國立交通大學

資訊科學與工程研究所

碩士論文

IEEE 802.11 無線網路下

具網路拓樸知覺無縫換手的設計與實作

Design and Implementation of a Topology-Aware Seamless Handover for IEEE 802.11 Wireless Networks

研 究 生：姜宜榮

指導教授：曾建超 教授

中 華 民 國 九 十 五 年 六 月

IEEE 802.11 無線網路下具網路拓樸知覺無縫換手的設計與實作

Design and Implementation of a Topology-Aware Seamless Handover for IEEE 802.11 Wireless Networks

研 究 生：姜宜榮　　Student: Yi-Rong Chiang

指導教授：曾建超　　Advisor: Chien-Chao Tseng

國 立 交 通 大 學

資 訊 科 學 與 工 程 研 究 所

碩 士 論 文

A Thesis

Submitted to Institute of Computer Science and Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

June 2006

Hsinchu, Taiwan, Republic of China

中 華 民 國　九 十 五　年　六　月

IEEE 802.11 無線網路下
具網路拓樸知覺無縫換手的設計與實作

研究生：姜宜榮　　指導教授：曾建超

國立交通大學資訊學院資訊科學與工程研究所

摘　要

本篇論文提出了一套具網路拓樸知覺的「目標基地台搜索與量測」機制，以縮短 IEEE 802.11 無線網路基地台換手過程的延遲。

近年來，以熱點型式出現的 IEEE 802.11 無線網路已經成功地部署在公眾場所，大量服務也隨之應運而生，最受矚目的莫若於 VoIP 等即時性多媒體應用服務。然而，目前普遍實作的 IEEE 802.11 無線網路，Client 端只有在目前通訊受到威脅、即將無法再繼續進行通訊的前一刻，才開始重新找尋目標基地台進行換手。在這樣的換手模式下，Client 端在執行換手程序之前會進行一段較完整的目標基地台搜索，導致整個換手過程時間過長，不符合 VoIP 等即時性應用服務低延遲的需求。

目前普遍實作的 IEEE 802.11 無線網路基地台換手過程，在邏輯上可以分離為基地台搜索與基地台連結 (association) 兩個獨立的 Phases；許多研究也顯示基地台搜索佔據整個換手過程高達 90%左右的時間。因此我們從改良「目標基地台搜索與量測」機制著手，讓 Client 端在使用過程中的適當時機，例如當服務基地台的訊號強度低於某個預先設定的臨界值時，開始預先找出目前服務基地台相鄰網路存取點的局部拓樸；接著使用 Non-AP-Aligned SyncScan 的技巧，在換手情況發生之前持續並且有效地對可能的目標基地台進行量測。我們所設計的換手方式，讓 Client 端在換手過程中不需要再進行基地台搜索，因而能縮短整個換手過程的延遲；此外換手前持續的量測數據，也提供了 Client 端更好的換手決策條件。

我們所設計與實作的具網路拓樸知覺無縫換手機制，可以只針對 Client 端進行實作層面的修改，完全符合 IEEE 802.11 標準，並且不需要額外的資源部署。我們使用 Prism 2 chipset 的無線網路卡，在 Linux 作業系統上，藉由修改 HostAP driver 的方式來實作我們所提出的機制。經由實測評估顯示我們的方法效果不錯，並且可以立即應用於已經佈署使用中的 IEEE 802.11 無線網路。

Design and Implementation of a Topology-Aware Seamless Handover for IEEE 802.11 Wireless Networks

Student：Yi-Rong Chiang　　Advisor：Dr. Chien-Chao Tseng

Institute of Computer Science and Engineering
College of Computer Science
National Chiao Tung University

Abstract

In this thesis, we propose a topology-aware "Target APs Search and Measurement" scheme that can reduce the handover latency of IEEE 802.11-based Wireless LANs. In recent years, 802.11-based wireless LANs, as Hot Spots, has been successfully and widely deployed in public places. Many services arise with the growth of 802.11-based wireless LAN, and VoIP and similar multimedia applications, in particular, have become the well-known services. However, most of the current implementation of 802.11-based networks only attempts a handover when a client's service degrades to a point where connectivity is threatened. With this handover strategy, a client must perform a full search before committing a handover such that the overall handover time is far longer than what can be tolerated by real-time applications such as VoIP services.

AP Discovery and AP Association are intrinsically two independent phases in IEEE 802.11-based WLANs. Furthermore, many studies have shown that AP Discovery constitutes more that 90% of the handover latency in most of the current implementation of 802.11 WLANs. Therefore we focus on improving AP Discovery process to reduce the handover latency of 802.11 WLANs and propose a topology-aware "Target APs Search and Measurement" scheme.

The underlying idea of the topology-aware "Target APs Search and Measurement" scheme is that a client will discover the APs neighbor to the client's current location and use the neighbor AP information to facilitate "Target APs Measurement." Therefore, along with the normal communication, a client determines the local APs topology at some proper time before the handover, for example, when the serving AP's signal strength gets below the pre-defined threshold. Later, the client measures these possible target APs regularly by using the local topology information and the proposed low-cost *Non-AP-Aligned SyncScan* technique. With the facility of the *continuously* tracking of nearby APs, a client could make better handover decision thus improve the handover delay.

Finally, our approach requires only the modification at the client side without any extra deployment at the network, and is completely compatible with IEEE 802.11 standards. We have implemented our design using the Prism 2 chipset WLAN card on Linux platform with the HostAP driver. Experimental results show that our approach is very effective and is suitable for the existing 802.11 wireless networks.

誌 謝

首先，我要感謝我的指導教授曾建超博士在這二年多來所提供給我的完整研究學習成長環境，讓我能夠在此環境中學習到許多相當寶貴的經驗、並且有機會親自深入仔細思考並探索該如何做一項有用的研究。

接著要感謝的是曹孝櫟教授，他提供給我許多有用的最新研究資料題材，並且帶給我許多具有挑戰性與啟發性的觀念，我深信這一篇論文能夠用這種實務性的基調呈現，大多是源自於曹老師所帶給我的影響。

再來我要感謝的是我的二位口試委員紀光輝教授與嚴力行教授，還有所謂第五位口試委員的王讚彬教授。他們在口試期間對我的論文的細心審查，並且提供給我許多相當寶貴且極具建設性的意見，讓我的論文能夠更加完善。

我也要感謝這二年來陪伴我一起成長、砥礪、自省、蛻變，實驗室的學長姊和好伙伴們。你們在這段時間的參與讓我的人生增色不少。

最後，我也必須向許多不列其名的老師、同學、朋友表達我的謝意，他們在我低潮的時候，不管是藉由其自身理念的傳達或是外在身體力行所帶給我的啟發，讓我能夠在黑暗中看到一絲光明。要感謝的人太多了，那就謝天吧！


E
S
A
1896

目 錄


E
S
A
1896

圖 目 錄


E
S
A
1896

表 目 錄


E
S
A
1896

第一章　緒論

1.1 研究動機

近年來，各種寬頻無線存取技術蓬勃發展。從電信網路領域的GSM、GPRS、UMTS系統的演進，到資料網路領域的WiFi、WiMax 等技術發展與標準的制定，未來處處可上網的情境已指日可待。以IEEE 802.11 標準[1] 為基礎的WLAN，由於其架設簡易、價格低廉且提供高速的頻寬，過去已經相當成功地以「熱點的型式」出現在公眾場所中，大量應用服務也隨之應運而生。一般普遍認為Voice over IP (VoIP) 是推動這股無線寬頻熱潮的背後推力，而相關的產品如 dual-module phones 的出現，甚至與電信網路整合的技術如 UMA[1] technology 的發展，都顯示了WLAN是未來 4G無線網路裡頭不可缺少的一環。

然而，目前WLAN無線網路的支援能力，對於VoIP等即時性應用服務 (real-time service) 的需求支援仍不足。在沒有適當的網路流量控管下，這些由即時性應用服務所周期性產生的聲音或影像串流，難保不受到網路封包遺失 (packet loss)、延遲 (delay)、以及傳送時間變動 (jitter) 的影響，尤其在網路擁塞的情況下會更為嚴重。為了進一步提供網路服務品質 (Quality of Service, QoS) 保證，IEEE發展了IEEE 802.11e服務品質保證規格標準[4]，提供在單一基地台下QoS資源分配機制。然而，對於具行動力的STA (Station[2]) 而言，快速地在基地台間順暢換手則是另一項維護網路QoS的基本需求，過長的換手過程會導致正在使用中的應用服務中斷或造成服務品質降低，進而影響到使用者的使用經驗與意願。

---

[1] Unlicensed Mobile Access (UMA) technology provides access to GSM and GPRS mobile services over unlicensed spectrum technologies, including Bluebooth and IEEE 802.11. Please consult http://www.umatechnology.org/ for more information.

[2] Any device that contains an IEEE 802.11 conformant medium access control (MAC) and physical layer (PHY) interface to the wireless medium (WM). This term is defined by IEEE 802.11 Standard [1].

無線網路下，影響整個換手過程優劣的因素主要有二項：第一是換手時機的選擇、第二是換手程序的長短。換手時機選擇的好壞，會決定無線通訊品質受到無線通訊通道好壞的影響程度；一般而言，我們會希望在使用中的通道還不算太糟的情況下，換到立即可提供更佳條件或者不久後可提供穩定及更佳的通訊環境的基地台，而這項性質往往必須取決於足夠目標基地台的量測資訊，使得 STA 可以在適當的時間點換手到適當的基地台。

由於 STA 在換手過程中會有一段不能通訊的時間，換手程序的長短則會影響使用中服務封包遺失 (packet loss) 個數和因換手造成的傳送時間變動 (jitter) 的變化性。使用者會希望換手過程愈短愈好、因而使得封包遺失數量愈少並且使得在目地端點封包與封包間到達的時間差 (inter-arrival time) 愈趨於一致。改良換手程序的長短可以讓換手過程直接受惠，在不影響安全性係數的條件下縮短換手程序，一直是相關研究的目標之一。

由於IEEE 802.11 標準[1] 並沒有制定WLAN換手過程的規範，製造商可以自由選擇實作，以作為產品及市場的區格。然而根據許多研究報告 [7]-[13] 指出，目前市面上常見的WLAN換手機制的效能，尚未能達到VoIP這類即時性應用服務的需求。因此，本篇論文即以此為出發點，深入探索WLAN的換手過程，嘗試釐清相關問題，並提供改善的做法。

1.2 研究目標

換手程序是影響整體換手過程一項重要的因素，IEEE在完成IEEE 802.11i增強安全協定標準[3] 與IEEE 802.11e服務品質保證規格標準[4] 後，針對需要快速換手需求的應用 (如VoIP)，草擬IEEE 802.11r基地台快速換手程序標準[6][3]。

草擬中的IEEE 802.11r基地台快速換手程序標準[6] 提到，STA換手進行過程中，邏輯上會歷經三個階段 (Stages)－

[3] 我們將 IEEE 正在制定中的 IEEE 802.11r 基地台快速換手程序標準[6] 以 **附錄 A** 的方式呈現，讓讀者可以了解「基地台快速換手程序」有那些快速換手的支援。

Discovery：行動端根據某種系統內部設計的演算法，不斷地尋找四周新的基地台，維護一串最新未來可能進行換手的目標基地台。因此 STA 能夠在察覺到通訊品質下降的情況下，立即做出反應，換手到最終目標基地台。

Resource establishment：在具有品質保證能力及加強等級安全機制下，STA 也許會先向目標基地台查詢、並且預先建立通訊安全加密金鑰以及配置無線傳輸頻寬資源，以確保新的基地台在STA 換手之後能立即提供STA 所需的頻寬以維護STA 正在進行中的通訊。

Transition：STA 執行的換手程序，將其無線存取點從舊基地台切換成新基地台，並且清除之前在舊基地台所使用的頻寬資源。

IEEE 802.11r基地台快速換手程序標準[6] 出現後，因換手程序所造成的延遲將可以獲得舒緩，同時該快速換手程序標準也支援服務品質協定的預先保留，進一步加強服務品質的保證。IEEE 802.11r基地台快速換手程序標準[6] 能有效縮短換手延遲的主要理由是該快速換手程序標準最佳化換手程序協定的協定花費 (protocol overhead)，大大減少了換手程序所需messages exchange的數量。然而即使有最佳化的換手程序協定的支援，STA若無法在正確的時序下執行相關的程序，或花費太多的資源在換手過程相關的維護而導致正常的傳輸時間減少，仍然會對整個換手過程優劣造成影響。

目前已經成功運行的電信行動網路的 STA，因為該行動網路的 radio technology 採用特殊設計的分時多工機制，電信行動網路的 STA 可以透過固定的時槽 (或稱 control channel)，進行周圍基地台的量測工作，並將量測資訊回報給基地台控制端點(Radio Network Controller, RNC)，最後由基地台控制端點執行符合整體網路運作利益的換手決策，這種換手方式稱為 Mobile-Assisted Network Control Handover。

然而根據IEEE 802.11 標準[1]，WLAN下的換手決策是由STA自行考量，雖然WLAN網路端不需要像電信網路那樣複雜的機制，但如果行動端沒有自己一套有效的目標基地台量測機制，將不足以在正確的時序下執行相關的換手程序，因而無法達成無縫換手的目標。

本篇論文 (Design and Implementation of a Topology-Aware Seamless Handover for IEEE 802.11 Wireless Networks) 的研究目標，在於設計並且實作一套適合於 IEEE 802.11 WLAN 無線網路下使用的無縫換手機制，讓使用 real-time service 的 STA 能快速並且順暢地進行換手。

為了設計一套適合WLAN使用的無縫換手機制，首先我們必須釐清目前常見的實作換手機制效能不佳的問題是什麼？接著針對這些問題進行改良，並且納入IEEE 802.11 標準[1] 規定之換手程序的需求，以系統運作的角度規劃我們換手機制所需的「基本功能方塊」、「換手決策方塊」，以及整個「換手機制的運作和各功能方塊之間的互動」。以實務性為出發點，我們希望由我們針對IEEE 802.11 無線網路設計的換手機制，能夠達到下列的目標：

1. 符合IEEE 802.11 標準[1] 的規範。
2. 僅利用IEEE 802.11 標準[1] 提供的各項基本服務。
3. 屬於實作層次上的改良。
4. 換手機制背後所需花費的代價相對較低、可以負擔，並是可行的。
5. STA 能夠有足夠的目標基地台資訊進行換手決策，經歷較佳的換手過程。

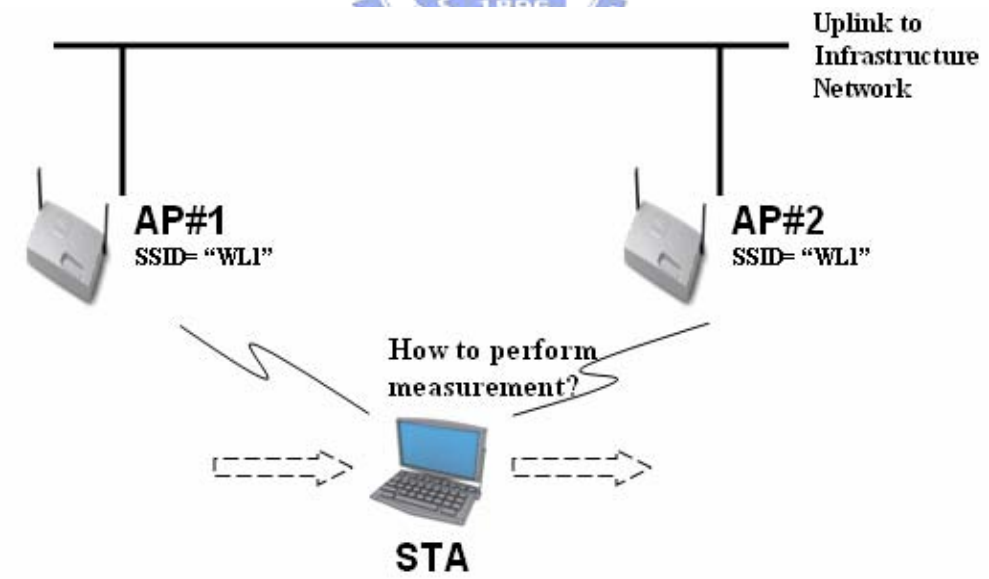


Figure 1–1 Diagram for a STA performing measurement during its movement

簡言之，我們的目標在於設計與實作一套有效率、使用低花費、並且以盡量不影響到正常網路系統使用效能為前提的內部換手機制，使得 STA 能有效率、持續不斷地進行其周圍可能換手的基地台的量測和資源維護的工作。我們所採取的設計基本理念，引入了 Topology-Aware 的概念，利用早期作業找出 STA 目標基地台的 Local Topology，最後針對這些數量不大的基地台，使用有效的方法進行量測與維護，即將目標基地台量測動作分散並且用在值得維護的目標基地台上。

1.3 章節簡介

本篇論文各章節簡述如下：

第一章　緒論，描述本論文的研究動機，以及本論文想要達到的目標。

第二章　背景知識與相關研究，說明本論文中討論主題的背景知識及相關研究，包含：無線網路下的換手動作、典型無線網路下使用的換手演算法、IEEE 802.11 無線網路目前普遍實作的換手程序、IEEE 802.11 無線網路換手機制的相關研究、及IEEE 802.11 無線網路安全協定與資源佈署進一步造成的延遲問題。

第三章　無縫換手機制之設計與架構，將說明我們所提出的無縫換手機制設計背後的原理與考量。包括：目標基地台搜索與量測機制使用的內部機制與設計原理、目標基地台搜索與量測機制的運作與無縫換手機制的整體運作。

第四章　無縫換手機制之實作，簡述系統實作所需了解的背景知識及我們所設計的無縫換手機制在系統層次的元件組成及功能運作。

第五章　效能評估，針對我們所設計的無縫換手機制系統實作，進行一連串相關的實際效能測試，最後提供相關的效能評估報告。

第六章　結論與未來工作，總結整篇論文，以及未來的研究方向。


E
S
A
1896

2.1 無線網路下的換手動作

無線傳輸和有線傳輸的主要差別在於傳輸的媒介 (media)，也就是傳輸通道(channel)。在傳統的有線傳輸，傳送端將電子訊號打在有外部遮蔽的金屬銅線上，訊號在金屬導線的導引下前進，因為有金屬導線的導引作用，訊號能量衰減率非常低，也因為外部遮蔽而較不易收到干擾。相對的，無線傳輸使用的傳輸媒介為空氣，訊號是以電磁波的方式散佈傳輸。由於沒有導引作用，訊號的能量快速衰減，其衰減程度與距離的平方甚至距離的四次方成正比；在這樣的傳輸模式下，能量是從訊號源的四面八方擴散與分佈。

2.1.1 微細胞基地台佈置 (Cellular Concept) 概念

由於無線傳輸的特性，想要單一基地台傳輸距離更遠、涵蓋範圍更大，便需要較大的發射能量；並且由於無外部遮蔽的因素，使用相同頻帶的訊號源會彼此干擾，相同空間中，相同頻帶下能夠使用的通訊容量便受到限制。

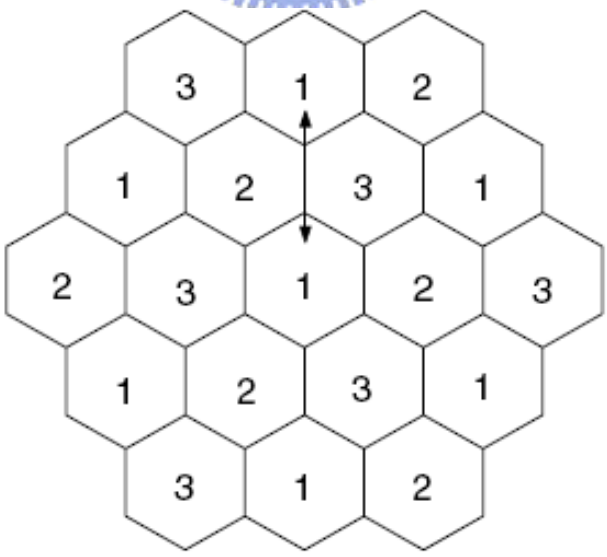


Figure 2–1 Cellular concept in wireless communications

(Source from [9])

為了減輕上述無線傳輸的問題，便有了微細胞基地台佈置概念 (cellular concept) 的出現。微細胞基地台佈置概念可以有效減輕無線傳輸的基本問題：**1.**頻譜使用效率 (spectrum efficiency) 及 **2.**使用容量 (user capacity) 的限制。單一基地台涵蓋的範圍可視為一正六角型的微細胞 (cell)，而大範圍的涵蓋可由多個微細胞組成。此外相鄰的微細胞使用不同的頻帶做傳輸，免除微細胞間彼此互相干擾的問題；至於不相鄰的微細胞則可以使用相同的頻帶，由於彼此的距離夠遠，對方的訊號到已方時已經相當低，因此可以視為雜訊的強度而不會對已方的通訊造成影響。在這樣的微細胞佈置概念下，頻帶可以重覆使用，因而總體的使用容量便可倍數增加。典型的頻帶分配只需要三個不互相干擾的頻帶即可涵蓋大範圍的面積，如 Figure 2–1 所示。

