

## 第十六章 Bluetooth 網路

由藍牙技術的應用領域到運作原理，本章都有詳細解說。

### 16-1 Bluetooth 技術簡介

『藍芽』( Bluetooth ) 技術最早是由 Ericsson 及 Nokia 兩家公司為連結行動電話及其它可攜式裝置，而共同發展的短距離無線通訊技術。為了使一般產品也能互通，在 1998 年 5 月，由 Ericsson、Nokia、Intel、Toshiba 及 IBM 等五家公司共同成立 Bluetooth SIG ( Special Interest Group )，之後又有 3COM、Lucent、Microsoft 及 Motorola 等公司相繼加入，共謀藍芽大業，希望共同建立新一代的短程無線通訊標準。基本上 Bluetooth Specification 是公開的，有意學習 Bluetooth 技術的讀者，甚至可將規格書下載下來，Bluetooth 官方網站為：

<http://www.bluetooth.org/specifications.htm>

『Bluetooth』名稱係取自西元 10 世紀丹麥國王 Harald II 的名字，紀念他統一丹麥和瑞典的貢獻。在藍芽的早期發展過程中，是由 Ericsson 負責無線射頻及基頻技術的研發，Nokia 則發展無線技術與行動電話之間所使用的軟體；另一方面，由 Intel 負責半導體晶片及傳輸軟體的開發，Toshiba 及 IBM 負責開發攜帶式電腦介面規格。經過這幾年的努力及推廣，全球大部份的資訊、通訊、半導體、消費性電子、網路及汽車等製造廠商，也都相繼加入為 Bluetooth SIG 會員，Bluetooth 儼然成為目前消費性產品最主要的通訊技術。另一方面，雖然 Bluetooth 也提出區域網路規範，但主要訴求是短距離之間的通訊，與 IEEE 802.11 系列有所區別；再者，若能將 Bluetooth 及 IEEE 802.11 網路連結起來，網路功能便能更深入到一般性消費產品，進而使網路應用更加廣闊。

藍芽技術到底能提供哪些服務，這是許多初學者迫切希望知道的答案，我們就以下幾個簡單的消費性產品，來介紹 Bluetooth 所能提供的服務：

- 三合一手機：如果手機內有 Bluetooth 裝置，那麼它不僅能隨時提供行動電話的功能，回到家後只要與桌上電話機座（也有 Bluetooth 裝置）連線，還可成為一般有線電話的無線話機；另外，它也能當成無線對講機，直接和另一個較近距離的手機彼此通訊。

因此，透過 Bluetooth 技術，便可將行動電話、有線電話、以及無線對講機結合成一體，幫您節省許多電話費用。

- **無線耳機**：如果隨身聽、手機、汽車音響、以及耳機都有 Bluetooth 裝置，該耳機就可成為無線耳機，來選擇聽取不同裝置的音響，不但可以節省許多不必要的連線，在開車時更成為最理想的免持聽筒。
- **互動式會議**：如果每部手提電腦上都有安裝 Bluetooth 裝置的話，便能構成一個最簡單的『對等網路』( Ad Hoc Network )，開會時可用來做互動式的資料傳輸。
- **整合電腦連線**：如果電腦主機、滑鼠、鍵盤、喇叭等等都有裝設 Bluetooth 技術，便可省去許多連接線，所有訊號皆利用無線 Bluetooth 通訊來完成。
- **數位化家電**：假設住家中的電冰箱、冷氣機、電視、以及防盜系統等設備都有 Bluetooth 裝置，出門在外時便可透過手機來觀察及控制家中的設備。

以上只是簡單介紹幾個與生活應用有關的範例，我們不難發現，雖然這些通訊都著重於短距離之間的連線，但確可提升許多便利性。然而 Bluetooth 的野心並不侷限於這些地方，它還可以連結一般網路，成為網路系統的前端使用設備。我們可將 Bluetooth 技術所能提供的功能區分為下列三大項：

- (1) **資料與聲音的存取點( Data and Voice Access Point )**：Bluetooth 裝置可即時( Real-time )收集聲音與資料，再以無線傳遞方式傳送給其它無線裝置或有線設備。也可反向成為其它固定裝置的資料與聲音輸出端點，成為一個理想的資料與聲音存取點。
- (2) **取代連線 ( Cable Replacement )**：透過無線 Bluetooth 裝置，可省略許多不必要的連線。
- (3) **對等網路 ( Ad Hoc Network )**：各種裝置 ( 電腦、冰箱、等等 ) 之間都可透過無線 Bluetooth 來互相通訊，成為對等網路。

## 16-2 Bluetooth 網路特性

由上一節的介紹，大略可以瞭解 Bluetooth 的應用範圍，也可以發現 Bluetooth 技術是整合所有電子裝置的最佳利器；但由於它的應用範圍廣闊，各種應用環境對 Bluetooth 技術都有不同的解說，在此我們以網路的觀點來敘述 Bluetooth 的特性如下：

- **傳輸速率**：1 Mbps ( 同一個 Piconet 網路內所有裝置共享 )。
- **網路範圍**：大約 10 公尺左右，室外加裝功率放大器可達 100 公尺。
- **網路環境**：一個 Bluetooth Radio 最多可包含十個 Piconet 網路，一個 Piconet 最多可包含 8 個活動的 Bluetooth 裝置 ( 有安裝 Bluetooth 技術的設備稱之 )。
- **通訊頻段**：2.4 GHz ISM 免授權頻段。
- **跳頻技術**：採用 FHSS 展頻技術，將 2.4 GHz 頻段劃分為 79 頻道 ( 其中 32 個頻道另做控制使用 )，每秒跳越 1600 次。
- **媒介存取**：採用主從式的集中處理，在一個 Piconet 網路中，由 Bluetooth 裝置中自由競爭選擇一個 Master 設備，由 Master 來分配其它 Slave 的媒介使用權，且 Master 和 Slave 之間的角色可隨時互換。
- **傳輸訊息**：可同時傳遞語音與資料的訊息。
- **編碼技巧**：語音訊號採用 CVSD ( Continuous Variable Slope Delta )，傳輸速率為 64 Kbps。數據資料採用 GFSK ( Gaussian Frequency Shift Keying ) 調變技術。
- **多工技術**：採用『分時雙工』( **Time Division Duplex** ) 機制。
- **連線方式**：提供兩種連接方式，一者為『同步連結導向』( **Synchronous Connection-Oriented, SCO** ) 鏈路，是針對語音通訊使用；另一則是針對資料傳輸的『非同步非連接』( **Asynchronous Connectionless Link, ACL** ) 鏈路。

### 16-3 Bluetooth 協定標準

既然 Bluetooth 包容了所有電子裝置的通訊系統，可想而知它的規格文件必然是非常豐富，大約有 1500 頁，但可將它區分為兩大類：『**核心規格**』( **Core Specification** ) 和『**草案規格**』( **Profile Specification** )。Core Specification 主要描述 Bluetooth 的協定堆疊架構，以及

各個層次應具有的功能，範圍可由無線電波到網路應用軟體之間的連結規範。Profile

Specification 是描述建立在 Bluetooth 技術上的各種應用領域，它是期望各種電子裝置都能夠有一個標準規範來實作 Bluetooth，使各設備之間的通訊較容易達成。簡單的說，Profile Specification 是架設在 Core Specification 上的各種應用標準，並且希望每一特殊應用都能有一標準規範來做依據。

### 16-3-1 協定堆疊

圖 16-1 是由 Core Specification 所制定的協定堆疊，其中包含 Core、Cable Replacement、Telephony Control、Adoption Protocol 等四種協定。Core Specification 包含有下列五個層次：

- (1) **無線電協定 (Radio Protocol)**: 描述有關無線電介面規格，包含電波頻段、跳頻技術、調變技術、以及發射功率等。
- (2) **基頻協定 (Baseband Protocol)**: 規範 Piconet 網路內的連線建立方式，其中包含定址方式、封包格式、時序、以及電源控制。
- (3) **鏈路管理協定 (Link Manager Protocol, LMP)**: 負責 Bluetooth 和其它設備之間連結鏈路的建立與管理，其中包含安全性規範，如認證 (Authentication) 和加密 (Encryption)，也包含控制與協調基頻封包的大小。
- (4) **邏輯鏈路控制與調適協定 (Logical Link Control and Adaptation Protocol, L2CAP)**: 調適上層的通訊協定和基頻協定之間的連結，L2CAP 提供連接導向和非連接等兩種連線方式。
- (5) **服務發現協定 (Service Discovery Protocol, SDP)**: 提供服務的裝置訊息、服務項目、以及特性的訊息供查詢，並提供兩個或兩個以上的 Bluetooth 裝置之間彼此建立連線。

至於 Cable Replacement Protocol 中的 RFCOMM (Radio Frequency Communication) 是提供一個『**虛擬串列埠口**』(Virtual Serial Port) 來取代電腦設備中的低速率傳輸埠口 (如 RS-232C)，以達到省略連線的功能。Telephony Control Protocol 是由 TCS BIN (Telephone Control Specification - Binary) 來製作，TCS BIN 可模擬電話系統的撥號、提起話機、忙線狀

態的處理，並且也提供行動電話的管理程序。Adopted Protocol 是依各種需求而加入的協定，譬如 PPP、IP、UDP、WAP 等等。

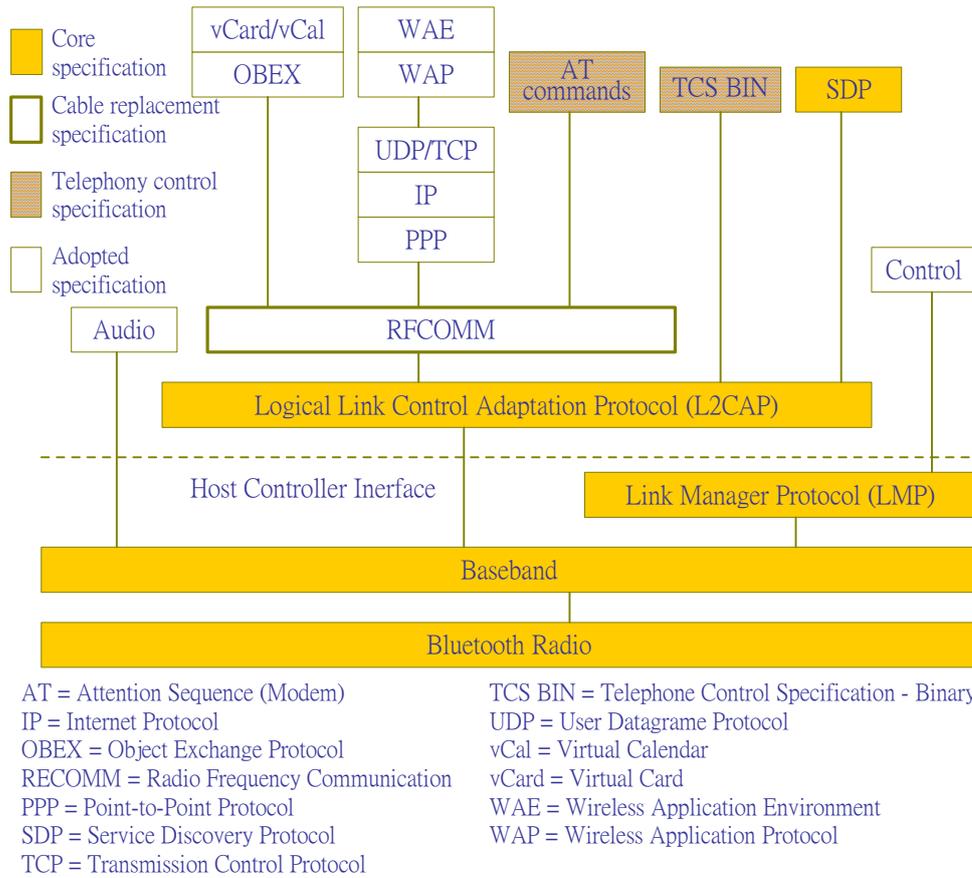


圖 16-1 Bluetooth 協定堆疊

圖 16-1 中的主機控制部份包含了 LMP、Baseband 和 Bluetooth Radio 三個功能層次，這三個功能層次在所有 Bluetooth 裝置上都必須具備，其它層次則依照各種裝置的環境需求來實現。至於在各種應用環境中，應該選用哪些層次，這也是讓人困擾的問題，我們用兩個範例來說明 Bluetooth 協定堆疊的選用。圖 16-2 (a) 為一般 Bluetooth 技術應用在網路連接的情形，主要是 RFCOMM 利用 PPP ( Point-to-Point Protocol ) 協定來連結高層的通訊協定 ( 如 IP )。而圖 16-2 (b) 則為三合一手機的協定堆疊，透過 TCS BIN 模擬電話撥接功能，而上層的通訊協可以是無線電話系統或基地台系統。

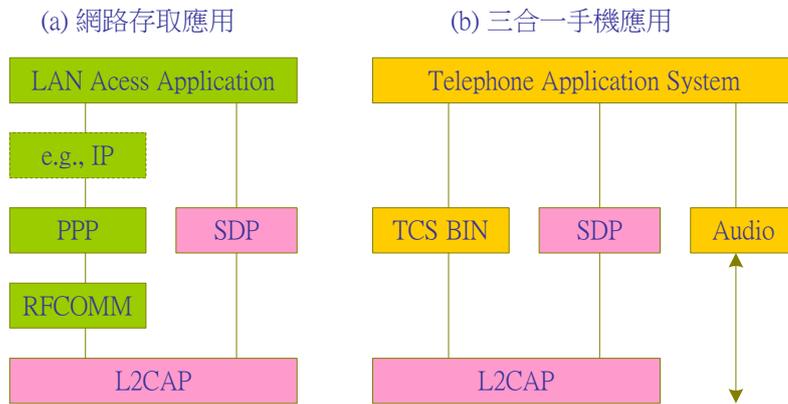


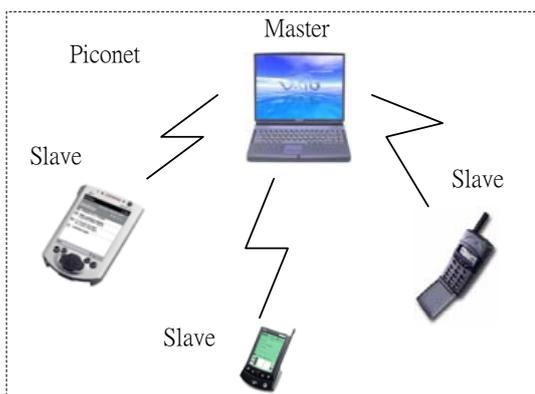
圖 16-2 Bluetooth 協定堆疊範例

### 16-3-2 網路架構

Bluetooth 的網路型態可由若干個 ( 最多 10 個 ) Piconet 網路構成一個較大的 Scatternet 網路，分別敘述如下：

#### (A) Piconet 網路

若干個 Bluetooth 裝置互相連結，即可成為一個 Piconet 網路，這是 Bluetooth 最基本的網路架構。在一個 Piconet 網路中，可經由競爭選擇來產生一個 Master，由它來管理該 Piconet 網路。Master 分配網路上所有 Slave 的媒介使用權，一個 Master 同時最多可與 7 個處於 Active 狀態的 Slave 裝置通訊 ( 如圖 16-3 所示 )。一般 Slave 裝置只和 Piconet 的時序同步但不傳輸資料，此種情況下稱之為 Park 狀態，一個 Piconet 網路可同時存在 256 個處於 Park 狀態的 Slave。雖然所有 Bluetooth 網路都在 2.4 GHz ISM 頻段上通訊，但可利用跳頻展頻的跳越順序來決定各個 Piconet 網路的通訊領域，每一 Piconet 網路上的跳越順序是由 Master 來決定，亦是利用 Master 裝置的位址來制定該網路的『**虛擬亂數序列**』 ( **Pseudo Noise Sequence, PNS** )，如此便可達到各個 Piconet 都有獨立的通訊領域。



