權值計算圖

【STEP 8】Greedy 演算法 (配置 GBR 與 non-GBR)

首先分配 GBR 與 non-GBR，針對不同的通道，找出尚未被分配的 UE，依照下列公式挑選 UE：

$$\arg \max_k \frac{r_k(t)}{R_k} * \frac{-\log(\delta_k)}{T_k} * D_k * \text{Exp}\left(\frac{1}{\text{Priority}_k}\right) \quad (9)$$

求出擁有最大優先權值的使用者 k，將此通道配置給 k。此通道配置完畢後，配置下一通道，已被配置的使用者 UE 則不會再被分配。如果 UE 要配置的 RB 數

大於 1，則暫時視為不同的 UE。

以圖 3-5 與圖 3-6 為例，在通道 1 中 GBR 與 non-GBR 之所有 UE，擁有最大的 AMLWDF 優先權值為 UE8，其值為 1.364，大於其它 UE，所以通道 1 將先配置給 UE8。依此類推，通道 2 會分配給擁有最大 AMLWDF 優先權值的 UE3。

	ch1	ch2	ch3	ch4	ch5	ch6	ch7	ch8	...	
UE 3	<b>1.235</b>	<u>1.345</u>	0.932	1.234	1.235	1.275	1.864	1.236	...	
UE 4	<b>0.823</b>	0.864	0.882	0.834	<u>0.847</u>	0.885	0.736	0.725	...	
UE 6	<b>0.436</b>	0.646	0.543	0.487	0.513	0.535	0.652	0.599	...	
UE 7	<b>0.988</b>	0.991	<u>1.035</u>	1.121	0.956	1.003	0.899	0.887	...	
UE10	<b>0.725</b>	0.779	0.713	0.689	0.711	<u>0.789</u>	0.764	0.748	...	
UE12	<b>0.534</b>	0.557	0.623	0.626	0.613	0.648	0.615	<u>0.678</u>	...	
UE14	<b>0.911</b>	0.934	0.789	<u>0.846</u>	0.942	0.915	0.648	0.877	...	
UE 1	<b>0.482</b>	0.613	0.453	0.513	0.435	0.497	<u>0.678</u>	0.598	...	
UE 2	<b>0.315</b>	0.378	0.394	0.366	0.315	0.376	0.381	0.394	...	
UE 8	<b>1.364</b>	1.387	1.456	1.351	1.299	1.371	1.451	1.427	...	
UE13	<b>0.245</b>	0.216	0.297	0.311	0.284	0.267	0.234	0.255	...	
UE 5	0.484	0.498	0.524	0.561	0.478	0.548	0.612	0.548	...	
UE 9	0.547	0.534	0.568	0.561	0.613	0.597	0.545	0.598	...	
UE11	0.635	0.658	0.594	0.587	0.631	0.547	0.694	0.580	...	

圖 3 - 5 GBR 與 non-GBR 之 Greedy 演算法步驟圖 I

	ch1	ch2	ch3	ch4	ch5	ch6	ch7	ch8	...	
UE 3	1.235	<b>1.345</b>	0.932	1.234	1.235	1.275	1.864	1.236	...	
UE 4	0.823	<b>0.864</b>	0.882	0.834	<u>0.847</u>	0.885	0.736	0.725	...	
UE 6	0.436	<b>0.646</b>	0.543	0.487	0.513	0.535	0.652	0.599	...	
UE 7	0.988	<b>0.991</b>	<u>1.035</u>	1.121	0.956	1.003	0.899	0.887	...	
UE10	0.725	<b>0.779</b>	0.713	0.689	0.711	<u>0.789</u>	0.764	0.748	...	
UE12	0.534	<b>0.557</b>	0.623	0.626	0.613	0.648	0.615	<u>0.678</u>	...	
UE14	0.911	<b>0.934</b>	0.789	<u>0.846</u>	0.942	0.915	0.648	0.877	...	
UE 1	0.482	<b>0.613</b>	0.453	0.513	0.435	0.497	<u>0.678</u>	0.598	...	
UE 2	0.315	<b>0.378</b>	0.394	0.366	0.315	0.376	0.381	0.394	...	
UE 8	<u>1.364</u>	1.387	1.456	1.351	1.299	1.371	1.451	1.427	...	
UE13	0.245	<b>0.216</b>	0.297	0.311	0.284	0.267	0.234	0.255	...	
UE 5	0.484	0.498	0.524	0.561	0.478	0.548	0.612	0.548	...	
UE 9	0.547	0.534	0.568	0.561	0.613	0.597	0.545	0.598	...	
UE11	0.635	0.658	0.594	0.587	0.631	0.547	0.694	0.580	...	

圖 3 - 6 GBR 與 non-GBR 之 Greedy 演算法步驟圖 II

當所有的使用者 k 都被分配完時，如果還有剩餘通道，則再挑選剩餘通道中擁有最大 AMLWDF 值的使用者，並且比較該通道此使用者的 AMLWDF 值是否優於此使用者之前配置的通道如果較優的話，則該通道分配給此使用者，之前的通道則空出。當所有的通道都執行完時，則分配 IMS。舉例來說，圖 3-7 中 GBR 與 non-GBR 之 UE 共有 11 位，所以當配置到第 11 個通道時，第一輪會配置完畢。通道 12、13、14 即是所謂剩餘通道。例如在通道 14 中擁有最大的 AMLWDF 優先權值者為 UE7，而 UE7 在之前是預先配置給通道 3，此時則必須比較 UE7 在通道 3 與在通道 14 的兩個 AMLWDF 優先權值何者較大，因為圖例中，UE7 在通道 14 的優先權值大於通道 3，所以需將通道 3 空出，再將通道 14 配置給 UE7，以完成此步驟。