在微細胞佈署的無線通訊環境下，當使用者移動到使用中基地台的邊緣時，便需要進行所謂換手 (handover) 的動手。本質上，換手過程只是使用者將無線通訊的通道換成目標基地台運作的通道，更換無線傳輸的存取點而已。換手的過程看似簡單，但要在正確的時間點切換到正確的基地台，使得使用者察覺不出因為換手所造成的暫時服務中斷，則是一項高深的藝術；無線通訊系統需要特別的設計及管理機制才有辦法達成這一項目標。

2.1.2 IEEE 802.11 無線網路架構

IEEE 802.11 無線網路正確的全名是無線區域網路 (Wireless LAN)，顧名思義即將區域網路 (Local Area Network, LAN) 的使用無線化，因此也有人稱 WLAN 是 Wireless Ethernet。由於定位在於 LAN 的無線化，WLAN 適合使用在室內的環境。相對於商業化的 Cellular System，IEEE 802.11 WLAN 使用的是 unlicensed 的 2.4 GHz ISM (Industrial, Science, Medicine) 頻帶。由於運作在沒有執照的頻帶 (unlicensed band)，傳輸功率會有一定的限制，並且通常會使用展頻 (spread spectrum) 的調變技術，避免與其它使用相同 ISM 頻帶的無線通訊互相干擾 (例如衛星通訊、雷達)。

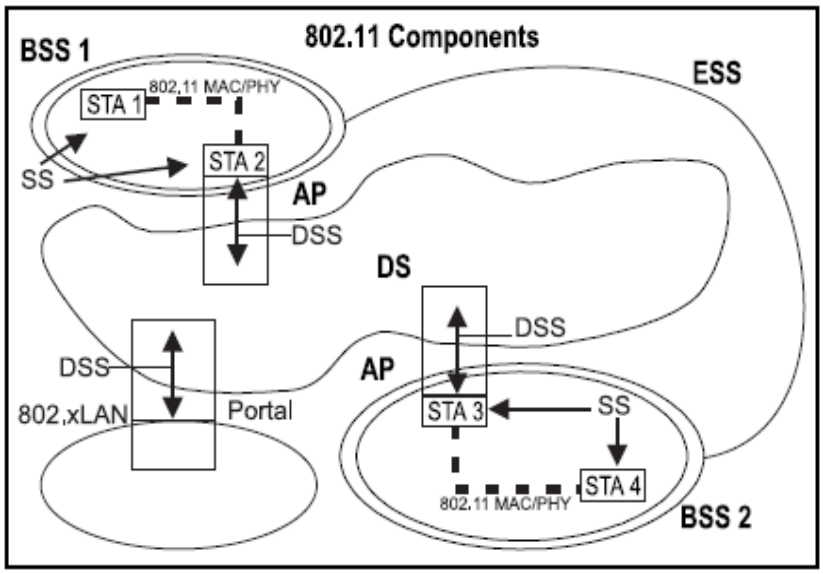


Figure 2–2 IEEE 802.11 architecture

(Source from [1])

IEEE 802.11 無線網路也採用了 2.1.1 中所提到的微細胞佈置概念，以IEEE 802.11標準[1] 的術語而言，一個cell是由多個運作在相同頻帶下、彼此在相互涵蓋的範圍下的STAs組成，稱為BSS (Basic Service Set)。若BSS中有一個特殊的STA，可以當作relay的端點，則此BSS下的STA不一定要全部相互涵蓋，只要每一個STA和此特殊的STA相互涵蓋即可，而這個特殊的STA則稱為基地台 (Access Point, AP)。

多個BSS可以經由Distribution System (DS) 相互連結而擴大WLAN的涵蓋範圍，這個由BSS拼湊出來、擴大的無線網路在概念還是屬於同一資料鏈結 (data link)，因此在不同BSS下的STA可以使用link layer address直接通訊；而這個由多個BSSs組成擴大的無線網路稱為Extended Service Set (ESS)。Figure 2–2 說明了上述的概念。

有了基本IEEE 802.11 架構的認識，現在進一步介紹相關的運作概念 [1]：

1. 同 BSS 下的 STAs 使用相同的 Wireless Medium (WM) 通訊。
2. 同BSS下的STAs並不直接通訊，而是經由AP再relay到對方[4]。

[4] IEEE 802.11e [4] 已經針對同 BSS 下的 STA 彼此通訊沒有效率的問題，提供了 optional Direct Link Service (DLS) 的 Station Service (SS)；同 BSS 下的 STAs 在使用 DLS 的 function 後，可以直接通訊。

3. 同ESS下、不同BSS的STA如果要通訊，要經由AP接上Distribution System (DS)，由DS的distribution service送到目標BSS的AP，再由目標BSS的AP relay給目標STA。DS使用Distribution System Medium (DSM) 通訊，802.11 標準[1] 沒有限定DS的實作。

4. IEEE 802.11 並不明確定義 WLAN 怎麼實作，而是定義邏輯上的共通介面。使用在 BSS 下的 function 稱為 Station Service (SS)；而使用在 DS 下的 function 稱為 Distribution System Service (DSS)。

5. STA 透過同 BSS 下的 AP 來使用 DSS。

6. 在相同的 ESS 下，同一時間點 STA 只能跟單一 AP 做連結 (association)。

而對於 WLAN 而言，使用者移動 (mobility) 的可能性可分成三類，說明如下：

1. No Transition: 使用者在連結 AP 下移動，沒有超過 AP 的涵蓋範圍。

2. BSS Transition: 使用者從一 AP 更改連結到另一個 AP，也就是本論文所討論的換手 (handover)。

3. ESS Transition: 使用者跨越ESS，到達另一個ESS，IEEE 802.11 標準[1] 並沒有支援此類的使用者移動。這類的移動，需要上層的網路協定處理，例如Mobile IP。

最後要提的是，由於功率的限制，WLAN 基地台的涵蓋範圍相對小了許多，也造成了使用者更頻繁的換手狀況。設計一套適合 WLAN 的換手機制，便更顯重要。

2.2 典型無線網路下使用的換手演算法

經過之前 2.1 的說明，我們大致可以了解無線網路運作背後的基本原理，接著我們要看的是典型無線網路的換手機制是如何設計的？

在無線網路下換手時機決策的問題，大致已經被研究得很完整，換手演算法要避免所謂ping-pong effect的情況發生，不要讓處在邊界的使用者因為左右的移動而造成短時間內執行來回換手程序的震盪效應。最常被拿來使用的方法是Relative Signal Strength with Hysteresis and Threshold, 在本論文中以Threshold-Based Handover Algorithm稱之。Figure 2–3 表示STA在兩個相臨的cell間移動，服務基地台 (Serving AP) 與目標基地台 (Target AP) 的SNR (Signal to Noise Ratio) 與時間的變化關係。

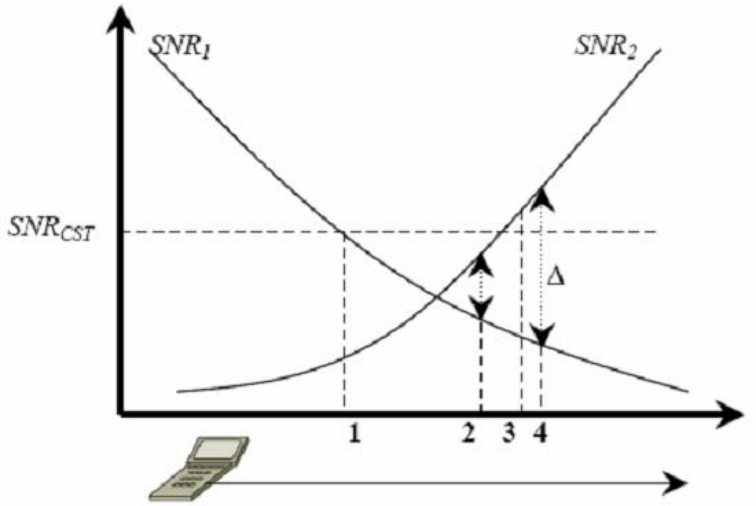


Figure 2–3 SNR change between old/new APs with Threshold-Based Handover Algorithm (Adapted from [18])

最佳的換手時間點在於正在服務基地台的訊號強度SNR_1 低於可以換手的訊號強度臨界值 SNR_{CST}，並且目標基地台的訊號強度SNR_2 大於服務基地台的訊號強度SNR_1 △值 (遲滯值) 以上才進行換手，在 Figure 2–3 中即為時間 4。用數學關係式表示即為：

$$SNR_1 < SNR_{CST}$$
$$SNR_2 > SNR_1 + \triangle$$

這樣的設計可以避免以下的情況產生：

1. 正在使用中的基地台還可以使用，但是因為有更強的基地台出現，而進行不必要的換手。

2. 在換到新的基地台後，使用者不會因為走回一、二步或無線通訊通道的多變性，又馬上換回舊基地台。使用者必須走到舊基地台的訊號強度 SNR_1 比新基地台的訊號強度 SNR_2 多△值以上才會再啟動另一次的換手程序，也就避免了上述的震盪效應。

不過這樣換手機制設計的前提，STA處在基地台邊界的時候，需要持續地測量所有可能的目標基地台，把 Figure 2–3 中目標基地台訊號強度變化線估計得愈準確愈好，其中sampling rate也是一項影響的factor。正在使用中的基地台的訊號強度變化線，因為正在使用中，可以隨時測量取得；而目標基地台的強度線則需要切換channel進行量測，在沒有充足的資訊或是特殊的安排下，目標基地台訊號強度變化線並不容易估計。因此，在大部分商業用途的無線通訊系統，都有提供某些特殊機制，讓STA在cell周圍時可以持續測量周遭所有可使用的基地台。例如：正在使用中的基地台告知STA在那些時間點切換到目標channel進行量測。

有一些研究H.H. Duong等[18] 想要利用Treshold-Based Handover Algorithm作為WLAN的換手演算法，但是由於目前IEEE 802.11 的標準[1] 並沒有提供周圍基地台資訊的機制[5]，因此只能夠使用標準所提供的active scan function做為量測的工具，其進行的方式如 Figure 2–4 所示。由圖中可看出當正在使用中的基地台的訊號強度SNR_1低於可以換手的臨界值 SNR_{CST}，STA即開始每隔T_{SI} (Scan Interval) 的時間利用active scan function執行Full Scanning (2.4.1.2)來對目標基地台進行量測。

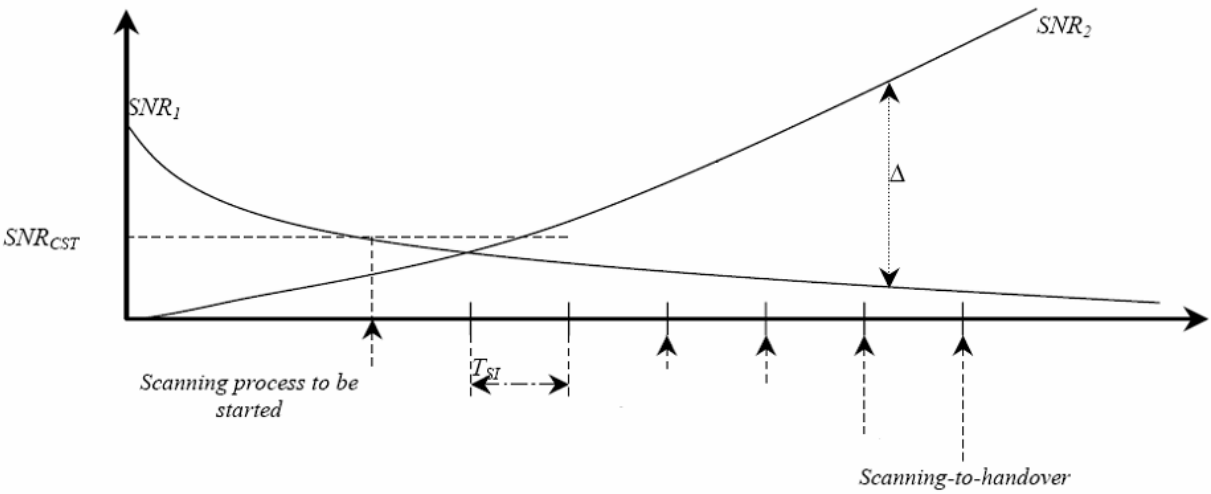


Figure 2–4 Timing diagram for using probe facility in 802.11 networks with Threshold-Based Handover Algorithm (Adapted from [18])

[5] 基於使用上的需求，IEEE 802.11 Working Group 也在 IEEE 802.11k[5] 開始著手制定有關 Radio Resource Measurement 相關的 Station Service (SS)，其中已經有提供 Neighbor Reports 的機制可以達到該效果。

這樣的設計基本上是不適合 IEEE 802.11 的無線網路的，其理由是 Full Scanning 作為量測工具不夠有效、並且花費太高。由於 Full Scanning 是利用 broadcast probe request 的方式對每個可能的 channel 偵測目標基地台，所有聽到 broadcast probe request frame 的基地台都會回應 unicast probe request frame，因此無線網路下的每個 STA 都得進行 Full Scanning 的代價太高，會造成不少的無線傳輸資源浪費在 Full Scanning 相關 frames 的傳輸上。

2.3 IEEE 802.11 無線網路目前普遍實作的換手程序[6]

觀察目前市售IEEE 802.11 的無線網路卡對於換手機制的處理，及A. Mishra等[7]的觀察分析報告指出，這些無線網路卡大部分只有實作監控服務基地台的連線傳輸品質，以此當作換手判斷的決策；並且這些普遍實作的換手程序，沒有將STA第一次進入基地台涵蓋範圍下使用無線網路及接下來後續換手的情況做出區別，都是用相同一套的處理機制。

當連線品質降低、開始有frames傳送不成功的過程中，網路卡會先試著用較低傳輸率、抗雜訊較強的調變方式來抵抗無線通道通訊品質的惡化；例如 802.11b規格下，STA 可以從 11 Mbps先降到 5.5 Mbps，再降到 2 Mbps及 1 Mbps。如果還是沒有辦法達到可接受的傳輸品質、大量frames仍然轉送失敗，則可視為STA已經走到連結基地台的最外緣。因此網路卡會決定換手，進行如 Figure 2–5 所示一連串的換手程序。

這樣設計的好處是，無線網路的網路端不需要複雜的管理機制，而且也可以節省目標基地台訊號量測的次數，只有在服務基地台完全無法使用的情況下，進行一次徹底的目標基地台搜索即可，這對 low-cost 定位的無線網路是比較實際的；畢竟，需要太複雜設定與維護的產品在使用上就是一個問題。

[6] IEEE 802.11 標準[1] 並沒有規定在 802.11-based 無線網路下的 STA 相關換手程序。因此我們討論的換手程序主要以目前普遍實作的換手程序為主。

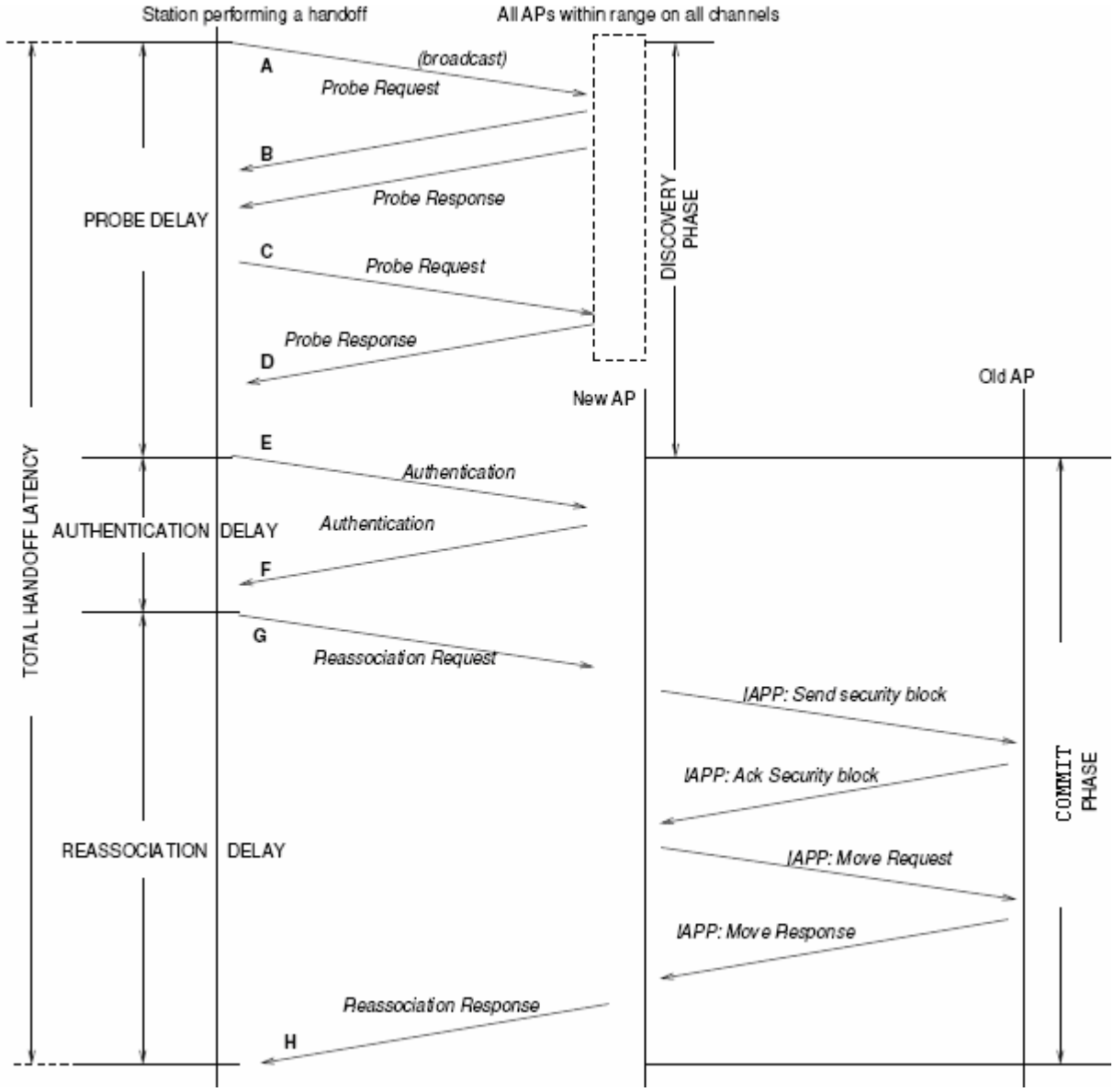


Figure 2–5 Observed handover process in major vendor's implementation

(Adapted from [7])

但是，這樣設計的結果是，換手過程需要比較長的時間，根據A. Mishra等[7] 觀察分析報告指出，換手過程多達 90%的時間花在搜尋目標基地台上，造成使用者服務中斷，這對不需要快速換手的應用 (如web, ftp) 影響不大，但對於有快速換手需求的即時應用 (如voice與video) 則有非常嚴重的影響。再者STA必須等到active scan function完成後才有辦法得知所有可能的目標基地台，以及是否已經到達ESS的邊緣，這對需要做vertical handover 的STA一樣不利。

對於常見實作的 802.11 換手程序，我們接著說明如下：

STA在服務基地台無法繼續使用後要進行換手，STA必須知道有那些基地台可以使用。因此，網路卡會先使用IEEE 802.11 標準[1] 所定義的active scan procedure ([1], Figure 2–7)來進行近端目標基地台的搜索偵測，也就是 Figure 2–5 中的Discovery Phase進行的部分。當完成近端基地台的搜索偵測後，網路卡得到一串可供換手的基地台，再利用換手演算法 (handover algorithm) 選出下一個連結的基地台，通常會是同網路 (ESS) 下訊號最強的基地台。接著網路卡進行link layer authentication exchanges，以及重新連結訊息交換 (reassociation exchanges)，當以上兩個程序都完成無誤後，STA即完成換手程序。

STA在決定不再使用舊基地台後 (決定換手)，再即時搜索目標基地台，在這個時間點搜索的結果保證是最新最正確的。P. Roshan & J. Leary所著一書[23] 稱呼這種實作在換手過程中的Discovery Phase為Roam-Time AP Discovery。

邏輯上可以把上述換手程序分成兩個部分，第一個部分是 Figure 2–5 中的 **"Discovery Phase"**，STA進行一段連續性的目標基地台搜索與量測動作。第二部分是 Figure 2–5 中的 **"Commit Phase"**，STA在這個phase進行的messages exchange是換手過程中，負責驅動換手前後基地台及STA本身狀態移動的訊息，我們稱在這個phase進行的messages exchange為**commit operations**。

Table 2–1 Two logical phases and other referred names

	Other referred names
Discovery Phase	Search Phase[8]、Scanning Phase[13]
Commit Phase	Execution Phase[8]、Re-authentication Phase[7]、Re-association Phase[13]、Transition Stage[6]

不同的學者對這二個Phases有不同的稱呼，但內涵是一樣的。Discovery Phase也有學者稱為Search Phase[8]、Scanning Phase[13]；而Commit Phase也有稱為Execution Phase[8]、Re-authentication Phase[7]、Re-association Phase[13]、Transition Stage[6]。我們將這些不同的稱呼，整理成 Table 2–1。

