

國立交通大學

電信工程研究所

碩士論文

LTE 下鏈路非連續接收機制中支援服務品質資源分配演算法

Downlink Resource Allocation under DRX Mechanism with  
QoS Support in LTE Network

研究生：林裕捷

指導教授：李程輝 教授

中華民國一〇二年七月

LTE 下鏈路非連續接收機制中支援服務品質資源分  
配演算法

Downlink Resource Allocation under DRX Mechanism  
with QoS Support in LTE Network

研究生：林裕捷

Student : Yu-Chieh Lin

指導教授：李程輝

Advisor : Tsern-Huei Lee

The logo of National Chiao Tung University is a large blue gear-like circle. Inside the circle, there is a stylized figure of a person holding a book and a torch. The text '國立交通大學' (National Chiao Tung University) is at the top, '電信工程研究所' (Institute of Communication Engineering) is in the middle, and '碩士論文' (Master's Thesis) is at the bottom.

國立交通大學  
電信工程研究所  
碩士論文

A Thesis

Submitted to Institute of Communication Engineering

College of Electrical and Computer Engineering

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Communication Engineering

July 2013

Hsinchu, Taiwan, Republic of China

中華民國一〇二年七月

# LTE 下鏈路非連續接收機制中支援服務品質資源分配演算法

學生：林裕捷

指導教授：李程輝

國立交通大學電信工程研究所碩士班

## 摘要

長程演進技術(LTE)為第三代合作伙伴(3GPP)發展第四代行動通訊過程中形成的無線網路系統。長程演進技術採用正交分頻多工存取 (OFDMA) 做為下行鏈路實體層的傳輸技術，其頻譜多樣性提供很好的網路效能。為了有效地使用無線資源，目前已經發展出許多不同目標的資源分配演算法。由於現今智慧終端普及，對於多媒體應用上網的需求呈現高度成長，保證即時性訊務的服務品質成為封包排程必須要克服的問題。

由於長程演進技術是藉由許多新穎技術構成的網路系統，使用者裝置必須擁有較複雜的計算能力增加了電力消耗。因此第三代合作伙伴針對下行鏈路提出非連續接收模式(DRX)來降低電力耗損，延長使用者裝置的電池壽命。使用者裝置在非連續接收模式中不再永遠保持清醒地監聽下傳控制通道(PDCCH)。當使用者裝置進入睡眠模式時會關閉射頻(RF)電路，停止接收來進演進型基地台(eNB)的訊息，以達到省電的效果。雖然非連續接收模式擁有很好的省電效能，但是其代價便是增加了封包延遲。因此在非連續接收模式中保證即時性訊務的服務品質變得更加困難。傳統的資源分配演算法不需要考慮使用者裝置會進入睡眠模式的問題，因此只在當前的訊框中

滿足訊務的需求。而在非連續接收模式中，資源分配必須考慮使用者裝置可能會在睡眠模式違反服務品質需求，況且現今尚未有一種演算法是針對在非連續接收模式中做資源分配。因此我們提出感知非連續接收模式的資源分配演算法降低非連續接收模式對訊務造成的影響。[1]利用訊務之延遲限制與資料遺失率計算『最小所需頻寬』，將資源分配定義為滿足訊務之最小所需頻寬下，最大化系統吞吐量之最佳化問題。而在非連續接收模式中，計算最小所需頻寬將改為預留服務品質策略，用來推算使用者裝置是否能夠進入睡眠模式。針對那些進入睡眠模式後會違反服務品質需求的使用者裝置，我們會動態地增加啟動非活動計時器(Inactivity Timer)的機會，防止它們進入睡眠模式並繼續保持傳輸。模擬結果顯示我們提出之資源分配演算法不但能夠降低非連續接收模式對系統吞吐量和平均遺失率的影響，還能夠保持良好的省電效能。

**關鍵字：**長程演進技術、非連續接收機制、正交分頻多工、資源分配

# Downlink Resource Allocation under DRX Mechanism with QoS Support in LTE Network

Student : Yu-Chieh Lin

Advisors : Prof. Tsern-Huei Lee

Institute of Communications Engineering  
National Chiao Tung University

## ABSTRACT

The UTRAN LTE specifications are developed by 3GPP. LTE adopts OFDMA as its downlink physical transmission technology which provides high user diversity and frequency diversity that improves network performance. In order to effectively use the limited wireless resources, there are many resource allocation algorithms which have been developed for guarantee of different targets. But nowadays, smartphones are very popular so that the multimedia applications are growing rapidly. Therefore, guaranteeing the QoS requirements of real-time traffics is a challenge for resource allocation algorithms.

LTE is composed of many novel technologies which cause more needs of UEs' computationally complex capability that drains its battery quickly. Thus, 3GPP proposed DRX mode for LTE downlink transmission to extend the UE's battery life. When UEs configure DRX mode, it is no longer for UEs to keep listening to PDCCH all the time. During sleeping period, UEs shutdown RF circuit and stop receiving information from eNB to achieve power saving. Obviously, the trade-off of DRX mode is the increased packet delay. Hence, in DRX mode, to guarantee the QoS requirements of real-time traffics will be more difficult. The conventional resource allocation algorithms try to satisfy the

requirements of the UE only during the basic dynamic scheduling time-domain unit and they do not take the effects of DRX mode into account. Therefore, we propose two DRX-aware resource allocation algorithms which decrease the effects in DRX mode. [1] exploits traffic delay bound and loss probability and running loss probability to calculate minimum requested bandwidth, in which resource allocation is defined as a problem of maximizing system throughput while the minimum requested bandwidth of traffic is guaranteed. In DRX mode, the calculation of the minimum requested bandwidth will adopt QoS provisioning policy to evaluate if the UE will violate its QoS requirements after entering sleeping mode. For those UEs which should not be allowed for entering sleeping mode, we exploit INACTIVITY-TIMER that makes them keep awake and prevents the violation of QoS requirements. Simulation results show that our proposed DRX-aware resource allocation not only decreases the harms of system throughput and the packet loss rate which are caused by DRX mode but also achieves good power saving performance.

**Keywords:** LTE, DRX, OFDMA, Resource Allocation

# 誌 謝

首先我要感謝我的指導教授—李程輝博士。在我的研究生涯遭遇困境時對我伸出援手，讓我可以他的指導下專心地研究。他總是不斷地提點我們做研究的精神以及過程中的每個環節，讓我們得以在漫漫研究道路中走在正確的方向。再來要感謝與我們一同參與會議的郭耀文教授和高榮鴻教授。每次會議三位教授一同討論和提出建議總是讓我受益匪淺，才能順利地提出構想並且實現出來。另外還要感謝實驗室的梓洋學長和承潔學姐，不論何時都願意放下手邊工作，盡可能地給予協助以解決我的問題。還有實驗室幽默的同學們，有時候沒有重點卻有笑點的對話，總能一解工作環境中的壓力，增添一份不一樣的色彩。

再來要感謝我的摯友們，在我研究的最後一年中遭遇人生低潮時一直相伴在左右，我才能縮短那段步履蹣跚的道路，提早回到做研究的正軌上並如期完成這份論文。還有一直支持我攻讀碩士學位的家人們，尤其是我的父母，讓我能選擇走上這段學習與歷練同等精彩的道路，並且提供安穩的環境讓我能專注在學術研究上。最後再一次感謝所有在這裡相遇、相伴的人們帶來的歡笑與淚水，它們交織出來的回憶以及帶來的成長便是這段期間最好的收穫。

2013/07 林裕捷

# 目 錄

摘 要 .....	i
ABSTRACT .....	iii
誌 謝 .....	v
圖 目 錄 .....	vii
表 目 錄 .....	viii
<b>第一章. 簡介 .....</b>	<b>1</b>
1.1. 研究動機 .....	1
1.2. 相關研究 .....	3
1.3. 論文架構 .....	4
<b>第二章. 系統模型 .....</b>	<b>5</b>
2.1. 系統環境 .....	5
2.2. 長程演進技術資源架構 .....	8
2.3. 傳統的資源分配機制 .....	10
2.4. 非連續接收機制 .....	14
<b>第三章. 在非連續接收模式中做資源分配 .....</b>	<b>16</b>
3.1. 問題描述 .....	16
3.2. 感知型非連續接收模式之資源分配 .....	17
3.2.1. 演算法一 .....	18
3.2.2. 演算法二 .....	20
<b>第四章. 模擬 .....</b>	<b>24</b>
4.1. 模擬環境 .....	24
4.2. 非連續接收模式對資源分配的影響 .....	26
4.3. 效能評估 .....	30
<b>第五章. 結論 .....</b>	<b>36</b>
<b>參考文獻 .....</b>	<b>37</b>

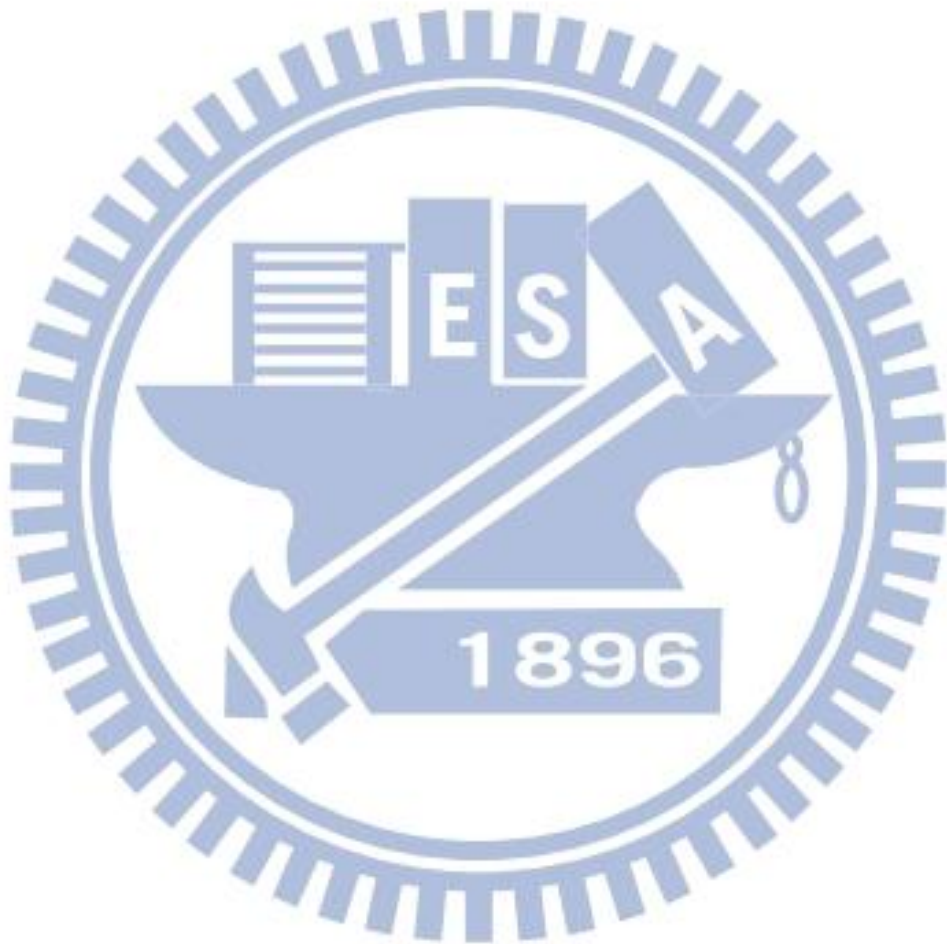


# 圖 目 錄

圖 1. 單一細胞的長程演進系統模型 .....	5
圖 2. 使用者佇列模型 .....	7
圖 3. 長程演進技術訊框架構(省略其他控制通道) .....	9
圖 4. 長程演進技術資源分配架構圖(省略控制) .....	10
圖 5. 長程演進技術非連續接收模式運作架構 .....	15
圖 6. 方法[1]資源分配架構圖 .....	17
圖 7. LTE-SIM 訊務管理架構圖 .....	24
圖 8. LTE-SIM 網路管理架構圖 .....	25
圖 9. 第一種情境中比較系統吞吐量 .....	28
圖 10. 第一種情境中比較平均遺失率 .....	28
圖 11. 第二種情境中不同的清醒區間定時器與非連續活動參數定時器對傳統資源分配演算法之影響 .....	30
圖 12. 第二種情境中比較系統吞吐量 .....	31
圖 13. 第二種情境中比較平均遺失率 .....	32
圖 14. 第二種情境中比較省電效能 .....	32
圖 15. 演算法二與傳統資源分配演算法在非連續接收模式中比較系統吞吐量 .....	33
圖 16. 演算法二與傳統資源分配演算法在非連續接收模式中比較平均遺失率 .....	34
圖 17. 演算法二與傳統資源分配演算法在非連續接收模式中比較省電效率 .....	35

# 表 目 錄

表 1. 模擬環境參數、訊務特性及服務品質需求設定.....	26
表 2. 非連續接收模式參數設定.....	30



# 第一章.

## 簡介

---

### 1.1. 研究動機

由於現今智慧型終端設備的普及，多媒體影音與行動上網的需求呈現高幅度的成長。因此第三代合作伙伴計劃(3rd Generation Partnership Project, 3GPP)提出之第四代行動通訊技術(4th Generation Mobile Telecommunication Standards)目的就是要提供更高速度、系統容量更大的無線網路。長程演進技術(Long Term Evolution, LTE)便是發展第四代行動通訊過程中形成的系統，也是目前營運商佈建的焦點。

長程演進技術的網路架構主要分為無線部份(Evolved Universal Terrestrial Radio Access Network, E-UTRAN)與核心網路部份(Evolved Packet Core, EPC)，並且在核心網路與使用者之間透過演進型基地台(Evolved Node B, eNB)執行所有無線電資源管理(Radio Resource Management, RRM)。無線部份的下行鏈路採用正交分頻多工存取(Orthogonal Frequency Division Multiple Access, OFDMA)技術將頻譜切割成許多子載波(sub-carrier)。根據使用者的需求、目前系統的負載量或是系統的配置，將會分配部份可使用的子載波給使用者傳送資料。正交分頻多工存取技術提升頻譜的利用效率以及能夠根據通道品質的條件選擇不同的通道傳輸增加抵抗通道衰弱能力，因此擁有更好的傳輸速率、使用者多樣性(user diversity)。但是在有限的頻譜資源中，封包排程便是無線資源管理其中一個重要的功能，它必須有效率地使用頻譜資源，選擇合適的使用者傳送封包，並且盡可能地滿足使用者對服務品質(Quality of Service, QoS)的要求。像是多媒體應用中的視頻串流(Video Streaming)、網路電話(Voice-over-IP, VoIP)...等等這些即時性訊務(Real-time Traffic, RT)，它們對服務品質的需求可區分為延遲敏感(delay sensitive)和遺