	ch1	ch2	ch3	ch4	ch5	ch6	ch7	ch8	...	ch14
UE 3	1.235	<u>1.345</u>	0.932	1.235	1.235	1.275	1.864	1.236	...	<b>1.013</b>
UE 4	0.823	0.864	0.882	0.834	<u>0.847</u>	0.885	0.736	0.725	...	<b>0.912</b>
UE 6	0.436	0.646	0.543	0.487	0.513	0.535	0.652	0.599	...	
UE 7	0.988	0.991	<del>1.005</del>	1.121	0.956	1.003	0.899	0.887	...	<b>1.271</b>
UE10	0.725	0.779	0.713	0.689	0.711	<u>0.789</u>	0.764	0.748	...	<b>0.845</b>
UE12	0.534	0.557	0.623	0.626	0.613	0.648	0.615	<u>0.678</u>	...	<b>0.711</b>
UE14	0.911	0.934	0.789	<u>0.846</u>	0.942	0.915	0.648	0.877	...	<b>0.697</b>
UE 1	0.482	0.613	0.453	0.513	0.435	0.497	<u>0.678</u>	0.598	...	<b>0.612</b>
UE 2	0.315	0.378	0.394	0.366	0.315	0.376	0.381	0.394	...	<b>0.401</b>
UE 8	<u>1.364</u>	1.387	1.456	1.351	1.299	1.371	1.451	1.427	...	<b>1.251</b>
UE13	0.245	0.216	0.297	0.311	0.284	0.267	0.234	0.255	...	<b>0.282</b>
UE 5	0.484	0.498	0.524	0.561	0.478	0.548	0.612	0.548	...	0.571
UE 9	0.547	0.534	0.568	0.561	0.613	0.597	0.545	0.598	...	0.594
UE11	0.635	0.658	0.594	0.587	0.631	0.547	0.694	0.580	...	0.561

圖 3 - 7 GBR 與 non-GBR 之 Greedy 演算法步驟圖 III

【STEP 9】Greedy 演算法 (配置 IMS)

針對資源型態為 IMS 的 UE，如同之前配置一樣。針對不同的通道，找出尚未被分配的 UE，依照 AMLWDF 值挑選擁有最大優先權值的使用者 k，將此通道配置給 k。此通道配置完畢之後，則配置下一通道，已被配置的使用者 UE 則不會再被分配。圖 3-8 中，剩餘三個通道與三位 IMS 之 UE 尚未被分配，故依據先前步驟，比較 AMLWDF 優先權值大小來決定配置結果。

	ch1	ch2	ch3	ch4	ch5	ch6	ch7	ch8	...	ch14
UE 3	1.235	1.345	0.932	1.234	1.235	1.275	1.864	1.236	...	1.013
UE 4	0.823	0.864	0.882	0.834	0.847	0.885	0.736	0.725	...	0.912
UE 6	0.436	0.646	0.543	0.487	0.513	0.535	0.652	0.599	...	0.577
UE 7	0.988	0.991	1.035	1.121	0.956	1.003	0.899	0.887	...	1.271
UE10	0.725	0.779	0.713	0.689	0.711	0.789	0.764	0.748	...	0.845
UE12	0.534	0.557	0.623	0.626	0.613	0.648	0.615	0.678	...	0.711
UE14	0.911	0.934	0.789	0.846	0.942	0.915	0.648	0.877	...	0.697
UE 1	0.482	0.613	0.453	0.513	0.435	0.497	0.678	0.598	...	0.612
UE 2	0.315	0.378	0.394	0.366	0.315	0.376	0.381	0.394	...	0.401
UE 8	1.364	1.387	1.456	1.351	1.299	1.371	1.451	1.427	...	1.251
UE13	0.245	0.216	0.297	0.311	0.284	0.267	0.234	0.255	...	0.282
<b>UE 5</b>	0.484	0.498	0.524	0.561	0.478	0.548	0.612	0.548	...	0.571
<b>UE 9</b>	0.547	0.534	0.568	0.561	0.613	0.597	0.545	0.598	...	0.594
<b>UE11</b>	0.635	0.658	0.594	0.587	0.631	0.547	0.694	0.580	...	0.561

圖 3 - 8 IMS 之 Greedy 演算法步驟圖

**【STEP 10】配置完成**

當所有的通道都有被挑選的 UE 時，則通道與使用者配置完畢。每個通道含有一個 RB，可根據此方陣的對應關係，找出 UE 該配置到何通道，亦即取得何 RB。此次 slot 到此分配完畢，如圖 3-9 所示，通道 1 對應 UE8，通道 2 對應 UE3，通道 3 對應 UE11，依此類推。

	ch1	ch2	ch3	ch4	ch5	ch6	ch7	ch8	...	ch14
<b>UE 3</b>	1.235	1.345	0.932	1.234	1.235	1.275	1.864	1.236	...	1.013
<b>UE 4</b>	0.823	0.864	0.882	0.834	0.847	0.885	0.736	0.725	...	0.912
<b>UE 6</b>	0.436	0.646	0.543	0.487	0.513	0.535	0.652	0.599	...	0.577
<b>UE 7</b>	0.988	0.991	1.035	1.121	0.956	1.003	0.899	0.887	...	1.271
<b>UE10</b>	0.725	0.779	0.713	0.689	0.711	0.789	0.764	0.748	...	0.845
<b>UE12</b>	0.534	0.557	0.623	0.626	0.613	0.648	0.615	0.678	...	0.711
<b>UE14</b>	0.911	0.934	0.789	0.846	0.942	0.915	0.648	0.877	...	0.697
<b>UE 1</b>	0.482	0.613	0.453	0.513	0.435	0.497	0.678	0.598	...	0.612
<b>UE 2</b>	0.315	0.378	0.394	0.366	0.315	0.376	0.381	0.394	...	0.401
<b>UE 8</b>	1.364	1.387	1.456	1.351	1.299	1.371	1.451	1.427	...	1.251
<b>UE13</b>	0.245	0.216	0.297	0.311	0.284	0.267	0.234	0.255	...	0.282
<b>UE 5</b>	0.484	0.498	0.524	0.561	0.478	0.548	0.612	0.548	...	0.571
<b>UE 9</b>	0.547	0.534	0.568	0.561	0.613	0.597	0.545	0.598	...	0.594
<b>UE11</b>	0.635	0.658	0.594	0.587	0.631	0.547	0.694	0.580	...	0.561