本篇論文會統一用法，使用 Discovery Phase 和 Commit Phase。

2.4 IEEE 802.11 無線網路換手機制的相關研究

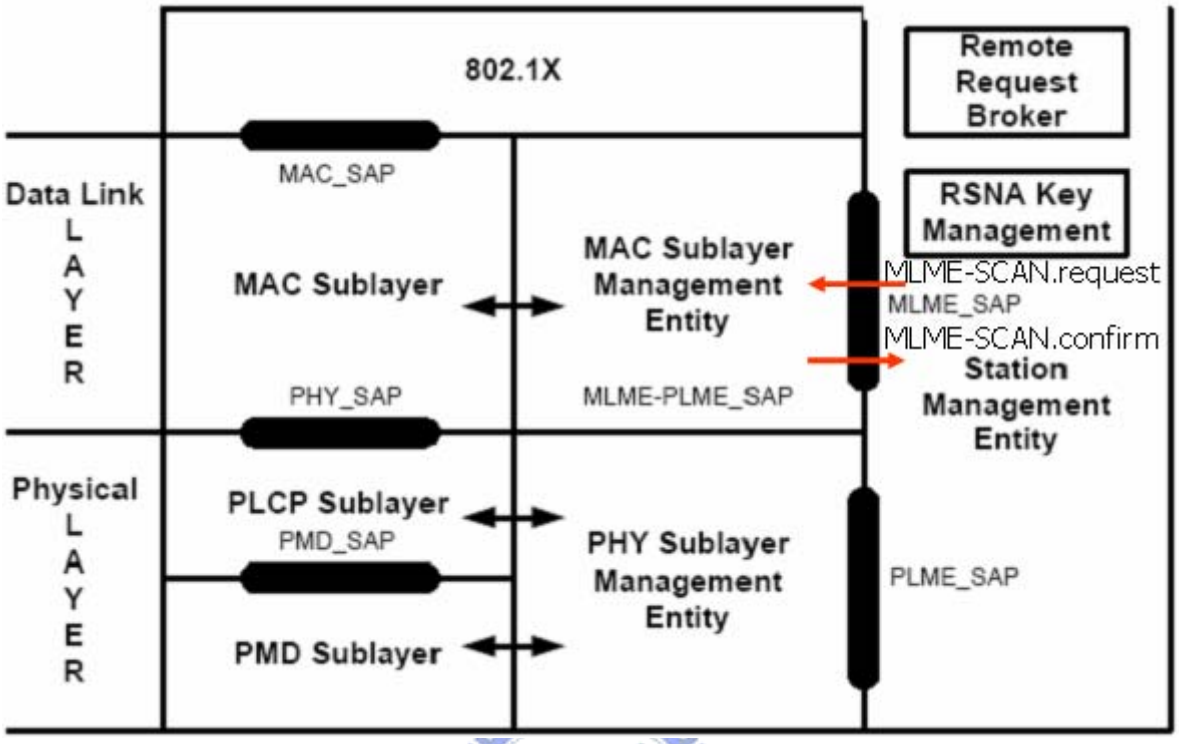


Figure 2–6 SME invokes MLME-SCAN.request and receives MLME-SCAN.confirm (Adapted from [6])

IEEE 802.11 標準[1] 提供了兩種進行基地台搜索 (AP Discovery) 的scan function： passive scan和active scan。如 Figure 2–6 所示，SME (Station Management Entity) 使用MLME_SAP提供的MLME-SCAN.request primitive進行active/passive scan；而active/passive scan的結果會以MLME-SCAN.confirm的方式回傳到SME。

顧名思義passive scan背動地等待基地台宣告它的存在，也就是等待基地台周期性發送的beacon；而active scan則採取主動的姿態，STA使用active scan function，會先發送broadcast destination的probe request frame，基地台在收到probe request後會回應unicast的probe response frame給STA[7]。若有多個目標基地台存在相同的channels，STA可以收到多個probe response的回應。

[7]active scan function 是 STA 發送 broadcast frame 詢問，由目標者使用 unicast frame 回應及提供訊息，而目標者也可以 broadcast 訊息的協定模式。在 network layer 下，STA 發送 router solicitation 主動詢問 router 的存在以及和 router 直接回應 router solicitation 及 router 定期 broadcast router advertisement 也是屬於這種訊息交換模式。

IEEE 802.11 標準[1] MAC機制並沒有針對這些probe request/response的frame有特殊傳送優先權，還是使用基本的DCF傳送機制。

目前大部分實作的 IEEE 802.11 無線網路基地台，beacon 發送的 interval 常預設為 100 TU (Time Unit)，而每個 TU 為 1024 μs。因此使用 passive scan 的情況下，STA 在每一個 channel 需要等待的時間在 worst case 情況下需要 100x1.024ms=102.4ms 才能得知是否有目標基地台的存在；又以 IEEE 802.11 無線網路目前普遍可用的 channel 數減去正在使用的 channel 是 10 個 channel，因此在毫無其它資訊下，STA 使用 passive scan 需要花 1.024 秒的時間才有辦法完成基地台搜索，得到所有可能目標基地台的資訊。

相較於passive scan 背動地等待基地台的宣告，active scan則採取較積極的作為，Figure 2–7 是從IEEE 802.11 標準[1] 中完整摘錄下來的active scan procedure：

For each channel to be scanned:

a) wait until the ProbeDelay time has expired or a PHY-RxStart.indication has been received;
b) Perform the Basic Access procedure (Distributed Coordination Function, DCF);
c) Send a probe with the broadcast destination, SSID, and broadcast BSSID;
d) Clear and start a ProbeTimer;
e) If PHY-CCA.indication(busy) has not been detected before the ProbeTimer reaches *MinChannelTime*, then clear NAV and scan the next channel, else when ProbeTimer reaches *MaxChannelTime*, process all received probe responses;

Clear NAV and scan the next channel.

Figure 2–7 Active scan procedure defined in IEEE 802.11 Standard

(Source from [1])

(a)項的程序，是為了STA剛換到新的channel，其medium busy的狀態 (Physical Sensing and Virtual Sensing[8]) 尚未與目標 channel同步，不可貿然送出frame，以免造成collision。因此在收到PHY-RxStart.indicatin或是等待ProbeDelay時候後即可確定Virtual Sensing的狀態。Matthew Gast所著介紹IEEE 802.11 無線網路的熱門書[22] 也注解等待ProbeDelay的時間，是為了避免在沒有基地台運作或是低使用率的channel下執行active scan function，因等待PHY-RxStart.indicatin所造成的Block現象。

(b)項所指是 IEEE 802.11 MAC 最基本的 Distributed Coordination Function (DCF) ，STA 要使用 DCF 搶到 channel 的使用權。(c)在成功送出 broadcast 的 probe request 之後，(d)(e)如果在 *MinChannelTime* 時間內 channel 都沒有 busy 的情況發生，表示目標 channel 沒有任何基地台存在；反之則目標 channel 有基地台存在，因而等待 *MaxChannelTime* 時間再進行下一個 channel 的搜索。

有不少的研究[8]、[11]、[12]、[13]等把(e)項解讀成：如果在*MinChannelTime*時間下沒有收到任何probe response則進行下一個channel的搜索，和IEEE 802.11 標準[1] 不符，應該是以訛傳訛的結果。不過在channel沒有人使用的情況下，上述兩者的結果是一樣的。

由上述的討論可得知，在沒有其它的資訊下，使用 active scan function 搜索基地台會比 passive scan function 快，其時間跟 scanned channel number *N*、*MinChannelTime*、和 *MaxChannelTime* 有直接的關係，並且：

$$N * MinChannelTime \le T_{\text{active scan}} \le N * MaxChannelTime$$

2.4.1 普遍實作之目標基地台搜索與量測程序的改良

部分研究以 active scan function 當作研究 WLAN 換手過程改良的重點，其中 active scan function 的三參數 scanned channel number *N*、*MinChannelTime*、和 *MaxChannelTime* 是最常見改善的部分。這些研究主要的想法不外乎是利用額外的資訊，移去「不必要搜索的 channel 數 (waste channel)」和「不必要的等待時間 (waste time) 」。

---

[8] Virtual Sensing 是 IEEE 802.11 標準[1] 下 MAC 採用 CSMA/CA 特有的 MAC function，用以輔助在 shared wireless media 的環境下，採用 distributed channel access 的無線通訊 MAC 協定僅使用 Physical Sensing 做為 Carrier Sensing 的不足。

2.4.1.1 Optimized Probe-Wait time

從一連串的實驗測量的分析結果，A. Mishra等[7] 提出，可以在無線網路offline的時間先對已經佈署好的基地台進行統計型的量測分析，並且依據這些統計的結果選取適當的*MinChannelTime*以及*MaxChannelTime*參數，使得大部分的基地台(例如 90%以上)都會在這些選取的時間內成功回傳probe response，因此在使用事先量測統計出來的*MinChannelTime*、*MaxChannelTime*，對Probe-Wait time便可以得到很好的tradeoff值。這篇報告的建議值*MinChannelTime*為 7ms而*MaxChannelTime*為 11ms。

另外H. Velayos[8] 建議的數據*MinChannelTime*為 1ms而*MaxChannelTime*為 10ms。這篇報告是利用Media Access Control的DCF function來做理論分析所推得的結果，如果channel在 1ms內沒有反應busy[9]，則可視為沒有目標基地台的存在。

2.4.1.2 Observed Scanning

在不知道目標基地台會被佈置在那些 channel 的情況下，STA 只好對每一個合法的 channel 都進行 active scan function，才能得到所有可能的目標基地台，這樣的基地台搜索方式稱為 Full Scanning。

相對於 Full Scanning，Observed Scanning 只對合法 channel 下的部分集合執行 active scan function，而這個 channel 部分集合可以是由之前 Full Scanning 的結果得到或是經由外部的資訊提供決定(例如已知某個 ESS 下的基地台只佈署在特定幾個的 channel 下，則 STA 在使用這個無線網路時，其 scan function 就可以只搜索這些特定的 channel)。

最常見的例子是，這個部分集合只包含 non-overlapping channels，因為這些 channel 最有可能使用來架設基地台；例如 2.4G ISM band 下的 channel #1, #6, 和 #11。在大部分的國家，這 3 個 channels 是最常用來架設基地台的 non-overlapping channels。STA 只搜索這些常用的 channel 可避免 Full Scanning 較長的基地台搜索，並且有極大的機率不會遺漏掉目標基地台的存在。

---

[9] Channel Busy 和沒有收到 Response 是兩回事，見 2.4 說明。

2.4.1.3 NG Probe Algorithm

M. Shin等[9] 建議在網路端建立基地台與基地台間的佈署關係及相關資訊，藉由這一項機制，網路端可以額外提供可能目標基地台資訊 (例如目標基地台所在的channel以及目標基地台的身份) 給STA，用以協助active scan function的進行。雖然需要付出額外管理基地台資訊的花費，但其實驗結果顯示，適當地使用額外目標基地台來協助active scan function進行的方式，可以比單純使用Full Scanning、Observed Scanning所需花費的時間縮減約 80%、30%以上。

M. Shin等[9] 提出的第一類基地台與基地台間的相關資訊，是相鄰關係 (Neighbor Relationship)，使用Neighbor Graph的資料結構來管理。STA可使用NG Probe Algorithm進行改良後的active scan function。這裡的相鄰關係是指：從基地台A可以換手到基地台B，則基地台B就是基地台A的相鄰基地台 (neighbor AP)，但反方向則不一定成立。

網路管理者可以根據地形和基地台的架設位置關係靜態設定基地台和基地台之間的相鄰關係，不過這種方式在無線通訊範圍沒有固定界線，並且易隨著時間變動的環境下，資訊不容易正確維護，管理上也較困難；網路管理者也可以使用基地台與基地台間通訊協定 (Inter-Access Point Protocol, IAPP[2]) 所提供的IAPP MOVE-Notify機制動態維護。

當基地台數量大的時候，動態機制維護是必要的。但即使有動態維護機制，在網路開始使用之前，也需要有基地台與基地台間相鄰資訊初始化的機制。關於這一點，M. Shin等[9] 使用的方法是：利用先期探訪的方式，把所有可能的路徑走一遍，利用網路端的IAPP協定便可以達到動態機制初始化。

此外M. Shin等[9] 也提到Neighbor Graph的資訊可以是集中管理的，儲存於特定的Information Server內；或者是分散管理的，儲存在每個基地台內。當使用者連結上基地台後，便可以透過使用中的基地台得到Neighbor Graph的資訊。Neighbor Graph的資訊可以是整個ESS下所有基地台相鄰關係的資訊，或只有正在使用中基地台的局部相鄰資訊，甚至特殊條件下局部資訊的子集合都是可能及可行的。M. Shin等[9] 的研究內容，採用了分散式管理的相鄰基地台關係，其利用IAPP動態機制維護在每個基地台上的局部Neighbor Graph。

```
for all channel i where any neighbor AP is running do
    Broadcast ProbeRequest on channel i
    Start probe timer
    while True do
        Read ProbeResponse
        if Medium is idle until MinT expires then
            break
        else if all APs on channel i have replied then
            break
        else if MaxT expires then
            break
        end if
    end while
end for
```

Figure 2–8 NG Probe Algorithm

有了Neighbor Graph的資訊後，STA可以使用這些資訊輔助active scan function的進行。使用Neighbor Graph資訊操作active scan function的程序如 Figure 2–8 所示。

利用 Neighbor Graph 的資訊，STA 只需要對目標基地台運作的 channels 進行 active scan function 即可，減少了不必要搜索的 channel 數；當 STA 在某一個 channel 都已經得到 Neighbor Graph 資訊中提供的目標基地台量測，便不需要再等待到 MaxChannelTime 就可以進行下一個 channel 進行搜索，減少不必要的等待時間。

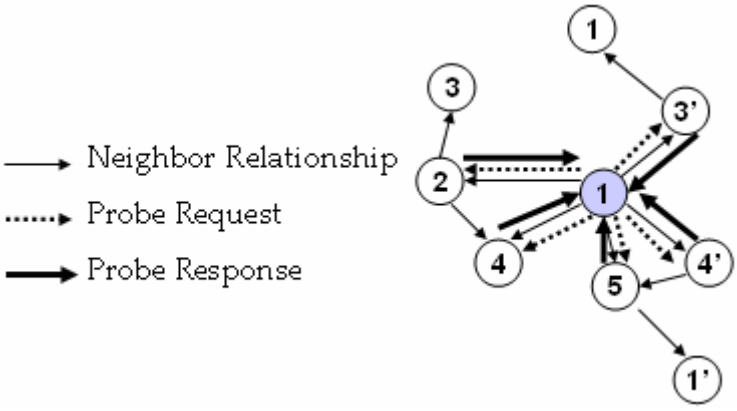


Figure 2–9 Demonstration of active scan using NG Probe Algorithm

如 Figure 2–9 所示，節點 1 是服務基地台，運作在channel #1 下，而Neighbor Graph提供的資訊顯示可能的目標基地台節點 2、節點 3′、節點 4、節點 4′、及節點 5，分別運作在channel #2, #3, #4, 和#5 下，STA有了這個資訊可以只對channel #2, #3, #4, #5 而不是所有的channels，執行active scan function。並且STA在量測到該channel的所有目標基地台的訊號強度後 (例如在channel #3 量到節點 3′後)，即可進行下一個channel的搜索。

2.4.1.4 NG-pruning Probe Algorithm

M. Shin等[9] 提出的第二類基地台與基地台間的相關資訊是Non-Overlapping Relationship，使用Non-Overlapping Graph的資料結構管理。STA可以更進一步使用NG–pruning Probe Algorithm來進行改良後的active scan function。這裡的Overlapping關係是指：若基地台A和基地台B間，存在有一地點，使得STA分別對基地台A和基地台B量測的訊號強度都在某臨界值以上 (隱含STA和基地台A和基地台B都可以有某程度以上的通訊品質)，則基地台A和基地台B互為Overlapping基地台。

由整個 ESS 下所有基地台為節點所形成的 Fully-Connected Graph 減去整個 ESS 下以基地台為節點、具有 Overlapping 關係的兩個基地台連成一 edge 所形成的 Overlapping Graph，即為整個 ESS 下以基地台為節點的 Non-Overlapping Graph。

由 Neighbor Graph 和 Overlapping Graph 的定義可推得：

$$\text{Neighbor Graph} \subseteq \text{Overlapping Graph}$$

Non-Overlapping Graph內的每一條edge所代表的意義是，若在某個地點量測基地台A的訊號強度大於可以使用的臨界值以上，表示另一基地台B在這個量測的地點一定無法在可用臨界值以上，因此我們可以更進一步的「剪去」基地台B，不需要再對基地台B進行量。M. Shin等[9] 指出，利用這一項資訊可以再進一步縮減不必要搜索的channel數和不必要等待的時間。

```
While not all neighbors APs are probed or pruned do
    Select channel i with maximum edge degree
    Broadcast ProbeRequest on channel i
    Start probe timer
    while True do
        Read ProbeResponse from APr
        if Medium is idle until MinT expires then
            break
        end if
        prune all APs non-overlapping with APr
        if all APs on channel i have replied or be pruned then
            break
        end if
        if MaxT expires then
            break
        end if
    end while
end while
```

Figure 2–10 NG-pruning Probe Algorithm

STA使用Neighbor Graph再加上Non-Overlapping Graph的資訊輔助active scan function的進行。其程序如 Figure 2–10 所示。

NG-pruning Probe Algorithm使用Neighbor Graph的資訊，及相鄰基地台的Non-Overlapping的資訊操作active scan function。其搜索的次序相對於NG Probe Algorithm則有特別的安排，是從Non-Overlapping Graph中order[10]最大的Neighbor Node點開始。

[10] 節點的 Order 指的是 Directed Graph 中，節點往外的 edge 數總合。例如：在 Figure 2–11 中，節點 2 的 order 為 2。

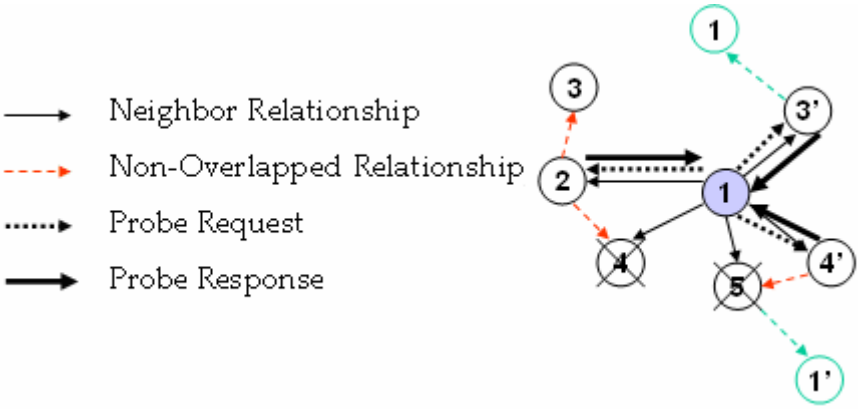


Figure 2–11 Demonstration of active scan using NG-pruning Probe Algorithm

如 Figure 2–11 所示，節點 1 是服務基地台，運作在channel #1 下，而Neighbor Graph提供的資訊顯示可能的目標基地台處在channel #2, #3, #4, 和#5 下。此外節點 2 的Non-Overlapping基地台有節點 3 和節點 4、節點 5 的Non-Overlapping基地台有節點 1′、節點 4′的Non-Overlapping基地台有節點 5，節點 3′的Non-Overlapping基地台有節點 1。Figure 2–11 中節點 2 是節點 1 所有相鄰基地台中，擁有最大Non-overlapping order的相鄰基地台。

STA 有了這個資訊可以只針對 channel #2, #3, #4, #5 執行 active scan function，根據Neighbor基地台的Non-Overlapping order大小，第一個搜索的是位在channel #2 的節點 2，其量測結果大於可使用的臨界值，根據 Non-Overlapping 基地台的關係，位在 channel #4 的節點 4 就可以被排除在接下來搜索名單內。但由於 channel #4 仍有節點 4′，所以仍然要搜索 channel #4，不過只要量測完節點 4′ 的訊號強度後，即可再進行下一個 channel 的搜索。利用以上程序可以對所有可能的目標基地台進行量測，並且在量測的過程中又可利用 Non-Overlapping 的性質，動態「剪去」不必要量測的基地台。

值的一提的是，這種基地台間 Non-Overlapping Relationship 的性質，最常出現在兩個彼此處於相對的方向 (opposite direction) 的基地台身上。適當的使用 Non-Overlapping 的資訊，的確可以進一步避免不必要的 channel 搜索以及不必要的等待。

2.4.1.5 Intelligent Channel Scan

K. Kwon and C. Lee [10] 以IEEE 802.11 標準[1] 下MAC所使用的DCF原理，及相關假設，設計了一套以Probe Response是否碰撞的事實，智慧地推斷是否該channel下的目標基地台已經全部偵測完畢，因而可以節省STA等待probe response的時間。

這個機制所根據的二個假設是：

1. probe response 有較高的優先權，即使基地台有待送的封包，在收到 probe request 後，基地台會優先而且「立刻」送出 probe response frame。
2. STA 可以從 data frame 截取 BSSID 的資訊。