### 圖 16-3 Piconet 網路

Bluetooth 網路並不使用分散式的媒介存取技術，而是採用集中控制法，由 Master 來分配頻道使用權給各個 Slave。此外，1 Mbps 的頻寬是由 Piconet 網路內所有裝置共享，所以 Slave 數目愈多，各個裝置所能分配到的頻寬就愈少。基本上，在一個 Piconet 網路內的 Slave 裝置之間並不互相通訊，每個 Slave 都只能和 Master 通訊，也只有 Master 才能控制整個 Piconet 網路內的傳輸。

依照 Bluetooth 的定義，所有裝置的地位是平等的，任何 Bluetooth 裝置都可以成為 Master 或 Slave。原則是首先提出連線要求者便成為 Master，被動和 Master 要求連線者便成為 Slave；也就是說，第一個發起 Piconet 網路連線者便是 Master。Master 也可以隨時退出，換其它 Slave 來承當 Master 工作。從另一方面來看，Bluetooth 利用 Master 來負責管理整個 Piconet 網路的工作，並沒有基地台的概念。

### (B) Scatternet 網路

在 Piconet 網路中的所有裝置共用 1 Mbps 的傳輸速率，所以當有更多的 Slave 加入時，每個 Slave 所能分配到的頻寬便將隨之下降。Bluetooth 的解決方法是減少單一 Piconet 的裝置數量，而讓較多個 Piconet 網路能互相通訊；此法事實上就是結合多個 Piconet 網路，構成一個較大的 Scatternet 網路，如圖 16-4 所示。Scatternet 的建構方法是，某一裝置雖然是某一個 Piconet 網路的 Master 成員，但它也可以成為另一個 Piconet 網路的 Slave 成員。基本上，每一個 Piconet 網路都有自己的跳頻順序，因此，Piconet 網路之間的頻道使用並不衝突。在一個 Scatternet 網路上，Master 裝置可以成為各個 Piconet 網路之間的溝通橋樑。

如圖 16-4 中筆記型電腦在自家網路是扮演 Master，但在另一個網路則是 Slave 角色。手機電話欲傳送資料給另一個 Piconet 網路的固定電話，首先手機電話將資料傳送給筆記型電腦 (Master)，筆記型電腦 (Slave) 再將資料傳送給另一網路的 PDA (Master)，最後 PDA 才將資料傳送給固定電話 (Slave)。

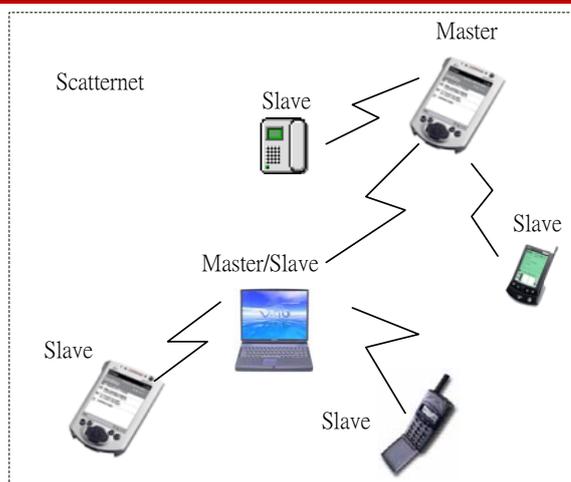


圖 16-4 Scatternet 網路

雖然每一個 Piconet 網路都有自己的跳頻順序，但當 Scatternet 網路內有太多 Piconet 時，也會產生碰撞的現象，使傳輸效率降低。但依照統計分析，當一個 Scatternet 網路內有 10 個 Piconet 時，每個 Piconet 所減少的資料傳輸量 ( Throughput ) 大約 10 %，似乎不會造成太大的影響。

## 16-4 無線電協定

在 Bluetooth 『無線電協定』( **Radio Protocol** ) 部份較為簡單，主要規範發射功率和使用頻段。Bluetooth Radio 的傳輸頻道是使用 2.4 GHz ISM 頻段，並採用跳頻展頻 ( FHSS ) 傳輸模式 ( 有關跳頻展頻技術請參考第十五章 )。然而不同國家所規劃的頻道也不盡相同。全世界大部份國家，包括北美和大部份歐洲地區，Bluetooth 都使用 ISM 中的 2.4 GHz ~ 2.4835 GHz 頻段，並佔滿整個 83.5 MHz 的頻寬；以每頻道 1 MHz 的頻寬，劃分為 79 個頻道；各頻道載波頻率可以表示為  $2402 + k$  MHz，其中  $k = 0, 1, 2, \dots, 78$ 。日本原來只使用 23 個頻道，但從 1999 年 10 月起開始也調整為 2.400 GHz ~ 2.4835 GHz ( 83.5 MHz 頻寬 )，改成和歐美相同的 79 個頻道。

Bluetooth 標準將 Bluetooth 裝置所能發射的功率區分為三個等級，分別為 100 mW ( 20 dBm )、2.5 mW ( 4 dBm ) 與 1mW ( 0 dBm )。一般功率 1 mW 時，傳輸距離是 10 cm 到 10 m，當發射來源接近或遠離接收站時，裝置本身會偵測訊號的增強或減弱，而來調整發射功率。當發射功率提高到 100 mW 時，傳輸距離可以擴大到 100 公尺。

Bluetooth 在訊號調變方面是採用 GFSK ( Gaussian FSK ) 方式。GFSK 調變方式類似一般 FSK 調變，但 GFSK 是利用一個中心頻率  $F$  來表示沒有資料傳送，而正向頻率偏移代表二進位的 1；負向頻率偏移代表二進位的 0，最小偏移量為 115 KHz ( $f_1$ )。在 GFSK 調變中，二進位 1 的頻率為  $F + f_1$ ；0 為  $F - f_1$  ( 有關 FSK 調變技術請參考本書 2-5 節說明 )。

另外，無線電規格和基頻規格有許多連帶性的關係，其中包含許多重要的參數，我們歸類在表 16-1 中，提供給讀者有較概括性的認識。

**表 16-1 無線電與基頻規格**

Topology	Up to 7 simultaneous links in a logical star
Modulation	GFSK
Peak data rate	1 Mbps
RF bandwidth	220 kHz (-3 db), 1 MHz (-20 db)
RF band	2.4 GHz ISM band
RF carrier	23/79
Carrier spacing	1 MHz
Transmit power	0.1 W
Piconet access	FH-TDD-TDMA
Frequency hop rate	1600 hops/s
Scatternet access	FH-CDMA

## 16-5 基頻協定

『基頻協定』( **Baseband protocol** ) 是整個 Bluetooth 技術的核心，它的規範顯然複雜了許多。基頻協定是位於無線電協定與 L2CAP 或 LMP 協定之間 ( 如圖 16-1 所示 )，負責整個 Bluetooth 網路的運作，但也大多著重於 Piconet 網路內的通訊處理。基頻協定在規格書裡佔了很大的份量，也包含了許多技術規範，以下我們以重點方式來介紹相關規格，以及其運作方式。

### 16-5-1 裝置位址型態

在敘述基頻規格之前，我們先認識一下如何表示某一個 Bluetooth 裝置的位址，有下列四種型態。

## (A) BD\_ADDR 位址

每一個 Bluetooth 裝置都有一個『藍芽裝置位址』(Bluetooth Device Address, BD\_ADDR)。這是工廠生產 Bluetooth 裝置時，便將它燒錄在裝置裡面，如同目前的 Ethernet 網路卡一樣，廠商在生產時便將 Ethernet Address 預先設定好了，以後也無法更改。但 BD\_ADDR 與 Ethernet 位址有很大的不同點，Ethernet 位址只是一個主機（或網路卡）的識別值，沒有其它用途；然而 BD\_ADDR 的內容還有許多用途。全世界所有 Bluetooth 裝置的 BD\_ADDR 都是由 Bluetooth SIG 協會負責分配與管理，製造商需要 BD\_ADDR 時都必須向 Bluetooth SIG 協會申請。

每一裝置的 BD\_ADDR 位址，在 Bluetooth 網路（Piconet 或 Scatternet）上是唯一的識別值，並且可說是 Bluetooth 技術的運算核心。幾乎所有負責 Bluetooth 系統正常運作的參數都是由 BD\_ADDR 計算出來的，譬如：跳頻順序（Hopping Sequence）、通道存取碼（Channel Access Code, CAC）與加密鑰匙（Encryption Key）皆是。

圖 16-5 為 BD\_ADDR 的結構圖，其中包含了 24 個位元的『較低位址部份』(Lower Address Part, LAP)、8 個位元的『較高位址部份』(Upper Address Part, UAP)、以及 16 個位元的『未定義位址部份』(Non-significant Address Part, NAP)。其中 UAP 是 Master 裝置用來計算跳頻順序使用，也就是說，在一個 Piconet 網路內的跳頻順序是由該網路 Master 裝置的 UAP 計算得來。

LAP 的用法就較為特別，每一個 Piconet 網路都有個唯一的 CAC( Channel Access Code ) 識別碼，就是由 Master 裝置的 LAP 計算出來的。LAP 的另一應用是，當 Slave 回應 Master 的翻頁（Paging）呼叫時，所使用的 DAC ( Device Access Code ) 識別碼，也是利用 Slave 裝置的 LAP 計算得的。另外，NAP 並沒有定義使用方法（只當識別使用）。



圖 16-5 BD\_ADDR 位址結構

## (B) AM\_ADDR 位址

在 Piconet 網路下，當 Slave 裝置為『活動狀態』( Active State ) 時，Master 將會分配給每個 Active 裝置一個『活動組員位址』( Active Member Address, AM\_ADDR )，Master 就是經由這個位址來辨識各個裝置。一般 Master 和 Slave 之間的通訊，是由 AM\_ADDR 來識別各個裝置，亦即當裝置 ( Master 或 Slave ) 收到封包時，由 AM\_ADDR 位址來判斷該封包是否傳送給自己。

簡單的說，AM\_ADDR 位址是 Piconet 網路上可以通訊之活動成員的識別值，之間通訊便利用這個位址來識別。AM\_ADDR 位址是由 3 個位元所組成，因此在同一個 Piconet 網路內最多可以存在 8 個活動裝置，而 Master 裝置的 AM\_ADDR 位址固定為 000。

### (C) PM\_ADDR 位址

當 Slave 裝置處於『停放狀態』( Park State ) 時，Master 都會分配一個『停放組員位址』( Parked Member Address, PM\_ADDR ) 給它。PM\_ADDR 是由 8 個位元所組成，因此在一個 Piconet 網路內，最多可容納 256 個處於 Park 狀態的裝置。

### (D) AR\_ADDR 位址

每一個處於 Park 狀態的 Slave，也會由 Master 分配到一個『存取要求位址』( Access Request Address, AR\_ADDR )。在一個 Piconet 網路內，Slave 裝置之間所分配的 AR\_ADDR 可能會相同，它並不是唯一識別值。AR\_ADDR 位址的用法是，當 Slave 欲由 Park 狀態轉換到 Active 狀態時，必須利用這個位址向 Master 要求 Slave-to-Master 時槽 ( 容後介紹 )。

## 16-5-2 實體通道

在 Piconet 網路上，是由一個 Master 和若干個 Slave 裝置所構成，亦即形成一個最基本的 Bluetooth 網路。而 Bluetooth 技術以每一頻寬為 1 MHz，將 2.4 GHz 頻段劃分為 79 個頻道，而傳遞訊息時，便在這 79 個頻道 ( 0, 1, ..., 78 ) 之間跳越；又因其每秒跳越 1600 次，所以每一個跳越的時槽為 625  $\mu$ s。另一方面，Bluetooth 採用『分時雙工』( Time Division Duplex, TDD ) 的多工技術，這種 TDD 技術是針對跳頻之跳越次序來分配 Master 和 Slave 間的媒介使用權，其中偶數頻道是由 Master 裝置使用，而奇數頻道則由 Slave 傳遞訊息，這種技術稱之為『跳頻分時雙工』( Frequency Hopping-Time Division Duplex, FH-TDD )，如

圖 16-6 所示 ( $k$  為偶數)。當任何一方傳送資料時，並不完全將整個時槽佔滿，以減少雙方的干擾。

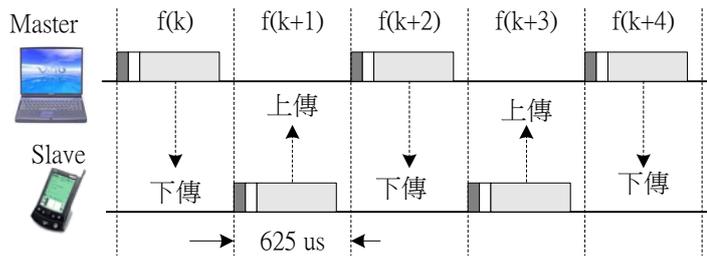


圖 16-6 跳頻分時雙工的傳輸方式

單就 TDD 多工技術來看，傳輸媒介的使用時間是以每個時槽  $625 \mu s$  為單位，將之劃分為若干個時槽，而通訊雙方則以交替方式使用時槽來傳送資料，亦即偶數時槽由 Master 傳送（又稱為 Master-to-Slave 時槽），而奇數時槽則由 Slave 使用（又稱為 Slave-to-Master 時槽）。FH-TDD 將這些時槽以頻率跳越方式，在不同的頻道上傳輸。

然而不論 Master 或 Slave，並非每次都只能使用一個時槽來傳送資料，有時為了提高傳輸速率，也可以連續使用 3 或 5 個時槽來傳送資料。連續傳送時槽可減少每一封包的封包標頭，如果每次都只能用一個時槽傳送，每一封包填入時槽時，都必須加入許多控制訊息。但由於 Master 和 Slave 各被限制只能固定於偶數或奇數時槽傳送，因此，連續時槽僅能為 1 個、3 個或 5 個，否則會跨越到對方（Master 跨到 Slave 或 Slave 跨到 Master）的起始傳送時槽。圖 16-7 為多時槽的傳送方式， $k$  為偶數表示 Master 傳送，若為奇數則表示 Slave 傳送。