圖 3 - 9 AMLWDF 配置完成圖



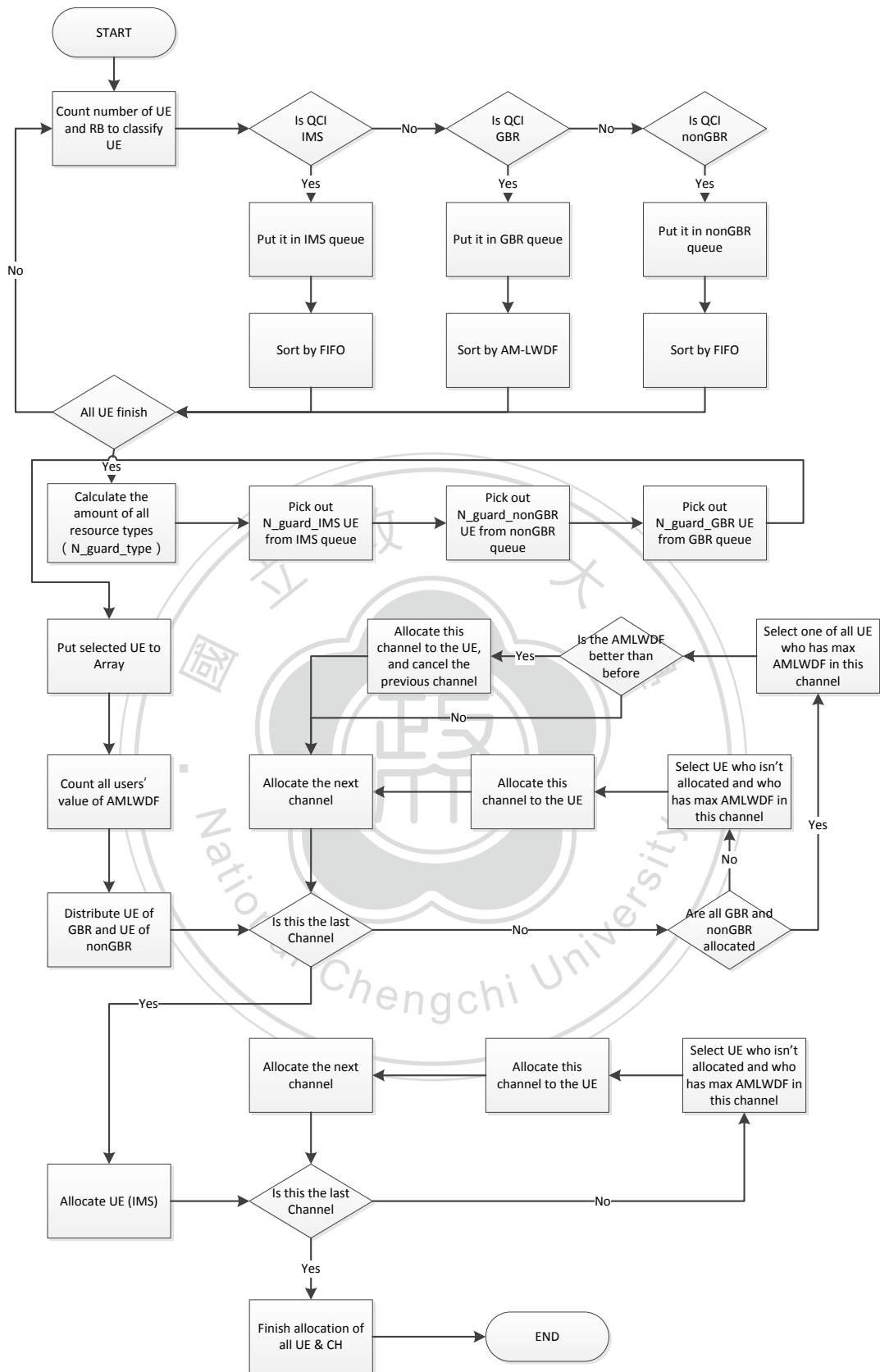


圖 3 - 10 AMLWDF 之流程圖



## 第四章 模擬實驗與結果分析

### 4.1 實驗環境與假設

實驗是使用 Network Simulator 3.13 模擬器[13]以及 Lena Project[14]模組進行，並使用 C++ 語言修改此模組，加入第三章的改進部份，以此撰寫腳本進行 LTE 網路實驗的模擬。

#### 4.1.1 評估指標

本實驗使用了四個評估指標，用來探討我們的方法與其它方法之間的優劣。其四個指標如下：

##### 【Jain's Fairness Index】

此為公平性指標，其值介於 0 與 1 之間，用來判斷所有使用者所傳輸的資料量是否平均。當每一位使用者在單位時間內所傳遞的資料均相等時，其值為 1。反之，如果每一位使用者所傳遞的資料量差距很大時，此指標值會小於 1。值愈大代表就資料量而言愈公平，值愈小代表愈不公平。公式如下：

$$J(x_1, x_2, \dots, x_n) = \frac{(\sum_{i=1}^n x_i)^2}{n * \sum_{i=1}^n x_i^2} \quad (10)$$

其中  $x_i$  代表使用者  $i$  的 throughput， $n$  為使用者 UE 的總數。

##### 【Throughput】

平均產能指的是單位時間內，使用者所傳輸的量。傳輸的量愈多，代表使用者在一定的時間內，接收到的資料量愈大，所花費全部資料的傳輸時間相對變短。而傳輸的量愈少，則代表使用者在一定的時間內，接收到的資料量愈少，所花費全部資料的傳輸時間相對較多。公式如下：

$$\text{Throughput} = \frac{\sum (\text{Packets} \times \text{Payload\_Size})}{\text{Total\_Time}} \quad (\text{bits/sec}) \quad (11)$$

### 【Delay Time】

平均延遲時間，指的是使用者在傳輸資料時，傳送端發出資料與接收端收到資料時的間隔時間。延遲時間通常受到網路流量的多寡而影響，延遲時間愈短代表網路狀況愈好，使用者可以在需求時間內收到資料。延遲時間愈長代表網路狀況不好，使用者收到資料時間較久，可能影響服務品質，也可能因為延遲時間超過封包傳輸的限制，導致收到超時無效的封包，造成資料被丟棄。公式如下：

$$\text{DelayTime} = \text{Time}_{\text{Arrival}} - \text{Time}_{\text{Enter}} \quad (\text{ms}) \quad (12)$$

### 【Jitter】

抖動率，指的是延遲時間彼此之間的變化量。抖動率愈小，代表網路傳輸愈穩定，抖動率愈大，代表每次封包延遲速率皆不同，才會造成差值也不同。因此，較好的網路品質，同時也會有較低的抖動率。公式如下：

$$\text{Jitter} = \text{Delay\_Time}_k - \text{Delay\_Time}_{k-1} \quad (\text{ms}) \quad (13)$$

其中 k 代表第 k 個封包，計算方式為第 k 個封包的延遲時間減去第 k-1 個封包的延遲時間。

## 4.1.2 模擬環境

模擬環境為一個都市環境，且每位使用者 UE 的速率不超過時速 3km/h。整體頻帶為 5MHz，每個單載波佔 12kHz，18 個單載波在一個 slot 時間單位即為一個 Resource Block。在同一個 slot 時間中，共有 25 個可用的 Resource Block。每個傳輸時間間隔 (TTI) 含有兩個 slot，共佔 1ms 時間。