K. Kwon and C. Lee [10] 的第 1 個假設看起來似乎合理，但以碰撞與否當作是決策的條件在現行的標準下是行不通的。以現有的 802.11 標準而言，並沒有針對probe response frame有特別傳送優先權的設計，仍遵循DCF、使用unicast frame機制傳送，並且probe request和probe response並非atomic operation[11]，因此雖然會有碰撞的機率，但並不是絕對的。第 1 個假設除非修改IEEE 802.11 標準[1]，否則不易達到。但修改已經運作的MAC protocol的機會從實務層面來考量並不大。

2.4.2 目標基地台搜索與量測機制的重新設計

除了針對Discovery Phase下所使用active scan function進行改良的相關研究之外，部分研究提出改良的做法是針對STA量測目標基地台的時機與模式進行重新設計。有的研究將 Figure 2–5 中的Discovery Phase提前，也有的研究提出完全不做 Figure 2–5 中的Discovery Phase，或者是改變目標基地台量測的模式，使其量測的行為不那麼集中，而改採用分散模式搜索目標基地台。這些ad hoc的方法，目標都是讓換手過程只剩下Commit Phase。

---

[11] 在 IEEE 802.11 標準的 MAC 機制，frame 和 frame 間如果必須相隔 SIFS (Short Inter-Frame Time) 時間，則稱這些 frame 組成一個 atomic operation。例如：傳送端送出 unicast data frame 後，接收端必須在 SIFS 後立刻送出 Ack frame，這是 IEEE 802.11 標準採用的 positive ack 機制，Data/Ack 組成 atomic operation。

將 Figure 2–5 中的Discovery Phase自換手過程中移除，主要是因Discovery Phase佔了換手過程超過 90%的時間，把Discovery Phase從換手過程拿掉，可以使換手過程時間縮短，減少換手延遲的時間。

2.4.2.1 Background Probing

換手過程 Figure 2–5 中的Commit Phase所做的messages exchange會影響換手前後的基地台、及STA內部的 802.11 Protocol State Machine (Figure A–3) 的改變，以及STA在DS上的連結點狀態的改變。因此這些messages exchange是不可移除或提前的。不過IEEE 802.11[1] 標準可以讓link layer authentication messages exchange提前執行；但因為link layer authentication messages exchange所佔的時間比例不大，提前執行對換手的效能改善並不會有太大的影響。

相對於Commit Phase而言，Figure 2–5 中的Discovery Phase佔了整個換手程序90%以上的時間[7]，為了改進換手延遲，在實作上可以提前執行Discovery Phase。如果再採取分散進行量測的模式，網路卡的行為模式會變成Transmit/Receive Phase[12] 和Discovery Phase呈現片斷交錯出現的情況。將量測基地台的動作提前並且與正常的資料傳輸平行進行的這種模式稱為 Background Probing。Background Probing可以採用active scan或者是passive scan；也可以單次只花較短的時間搜索少量的channels，目的是不要干擾或犧牲網路卡正常的傳送動作。

Background Probing要如何進行及如何評量效能通常就沒有標準可言，這也是一門實作層次的Black Art。大部分的實作會以服務基地台的訊號強度為指標，設計起動Background Probing的Cell Search Threshold。

Background Probing也有不同的名稱：P. Roshan and J. Leary所著一書[23] 稱之為 ”Preemptive AP Discovery”，H. Velayos等[8] 稱為 ”Detection and Search Phase Running in Parallel”，而N. Mustafa[12] 等稱之為 ”Pre-Scanning”。

---

[12] 相對於 Discovery Phase、Commit Phase，我們使用 Transmit/Receive Phase 指稱網路卡在連接基地台後，正常傳送、接收封包的期間。

2.4.2.2 Selective Scanning and Caching

S. Shin等[11] 使用了計算機架構memory hierarchy常用的概念cache技術，把Figure 2–5 中的Discovery Phase從換手程序中移除。Scan function找到的目標基地台資訊，就好比是CPU對速度相對較慢的Memory讀取資料；目標基地台Caching的機制，就如同CPU將已經讀取過、沒有變動的資料暫存在可以快速access的cache內，供下次讀取時，能快速access。

Table 2–2 Cache structure for Selective Scanning and Caching

	Key	Best AP	Second Best AP
1	MAC1 (Ch1)	MAC2 (Ch2)	MAC3 (Ch3)
..			
10			

(Source from [11])

Table 2–2 是 Selective Scanning and Caching 方法所用的cache structure，以服務基地台當作是Key (相對於memory address)，而由active scan function所得到的最強的前兩個基地台當作內容 (相對於memory content)。

每次需要進行換手時，就先查詢 cache structure 是否有以服務基地台的 MAC address 為 key 的 entry，若 cache hit 則直接對該 entry 內容所提供的基地台進行連結的動作，若第一個失敗則再試第二個基地台，都失敗則改採 Selective Scanning 進行即時搜索。

Cache structure內entry的新增和修改時機，發生在cache miss或cache hit但entry內容所提供的基地台卻無法進行連結的情況下。在這些情況下，STA會使用與 Figure 2–5 相似的換手程序，先進行Discovery Phase，然後再根據Discovery的結果進行Commit Phase。不同的地方在於STA在Discovery Phase使用的目標基地台搜索量測機制是採用Selective Scanning、而非Full Scanning。STA在完成換手程序後，會將在Discovery Phase搜索到訊號強度排名前二的基地台資料，新加入一筆或更改原本就已存在於cache structure中、以換手之前基地台的MAC address為key的entry。

據S. Shin等[11] 說法，當cache hit後嘗試二次猜測性的目標基地台連結都失敗的時間可以利用實作技巧使其不會超過 6 ms，因此額外的penalty不會太大，但有極大的機率可以順利連結上目標基地台。而cache miss所使用的 Selective Scanning也會比Full Scanning效果好，因此整體效果會比市售普遍實作的換手機制要好。

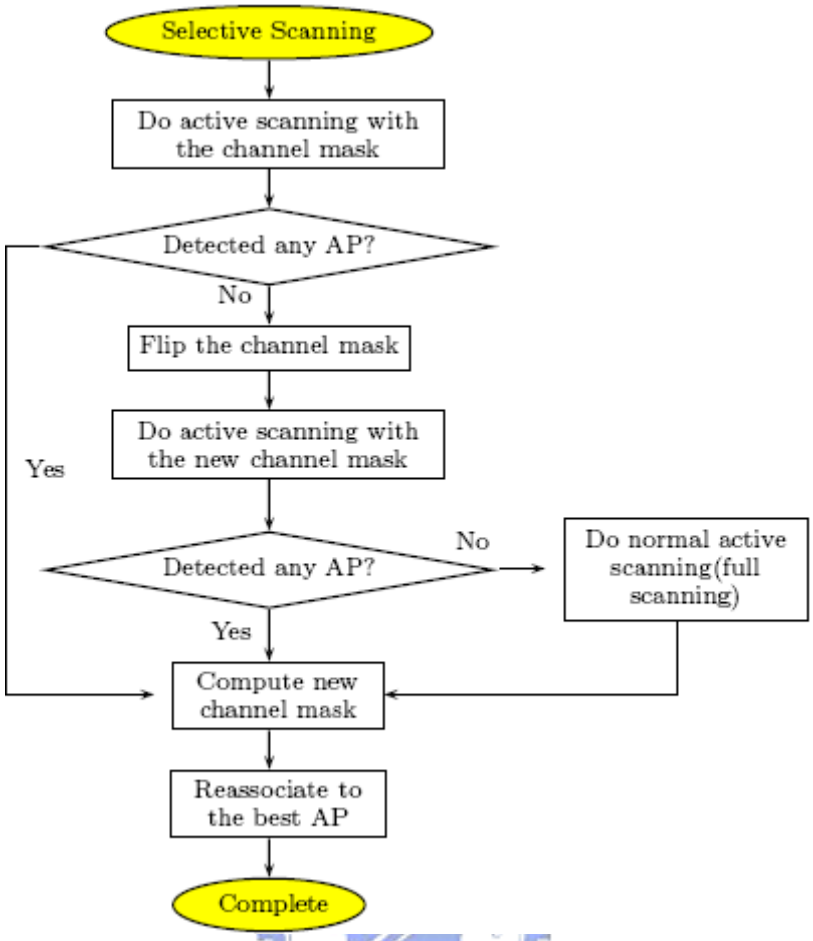


Figure 2–12 Selective Scanning procedure

(Source from [11])

Selective Scanning基本上就是 2.4.1.2 所提的 Observed Scanning，不過額外使用一個channel mask bitmap資料結構，記載選取的channel子集合，當第一次 Observed Scanning 結束後也找不到任何目標基地台時，便將channel mask反轉產生新的channel mask，再搜索另一半的channels。Figure 2–12 圖解了Selective Scanning procedure。

Selective Scanning and Caching最佳的效能出現在cache hit的情況，並且cache所提供的目標基地台正是正確的基地台，而最差的情況為cache miss或是一開始cache內沒有資訊的時候 (所謂cache cold start cost)，需要大約Selective Scanning花費的時間。這個方法在某些環境下的確可以看出其效果，但變動性和不確定性仍大。

2.4.2.3 Pre-Scanning and Dynamic Caching

N. Mustafa等[12] 基於S. Shin等[11] 的概念，考量cache可能有不夠新鮮的情況，將 2.4.2.2 Selective Scanning and Caching 和 2.4.2.1 Background Probing 融合成 Pre-Scanning and Dynamic Caching 的技巧。根據N. Mustafa等[12] 的說法，Pre-Scanning可以保證cache的新鮮性及正確性，換手效能會比 Selective Scanning and Caching 來的好。

2.4.2.4 SyncScan

觀察IEEE 802.11 無線網路下的基地台可以發現，基地台會定期經由broadcast的方式發送beacon訊息供STA更新及同步狀態使用，通常使用的週期是 100 TU[13]，約 102.4 ms。如果STA可以知道目標基地台發送beacon的時間，STA便可以直接使用passive scan function，在預期的時間區段內切換到目標channel，背動地等待基地台定期beacon的發送。在超過預期的時間區段後，即可再切換回原來運作的channel，重新繼續原本的封包傳送。在有基地台發送beacon時間資訊的條件下，STA便不需要等待完整的beacon interval即可搜索及量測目標基地台。這種在STA有目標基地台發送beacon時序資訊，因而保留短暫時間用來和目標基地台beacon發送做同步量測的動作的模式，本篇論文稱其為Synchronized Scan (SyncScan)[14]。

I. Ramani and S. Savage [13] 指出IEEE 802.11 標準[1] 對於相鄰基地台間beacon發送時序的關係，並沒有特別的限定。I. Ramani and S. Savage [13] 於是利用這個實作層次的自由度，讓每個基地台發送beacon的時間有特殊的安排，STA就可以不用外部資訊即可得知每個目標基地台發送beacon的時間區間，因此STA就可以在特定、預期beacon會發送的時間區間，切換channel去執行passive scan function，以判斷是否有目標基地台存在並且可以順道量測目標基地的訊號強度。

---

[13] 1 TU = 1024 μs in IEEE 802.11 標準[1]。

[14] 本文所稱的 SyncScan 定義與 I. Ramani and S. Savage [13] 所定義範圍不同，見註[15]。

I. Ramani and S. Savage [13] 採取的作法是，設計一套依據channel number安排好的週期性基地台beacon發送計劃時刻表，讓每個運作在不同channel下的基地台錯開發送beacon的時間。而運作在相同channel下的基地台，在發送beacon時再隨機微調目標時間點，以避免相同channel下基地台發送beacon可能的碰撞。

有了這個安排好的beacon發送計劃時刻表，STA便可以把 Figure 2–5 中的Discovery Phase從換手程序中移除，只剩下Commit Phase，加快換手速度；此外STA還可以持續地對目標基地台進行一連串的訊號量測動作，這種目標基地台探索與量測模式，可提供更多的資訊進行換手決策。

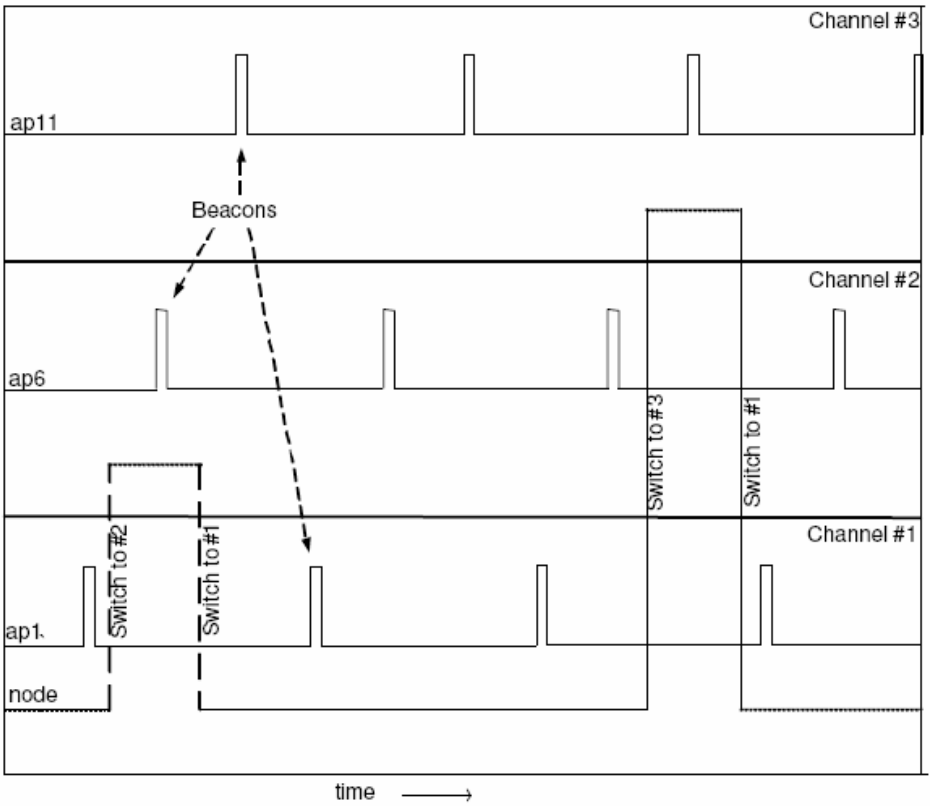


Figure 2–13 Timing diagram for SyncScan operations

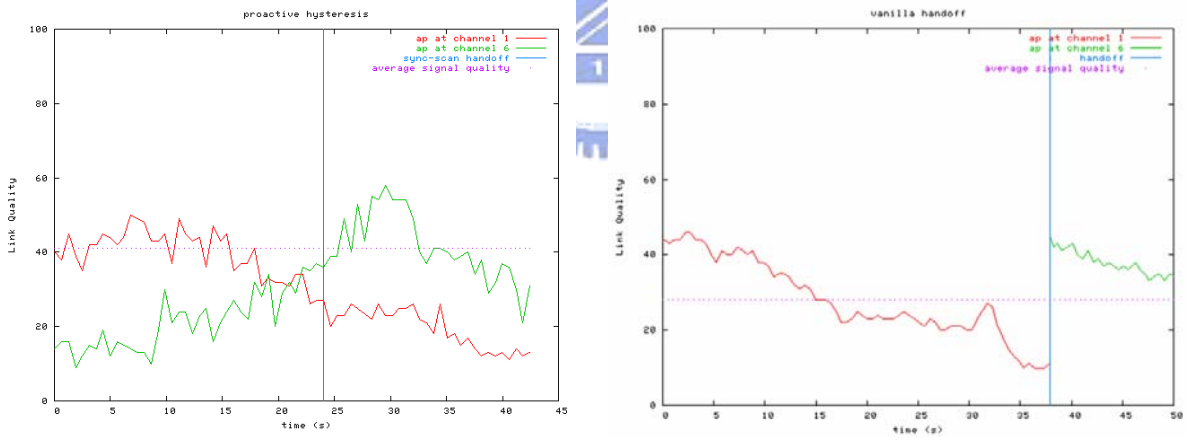
(Source from [13])

以下進一步說明I. Ramani and S. Savage [13] 所設計的beacon發送計劃時刻表的詳細規格：

如 Figure 2–13 所示，工作在channel #1 的基地台在t時間點發送beacon，工作在channel #2 的基地台在t+d時間點發送beacon，而工作在channel #3 的基地台在t+2d時間點發送beacon。因此，運作在channel #c的基地台連結的STA，若想要偵測或量測channel #(c+1)下的基地台，只要在目前的channel下聽到beacon後的d時間切換到channel #(c+1)即可。I. Ramani and S. Savage [13] 稱呼此目標基地台探索與量測模式為SyncScan，但本論文稱為這種作法為AP-Aligned SyncScan[15]。

根據上述基地台 beacon 發送時間表的安排，對於 total 為 n 個 channel，以 d 時間隔開發送時間的 IEEE 802.11 無線網路，會有以下的結論：

1. 運作在相同 channel 下的基地台，發送 beacon 的 phase 是一樣或相近的。
2. 每個基地台發送 beacon 的週期是一樣的，長度為 (n-1)*d。
3. STA 可以按照 beacon 發送時間表，從自己連結的基地台發送 beacon 的時間開始點開始，每格 d 時間切換一次目標 channel，在一個 beacon 發送週期後，把所有目標基地台偵測一次。



(a) Under Threshold-based algorithm (b) By Observed vendor implementation

Figure 2–14 AP's SNR diagram measured by STA for two AP discovery schemes (Source from [13])

[15] I. Ramani and S. Savage [13] 定義此作法為 SyncScan，與本論文所指的 SyncScan 範圍有些微的差距。本論文進一步依據基地台有無特別的 beacon 發送時刻表(Schedule)，將 SyncScan 再分為 AP-Aligned SyncScan 和 Non-AP-Aligned SyncScan。

據I. Ramani and S. Savage [13] 表示，AP-Aligned SyncScan的目標基地台探索與量測模式，可以有以下的好處：

1. AP-Aligned SyncScan的目標基地台探索與量測模式，可以有效取代掉 Figure 2–5 中的Discovery Phase，因而使得換手程序只剩下 Figure 2–5 中的Commit Phase，增進換手的效能(約幾個millisecond)。

2. STA的換手時機決策，可以使用 2.2 提到的Threshold-Based Handover Algorithm，使得STA做出更好的判斷，使得平均傳送與接收使用的訊號強度較佳，提昇無線通訊傳輸品質，如 Figure 2–14 所示。

3. STA可以利用這些持續量測到的目標基地台訊號，使用RADAR[15] 之類client端定位技術，達到持續性的location tracking的功能。

2.4.3 非使用目標基地台搜索與量測機制的換手方式

2.4.3.1 Location-based Fast Handoff

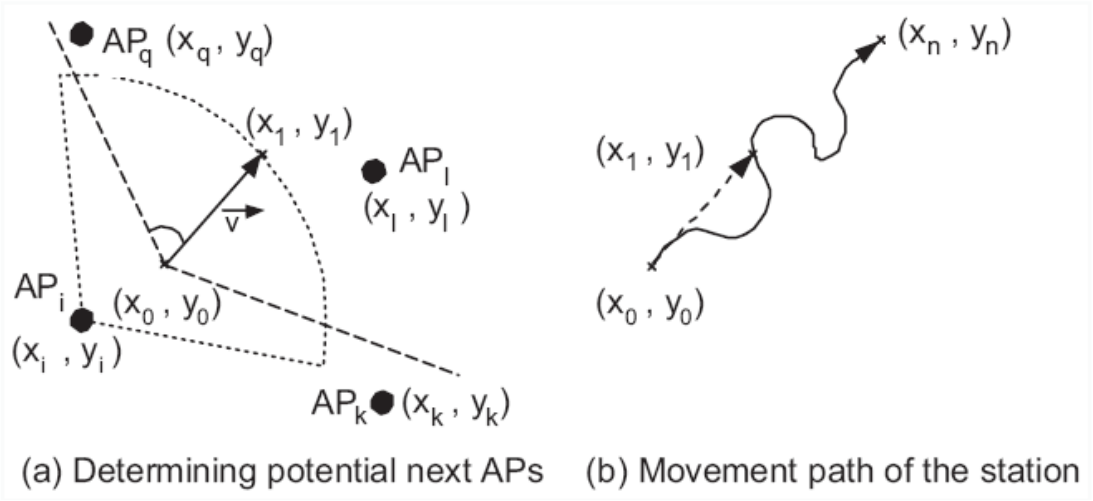


Figure 2–15 Deriving candidates AP sets on movement behavior

(Source from [14])

C. C. Tseng等[14] 提出額外引入另一dimension的資訊：地理資訊座標位置。利用這個dimension的資訊，STA可以將 Figure 2–5 中的Discovery Phase自換手程序中移除。而地理資訊的來源可以是GPS、sensor network、或是其它localization技術。

如 Figure 2–15 所示，STA從特定的location server獲得帶有座標位置的網路佈建資訊圖，再依據相隔兩次定位 (Positioning) 所得的座標位置推得行進方向，便可以在帶有座標位置的網路佈建資訊圖中推出目標基地台的次集合，當服務基地台弱到某程度的時候，STA可以不做任何的目標基地台搜索的動作，直接從次集合中和行進的方向推得最有可能的基地台，直接進行 Figure 2–5 中的Commit Phase相關的messages exchange。C. C. Tseng等[14] 稱呼僅執行 Figure 2–5 中的Commit Phase下的commit operations為AP Direct Association。