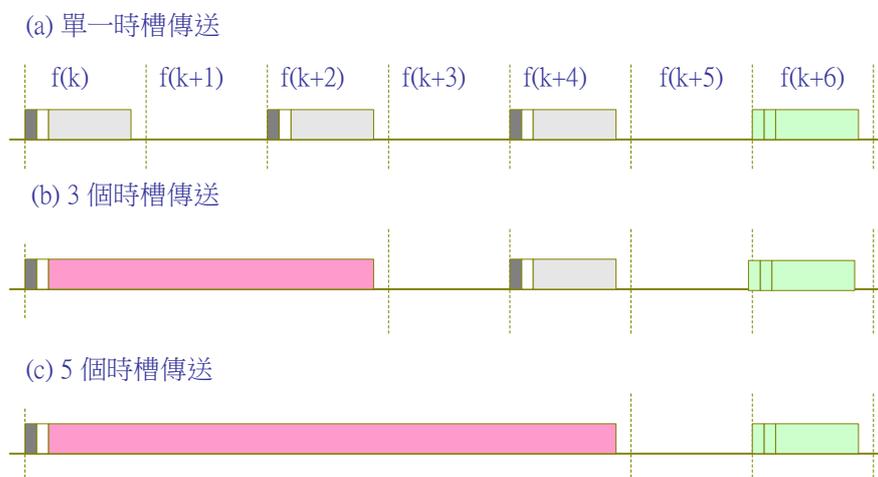


圖 16-7 多時槽傳送方式

### 16-5-3 實體鏈路

就我們所瞭解，Bluetooth 技術具有同時傳送語音與資料的功能，主要是它的『實體鏈路』( Physical Link ) 提供有『電路交換』( Circuit-switch ) 與『封包交換』( Packet-switch ) 等兩種連線型態，以下分別說明這兩種連線型態。

#### (A) 同步連接導向連線

『同步連接導向』( Synchronous Connection-Oriented, SCO ) 連線是屬於電路交換的同步傳輸型態，並且是全雙工連線。當 Master 與某一個 Slave 之間建立 SCO 連線後，不管雙方是否有資料傳送，系統都會預留固定時槽給 Master 和 Slave 使用，而其它 Slave 就不能使用這些時槽來傳輸資料，如圖 16-8 所示。一般 SCO 連線都使用在語音傳輸，每一 SCO 連線都以一個時槽來傳送，並支援 64 Kbps 的傳輸速率 ( HV1 ~ HV3 格式)。當 SCO 連線建立後，Master 和 Slave 之間便可透過該連線來通話，而 Master 無需事先詢問 ( Polling ) Slave。此外 SCO 連線是屬於點對點 ( Point-to-Point ) 的連線，也就是 Master 和 Slave 之間所建立的一對一直接連線。

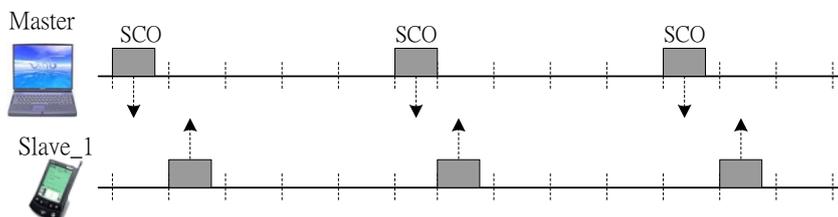


圖 16-8 Master 與 Slave 之間的 SCO 連線傳輸

由於語音傳輸並不適合因受干擾而重新傳送，因此 SCO 連線並沒有自動重新傳送 ( ARQ ) 的機制；但有時候為了提高通話品質，可採用較嚴格的編碼技巧，或者採用較高等的『錯誤修正』( Error Correction ) 技術，因此有 HV1 ~ HV3 的封包格式，可依不同使用環境來選擇。一般來講，如果語音傳輸受到干擾，只要能分辨聲音即可，要求不會太高；但如果在數位音響部份，那就另當別論了。

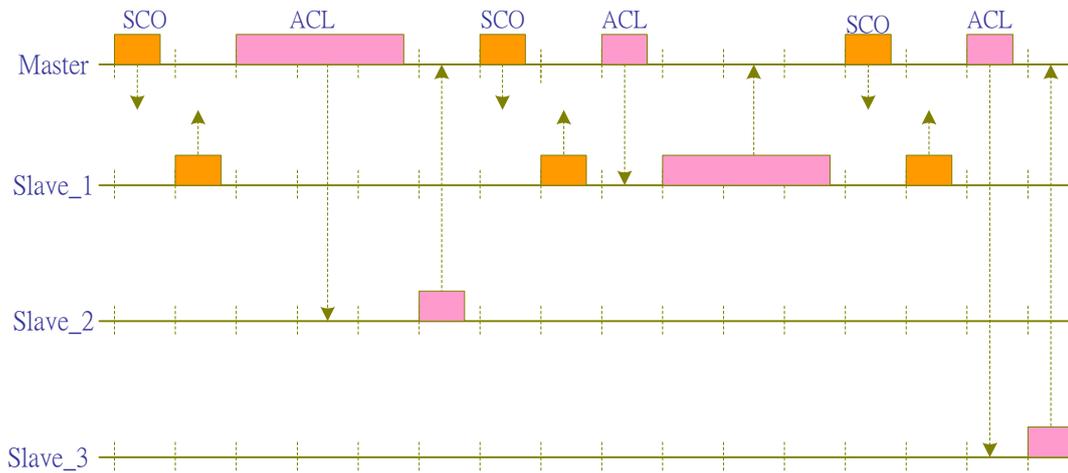
#### (B) 非同步非連接連線

『非同步非連接』( Asynchronous Connection-Less, ACL ) 連線是屬於封包交換的非同步傳輸模式，也是屬於全雙工連線。在一個 Piconet 網路中，是由 Master 來分配 ACL 連線

的使用，任何 Slave 傳送資料之前，必須經由 Master 的詢問 ( Polling ) 並同意之後才可以傳送資料。因為 Piconet 網路內只允許 Master 和 Slave 之間傳輸資料，所以 Bluetooth 提供 Master 和 Slave 之間有點對點 ( Point-to-Point ) 與單點對多點 ( Point-to-Multipoint ) 兩種連線。單點對多點連線表示可以由 Master 對多個 Slave 進行通訊，這是因為 Slave 和 Slave 之間的通訊必須透過 Master 轉送才行。另外 ACL 也提供 Master 裝置對所有 Slave 廣播的功能。

在實體層傳輸方面，SCO 和 ACL 連線都在跳頻的時槽中傳送，SCO 連線都是以單一時槽來傳輸，然而 ACL 可選擇多時槽方式傳輸。因為 SCO 的時槽都是固定的，所以 Master 在分配 ACL 時槽時，必須避開 SCO 的時槽，也就是說，SCO 的時槽優先權較高，而 ACL 只能選擇時槽還有空間的時候才可以佔用。ACL 的多時槽設計是針對爆發性資料的傳輸使用，也就是說，Bluetooth 裝置常處於空閒狀態，當需要傳輸時，假使有大量資料要傳送、且此時又必須快速的傳送資料，才會使用多時槽傳輸。圖 16-9 為一個 Piconet 網路，內有 SCO 和 ACL 同時存在時，封包資料傳遞的情形，其中 SCO 連線已固定在某些時槽上傳送，ACL 僅能使用空間的時槽。圖 16-9 中 ACL 時槽使用情況如下：其一是 Master 傳送一筆佔用三個時槽的資料給 Slave\_2，緊接著下一個時槽由 Slave 回應確認訊號給 Master；另一則是 Master 詢問 Slave\_2 是否有資料要傳送，而 Slave\_2 則發送一個連續三個時槽的資料給 Master；其三是 Master 詢問 Slave\_3，而 Slave\_3 傳送一個時槽的資料給 Master。

這裡要特別注意的是，Bluetooth 網路是利用廣播方式來傳輸，當 Master 欲詢問或傳送訊息給某一個 Slave 時，便將訊號發射出去；每一個 Slave 收到封包後，由封包上的位址 ( AM\_ADDR ) 來判斷是否是要傳送給自己，如果是自己的位址，便將該封包收取；否則便將其拋棄。



**圖 16-9 SCO 與 ACL 連線的時槽傳輸情形**

一般 ACL 連線都使用於資料傳輸，或者是經壓縮處理後的語音或視訊訊息，然而一般網路的上層通訊定（如 IP 或 PPP）都屬於較長封包的傳輸，如果下層利用較短封包的 Bluetooth 來實現，上層封包必須先分割為若干個較短的封包。為了要能合乎各種情況的應用，ACL 支援對稱和非對稱兩種傳輸速率。在非對稱模式下，雖然 Piconet 網路的頻寬為 1 Mbps，但須扣除其它控制訊號的傳輸，因此，下傳最高速率為 721 Kbps ( Master-to-Slave )，而上傳速率更只有 57.6 Kbps ( Slave-to-Master )。在對稱模式下，上傳和下傳速率皆為 432.6 Kbps。然而為了提高傳輸資料的可靠性，ACL 提供有自動重送機制 ( ARQ )，若接收端收到的資料在傳送中受到干擾而發生錯誤，則可要求對方重新傳送。

為了提高傳輸效益，在 Bluetooth 規格中允許 Master 和每一個 Slave 之間可建立多條 SCO 連線，但僅能建立一條 ACL 連線。在建立或關閉 SCO 連線時，必須先建立 ACL 連線來傳遞控制訊號，但建立 ACL 連線比 SCO 連線來得容易。

### 16-5-4 封包格式

Bluetooth 將填入時槽內的訊息單元（未經編碼處理）稱為『封包』( Packet )（一般網路稱為訊框），隨著傳送不同的訊息（語音或資料），或使用時槽的多寡（1、3 或 5 個時槽），會衍生許多不同的封包格式，但它們都有一定的基本格式。圖 16-10 為 Bluetooth 的封包格式，包含了存取碼 ( Access Code )、標頭 ( Header ) 和承載 ( Payload ) 等三個主要欄位；但並不表示所有訊息封包都具有這些欄位，還是要依照該封包所傳送的訊息而定，以下分別說明各欄位的功能。

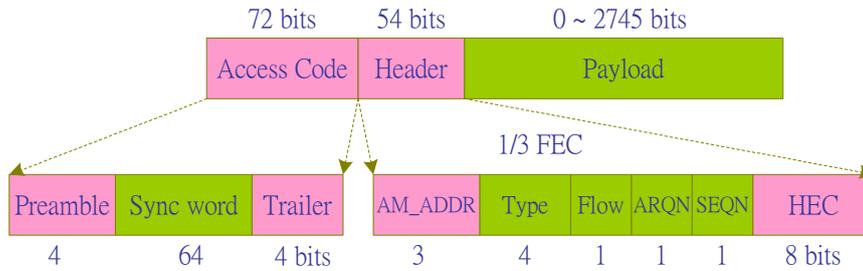


圖 16-10 Bluetooth 封包格式

### (A) 存取碼 ( Access Code )

存取碼欄位包含了 Preamble、Sync Word 與 Trailer 等三個子欄位。其中 Preamble 和 Trailer 子欄位都是 4 個位元，是用來做直流補償 ( DC Offset Compensation ) 使用，如果 Sync Word 的最低位元 ( LSB ) 為 1 時，則 Preamble 為 1010；否則為 0101。Trailer 是由 Sync Word 的最高位元 ( MSB ) 決定，如果為 1，則 Trailer 為 1010；否則為 0101。

然而 Sync Word 欄位有 64 個位元長度，它的內容是由某一個裝置的 LAP 位址經過計算處理後的值，這個計算處理是將 LAP 位址 ( 24 位元 ) 與虛擬亂數序列 ( Pseudo-Noise, PNS ) 經過 XOR 運算，並展開成為 64 位元。使用不同裝置的 LAP 位址來計算，會形成不同的 Access Code 型態，以下我們就由 Access Code 的型態來介紹它是由何種裝置計算得來；

- 『通道存取碼』 ( Channel Access Code, CAC )：Sync Word 是由 Master 裝置之 BD\_ADDR 位址內的 LAP ( Lower Address Part ) 計算而來。CAC 表示一個 Piconet 網路的識別碼，在同一個 Piconet 網路內的 Master 和 Slave 都用這個碼來表示，避免和其它 Piconet 互相干擾，同時防止他人竊聽。
- 『裝置存取碼』 ( Device Access Code, DAC )：當 Master 對 Slave 發出『翻頁』 ( Paging ) 呼叫，或 Slave 回應 Master 的 Paging 呼叫的訊號時，其呼叫訊號的存取碼就是 DAC，DAC 是由 Slave 的 BD\_ADDR 位址內的 LAP 計算所得的。
- 『詢問存取碼』 ( Inquiry Access Code, IAC )：由某一特殊裝置的位址 ( LAP ) 計算得來。IAC 是 Master 查詢 ( inquiry ) 某一特殊裝置時所使用的查詢訊號。

### (B) 標頭 ( Header )

由於標頭欄位是包含很重要的控制訊息，因此將原來的 18 位元經過 1/3 FEC ( Forward Error Correction ) 編碼後，成為 54 位元長度。標頭欄位包含下列訊息 ( 如圖 16-10 所示 )：

- (1) **AM\_ADDR**：長度為 3 位元。AM-ADDR 為『活動成員位址』( Active Member Address )，亦即此封包的目的位址。任何一個 Bluetooth 裝置在 Bluetooth 網路上都有一個唯一的裝置識別碼，它是由 Access Code 欄位內的 CAC、DAC 或 IAC 位址，和 AM\_ADDR 的組合而成。
- (2) **Type**：長度為 4 個位元。Type 是用來描述封包的格式。在 Bluetooth 網路上，封包可分為共同 ( Common )、同步連接導向 ( SCO ) 與非同步非連接 ( ACL ) 等三種格式，如表 16-2 所示。
- (3) **Flow**：長度為 1 個位元。Flow 旗號是用來針對 ACL 連線作流量控制之用，對 SCO 連線則沒有作用。當 Flow = 1 表示緩衝器還有空間，對方可以繼續傳送資料；如果 Flow = 0，表示緩衝器已滿，要求對方暫停傳送。
- (4) **ARQN**：長度為 1 個位元。ARQN ( Automatic Request Not ) 是要求對方重新傳送的旗號，“1”表示接收正常；“0”則表示經錯誤檢查後，發現封包已發生錯誤，要求對方重送。
- (5) **SEQN**：長度為 1 個位元。SEQN ( Sequence Number Not ) 用來表示該封包是否為重送的封包，“1”代表不是重送封包。SEQN 的功能必須和 ARQN 互相配合使用，它的功能就如同 TCP 協定中的 ACK 和 SYN 位元 ( 有關 TCP 協定請參考第十三章說明 )，都是作為傳輸流量控制。
- (6) **HEC**：8 位元的 HEC ( Header Error Check ) 是針對封包標頭做錯誤控制的檢查碼。

上述的 Flow、ARQN 和 SEQN 等三個旗號都是針對 ACL 連線做傳輸流量的控制使用 ( 採用 Stop-and-Wait 流量控制法，請參考第三章說明 )，如果該封包是使用於 SCO 連線，則這三個旗號便沒有任何功能。

