

LTE 媒體存取控制 (Medium Access Control)

通訊協定(2)

工研院資通所新興無線應用技術組 林佑恩 王淑賢

摘要

在上個月的 LTE 媒體存取控制 (Medium Access Control) 通訊協定(1)介紹文章中，我們了解了 MAC 子層在 LTE UU 介面中所處的位置與服務內容。除了上次介紹過的資料傳輸、錯誤重傳機制、與控制訊息發送等功能，這次我們將先針對 LTE MAC 的多工與解多工(即 MAC PDU 的組成)做詳細介紹，之後我們將介紹 MAC 層對連線的初始管理(隨機存取)與對資料的排程服務。

一、 MAC 協議數據單元(Protocol Data Unit, PDU)

由於 LTE MAC 層負責匹配與轉換 RLC 層的資料與 PHY 層的資料，在本章中我們將更詳細的說明 MAC SDU (即 logical channel 上承載的數據單元，例如 RLC PDU)與 MAC PDU (即 transport channel 上承載的數據單元，或稱為一個 PHY 層的 transport block)的對應關係。

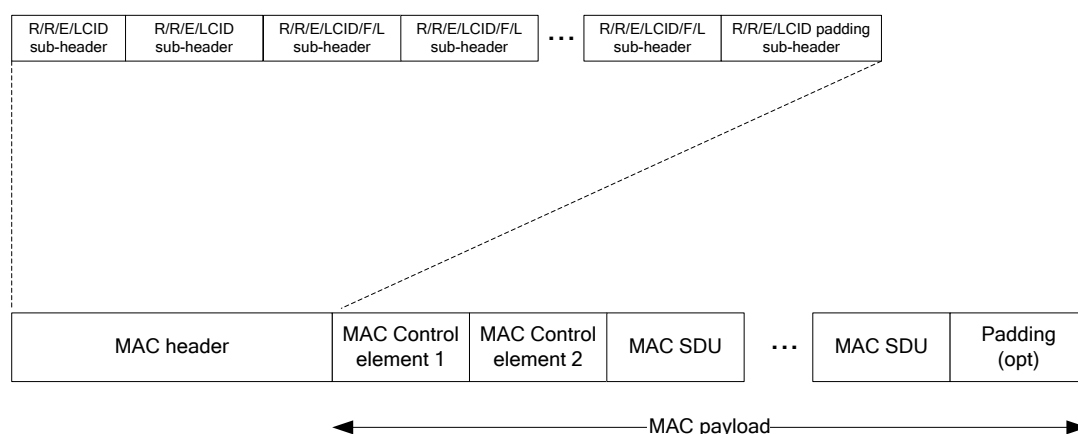
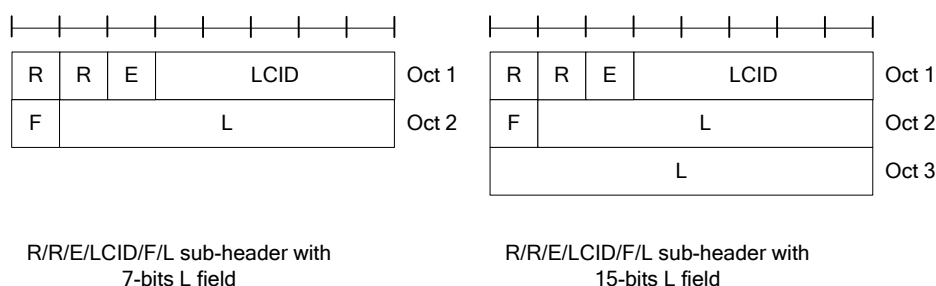


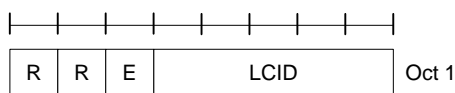
圖1 MAC PDU 架構圖

圖 1 說明了一個基本的 MAC PDU 架構，一個 MAC PDU 就相當於 PHY 層在每個 subframe (子禎，長度為 1 ms)的時間內所能傳送的 data，MAC PDU 的長度在

不同的時間點可能是不一樣的，會根據 PHY 層的通道狀況與資源配置方式而不同，但相同的是每個 MAC PDU 長度的基本單位為 1 Byte (8 bits)。MAC PDU 的基本組成為一個或多個的 MAC 子標頭(subheader)，搭配相對應的 MAC Payload：MAC 控制單元、MAC SDU、或是 padding。原則上每個 subheader 會對應一個 MAC 控制單元、MAC SDU、或是 padding，但是也有例外，比如：DRX 控制訊息就只有 subheader 沒有 payload 的部分；當同時有多個使用者設備發送 random access preamble 時，基站會在 Random Access Response 這個訊息的前端加上 Backoff indicator subheader；還有當 MAC 處於 transparent mode 時，MAC 會直接把一個 MAC SDU 原封不動地當成一個 MAC PDU 來傳送。



(a) R/R/E/LCID/F/L MAC 子標頭



(b) R/R/E/LCID MAC 子標頭

圖2 MAC 子標頭格式

圖 2 則顯示了最常見的兩大類 MAC subheader 格式，根據其對應的 Payload 種類與特性的不同，會選用不同的 subheader 來搭配。MAC subheader 必定會包含 E(Extension field)欄位與 LCID(logical channel identity)欄位(MAC RAR 則以 Preamble ID 作為辨識，在第二章節中將會說明)。當 E 欄位等於 1 時代表目前的 subheader 後仍有其他 subheader，反之則代表這是最後一個 subheader。而 LCID 的欄位則代表這個 subheader 與其對應的 MAC payload 所乘載的訊息種類，LCID 的值與其對應的意義如表 1 與表 2。當 MAC subheader 所對應的 MAC SDU 種類為像 RLC PDU 這種長度不一的數據單元時，需在 MAC subheader 中額外標示 MAC SDU 的長度，目前 LTE 支援的 MAC SDU 長度最多為 $2^{15}-1=65534$ Bytes，當 MAC SDU 長度落於 $0\sim 2^7-1$ Bytes 時，MAC subheader 會在 F(Format field)填上 0，接著用 7 bits 來表示 MAC SDU 的長度；若 MAC SDU 的長度落於 $2^7\sim 2^{15}-1$ 時，MAC subheader

的 F 欄位則填上 1，再用接下來的 15 bits 標示 MAC SDU 的長度。另外 subheader 中的 R(Reserved bit)欄位則為保留字元，一律填上 0。藉由 subheader 中所標示的各個欄位，MAC 在收到 PDU 後便可以準確地從其對應的 MAC SDU 中將訊息解多工出來。但 MAC PDU 的最後一個 SDU 若為 RLC PDU 的話，因為已是 MAC PDU 的最後一部分了，此 RLC PDU 所對應的 MAC subheader 便不需要標示其長度(長度就為前一個 MAC SDU 的結尾一直到 MAC PDU 的結尾)，因此最後一個 MAC SDU 的 Subheader 格式就固定為 R/R/E/LCID 共 1 Byte。

表1 DL-SCH 上的 LCID 欄位代表意義

LCID 欄位的二進位數值	代表意義
00000	Downlink CCCH
00001-01010	Identity of the logical channel
01011-11010	Reserved
11011	Activation/Deactivation
11100	UE Contention Resolution Identity
11101	Timing Advance Command
11110	DRX Command
11111	Padding

表2 UL-SCH 上的 LCID 欄位代表意義

LCID 欄位的二進位數值	代表意義
00000	Uplink CCCH
00001-01010	Identity of the logical channel
01011-11000	Reserved
11001	Extended Power Headroom Report
11010	Power Headroom Report
11011	C-RNTI
11100	Truncated BSR
11101	Short BSR
11110	Long BSR
11111	Padding