參數	設定
Cellular Layout	omnidirectional antenna
Radius	1km
System Bandwidth	5MHz
Subcarriers per RB	12
RB Bandwidth	180kHz
Number of RBs	25
Cell-Level User Distribution	Uniform
Number of Active Users in Cell	10 ~ 60
Transmission Time Interval (TTI)	1ms
HARQ Process	OFF
Modulation and Coding Rate Setting	QPSK: 1/3, 1/2, 2/3, 3/4 16QAM: 1/2, 2/3, 3/4 64QAM: 1/2, 2/3, 3/4

表 4 - 1 模擬環境參數

## 4.2 實驗數據與分析

### 4.2.1 實驗一

在本次實驗中，我們會計算出每個使用者 UE 的產能，根據 Jain's Fairness Index 公式求出公平性指標數值，來驗證各個演算法在公平性上的差異。以及針

對不同的使用者 UE，求出不同 Resource Type 的產能差異。

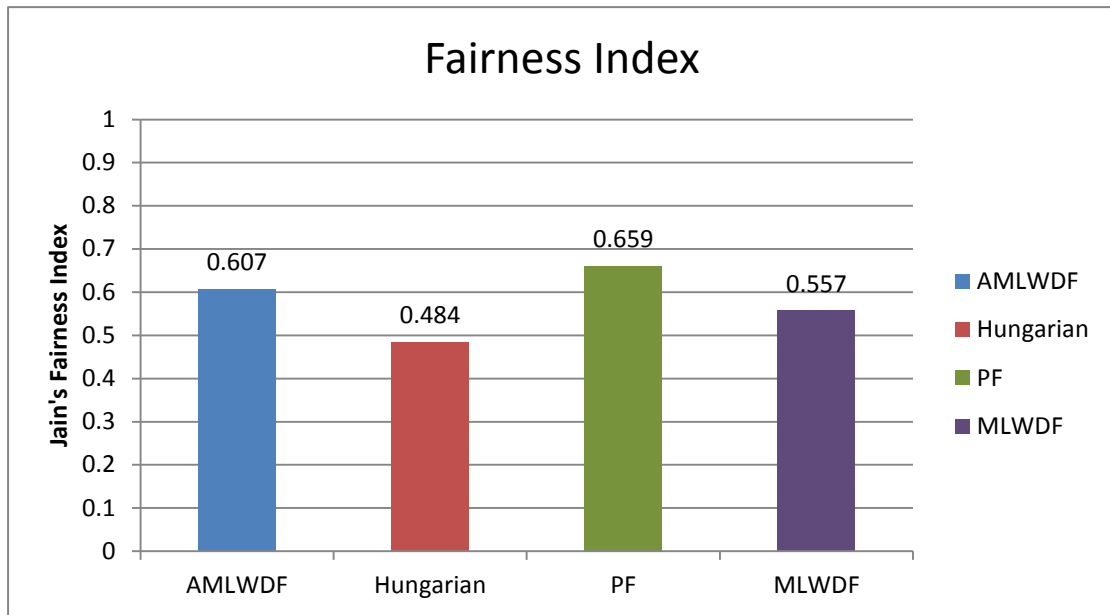


圖 4 - 1 Jain' Fairness Index

Algorithm	Improvement (Fairness Index)
Hungarian	25.44%
PF	-7.95%
MLWDF	9.03%

表 4 - 2 AMLWDF 之公平性改進比例

我們所提出的方法為 AMLWDF，上圖中 AMLWDF 方法的公平性略遜於 PF 演算法，但優於 Hungarian 與 MLWDF 演算法。PF 演算法並不會針對資源型態而有所差異，所有資源型態的資料都一視同仁，唯一的影響就是使用者 UE 的通道品質，通道品質差的 UE，最後會取得資源，通常品質好的 UE 則會慢慢的失去資源，故以長時間來看，每位使用者所取得的資源量大約相同，所以能擁有較佳的公平性。AMLWDF 演算法與 MLWDF 皆考慮 QoS，在各種不同的資源型態中，GBR 會擁有較高的優先權，容易取得較多的資源，而 non-GBR 的優先權次之，故取得較少的資源，

所以公平性皆不若 PF 來的好。但 AMLWDF 演算法採取了針對各個通道與使用者 UE 的公平分配，在 GBR 與 non-GBR 做平衡，故公平性高於 MLWDF。而 Hungarian 雖然針對通道的 throughput 最大化，卻並未考慮優先權平衡的機制與使用者持有資源時間長短的控制，造成通道品質好的 UE 一旦搶到資源，就容易造成長時間佔據，通道品質差的 UE 則容易因長時間搶不到資源而餓死 (Starvation)，因資源分配極不均勻，故其公平性最差。