失敏感(loss sensitive)，保證這些需求是封包排程的一項挑戰，目前也已經存在許多針對此主題的相關研究。

由於長程演進技術是由多項新穎技術組成的網路系統以達到高速的網路速率，使用者裝置(User Equipment, UE)必須具備複雜的運算能力因而增加電力消耗，縮短使用者裝置的使用時間反而成為其致命傷。所以第三代合作伙伴計劃針對下行鏈路提出一個解決方案—非連續接收(Discontinuous Reception, DRX)[2]機制來延長使用者裝置的電力壽命。在非連續接收模式中，使用者裝置會週期性的清醒一段時間監聽實體下傳控制通道(Physical Downlink Control Channel, PDCCH)，檢查是否有資料需要接收。一旦使用者裝置被通知接收封包的指令，便會延長其清醒時間繼續接收緩衝器(buffer)裡的資料。若使用者裝置在清醒時間都未收到接收封包指令則會進入睡眠模式，此時所有的下行鏈路資料都將儲存在演進型基地台的緩衝器中直到下一次使用者裝置回到清醒模式才能繼續傳輸。雖然非連續接收機制能有效降低使用者裝置的電力消耗，但其代價便是增加封包延遲、遺失率還有降低系統的吞吐量(throughput)。

從[3]可看出非連續接收機制的省電效率除了與參數的配置有關，也被使用者數量所影響。原因在於當使用者裝置數量增加使得競爭資源越趨激烈，使用者裝置需要更多的傳輸時間才能消化完緩衝器中的資料，因此增加電力的消耗。另外，若是缺乏目的地延長使用者裝置的清醒時間，不只降低省電效能更對提升系統吞吐量沒有幫助。傳統的資源分配演算法只會在分配資源的基本時間單位內盡可能地滿足使用者裝置的需求，並未考慮在非連續接收模式中使用者裝置會有一段睡眠期間對訊務造成的影響。因此我們認為需要針對非連續接收模式設計感知型的資源分配演算法，才能夠讓使用者裝置在節省電力的前提下降低系統效能和服務品質受到的影響。方法[1]利用訊務之延遲限制與資料遺失率計算滿足遺失率容忍度的『最小所需頻寬』，將資源分配問題定義為滿足訊務的最小所需頻寬下，最大化系統吞吐量之最佳化問題。我們可以將服務品質預留策略(QoS provisioning policy)套用在計算最小所需頻寬中，用來判斷使用者裝置在進入睡眠模式後

是否會違反其服務品質需求。若使用者裝置在進入睡眠模式後會違反服務品質需求則提升其延長清醒時間的機會增加參與資源分配的時間，直到不會在進入睡眠模式後違反服務品質需求為止。模擬結果將會顯示，我們提出的感知非連續接收模式之資源分配演算法將會降低系統吞吐量和封包遺失率受到的影響還能保持很好的省電效率。

## 1.2. 相關研究

長程演進技術採用正交分頻多工存取[4]做為其下行鏈路實體層的傳輸技術，將傳輸資源定義於頻域(frequency domain)和時域(time domain)組成的二維空間，以達到使用者多樣性(user diversity)和頻譜多樣性(frequency diversity)。封包排程除了要保持系統有好的吞吐量，還要考慮分配的公平性及使用者對服務品質的需求。針對這些目標，以下將簡介幾個本篇論文參考的資源分配演算法。

最常見的網路系統效能標的便是整體系統吞吐量。最大化速率演算法(max-rate)[5]以最大化系統吞吐量為目標，每次做資源分配時皆由通道品質最好的使用者裝置開始分配直到資源用完為止。這種方式雖然會得到最好的系統吞吐量，卻造成通道品質不好的使用者裝置很難競爭到資源，除非其通道品質有所提升才能改善挨餓的情形。

等比例公平演算法(Proportional Fairness, PF)[6]以平衡系統吞吐量及資源分配公平性為目標。但其原生是從分碼多工存取(Code Division Multiple Access, CDMA)系統中探討非即時性訊務(Non-real-time Traffic, NRT)之資源分配而來，並沒有保證任何服務品質。因此最大延遲優先演算法(Modified Largest Delay First, MLWDF)[7]改良了等比例公平演算法，分配資源的過程中將訊務流遺失率需求、延遲限制以及佇列中最前線(head-of-line)封包延遲時間一併納入計算，因此能支援不同服務品質需求的即時性訊務。顯然地，在非連續接收模式中封包延遲的情形將會更加嚴重，若只考慮最前線的封包延遲對保證服務品質不是最佳的。

方法[1]假設其服務品質需求為延遲限制與資料遺失率，將資源分配分成兩個回合。第一回合先針對即時性訊務，根據延遲限制與資料遺失率計算最小所需頻寬，然後將資源分配定義為滿足即時性訊務之最小所需頻寬下，最大化系統吞吐量之問題。若即時性訊務流之最小所需頻寬皆被滿足且資源尚未耗盡則會進入第二回合，同時考慮非即時性訊務與即時性訊務剩餘的資料量，以最大化速率演算法分配其餘資源。雖然此演算法能同時處理即時性與非即時性訊務的資源分配，在滿足服務品質需求的前提下較其他資源分配演算法提供更好的系統吞吐量。但是未考慮非連續接收模式所造成之影響，將會降低保障即時性訊務服務品質的能力。

### 1.3. 論文架構

接下來的章節中，第二章會先說明本篇論文使用的系統環境。接著會詳細的說明長程演進技術下行鏈路的訊框以及資源分配架構。再來會深入的說明前一節相關的資源分配演算法的每一個步驟。最後是長程演進技術在非連續接收機制底下的運作模式。

第三章一開始會完整的探討在非連續接收機制底下做資源分配將會碰上的問題。接著我們會提出兩個感知型非連續接收模式的資源分配演算法，盡可能地降低非連續接收模式對網路效能造成的影響。

第四章會先說明模擬環境的參數設定以及模擬器的基本架構，我們將在一個系統層級的模擬器下比較本篇論文的實驗結果。接著我們會先模擬傳統資源分配演算法之效能以更瞭解其原始特性。再來我們模擬幾種不同的非連續接收參數配置，分析資源分配演算法受到的影響以選出一組合適的非連續接收參數設定。最後以模擬結果比較傳統的資源分配演算法與我們提出之資源分配演算法在非連續接收模式中的效能。比較標的將包含系統的吞吐量、整體的資料遺失率和省電效率。

最後，本篇論文將在第五章中做一個完整的總結。

## 第二章.

### 系統模型

#### 2.1. 系統環境

本篇論文中，我們的環境為一個演進型基地台連接多個使用者的單一細胞 (single-cell) 分頻多工存取之長程演進技術系統，並且只考慮下行鏈路的資料傳輸(如圖 1)。使用者在每一次通話(session)期間只會使用一條即時性或者非即時性其中一種型態的訊務。其中  $\Gamma_{RT}$  和  $\Gamma_{NRT}$  分別表示使用即時性和非即時性訊務的使用者集合。接著  $\Gamma = \Gamma_{RT} \cup \Gamma_{NRT}$  表示所有使用者的集合。

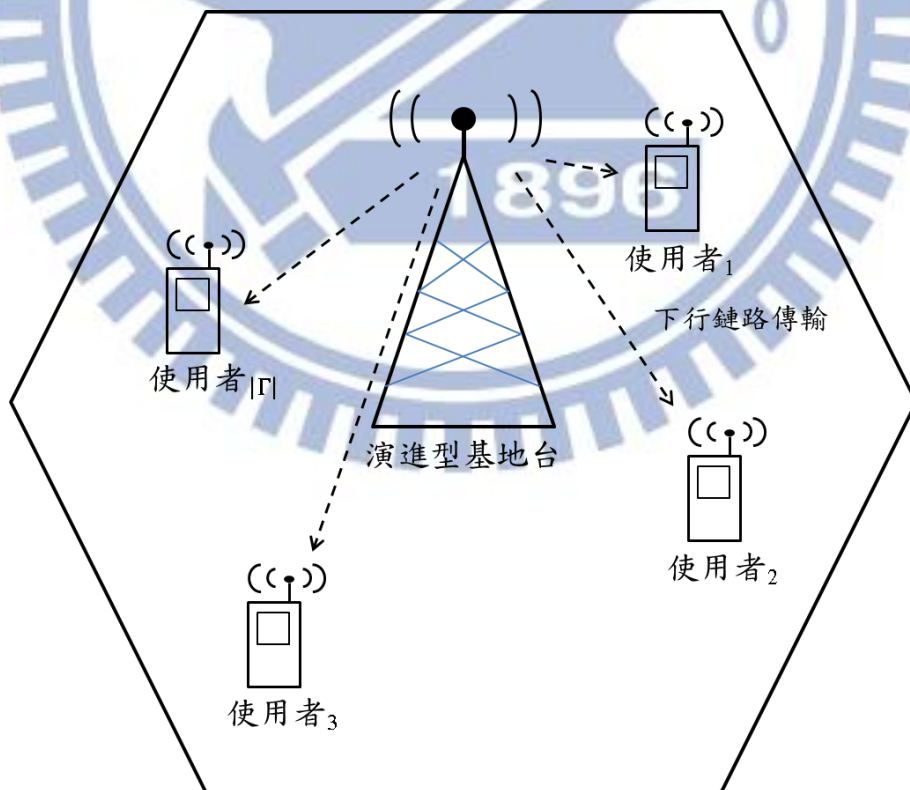


圖 1. 單一細胞的長程演進系統模型

分頻多工存取技術中，時間軸被分割成區間為 $T_{frame}$ 的訊框，每個訊框再切分成許多區間為 $T_{subframe}$ 的子訊框，每個子訊框由兩個時槽(time-slot)所組成。頻域上，一個訊框則被切分成 $M$ 個子頻道。在長程演進技術中，動態排程(dynamic scheduling)的時間間隔為 $T_{subframe}$ [8]，演進型基地台會分配一條佇列 $Queue_n$ 給使用者 $n$ (如圖 2)存放正在使用的訊務資料。針對使用即時性訊務的使用者，我們將其對服務品質的要求定義為延遲上限(delay bound)和遺失率容忍度(loss-rate tolerance)，分別用 $D_n \cdot T_{subframe}$ 和 $P_n$ 來表示，並且佇列可再切分成許多子佇列 $Queue_n^d$   $1 \leq d \leq D_n$ ，每個子佇列中的資料還能夠存活於 $Queue_n$ 中 $d \cdot T_{subframe}$ 的時間長度，資料延遲若超過延遲限制將會被丟棄。而 $Q_n[t]$ 和 $Q_n^d[t]$ 則表示在 $t^{th}$ 子訊框的開始時 $Queue_n$ 和 $Queue_n^d$ 的長度。





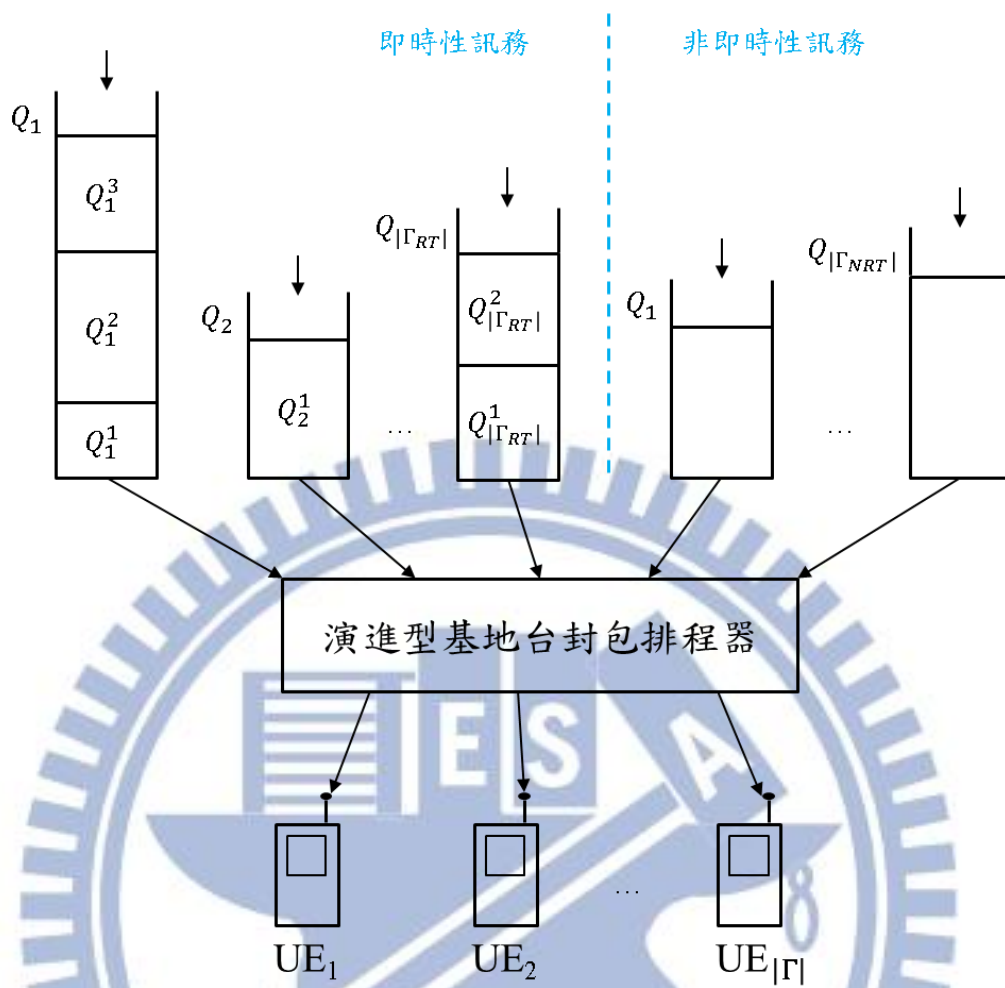


圖 2. 使用者佇列模型

## 2.2. 長程演進技術資源架構

在長程演進技術中，演進型基地台負責執行無線電資源管理。無線電資源管理可以由上到下分為三層。協定第三層(Layer 3)包含資料匯聚通訊協定(Packet Data Convergence Protocol, PDCP)對上層封包進行資料保密與壓縮；無線資源控制(Radio Resource Control, RRC)管理每一條訊務的資料，根據訊務不同的服務品質需求以承載(bearer)的方式對進行分類。協定第二層(Layer 2)包括無線電連結控制(Radio Link Control, RLC)負責資料的重傳以及對媒體存取控制(Media Access Control, MAC)層進行訊框切割和重組；媒體存取控制主要負責無線電資源分配，考量系統使用者的需求、目前系統承載...等等因素有效地使用頻譜資源。協定第一層(Layer 1)為實體層(Physical Layer, PHY)，負責通道品質指示管理(Channel Quality Indicator Manager)以及在無線通道上進行資料傳送。無線電資源管理可分為半動態(semi-dynamic)機制和動態(dynamic)機制[4]。協定第三層屬於半動態機制，也就是在承載被建立時或是每隔一段區間才會被執行。而協定第二層與協定第一層則屬於動態機制，被定義為每個傳送時間間隔(Transmission Time Interval, TTI)都要執行一次。在長程演進技術中，傳送時間間隔的長度為 $T_{subframe}$ 。