二、 隨機存取過程

隨機存取過程(Random access procedure)是由 UE 發起的程序，目的是為了讓

UE 可以與 eNB 建立 RRC 連線狀態，或是當連線中斷或雙方的傳送時間沒有同步在一起時，讓 UE 可以重新連線或是更新同步狀態。

為了讓 UE 知道如何進行 LTE 隨機存取過程，eNB 會先在廣播系統訊息中分配特定的頻段與時間，讓 UE 可以在這些通道資源上發送 Random Access Preamble (隨機存取過程的前導訊息，又被稱為 Msg 1)，如果 Preamble 有正確被 eNB 接收到，eNB 就會發送 Random Access Response (RAR, 又被稱作 Msg 2)，告知 UE 可以繼續發送後續的身分認證等訊息。

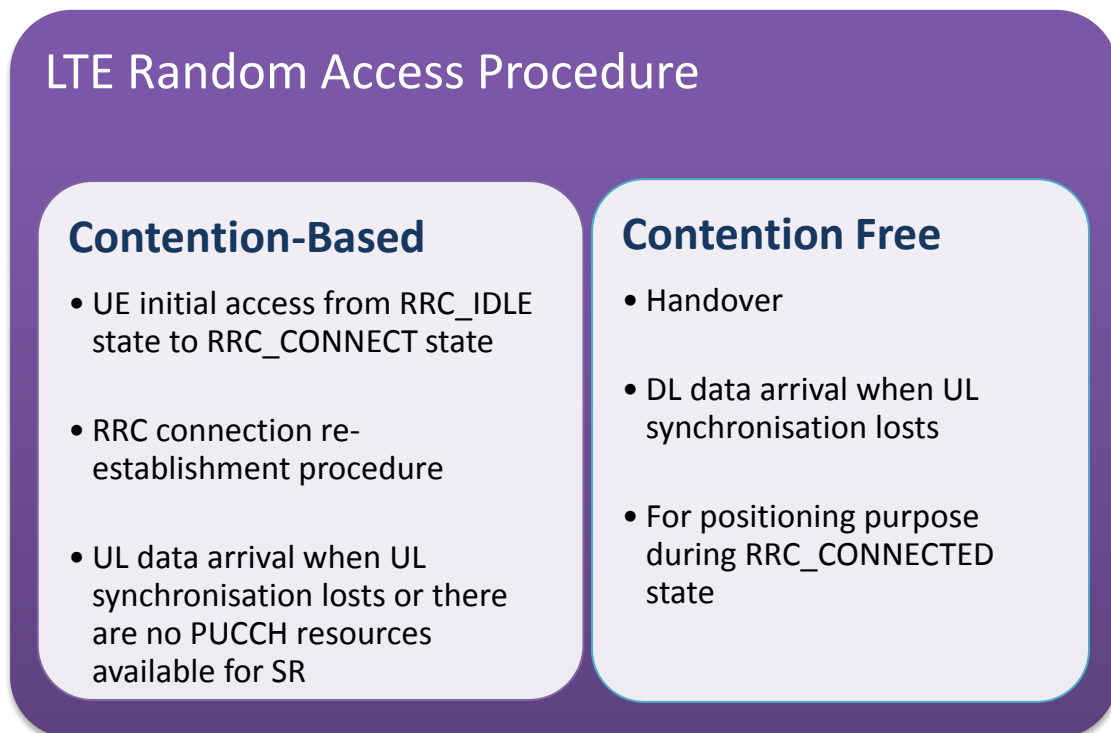


圖 3 競爭式與非競爭式隨機存取過程的觸發原因

根據不同的狀況與需求，隨機存取過程可分為競爭式(contention-based)或是非競爭式(contention free)，如圖 3 所示，此兩者的分別在於 UE 是否被告知特定的 preamble。例如 UE 初次連上 eNB，或是 UE 要傳資料卻發現與基地台失去同步或斷線，就會隨機從 eNB 廣播的系統訊息設定中，選定一個合法的 random access preamble 來進行 contention based 的 random access，在競爭式的隨機存取過程中，若有多個 UE 同時選到了同一個 preamble，就會造成碰撞，若基站無法解出 preamble 信號，可以通知 UE 進行隨機後退(random backoff)並再次嘗試發送 preamble；若 eNB 正確收到 preamble，就會在 RAR 中指定 UE 的暫時性身分辨識碼、上行同步參數、上行頻寬分配等等訊息，UE 可繼續發送本身的身分辨識(通常稱為 Msg 3)，並等待 eNB 發送 Contention Resolution 訊息與 RRC 連線訊息(通常稱為 Msg4)，以完成隨機存取的任務。除此之外，當 UE 需要換手(handover)到新的基站、或是基站要送資料給 UE 時發現 UE 不同步、或是 eNB 需要更新 UE

的位置.....等等狀況，eNB 會主動告知 UE 需要重新進行隨機存取過程，同時也會告知 UE 一個特定的 preamble 來進行連線，這時候這個 preamble 是唯一指定的，所以不會發生 preamble 碰撞的狀況，整個隨機存取過程都是 contention free，而且由於隨機存取過程是由基站觸發、UE 執行，所以基站不需要再請 UE 繼續發送身分認證訊息就已經知道 UE 的身份，非競爭式的隨機存取過程只需要交換到 RAR (即 Msg 2)即可結束。

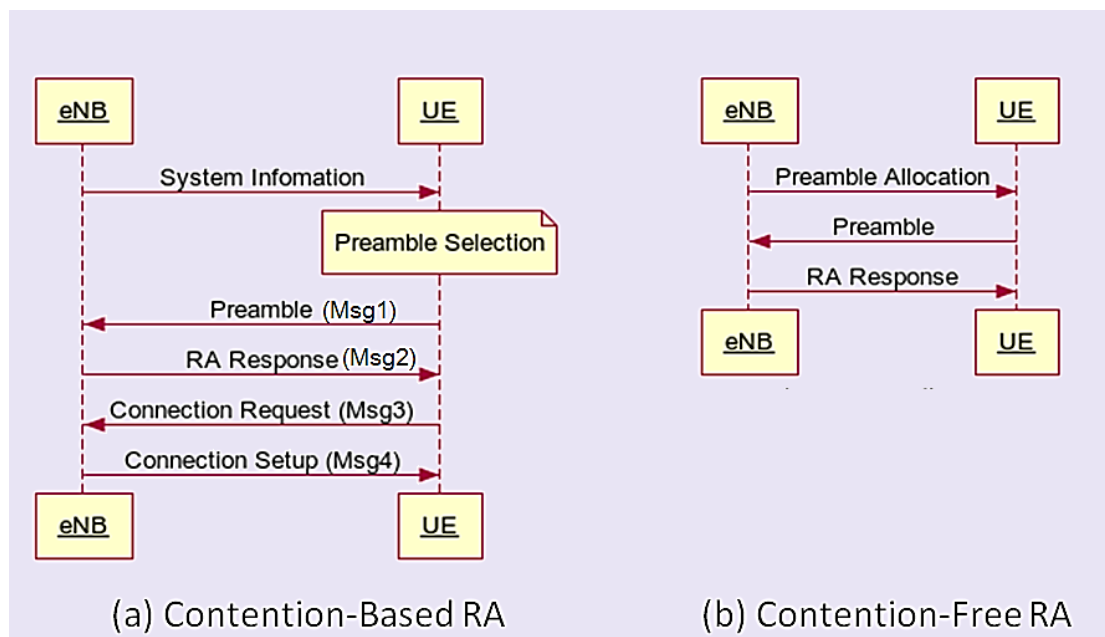


圖 4 基站(eNB)與使用者設備(UE)隨機存取過程中的訊息交換

圖 4 顯示 eNB 與 UE 端隨機存取過程的 message 交換流程圖，我們可以很清楚的看到 Msg1 至 Msg4 的交換互動，以下將再對每個 Msg 的用途與意義分別做詳細的介紹。

1. Preamble (Msg1)

LTE 規範了 64 組隨機存取過程可用的 preamble 讓 UE 來發起隨機存取過程，每一組 preamble 就對應於通道上某個頻段與時段的資源。eNB 在系統訊息中指定 Contention Based 的隨機存取過程可使用的 preamble 組合，而其他剩下來的 preamble 則會被保留為 Contention Free RA 使用。當 UE 想要發起隨機存取過程時，就會在 Contention Based 可用的 preamble 隨機選取一個，或是使用 eNB 先前指定的 Contention Free 的 preamble，在其對應的通道資源上發送前導訊息。如果 eNB 端接下來並沒有任何回應，則 UE 可以增加傳輸 power 再次發送 preamble，直到達到發送上限或是收到 eNB 的回應為止。隨機存取的 preamble 選取範圍、傳送 power、發送上限等等，都會載在 eNB