**表 16-2 Bluetooth 封包格式**

Type Code	PhysicalLink	Name	Number of Slots	Description
0000	Common	Null	1	沒有承載欄位，主要使用於接收端回應給傳

				送端 ARQN 或 Flow 旗號。無確認功能。
0001	Common	Poll	1	沒有承載欄位，使用於 Master 詢問 (Poll) Slave 時使用。有確認功能。
0010	Common	FHS	1	展現傳送端的裝置位址及時序的特殊控制封包，使用於回應 Master 的 Paging Response、Inquiry Response，以及跳頻時序的同步。經 2/3 FEC 編碼。
0011	Common	DM1	1	提供控制訊息，並且可攜帶使用這資料。採 16-bit CRC 及 2/3 FEC 編碼。
0101	SCO	HV1	1	攜帶 10 Bytes 訊息，典型使用於 64 Kbps 語音傳輸，1/3 FEC 編碼。
0110	SCO	HV2	1	攜帶 20 Bytes 訊息，典型使用於 64 Kbps 語音傳輸，採 2/3 FEC 編碼。
0111	SCO	HV3	1	攜帶 30 Bytes 訊息，典型使用於 64 Kbps 語音傳輸，無 FEC 編碼。
1000	SCO	DV	1	組合 150 bits 資料與 50 bits 語音訊息，資料部份經 2/3 FEC 編碼。
0100	ACL	DH1	1	攜帶 28 Bytes 資料及 16-bit CRC，沒有 FEC 編碼。典型使用於高速資料傳輸。
1001	ACL	AUX 1	1	攜帶 30 Bytes 資料，沒有 CRC 及 FEC 編碼。典型使用於高速資料傳輸。
1010	ACL	DM3	3	攜帶 123 Bytes 資料及 16-bit CRC，採 2/3 FEC 編碼。典型使用於高速資料傳輸。
1011	ACL	DH3	3	攜帶 185 Bytes 資料及 16-bit CRC，沒有 FEC 編碼。典型使用於高速資料傳輸。
1110	ACL	DM5	5	攜帶 226 Bytes 資料及 16-bit CRC，採 2/3 FEC 編碼。典型使用於高速資料傳輸。
1111	ACL	DH5	5	攜帶 341 Bytes 資料，及 16-bit CRC，沒有 FEC 編碼。典型使用於高速資料傳輸。

### (C) 承載 (Payload)

隨著不同的封包型態，承載欄位的長度會有所不同 (0 ~ 2745 位元)，甚至承載欄位的格式也會不一樣，大略可區分為：單一時槽、多時槽和語音封包等三種格式，前兩者是針對資

料封包，後者為承載語音使用。但單一時槽封包也有可能同時承載語音和資料(如 DV 封包)。單一時槽和多時槽都包含：『承載標頭』(Payload Header)、『承載實體』(Payload Body)、以及 CRC 等三個欄位，如圖 16-11 所示。

基本上，承載欄位所存放的內容，大多是由上層的協定資料單元 (Protocol Data Unit, PDU) 所包裝而來的，因此內容會和上層通訊協定較有關聯。如圖 16-11 所示，基頻協定的上層是 L2CAP、LMP 和 Audio 等三個協定，其中 LMP 是關於控制訊息；Audio 是語音通訊；而 L2CAP 是連結其它通訊協定使用，其中可能傳送資料或封包語音、視訊。有了這些概念後，我們再來看承載欄位內的子欄位會較容易，如圖 16-11 中無論單一時槽或多時槽 (3 個或 5 個時槽)，子欄位的功能都相同，如下敘述：

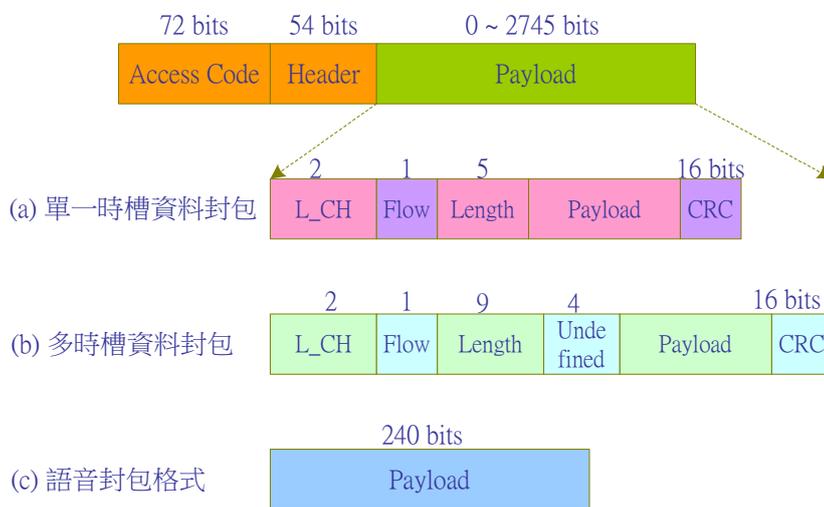


圖 16-11 Bluetooth 封包的承載欄位

- **L\_CH ( Logical Link Channel )**：表示邏輯鏈路識別 (以序號表示)。如果上層協定是 LMP，則 L\_CH = 11；如果是 L2CAP 的起始連線，則 L\_CH = 10；如果是 L2CAP 的連續連線或分段訊息 (Fragment)，則 L\_CH = 01；其它邏輯鏈路為 00。
- **Flow**：此旗號是用於 L2CAP 連線的流量控制。Flow = 1 表示還有緩衝器框間；Flow = 0 表示要求對方暫停傳送。
- **Length**：表示此封包的承載長度，包含承載標頭 (Payload Header)、承載實體 (Payload Body)、以及 CRC 的長度，以位元組為單位。

如果封包格式為語音訊息，則承載內容為 240 位元的語音編碼，如圖 16-11 (c) 所示，典型的應用都是使用於 SCO 連線上。但 Bluetooth 網路允許語音和資料可同時傳送，也就

是承載內可能同時存在語音編碼和資料的包裝，如表 16-2 中的 DV 封包。在 DV 封包的承載內，語音資料固定為 80 個位元，而一般數據資料的長度則為 150 個位元；但 DV 封包內沒有 Payload Header 欄位。

### 16-5-5 數據資料封包

由表 16-2 中可以發現有 6 種數據資料封包格式(如圖 16-11 (a) 與 (b) 所示)，也都應用在 ACL 通訊連線上，我們將這些封包的特性整合於表 16-3，並分別敘述如下：

- **DM1 ( Data-Medium Rate 1 )**: DM1 的 Payload 部份是由 18 個資訊位元組 ( Information Bytes ) 再加上 16 位元的 CRC 錯誤檢查碼所構成，這些訊息都接受過 2/3 FEC 編碼。資訊位元組內包含一個位元組的 Payload Header。DM1 佔用一個時槽。
- **DH1 ( Data-High Rate 1 )**: DH1 的 Payload 欄位包含 28 個資訊位元組及 16 位元的 CRC 檢查碼，兩者都沒有經過 FEC 編碼處理。資訊位元組中包含 1 個位元組的 Payload Header。DH1 佔用一個時槽。
- **DM3 ( Data-Medium Rate 3 )**: DM3 的 Payload 部份是由 123 個資訊位元組再加上 16 位元的 CRC 錯誤檢查碼所構成，這些訊息都接過 2/3 FEC 編碼。資訊位元組內包含 2 個位元組的 Payload Header。DM1 佔用一個時槽。
- **DH3 ( Data-High Rate 3 )**: 與 DH1 非常相似，Payload 欄位是由 185 個資訊位元組與 16 個位元的 CRC 檢查碼所構成，兩者都沒有經過任何 FEC 編碼處理。Payload Header 為 2 個位元組，需佔用 3 個時槽。
- **DM5 ( Data-Medium Rate 5 )**: 幾乎與 DM3 相同，Payload 部份是由 226 個資訊位元組再加上 16 位元的 CRC 錯誤檢查碼所構成，這些訊息都接過 2/3 FEC 編碼。DM5 的 Payload Header 為 2 個位元組，並佔用 5 個時槽。
- **DH5 ( Data-High Rate 5 )**: 與 DH3 相似，Payload 欄位是由 341 個資訊位元組與 16 個位元的 CRC 檢查碼所構成，兩者都沒有經過任何 FEC 編碼處理。Payload Header 為 2 個位元組，共佔用 5 個時槽。

- **AUX1**：與 DH1 相似，其 Payload 欄位包含 30 個資訊位元組，但沒有 CRC 檢查碼及任何 FEC 編碼處理。又 Payload Header 的長度為一個位元組，並佔用一個時槽。

表 16-3 ACL 連線的資料封包形式

Type	Number of Slots	User Payload (Bytes)	FEC	CRC	Symmetric Max. Rate (Kbits)	Asymmetric Max Rate (Kbits)	
						Forward	Reverse
DM1	1	0 ~ 17	2/3	Yes	108.8	108.8	108.8
DH1	1	0 ~ 27	No	Yes	172.8	172.8	172.8
DM3	3	0 ~ 121	2/3	Yes	258.1	387.2	54.4
DH3	3	0 ~ 183	No	Yes	390.4	585.6	86.4
DM5	5	0 ~ 224	2/3	Yes	286.7	477.8	36.3
DH5	5	0 ~ 339	No	Yes	433.9	723.2	57.6
AUX1	1	0 ~ 29	No	No	185.6	185.6	185.6

在不同的環境下，可分別選擇上述各種不同的封包格式來包裝，但每一種封包格式都會佔用一個或二個位元組的 Payload Header。由表 16-3 可觀察到，在非對稱傳輸模式下，下傳最高可達 723.2 Kbps 的傳輸速率，但上傳則僅有 57.6 Kbps。所謂非對稱傳輸模式是『上行』( Up-Link ) 和『下行』( Down-Link ) 係採用不同的封包格式。譬如下行採用 DH5 ( 佔用 5 個時槽 )，而上行使用 DH1 ( 佔用 1 個時槽 ) 的 ACL 通訊連線，表示如果沒有 SCO 連線佔用其它時槽的話，則每 6 個時槽週期中有 5 個下行時槽 ( DH5 ) 及 1 個上行時槽 ( DH1 )，其上/下行傳輸速率計算如下 ( 每秒 1600 個時槽 )：

$$\text{下行速率} = (339 \text{ Bytes}) \times (8 \text{ bits/Byte}) \times (1600 \text{ slots/sec} \div 6 \text{ slots}) = 723.2 \text{ Kbps}$$

$$\text{上行速率} = (27 \text{ Bytes}) \times (8 \text{ bits/Byte}) \times (1600 \text{ slots/sec} \div 6 \text{ slots}) = 57.6 \text{ Kbps}$$

對稱傳輸表示上行與下行都採用同樣的封包格式，例如兩者都採用 DH5 ( 佔用 5 個時槽 ) 封包傳送，在沒有其它 SCO 連線的情況下，每 10 個時槽中就有 5 時槽上行、及 5 個時槽下行，其傳輸速率計算如下：

$$\begin{aligned} \text{對稱傳輸速率} &= (339 \text{ Bytes}) \times (8 \text{ bits/Byte}) \times (1600 \text{ slots/sec} \div 10 \text{ slots}) \\ &= 433.9 \text{ Kbps} \end{aligned}$$

## 16-5-6 語音封包

Bluetooth 封包格式有 5 種語音封包，如表 16-2 所示，這 5 種封包都是利用『**同步連接導向**』(SCO)連線來傳輸。語音封包都是佔用一個時槽，並且以對稱模式來傳輸。基本上，語音傳輸都是屬於交談式的對話，並不適合重新傳送的 ARQ/CRC 機制，Bluetooth 為了提高語音的品質及可靠度，所以將語音封包都先經過 FEC 編碼處理 (如 HV1 和 HV2)。但有一些應用並不需要太好的音效品質，因此也可以採用沒有經過 FEC 編碼的 HV3 格式。但無論採用哪一種格式，都只用到一個時槽，因而傳輸速率都限制在 64 Kbps (雙向傳輸) 以內，但以封包間隔來區分傳輸音效的良寡。各種封包的特性整合於表 16-4，並分別敘述如下：

- (1) **HV1 ( High Quality Voice 1 )**: HV1 的 Payload 所存放的是將 10 個資訊位元組 ( Information Bytes ) 80 個位元 ) 經過 1/3 FEC 編碼後所成的 240 位元資料。HV1 支援 1.25 ms 的語音間隔，傳輸速率為 64 Kbps，封包間隔為 2。
- (2) **HV2 ( High Quality Voice 2 )**: HV2 的 Payload 所存放的是 20 個資訊位元組 ( 160 個位元 ) 經過 2/3 FEC 編碼後成為 240 位元的資料。HV2 支援 2.5 ms 的語音間隔，傳輸速率為 64 Kbps，封包間隔為 4。
- (3) **HV3 ( High Quality Voice 3 )**: HV3 的 Payload 中存放了 30 個資訊位元組 ( 240 個位元 )，但沒有經過經過任何 FEC 編碼處理。HV3 支援 3.75 ms 的語音間隔，傳輸速率為 64 Kbps，封包間隔為 6。
- (4) **DV ( Data Voice )**: DV 的 Payload 分為語音和數據資料兩個部份，語音資料含有 80 個位元，沒有經過任何 FEC 編碼。數據資料包含 0~10 個資訊位元組與 16 個位元的 CRC 錯誤檢查碼，兩者都經過 2/3 FEC 編碼。由 DV 所傳送的資料並沒有 ARQ 機制，僅能以 CRC 來檢查資料的正確性。

表 16-4 SCO 連線的語音封包形式

Type	Payload Header (Bytes)	User Payload (Bytes)	FEC	CRC	Symmetric Max. Rate (Kbits)
HV1	N/A	10	1/3	No	64.0

HV2	N/A	20	2/3	No	64.0
HV3	N/A	30	No	No	64.0
DV	1D	10+(0 ~ 10)D	2/3D	YesD	64.0 + 57.6D

由以上敘述可以發現，如採用較高品質的 HV1 來傳輸語音，通訊雙方都是每兩個時槽傳送一個 HV1 封包，等於將時槽全部佔滿 (SCO 連線)，也就無法再建立其它通訊連線了 (SCO 或 ACL 連線)。相對的，使用傳輸品質較差的 HV3，通訊雙方的時槽間隔為 6，亦是每 6 個時槽才只佔用 2 個時槽，剩下來 4 個時槽可用來建立其它 SCO 或 ACL 連線。至於到底要選用何種封包格式，則可依使用環境來決定。