長程演進技術下行鏈路的無線通道是基於正交分頻多工存取技術進行傳輸，無線電資源是由時域和頻域的二維空間定義而成。在時域上，訊框的長度為十毫秒。圖 3 為訊框與子訊框在時域中分割的方式，每個訊框中由十個長度為一毫秒的子訊框組成，子訊框的長度等同於一個傳送時間間隔。子訊框中的資源被切割成兩個時槽，每個時槽的長度為零點五毫秒。傳送時間間隔中包含十四個正交分頻多重符號(OFDM symbol)，其中第一到第三的正交分頻多重符號稱作實體下傳控制通道負責傳送資源分配訊息。而訊務資料將會被放在其餘的正交分頻多重符號中傳送，稱作下傳共享通道(Physical Downlink Shared Channel, PDSCH)。在頻域上，頻譜可以被切割成許多相同寬度的子通道，每一個子通道包含十二個寬度為一百八十千赫茲的子載波。在長程演進技術實體層中，時域上一個時槽與頻域上一個子通道構成最小的資源單位稱作資源區塊(resource block, RB)

(如圖 4)。根據不同的系統頻寬，一個子訊框中將會擁有不同數量的資源區塊。另外，時域上連續兩個資源區塊組成一個資源區塊對(RB pair)，封包排程器每次在子訊框中做資源分配時以資源區塊對為最小單位分配給使用者[9]。當使用者獲得某個資源區塊對時，表示該使用者佔用子訊框中該資源區塊對所在的整個子頻道，一個子頻道只會由一個使用者佔用。

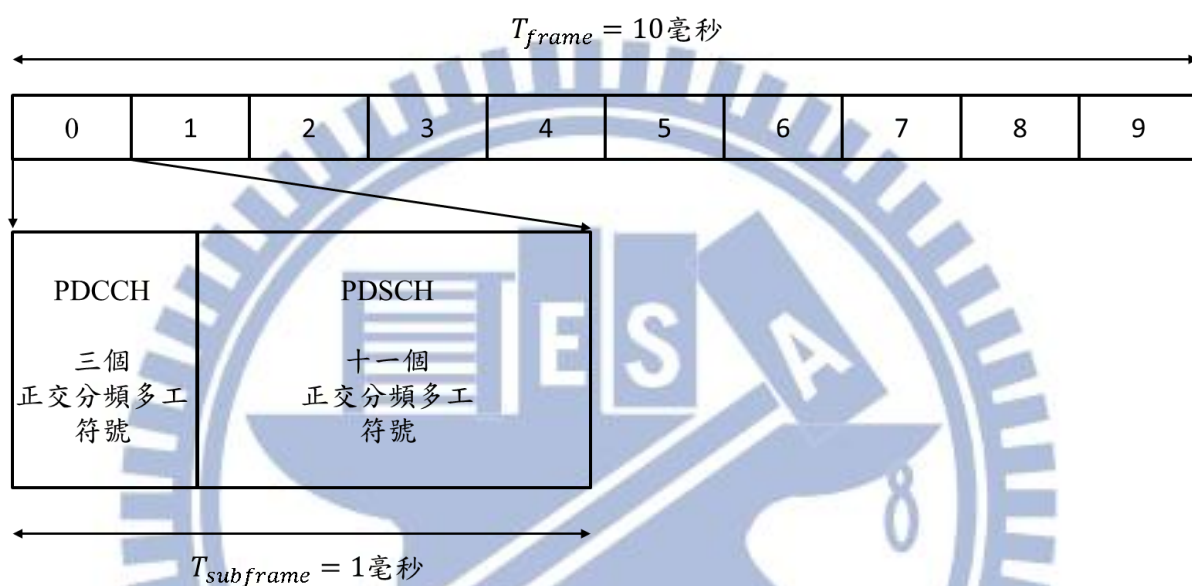


圖 3. 長程演進技術訊框架構(省略其他控制通道)

實體下傳控制通道的訊息中，除了有子訊框中資源分配的結果還包含使用者可以使用的調變及編碼方式(Modulation and Coding Scheme, MCS)。使用者裝置必須探測通道品質並且週期性地回報通道品質指示(Channel Quality Indicator, CQI)給演進型基地台進行排程決定和連結調整。演進型基地台會根據使用者裝置的通道品質決定可以使用的調變及編碼方式，並將結果放置於實體下傳控制通道中通知使用者裝置。而為了提升可靠性，實體下傳控制通道一律採用四相移鍵控(QPSK)的調變方式進行傳輸[4]。當使用者解碼實體下傳控制通道獲得某個數量的資源區塊對後，使用的調變及編碼方式將會決定在此傳送時間間隔中可以傳送的資料數量。

在我們的系統中，將會採用適應性調變編碼(Adaptive Modulation and Coding, AMC)

根據使用者裝置的通道品質選擇適合的調變及編碼方式，以提升傳輸品質的可靠性。使用者裝置採用理想的方式回報通道品質指示不考慮回報延遲，因此每隔一個 $T_{subframe}$ 的時間便傳送通道品質指示給演進型基地台進行連結調整。使用者裝置根據可以使用的調變及編碼方式便可計算出在每個子通道上可達到的瞬時速率(instantaneous rate)。

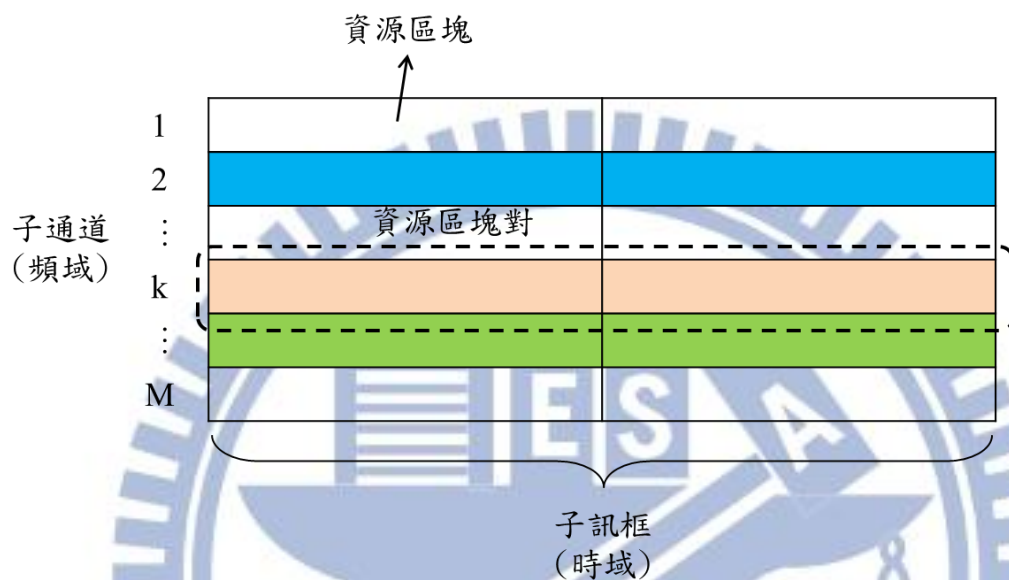


圖 4. 長程演進技術資源分配架構圖(省略控制)

### 2.3. 傳統的資源分配機制

資源分配演算法會在每一個子訊框開始時執行資源分配，我們將此子訊框標示為 $t^{th}$ 子訊框。 $r_{n,m}[t]$ 表示在 $t^{th}$ 子訊框開始時，使用者 $n$ 在子通道 $m$ 可以達到的瞬時速率。 $\bar{r}_n[t]$ 表示統計到 $t^{th}$ 子訊框，使用者 $n$ 的平均傳送速率。 $u_{n,m}$ 表示使用者 $n$ 在子頻道 $m$ 上的度量衡。最後用 $R_n[t]$ 表示使用者 $n$ 在 $t^{th}$ 子訊框中可傳送的資源總量。

#### A. 等比例公平演算法(PF)

由於等比例公平演算法是從分碼多工存取系統中發展而來，資源分配的基本單位只包含時域的概念，因此在分配資源時只要在該時間點挑出一個度量衡(metric)最大的使用

者裝置。但在分頻多工存取系統中，資源單位是由時域與頻域的二維空間所組成，我們可以將子訊框裡每個資源區塊獨立地做分配[10]，每次將資源區塊對分配給在該子頻道上度量衡最大的使用者裝置，接著更新該使用者裝置的平均傳送速率，再進行下一個資源區塊對的分配，直到耗盡所有資源區塊對為止。以下是在一個子訊框中，等比例公平演算法的完整步驟：

- 1)  $u_{n^*, m^*} = \max_{n \in \Gamma, 1 \leq m \leq M} \{r_{n, m}[t] / \bar{r}_n[t]\}$
- 2)  $Q_{n^*}[t] = (Q_{n^*}[t] - r_{n^*, m^*}[t])^+$  and update  $R_{n^*}[t]$  and  $\bar{r}_{n^*}[t]$
- 3) Replace  $u_{n, m} = 0 \forall n \in \Gamma$   
If  $Q_{n^*}[t] = 0$  then replace  $u_{n^*, m} = 0 \forall m$

這三步驟會重覆執行直到  $u_{n, m} = 0 \forall n \forall m$  為止。

雖然等比例公平演算法提供很好的資源分配公平性，但是沒有考慮即時性訊務的延遲限制或是遺失率容忍度，無法為使用即時性訊務流的使用者提供任何的服務品質保證。當啟動非連續接收模式後封包延遲將會更加嚴重，服務品質會受到更大的損害。

#### B. 最大延遲優先演算法(M-LWDF)

最大延遲優先演算法的目標是保證使用者  $n \in \Gamma_{RT}$  的遺失率  $P(W_n > D_n) < P_n$ ，度量衡  $u_{n, m} = a_n \cdot W_n[t] \cdot r_{n, m}[t] / \bar{r}_n[t]$ 。其中  $W_n[t]$  為  $Queue_n$  在  $t^{th}$  子訊框開始時的最前線封包延遲， $a_n$  是一個為正的常數，其中  $a_n = -\log P_n / D_n$  已經被證明是一個很好的參數設定值。整個演算法執行的步驟如下：

- 1)  $u_{n^*, m^*} = \max_{n \in \Gamma, 1 \leq m \leq M} \{a_n \cdot r_{n, m}[t] / \bar{r}_n[t]\}$
- 2)  $Q_{n^*}[t] = (Q_{n^*}[t] - r_{n^*, m^*}[t])^+$  and update  $R_{n^*}[t]$  and  $\bar{r}_{n^*}[t]$

3) Replace  $u_{n, m^*} = 0 \forall n \in \Gamma$

If  $Q_n^*[t] = 0$  then replace  $u_{n^*, m} = 0 \forall m$

這三步驟會重覆執行直到  $u_{n, m} = 0 \forall n \forall m$  為止。

根據最大延遲優先演算法的度量衡計算方式，使用者的優先權會受到延遲限制、遺失率容忍度和佇列最前線的封包延遲影響，藉以保證即時性訊務的服務品質。但是並沒有針對非即時性訊務做參數設定的討論，因此我們令非即時性訊務的  $a_n = 1$ 。很顯然地，在非連續接收模式中分配資源時只考慮最前線的封包延遲對於保證服務品質並不是最佳的。

### C. 方法[1]

方法[1]的目標是保證即時性訊務的服務品質前提下，最大化系統的吞吐量。在  $t^{th}$  子訊框開始時將資源分配分成兩個回合進行：第一回合中只考慮使用者  $n \in \Gamma_{RT}$  定義其在  $t^{th}$  子訊框結束時的當前遺失率(running loss rate)

$$P_n[t] = \frac{L_n[t]}{S_n[t] + L_n[t]} \quad (1)$$

其中  $S_n[t]$  和  $L_n[t]$  分別表示在  $t^{th}$  子訊框結束時，使用者  $n$  累積傳送和遺失的資料量。依照當前遺失率的定義，根據使用者  $n$  的延遲限制、佇列長度和在  $t^{th}$  子訊框中使用者  $n$  可能得到的資源數量，使用者  $n$  在  $t^{th}$  子訊框結束時的當前遺失率可以修改成

$$P_n[t] = \frac{S_n[t-1] + L_n[t-1] + (Q_n^1[t] - R_n[t])^+}{S_n[t-1] + L_n[t-1] + \max(Q_n^1[t], R_n[t])}, \quad 0 \leq R_n[t] \leq Q_n[t] \quad (2)$$