發送的 System Information 2 中。

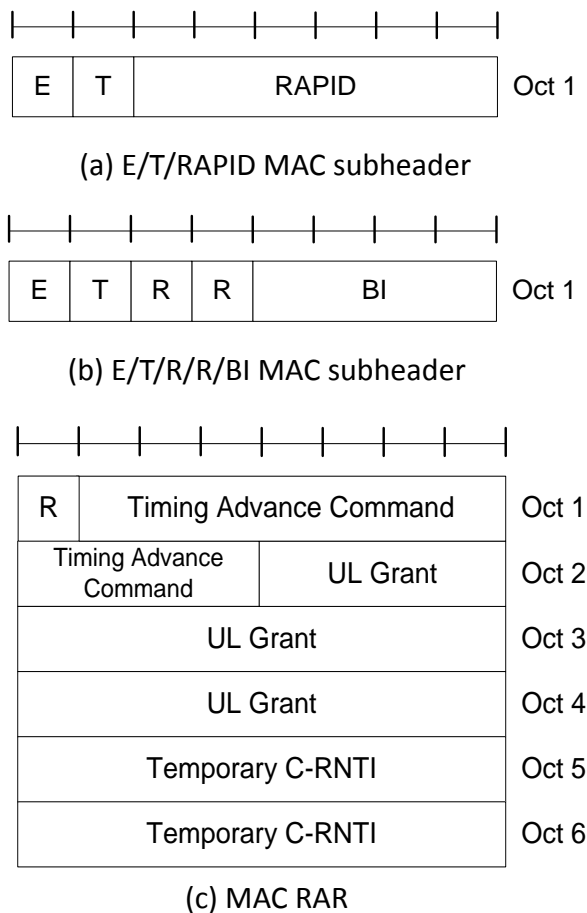


圖 5 MAC Random Access Response (RAR) 封包格式

2. Random Access Response (Msg2)

當 eNB 成功收到 UE 的 preamble 後，會發送 random access response 信息給 UE，其信息內會載明三條消息：(1). 上行同步指令(Timing Advance Command)，用於同步 UE 與 eNB 間上行資料的傳輸時間；(2). 上行通道資源(UL Grant)，用於告知 UE 下一筆上行資料的傳輸資源；(3). UE 臨時身分辨識碼(Temporary C-RNTI)，用於 UE 初次身分認證，當隨機存取過程順利結束後，此臨時身分認證就會變成 UE 與 eNB 間的正式的身分辨識。但如果 eNB 認為 preamble 發生碰撞或是目前沒有上行資源可以分配給 UE，會發送 Random Backoff 的訊息給 UE，請 UE 隨機地等待一段時間再次發送 preamble。RAR 的封包格式如圖 5 所示，其中圖 5 (a)與(b)是 eNB MAC 子層發送 RAR 的子標頭，當 eNB 成功收到某個 UE 的 preamble 後，會發送格式(a)的子標頭，載明此 UE 的 preamble index (0 到 63 共 64 個組合)，若 eNB 需要 UE 隨機後

退再次嘗試發送 **preamble**，則會發送格式(b)的子標頭。子標頭的 **T** 欄位則用來判別子標頭的格式(a)與(b)，其對應的 **T** 欄位要分別填上字元 **1** 與 **0**。而圖 5 (c)則為當 **preamble** 被 eNB 正確收到後，eNB 所回覆的 RAR 的載酬，載明上述與 UE 相關的三條消息。

3. RRC Connection Request (Msg3)

當 UE 收到 RAR 後，就會在其中指定的上行通道資源上發送 RRC Connection Request Message，一般稱作 Message 3。Message 3 中會帶有 UE 的身分辨識標誌與發起隨機存取過程的原因。UE 的身分辨識標誌一般來說可能是 UE 之前曾經向網路註冊過的 Temporary Mobile Subscriber Identity (TMSI)，TMSI 是具有地域性的編號，當 UE 移動到其他的網路下就會重新再取得一個新的 TMSI。而當 UE 在目前的網路下並沒有 TMSI 時，就會隨機地傳送 5 bytes 的亂數碼當作暫時性的身分辨識標誌。但無論 UE 傳送 TMSI 或是亂數碼當作身分辨識標誌，為了區別 UE 是否有順利地完成隨機存取過程，eNB 需要在 Contention Resolution 計時器截止前將 UE 的身分辨識標誌載在 MAC 控制訊息上回傳給 UE，作為區別 UE 的辨識訊息，讓 UE 判別隨機存取過程是否結束、與 Layer 2 連線是否建立完成。或許大家會認為 UE 既然收到 RAR，為何還需要利用 Contention Resolution 控制訊息來再次確認隨機存取過程是否順利結束呢？這是因為當多個 UE 同時進行隨機存取過程時，有可能兩個以上的 UE 會同時選擇同樣的 **preamble** 發送，也就是說這兩個 UE 會在相同的通道資源上發送一模一樣的隨機存取前導訊息。此時 eNB 若成功收到 **preamble**，只能針對收到的 **preamble** 發送 RAR 以安排 Message 3 的上行資源，無法區別此 **preamble** 到底是由哪個 UE 發送出來的。然而若兩個以上的 UE 也成功收到 RAR，就會在相同的上行資源上發送 Message 3，Message 3 中所帶有的 UE 身分辨識標誌在此時就會發揮作用，由於不同 UE 具有不同的身分辨識標誌(身分辨識標誌相同的機率趨近於 0)，因此 eNB 此時就可以選取其中一個 UE(甚至是不選取)來發送 Contention Resolution 控制訊息，因此可以保證唯有最多一個 UE 在此競爭中勝出並取得後續的通道資源與身分辨識，也唯有這個競爭勝出的 UE 可以將 Temporary C-RNTI 當作自己的 C-RNTI 進行後續的資料收送。

值得注意的是，Message 3 也是 UE 第一筆受到 HARQ 保護的上行訊息，當 UE 送出 Message 3 後，如未順利收到 eNB 的 ACK 回應，就會再次發送 Message 3 直到 UE 收到 eNB 回應或是達到 Msg3 的最大 HARQ 重傳次數。

4. RRC Connection Setup (Msg4)

當 eNB 收到 RRC Connection Request (Msg3)後，MAC 層會通知 RRC 層，若 RRC 可識別此 UE 並允許連線的話，就會發送 RRC Connection Setup 的訊息給 UE，此 Message 一般也稱作 Message 4。Message 4 帶有 SRB1 與 DRB

的訊息，讓 UE 與 eNB 端開始建立 Control Plane 與 User Plane 的訊息通道。除此之外，Message 4 也可能帶有一些上行通道的訊息，比如上行方向的 power control、UE 端 MAC 層的重要組態諸如計時器設定或是 HARQ 參數、UE 回報通道品質指標(Channel Quality Index, CQI)的組態、上行方向排程相關的組態諸如 Scheduling Request 的通道資源.....等等。

三、邏輯通道(logical channel)排程

資料排程是 MAC 子層很重要的任務，MAC 層必須要將 Layer 3 或更上層丟下來的控制或是資料封包組成 MAC PDU，並在適當的無線通道資源上發送。在以下連續三個章節裡，我們將詳細說明 MAC 子層的資料排程所需要執行的任務與考量，首先要介紹的是 MAC 子層如何決定每個邏輯通道的重要性與資料排程量，之後將討論 MAC 如何將這些資料安排在無線通道資源上發送。