### 16-5-7 錯誤控制

『**錯誤控制**』( **Error Control** )在一般通訊系統中是非常重要的機制。接收端收到訊息後，如何來判斷該訊息是否正確？假使確定訊息已發生錯誤，應以何種方式來要求對方重送、或者自行將錯誤修正，這就是錯誤控制的功能。一般網路上所傳遞的訊息採用長訊框方式居多，且大多是採用『**錯誤檢出**』( **Error Detection** )方式，如 CRC 檢查機制。當接收端由錯誤檢出機制發現所接收的訊息已發生錯誤，如何要求對方重送，這便是『**自動重複請求**』( **ARQ** )機制。如果傳輸端傳輸訊息之前，將訊息做特殊的編碼 (如 1/3 FEC)，接收端收到訊息後，不但可以檢查出該訊息是否發生錯誤，並且可以將錯誤修正回來，這就是『**錯誤修正**』( **Error Correction** )機制。雖然錯誤修正的功能較強，但必須付出較龐大的代價，經過錯誤修正編碼的資料量，會比原來的資料多了許多，也就是會浪費許多頻寬來傳輸這些多餘的資料，這些多餘的資料又稱之為『**多餘碼**』( **Redundancy Code** )。有關 CRC 和 ARQ 機制請參考第三章說明，這裡不再重複。

Bluetooth 的通訊環境是在 2.4 GHz ISM 頻段上傳輸，目前這個頻段還有許多其它網路在使用，訊號之間干擾非常嚴重，雖然採用跳頻技術可以避免其它訊號的干擾，但也無法達到百分之百的避免。

因此，Bluetooth 制定了嚴謹的錯誤控制機制，除了錯誤檢出的自動重複請求 (ARQ 機制)外，也有錯誤修正的功能。基本上，錯誤修正是針對不適合重送的語音資料，而 ARQ 機制是針對可以重送的數據資料。

Bluetooth 在 Baseband 規範中制定了 1/3 FEC( Forward Error Correction )和 2/3 FEC 兩種『**順向錯誤修正**』機制。1/3 FEC 表示資料經過錯誤修正編碼後，會成為原來資料的三倍，它的傳輸效益便剩下 1/3。譬如，80 位元的資料經過 1/3 FEC 編碼後，便成為 240 個位元長的訊息，其實真正的資料只有訊息的 1/3。2/3 FEC 編碼表示編碼後，原來資料只佔 2/3，譬如資料長度為 10 個位元，編碼後成為 15 個位元。基本上 1/3 FEC 的錯誤修正能力比 2/3 FEC 強，但相對應的，1/3 FEC 有較長的多餘碼，也會影響傳輸效益，如 HV2 的傳輸效益比 HV1 高（有關 FEC 演算法請參考 “Error Control Coding: Fundamentals and Applications” , SHU LIN）。

## 16-6 基頻通道控制

『**基頻通道控制**』（**Baseband Channel Control**）是基頻協定中一個重要的功能，主要是負責建立 SCO 和 ACL 連線的工作，並且管理 Bluetooth 裝置的各種狀態轉換。Bluetooth 網路最主要的功能是『**取代連線**』（**Cable Replacement**），而以無線通訊來省略有線連線；但這些無線裝置大多是使用電池供應電源，因此，如何發揮省電的功能，也是 Bluetooth 網路一個很大的挑戰。另一方面，這些無線裝置也必須隨時待命，它可能隨時會被要求連線，所以即使在各種省電情況下，還是必須隨時和主設備（Master）保持連線狀態。Bluetooth 為了要使各種裝置能具有省電和隨時被呼叫的功能，特別將裝置的運作狀態依照使用情形區分為好幾個層次，越高層次者越耗電力，但容易被呼叫；而較低層次者雖然較省電力，但卻不容易被呼叫。這些裝置狀態的改變也是通道控制主要工作項目之一，我們由下列幾個重點來介紹基頻協定的通道控制；至於其詳細的運作程序，請參考 Bluetooth 的 Core Specification。

### 16-6-1 共同封包

Bluetooth 定義有 ID、NULL、POLL、FHS 與 DMI 等五個『**共同封包**』（**Common Packet**），這五個封包是用來做基頻通道控制使用，如表 16-2 所示。圖 16-10 所呈現的是各種封包的格式，功能介紹如下：

- (1) **ID ( Identify Packet )**：識別裝置本身。當 Master 發送『**查詢**』（**Inquiry**）或『**翻頁呼叫**』（**Paging**）訊號時，所使用的封包就是 ID 封包；或者當 Slave 處理『**翻頁回應**』（**Page Response**）時，也是傳送 ID 封包。也就是說，這是發送者用來表明自己

身份的封包。ID 封包只有 Access Code 部份(請參考圖 16-10)，長度為 68 個位元。由於沒有 Header 欄位，自然沒有 Type 子欄位，因此並未出現在表 16-12 中。

- (2) **NULL( Null Packet )**: Null 封包沒有 Payload 欄位，只有 Access Code 和 Header 兩個欄位。Null 封包的主要功能是作為交換訊息之用，負責將封包內的 ARQN 與 Flow 旗號傳送給呼叫設備，而不需要對方回應 ( Acknowledge )。
- (3) **POLL ( Poll Packet )**: 如同 Null 封包一樣，Poll 封包也沒有 Payload 部份，只有 Access Code 和 Header 兩個欄位。Poll 封包的主要用途是供 Master 輪詢各個 Slave 是否準備傳送資料，但它需要對方回應。
- (4) **DM1 ( Data Medium Rate Packet 1 )**: DM1 封包可能出現於 SCO 或 ACL 連線上，是作為控制訊息的傳遞使用。在 SCO 連線上藉著 DM1 來傳送控制訊息，並且可中斷該連線；而在 ACL 連線內則藉著 DM1 封包來傳送使用者資料，或建立 SCO 連線。
- (5) **FHS ( Frequency Hop Synchronization Packet )**: 『跳頻同步封包』。FHS 封包使用於 Master Response、Inquiry Response、以及 Master Slave 交換等狀態下，供傳遞訊息使用。在 Master Response 與 Master Slave 交換狀態下，FHS 封包會被一直發送，直到收到對方的確認訊息或逾時；然而在 Inquiry Response 狀態下，一方在送出 FHS 封包後並不會等待對方回應確認訊息。

FHS 封包含有 Access Code、Header 與 Payload 三個部份(如同 Bluetooth 封包基本格式，圖 16-10)；而且 Payload 包含著重要的控制訊息，如圖 16-12 所示。FHS 封包的 Payload 部份是由 144 個位元的控制訊息和 16 個位元的 CRC 所組成，這 160 個位元經過 2/3 FEC 編碼後，成為 240 個位元。Payload 欄位上的控制訊息與 Bluetooth 系統的運作方式有很大的關，各控制訊息功能如下：

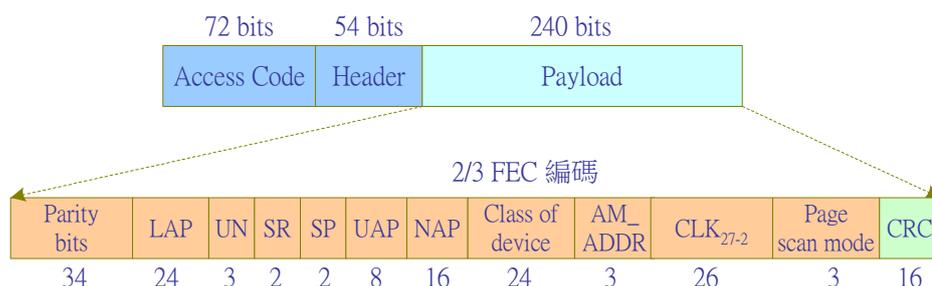


圖 16-12 FHS 封包格式

- **Parity bits**：是由封包前面 Access Code 內的 Sync Word 計算而來，然而 Sync Word 是由 LAP 位址計算出來的。
- **LAP ( Lower Address Part )**：存放傳送該封包裝置的 LAP 位址。
- **UN ( Undefined )**：一般都存放 00。
- **SR ( Scan Repetition )**：表示連續發送 Page Scan 封包的間隔時間，有 R0、R1 與 R2 三種間隔方式。
- **SP ( Scan Period )**：表示在傳送 Inquiry Response 訊息後，執行強制 Page Scan 模式的週期。
- **UAP ( Upper Address Part )**：傳送該 FHS 封包裝置的 UAP 位址。
- **Class of device**：表示傳送此封包的裝置類別。Bluetooth 有將各種裝置編排裝置類別序號。
- **AM\_ADDR ( Active Member Address )**：包含傳送裝置的 AM\_ADDR 位址。
- **CLK27-2**：此欄位存放傳送者的『原始系統時序』( Native System Clock )。
- **Page scan mode**：此欄位是表示呼叫掃描模式 ( Page Scan Mode )，目前制定有 Mandatory Scan Mode、Optional Scan Mode I、Optional Scan Mode II 與 Optional Scan Mode III 等四種。

FHS 封包包含著一個即時時序 ( Real-time Clock ) 訊息 ( CLK27-2 )，每次重送 FHS 封包之前，此時序訊息便被更新。然而 FHS 封包也許會經由各個裝置來傳送，除了 Master 外，每一傳送者都會將時序加入自己的偏移量 ( Offset )，Master 再利用這些偏移量來調整時序，使整個網路的裝置達到同步狀態。

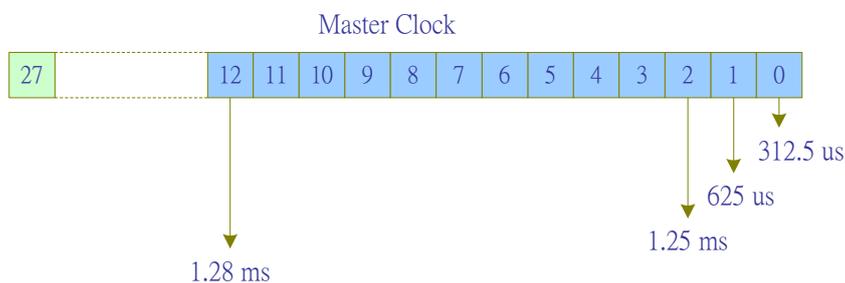
## 16-6-2 時序型態

每一個裝置都有一個內部時序，並且裝置之間的時序必須保持同步，才能順利執行跳頻展頻的功能，以及正常傳送與接收訊息。Bluetooth 利用一個稱之為『藍芽時序』( Bluetooth

**Clock** )，來達到 Piconet 網路內裝置之間的同步。Bluetooth Clock 是一個自由執行的『**原始時序**』( **Native Clock** )，並且從不調整與關閉，裝置之間僅能藉由加入偏移量，來達成暫時性的相互同步。值得注意的是，Bluetooth Clock 和一般時間的時序並無關聯，而是利用一個 28 位元的計數器 ( Counter )。此計數器可以起始於任何值，可能計數的範圍是由 0 到  $2^{28}-1$ ，而每一計數單位的值是 312.5 us ( 亦即 3.2 KHz 的計數頻率 )。

整個 Piconet 網路通道的跳頻和時序，都依照 Master 裝置的 Bluetooth Clock 來達成。當 Piconet 網路建立時，Master 以自己的時序和 Slave 通訊，每一個 Slave 在原始時序上加入偏移量來同步於 Master 的時序；然而此時序是自由執行 ( Free-running )，因此偏移量會隨時被更新。

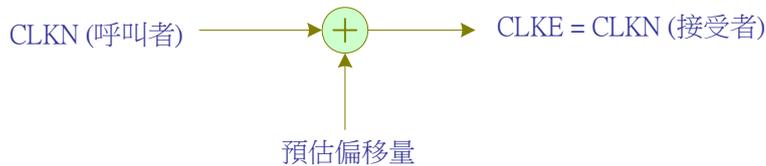
Bluetooth Clock 可產生四種關鍵性的週期來觸發裝置的某些事件，這四個週期對 Bluetooth 系統非常重要，它們是 312.5 us、625 us、1.25 ms 與 1.28 s，分別位於原始時序的 CLK0、CLK1、CLK3 與 CLK12，如圖 16-13 所示。其中 Master-to-Slave 時槽的時間是當 CLK0 與 CLK1 皆為 0 時，也就是偶數時槽。



**圖 16-13 Bluetooth 時序**

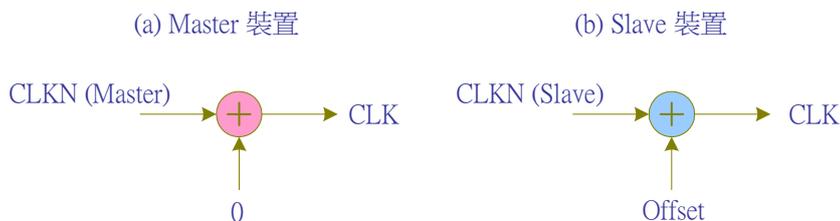
在不同的情況下，會有下列三種型態的時序出現：

- **CLKN ( Native Clock )** : CLKN 是自由執行 ( Free-running ) 的原始時序，既是裝置內時序震盪器的參考，也是其它時序型態的參考值。在 Bluetooth 裝置內，CLKN 的精準範圍至少要保持在一  $\pm 20$  ppm 左右；至於在較低消耗功率狀態下，如 Standby、Hold、Park 與 Sniff，則容許較鬆散的精準度 (  $\pm 250$  ppm )。
- **CLKE ( Estimated Clock )** : CLKE 是由 CLKN 加上偏移量而得的預估時序。接受者 ( Slave ) 利用此預估時序來預估呼叫者 ( Master ) 的原始時序，亦即接收者將偏移量加入 CLKN 後，可得到接近於呼叫者的 CLKE 時序，如圖 16-14 所示。



**圖 16-14 CLKE 的產生**

- **CLK ( Master Clock )**: CLK 是 Piconet 網路內的主要時序，也是依照 CLKN 時序計算得來的。所有 Bluetooth 裝置都是利用 CLK 時序來排序傳送和接收的行程。圖 16-15 (a) 為 Master 裝置產生 CLK 的方法，它所加入的偏移量為 0，也就是說，Master 裝置的 CLK 是和 CLKN 保持同步的。另外，圖 16-15 (b) 為 Slave 裝置產生 CLK 的方法，它是利用各自的 CLKN (如圖 16-14 所產生) 加入某些偏移量來達到本身的同步。



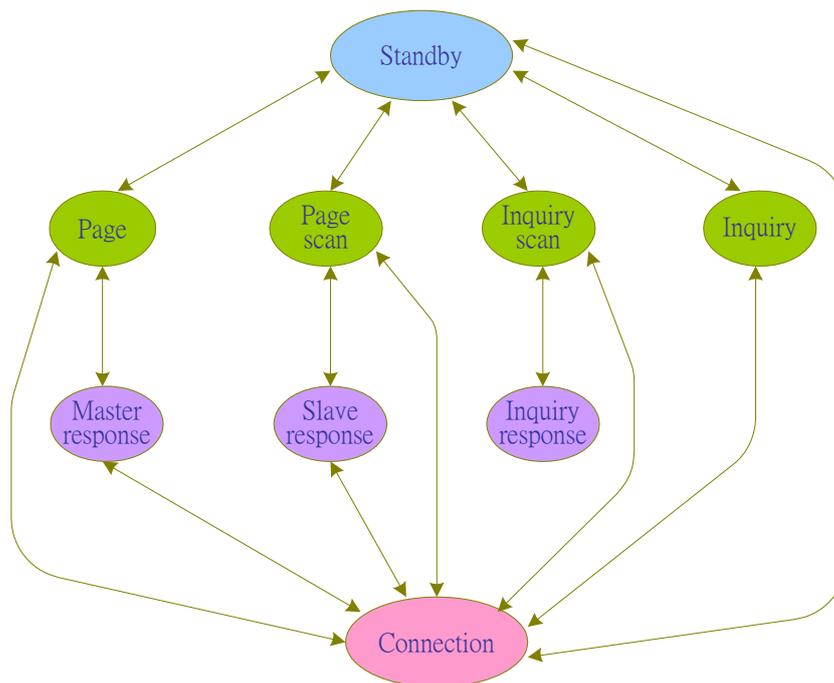
**圖 16-15 CLK 時序的產生**

### 16-6-3 鏈路控制狀態

Bluetooth 裝置在不同環境下，會有各種不同的『鏈路控制狀態』( Link Controller State ) 的演變，譬如裝置剛啟動電源時，如何由『預備狀態』( Standby State ) 進入『活動狀態』( Active State )；又當裝置長時間停頓時，應如何再進入省電模式。反過來說，裝置要如何由省電模式回到活動模式，或者如何將裝置停頓下來並回到預備狀態。在 Bluetooth 規格中，裝置有兩種主要工作狀態：『預備狀態』與『連線狀態』( Connection State )。