接著根據遺失率容忍度，令  $P_n^*[t] = P_n$  計算出在  $t^{th}$  子訊框中滿足遺失率容忍度所需要最小所需頻寬  $R_n^*[t]$ ，若  $R_n^*[t] > 0$  則將使用者  $n$  加入集合  $\emptyset_1$ ，接著將資源分配定義為滿足最小頻寬下最大化系統吞吐量將之問題。第二回合考慮使用者  $n \in \Gamma$ ，以最大速率演算法分配剩餘資源。以下是此演算法在兩個回合中執行的完整步驟：

### First Round

$$1) \quad u_{n^*, m^*} = \max_{n \in \phi_1, 1 \leq m \leq M} \{ r_{n, m}[t] \}$$

$$2) \quad R_{n^*}^*[t] = \left( R_{n^*}^*[t] - r_{n^*, m^*}[t] \right)^+$$

*If  $R_{n^*}^*[t] = 0$  then remove user  $n^*$  from  $\phi_1$*

$$Q_{n^*}^*[t] = \left( Q_{n^*}^*[t] - r_{n^*, m^*}[t] \right)^+ \text{ and update } R_{n^*}^*[t]$$

$$3) \quad \text{Replace } u_{n, m^*} = 0 \quad \forall n \in \Gamma$$

*if  $Q_{n^*}^*[t] = 0$  then replace  $u_{n^*, m} = 0 \quad \forall m$ ,*

這三步驟會重覆執行，若  $R_n^*[t] = 0 \quad \forall n \in \Gamma_{RT}$  且  $u_{n, m} \neq 0 \quad \exists n \exists m$  則進入第二回合。否則  $u_{n, m} = 0 \quad \forall n \forall m$  則結束此演算法。

### Second Round

$$1) \quad u_{n^*, m^*} = \max_{n \in \Gamma, 1 \leq m \leq M} \{ r_{n, m}[t] \}$$

$$2) \quad Q_{n^*}^*[t] = \left( Q_{n^*}^*[t] - r_{n^*, m^*}[t] \right)^+ \text{ and update } R_{n^*}^*[t]$$

$$3) \quad \text{Replace } u_{n, m^*} = 0 \quad \forall n \in \Gamma$$

*If  $Q_{n^*}^*[t] = 0$  then replace  $u_{n^*, m} = 0 \quad \forall m$*

這三個步驟會重覆執行直到  $u_{n^*, m} = 0 \quad \forall n \forall m$  為止。

此演算法首先讓即時性訊務獲得滿足遺失率容忍度的最小所需頻寬，接著再以最大化系統吞吐量的方式分配資源，因此能夠同時支援即時性與非即時性的訊務流。但是在非連續接收模式中，使用者裝置會在清醒與睡眠模式之間切換，若計算即時性訊務的最小所需頻寬時只考慮當前子訊框中會遺失的資料量，一旦使用者裝置進入睡眠模式時極有可能就會違反其服務品質需求。

## 2.4. 非連續接收機制

在長程演進技術中，使用者裝置可以在RRC\_CONNECTED的狀態啟動下行鏈路非連續接收模式[11]，並且會在清醒和睡眠模式之間做切換(如圖 5)(高起的部份為清醒時間)。使用者裝置清醒時會監聽實體下傳控制通道，判斷是否有資料需要接收；睡眠模式時，使用者裝置會關閉射頻電路，停止接收封包以達到節省電力之效果。演進型基地台可以針對每一個使用者裝置配置一組非連續接收模式的參數，以下為非連續接收模式各參數的說明[12]：

- 清醒區間定時器(On-duration Timer,  $t_{ON}$ )說明使用者裝置在每一個非連續接收週期開始時，需要保持連續清醒的子訊框個數，以讀取實體下傳控制通道的訊息。可設定的長度為1, 2, 3, 4, 5, 6, 8, 10, 20, 30, 40, 50, 60, 80, 100, 200個子訊框。
- 非活動定時器(Inactivity Timer,  $t_I$ )說明使用者裝置需要繼續保持連續清醒的子訊框個數。每當使用者裝置收到實體下傳控制通道通知有資料需要接收後，非活動定時器將會被重置。可設定的長度為1, 2, 3, 4, 5, 6, 8, 10, 20, 30, 40, 50, 60, ...個子訊框。
- 短非連續接收週期(Short DRX Cycle,  $T_{C_S}$ )為每一個短非連續接收週期的長度。短非連續接收週期為選擇性使用。可設定的長度為 $2^n, n = 1, \dots, 9$ 和 $5 \cdot 2^n, n = 1, \dots, 6$ 個子訊框。
- 短非連續接收週期定時器(Short DRX Cycle Timer,  $n_S$ )為使用者裝置連續進入 $n_S$ 個短非連續接收週期後則進入長非連續接收週期。每當使用者裝置收到實體下傳控制通道通知有資料需要接收後，短非連續接收週期定時器將會被重置。
- 長非連續接收週期(Long DRX Cycle,  $T_{C_L}$ )為每一個長非連續接收週期的長度。當 $n_S = 0$ 後才會進入長非連續接收週期。可設定的長度為 $2^n, n = 1, \dots, 9$ 和 $5 \cdot 2^n, n = 1, \dots, 9$ 個子訊框。



- 非連續接收偏移值(DRX Offset,  $T_{Offset}$ )用來獲得非連續接收週期的開始子訊框位置。當 $(SFN \cdot 10 + n)\%T_C = T_{Offset}$ 時便是非連續接收週期的第一個子訊框，其中  $SFN$ 為訊框編號； $n$ 為子訊框編號； $T_C$ 則等於 $T_{C\_S}$ 或 $T_{C\_L}$ 。

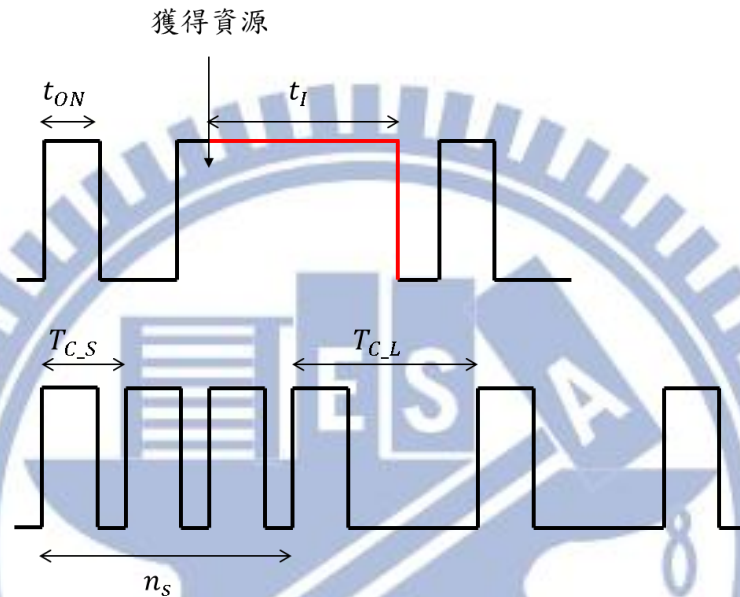


圖 5. 長程演進技術非連續接收模式運作架構

使用者裝置每次進入非連續接收週期便會連續地監聽 $t_{ON}$ 個子訊框的實體下傳控制通道，一旦接收到傳輸資料的通知便會延長 $t_I$ 個子訊框的清醒時間，繼續接收在佇列中的封包。反之，若使用者裝置在清醒時間結束前( $t_{ON} = 0$ 和 $t_I = 0$ )都未收到實體下傳控制通道的通知便會進入睡眠模式，直到下一個非連續接收週期才會回到清醒模式繼續監聽實體下傳控制通道。為了要簡化非連續接收模式的複雜度，我們只採用一種非連續接收週期，長度設定為小於即時性訊務的延遲限制。這樣設置的方式將不會有封包在一個非連續接收週期中到來，但卻在下一個非連續接收週期開始前就被丟棄。

## 第三章.

# 在非連續接收模式中做資源分配

---

### 3.1. 問題描述

當使用者裝置啟用了非連續接收模式後，其狀態便會在清醒模式和睡眠模式之間做切換，一旦進入睡眠便會暫時關閉其射頻電路，暫停接收來自演進型基地台的訊息，必須要等到下一個非連續接收週期啟動清醒區間定時器，使用者裝置才會再次接收實體下傳控制通道的訊息。若使用者裝置在清醒模式監聽到實體下傳控制通道通知有資料需要接收，非活動定時器將會被重置以延長使用者裝置的清醒時間。非活動定時器的原本目的是希望使用者裝置繼續保持清醒以消化在佇列中等待傳送的資料。倘若使用者裝置因非活動定時器重置而延長清醒時間，但是佇列中並沒有資料在等待傳送；或是佇列中有資料，但使用者裝置進入睡眠模式到下一個非連續接收週期前也不會違反其服務品質需求。此時這段被延長的清醒時間不但對提升系統吞吐量沒有幫助，更傷害非連續接收模式的省電效率。因此非活動定時器的啟動時機顯得格外重要，必須確保其啟動的目的才能發揮好的效果。

在長程演進技術中，傳統的封包排程器會在每個子訊框開始時執行資源分配演算法，並且根據該資源分配的方針(分配公平性、保證延遲限制、保證封包遺失率...等等)在傳送時間間隔內盡可能地滿足使用者裝置的需求。但很不幸地，非連續接收模式的特性使得使用者裝置不再是永遠保持清醒的狀態。若分配資源只保證使用者裝置在傳送時間間隔內滿足其需求，一旦它進入睡眠後無法參與資源分配時就有可能違反其需求。因此在非連續接收模式中分配資源只有保證使用者裝置在當前的子訊框內滿足其需求是不夠的，必須將使用者裝置進入睡眠模式後的狀態一起納入考慮。下一節我們將針對非連續

接收模式提出兩種感知型的資源分配演算法。

### 3.2. 感知型非連續接收模式之資源分配

我們於先前一章中提到方法[1]針對即時性訊務會根據其延遲限制和資料遺失率計算在當前的子訊框中滿足遺失率容忍度的最小所需頻寬(如圖 6)。方法[1]中計算滿足遺失率容忍度的最小所需頻寬時採用的是緊急策略(urgent policy)模式，因此只有考慮在當前子訊框結束時會遺失的資料量 $Q_n^1[t]$ 。為了要因應使用者裝置在非連續接收模式中，有可能在進入睡眠狀態後違反服務品質需求，我們在計算最小所需頻寬時將策略改為預留服務品質策略模式，保證使用者裝置在未來 $d$ 個子訊框中的服務品質，因此計算最小所需頻寬時考慮資料遺失量將為 $\sum_{i=1}^d Q_n^i[t]$ 。首先定義 $P_n^d[t]$ 為使用者 $n$ 在 $t^{th}$ 子訊框結束時考慮 $\sum_{i=1}^d Q_n^i[t]$ 的當前遺失率

$$P_n^d[t] = \frac{S_n[t-1] + L_n[t-1] + (\sum_{i=1}^d Q_n^i[t] - R_n[t])^+}{S_n[t-1] + L_n[t-1] + \max(\sum_{i=1}^d Q_n^i[t], R_n[t])}, \quad 0 \leq R_n[t] \leq Q_n[t] \quad (3)$$

接著根據遺失率容忍度，令 $P_n^*[t] = P_n$ 即可算出預留未來 $d$ 個子訊框中都能滿足遺失率容忍度的最小所需頻寬。

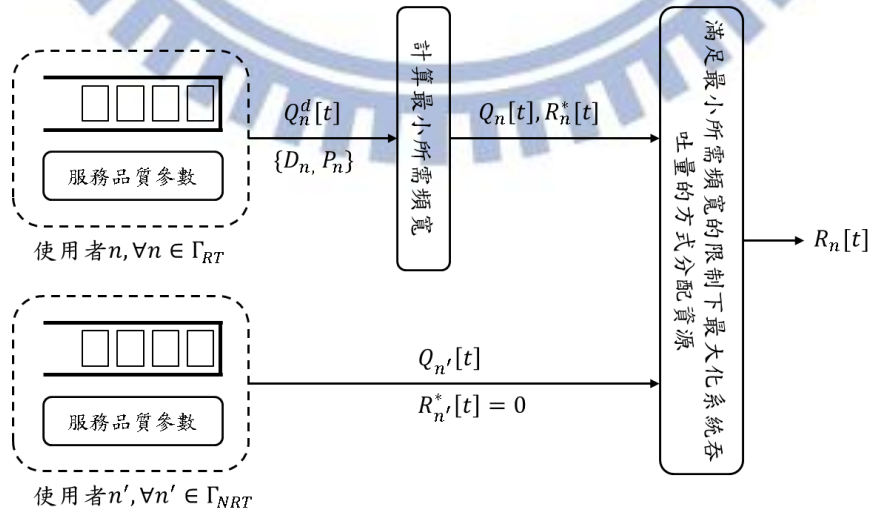


圖 6. 方法[1]資源分配架構圖

在非連續接收模式中，若 $t_{ON} > 0$ 或是 $t_I > 0$ 使用者裝置便會保持清醒狀態監聽實體下傳控制通道，一旦接收到通知有資料需要接收，非活動定時器將會被重置以延長使用者的清醒時間繼續保持傳輸。若使用者裝置尚未到達最後一個清醒的子訊框時，我們不採用服務品質預留策略，其原因有兩點：(1)使用者裝置還有剩餘的清醒時間去滿足對服務品質的需求；(2)過早預留資源將減少其他的使用者裝置可以得到的資源。而當使用者裝置到達最後一個清醒的子訊框時，我們才以預留服務品質的模式來判斷使用者裝置是否會因為進入睡眠模式而違反服務品質需求。在接下來的小節中，我們將提出兩種感知型非連續接收資源分配演算法。另外針對使用者 $n \in \Gamma_{RT}$ 定義 $t_R$ 為該使用者裝置距離到達下一個非連續接收週期的子訊框個數。

### 3.2.1. 演算法一

假設有一個使用者裝置連結即時性訊務，而且非連續接收模式沒有啟用非活動定時器，這樣此使用者裝置只會在每個非連續接收週期中的清醒區間監聽實體下傳控制通道，其餘時間都將進入睡眠模式。基於此概念，最直覺的方式便是使用者裝置在清醒模式時要盡可能地被服務，防止使用者裝置進入睡眠模式後違反其服務品質需求。因此演算法一是透過預留資源的方式，讓使用者裝置在睡眠模式前能盡量獲得足夠多的資源，就能降低在睡眠模式中違反服務品質的機會。當使用者 $n \in \Gamma_{RT}$ 在非連續接收模式中到達最後一個清醒的子訊框，封包排程器便計算 $P_n^{t_R}[t]$ ，接著令 $P_n^{t_R}[t] = P_n$ 計算出 $R_n^*[t]$ 。這裡計算出的 $R_n^*[t]$ 的意義將是使用者 $n$ 在到達下一個非連續接收週期前皆不會違反其遺失率容忍度的最小所需頻寬。若 $R_n^*[t] > 0$ 則表示使用者裝置 $n$ 對頻寬有需求，如果沒有得到任何頻寬而進入睡眠模式的話，使用者裝置將會在下次回到清醒模式前違反其遺失率容忍度。

---

**Algorithm 1** DRX-aware Resource Allocation at the beginning of  $t^{th}$  subframe
 

---

Input:  $\{\{D_n, P_n\}, \{Q_n^d[t]\}_{d=1,2,\dots,D_n}\}_{\forall n \in \Gamma_{RT}}, \{Q_n[t]\}_{\forall n \in \Gamma}$

Output:  $\{R_n[t]\}_{\forall n \in \Gamma}$

Variables:

$\phi_1$ : The set of RTs which the minimum requested bandwidth is bigger than zero.