這個方法需要相對較精確且額外的定位技術，其佈署代價和所能獲得的效益仍待進一步的評估。利用定位技術輔助換手程序，據C. C. Tseng等[14] 的說法，可以將可能目標基地台的數量縮小，在Robust Security Network下使用繁複的安全協定的換手程序的情況下，對於STA在執行Pre-authentication及Proactive Key Distribution等預先安全佈署機制，會有很大的幫助。

2.5 IEEE 802.11 無線網路安全協定與資源佈署進一步造成的延遲問題

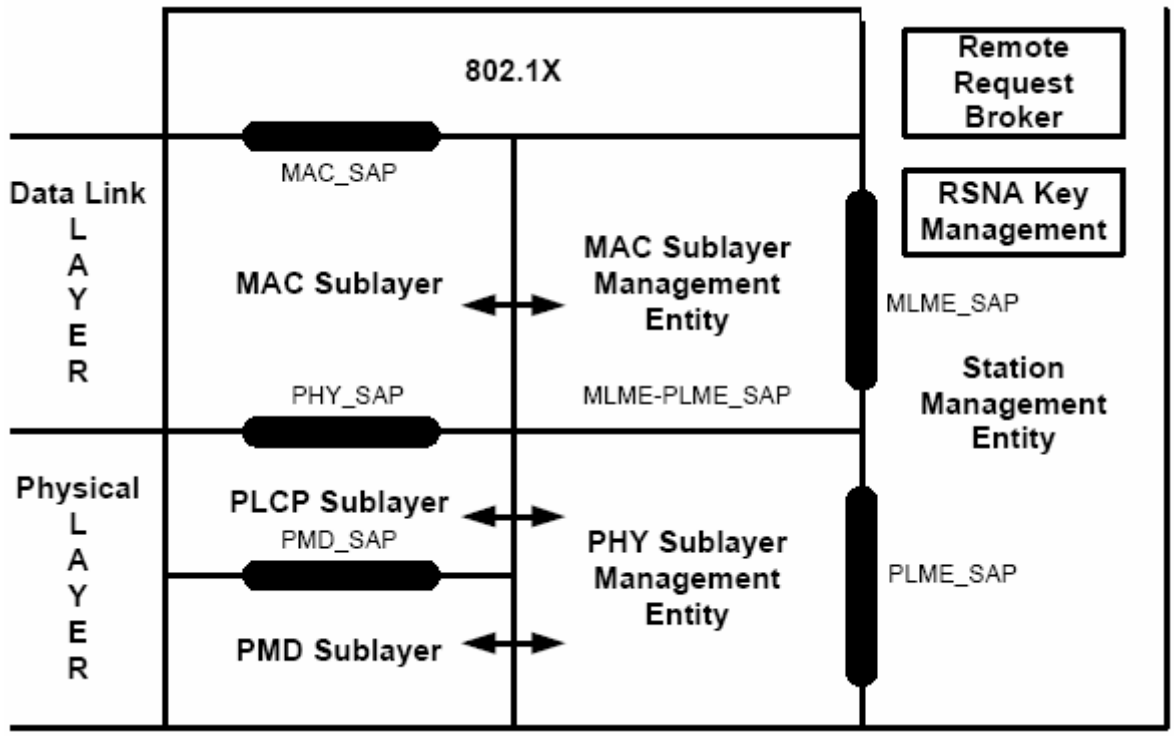


Figure 2–16 802.11 basic reference model

(Source from [6])

最初IEEE 802.11 標準[1] 採用的安全協定稱為WEP(Wired Equivalent Privacy)，提供了link layer authentication和簡單的ciphering功能。但是由於WEP安全協定下的認證機制使用的是static key，並且直接拿來當作ciphering key，使得非但達不到安全的要求，反而造成更大的不安全。

IEEE 802.11 Working Group後來發展新的IEEE 802.11i增強安全協定標準[3]，採用802.1X Port-based、可擴展式認證協定 (Extensible Authentication Protocol) 為架構，將Authentication和Access Control的層次，從 Figure 2–16 中的MAC Sublayer功能方塊，橫跨到 802.1X功能方塊。而原來的link layer authentication則保留並使用Open System不再做認證，以維護原始IEEE 802.11 Protocol State Machine (Figure A–3)的一致性。

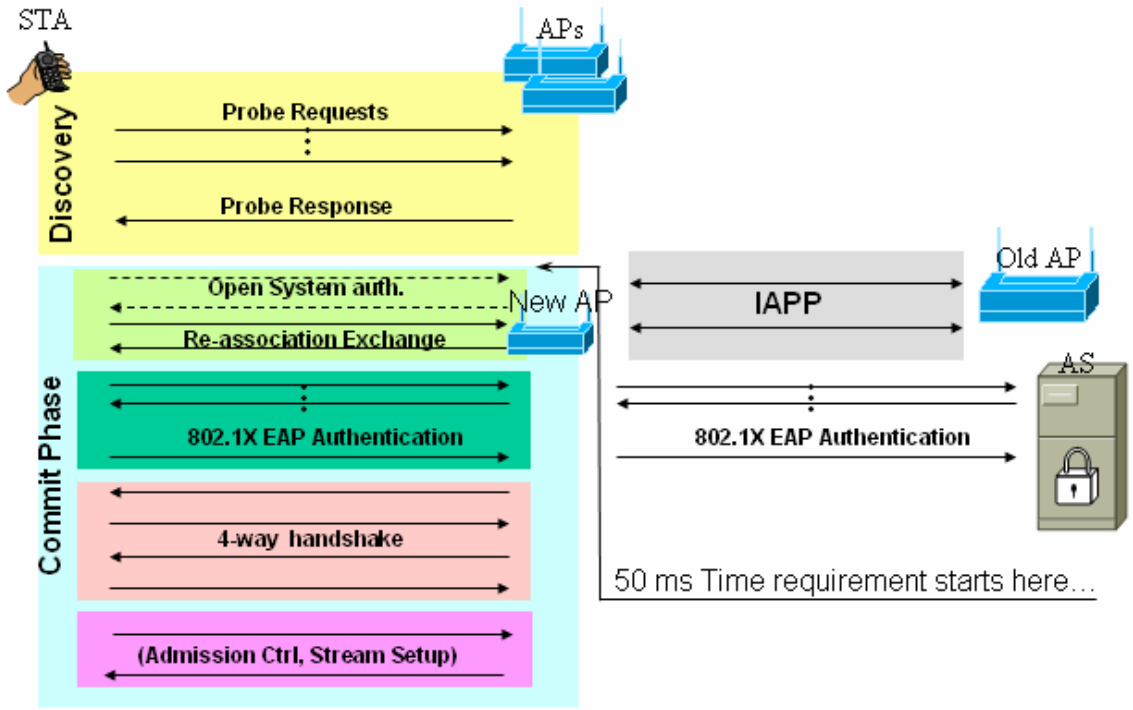


Figure 2–17 Handover messages exchange under 802.11i RSN

IEEE 802.11i增強安全協定標準[3] 新增 4-Way Handshake做為Access Control的驗證，並且使用dynamic key的認證機制。而傳送資料用的ciphering key則是在 4-Way Handshake的驗證程序中動態產生。802.1X認證是由一連串的EAP認證訊息完成以User為基礎的認證程序。在完成 802.1X認證後，STA會動態產生 4-Way Handshake驗證需要的PMK (Pairwise Master Key)[3]。

STA第一次進入該無線網路時，這些相關的認證程序必須在 Figure 2–5 中的Commit Phase相關messages exchange完成之後才能執行，如 Figure 2–17 所示。因此可視為Commit Phase的延長。

部分研究A. Mishra [16]、[17] 在IEEE 802.11i增強安全協定標準[3] 制定過程中，針對PMK架構與使用的設計提出網路端具有Proactive Key Distribution能力的安全協定設計，解決佈署加強安全協定機制後換手程序過長的問題。IEEE 802.11F [2] 也制定了相關的IAPP CACHE-notify機制提供Proactive PMK Key Distribution的功能。不過IEEE 802.11i增強安全協定標準[3] 最後底定標準，對於PMK預先佈署的做法，採用的是接下來介紹的 802.1X Pre-Authentication。

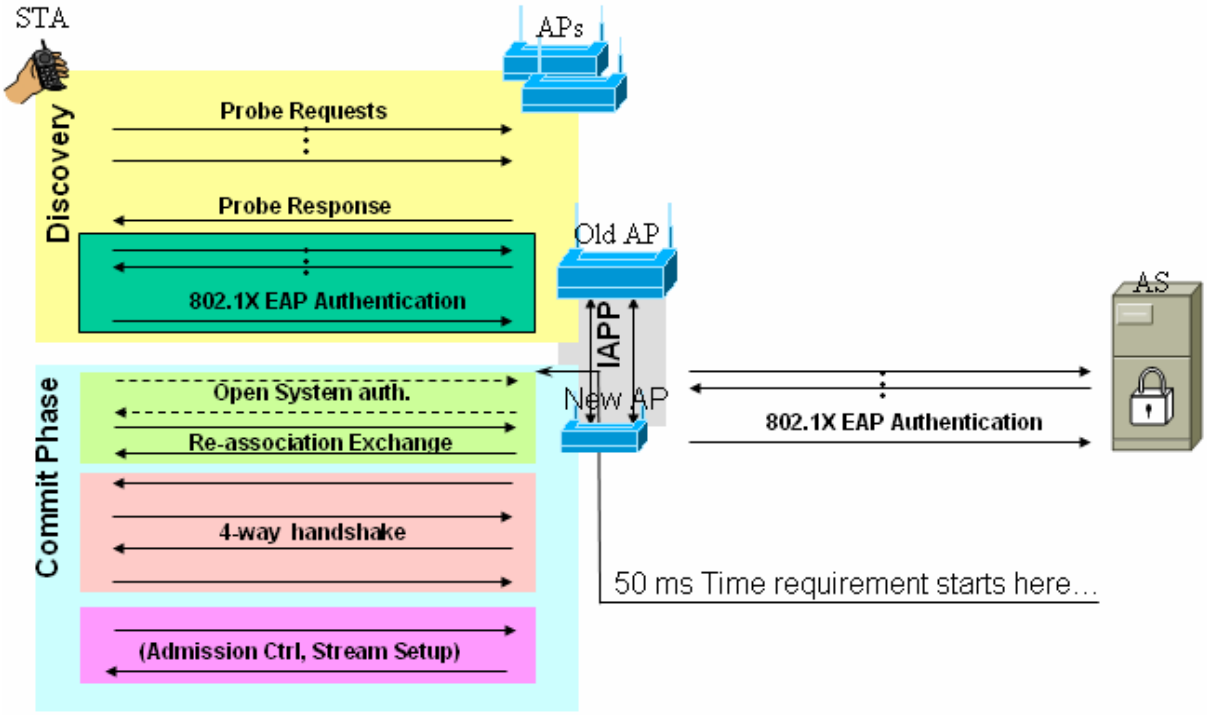


Figure 2–18 Handover messages exchange under 802.11i RSN using Pre-authentication

在接下來的換手程序中，若目標基地台沒有PMK可供驗證，IEEE 802.11i增強安全協定標準[3] 提供STA可使用Pre-Authentication的機制，如 Figure 2–18 所示，經服務基地台到目標基地台預先執行 802.1X的認證程序，將PMK預先佈署到目標基地台，然而 4-Way Handshake的驗證機制仍然必須在原始的Commit Phase完成之後執行。

若STA使用IEEE 802.11e服務品質保證規格標準[4] 所提供的QoS功能，也需要在連結上基地台且完成驗證之後才能執行。因此在具有IEEE 802.11i增強安全協定標準[3] 與IEEE 802.11e服務品質保證規格標準[4] 佈署的IEEE 802.11 無線網路，會進一步增加Commit Phase的長度，使得換手延遲增長。

2.6 背景知識與相關研究小結

影響 STA 在無線網路下的換手過程優劣的主要因素有二部分：第一是「換手時機的決策」、第二是「換手程序的長度」。

第一點「換手時機的決策」機制，是由「目標基地台搜索與量測機制」和以根據前者獲得的目標基地台量測資訊、及相關換手程序資訊為輸入的「換手決策演算法(Handover Algorithm)」所組成。

「換手決策演算法」中以目標基地台量測資訊為主要演算核心的部分，則以 2.2 中的Relative Signal Strength with Hysteresis and Threshold最著名，在本論文中以Threshold-Based Handover Algorithm稱呼，其效果在STA具有持續性的目標基地台量測數據下會更加顯著。

2.3 提到IEEE 802.11 無線網路目前普遍實作的換手模式是採取在執行換手程序前才進行「目標基地台搜索與量測」。這種換手模式對於無快速換手需求的應用是可行的，其優點是不需要特別的基地台管理機制，並且「目標基地台搜索與量測」的結果是最新也最符合換手環境現狀的。

Table 2–3 Useful target APs information

目標基地台資訊類別	章節	可改善參數
回應時間底限資訊	2.4.1.1	MinChannelTime、MaxChannelTime
工作的 channels 資訊	2.4.1.2	Number of channels
基地台相鄰關係資訊	2.4.1.3	MaxChannelTime、Number of channels
基地台 Non-Overlapping 關係資訊	2.4.1.4	MaxChannelTime、Number of channels

2.4.1 整理了一些以不更改「IEEE 802.11 無線網路目前普遍實作的換手模式」為前提、僅針對其Discovery Phase內「目標基地台搜索與量測」方式進行改進的相關研究。「額外目標基地台的資訊」是縮減active scan function執行時間的關鍵。有用的「額外目標基地台的資訊」可以是：目標基地台回應時間的底限資訊(2.4.1.1)、基地台工作的channels資訊(2.4.1.2)、基地台相鄰關係資訊(2.4.1.3)、基地台Non-Overlapping關係資訊 (2.4.1.4) 等，都可以協助active scan function對於MinChannelTime、MaxChannelTime及目標channels的選擇決策。我們將這些目標基地台資訊類別及其可改善active scan function的參數，整理如 Table 2–3 所示。

2.4.2 收集了一些ad hoc的方法，更改「IEEE 802.11 無線網路目前普遍實作的換手模式」，將Discovery Phase從換手過程中移除，即重新設計「目標基地台搜索與量測」機制，使得整個換手過程只有執行commit operations的Commit Phase。這些方法有概念性質的 Background Probing(2.4.2.1)、利用Caching技巧在STA端自行建立相鄰基地台資訊的 Selective Scanning and Caching(2.4.2.2)、融合Background Probing及Selective Scanning and Caching 的 Pre-Scanning and Dynamic Caching(2.4.2.3)及能夠提供持續性目標基地台搜索與量測功能的 SyncScan (2.4.2.4)。

值得注意的是，IEEE針對無線電通訊通道資源量測 (Radio Resource Measurement)這項議題，正草擬IEEE 802.11k無線電通訊通道資源量測標準[5]，這個正在制定中的量測標準可提供各式各樣有關通道資源量測與資訊傳達的機制。這些通道資源的量測資訊，除了可以幫助無線通訊進行傳輸能量管理 (Transmission Power Control, TPC)、傳輸通道選擇 (Dynamic Frequency Selection, DFS) 之外，其中Neighbor Report機制、和相關的量測服務都有助於「目標基地台搜索與量測」和「換手時機決策」的進行。

第二點「換手程序的長短」則跟「換手程序標準」有直接關連。換手程序中 Commit Phase 進行的 messages exchange 主要牽涉到：

1. 換手前後基地台、DS、與 STA 內部 Radio Link 狀態的改變驅動。
2. 安全協定相關 Context 的初始與佈署。
3. 服務品質保證協定相關 Context 的初始與佈署。

IEEE目前針對快速換手需求的應用服務，正草擬制定IEEE 802.11r基地台快速換手程序標準[6] 以減少Commit Phase所需花費的時間。此基地台快速換手程序可以有效的解決IEEE 802.11 無線網路在加強型安全協定與服務品質保證協定部署後，換手程序在各類協定間使用的messages exchanges不夠最佳化所造成的延遲問題。

根據IEEE 802.11r基地台快速換手程序標準[6] 所提供的Over the DS模式下的快速換手程序，可以讓Commit Phase (參考Figure 2–5、Figure 2–17、及Figure 2–18)只剩下(Re-)association request/response這兩個必備的、會更動radio link 狀態的messages exchange。本篇論文整理了IEEE 802.11r基地台快速換手程序標準[6] 所提供的相關快速換手程序，以 **附錄A** 的方式呈現，供讀者參考。

第三章　無縫換手機制之設計與架構

我們在前一章的 2.6 談到：影響STA在無線網路下換手過程優劣的主要因素有二部分：第一是**「換手時機的決策」**、第二是**「換手程序的長短」**。

而**「換手時機的決策」**機制，是由**「目標基地台搜索與量測機制」**與以根據前者獲得的目標基地台量測資訊、相關換手程序資訊為輸入的**「換手決策演算法 (Handover Algorithm)」**所組成。仔細思考「換手時機的決策」機制的運作，我們可以發現到「換手決策演算法」的選擇與設計，也會跟「目標基地台搜索與量測機制」所能提供的目標基地台量測資訊的程度高度相關。更進一步思考，換手時機的決策機制，也必須提供「換手程序」中某些預先佈署動作所需的事件驅動。

由上述討論不難理解：**「目標基地台搜索與量測機制」**是整個換手機制設計的核心。「換手決策演算法」及相關「換手程序」動作都會根據「目標基地台搜索與量測機制」進行事件與狀態驅動。因此我們必須設計一套適合 IEEE 802.11 無線網路使用的「目標基地台搜索與量測機制」，以便接下來整個無縫換手機制的設計。

3.1 目標基地台搜索與量測機制使用的內部機制與設計原理

我們設計的「目標基地台搜索與量測」機制，可分成「目標基地台搜索機制」與「目標基地台量測機制」。

其中**「目標基地台搜索機制」**會用到「Further Observed Scanning搜索機制」、「Packet Loss Prevention Mechanism」及「AP-specific Probe Facility」；而**「目標基地台量測機制」**會使用到「Non-AP-Aligned SyncScan機制」、「Packet Loss Prevention Mechanism」及「AP-specific Probe Facility」。其原理與運作介紹如下：

3.1.1 Non-AP-Aligned SyncScan

為了達到提供STA足夠的目標基地台資訊以供更好的換手決策所需的目標，我們設計的「目標基地台搜索與量測機制」必須能夠提供持續性目標基地台量測的能力。回顧2.4.2.4 SyncScan 的核心精神：如果STA可以知道目標基地台發送beacon的時間，STA便可以直接使用passive scan function，以背動等待目標基地台發送beacon的模式，對目標基地台進行量測。

雖然I. Ramani and S. Savage [13] 所提的AP-Aligned SyncScan：「設計一套依據channel number安排好的週期性基地台beacon發送計劃時刻表，讓每個基地台都按照這個beacon發送計劃時刻發送beacon。」可以讓STA獲得所有目標基地台發送beacon的時間資訊，但這個方法的前提是所有的基地台都要按表操課，這對世界上已經佈署的眾多基地台，似乎不是一件容易的事。

於是現在的問題變成：「除了使用一套依據 channel number 安排好的週期性基地台 beacon 發送計劃時刻表之外，還有沒有其它方式，可以讓 STA 獲得目標基地台發送 beacon 的時間？」

答案是肯定的！！

根據IEEE 802.11 標準[1] 11.1 的規定，同一BSS底下的所有STAs需要做timing synchronization的動作。而在Infrucstrature Mode下的BSS是由基地台採取主導的地位。基地台透過每隔Beacon Interval的時間發送Beacon，在Beacon中提供基地台使用的相關 timing information 供其它 STAs 做 synchronization 之用。除此之外，Probe Response同樣也帶有這些相關基地台的timing information。

IEEE 802.11 標準[1] 11.1.2.1 也規定了基地台beacon發送的時間點，稱為TBTTs (Target Beacon Transmit Time)。基地台除了每隔Beacon Interval發送一次Beacon之外，第一次發送的時間是從基地台內部microsecond resolution的clock為 0 的那一刻開始。由於發送Beacon Frame的機制仍然是使用DCF，所以基地台有可以無法在規定的TBTT傳送出Beacon，而是在稍微delay的時間送出，如 Figure 3–1 所示。

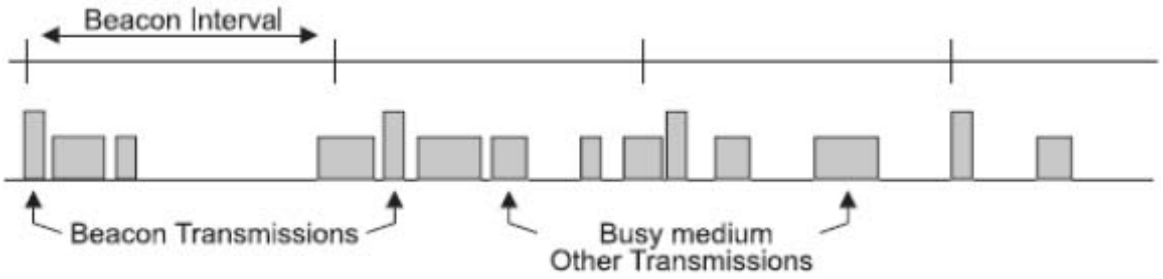


Figure 3–1 Beacon Transmission on a busy network

(Source from [1])