在連線狀態時，表示該裝置處於可通訊狀態，且 Master 與 Slave 裝置都使用相同的通道存取碼 ( Channel Access Code, CAC ) 與相同的跳頻序列 ( 兩者皆由 Master 的 BD\_ADDR 演算而得 )。至於預備狀態則是裝置剛啟動電源的情況，此時該裝置並不隸屬於任何 Piconet 的成員，也沒有相對應的 CAC 碼及跳頻序列。

裝置要如何由預備狀態轉換到連線狀態，這可要經過一連串的轉換過程才可以達成。每一個 Piconet 都由 Master 裝置來負責這些連結的工作，由它來呼叫預備狀態的裝置，並使這些裝置成為網路的成員 ( Slave )。也就是說，Master 和 Slave 之間必須經過一連串的『查詢』( Inquiry ) 和『翻頁』( Paging ) 呼叫程序，才能使該裝置進入連線狀態，這一連串的轉換過程會有許多狀態出現，稱之為『中間次狀態』( Intermediate Substate )，如圖 16-16 所示。中間次狀態包含有 Inquiry、Inquiry Scan、Inquiry Response、Page、Page Scan、Master Response、Slave Response 等七種，分別敘述如下：



**圖 16-16 Bluetooth 狀態變遷圖**

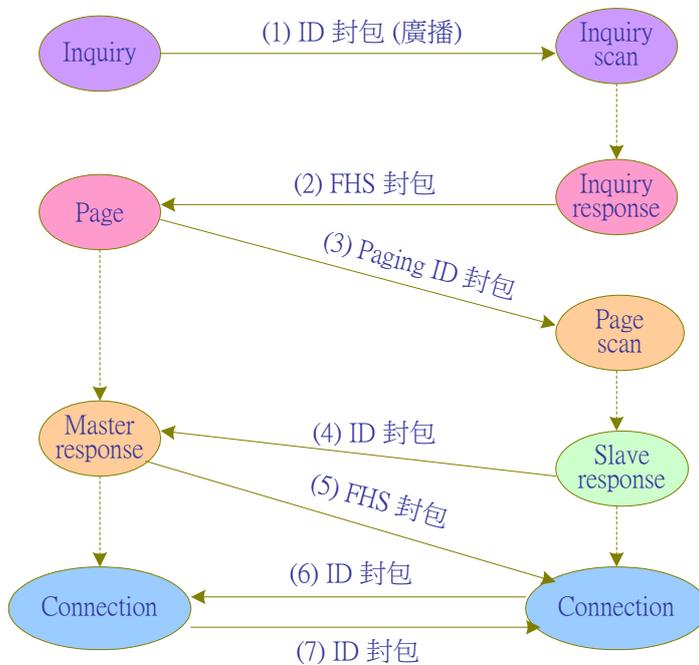
- (1) **Inquiry**：Master 裝置正在查詢是否有新裝置欲加入網路時的狀態。
- (2) **Inquiry scan**：指 Slave 裝置 ( 或新啟動裝置 ) 正在聆聽 Inquiry 呼叫時的狀態。
- (3) **Inquiry Response**：裝置在 Inquiry Scan 狀態下收到 Inquiry 呼叫後，隨即進入 Inquiry Response 狀態。可能會有許多新裝置同時收到 Master 的 Inquiry 呼叫，並同時進入 Inquiry Response 狀態，裝置在此狀態下會決定是否允許進入、或是和其它裝置碰撞的處理。
- (4) **Page**：在進入翻頁 ( Page ) 狀態時，Master 已經有某些要求進入網路的名單、以及其詳細資料 ( 如裝置類別 )。因此 Master 會依照此名單 ( 或裝置的 DAC 位址 ) 來個別呼叫，有如翻頁點名 ( Paging ) 的動作一樣。

(5) **Page Scan**：裝置在 Page Scan 狀態下，會聆聽是否有自己 DAC 位址的 Page 呼叫。

(6) **Master Response**：Master 裝置發送 Paging 呼叫後，若得到對方的回應，便進入 Master Response 狀態，準備處理讓 Slave 進入連結狀態的動作。

(7) **Slave Response**：Slave 裝置收到自己 DAC 位址的 Paging 呼叫後，便可進入 Slave Response 狀態，準備處理進入連結狀態的動作。

基本上，Master 裝置會週期性的廣播 Inquiry 查詢訊號，詢問是否有新的裝置欲加入網路；待新裝置接受 Inquiry 呼叫後，再以 page 程序使之加入網路，而進入連結狀態，成為網路中的一份子。處理程序如圖 16-17 所示，並說明如下：



**圖 16-17 Inquiry 與 Page 處理程序**

**(A) 詢問處理程序 ( Inquiry Procedure )**

在 Bluetooth 裝置還未進入連線狀態之前，並不隸屬於任何 Piconet 網路的成員，也未與任何 Master 之間的時序達到同步，並且跳頻順序也不相同，因此，Master 裝置必須利用特殊的方法，才能執行 Inquiry 與 Paging 呼叫。Master 執行 Inquiry 程序的特殊方法，是利用某些特定頻道來廣播 ID 封包，被呼叫者也在這些頻道上接收訊號。這些特定頻道是 ISM 頻段 79 個頻道中的 32 頻道，到底是哪 32 個頻道，乃依照各國的規定。Bluetooth 將

這 32 個頻道分為兩個 Train，稱為 Train A 與 Train B，每個 Train 包含了 16 不同的頻道。Master 裝置首先在 Train A 頻道上廣播查詢 ( Inquiry ) 訊號，如果沒有任何裝置回應，再到 Train B 頻道上廣播。在每一頻道上廣播時，會採用跳頻技術；但跳頻序列是公開性的，並不侷限於哪一個 Piconet 網路。

Inquiry 呼叫程序一開始是由 Master 裝置進入 Inquiry 狀態，準備詢問是否有新的裝置欲加入 Piconet 網路；同時某些裝置啟動電源，也準備加入網路運作而進入 Inquiry Scan 狀態，聆聽廣播封包 ( ID 封包 )。接下來，Master 裝置在特殊頻道上 ( Train A 或 Train B ) 發送 ID 封包 ( 圖 16-17 訊號 (1) )，該封包含著 IAC ( Inquiry Access Code )；相對地，Slave 裝置也由特殊頻道上收到 ID 封包，知道這是 Master 邀請加入網路的查詢訊號，便進入 Inquiry Response 狀態，並於隨機時間後發送『跳頻同步封包』( FHS 封包，訊號 (2) )給 Master 裝置，FHS 封包內包含著 Slave 裝置本身的 BD\_ADDR ( Bluetooth Device Address )、原始時序 ( Native Clock ) 與裝置類別 ( Class of Device ) 訊息。由於同一時間可能有許多裝置被叫醒 ( Wake-up )，所以在此採用隨機時間，以減少回應 FHS 封包時碰撞的機會，如果某一裝置的 FHS 封包和其它裝置碰撞，則必須回到 Inquiry Scan 狀態，等待下一個 ID 封包。

Master 裝置也許會在收到一個以上的 FHS 封包後，便進入 Page 狀態，執行下一個 Page 程序。同樣地，Slave 裝置在成功發送 FHS 封包後，也會進入 Page Scan 狀態。這裡特別要注意的是，Master 裝置是週期性 ( 每隔 1.28 秒 ) 的進入 Inquiry 狀態，詢問是否有新的裝置欲加入網路，因此所接收到個別裝置的 FHS 封包也會累積起來。

## (B) 翻頁處理程序 ( Page Procedure )

在翻閱呼叫 ( Paging ) 時，同樣是使用到 ISM 頻段上的 32 個特殊頻道。Master 裝置進入 Page 狀態後，可能已收到若干個裝置回應的 FHS 封包，此時必須分別針對每一個裝置做 Paging 的處理。Master 裝置針對每一 Slave 的 BD\_ADDR 來計算『翻頁跳頻序列』( Page Frequency-Hopping Sequence )，並以 ID 封包回應給 Slave 裝置 ( 訊號 (3) )，ID 封包內包含著 Slave DAC ( Device Access Code ) 的 LAP 位址。Slave 裝置在 Page Scan 狀態下收到 Master 的 ID 封包後，便進入 Slave Response 狀態，同時也以翻頁跳頻序列回應相同的 DAC ID 封包給 Master 裝置 ( 訊號 (4) )。接下來，Master 裝置收到 Slave 的 DAC ID 封包後進入 Master Response 狀態，利用下一個 Master-to-Slave 時槽發送一個 FHS 封包給

Slave( 訊號 (5) )，此 FHS 封包包含有 Master 裝置的 BD\_ADDR、以及即時的 Bluetooth 時序的值。緊接著，Slave 裝置回應一個 DAC ID 封包給 Master( 訊號 (6) )，表示有收到 FHS 封包。而 Master 也回應一個 ID 封包給 Slave ( 訊號 (7) )，表示歡迎加入 Piconet 行列。

從此 Slave 擁有 Master 裝置的 BD\_ADDR 與 CLKN，可以計算出 Master 的跳頻順序以及時序的同步，藉此進入連線狀態 ( Connection State )。同時 Master 也分配一個 AM\_ADDR 位址給該 Slave，爾後便利用此位址來通訊，此裝置也正式成為 Piconet 的成員。然而 Master 裝置會連續向欲加入網路的成員做 Page Procedure，一直到全部完成或逾時，才回到連結狀態。

### 16-6-4 連結狀態

每一個 Slave 裝置的『**連結狀態**』( **Connection State** ) 是起始於 Master 裝置的 Poll 封包呼叫，該封包可證實 Slave 是否轉換到 Master 的時序和跳頻序列，如果轉換成功，Slave 可以回覆任何封包。為了節省功率消耗與對其它裝置的干擾，當 Slave 裝置長時間不需要傳遞訊息，但又希望與 Master 裝置保持連線狀態時，Slave 可以選擇進入不同層次的狀態。Slave 在連結狀態下有：Active、Sniff、Hold 與 Park 等四種次狀態，在前面三種狀態下，Slave 還保有 AM\_ADDR 位址；但在 Park 狀態下則已釋放 AM\_ADDR，而以 PM\_ADDR 來取代。另外，由於 Master 裝置只能保持在 Active 狀態下，因此以下僅說明 Slave 的各種次狀態。

#### **(A) Active 狀態 ( Active State )**

Active 狀態是 Slave 與 Master 保持連線的一般狀態，此時 Slave 裝置擁有 AM\_ADDR 位址、以及與 Piconet 網路相同的跳頻順序。在此狀態下，Slave 裝置隨時保持在通訊狀態中，並必須隨時接收及回應 Master 裝置的輪詢 ( Polling )。

#### **(B) Sniff 狀態 ( Sniff State )**

如果 Slave 裝置不時常傳送訊息，便可進入 Sniff 狀態以減少功率消耗。在 Sniff 狀態下，Slave 裝置並不隨時接收 Master 的 Polling 呼叫，而是以一個 Sniff Interval 間隔為週期來聆聽 Master 的呼叫，因此，可能會有一些來自 Master 的呼叫沒接受到。在此狀態下，Slave 還保有 AM\_ADDR 位址、以及 SCO 與 ACL 連線。

### (C) Hold 狀態 ( Hold State )

當 Slave 裝置進入 Hold 狀態時，將暫停支援 ACL 連線，但仍然保持 SCO 連線的通訊，此時 Slave 裝置僅依照 Hold Interval 週期時間來聆聽 Master 的呼叫( Polling )，而 Hold Interval 的數值由上層應用程式制定。進入 Hold 狀態的應用非常普遍，譬如無線耳機和 CD 音響之間的通訊，當兩裝置建立連線後極少再有其它控制訊號的傳輸，而僅以 SCO 連線來傳輸語音訊號，如此無線耳機 ( Slave 裝置 ) 只要進入 Hold 狀態，便能減少無線耳機發射頻率的機會。當它們之間需要傳遞控制訊息時，無線耳機只要週期性的聆聽 CD 音響( Master 裝置 ) 的 Polling 呼叫即可；當然即時性會比 Active 狀態差一些 ( 也許 Master 呼叫好幾次 Slave 才會收到 )。Slave 在 Hold 狀態時還是保持著 AM\_ADDR 位址，跳頻時序也同步於 Piconet 網路。

### (D) Park 狀態 ( Park State )

當 Slave 裝置不需要傳送與接收資料時，若希望更節省消耗功率、又不願離開 Piconet 網路，便可以進入 Park 狀態。Slave 進入 Park 狀態之前，將會釋放掉 AM\_ADDR，並由 Master 分配得到一個 PM\_ADDR ( Park Member Address ) 及 AR\_ADDR ( Access Request Address )。當 Slave 進入 Park 狀態後，即停止所有和 Master 之間的通訊 ( 包括 SCO 及 ACL )，也不聆聽 Master 的 Polling 呼叫。在 Park 狀態下的裝置，仍然與 Piconet 網路保持相同的跳頻順序。同一 Piconet 網路下可以容納 256 個 Park 狀態的 Slave，又 Slave 之間所分配到的 AR\_ADDR 可能會相同。

Master 裝置為了與 Piconet 網路內所有 Park 狀態下的 Slave 通訊，它會週期性的在 Master-to-Slave 廣播頻道 ( Beacon Channel, BC ) 上發出一些訊號；然而 Slave 裝置也會依照 Park Interval 週期性的接收訊息。當 Master 裝置希望喚醒 ( Unpark ) 某裝置時，就在 BC 頻道上發送 Slave 裝置的 PM\_ADDR 位址，並同時指定 Slave 進入 Active 狀態時的 AM\_ADDR 位址。Master 可經由 BC 頻道同時喚醒多個 Slave 裝置；而當 Slave 裝置欲回到 Active 狀態時，也是在 BC 頻道上以 AR\_ADDR 位址向 Master 要求可傳送的 Slave-to-Master 時槽。Master 收到 Slave 的要求後，會再發出控制訊息以喚醒處於 Park 狀態的 Slave。