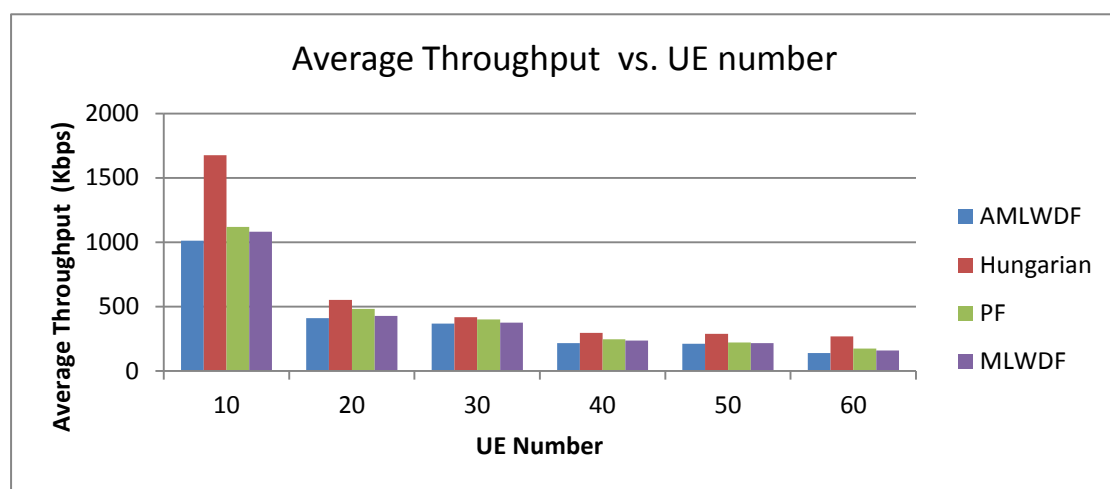


圖 4 - 2 Average Throughput 與 UE Number 關係圖

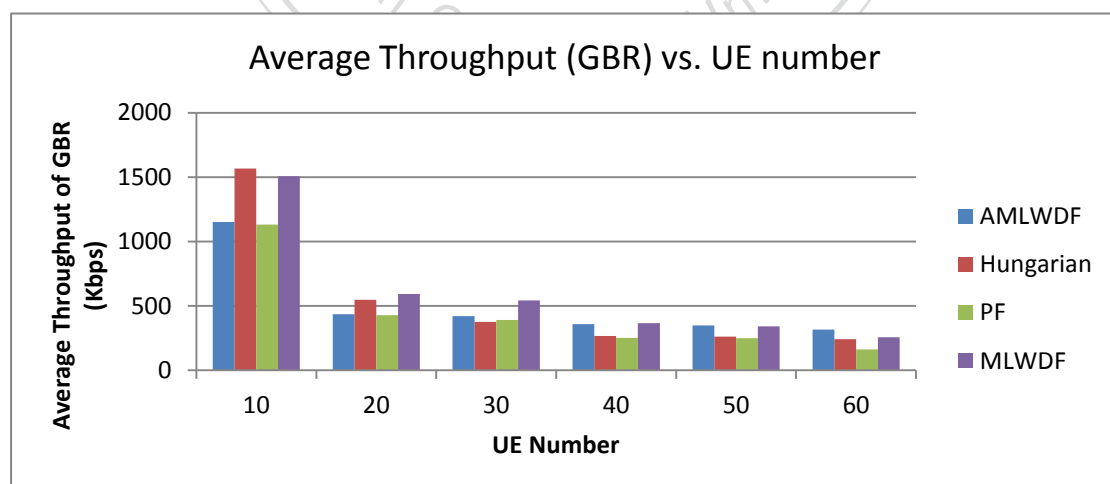


圖 4 - 3 Average Throughput of GBR 與 UE Number 關係圖

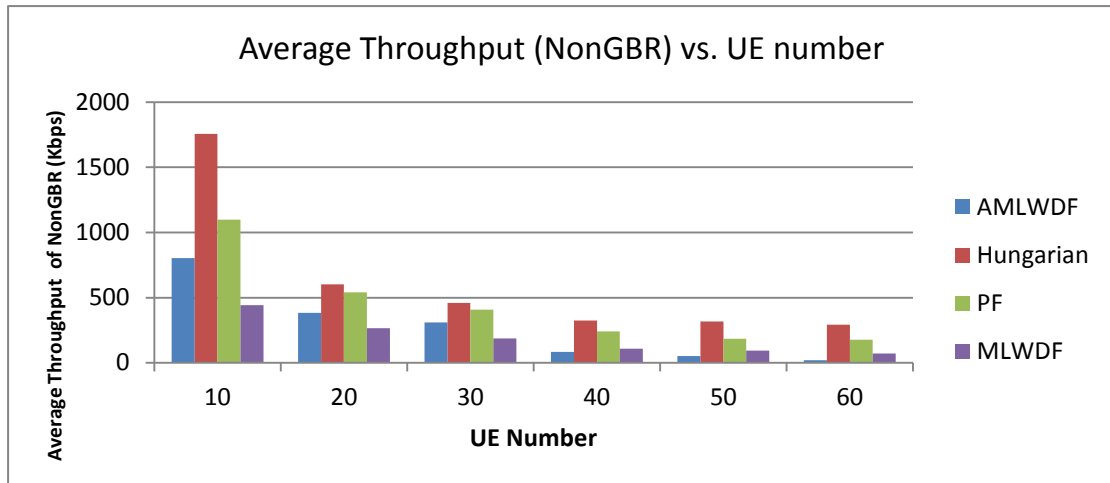


圖 4 - 4 Average Throughput of Non-GBR 與 UE Number 關係圖

Algorithm	Avg. throughput	GBR avg. throughput	non-GBR avg. throughput
Hungarian	-32.65%	-7.05%	-55.93%
PF	-10.95%	15.93%	-37.56%
MLWDF	-5.72%	-15.94%	41.67%

表 4 - 3 AMLWDF 之產能改進比例

在產能方面，Hungarian 演算法無論是在 GBR 或是 non-GBR 皆能擁有最佳的產能，其原因是資源會分配給擁有最佳通道品質的 UE，因資源長期被佔據，所以能一直擁有最高的產能。PF 演算法並無考慮 QoS，所以不會針對不同的資源型態而有所調整，且擁有均衡的機制調整優先權，所以 GBR 與 non-GBR 皆能擁有不錯的產能，但因不限定資源只能分配給最佳通道的 UE，所以產能較 Hungarian 差。AMLWDF 演算法因公平性的機制，導致 UE 量少時分配給 GBR 的比例，少於 UE 量多時分配給 GBR 的比例，以致於 UE 數 10 到 30 人時，產能較 MLWDF 低。但 UE 數超過 40 人以上時，GBR 的比例有漸升的趨勢，所以 GBR 的產能會高於 MLWDF 演算法。相反的，在 non-GBR 面，AMLWDF 在使用者 UE 較少時，會分配給 non-GBR 較多比例的資源，一旦 UE 數量變多，non-GBR 的比例慢慢降低，因此 non-GBR 的產能會從一開始高於 MLWDF 演算法，而人數暴增時，產能則會少於 MLWDF 演算

法。

## 4.2.2 實驗二

本實驗觀察每位使用者 UE 所收到的封包，查看使用者多寡對於不同演算法所造成延遲與抖動的差異。一般來說，較好的網路環境，封包的延遲時間較短，因為封包在網路中不太容易碰撞或是延遲太久而造成遺失。以及好的網路環境，使得每個封包的傳遞時間不致於變動太大。