$R_n[t]$ : The bandwidth allocated to user  $n$  at  $t^{th}$  subframe.

for ( $n = 1$  to  $|\Gamma|$ )

  if (user  $n \in \Gamma_{RT}$ ) then

    if (user  $n$  remains last awake subframe) then

$R_n^*[t]$  is calculated according to  $P_n^{tR}[t]$

    else

$R_n^*[t]$  is calculated according to  $P_n^1[t]$

    end if

    if ( $R_n^*[t] > 0$ ) then

      user  $n \in \phi_1$

    end if

  else

$R_n^*[t]$  is set zero

  end if

end for

while ( $u_{n,m} \neq 0 \exists m \exists n$ ) do

  if ( $\phi_1 \neq \emptyset$ ) then

$$u_{n^*, m^*} = \max_{n \in \phi_1, 1 \leq m \leq M} \{r_{n, m}[t]\}$$

$$R_{n^*}^*[t] = \max\{0, R_{n^*}^*[t] - r_{n^*, m^*}[t]\}$$

  if ( $R_{n^*}^*[t] == 0$ ) then

    remove user  $n^*$  from  $\phi_1$

  end if

  else

$$u_{n^*, m^*} = \max_{n \in \Gamma, 1 \leq m \leq M} \{r_{n, m}[t]\}$$

  end if

$$Q_{n^*}^*[t] = \max\{0, Q_{n^*}^*[t] - r_{n^*, m^*}[t]\}$$

$$R_{n^*}^*[t] += r_{n^*, m^*}[t]$$

```

replace  $u_{n, m^*} = 0 \forall n \in \Gamma$ 
if ( $Q_n^*[t] == 0$ ) then
    replace  $u_{n^*, m} = 0 \forall m$ 
end if
end while

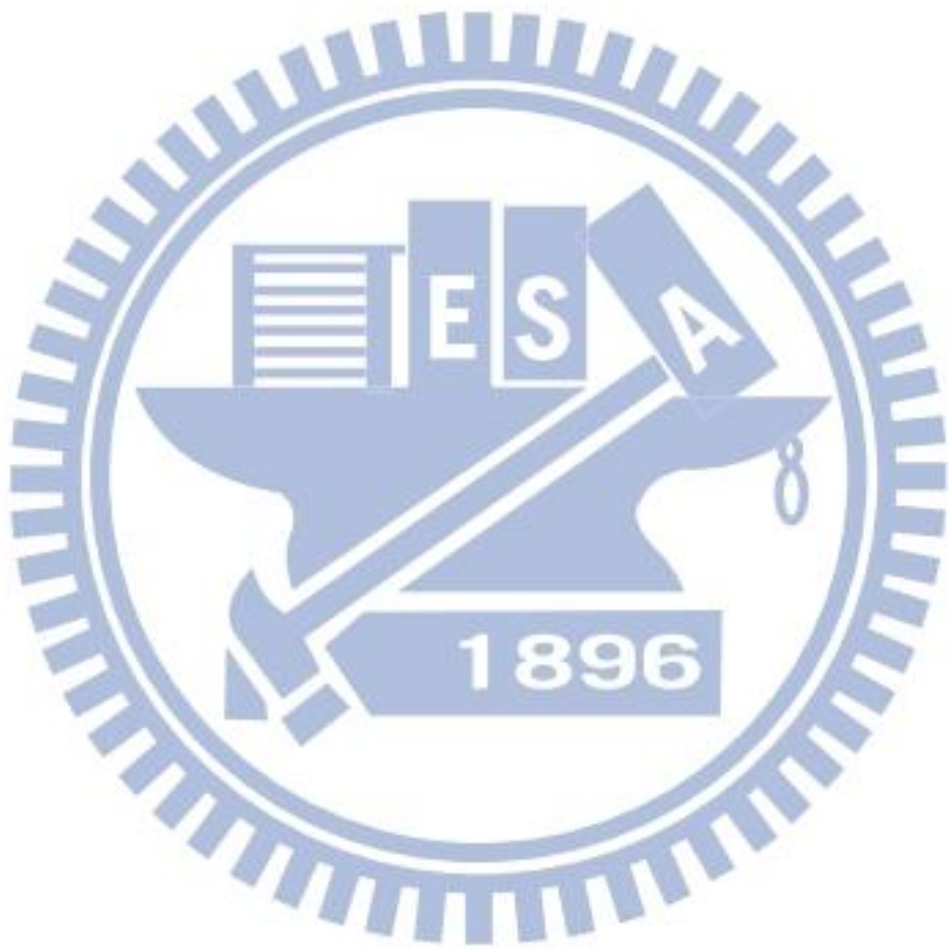
```

---

### 3.2.2. 演算法二

前一節演算法的主要構想是預留資源給即將進入睡眠模式的使用者裝置，雖然這樣是較直覺地防止在睡眠模式中違反服務品質的方式，但是並沒有運用到非活動定時器的特點，就是延長使用者裝置的清醒時間以繼續接收在佇列中等待的資料。若使用者裝置是在進入睡眠模式後才會違反服務品質需求，只要能夠啟動其非活動定時器延長清醒時間就能繼續參與資源分配。因此我們提出演算法二，運用非活動定時器防止那些會在睡眠模式中違反服務品質需求的使用者裝置保持清醒狀態。首先，一樣在使用者  $n \in \Gamma_{RT}$  到達最後一個清醒的子訊框時判斷是否會在睡眠模式違反遺失率容忍度。但與演算法一不同的是這裡我們的判斷將分成三種結果

- 1) 令  $P_n^1[t] = P_n$  計算出  $R_n^*[t]$  大於零：這意謂使用者  $n$  在此子訊框有最小頻寬需求，若被分配到資源就會啟動非活動定時器，因此最小所需頻寬維持此計算結果。
- 2) 令  $P_n^1[t] = P_n$  計算出  $R_n^*[t]$  等於零，但是令  $P_n^{tR}[t] = P_n$  計算出  $R_n^*[t]$  大於零：這表示使用者  $n$  在此子訊框中雖然沒有最小頻寬需求，但若進入睡眠模式將會違反遺失率容忍度。因此我們不需要過早預留資源，只要令  $R_n^*[t] = 1$  使得使用者裝置至少有一個資源區塊的需求。若使用者裝置有得到資源而啟動非活動定時器就能繼續保持清醒，防止進入睡眠模式違反其遺失率容忍度。
- 3) 令  $P_n^1[t] = P_n$  計算出  $R_n^*[t]$  和令  $P_n^{tR}[t] = P_n$  計算出  $R_n^*[t]$  皆等於零：這表示使用者  $n$  在到達下一個非連需傳輸週期前的子訊框中，沒有任何頻寬需求也不會違反其遺失率容忍度。因此最小所需頻寬設成零，將資源保留給其他使用者裝置。



---

**Algorithm 2** DRX-aware Resource Allocation at the beginning of  $t^{th}$  subframe
 

---

Input:  $\{\{D_n, P_n\}, \{Q_n^d[t]\}_{d=1,2,\dots,D_n}\}_{\forall n \in \Gamma_{RT}}, \{Q_n[t]\}_{\forall n \in \Gamma}$

Output:  $\{R_n[t]\}_{\forall n \in \Gamma}$

Variables:

$\phi_1$ : The set of RTs which the minimum requested bandwidth is bigger than zero.

$R_n[t]$ : The bandwidth allocated to user  $n$  at  $t^{th}$  subframe.

for ( $n = 1$  to  $|\Gamma|$ )

  if (user  $n \in \Gamma_{RT}$ ) then

$R_n^*[t]$  is calculated according to  $P_n^1[t]$

    if (user  $n$  remains last awake subframe &&  $R_n^*[t] == 0$ ) then

$R_n^*[t]$  is calculated according to  $P_n^{tR}[t]$

      if ( $R_n^*[t] > 0$ ) then

$R_n^*[t] = 1$

      end if

    end if

    if ( $R_n^*[t] > 0$ ) then

      user  $n \in \phi_1$

    end if

  else

$R_n^*[t]$  is set zero

  end if

end for

while ( $u_{n,m} \neq 0 \exists m \exists n$ ) do

  if ( $\phi_1 \neq \emptyset$ ) then

$$u_{n^*, m^*} = \max_{n \in \phi_1, 1 \leq m \leq M} \{r_{n, m}[t]\}$$

$$R_{n^*}^*[t] = \max\{0, R_{n^*}^*[t] - r_{n^*, m^*}[t]\}$$

  if ( $R_{n^*}^*[t] == 0$ ) then

    remove user  $n^*$  from  $\phi_1$

  end if

  else

$$u_{n^*, m^*} = \max_{n \in \Gamma, 1 \leq m \leq M} \{r_{n, m}[t]\}$$

  end if



$$Q_n^*[t] = \max\{0, Q_n^*[t] - r_{n^*, m^*}[t]\}$$

$$R_n^*[t] += r_{n^*, m^*}[t]$$

*replace*  $u_{n, m^*} = 0 \forall n \in \Gamma$

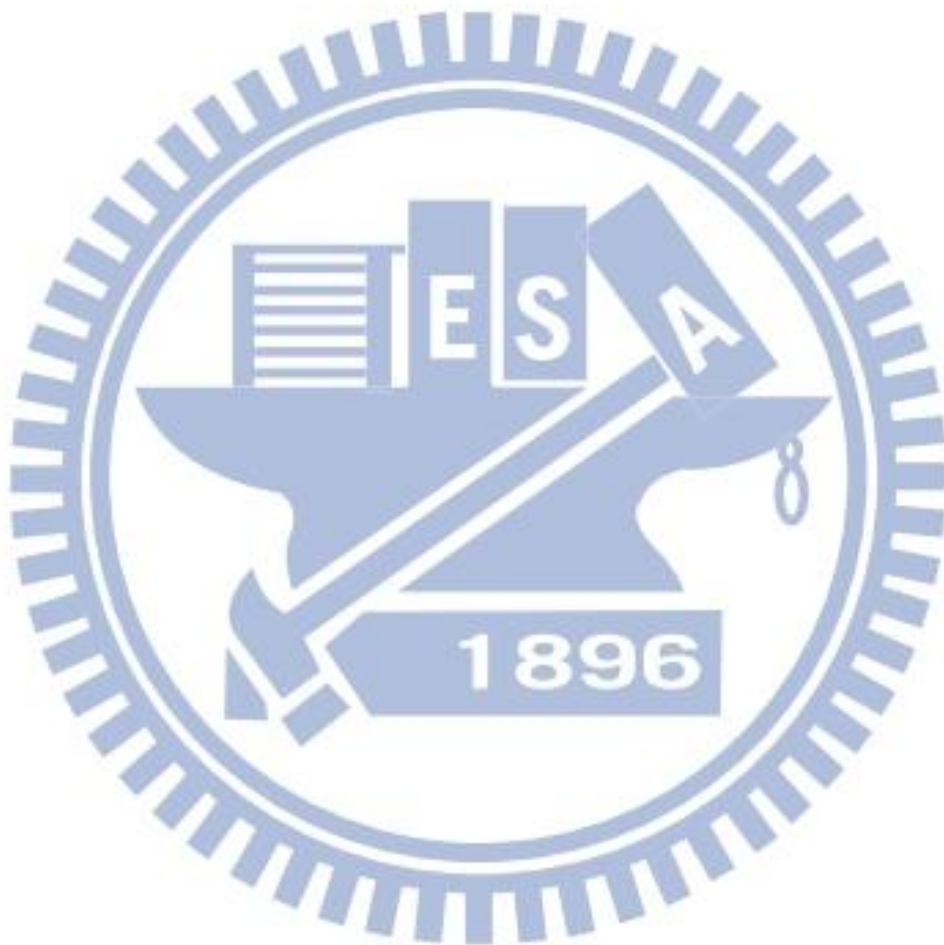
if ( $Q_n^*[t] == 0$ ) then

*replace*  $u_{n^*, m} = 0 \forall m$

end if

end while

---

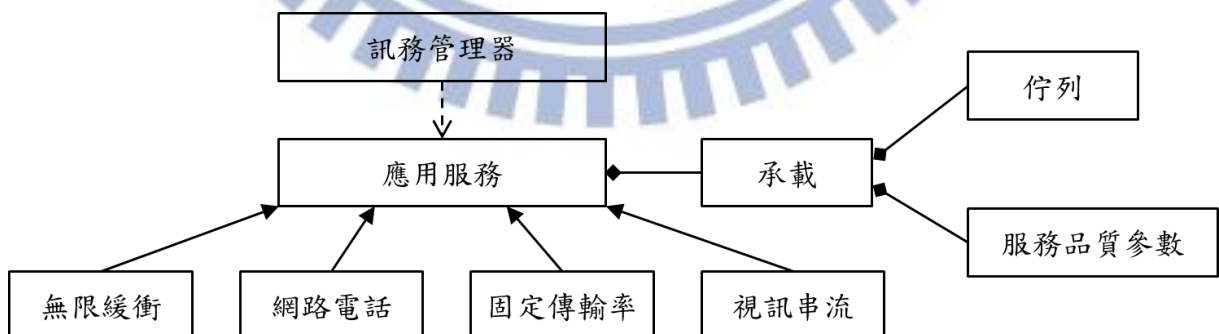


## 第四章.

### 模擬

#### 4.1. 模擬環境

本章節中，我們將使用*LTE-Sim*模擬器[13]執行演進型技術系統層級的模擬。*LTE-Sim*是一個開放源碼基於C++的模擬器，其設計的理念是希望能完整實現演進型技術協定層，提供演進型技術的研究者們便於修改、增加且公平比較的模擬平台。協定層方面，*LTE-Sim*支援：無線資源控制管理承載，並且能根據不同的即時性訊務對其承載設定不同的服務品質需求(如圖 7)；資料匯聚通訊協定對上層封包進行資料保密和壓縮；媒體存取控制執行無線資源管理，封包排程器被定義於此層進行上行鍊路與下行鍊路的資源分配；實體層儲存和管理演進型技術中節點之間的無線通道資訊，並且也實現了無線通道傳遞衰減模型。由於*LTE-Sim*尚未支援非連續接收模式，根據[11]我們將非連續接收模組定義於媒體存取控制層(如圖 8)，並且增加演算法一和演算法二的封包排程器，執行我們提出的資源分配演算法。