在上個月的 LTE MAC 介紹文章中，我們曾經提過 MAC 層會從邏輯通道上取得上層的資料封包，多工為 MAC PDU 後，再傳遞給實體層傳送。LTE 中規範每個 UE 與 eNB 在做資料收送時，可以擁有編號 0 至 10 最多共 11 個邏輯通道，每個邏輯通道的 Priority 由 RRC 端指定，Priority 數值越小代表優先順序越高，當 UE 被分配到上傳通道資源時，原則上會依照每個邏輯通道的優先序抓取資料。然而，為了避免每次組 MAC PDU 時都從優先序高的邏輯通道抓取資料，造成優先序低的邏輯通道的資料永遠送不出去的狀況，LTE MAC 子層的協定針對每條邏輯通道，例如邏輯通道 j，定義了所謂的 B_j 值，讓 UE 計算每次該從每條邏輯通道中取出多少資料量。 B_j 值會隨著每個邏輯通道的 Prioritized Bit Rate (PBR)、Bucket Size Duration (BSD)、與時間而改變，會動態地計算每條邏輯通道的頻寬需求，並讓 MAC PDU 的多工處理可以盡量符合所有邏輯通道的頻寬需求。PBR 是每個邏輯通道所被分配到的頻寬，PBR 由上層指定，假設實體層的頻寬資源足夠的話，那麼每個邏輯通道在一個子訊框內就可以送出 PBR 乘上 10^{-3} 秒(即 1ms，一個子訊框的時間長度)這麼多的資料量。BSD 是每個邏輯通道一次最多可以預先儲備的資料量，假設 BSD=1 秒，則代表每個邏輯通道最多可以預儲備 PBR 乘上 1 秒的資料量，BSD 一樣是由上層指定。

根據 PBR 與 BSD 的定義，UE 的邏輯通道 j 一開始的 B_j 值會初始為 0，之後每個子訊框會將 B_j 累加 PBR 乘上 10^{-3} 秒，但是 B_j 值若超過 PBR 乘上 BSD 則不可以再往上加，一律維持在 PBR 乘上 BSD 這個數量。當 UE 獲得上傳頻寬時，就會依照邏輯通道的 Priority 順序開始抓取資料，針對第 j 條邏輯通道抓取適量的 RLC PDU 後，就會將第 j 條的 B_j 值扣掉方才抓取的資料量。 B_j 值有可能因為一次抓取太多 RLC PDU 而變成負值，當 B_j 變為負值時，該邏輯通道就暫時不能發送資料，要直到 B_j 值隨著時間上升為正值之後才能再次從中抓取資料傳送。若當所有邏

輯通道都抓取一輪後仍有剩餘的上傳頻寬，則會再輪回優先序高的邏輯通道重新抓取資料，第二輪的資料抓取則不必考慮 Bj 值，直接按照邏輯通道的優先序抓取 RLC PDU 直到上傳頻寬被耗盡為止。只要 UE 根據這個原則抓取資料的話，當實體層有足夠頻寬時，每條邏輯通道的傳輸量都可以滿足上層所分配的 PBR，也不會造成優先序高的邏輯通道永遠占滿頻寬、優先序低的邏輯通道無法傳送的狀況。

MAC 除了從邏輯通道抓取資料組成 MAC PDU，也會將 MAC 子層的控制訊息放入 PDU 中，部分的 MAC 控制訊息其排程優先序是比邏輯通道還要高的，例如以上行方向的排程上，LTE 規定 UE 應該以以下的順序由高至低來安排 MAC 控制訊息與邏輯通道資料間的優先序：

- (1). C-RNTI MAC 控制單元與 CCCH 邏輯通道資料；
- (2). BSR MAC 控制單元 (不包含 padding BSR，即利用原本要發送 padding 的多餘空間發出的 BSR)；
- (3). PHR MAC 控制單元；
- (4). 除了 CCCH 以外的邏輯通道資料；
- (5). Padding BSR MAC 控制單元。

當考量以上所述的優先序之後，UE 就可以適當的將各類訊息公平地發送至 eNB。

四、動態資料排程

MAC 子層的重要工作之一就是處理無線資源的分配，使得 MAC 的 PDUs 可以在無線資源上順利地被送出和接收。在了解無線資源分配之前，首先必需先了解在 LTE 的規範下，無線資源是如何被規劃的。

1. 無線資源(Radio Resource)

無線資源最基本可被區分成兩個維度：時域(Time Domain)和頻域(Frequency Domain)。從時域來看，LTE 在時域上的運作是以系統幀(System Frame)為最大單位，如圖 66 所示，一張系統幀是以 10 毫秒(ms)為單位，系統幀的號碼由 0、1、...編號至 1023，共 1024 個號碼可供重覆循環使用，因此約每 10 秒系統幀的編號會重新開始。

在系統幀下又可切割成 10 張子幀，從子幀 0、1、...到子幀 9，共 10 張子幀，每張子幀長度為 1 毫秒。在 LTE 中，傳輸時間間隔(Transmission Time Interval; TTI)恰為一張子幀，即 1 毫秒。換句話說，對於一支用戶端裝置(UE)而言，它是以一張張的子幀作為單位來接收 MAC PDUs。或者說，接收、傳送一筆 MAC PDU 需固定花 1 毫秒的時間，然而 MAC PDU 有大有小，這就取決於使用的調變編碼的階數(Modulation and Coding Scheme)，以及頻域上使用資源的多寡，這部分稍後再作介紹。

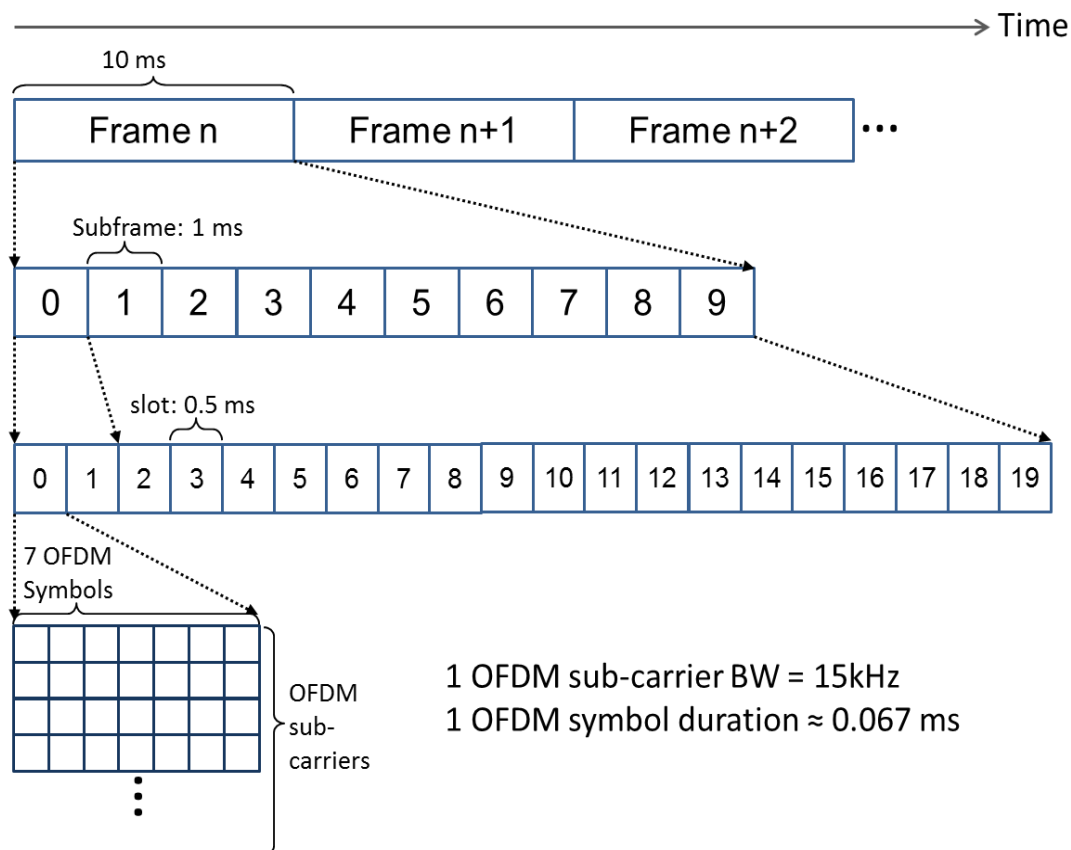


圖 6 LTE 無線資源在時域上的規劃圖

在一張子幀下又可再切割成 2 個時槽(Time Slot)，每個時槽為 0.5 毫秒。一張系統幀內的時槽共有 20 個，分別由編號 0、1、...直到編號 19。一個時槽的時間長度恰等同於一個資源方塊(Resource Block; RB)的時間長度。再深入拆解到 OFDM 的 symbol 長度時，可發現在一般 Cyclic prefix 長度下，一個時槽(或資源方塊)的時間長度是由 7 個 OFDM symbol 所組成，一個 OFDM symbol 長度大約是 0.067 毫秒。此外，一個 OFDM 的子載波(OFDM Sub-carrier)的頻寬是 15kHz，其倒數約為 0.067 毫秒，正符合 OFDM 的 symbol 的時間長度。