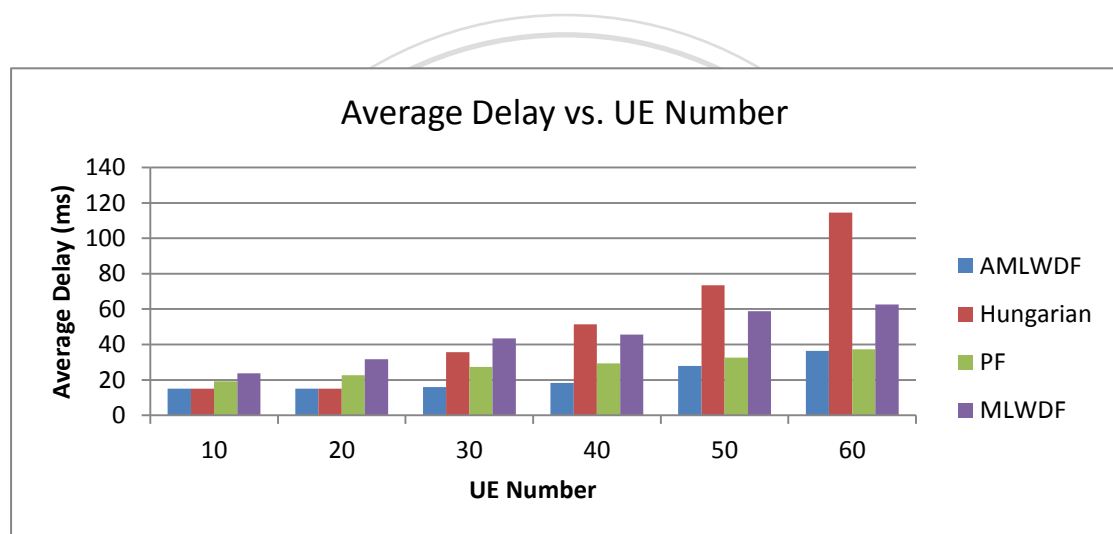


圖 4 - 5 Average Delay Time 與 UE Number 關係圖

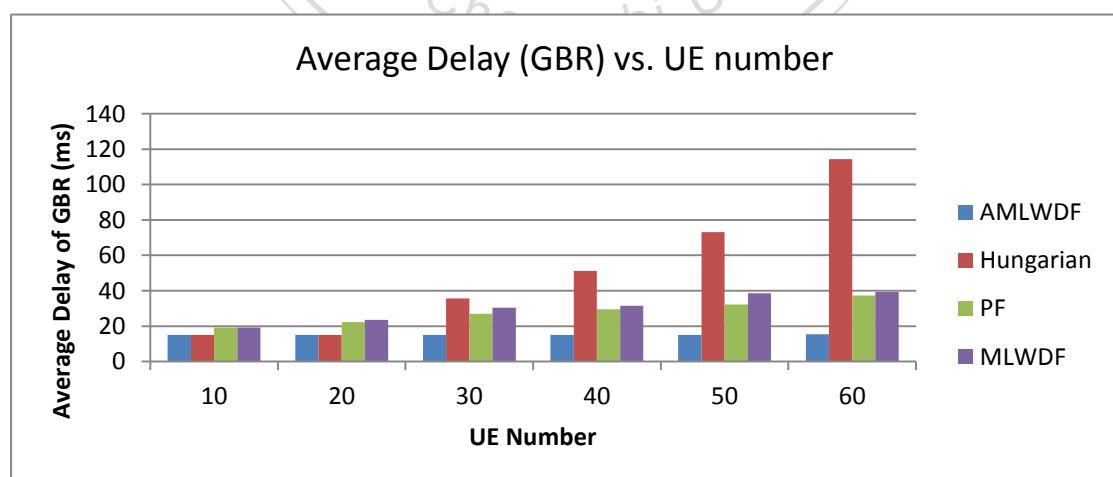


圖 4 - 6 Average Delay Time of GBR 與 UE Number 關係圖

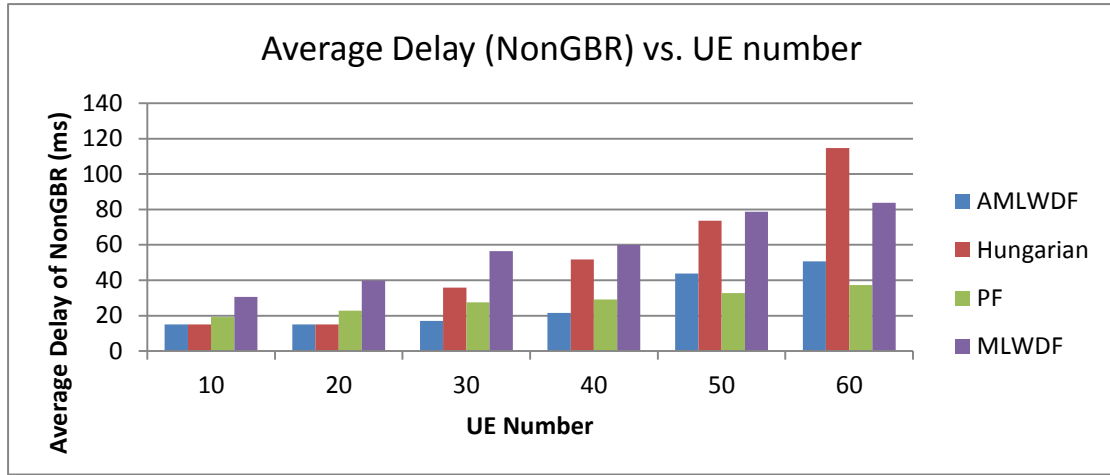


圖 4 - 7 Average Delay Time of Non-GBR 與 UE Number 關係圖

Algorithm	Avg. Delay	GBR avg. delay	Non-GBR avg delay
Hungarian	57.86%	70.26%	46.69%
PF	23.55%	45.99%	3.43%
MLWDF	51.68%	50.35%	53.32%

表 4 - 4 AMLWDF 之延遲時間改進比例

AMLWDF 演算法因為在每一個 slot 時間，都會配置資源給 GBR 與 non-GBR，所以平均延遲時間皆會低於其它的演算法。PF 演算法則因為優先權會有所變動，而無法預測同個 UE 下一次取得資源的時間，所以延遲時間較長。MLWDF 演算法則因為考量 QoS，使得優先權的變動更大，造成平均延遲時間更長。Hungarian 演算法在 UE 少時，能有效分配資源，使得每一個 UE 都能穩定取得資源，但當 UE 數量變多時，會造成部分的 UE 有餓死的狀態，所以延遲時間會暴增。

在考量 QoS 的情況下，AMLWDF 演算法皆能讓 GBR 的資料在有效時間內傳遞出去，所以延遲時間不致於有太大的變化。但 non-GBR 的資料會因為資源塊較少，使得延遲時間隨人數而上升。MLWDF 演算法雖也考量 QoS，但並不能保證在實驗模擬時間內一定能取得資源，所以延遲時間大於 AMLWDF 演算法。