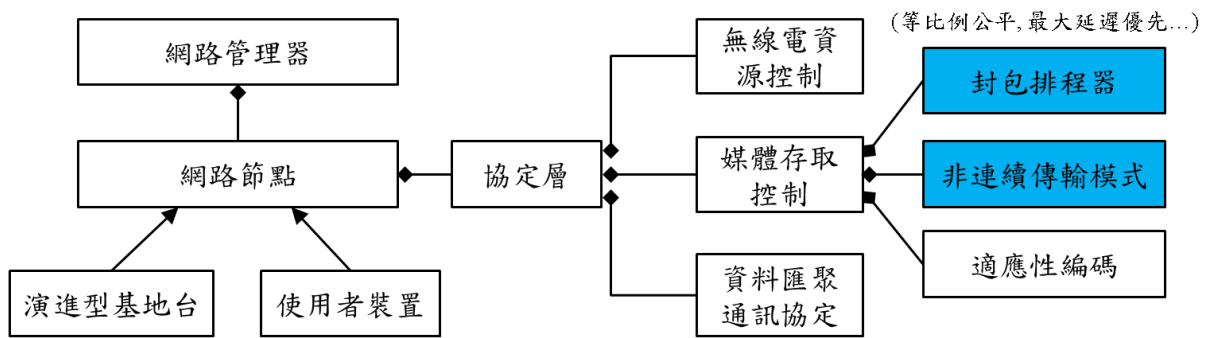


圖 8. LTE-Sim 網路管理架構圖

我們的模擬中，使用者裝置會均勻分佈在演進型基地半徑一公里的圓形區域內，並且只考慮下行鍊路的傳輸。使用者裝置在一段通話期間只會使用即時性或是非即時性其中一種型態的訊務流。其中即時性我們採用視頻串流真實的追蹤檔案(trace file)；非即時訊務我們採用無限緩衝(Infinite Buffer)的訊務，也就是佇列中永遠都會有封包等待被傳送。傳輸通道品質會隨著時間改變，使用者裝置回報的通道品質指示採用理想的方式，不會有回報延遲。演進型基地台會採用適應性調變編碼，根據使用者裝置目前的通道品質選擇可以使用的調變等級，並透過實體下傳控制通道告知使用者裝置。其他關於環境參數、訊務特性、服務品質需求設定將被統整在表 1。另外非連續接收模式的參數設定將會在下節中討論。

我們的模擬將分成兩種情境，每一個情境中 $|\Gamma_{NRT}| = 10$ 且非即時性訊務的最小所需頻寬都設為零。在第一種情境沒有啟用非連續接收模式且會有不同個數的使用者裝置連結即時性訊務；第二種情境有啟用非連續接收模式並且 $|\Gamma_{RT}| = 20$ 。執行模擬器的電腦配備為Intel Core i5 3.2GHz之CPU，記憶體為8GB，模擬時間長度為三十萬個子訊框。

表 1. 模擬環境參數、訊務特性及服務品質需求設定

環境參數	
細胞半徑	1 km
使用者分佈	均勻分佈
頻寬	5 MHz
通道模型	瑞利衰減通道
使用者移動速度	3 km/hr
訊框長度	10 ms
子訊框長度	1 ms
時槽長度	0.5 ms
資源區塊對個數	25
每個資源區塊對中的子載波個數	12
子載波寬度	15 kHz
回報通道品質間隔	1 ms
訊務特性	
內容	實際的影片檔案
影片格式	H.264
平均封包到達時間	40 ms
平均封包大小	1228 bytes
延遲限制	160 ms
遺失率容忍度	10%

## 4.2. 非連續接收模式對資源分配的影響

在這小節中，我們將針對等比例公平演算法、最大延遲優先演算法以及方法[1]比較系統吞吐量和平均遺失率在非連續接收模式中受到的影響，並且分析非連續接收模式參數的設定方式。其中系統吞吐量的計算方式為：

$$\frac{\text{所有使用者接收封包的總量}}{\text{模擬時間}} \quad (4)$$

平均封包遺失率的計算方式為：

$$1 - \frac{\text{所有使用者接收封包的總數}}{\text{所有使用者傳送封包的總數}} \quad (5)$$

最後，省電效率的計算方式為：

$$\frac{\text{所有使用者睡眠時間的總合}}{\text{模擬時間}} \quad (6)$$

我們在第二章提過為了簡化非連續接收機制，因此只使用一種非連續接收週期，並且長度設定為小於即時性訊務的延遲限制，所以長度為一百二十八個子訊框。每個使用者裝置的非連續接收偏移值則為均勻分佈在非連續接收週期的區間內。接下來我們將用模擬的方式，探討清醒區間定時器和非活動定時器對效能的影響以及合理的設定方式。圖 9和圖 10為沒有啟動非連續接收模式時的系統吞吐量和平均遺失率。可以看到方法 [1]在保證即時性訊務的最小所需頻寬下，最大化系統吞吐量的方式分配資源，因此在系統吞吐量的表現是三個之中最佳的。而最大延遲優先演算法保障延遲較大的即時性訊務有較高的優先權，因此系統吞吐量較等比例公平演算法差一些。等比例公平演算法雖然在吞吐量上表現較最大延遲優先演算法稍好，但是其因為沒有保證任何的服務品質需求，因此在 $|\Gamma_{RT}| = 20$ 時平均遺失率就已超過遺失率容忍度。最大延遲優先演算法保障延遲較大的即時性訊務有較高優先權，因此在 $|\Gamma_{RT}| = 30$ 之前平均遺失率表現最佳，但代價就是犧牲了系統吞吐量。方法[1]在 $|\Gamma_{RT}| = 30$ 之前，即時性訊務的封包遺失率會一直保持在百分之十左右，原因在於其分配資源方式只優先保障即時性訊務滿足遺失率容忍度的最小頻寬，接著以資源以最大化系統吞吐量的方式分配剩餘資源，藉此同時兼顧吞吐量和服務品質需求。

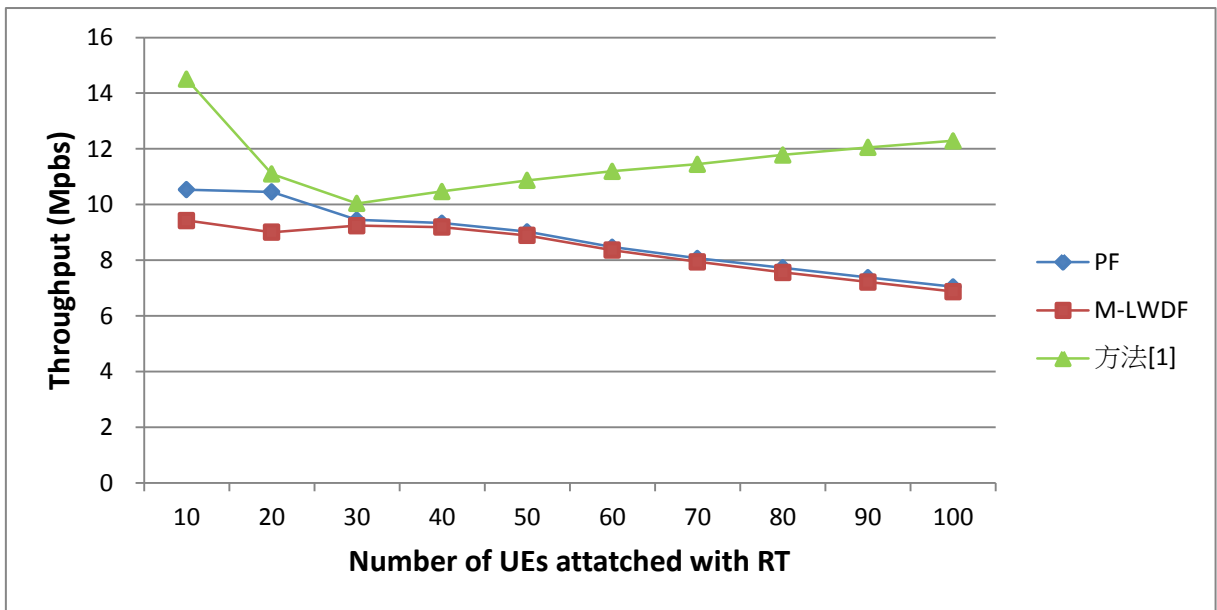


圖 9. 第一種情境中比較系統吞吐量

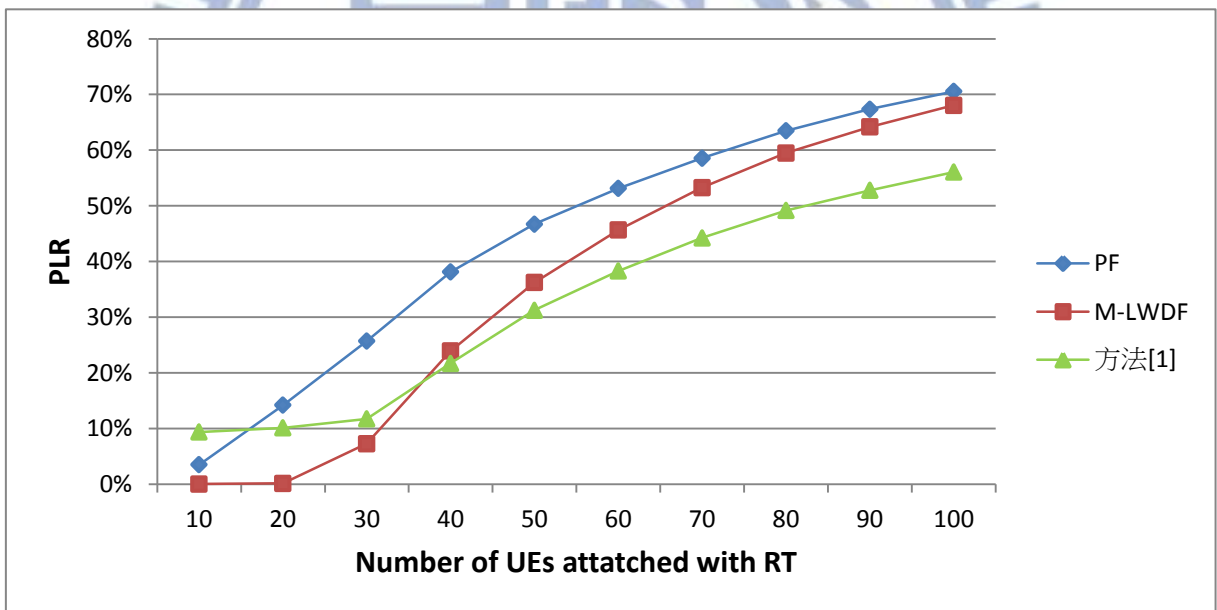
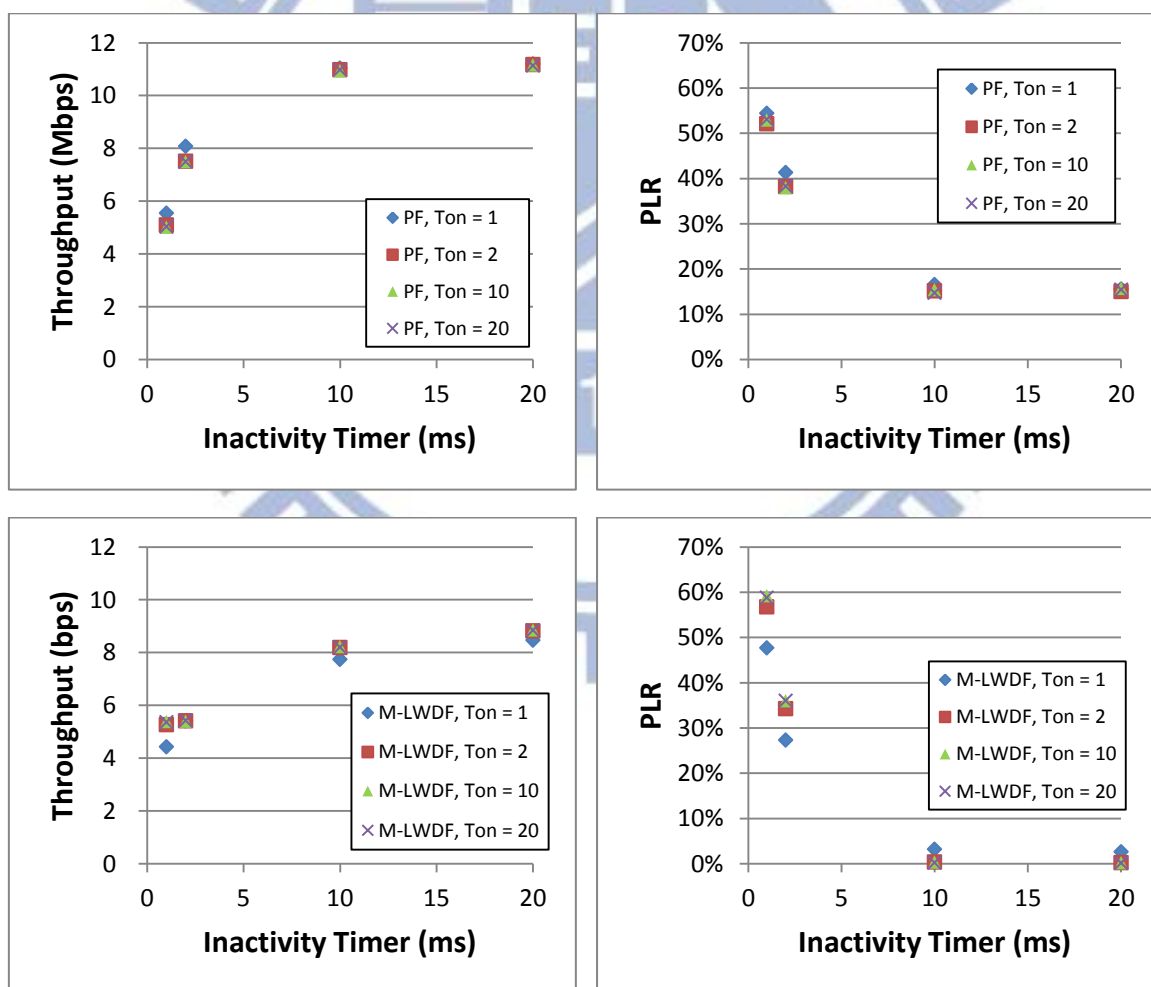