由於 LTE 的物理層使用 OFDM 的傳輸技術，因此從頻域方面來看無線資源，則頻域被切割成數個正交獨立的子載波，即每個 OFDM 子載波可承載不同的 bit stream 同時傳輸。

在了解 LTE 下無線資源的切割及規劃方式後，對於什麼是資源元素(Resource Element; RE)、什麼是資源方塊(Resource Block)的定義就比較容易了解。

所謂的資源元素(RE)就是定義在時域上 1 個 OFDM symbol 的長度，且在頻域上佔用 1 個 OFDM 子載波的無線資源，即 $0.067 \text{ ms} \times 15 \text{ kHz}$ 。在一個資源元素上，若使用的調變階數為 2 階(例如：QPSK)，則一個資

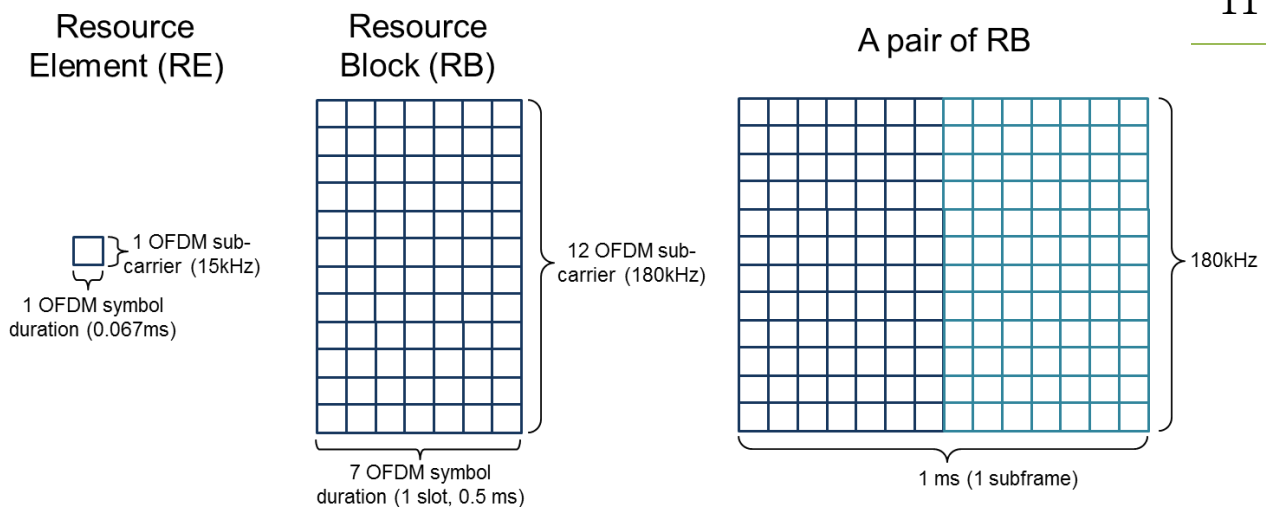


圖 7 資源元素(Resource Element)及資源方塊(Resource Block)的示意圖

源元素可承載 2 個 bits；若使用的調變階數為 6 階(例如：64 QAM)，則一個資源元素可承載 6 個 bits。

所謂的資源方塊(RB)就是在時域和頻域上連續的資源元素(RE)所形成的集合。在 LTE 的規範中，其定義一個資源方塊就是頻域上連續 12 個 OFDM 子載波，且在時域上連續 7 個 OFDM symbol 的時間長。如圖 7 所示，即一個資源方塊是由 84 (12 x 7)個資源元素所組成。若每個資源元素可承載 6 bits，則一個資源方塊可承載 63 bytes。

在 LTE 的範規中，對於下行的物理共享通道(Physical Downlink Shared Channel; PDSCH)或上行的物理共享通道(Physical Uplink Shared Channel; PUSCH)，所採取的無線資源分配之最小單位是以 1 對資源方塊進行分配。所謂的 1 對資源方塊指的是在時間軸上兩個相鄰的資源方塊，如圖 所示。由於 1 對資源方塊是佔 1 毫秒，等同於 1 個系統子幀或 1 個 TTI 長度，所以 MAC 層的資源分配在每個 TTI 內只需(只能)在頻域上挑選不同的資源方塊對，而不進行時間軸上資源方塊的挑選及配置，也因此 MAC 在下行控制訊息(Downlink Control Information; DCI)中的資源指示值(Resource Indication Value; RIV)只描述資源方塊在頻域位置上的編號。

1. 控制通道的資源分配

基站端是透過下行的控制通道(Physical Downlink Control Channel; PDCCH)來告知用戶端裝置(UE)下行和上行資料通道(PDSCH 和 PUSCH)的無線資源分配結果，這支用來告知 UE 的訊息稱作「下行控制訊息」(DCI)。DCI 是經由 PDCCH 來傳送給 UE 端，而 PDCCH 在無線資源的分配在是佔每個下行子幀的前 1 到 3 個 OFDM symbol 數，取決於有多少 DCI 需要

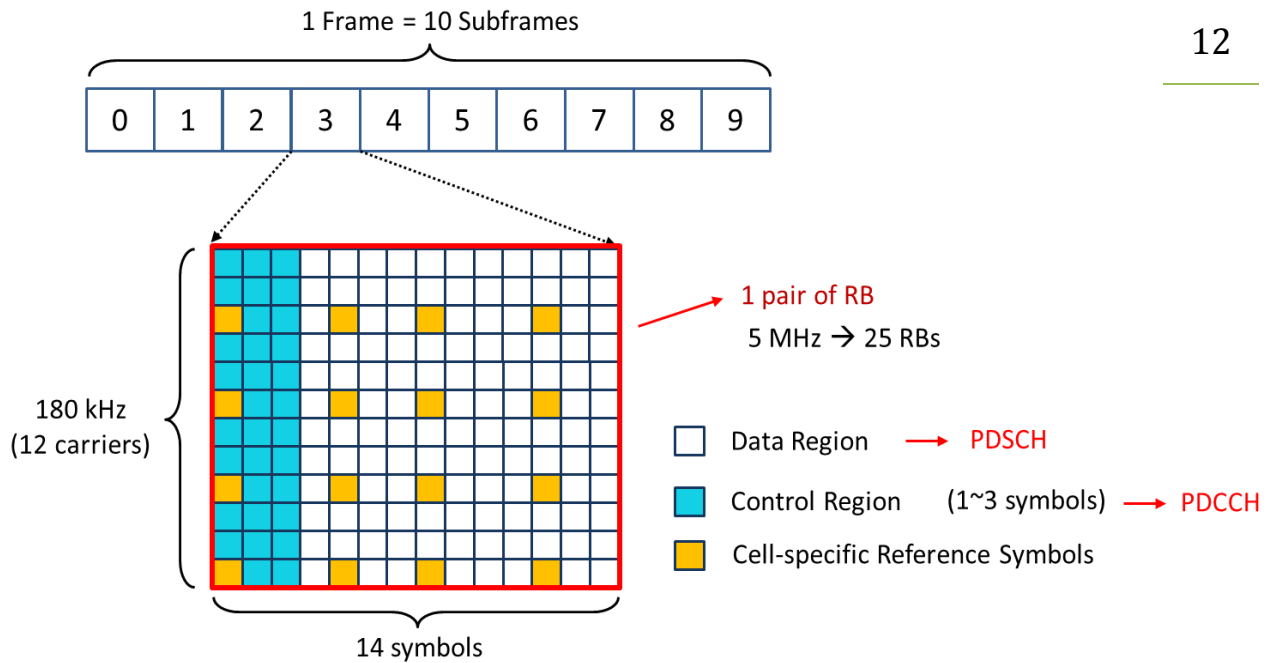


圖 8 下行的控制通道和資料通道在無線資源上的規劃

傳送。如圖 8 所示，在此例子中，一張下行子幀的前 3 個 OFDM symbols 被規劃成 PDCCH。每張子幀 PDCCH 所佔用的 OFDM symbol 數可隨需求做調整，佔用的 OFDM symbol 數經由 PCFICH(Physical Control Format Indicator Channel)告知，PCFICH 固定於子幀的第一個 OFDM symbol 發送。