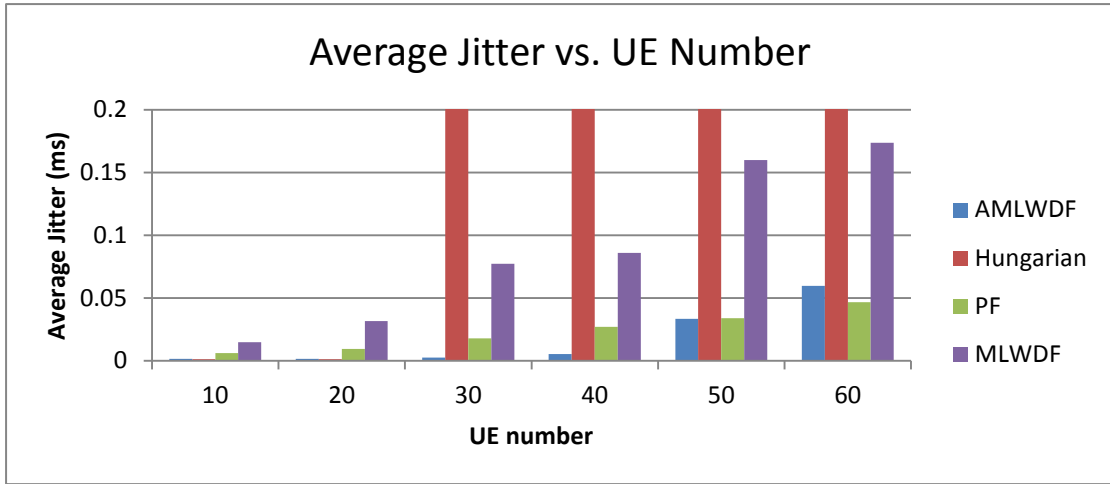


圖 4 - 8 Average Jitter 與 UE Number 關係圖

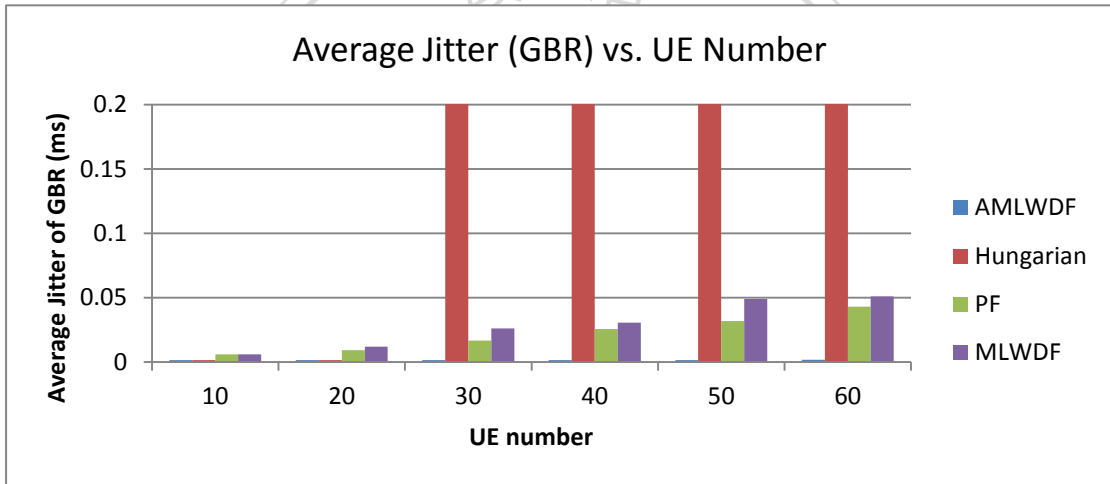


圖 4 - 9 Average Jitter of GBR 與 UE Number 關係圖

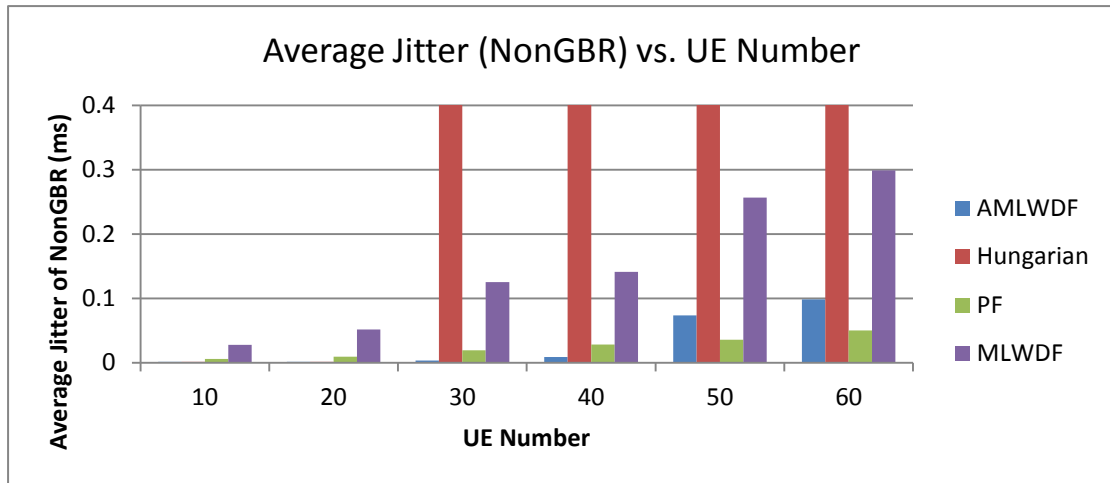


圖 4 - 10 Average Jitter of Non-GBR 與 UE Number 關係圖

Algorithm	Avg. jitter	GBR avg. jitter	non-GBR avg. jitter
Hungarian	99.97%	99.99%	99.95%
PF	26.16%	92.85%	-25.54%
MLWDF	80.84%	94.59%	79.12%