Beacon frame 與 Probe Response frame 這兩種 management frame 具有類似的結構，並且攜帶許多基地台運作的相關資訊。從這些攜帶的資訊當中，經由 Timestamp 及 Beacon Interval 欄位的資訊，我們就可以推得基地台發送 beacon 的時序。

Timestamp 欄位記載的是基地台發送 Beacon frame 或 Probe Response frame 那一瞬間，內部以 microsecond 為解析度的 clock 值，Timestamp 欄位長度為 8 bytes。而 Beacon Interval 欄位記載是以 TU (Time Unit) 為單位的 beacon 發送周期，且 1 TU = 1.024 ms = 1024 μs。STA 從 Beacon frame 與 Probe Response frame 內的 Timestamp 及 Beacon Interval 資訊，推算基地台發送 beacon 時序的方法如下：

(a)Beacon Frame：

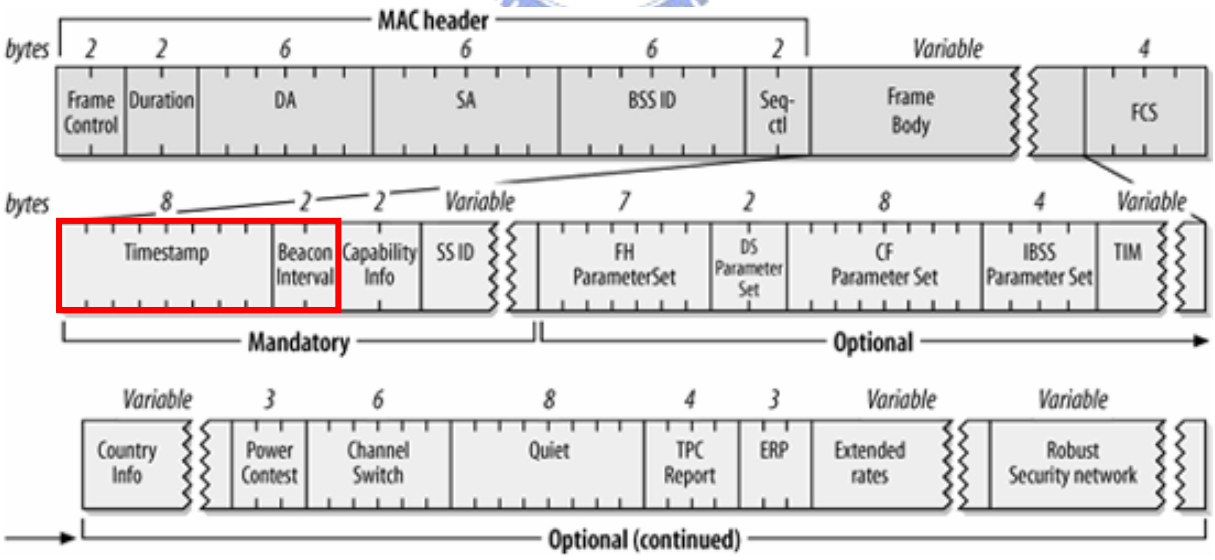


Figure 3–2 Beacon frame

(Source from [22])

如果 STA 在以 millisecond 為單位的時間點 t_0 收到一個基地台發送的 Beacon frame，其中在 Beacon frame 內的 Timestamp 欄位的值為 **C**，而 Beacon Interval 欄位的值為 **T**。則基地台會在 STA 的時間點

$$t = t_0 - (C \% (T*1024))/1000 + n * T * 1.024 \quad n = 1, 2, 3 \ldots$$

發送 beacon。

當發送的 Beacon 是在正常的狀況（例如：沒有延遲或碰撞後重送），

$$(C \% (T*1024))/1000$$

這一項會等於 0，即基地台在 STA 的時間點

$$t = t_0 + n * T * 1.024 \quad n = 1, 2, 3 \ldots$$

發送 beacon。

(b) Probe Response Frame：

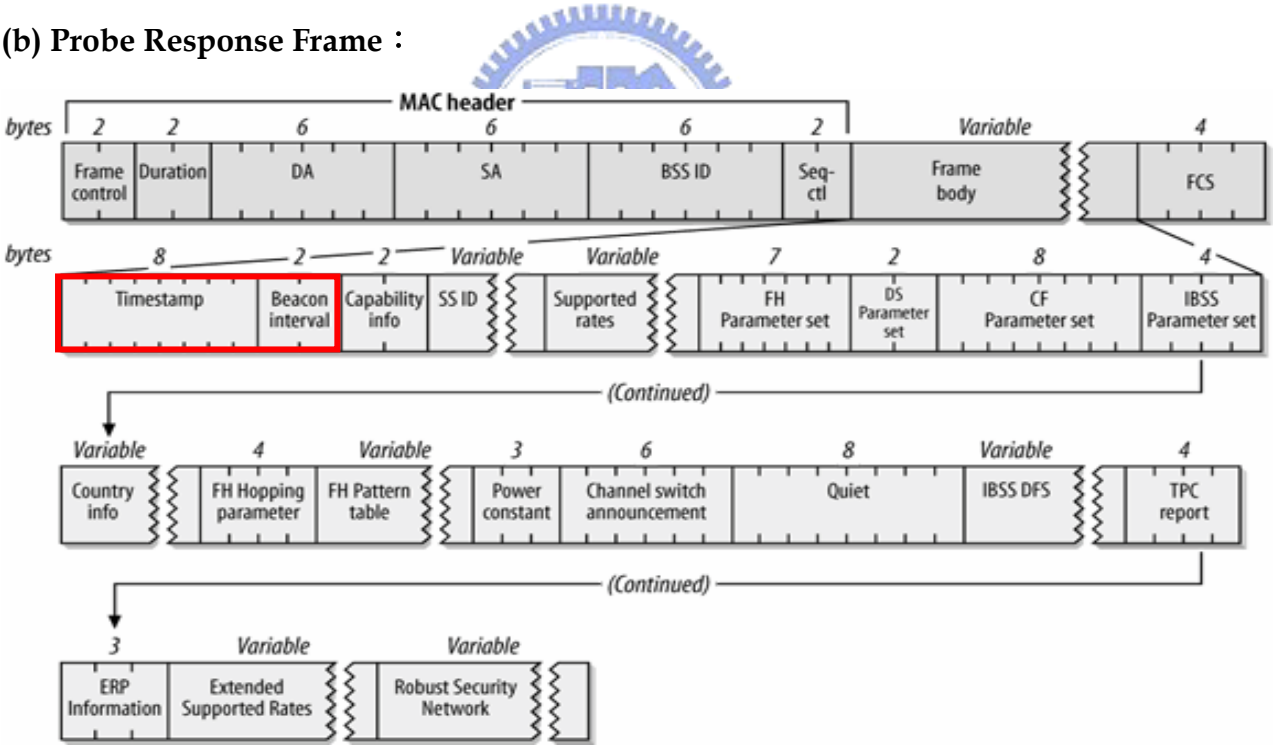


Figure 3–3 Probe Response frame

(Source from [22])

如果 STA 在以 millisecond 為單位的時間點 t_0 收到一個基地台發送的 Probe Response frame，其中在 Probe Response frame 內的 Timestamp 欄位的值為 **C**，而 Beacon Interval 欄位的值為 **T**。則基地台會在 STA 的時間點

$$t = t_0 - (C \% (T*1024))/1000 + n * T * 1.024 \qquad n = 1, 2, 3 \ldots$$

發送 beacon。

因此，在STA第一次收到目標基地台發送的Beacon Frame或是Probe Response frame之後，STA就擁有該目標基地台發送beacon的時間表。STA就可以根據自己的需求，在適當的時間點對該目標基地台執行 SyncScan，進行目標基地台的量測動作。利用這種方法進行的SyncScan有以下好處：

1. 基地台不需要使用統一的 beacon 發送時刻表。
2. 基地台不需要進行修改。
3. 僅需對 STA 進行修改。
4. STA 進行目標基地台量測時，除了第一次之外，因為使用 passive scan function，不會造成額外的網路流量。

本篇論文稱呼這種 SyncScan 模式為 Non-AP-Aligned SyncScan，這也是我們「目標基地台量測機制」的量測機制的核心。

3.1.2 AP-specific Probe Facility

MLME-SCAN.request	(BSSType, BSSID, SSID, ScanType, ProbeDelay, ChannelList, MinChannelTime, MaxChannelTime)

Figure 3–4 MLME-SCAN.reqeust primitive

根據IEEE 802.11標準[1]的說明(參考Figure 2–6 SME invokes MLME-SCAN.request and receives MLME-SCAN.confirm)，在Station Management Entity (SME)透過Figure 2–16中的MLME-SAP呼叫如Figure 3–4中的MLME-SCAN.request() primitive之後，Figure 2–16中MAC Sublayer Management Entity會依據MLME-SCAN.request() primitive的參數，透過MAC Sublayer，在每一個ChannelList參數所列的channels發送Probe Request Frame。如果是呼叫active scan function，在每一個channel下，MAC都應該進行如Figure 2–7所示的active scan procedure，當所有的channels都完成active scan之後，MAC Sublayer Management Entity會將結果使用MLME-SCAN.confirm() primitive回傳給Station Management Entity。

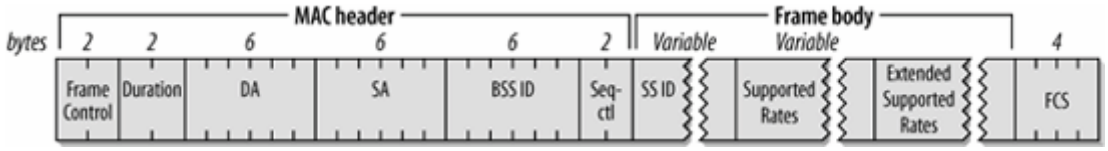


Figure 3–5 Probe Request frame

(Source from [22])

如Figure 3–5所示的Probe Request frame format，DA欄位是指接收者的MAC address，Probe Request frame通常使用broadcast address。而SA欄位是發送者的MAC address。BSSID欄位是BSS的代號，通常是基地台的MAC address，如果是broadcast address，則代表不限定接收的基地台。SSID欄位可供STA指定目標基地台搜索的無線網路代號，Probe Request frame通常使用Null SSID以表示不限定。

原始的active scan function如Figure 2–7所示，會使用broadcast DA、broadcast BSSID、不限定SSID的方式在目標channel下發送Probe Request frame。所有在目標channel下的基地台在接收到Probe Request frame之後，會用Unicast方式傳送Probe Response frame (如Figure 3–3所示)給STA。

STA在一個工作channel下發送一個broadcast Probe Request frame，可以獲得多個Probe Response frames，這就是最保守的「目標基地台搜索機制」。

active scan function還能怎麼用？還能拿來做什麼？

3.1.2.1 Specify ChannelList in MLME-SCAN.request primitive

Station Management Entity (SME)可以指定 MLME-SCAN.request() primitive ChannelList 的參數，使得 active scan function 只對指定的 channels 下的目標基地台進行量測。例如：STA 可以一次只指定單一 channel，將多個預選量測的目標基地台依運作 channel 進行分散且分成多次進行。MAC 每一次只會對該 channel 下的目標基地台進行量測，可以縮短因量測基地台所造成正常資料傳輸與接收暫時中斷的時間，許多研究都沒有特別考量這一項因素。

3.1.2.2 Specify SSID in Probe Request Frame

觀察MLME-SCAN.request() primitive提供的介面，STA可以指定特定的SSID；而IEEE 802.11 標準[1] 也規定基地台必須收到Probe Request frame內的SSID是「不指定特別SSID」或是「指定的SSID就是基地台本身使用的SSID」的情況下才能夠回覆Probe Response frame。因此，若STA只想對本身使用的ESS下的基地台進行量測，可以指定active scan function使用的SSID，使得不相干的基地台不會回覆Probe Response frame，節省無線傳輸的資源及STA等待結果的時間。

3.1.2.3 Specify BSSID in Probe Request Frame

再進一步觀察MLME-SCAN.request() primitive提供的介面，STA也可以指定特定的BSSID；而IEEE 802.11 標準[1] 也規定基地台必須收到Probe Request frame內的BSSID是「broadcast BSSID」或是「BSSID欄位就是基地台本身使用的BSSID」才能夠回覆Probe Response frame。因此，若STA只想對特定的基地台進行量測，可以指定active scan function使用的BSSID，使得其它基地台不會回覆Probe Response frame，節省無線傳輸的資訊，並且STA的MAC在接收到量測目標基地台回覆的Probe Response frame後，因為不會有兩個基地台同時具備相同的BSSID，即可提前結束等待其它的Probe Response frame。

3.1.2.4 Unicast Probe Request

若STA已知目標基地台的相關資訊：工作channel、BSSID、SSID的情況下要進行目標基地台的量測動作，3.1.2.1－3.1.2.3說明了STA可以透過MLME-SCAN.request() primitive相關參數的指定，達到僅由STA的MAC傳送「指定SSID、BSSID的broadcast Probe Request frame」，接著等待目標基地台回傳單一的Probe Response frame即可。藉由Probe Response frame的接收，STA的PHY可以提供接收該Probe Response frame的訊號強度，達到所謂的「目標基地台量測[16]」。

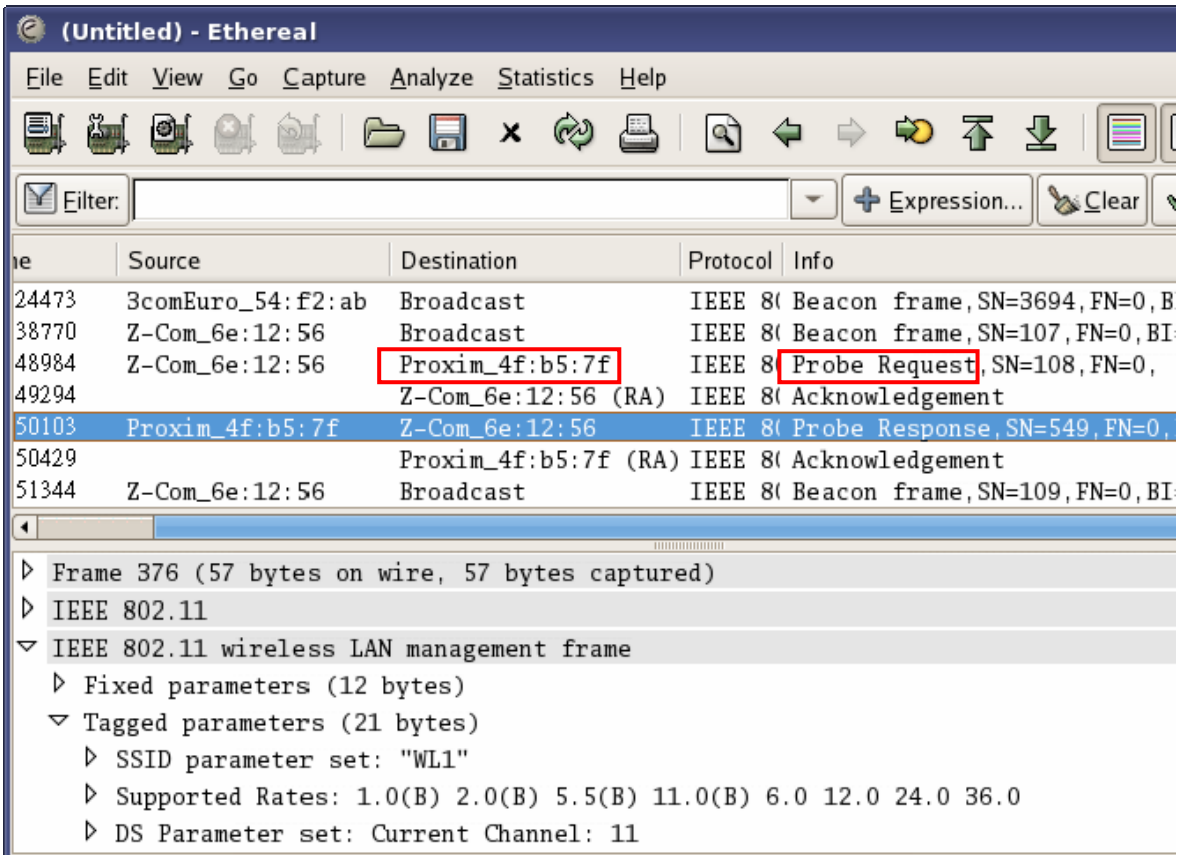


Figure 3–6 Captured unicast Probe Request/Response

[16]目標基地台量測是要測量基地台發送的 frame 在 STA 端的 SNR 的值，在 active scan 進行時，STA 量測是目標基地台送給 STA 的 unicast Probe Response frame；在 passive scan 進行時，STA 量測的是目標基地台發送的 broadcast Beacon frame。事實上，STA 只要量測到目標基地台發送的 frame 即可。由於 STA 並不知道目標基地台什麼時候會發送 frame，因此如果要背動量測，我們通常選擇基地台固定會發送的 Beacon 來做量測，在 TBTT 的時間切換到目標基地台運作的 channel 等待目標基地台的 frames 傳送，有很大的可能可以聽到 Beacon frame。

試問此指定SSID、BSSID的Probe Request frame是否可以利用unicast機制傳送？據筆者對IEEE 802.11 標準[1] 所制定MAC運作機制的理解，unicast/broadcast frame依據DCF運作的傳送機制和frame傳送的內容兩者是分開的，沒有直接關連。而STA進行所謂的量測過程，也只是藉由接收目標基地台傳送的frame時，由STA的PHY提供的接收frame所量測到的訊號強度資訊。

因此，只要STA的MAC將Probe Request frame的DA欄位改成目標基地台的MAC address，則此Probe Request frame即為unicast Probe Request，MAC需要依照DCF下unicast frame傳送的機制進行傳送和接收，在SIFS時間後回應Ack。根據我們的實驗結果，unicast Probe Request確實可正確運作，見 Figure 3–6 由sniff所攫取到的unicast Probe Request frame示意圖。

那麼在 STA 擁有目標基地台的相關資訊後，使用 unicast Probe Request 可以有什麼好處呢？

就傳送 Probe Request frame 本身而言是沒有多大的差別，目標基地台都可以接收到該 Probe Request frame，但是因為 unicast Probe Request 是使用 unicast 傳送機制，因此目標基地台必須使用 Positive Ack 回應 STA 傳送成功。若 STA 只是要單純進行目標基地台的量測，可以藉由目標基地台回應的 Ack 的接收機會，經由 STA 的 PHY 得到傳送 Ack 的訊號強度資訊即可。不需要再等待接收 Probe Response frame。此外 STA 的 MAC 也可以藉由 Ack 正確接收與否，判斷目標基地是否存在。

不過上述機制還有一些需要考量的議題：

1. STA 的 MAC 需要有 PHY 提供的 primitive，以獲得接收到的 Ack Frame 時的訊號強度資訊，接下來才能提供 Station Management Entity (SME) 量測到的強度資訊。

2. 目標基地台仍然會回傳 Probe Response frame，因此 Probe Request frame 需要有指定目標基地台不回應 Probe Response frame 的識別，否則 STA 提前離開工作 channel，會造成目標基地台傳送 Probe Response frame 的失敗，引發目標基地台接下來的嘗試重送，這會浪費目標 channel 的無線通訊資源。

我們設計的「目標基地台量測機制」，只有在 STA 使用 Non-AP-Aligned SyncScan 但 miss 掉目標基地台發送的 beacon，又必須量測到目標基地台的條件下或者在已知特定 SSID、BSSID、及目標基地台運作的 channel 時，才會使用 unicast Probe Request，對目標基地台進行「主動量測」。

```
124288-399327: prism2_ioctl_siwscan: enter
124289-399327: my_prism2_request_hostscan: ssid=<NULL>, channels=9
124312-399384: MGMT:PROBE_RESP: ch10 00:0E:A6:39:58:BF WIN (1/8)
124314-399385: MGMT:PROBE_RESP: ch11 00:60:B3:13:03:5D toy (2/8)
124316-399386: MGMT:PROBE_RESP: ch 9 00:20:A6:4F:B5:7F WL1 (3/8)
124329-399413: prism2_ioctl_giwscan_sta: ready to leave
```