在一張子幀決定 PDCCH 所使用的 OFDM symbol 數後，便可算出有多少資源元素(RE)落於 PDCCH。在 LTE 中，定義每 4 個資源元素成為 1 個資源元素組(RE Group; REG)，並將 9 個資源元素組(REG)定義成 1 個控制通道元素(Control Channel Element; CCE)。在下行控制通道(PDCCH)上，控制通道元素(CCE)為最基本的資源分配單位。當有一筆下行控制訊息(DCI)要傳送時，就需要分配適當的 CCE 數量供其傳輸使用。

分配的 CCE 數量需要遵守一定的規範，如表 3 所示，對於共通區段(Common Region)內所要發放的下行控制訊息，一筆訊息只可分配 4 個或 8 個 CCE 用來傳輸，又稱作「聚合級」(Aggregation Level)。而對於 UE 特定(UE-specific Region)的下行控制訊息，則可分配 1、2、4 或 8 個不同數量的 CCE 來傳輸。

此外，所分配的 CCE 位置起始位置只能在聚合級的倍數，並按照一定的公式配置。如在無載波聚合的情況下，一筆下行控制訊息的起始位置，由 UE 的 C-RNTI 及所在的 subframe number 所決定：

C-RNTI=257, subframe 0

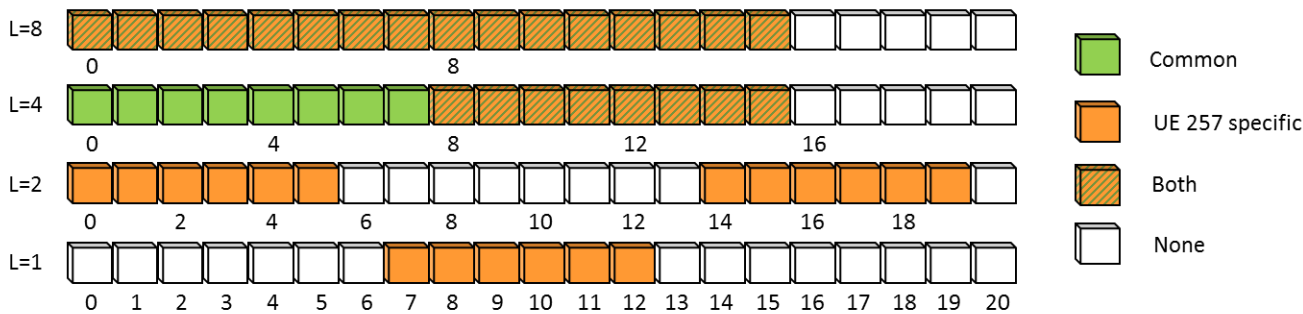


圖 9 UE C-RNTI = 257 在 subframe 0 時，各種聚合級下可能分配 CCE 的位置分布。

$$L \left\{ (Y_k + m) \bmod \left\lfloor N_{\text{CCE},k} / L \right\rfloor \right\}$$

$$Y_k = (39827 \cdot Y_{k-1}) \bmod 65537$$

$$Y_{-1} = \text{CRNTI}$$

其中的 k 表示所在的 subframe 編號， m 表示第 m 個候選的 CCE 起始位置。圖 9 說明一個 UE 其 C-RNTI 為 257 在 subframe 0 時，各種聚合級下可能被分配到 CCE 的位置分佈，其中共通區段和特定區段可能彼此重疊。

UE 端依照表 3 中可能被分配到的 CCE 位置，在其上進行搜尋，尋找可能的下行控制訊息。若 UE 能順利解出下行控制訊息，則可獲取下行或上行資料通道上的資源分配結果。

2. 資料通道的資源分配

在 LTE 的規範中定義了 3 種不同的資源方塊分配方式來分配無線資源給資料進行傳輸，其名稱分別是 Type 0、Type 1 及 Type 2。以下將針對三種不同的資源分配方式作介紹。

表 3 PDCCH candidates monitored by a UE (TS36.213 Table 9.1.1-1)

Type	Search space		Number of PDCCH candidates (m)
	Aggregation level (L)	Region Size [in CCEs]	
UE-specific	1	6	6
	2	12	6
	4	8	2
	8	16	2
Common	4	16	4
	8	16	2

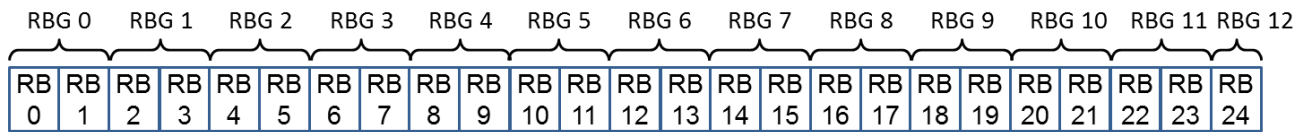


圖 10 頻寬 5MHz 下，25 個資源方塊分組成 13 個資源方塊組的情況

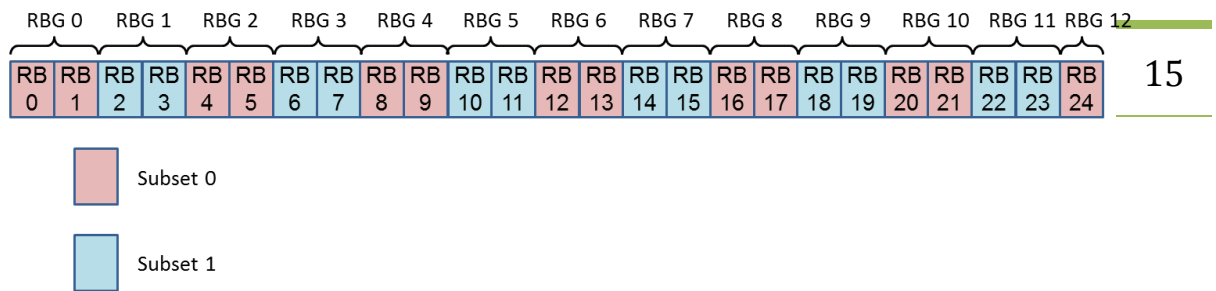
表 4 不同頻寬下資源方塊組的大小

System Bandwidth	RBG Size
N_{RB}^{DL}	(P)
≤ 10	1
11 – 26	2
27 – 63	3
64 – 110	4

在 Type 0 中，一張系統子幀中的資源方塊被分組成數個資源方塊組(Resource Block Group; RBG)，一個資源方塊組的大小取決於系統所使用的頻寬，如表 4 所示。

在資源方塊分組成資源方塊組的過程中，若最後一個資源方塊組內的資源方塊數不足以達到規定的資源方塊組大小時，仍然以一個資源方塊組自居。以 5Mhz 的頻寬為例，一共有 25 個資源方塊可用，每兩個資源方塊組一個資源方塊組，所以一共有 13 個資源方塊組，其中最後一個資源方塊組(第 13 組)的大小只有一個資源方塊，如圖 10 所示。資源方塊及資源方塊組的編號是依順序由頻域軸上的低頻至高頻進行編號。

在 Type 0 的資源分配過程中，只能以資源方塊組為單位進行分配，若是需要傳輸的資料量是小而零散時，就會浪費較多的資源方塊。特別是在大頻寬下，原本用 1 個資源方塊就可傳送完的資料，卻需要用 4 資源方塊所形成的資源方塊組來傳送，浪費了 3 個資源方塊。