圖 10. 第一種情境中比較平均遺失率

圖 11 為這三種資源分配演算法在第二種情境中，套用不同清醒區間定時器與非活動定時器的系統吞吐量與平均遺失率。從其中可以發現不同長度的清醒區間定時器對於效能的影響並不大，原因在於清醒區間定時器的目的是為了保證使用者裝置在每一個非

連續週期中都能夠回到清醒模式一段時間監聽實體下傳控制通道是否會有資料需要傳輸，避免資料在還未傳送前便被丟棄。因此增加清醒區間定時器只是提升使用者裝置在每個非連續接收週期中獲得第一次資源的機會。而非活動定時器的目的是當使用者裝置獲得資源後，延長其清醒時間繼續接收在佇列中的封包，因此才能有效地消化佇列的長度。由此實驗結果可得知，在每一個非連續接收週期中，清醒定時器的長度只要足夠讓使用者裝置在一個非連續接收週期中獲得第一次資源即可，接著延長清醒時間的非活動定時器才是保持傳輸並且改善效能的關鍵。因此在接下來的模擬中，清醒區間定時器的長度將會被設定為二個子訊框，其他非連續接收模式的相關參數統整在表 2。



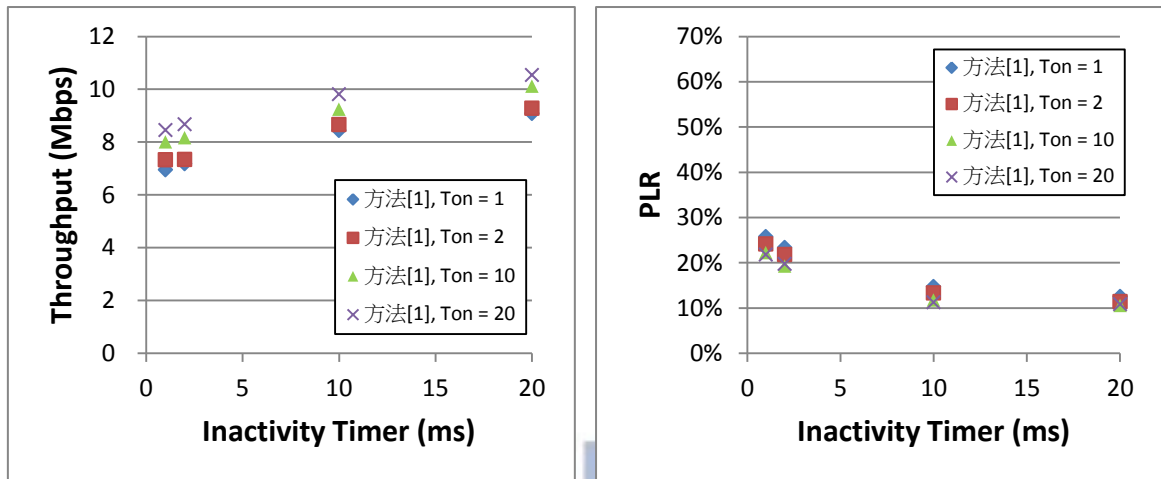


圖 11. 第二種情境中不同的清醒區間定時器與非連續活動參數定時器對傳統資源分配演算法之影響

表 2. 非連續接收模式參數設定

非連續接收週期參數	
清醒區間定時器	2 ms
非活動計時器	1, 2, 5, 10, 20 ms
短非連續接收週期	無
短非連續接收週期計時器	無
長非連續接收週期	128 ms
偏移值	均勻分佈

### 4.3. 效能評估

首先在第二種情境下比較方法[1]和我們提出的兩種感知型非連續接收模式的資源分配演算法，針對不同長度的非活動定時器比較系統吞吐量、平均遺失率和省電效率。圖 12 可以看出演算法一與演算法二在系統吞吐量方面都比方法[1]較高。因為演算法一



會預留資源給剩下最後一個清醒時間的使用者裝置，保證該使用者在未來的子訊框都能滿足遺失率容忍度，因而提升使用者裝置對最小頻寬的需求。基於此原因，通道品質較好的使用者裝置會獲得更多的資源區塊對以滿足提升過後的最小頻寬需求，也因此提升了系統吞吐量。但是從圖 13 可以發現演算法一預留資源給即將進入睡眠模式的使用者裝置，將會減少其他也有最小頻寬需求的使用者裝置可以得到的資源，因此平均遺失率反而較方法[1]還高。演算法二不做預留資源，而是利用非活動定時器延長那些會在睡眠模式中違反遺失率容忍度的使用者裝置繼續保持清醒去參與資源分配，慢慢消化佇列中的資料。這樣的方式會增加使用者裝置的清醒時間接收更多封包，因此提升系統吞吐量，而且在不會影響其他對最小頻寬有需求的使用者的前提下，有效降低封包遺失率。由於這三種方法為了最大化系統吞吐量，分配資源不論是在滿足即時性訊務最小頻寬需求的第一回合或是考慮所有訊務的第二回合，都會從通道品質最好的使用者裝置開始分配。在滿足遺失率容忍度的情形下，只有通道較好的使用者裝置有機會啟動非活動定時器延長清醒時間，因此圖 14 顯示提升清醒區間定時器和非活動定時器，這三個演算法都能保持很好的省電效率。但是因為演算法一和演算法二會針對在睡眠中違反遺失率容忍度的使用者裝置增加啟動非活動定時器的機會，因此會稍微提升電力消耗。

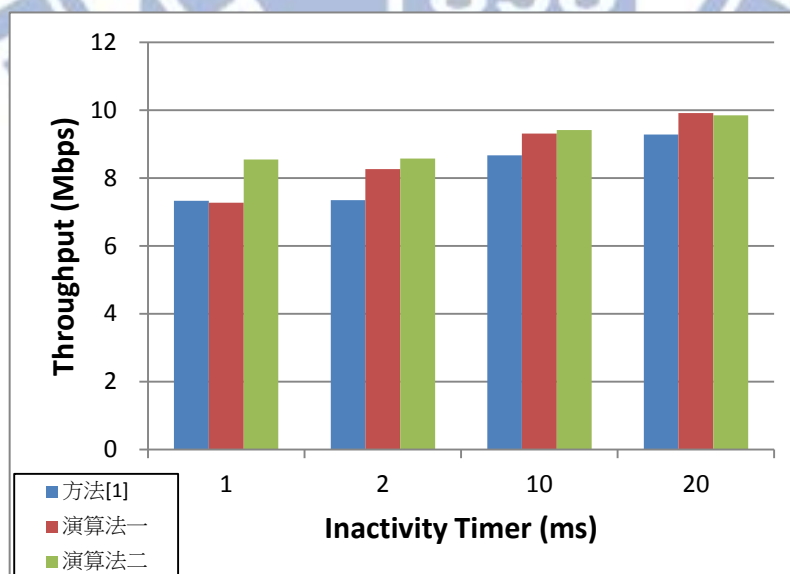


圖 12. 第二種情境中比較系統吞吐量

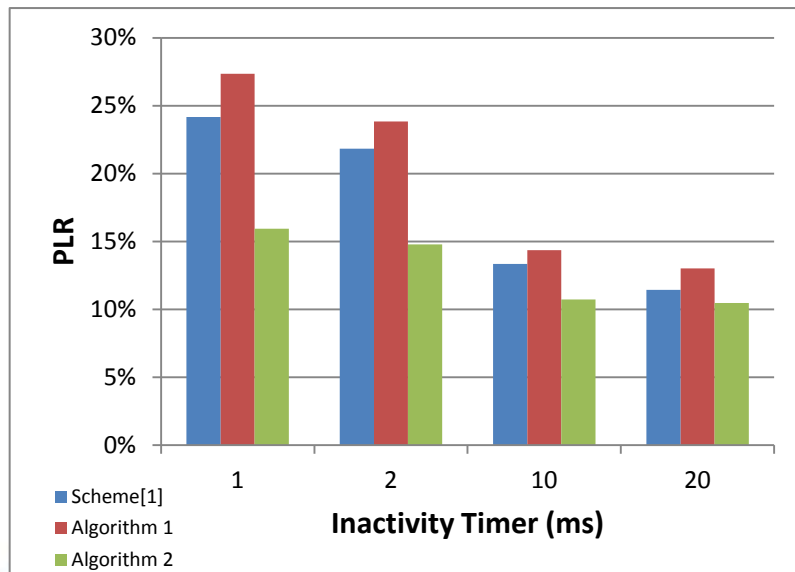


圖 13. 第二種情境中比較平均遺失率

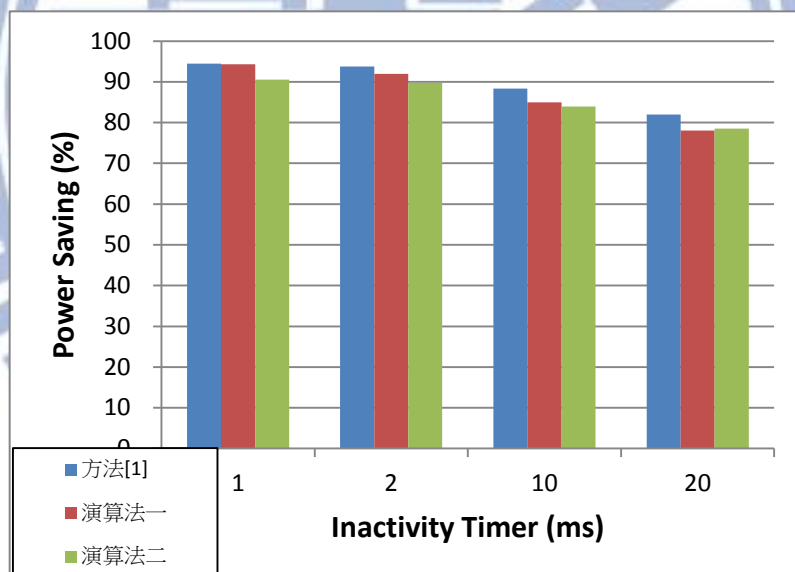


圖 14. 第二種情境中比較省電效能

雖然演算法一以較直覺的方式保障使用者裝置進入睡眠模式前能獲得足夠的頻寬，但沒有充份利用非連續活動定時器以及過早佔用資源反而損害了整體的平均遺失率。因此下一段中，我們只比較等比例公平、最大延遲優先、方法[1]和演算法二的系統吞吐量、

平均遺失率和省電效能。

圖 15 可以看到等比例公平演算法的系統吞吐量很容易受到非活動定時器的影響。當非活動定時器很小時，使用者裝置被通知有資料需要接收而啟動非活動定時器，因為延長清醒時間不夠久，基於公平的分配方式無法讓使用者裝置再次獲得資源以延續傳輸資料。但是當非活動定時器夠大時，使用者裝置每延長一次清醒時間便足以在公平的原則下再次獲得資源以延續傳輸。因此當非活動定時器足夠大時，等比例公平演算法能夠平均地讓使用者裝置保持清醒接收在佇列中的資料，所以會有最好的系統吞吐量。最大延遲優先演算法雖然也有保障一定程度的資源分配公平性，但是在非連續接收模式中封包延遲更加嚴重，分配資源時都會以即時性訊務為優先，因此系統吞吐量的表現並不好。方法[1]和演算法二如前段所敘述的，在保障即時性訊務滿足遺失率容忍度的最小頻寬後最大化系統吞吐量，因此在吞吐量方面都有不錯的效能。但是演算法二會讓那些進入睡眠模式後會違反遺失率容忍度的使用者裝置保持清醒獲得更多的傳輸機會，因而會有更好的系統吞吐量。