由上述介紹，可以瞭解 Slave 的四種狀態若依功率消耗大小來排列，依序為 Active、Sniff、Hold 及 Park 狀態；若以週期時間間隔來比較，則為 Park Interval > Hold Interval > Sniff Interval。功率消耗越低時，動作週期越長，也就是以延長 Slave 的反應時間來節省耗電量(週期間隔時間大小是由 Link Manager 所制定)。

## 16-7 鏈路管理協定

『鏈路管理協定』( Link Manager Protocol, LMP ) 是針對基頻層次各項功能的協調控制而定。在每一個 Bluetooth 裝置上都有一個鏈路管理層 ( Link Manager )，裝置之間的鏈路控制事項，就是利用 LMP 協定來互相協調與通訊，如圖 16-18 所示。LMP 層位於基頻 ( Baseband ) 層次之上，接受更高層次 ( 或應用軟體 ) 的命令，向下傳給基頻層。大部份的基頻控制事項都是來自鏈路管理層的控制，而這些控制事項是經由 LMP 協定和通訊對方協調而來的。

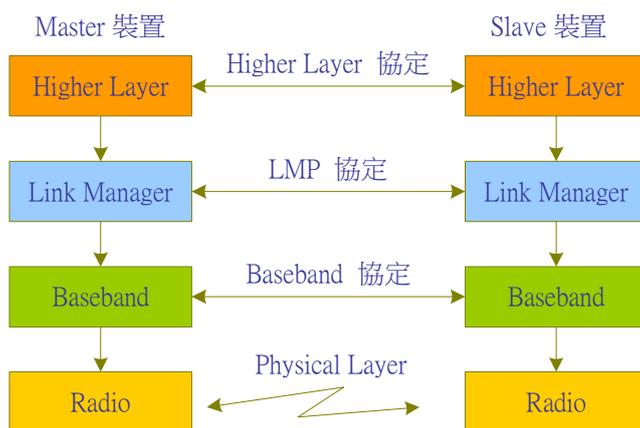


圖 16-18 LMP 協定關係

Master 與 Slave 之間的鏈路管理者是以 LMP PDU 來互相溝通，由 LMP PDU 的描述，大略可將 LMP 的服務區分為 24 個群組，如表 16-5 所示。每一種服務雙方都必須交換一個或多個訊息來達成。表 16-5 中有兩個一般回應 ( General Response ) 訊息是用來回覆其它控制 PDU 所用，其中 accepted PDU 內的工作碼 ( Opcode ) 表示接受控制項目；not\_accepted PDU 封包內的工作碼表示拒絕控制項目。LMP 的控制項目可區分為四大類，以下分別介紹之：

- (1) **安全服務 ( Security Service )**: 提供 Master 與 Slave 裝置之間( 或 Piconet 網路內 ) 的安全性管理機制，譬如認證方式、加密演算法、交換鏈路密匙等等。

- (2) **時序與同步 ( Time/Synchronous )**: 制定 Piconet 網路成員的時序同步機制，譬如時序偏移量要求、時槽偏移量訊息、以及時序校準訊息要求等。
- (3) **站台能力 ( Station Capability )**: LMP 包含兩個 PDU 來交換通訊雙方的站台訊息，如 LMP 版本與該站台所能提供的特色 ( Supported Feature )。
- (4) **模式控制 ( Mode Control )**: 有關 Piconet 網路內的各項控制都包含在此類別內，譬如 Master 與 Slave 之間角色互換的規則，以及 Hold、Sniff、Park 模式的切換運作方式，並包含 SCO 與 ACL 連線建立方式與鏈路服務品質管理等；這些控制項目的運作方式都由 LMP 所提供。

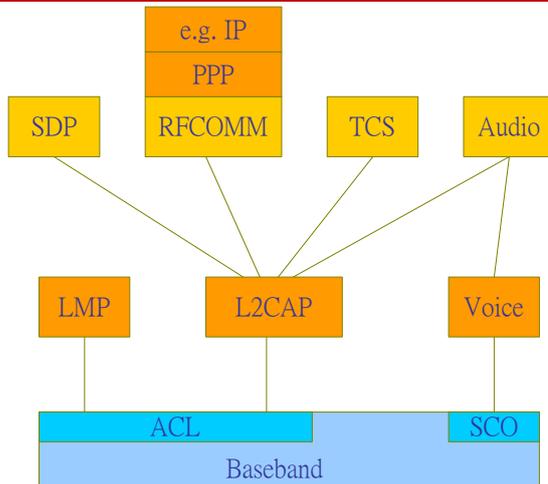
**表 16-5 鏈路管理協定的協定資料單元**

Function	PDU's
General response	Accepted, not_accepted
Security Service	
Authentication	au_rand, sres
Pairing	in_rend, au_rand, sres, comb_key, unit_key
Change link key	comb_key
Change current link key	temp_rand, temp_key, use_semi_permament_key
Encryption	encryption_mode_req, encryption_key_size_req start_encryption_req, stop_encryption_req
Time/Synchronization	
Clock offset request	clkoffset_req, clkoffset_res
Slot offset information	clot_offset
Timing accuracy information request	timing_accuracy_req, timing_accuracy_res

Station Capability	
LMP version	version_req, version_res
Supported features	feature_req, feature_res
Mode Control	
Switch master/slave role	switch_req
Name request	name_req, name_res
Detach	deatch
Hold mode	hold, hold_req
Sniff mode	sniff, sniff_req, unsniff_req
Park mode	park_req, park, set_broadcast_window, modify_beacon, unpark_PM_ADDR_req, unpark_BD_ADDR_req
Power control	incr_power_req, decr_power_req, max_power, min_power
Channel quality-driven change between DM and DH	auto_rate, preferred_rate
Quality of service	quality_of_service, quality_of_service_req
SCO link	SCO_link_req, remove_SCO_link_req
Control of multislot packets	max_slot, max_slot_req
Paging scheme	page_mode_req, page_scan_mode_req
Link supervision	supervision_timeout

## 16-8 邏輯鏈路控制與調適協定

『邏輯鏈路控制與調適協定』( **Logical Link Control and Adaptation Protocol**, 簡稱為 **L2CAP** ) 的功能如同 IEEE 802.2 LLC ( Logical Link Control ) 協定一樣，都是處理上層通訊協定連結到共享網路媒介的鏈路控制，包含有網路媒介的多工處理、協定單元的分段與組合、錯誤檢出與重送、以及通訊連線的建立與終止等等。圖 16-19 為 L2CAP 層與其它高層通訊協定的關係，L2CAP 協定為了能連結各種高層協定，制定了一系列的標準介面，其它通訊軟體只要能合乎標準介面規格，便能順利連結到 L2CAP 層次，再由 L2CAP 層來呼叫以及連結到基頻 ( Baseband ) 層。也就是說，通訊協定只要能符合 L2CAP 的介面標準，便能將它的應用軟體安裝在 Bluetooth 裝置上。



**圖 16-19 L2CAP 層與高層協定的關係**

目前 Bluetooth 規格中已有許多上層協定的標準，相信近來會有更多的通訊協定加入標準之中。我們用圖 16-19 為範例，來介紹一些上層通訊協定的運用，這對我們認識 Bluetooth 網路非常有幫助，敘述如下：

- **RFCOMM ( Radio Frequency Communication )**：『射頻通訊』( RFCOMM ) 協定是 Bluetooth SIG 協會利用現有的 GSM TS07.10 標準制定而成，主要功能是模擬 RS-232 串列埠口內的控制與資料訊號，也就是模擬 RS-232 串列的通訊協定。因此 RFCOMM 上層可以連接 PPP ( Point-to-Point Protocol ) ( 類似 Modem 功能 )，並可透過 PPP 協定連結到其它通訊軟體，譬如 IP 或 IPX 協定。由此可見，L2CAP 只要透過 RECOMM 層的連結功能，便能銜接到 Internet 網路，而目前透過 PPP 協定的應用軟體已經非常普遍。
- **TCS ( Telephone Control Specification )**：『電話控制規格』( TCS ) 包括 TCS Binary 與 AT-Command 兩部份，TCS Binary 是 Bluetooth SIG 協會基於 ITU-T Recommendation Q.931 標準所制定的協定，定義在 Bluetooth 裝置之間建立語音 ( Speech ) 與資料 ( Data ) 通訊所需的呼叫控制命令 ( Call Control Signalling )，使 Bluetooth 裝置能與傳統的電話設備相結合。另外，AT-Command 命令是根據 ITU-T V.250 與 ETSI 300 916 ( GSM 07.07 ) 標準，定義控制行動電話與數據機之間的指令集。因此，透過 TCS 協定便可連結一般有線電話或行動電話。
- **SDP ( Service Discovery Protocol )**：『服務發現協定』( SDP ) 是使用於發現網路中有哪些可用的服務、以及這些服務特性的通訊協定。由於 Bluetooth 裝置( 無線網路 )

攜帶方便，移動中的網路環境必定隨時變化：當裝置進入新的網路範圍時，必須搜尋新的環境是否能給予服務。它的運作方式是當某一裝置 ( Client ) 進入新的網路服務範圍時，即由 SDP 協定尋找該網路 ( 或稱 Server ) 所能提供的服務類型；而當裝置 ( Client ) 離開網路範圍時，SDP 協定也能偵測出該網路 ( 或 Server ) 所提供的服務已經不存在。

- **Voice**：一般語音 ( Voice ) 傳輸都是利用同步傳輸 ( SCO 連線 )，如果針對封包式 ( 經過壓縮處理 ) 的語音，也可透過 L2CAP 來利用非同步傳輸 ( ACL 連線 )。

### 16-8-1 多工與邏輯通道

基本上，L2CAP 僅提供非同步的 ACL 連線，並不提供同步通訊的 SCO 連線。另外，在 ACL 連線中的 AUX1 封包也被禁止使用，因為 AUX1 封包並不提供 CRC 錯誤檢出的功能。傳送資料若無法被檢測出是否有錯誤發生，也就無法執行重送的功能，而 L2CAP 大多連結應用系統，因此不適合使用 AUX1 封包。

圖 16-20 為 L2CAP 多工處理的功能圖，在 L2CAP 層提供多個『邏輯通道』( **Logical Channel** ) 存取點，每一通道給予一個『邏輯通道識別碼』( **logical Channel Identifier, CID** )。上層的通訊軟體只要銜接到一個邏輯通道，便能夠連結到 L2CAP 層的通訊軟體。裝置內系統大多是使用分時多工的方法，來輪流服務已銜接上的邏輯通道；但對於任何通訊軟體而言，卻好像獨享 L2CAP 協定的通訊軟體 ( 如同一般分時多工系統的功能 )。依照邏輯通道的傳輸型態可區分為：

- (1) **資料通道 ( Data Channel )**：只供傳輸資料使用，又可區分為連接導向 ( Connection-oriented ) 與非連接 ( Connectionless ) 資料連線，其中連接導向為雙向傳輸的『全雙工』( **Duplex** ) 模式，而非連接方式為單工模式 ( Simplex )。
- (2) **訊號通道 ( Signaling Channel )**：只供傳遞控制訊號使用，有關建立或終止連線、以及相關控制訊號皆由此通道傳遞。

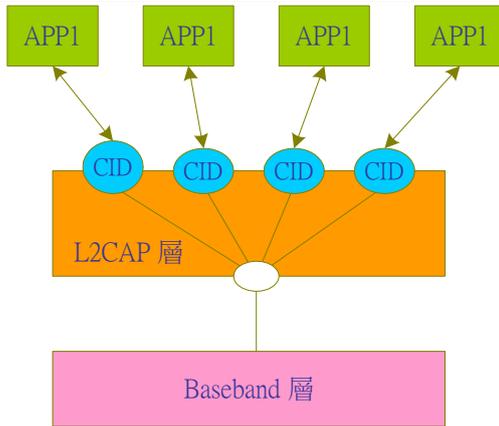


圖 16-20 L2CAP 層的邏輯通道

CID 在 L2CAP 協定中是非常重要的識別碼，每一裝置內的每一個 CID 識別碼分別代表一條與遠端裝置 ( L2CAP 協定 ) 的連接，而遠端裝置也以一個 CID 碼來對應。CID 碼共有 16 個位元，可表示範圍是 0 ~ 65535，通常是在建立邏輯通道時，再動態分配 ( Dynamically Allocate ) 給上層通訊軟體；但 0x0000 未指定使用、0x0001 保留給『訊號通道』( Signaling Channel ) 使用、0x0002 保留給『非連接接收通道』( Connectionless Reception Channel ) 使用、另 0x0003 ~ 0x003F 保留未來發展使用，其餘識別碼則可以任意指定使用 ( 連接導向的識別碼 )。由於每一裝置的非連線接收通道只有一個，並且固定在 0x0002 的 CID 識別碼上，因此每一裝置只能接受一條非連接連線，但可針對多個裝置發送多條非連接連線。CID 識別碼與通道型態之關係如表 16-6 所示。

表 16-6 CID 識別碼分類

通道類別	本地 CID 識別碼	遠端 CID 識別碼
連接導向	動態指定	動態指定
非連接資料	動態指定	0x0002 ( 固定 )
訊號	0x0001 ( 固定 )	0x0001 ( 固定 )

圖 16-21 為 L2CAP 的連結範例，其中裝置 #2 ( Master 裝置 ) 分別與裝置 #1、裝置 #3 建立連接導向連線，其中和裝置 #1 更建立起兩條連線。另一方面，裝置 #2 也分別和裝置 #4 與 #5 各建立一條非連接方式的連線。為了能保持及控制這些資料連線，裝置 #2 分別與各個裝置之間建立一條控制連線，由之間的控制連線來管理這些資料連線的起始或終止。

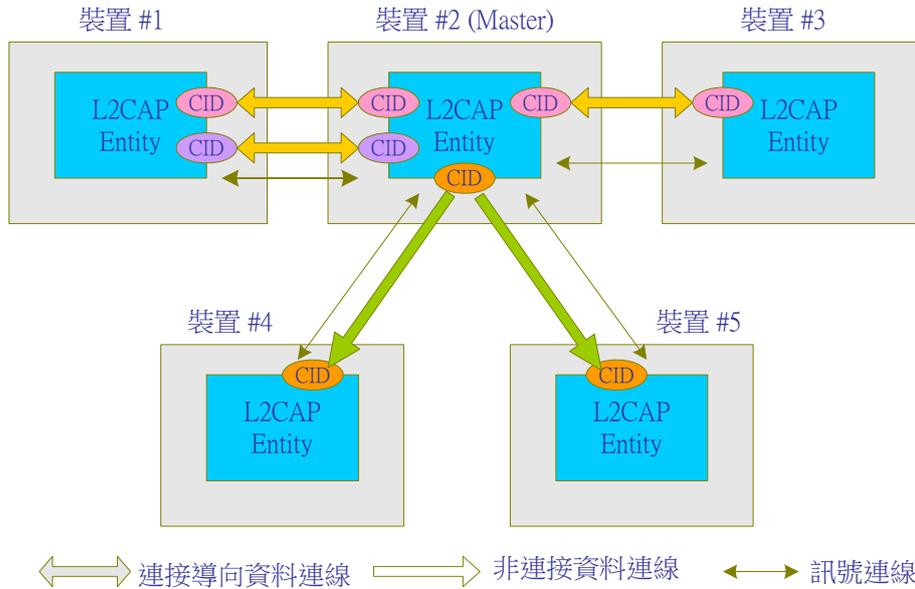


圖 16-21 邏輯通道連接範例

其實 L2CAP 的資料通道與訊號通道對下層通訊協定的連接並不相同，訊號通道是傳輸有關 ACL 連線的建立與終止控制訊息，譬如所建立的資料通道是單一時槽或多時槽、以及希望基頻層所傳送的封包格式等等，因此訊號通道是連結到 LMP ( Link Manager Protocol ) 層次上。L2CAP 接受上層通訊協定的要求，透過訊號通道將所期望的通訊方式傳輸給 LMP 層，再由 LMP 層下達控制命令給基頻 ( Baseband ) 層，由基頻層建立相關傳輸時槽以及封包格式。然而資料連線只負責傳遞資料，其中並不包含相關的控制訊息，因此只要連結到基頻層即可，如圖 16-22 所示。