15

圖 11 頻寬 5MHz 下，Type 1 將 25 個資源方塊分成 2 個子集合的情況

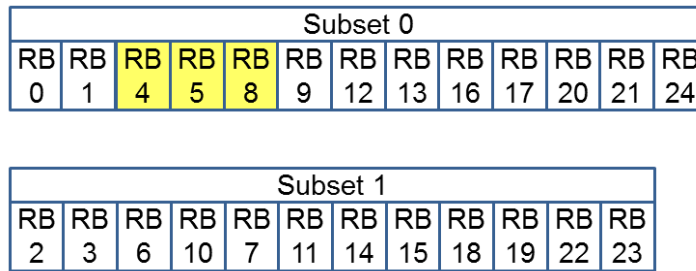


圖 12 在 Type 1 資源分配中，子集合內的資源方塊可各別定址

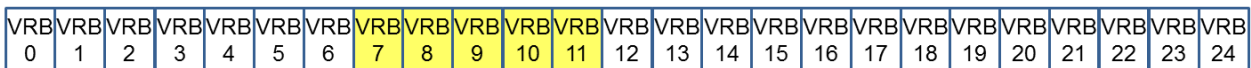


圖 13 Type 2 的資源分配方式：指定虛擬資源方塊的起始位置及使用的數量

在 Type 1 的資源分配中，改善了 Type 0 中資源分配顆粒較粗的問題，但付出較高的複雜度來維護資料結構。首先，資源方塊會被分成資源方塊子集合(Subset)，子集合的數量隨系統頻寬調整，數量上如同資源方塊組的大小，如表 4。在決定完子集合的數量後，再依序分配各個資源方塊組至子集合內，分配的方式以 Round Robin 的方式進行。以 5MHz 的 25 個資源方塊為例，其會被分割成 2 個子集合，如圖 11 所示，在子集合 0 中會包含資源方塊組 0、2、4 … 12；而子集合 1 中會包含資源方塊組 1、3、5 … 11。在一個子集合內，資源分配的最小單位是資源方塊，而不是資源方塊組，與 Type 0 相比，Type 1 是比較細緻的資源分配方式。在規劃好資源分配的資料結構後，在分配資源給 UE 時，最先開始需決定使用那一個子集合，在決定子集合後，再決定使用該子集合內的那一些資源方塊。以圖 12 為例，子集合 0 先被選取，然後子集合 0 中的資源方塊 4、5、8 被分配給 UE 作為資料傳輸使用。

最後一種資源分配方式是 Type 2，在 Type2 中引入了虛擬資源方塊 (Virtual Resource Block; VRB)的概念，對於 MAC 層而言，資源方塊已被抽象化。MAC 層只需指定欲使用的資源方塊起始編號，以及使用資源方塊的個數，不需管實際上資料是放在那一個實體的資源方塊來傳輸，

例如：是位於高頻的資源方塊，還是低頻的？至於虛擬資源方塊如何對應到實體資源方塊則是由 PHY 層依照一定的規則來安排。MAC 層為了方便管理資源的使用量，其可規劃出等同實體資源方塊數量的虛擬資源方塊，以 N_{RB} 表示。在分配資源時，被分配的資源之起始位置可以是 $0 \sim N_{RB}-1$ ，使用的資源數量則可由起始位置開始連續分配，最多直到結尾。換句話說，被分配的資源數量可能是 $1 \sim N_{RB}$ 。以圖 13 為例，Type 2 的資源分配方式可指定由資源方塊 7 的位置開始，連續分配 5 個資源方塊 (VRB 7、8、9、10、11) 給某一個 UE 作為資料傳輸使用。在三種不同的資源分配方式中，對 MAC 層而言，Type 2 是最簡單又兼具彈性的一種方式。

五、 Semi-Persistent Scheduling (SPS) 半靜態資料排程

大部分無線通訊網路的封包傳輸模式，都是採用動態資料排程 (Dynamic Scheduling)，當有資料要傳送的時候，才去索要通道資源與頻寬。例如在 LTE 規範中，UE 要發送上行的資料時是用 Scheduling Request 或是觸發 Random Access Procedure 的方式來要上行的頻寬，而下行資料 eNB 會在實體層的 Physical Downlink Control Channel (PDCCH) 發送下行資料的控制訊息 (Downlink Control Information, DCI) 或是 Paging Information，UE 會固定去收聽實體層 PDCCH 發出的下行頻寬訊息，然後去指定的通道資源上接收屬於自己的下行資料。

但 LTE 上若乘載類似 voice 的訊息，dynamic scheduling 的方式並不適合，因為此種方式在使用者的通話期間會持續地要空中資源，並且 voice 本身的資料量不大、週期固定，反而會造成 PDCCH 上控制訊息發送的大量負擔。

使用半靜態資料排程 (SPS) 的理由就是針對類似 voice 這種週期較固定的小封包資料，在固定時間內進行週期性排程，eNB 與 UE 間只需要進行第一次與最後一次空中資源的排程控制訊息交換，就可以在固定位置與時間的空中資源上逕行發送上/下行資料，中間不需要再針對每筆上/下行資料交換排程控制訊息。當發現沒有週期性小封包要發送時，就可以釋放空中資源，等到有類似特性的資料量則再次請求 SPS 即可，這樣即可省下大量控制訊息的發送。

1. SPS 組態管理

原則上 LTE 系統中 SPS 的組態、啟動、與釋放是由 eNB 端的 RRC 層所控制，RRC 隨時都可以進行 SPS 的組態管理、啟動、與釋放，上行方向與下行方向的 SPS 可以分別進行組態管理，亦不需要同時啟動或釋放。RRC 會發送 SPS 的以下組態給 Layer 2 以下的模組，用於 SPS 資源

管理與流程控制：

- SPS C-RNTI：為了與一般的動態排程有所區隔，具有 SPS 能力的 UE 會使用另外一個身分認證，我們稱作 SPS C-RNTI，用來進行 SPS 的啟動、釋放、與週期性的資料收送。
- SPS 下行方向的組態參數，包含 SPSIntervalDL 與 numberOfConfSPS-Processes：SPSIntervalDL 指明了下行方向 SPS 傳輸的週期，最小可為 10 個子禎(10 ms)，最大可為 640 個子禎(640 ms)；而 numberOfConfSPS-Processes 則指定 SPS 所使用的 HARQ Process ID 數量。動態排程與半靜態排程的資料都可以受到 HARQ 的保護，也會共用所有的 HARQ Process ID，不過動態排程的優先序比半靜態排程還高，若半靜態排程的 HARQ Process ID 或空中資源被指定給動態排程的資料使用，則半靜態排程的資料就暫時無法發送。
- SPS 上行方向的組態參數，包含 SPSIntervalUL、implicitReleaseAfter、twoIntervalsConfig 三大參數：SPSIntervalUL 指明了上行方向 SPS 傳輸的週期，最大最小值一樣是 10 個子禎與 640 個子禎(10 ms 至 640 ms)；implicitReleaseAfter 提供 UE 端主動要求釋放 SPS 資源的管道，當 eNB 端的 RRC 層啟動了上行方向的 SPS 但是 UE 端發現沒有多餘 SPS 資料要上傳時，就會在每次 SPS 的資源上發送全部 padding 的 MAC PDU，eNB 端會累計上行方向的全部 padding 的 MAC PDU 數量，當達到 implicitReleaseAfter 的數量後就會釋放 uplink SPS 資源。twoIntervalsConfig 是為了方便 TDD 模式下的 SPS 傳輸，由於 TDD 模式的上/下行組態每 10 ms 會重複一次，因此 SPS 的排程也必須要考慮 TDD 的這種特性來設計資料週期。

根據 RRC 給予的組態，當 SPS 在系統時間第 SFN_{start} 個 Frame 與第 $subframe_{start}$ 個子禎被啟動時，第 N 個下行方向的傳輸資源會在第 SFN 個 Frame 與第 $subframe$ 個子禎上發生，其對應的計算公式如下：

$$(10 * SFN + subframe) =$$

$$[(10 * SFN_{start} + subframe_{start}) + N * SPSIntervalDL] \text{ modulo } 10240$$