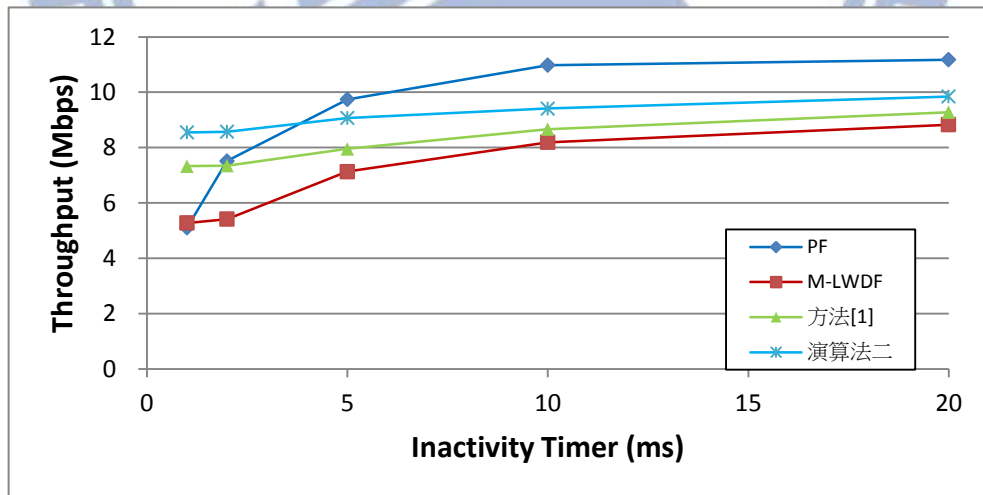


圖 15. 演算法二與傳統資源分配演算法在非連續接收模式中比較系統吞吐量

圖 16 中顯示等比例公平演算法的平均遺失率是最高的，原因在於資源分配時

沒有考慮使用者裝置的服務品質需求。在非連續接收模式中，最大延遲優先演算法將會給予即時性訊務很高的優先權，因此當非活動定時器夠大時，連結即時性訊務的使用者裝置很容易被分配到資源而延長清醒時間，所以能夠達到很低的平均遺失率。但不論是等比例公平或是最大延遲優先演算法，都會保障資源分配的公平性，非活動定時器一定要夠大才能延續使用者裝置繼續傳輸資料，因此系統效能很容易受到不同非活動定時器的設定影響。方法[1]想盡可能地保障即時性訊務獲得滿足遺失率容忍度的最小頻寬，但在非連續接收模式中遺失率將會提升使得即時性訊務的最小頻寬需求增加，不容易滿足所有連結即時性訊務的使用者裝置的最小所需頻寬，因此平均遺失率會高於遺失率容忍度。另外，因為沒有考慮使用者裝置進入睡眠模式後是否會違反遺失率容忍度，使其在非活動定時器較小時遺失率會上升較多。演算法二雖然在非活動定時器較小時也無法將遺失率保持在遺失率容忍度附近，但是因為會針對進入睡眠模式後會違反遺失率容忍度的使用者裝置增加其啟動非活動定時器的機會，因此非活動定時器很小時，平均遺失率只會稍微上升，並且隨著增加非活動計時器較早便能將遺失率降低至遺失率容忍度。

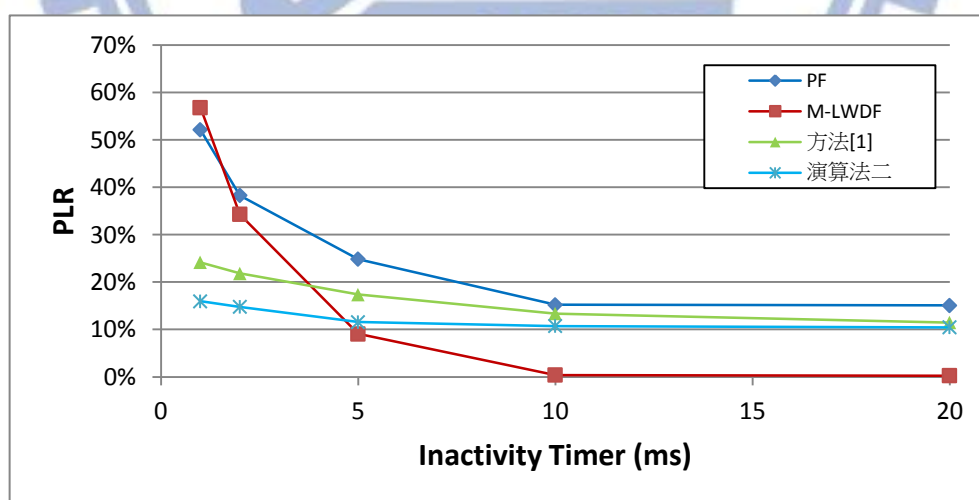


圖 16. 演算法二與傳統資源分配演算法在非連續接收模式中比較平均遺失率

圖 17顯示出等比例公平與最大延遲優先演算法都會保障資源分配的公平性，在非活動定時器較大時平均地延長使用者裝置的清醒時間，因此降低了省電效能。不過最大延遲優先演算法中，延遲較嚴重的那些特定使用者裝置較容易獲得資源延長清醒時間，因此整體省電效能較等比例公平演算法好。方法[1]就如前段所敘述，在滿足即時性訊務最小所需頻寬的前提下，通道較佳的使用者裝置較有機會保持清醒，因此整體省電效能最好。演算法二為了防止使用者裝置進入睡眠模式後違反遺失率容忍度，因此會增加啟動非活動定時器的機會所以較方法[1]耗電，不過最大只多出將近百分之四的耗電量卻能夠提升了百分之十八的系統吞吐量及降低百分八的平均遺失率。

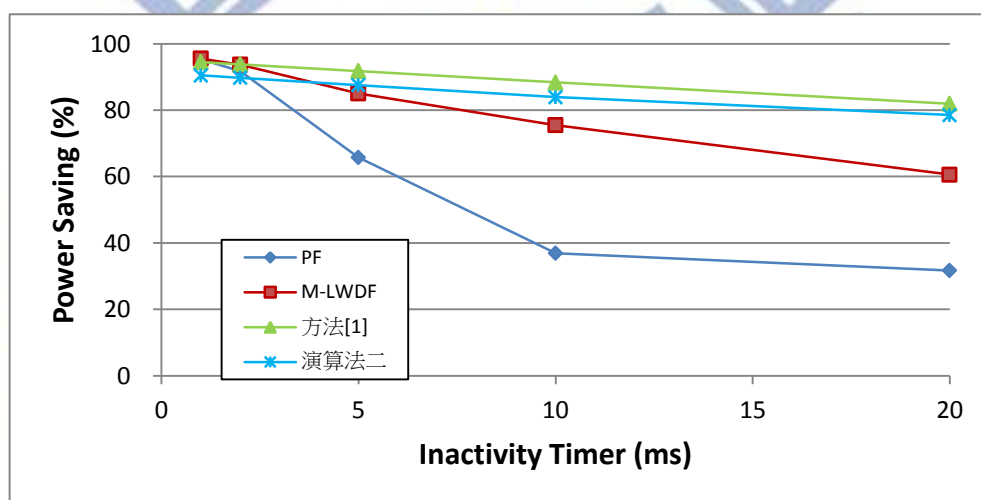


圖 17. 演算法二與傳統資源分配演算法在非連續接收模式中比較省電效率

## 第五章.

### 結論

---

我們在這篇論文中提出兩個感知型非連續接收模式的資源分配演算法，嘗試降低非連續接收模式對網路效能的傷害，並且盡可能地保證即時性訊務的服務品質且維持好的省電效能。我們將方法[1]根據即時性訊務的遺失率容忍度以及當前遺失率計算最小所需頻寬延伸成推算在一個非連續接收週期的剩餘時間中是否會違反其遺失率需求。為了讓使用者裝置在一個非連續接收週期中都能夠滿足遺失率需求，演算法一嘗試讓使用者裝置在進入睡眠前便能獲得足夠的頻寬，而演算法二則是嘗試增加啟動非活動計時器的機會以延長清醒時間繼續參與資源分配。

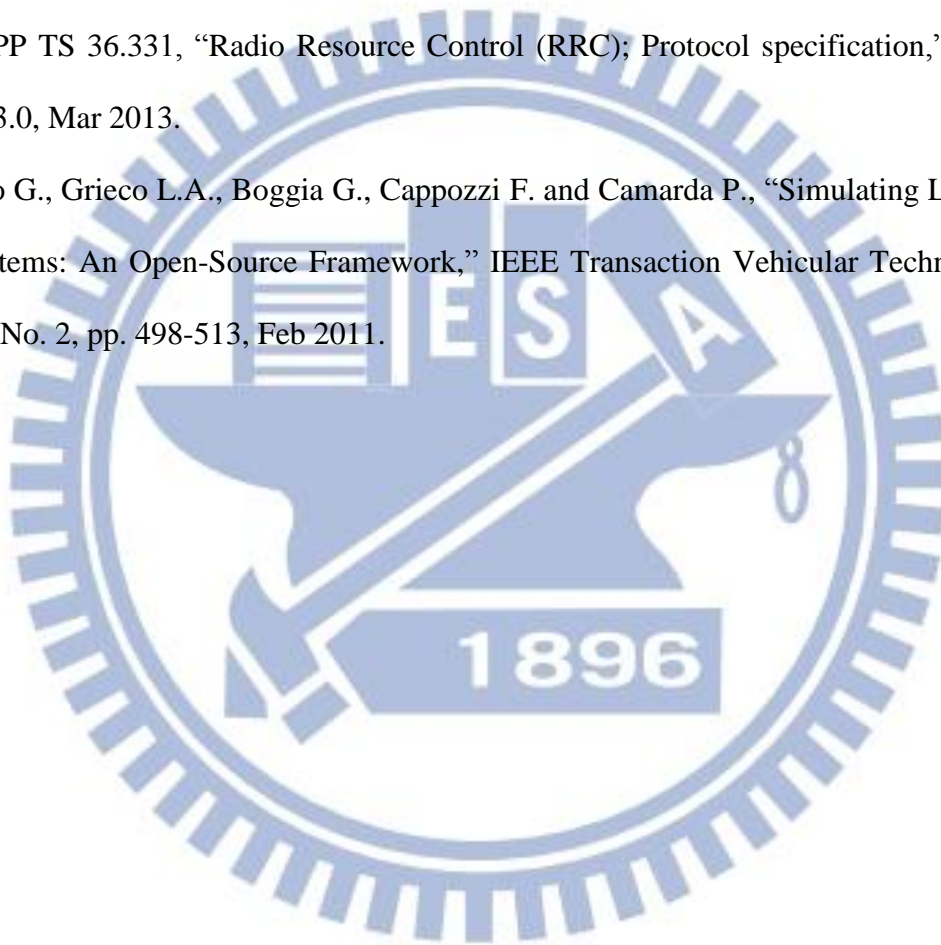
實驗結果顯示，雖然演算法一能夠提升系統吞吐量，但是過早佔用資源的代價便是增加其他使用者裝置的封包遺失率。而方法二不但能夠提升系統吞吐量，有效降低封包遺失率，而且只針對會在睡眠區間違反遺失率容忍度的使用者裝置延長清醒時間的方式仍然保持很好的省電效率。不過在這次的研究中，我們注意到非連續接收模式的參數設定對於每個資源分配演算法會造成不同的影響，像是等比例公平演算法平均分配資源的特性，只要使用者裝置每次清醒時間夠長時便能一直延續傳輸時間反而讓系統吞吐量超越方法[1]。由於第三代合作伙伴計劃並沒有規範資源分配的方法，因此各家營運商皆有自己的資源分配技術。對於不同的資源分配方法，如何配置合適的非連續接收模式的參數讓使用者裝置能夠在保持省電的情形下維持良好的服務品質將是未來需要探討的。

## 參考文獻

---

- [1] Tsern-Huei Lee and Yu-Wen Huang, "Resource Allocation for Real-Time and Non-Real-Time Traffic in OFDMA-Based Systems," IEEE Transaction on Communications, Vol. 60, No.3, pp. 851-861, Mar 2012
- [2] Bontu C.S. and Illidge E., "DRX Mechanism for Power Saving in LTE," IEEE Communications Magazine, Vol. 47, No. 6, pp. 48-55, June 2009.
- [3] Wigard J., Kolding T., Dalsgaard L. and Coletti C., "On the User Performance of LTE UE Power Savings Schemes with Discontinuous Reception in LTE," ICCW 2009, June 2009.
- [4] Kim R.Y. and Mohanty S., "Advanced Power Management Techniques in Next-generation Wireless Networks," IEEE Communications Magazine, Vol.48, No.5, pp. 94-102, May 2010.
- [5] Ramli H.A.M., Basukala R., Sandrasegaran K., Patachaianand R., "Performance of Well Known Packet Scheduling Algorithms in the Downlink 3GPP LTE System," MICC 2009, Dec 2009.
- [6] Jalali A., Padovani R. and Pankaj R., "Data Throughput of CDMA-HDR a High Efficiency-high Data Rate Personal Communication Wireless System," IEEE VTC 2000-Spring, Vol. 3, pp. 1854-1858, May 2000.
- [7] Andrews M, Kumaran K., Ramanan K., Stolyar A., Whiting P. and Vijayakumar R., "Providing quality of service over a shared wireless link," IEEE Communications Magazine, Vol. 39, No.2, pp. 150-154, Feb 2001.
- [8] Dahlman E. and Parkvall S., 4G LTE/LTE-Advanced for Mobile Broadband, Academic Press, 2011.


- [9] 3GPP TS 36.331, "Physical layer procedures," Rel. 11, v. 11.0.0, Sep 2012.
- [10] Wengerter C., Ohlhorst J. and Von Elbwart A.G.E., "Fairness and throughput analysis for generalized proportional fair frequency scheduling in OFDMA," IEEE VTC 2005-Spring, Vol. 3, pp. 1903-1907, June 2005.
- [11] 3GPP TS 36.321, "Medium Access Control (MAC) Protocol Specification," Rel. 11, v. 11.1.0, Dec 2012.
- [12] 3GPP TS 36.331, "Radio Resource Control (RRC); Protocol specification," Rel. 11, v. 11.3.0, Mar 2013.
- [13] Piro G., Grieco L.A., Boggia G., Cappozzi F. and Camarda P., "Simulating LTE Cellular Systems: An Open-Source Framework," IEEE Transaction Vehicular Technology, Vol. 60, No. 2, pp. 498-513, Feb 2011.





102

碩  
士  
論  
文

「」系統非連續接收之省電兼顧服務品質下行鏈路資源  
分配演算法

交通大學

電機學院  
電信工程研究所碩士班

林  
裕  
捷