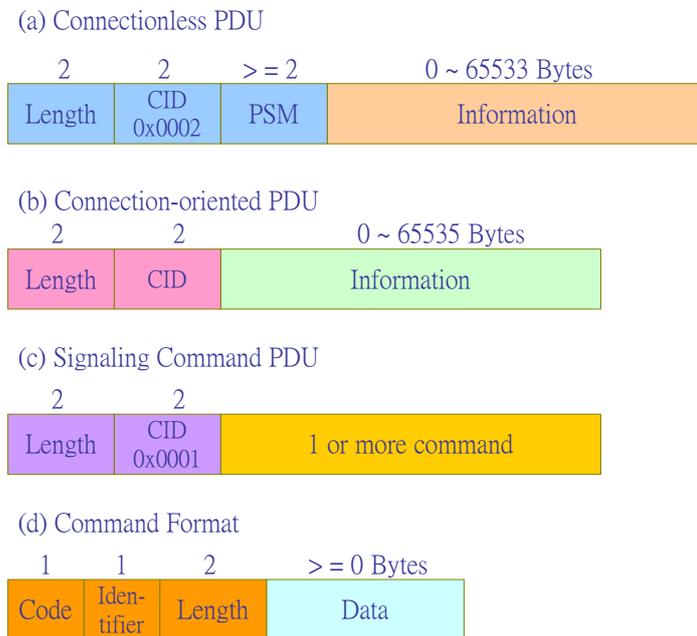


圖 16-22 訊號通道與資料通道

### 16-8-2 L2CAP 封包格式

圖 16-23 為 L2CAP 封包格式，其中包含非連接 ( Connectionless )、連接導向 ( Connection-oriented ) 與訊號命令 ( Signaling Command ) 等三種格式，在訊號命令格式中，

每個封包也許會包含若干個命令，每一個命令格式如圖 16-23 (d) 所示。L2CAP 封包內各欄位功能如下：



**圖 16-23 L2CAP 封包格式**

- **Length**：表示此封包所承載訊息的長度，其中也包含 PSM 欄位（如非連接方式的封包）的長度，單位為位元組。
- **Channel ID ( CID )**：通道識別碼，表示該封包目的地的 CID 識別碼，如非連接封包則固定為 0x0002，而訊號命令封包則固定為 0x0001。
- **PSM ( Protocol Service Multiplex )**：『協定服務多工』。在非連接封包上多出來一個 PSM 欄位，此欄位用來標明本封包所承載的訊息是屬於哪一個通訊協定的。PSM 內容的規範是由 ISO 3309 規格擴充而來，該值必須為奇數 ( ODD )，且最高位元必須為 0；目前 Bluetooth 所規範的通訊協定與 PSM 值的分配如表 16-7 所示。
- **Information**：訊息。此欄位承載上層通訊協定的封包。

由上面的敘述可以發現，各種封包內並沒有來源 CID ( Source CID ) 欄位，表示對連接導向封包而言，在同一個裝置上的上層應用軟體 ( L2CAP 的上一層 ) 只能給予一條連線，否則便無法分辨。譬如，上層通訊協定為 RFCOMM，則通訊雙方僅能針對 RFCOMM 建立一條連線。至於非連接封包，因為它是動態的連線，傳遞完資料後該連線就消失，因此利用 PSM

欄位來分辨即可。另一者，訊號命令封包本來就固定在 0x0001 位址上，也不需要分辨，而且它也是非連接方式，訊號傳遞完成後該連線便自然消失。

**表 16-7 PSM 數值規範**

PSM 數值	描述
0x0001	服務發現協定 ( SDP )
0x0003	射頻通訊協定 ( RFCOMM )
0x0005	電話控制協定 ( TCP )
< 0x1000	保留
0x0001 ~ 0xFFFF	動態指定

訊號命令封包與連接導向封包有相同的封包標頭，但它的 CID 數值固定為 0x0001，且其承載訊息部份也許會包含若干個命令，每一個命令的格式如圖 16-23 (d) 所示，各欄位功能如下：

- (1) **Code**：表示此命令型態，表 16-8 為目前 Bluetooth 所制定的命令彙集。
- (2) **Identifier**：回應命令 ( Response ) 時，用來標示對應到哪一個要求命令 ( Request )。每一個要求命令都會設定一個識別值，而對方回應時便依照該識別值來標示自己是針對哪一個命令所做的回覆。
- (3) **Length**：表示該命令所承載資料 ( 或命令參數 ) 的長度，單位為位元組。
- (4) **Data**：該命令所需的資料或參數。

表 16-8 為 L2CAP 訊號命令的彙集，各種命令會依實際需要承載一些資料或參數，而這些資料參數都有一定的格式。也就是說，Data 欄位會依照各種命令再劃分為若干個子欄位。有關這些子欄位如何定義，請參考『**Bluetooth Core Specification Part D**』，本書限於篇幅不再一一介紹。

**表 16-8 L2CAP 訊號命令彙集**

Code	描述	參數 ( 或資料 )
0x01	Command reject	Reason
0x02	Connection request	PSM, Source CID

0x03	Connection response	Destination CID, Source CID, Result, Status
0x04	Configure request	Destination CID, Flags, Options
0x05	Configure response	Source CID, Flags, Result, Options
0x06	Disconnection request	Destination CID, Source CID
0x07	Disconnection response	Destination CID, Source CID
0x08	Echo request	Data (optional)
0x09	Echo response	Data (optional)
0x0A	Information request	InfoType
0x0B	Information response	InfoType, Result, Data (optional)

### 16-8-3 封包分段與重組

L2CAP 協定建立邏輯通道之前，雙方必須利用訊號通道協議出『最大傳輸量』( **Maximum Transmission Unit, MTU** )、以及所欲建立連線的『服務品質』( **Quality of Service, QoS** )。由於基頻傳輸是利用無線電波，每次所能傳送的封包比較小，因此，一般 L2CAP 協定封包傳送到基頻層時，大多已經過分段處理 ( segment )；接收端收到一連串的分包後，再將其重組 ( Reassembly ) 還原成 L2CAP 封包格式。MTU 就是表示每個傳送封包的大小，也就是說，如果 L2CAP 封包超過雙方協議的 MTU 大小，L2CAP 層傳送封包給基頻層時，該封包便必須經過分段處理。( 如圖 16-24 所示 )

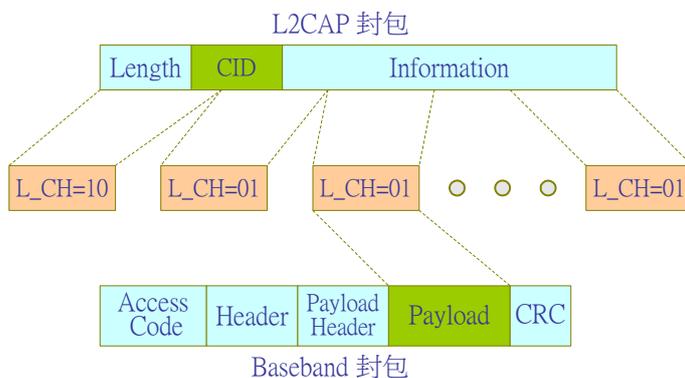


圖 16-24 L2CAP 協定的封包分段

圖 16-24 為 L2CAP 與基頻層之間的封包分段情形，分段後的第一個封包 ( 或未經分段處理的封包 ) 將其 Payload Header 內的 L\_CH 欄位設定為 10，而緊跟其後的分段封包則將自己的 L\_CH 設定為 01 ( 請參考圖 16-11 )。分段後的封包便依照傳輸格式 ( DM1、DH1、DM3、DH3、DM5 或 DM6 )，決定是否將資料經過 1/3 FEC 或 2/3 FEC 編碼，並計算 CRC 檢查值且填入 CRC 欄位上。

接收端收到資料後，依照 CRC 檢查碼診斷該封包是否發生錯誤，如發生錯誤便可以要求對方重送。這裡最重要的一點是，L2CAP 所建立的邏輯鏈路是使用『停止與等待』

( Stop-and-Wait ) 的流量控制法，傳送端每傳送一個封包，便須等待對方回應是否正常接收；而接收端每收到一個封包後，亦須經過 CRC 檢查再決定是否要求對方重送或繼續傳送下一個封包，因此不會涉及緩衝器不足的問題 ( 有關流量控制請參考本書第三章說明 )。

### 16-8-4 層次介面

L2CAP 的層次介面 ( Layer Interface ) 如同 IEEE 802.2 LLC 一樣，具有四個基礎呼叫 ( Primitive )：Request、Indication、Confirm 與 Response。無論 L2CAP 的命令呼叫或資料傳遞呼叫，其運作方式都如同圖 16-25 所示，上層協定可能是 RFCOMM、TCS 或 SDP，而下層協定也許是 LMP 或 Baseband。

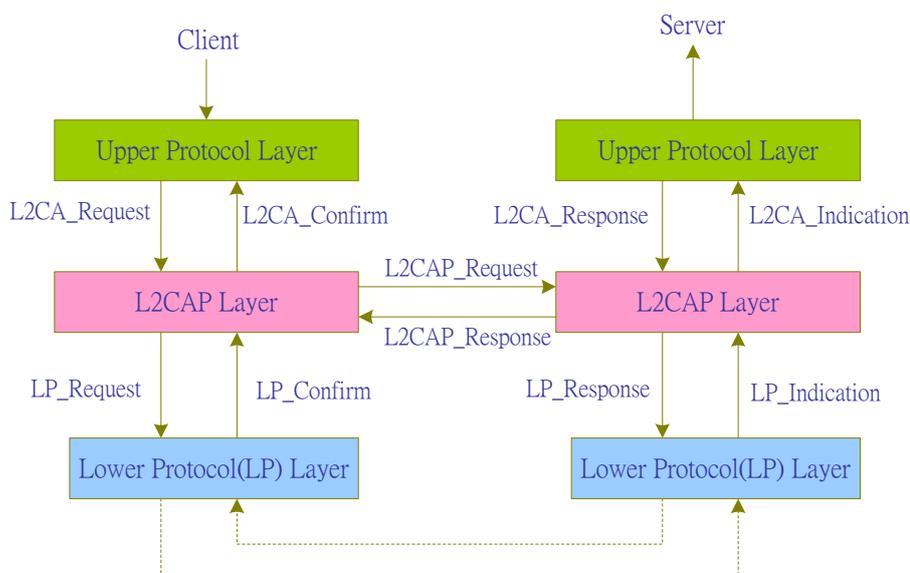


圖 16-25 L2CAP 層次介面的交談

L2CAP 傳遞到上下層的命令是由許多不同的事件 ( Event ) 與命令 ( Command ) 所組成。事件 ( Event ) 包括上層發出的連線要求 ( Connection Request )、資料寫入要求 ( Data Write Request )、或是中斷連線要求 ( Disconnection Request )，下層可能是發出連線要求或中斷連線等事件。每一事件會延伸若干個命令出來，譬如圖 16-25 中，客戶端 ( Client ) 的上層協定發出 L2CA\_Request 命令 ( 要求事件 ) 給 L2CAP 協定，由 L2CAP 協定產生 L2CAP\_Request 命令 ( 要求事件 )，該要求事件是由 LP\_Request 命令來達成。而伺服器端 ( Server ) 下層的 LP\_Response 命令，則用來回應連線要求事件 ( L2CAP\_Response )。

各種資料及訊號通道的建立方式，大多如同圖 16-25 所示，至於各種命令的運作方式，請參考『Bluetooth Core Specification Park D』。

## 習題

1. 請舉五個範例來說明 Bluetooth 技術的應用。
2. 何謂『藍芽技術』( Bluetooth Technique )？它的功能可區分為哪三大項？
3. 請說明 Bluetooth 的傳輸技術？以及使用何種頻段來傳輸？
4. 何謂『Piconet』網路？何謂『Scatternet』網路？
5. 請說明 Bluetooth 網路中 Master 與 Slave 裝置的產生方式。
6. 請說明 Bluetooth 網路的媒介存取技術為何？
7. 請說明 Piconet 網路中，共通的跳頻順序和通道存取碼 ( Channel Access Code ) 是如何產生的？
8. 何謂『跳頻分時雙工』( FH-TDD ) 傳輸技術？
9. 為何基頻的多時槽傳輸只限制於 1 個、3 個 或 5 個時槽？
10. 請說明『同步連接導向』( SCO ) 連線的特性？以及使用時機？
11. 請說明『非同步非連接』( ACL ) 連線的特性？以及使用時機？
12. 在 ACL 連線上，如果下行採用 DM5 封包、上行採用 DH1 封包，請計算出最高傳輸率為何？
13. 如果無線耳機與高音質的數位音響連線，應該採用何種語音封包格式？如果採用最佳封包，是否還可以建立其它連線 ( ACL 或 SCO )？為何？
14. 在 ACL 連線中，請說明自動重送請求 ( ARQ ) 機制的運作方式？
15. 何謂『錯誤修正』( Error Correction )？在 Bluetooth 技術中採用哪兩種錯誤修正方法？兩者之間有何優異點？

16. 何謂『藍芽時序』( Bluetooth Clock )？請說明其功能及特性。
17. 請分別說明 Slave 裝置在 Active、Sniff、Hold 與 Park 狀態的特性，以及如何達到省電的功能？
18. 請說明『鏈路管理協定』( LMP ) 的功能。
19. 請說明『邏輯鏈路控制與適應協定』( L2CAP ) 的功能？
20. 請說明『射頻通訊』( RFCOMM ) 協定的功能？
21. 請說明『電話控制規格』( TCS ) 的功能？
22. 請說明『服務發現協定』( SDP ) 的功能？
23. 請說明 L2CAP 協定的多工技術。
24. L2CAP 有哪兩種邏輯通道？請分別說明其使用時機？
25. 請說明 L2CAP 層的資料傳輸是採用何種流量控制法？為什麼？
26. 請簡略說明 L2CAP 介面程式的運作方式。