另一方面，若是上行 SPS 在第 SFN_{start} 個 Frame 與第 $subframe_{start}$ 個子禎被啟動時，第 N 個上行方向的傳輸資源會在第 SFN 個 Frame 與第 $subframe$ 個子禎上發生，其對應的計算公式如下：

$$(10 * SFN + subframe) =$$

$$[(10 * SFN_{start} + subframe_{start}) + N * SPSIntervalUL +$$

$$Subframe_Offset * (N \% 2)] \text{ modulo } 10240$$

其中 Subframe_Offset 的參數當 twoIntervalsConfig 沒有被觸發時設為 0，當 twoIntervalsConfig 被觸發時則 Subframe_Offset 的值需參考表 5。

值得注意的是，RRC 發送 SPS 的組態並不同於要馬上啟動 SPS 模式，SPS 的啟動與釋放都需要再額外進行通知，SPS 的組態也可以重新發送、再次啟動與釋放，是很彈性的資料排程模式。

表 5 SPS 下行方向 Subframe_Offset 值表

TDD UL/DL configuration	Position of initial Semi-Persistent grant	Subframe_Offset value (ms)
0	N/A	0
1	Subframes 2 and 7	1
	Subframes 3 and 8	-1
2	Subframe 2	5
	Subframe 7	-5
3	Subframes 2 and 3	1
	Subframe 4	-2
4	Subframe 2	1
	Subframe 3	-1
5	N/A	0
6	N/A	0

2. SPS 資源的啟動與釋放

在收到 RRC 層的 SPS 組態後，只代表 eNB 與 UE 之間可以建立 SPS 連線與資源管理，要等到 eNB 在 PDCCH 上送出特殊的下行控制信令給特定的 SPS C-RNTI 後，才會開始啟動 SPS 的連線。同樣的，釋放 SPS 資源也會藉由特殊的下行控制信令來觸發。這些特殊的 PDCCH 控制信令有一個共通的特點就是 New Data Indicator (NDI) 的欄位都固定填 0，除此之外，表 6 與表 7 分別列出啟動與釋放 SPS 資源的 PDCCH 控制信令格式的其餘重要欄位，我們可以看到，除了 DCI format 0 是用於 uplink 資源分配所以沒有 HARQ process ID 欄位之外，其餘的 DCI 信令中的

HARQ process ID 欄位也一律填 0。而傳送下行 SPS 資料所對應的 HARQ process ID 則根據以下公式計算：

$$\text{HARQ Process ID} = \left[\text{floor}(\text{CURRENT_TTI} / (\text{SPSIntervalDL})) \right] \text{ modulo } \text{numberOfConfSPS-Processes}$$

上述的 $\text{CURRENT_TTI} = [(\text{SFN} * 10) + \text{subframe}]$ 。而上行方向的 HARQProcess ID 則一樣按照 subframe 的值給定，無需特別換算。

另外要注意的是 DCI format 2 系列的格式因為 MIMO 支援同時發送兩個 MAC PDU，因此一個 DCI 信令會分成兩大部分，分別指明兩個 MAC PDU 的組態，因此某個 MAC PDU 的實體層資源要啟動 SPS 時，需在相對應的 MAC PDU 的 DCI 欄位上設定特殊格式。而下行 SPS 所對應的 HARQ 需要在 Physical Uplink Control Channel (PUCCH) 的那些資源上周期性的傳送，則一併在 DCI 的 TPC command 中給予，上層會指定 4 個可供使用的 PUCCH 資源組態，而 TPC command 的值與其對應的 PUCCH 資源如表 8 所示。

表 6 PDCCH 上啟動 SPS 的控制信令特殊格式

	DCI format 0	DCI format 1/1A	DCI format 2/2A/2B/2C/2D
TPC command for scheduled PUSCH	set to '00'	N/A	N/A
Cyclic shift DM RS	set to '000'	N/A	N/A
Modulation and coding scheme and redundancy version	MSB is set to '0'	N/A	N/A
HARQ process number	N/A	FDD: set to '000' TDD: set to '0000'	FDD: set to '000' TDD: set to '0000'
Modulation and coding scheme	N/A	MSB is set to '0'	For the enabled transport block: MSB is set to '0'
Redundancy version	N/A	set to '00'	For the enabled transport block: set to '00'

表 7 PDCCH 上釋放 SPS 的控制信令特殊格式

	DCI format 0	DCI format 1A
TPC command for scheduled PUSCH	set to '00'	N/A
Cyclic shift DM RS	set to '000'	N/A
Modulation and coding scheme and redundancy version	set to '11111'	N/A
Resource block assignment and hopping resource allocation	Set to all '1's	N/A
HARQ process number	N/A	FDD: set to '000' TDD: set to '0000'
Modulation and coding scheme	N/A	set to '11111'
Redundancy version	N/A	set to '00'
Resource block assignment	N/A	Set to all '1's

表 8 TPC Command 數值代表的 PUCCH 資源

Value of 'TPC command for PUCCH'	$n_{\text{PUCCH}}^{(1,p)}$
'00'	The first PUCCH resource value configured by the higher layers
'01'	The second PUCCH resource value configured by the higher layers
'10'	The third PUCCH resource value configured by the higher layers
'11'	The fourth PUCCH resource value configured by the higher layers

SPS 的釋放除了由 eNB 端藉由 DCI 的特殊信令來觸發，如同之前所說的，也可以由 UE 觸發，當 UE 發送連續的全部 padding 的 MAC PDU 次數達到 implicitReleaseAfter 後，SPS 的資源就要馬上釋出。

3. SPS 資料的混和式錯誤重傳

透過 SPS 資源所發送的資料若發生接收錯誤，當傳送端收到 HARQ NAK 後依然需要準備重傳的動作。在下行方向的資料混和式錯誤重傳因為採用異步式(asynchronous)且非自適應式(non-adaptive)的機制，因此 SPS

資料若發生重傳，就要使用動態排程的方式另外安排重傳的資源。在上行方向的 SPS 資料若遇到 HARQ NAK 則因為可採用自適應或是非自適應式重傳，所以可以選擇繼續在原本的 SPS 資源上重新傳送或是另行以動態排程的方式來重傳。而由於 SPS 新傳資料的 NDI 欄位都是填 0，所以當安排 SPS 重傳資源時，NDI 欄位就會改為 1，用以標註這份資料為 SPS 重傳。

六、 結論

藉由這兩個月的 LTE MAC 通訊協定介紹，我們了解了 MAC 子層的目的是提供高效能的資料傳輸與網路存取管理的服務，針對 MAC 層所提供的各項服務細節，例如：邏輯通道與傳輸通道的對應關係、MAC PDU 的組成、隨機存取過程、錯誤重傳機制、資料排程服務.....等等，我們也一一地作了詳細介紹。在近幾年與未來幾年的規劃上，LTE 的 MAC 子層也將開發更多元的服務，諸如能提供更大頻寬的載波聚合(Carrier Aggregation)、或是讓使用者更省電的 Discontinuous Reception (DRX) 功能.....等等，這些更新更優化的技術，也將會讓 MAC 子層在 LTE 的通訊棧中更顯重要，亦值得大家多多關注 MAC 子層的新興技術。

參考文獻

- [1] 3GPP TS 36.300: "Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN) Overall description," v12.0.0, December, 2013.
- [2] 3GPP TS 36.321: "Evolved Universal Terrestrial Radio Access (E-UTRA) Medium Access Control (MAC) protocol specification (Release 12)," v12.0.0, December, 2013.
- [3] 3GPP TS 36.213: "Evolved Universal Terrestrial Radio Access (E-UTRA) Physical layer procedures (Release 12)," v12.0.0, December, 2013.
- [4] 3GPP TS 36.331: "Evolved Universal Terrestrial Radio Access (E-UTRA) Radio Resource Control (RRC) Protocol specification (Release 12)" v12.0.0, December, 2013.