Figure 3–7 Output Logging messages from our experiment

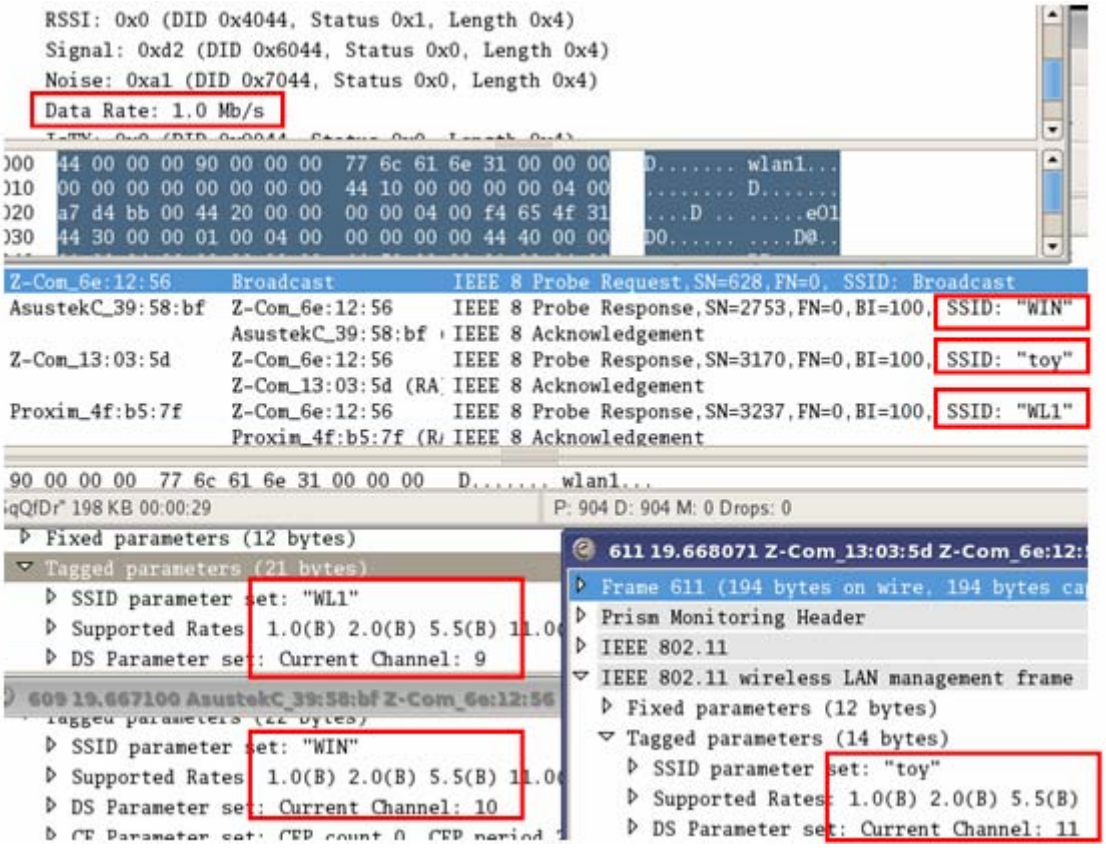


Figure 3–8 Captured Probe Response frames in overlapping channels environment

The Probe Request frame was sent in channel #9

3.1.3 Further Observed Scanning in Overlapping Channels Environment

我們利用實際的硬體設備，在channel #9 操作發送broadcast Probe Request，在802.11b運作的 2.4GHz overlapping channels的環境下，我們從硬體設備 (如Figure 3–7) 與sniff的偵測 (如Figure 3–8) 所接收到的Probe Response frames結果發現一個現象，那就是STA可以接收到運作在channel #9, #10, #11 基地台所回應發送出來的Probe Response frames。

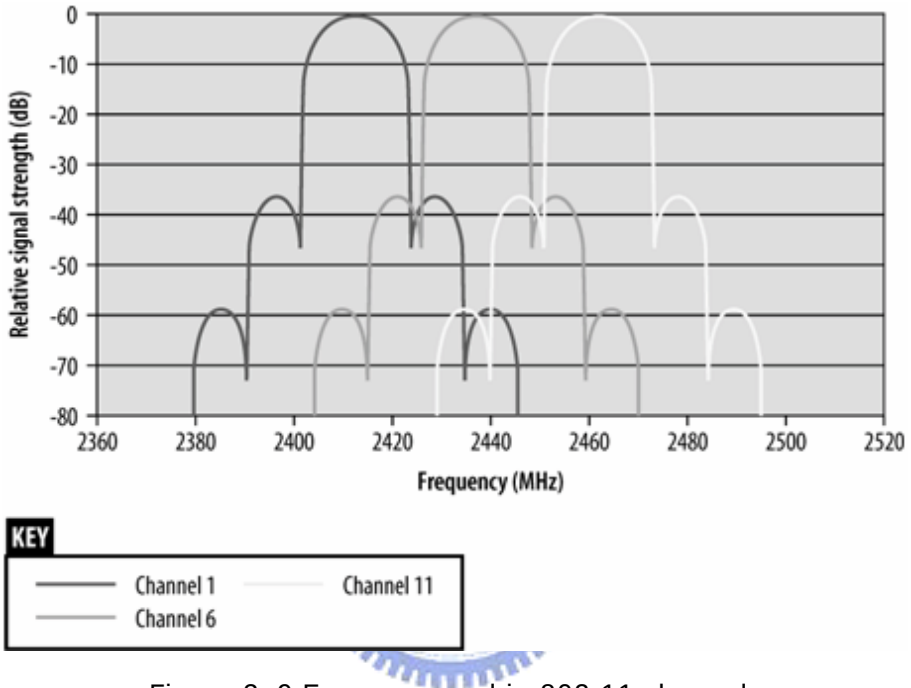


Figure 3–9 Energy spread in 802.11 channels

(Source from [22])

這個現象發生的原因主要是因為：在802.11b使用的2.4 GHz下規定的11個channels之中，只有間隔 5 個channel寬度的#1, #6, #11 符合Non-overlapping的條件，而channel #10, #11 因為離channel #9 太近，所以會有互相干擾的情況產生。Figure 3–9 是STA的PHY使用 802.11b DSSS調變技術傳送frame時，在周圍相鄰頻率下能量的分佈圖。

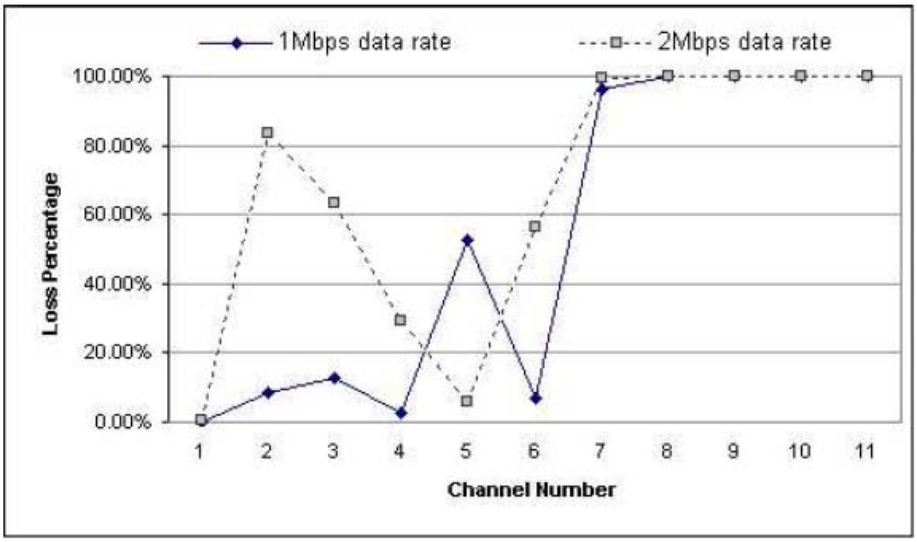


Figure 3–10 Loss percentage of frames in neighboring channels.

The traffic was sent on channel 1. (Source from [7])

我們在A. Mishra等[7] 的實驗量測報告中，也發現相同的現象的報告。A. Mishra等[7] 為了觀察市售無線網路卡換手的動作，需要使用Sniff記錄STA完整的換手過程，但是A. Mishra等[7] 只使用了運作在channel #1, #6, #11 下的三個sniff即可將 11 個channel下，大部分的Probe Request/Response的frames全部攫取到，其原因也是利用overlapping channel的特性。

Figure 3–10 是A. Mishra等[7] 研究此特性所做的實驗結果。他們在channel #1 分別以 1 Mbps、及 2 Mbps傳輸速度用的調變方法持續傳送frames，接著在其它channels放置sniff攫取這些frames，Figure 3–10 表示了在其它channels下攫取不到、遺失frames的百分比。我們可以發現到，在相鄰 3 個channels以內的Overlapping channel，以 1 Mbps傳輸速度用的調變方法傳送的frames，有高達 90%以上的frames可以被正確傳送與接收。

回到之前 Probe Request/Response 的討論，channel #10, #11 的基地台因為能收到在 channel #9 發送的 Probe Request frame，因此回應了 Probe Response frames，而運作在 channel #9 下的 STA 也能正確接收到在 channel #10, #11 基地台發送的 Probe Response frames。

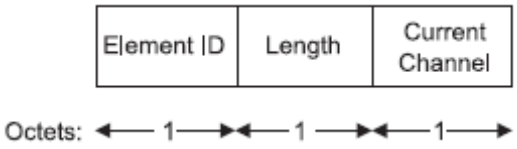


Figure 3–11 DS Parameter Set IE

(Source from [1])

我們在IEEE 802.11k無線電通訊通道資源量測標準[5] 已經看到了對於這個現象的相關修正。新修正的標準將Probe Request frame(Figure 3–5)，新增一項原本只出現在Beacon frame(Figure 3–2)及Probe Response frame(Figure 3–3)下如 Figure 3–11 所示的DS Parameter Set IE。這個Information Element記載了使用DSSS調變方法其運作的channel number。STA可以經由Beacon/Probe Response frame中的DS Parameter Set IE識別目標基地台運作的channel number，而目標基地台也可以經由Probe Request frame中的DS Parameter Set IE識別發送STA運作的channel number。

新修正的標準沒有硬性規定 STA 發送的 Probe Request frame 一定要攜帶 DS Parameter Set IE，STA 可以視需求自由選擇攜帶與否；但標準本身規定目標基地台在收到帶有 DS Parameter Set IE 的 Probe Request frame 時，本身必須也是運作在相同 channel 下才能回應 Probe Response frame。

因此，如果 STA 需要對運作在 2.4 GHz 11 個 channels 下的目標基地台進行搜索，只需要先針對 channel #1, #6, #11 執行發送不攜帶 DS Parameter Set 的 Probe Request frame 的 active scan function，就可以掌握大部分基地台的資訊，若需要正確的基地台量測數據，STA 可以再執行第二輪在有切換到正確運作的 channel 下、對目標基地台所做的個別量測。

我們可以利用overlapping channel下產生的特性[17]，使用上述的程序，有效地搜索未知的目標基地台。在本論文稱使用上述程序進行目標基地台搜索的模式為 Further Observed Scanning。

3.1.4 Packet Loss Prevention Mechanism

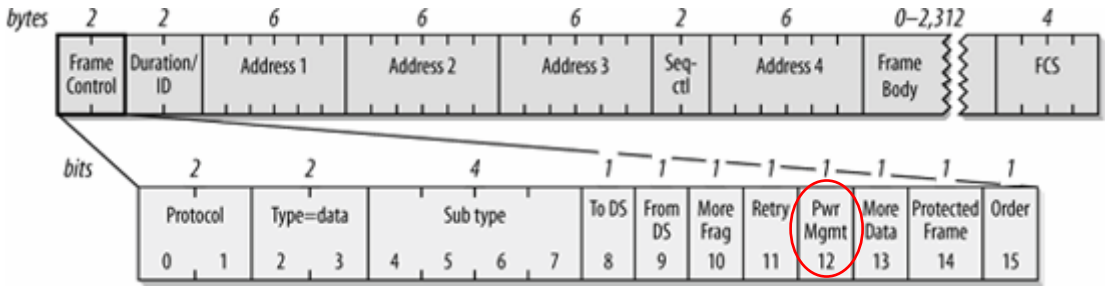


Figure 3–12 Power management bit in Frame Control field

(Source from [22])

當 STA 執行「目標基地台搜索與量測」期間，STA 必須切換到目標 channels 進行相關 frames 的傳送與接收，將無法正常傳送與接收來自狀態上仍屬連結的基地台的 frames。這段期間除了 STA 不能傳送正常的 data frames 到連結基地台之外，若連結基地台有 frames 要傳送給 STA，將造成無人回應 Ack 的傳送失敗。

所幸，IEEE 802.11 標準[1] 提供的Power Save mechanism可以幫得上忙，STA有管道通知基地台目前它的power management狀態。STA的power management狀態傳送，是使用frame header中、Frame Control欄位下的Power Management Bit (Figure 3–12)識別。

[17] 這項 Overlapping Channel 的特性只對使用 1 Mbps 傳送速率的調變方法有較高的可靠度，Probe Request/Response Frame 都必須使用 1Mbps 傳送速率的調變方法傳送。所幸 IEEE 802.11 標準[1] 有規定，基地台傳送 Beacon Frame 及回應 Probe Response Frame 使用的調變方法，要以最多 STA 或目標 STA 能正確接收為最高原則。基地台通常會使用 1 Mbps 傳送速率的調變方法傳送 Beacon Frame；並且針對 Probe Response Frame 使用的調變方法，會依 STA 所送 Probe Request 使用的調變方法傳送。

Power Management Bit set (Figure 3–13) 代表STA即將進入Power Save模式，基地台應該幫STA buffer送給STA的frames一段時間；而Power Management Bit clear (Figure 3–14) 表示STA沒有處在Power Save模式，基地台必須對STA的狀態進行更新。Figure 3–13、Figure 3–14 中的Data frame雖然沒有frame body，但並不表示只能使用這類subtype為null的data frame；相反的，這類subtype為null的data frame的出現，是為了確保即使STA沒有任何資訊可以傳輸，仍然可以將Power Save狀態通報給基地台。

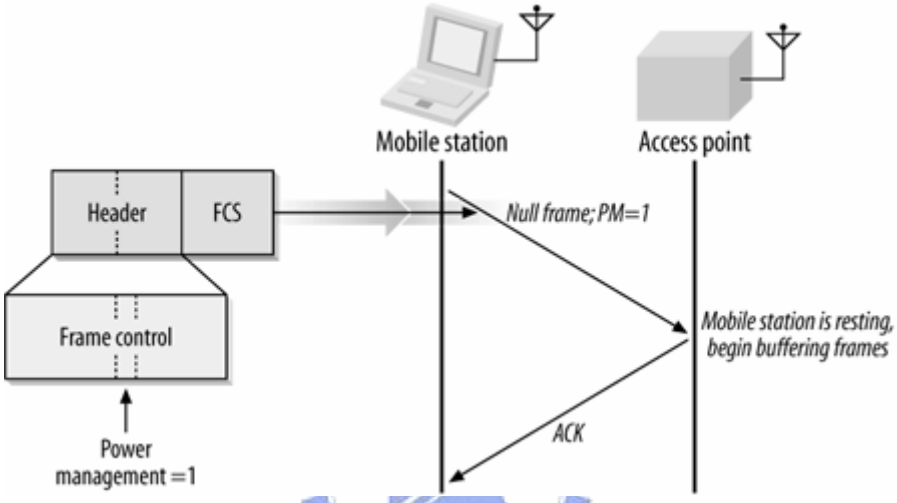


Figure 3–13 Data frame of subtype Null with PwrMgmt Bit set

(Source from [22])

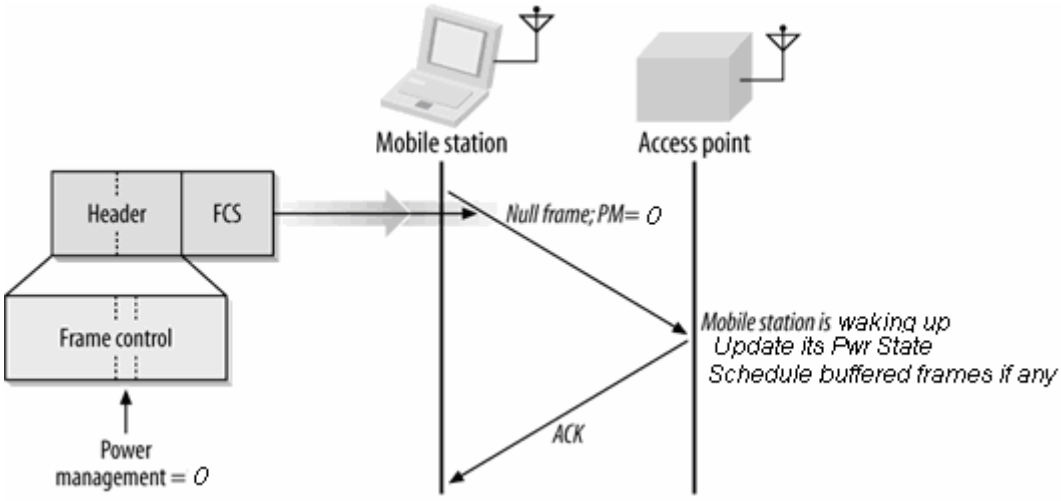


Figure 3–14 Data frame of subtype Null with PwrMgmt Bit clear

(Adapted from [22])

IEEE 802.11 標準[1] 另外也提供了一項STA從Power Save狀態回覆正常狀態對連結基地台查詢並請求傳送buffered frames的機制。STA可以使用subtype為PS-Poll的control frame向連結基地台查詢，基地台在收到PS-Poll frame之後，可以立即回覆buffered frames (Figure 3–15) 或是延緩傳送buffered frames，僅以Ack回應STA的查詢 (Figure 3–16)。

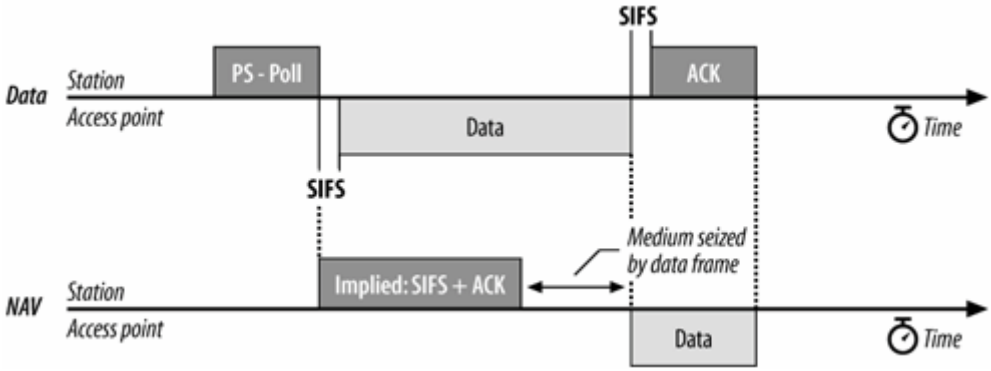


Figure 3–15 Immediate PS-Poll response

(Source from [22])

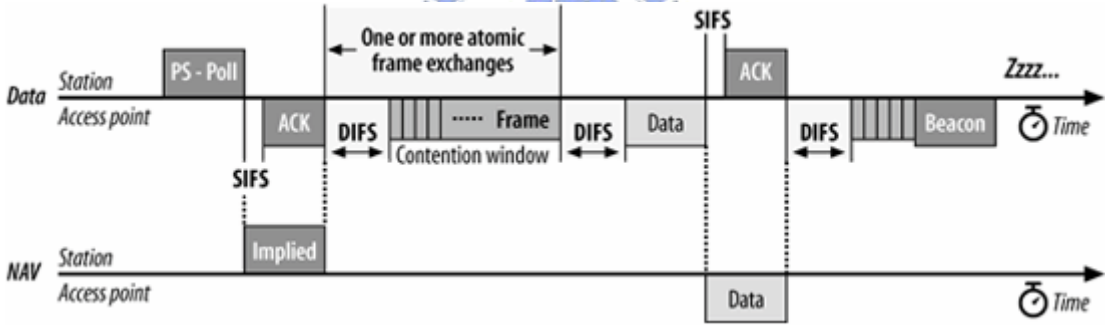


Figure 3–16 Deferred PS-Poll response

(Source from [22])

雖然 power management 機制的主要目的是為了電源使用的考量，STA 適當地關掉無線收發器可以延長電池的使用時間。然而，利用這個 power management 機制下基地台提供的 buffering 功能，可以避免 STA 在進行「目標基地台搜索與量測」期間可能造成的封包遺失情況。