表 4 - 5 AMLWDF 之抖動率改進比例

AMLWDF 演算法因為在固定時間內，每一個 UE 一定能分配到資源，所以延遲時間很穩定，使得抖動率並不明顯。PF 因為常常要調整優先權，所以同個 UE 取得資源的時間並不固定，造成延遲時間經常變動，所以抖動率較大。MLWDF 則是讓 GBR 擁有較高的優先權，如果 non-GBR 長時間未獲得資源，non-GBR 的優先權值在往後的時間會被大幅提高取得資源以避免餓死，當取得資源後，優先權值又會被降低讓 GBR 能擁有較多可用的資源，造成延遲時間變動過大，因此抖動率較高。而 Hungarian 在 UE 數少時能有效地分配資源，一旦人數太多，容易造成大多數的 UE 長時間等待，因此延遲時間極端不平衡，所以抖動率遠高於其它演算法。

## 第五章 結論與未來研究

### 5.1 結論

本論文中，分析了下行鏈路的排程法則，並參考相關文獻，針對 OFDMA 系統在頻寬的分配上做更進一步的研究。

為了有效維持服務品質與提升公平性，我們調整了允入控制及資源分配法則。讓較即時的資料能優先處理，以維持服務品質。但此方法仍有所限制，當使用者人數過多時，雖然仍然能在公平的情況下，保證服務品質中的延遲時間及抖動率，卻造成非即時的資料延遲時間增加且產能下降。高公平性雖然不容易造成等待時間過長，能讓延遲時間不會過長，以及讓每位使用者都能儘量獲得資源，但付出的代價就是總體產能的下降。本論文提出 AMLWDF 演算法，能兼具 Proportional Fair 優先權協調的特色，也考量了 MLWDF 演算法的服務品質保證，以及 Hungarian 的相對公平性。

在延遲時間方面，AMLWDF 優於 Hungarian 演算法 57.86%；優於 MLWDF 演算法 51.68%；優於 Proportional Fair 演算法 23.55%。在抖動率更有顯著的提升，AMLWDF 優於 Hungarian 演算法 99.97%；優於 MLWDF 演算法 80.84%；優於 Proportional Fair 演算法 26.16%。原因在於 AMLWDF 演算法對於即時的封包能保證其高優先權，盡可能優先配置資源塊給擁有即時封包的使用者。反之，非即時封包因優先權較低，分配到資源塊的數量略少於即時性封包，故延遲時間較即時封包的延遲時間長。反觀 Proportional Fair 與 MLWDF 演算法並不能保證每個封包在某段時間內一定能取得資源塊，所以延遲時間與抖動率皆高於 AMLWDF 演算法。

在公平性方面，AMLWDF 演算法在 UE 數量與通道數量相仿時，其公平性最佳。與 Hungarian 相比進步了 25.44%；與 MLWDF 相比進步了 9.03%，兩者都能有明顯進步，但在與 Proportional Fair 的比較，則退步 7.95%。其原因在於：AMLWDF 平衡了 GBR 與 non-GBR 之比重，所以能優於 MLWDF 與 Hungarian，但 Proportional

Fair 不考慮資源型態，故可針對各個資源塊做較佳的調整。

在產能方面，AMLWDF 與 Hungarian 比較時，平均下降了 32.65%，與 MLWDF 比較時，平均下降了 5.72%，與 Proportional Fair 比較時，則平均下降了 10.95%。Hungarian 可視為整體系統產能的上限，Proportional Fair 與 MLWDF 因為會將資源優先分配給通道品質較佳的使用者，所以平均產能相比之下會較高。但 Proportional Fair 未考量資源型態，所以在 GBR 的表現中，AMLWDF 仍高出 15.93%。

## 5.2 未來研究

本篇論文提到公平性與產能很難取得平衡，顧及公平性就必然犧牲產能。在著重公平性的原則下，未來可針對較彈性的延遲時間與產能，求其平衡點，來有效地提高產能。此外本篇論文因著重在排程的設計，尚未考量重傳機制以及封包遺失，未來也可針對封包重傳的次數考量優先權。

## 参考文献

- [1] 3GPP, TS 36.321, Evolved Universal Terrestrial Radio Access (E-UTRA); Medium Access Control (MAC) protocol specification, ver. 8.12.0.
- [2] 3GPP, TS 36.322, Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Link Control (RLC) protocol specification, ver. 8.8.0.
- [3] 3GPP, TS 36.323, Evolved Universal Terrestrial Radio Access (E-UTRA); Packet Data Convergence Protocol (PDCP) specification, ver. 8.6.0.
- [4] 3GPP TS 36.300, Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2, ver. 10.5.0.
- [5] C. Mehlhruer, M. Wrulich, J.C Ikuno, D.Bosanska, and M. Rupp, "Simulating the Long Term Evolution Physical Layer," in *Proc. of the 17th European Signal processing conference (EUSIPCO 2009)*, Glasgow, Scotland, Aug. 2009.
- [6] H. Fattah and H. Alnuweiri, "A Cross-Layer Design for Dynamic Resource Block Allocation in 3G Long Term Evolution System," *Mobile Adhoc and Sensor Systems, IEEE*, pp.929-934, Oct 2009.
- [7] S. Schwarz, C. Mehlhruer and M. Rupp, "Low Complexity Approximate Maximum Throughput Scheduling for LTE," *44th Annual Asilomar conference on Signals, Systems and Computers*, California, USA, 2010.
- [8] JM Holtzman, QUALCOMM Inc., "Asymptotic Analysis of Proportional Fair Algorithm," *Personal, Indoor and Mobile Radio Communications (IEEE PIMRC)*, vol. 2, page(s): F-33-F-37, Aug. 2001.
- [9] R. Basukala, H.A. Mohd Ramli, and K. Sandrasegaran, "Performance Analysis of EXP/PF and M-LWDF in Downlink 3GPP LTE System," in *Proc. of the 1st Asian Himalayas International Conference on Internet (AH-ICI'09)*, pp.1-5, Aug. 2009.
- [10] O. Nwamadi, X. Zhu and A. K. Nandi, "Dynamic Physical Resource Block Allocation Algorithms for Uplink Long Term Evolution," in *proc. 16th European signal processing conference, Eusipco*, Lausanne, Aug. 2011.
- [11] 3GPP TS 23.107, Quality of Service (QoS) Concept and Architecture,

ver. 3.9.0.

- [12] Agilent Technologies, “3GPP Long Term Evolution: System Overview, Product Development and Test Challenges,” pp.81-86, June 2009.
- [13] “The Network Simulator ns-3” (NS-3). Retrieved:  
<http://www.nsnam.org/docs/release/3.13/tutorial/singlehtml/index.html>,  
February 2012.
- [14] “Ns-3 LTE Module Documentation-LENA M2 Documentation” Open Source Project. Retrieved: <http://lena.cttc.es/manual/> , February 2012.