3.2 目標基地台搜索與量測機制的運作

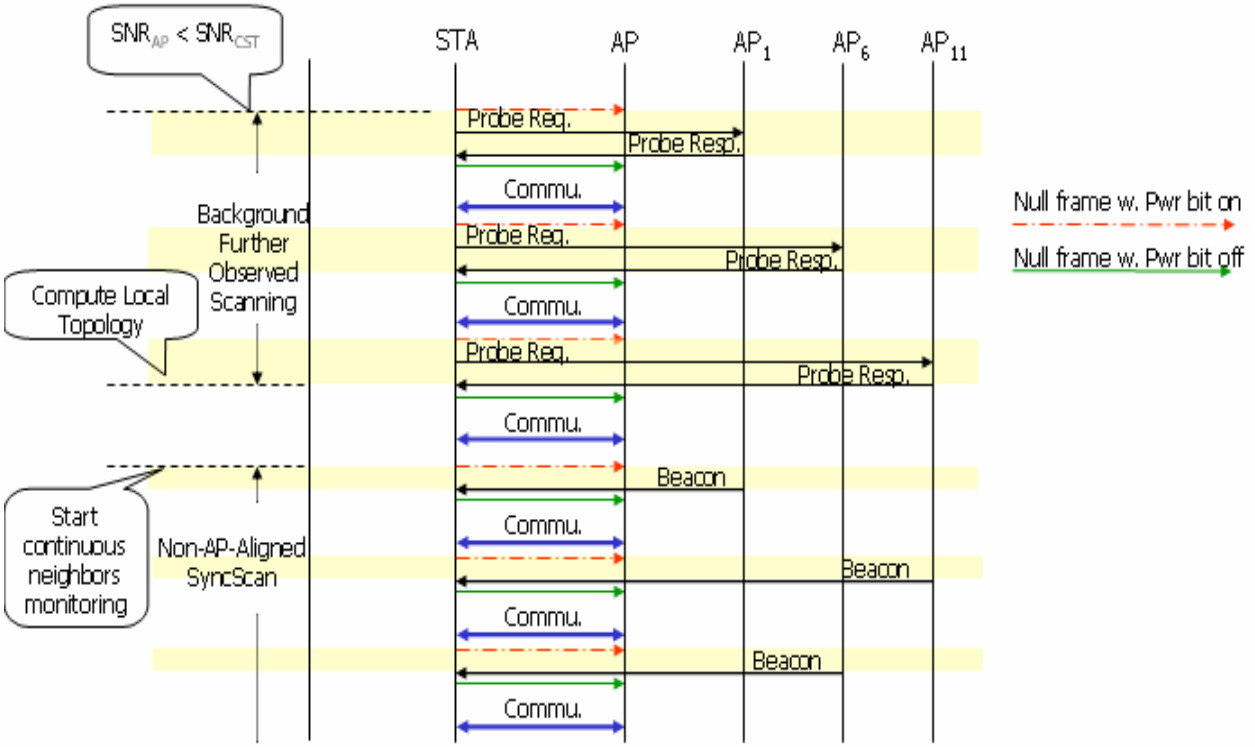


Figure 3–17 AP Discovery and Measurement example without NG support

STA在執行換手程序commit operations前才執行完整一連串的「目標基地台搜索與量測」動作，如 2.3 所提及的目前普遍實作的換手程序，因「目標基地台搜索與量測」程序過長，不適合需要快速換手的應用；而周期性不間斷地進行「目標基地台搜索與量測」，如 2.4.2.4 所提及的SyncScan，雖然STA不會造成額外的網路流量，然而確需要暫停正常資料的傳送與接受，也是件不小的負擔。更重要的是，周期性不間斷地進行「目標基地台搜索與量測」，會造成省電機制所能達到的效果有限。

我們所設計的「目標基地台搜索與量測機制」採取折衷的方式，根據服務基地台服務的訊號強度，區分出針對於「目標基地台搜索與量測」層面，STA 所處的 Stop Cell Search 和 Regular Cell Search 兩種狀態。

當服務基地台的訊號強度大於 SNR_{CST} 時，STA 處於 Stop Cell Search 狀態，完全不進行目標基地台的搜索與量測；當服務基地台的訊號強度小於 SNR_{CST} 時，STA 進入 Regular Cell Search 狀態，也啟動目標基地台搜索與量測的程序。

首先，STA必須先進行目標基地台的搜索。在無法提供Neighbor Graph基地台資訊的服務基地台下，STA必須自己進行目標基地台的蒐集。我們採取的搜索機制是Background Further Observed Scanning。如 Figure 3–17 所示，STA僅對channel #1, #6, #11 進行Background Scanning。利用 Further Observed Scanning 的原理，STA可搜索到大部分的目標基地台的資訊。接著STA可以建構自己的目標基地台Local Topology結構，使用 Non-AP-Aligned SyncScan 機制，有效率地進行目標基地台量測動作。

要注意的是，使用Background Further Observed Scanning 及 Non-AP-Aligned SyncScan 並不會減少搜索量測動作本身需要的時間，倒是這種分散目標基地台量測的模式，避免掉因接連進行一連串搜索量測動作所會造成的明顯正常傳輸暫停的現象。

如果服務基地台有提供Neighbor Graph基地台資訊，如 Figure 3–18 所示，STA可以針對這些參考性質的資訊，使用Unicast Probe的方式進行偵測，並且將STA量測到的最新資訊提供給服務基地台參考，進行動態的維護工作。在這樣的網路環境下，STA只要根據網路端提供的基地台資訊進行偵測即可，不需要額外再進行目標基地台的搜索作動作，可以省下不少的資源，並且不會遺漏掉可能的目標基地台。

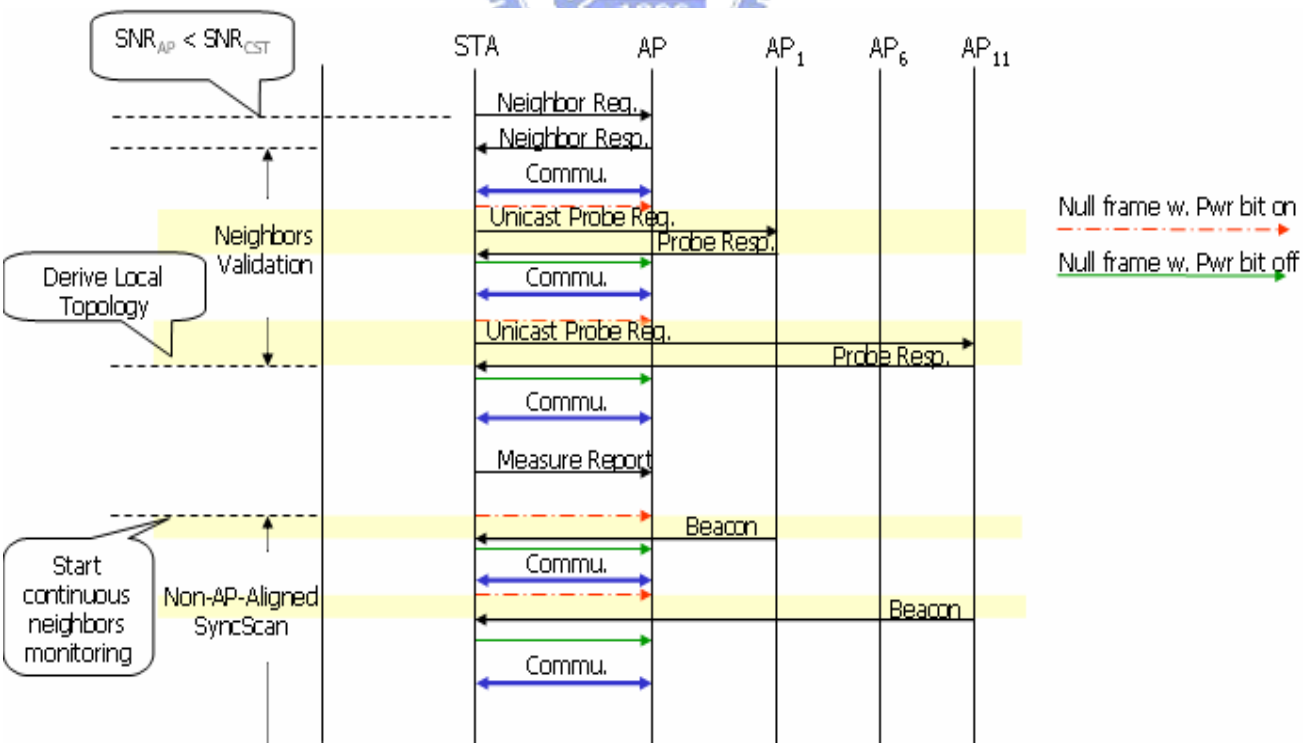


Figure 3–18 AP Discovery and Measurement example with NG support

3.3 無縫換手機制的整體運作

我們設計的無縫換手機制，是以「目標基地台搜索與量測機制」為運作核心骨架的換手機制。換手時機的相關決策是由換手決策演算法經相關目標基地台量測數據所進行演算的決定，接著執行符合 IEEE 802.11 標準相關規範的換手程序與適當的預先佈署程序。

首先，我們先看看「換手時機的決策」又可細分成那些子決策事項？

1. 是否有換手的需要？ (Determination of the need for a handover)

若服務基地台的訊號強度低於某 Threshold，例如 Cell Search Threshold，則 STA 可能會進行換手，即 STA 有換手的需求，但上述的條件並非一定會導致立即決策換手，還會考量鄰近基地台與服務基地台彼此間佈署的相對位置而定。其理由是：若服務基地台的訊號強度還大於目標基地台的訊號強度，立即進行換手顯然並不會獲得好處。

2. 換手的目標基地台為何？ (Selection of which AP to handover to)

如果 STA 有持續性目標基地台搜索與量測的機制，在最終決策換手、執行 commit operations 前的目標基地台可能還只是一基地台的集合。不過，最終目標基地台在 STA 執行 commit operations 前一刻一定會確定。

3. 什麼時間點進行換手程序？ (Determination of when to handover)

明確講就是什麼時間點執行 commit operations。這會由「換手決策演算法」根據持續性的目標基地台量測數據經演算決定。

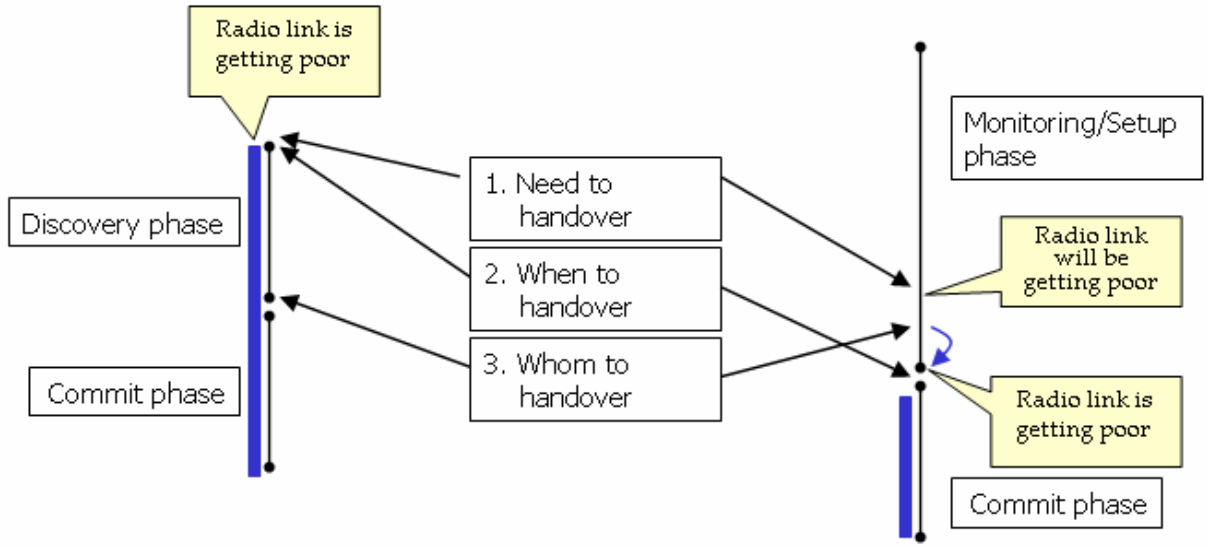


Figure 3–19 Transition decision timeline of 2 possible determination behaviors

Figure 3–19 顯示了普遍實作的換手決策過程和我們採取的換手決策過程的差異，其中在旁邊的藍色長柱表示換手過程所花的時間。左邊是普遍實作的換手決策過程(2.3)，當服務基地台無法再提供可接受的傳送品質質，STA才完成 1. 3.項的決策，接著進行一完整的目標基地台搜索與量測的動作，最後執行commit operations進行換手。右邊則是我們實作的換手機制在整個換手決策的時間進程，當服務基地台的訊號強度低於某臨界值時，STA即完成 1.項的決策，找出未來可能換手的Local Topology，並且開始持續地進行目標基地台的量測動作，在這個量測的過程中，根據換手決策演算法接著會完成 3.及 2.項的決策。由 Figure 3–19 可看出我們設計的換手機制，換手過程除去了Discovery Phase，能有效地縮短換手延遲時間。

基於我們所採用的「目標基地台搜索與量測」的運作，「換手決策機制」演算法根據目前服務基地台的訊號強度範圍，將換手過程分成主要三個 stages，說明如後：

1. Neighbors Gathering Stage (Discovery)[18]
2. Neighbor Monitoring Stage (Resource Establish)
3. Commit Stage (Transition)

[18] 括號內的名稱，是對應 IEEE 802.11r 基地台快速換手程序標準[6] 針對 BSS Transition 定義的 Stages。參考 1.2 的說明。

3.3.1 Neighbors Gathering Stage (Discovery)

當目前服務基地台的訊號強度低於SNR_{CST}時，進入 Figure 3–20 中的ND State，驅使STA進行鄰近目標基地台的搜索動作。如 Figure 3–18 及 Figure 3–17 所示，STA依據目前服務基地台是否提供參考性質的相鄰基地台資訊與否，進行目標基地台的偵測或/和搜索。在這個Stage最後，STA會有產生一個經過確認與過濾的Local AP Topology。

要注意的是，雖然我們使用Background Further Observed Scanning 目標基地台搜索方式，但是這個架構並不會排除其它更好的目標基地台搜索的方法，例如：NG Probe (2.4.1.3)、NG-pruning Probe (2.4.1.4)等方法。

3.3.2 Neighbor Monitoring Stage (Resource Establishment)

在完成目標基地台的搜索、產生Local Topology之後，接著STA進入 Figure 3–20中的RCS State，根據即時產生的Local AP Topology，使用 Non-AP-Aligned SyncScan機制對Local AP Topology內的目標基地台進行持續性的量測。在這Neighbor Monitoring期間，STA的「換手決策演算法」可能會根據這些持續性量測的資訊，判別STA接近那些目標基地台，因而驅動執行IEEE 802.11 相關標準所規定的換手程序下所提供的金鑰、QoS資源預先佈署程序。例如IEEE 802.11i增強安全協定標準[3] 所提供Pre-Authentication機制預先佈署PMK驗證金鑰、IEEE 802.11r基地台快速換手程序標準[6] 提供的Pre-Reservation機制，STA可預先設定PMK-R1 驗證金鑰及QoS資源預先佈署的動作。

STA 的「換手決策演算法」可能會並且可以根據這些持續性量測的資訊，進一步調整某些目標基地台的量測頻率，以節省相關資源。

3.3.3 Commit Stage (Transition)

最後，STA的「換手決策演算法」會選出最終的目標基地台，進入 Figure 3–20 中的HO State，進行接下的commit operations，完成換手程序。

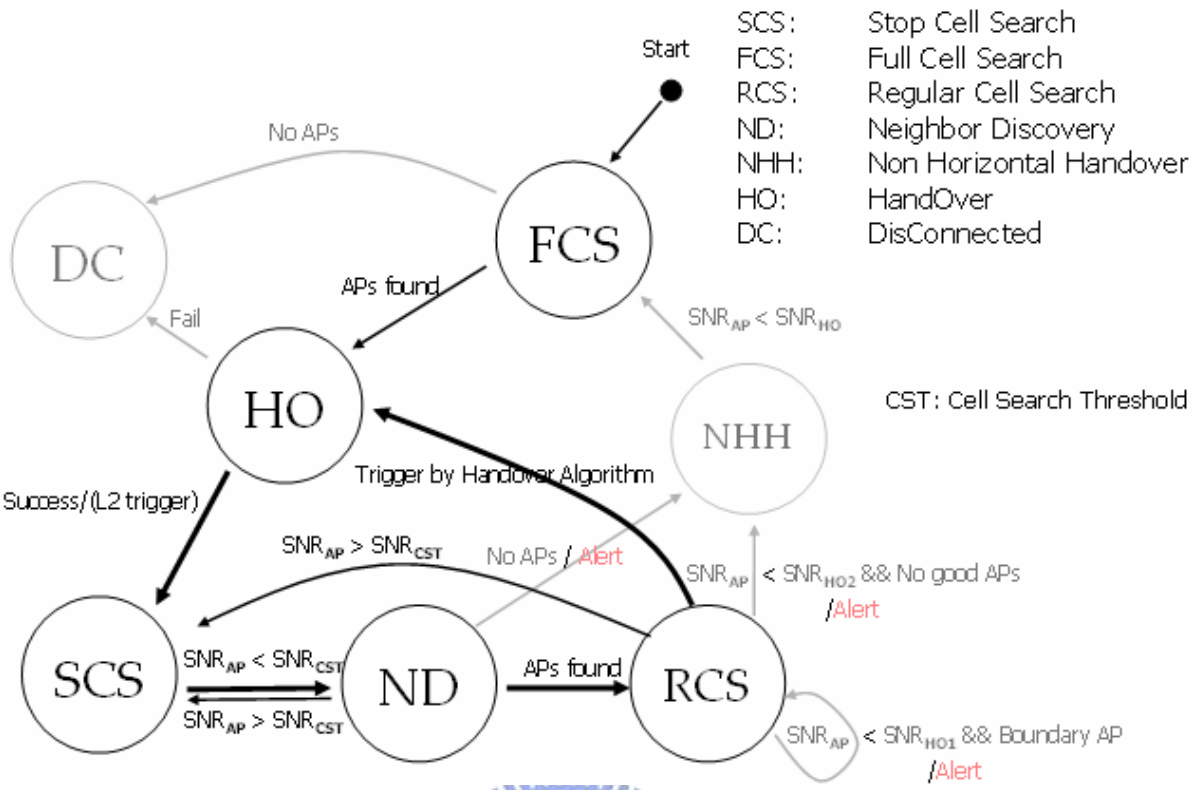


Figure 3–20 State machine used by our seamless handover scheme

Figure 3–20 中的粗黑色實線是STA在同一個ESS下進行基地台間的換手最常驅使行進的狀態轉換路徑。若服務基地台訊號強度SNR_{AP}大於SNR_{CST}時，表示STA與服務基地台的連線良好，不需進行目標基地台的搜索與量測。當服務基地台訊號強度SNR_{AP}小於SNR_{CST}時進入ND狀態，表示STA已經往某個方向移動，需要利用目標基地台搜索機制找出STA此時的Local AP Topology，完全目標基地台搜索後即進入RCS狀態，開始利用Non-AP-Aligned SyncScan對目標基地台進行持續性地的量測動作。最後利用Handover Algorithm與量測的數據演算出換手的時機，進入HO狀態執行換手標準程序進行基地台的換手，完成基地台換手後再進入SCS狀態。

STA一開始進入ESS的情況在 Figure 3–20 中則是從start初始點開始，接著進入FCS狀態執行Full Scanning完整找出可使用的基地台，接著進入HO狀態進行基地台連結，接著進入SCS狀態。

Figure 3–20 中的Alert動作，是在通知上層Mobility Management Entity即將無法繼續與相同ESS下的基地台存取使用無線網路，Mobility Management Entity可能決定進行Vertical Handover連結其它類型的無線網路或是下達連結其它ESS的命令。Alert動作可能發生在ND狀態以及RCS狀態。

4.1 實作層面功能分割考量：Policy vs. Mechanism

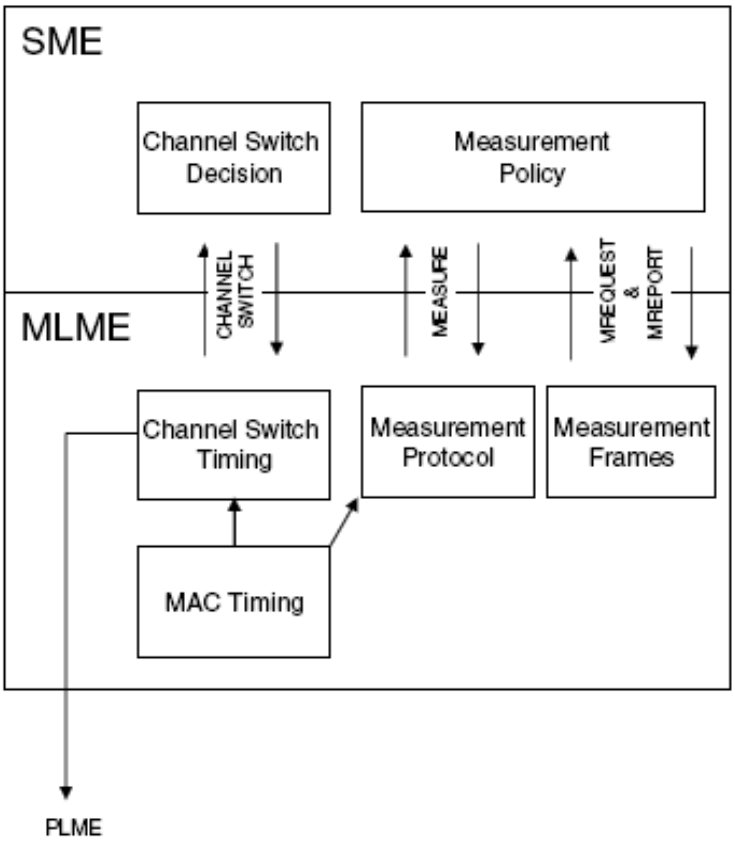


Figure 4–1 Layer management model

以實作的角度來看，可以將換手機制的運作，分成兩個層次來看。第一是所謂的 policy functional block，第二是內部實作驅動底層硬體動作的 abstraction layer，有時候也稱為 mechanism functional block。

以目標基地台搜索與量測機制的運作為例，如 Figure 4–1 所示，我們可以分成進行決策演算的measurement policy functional block以及底層實作對進行一個抽像量測動作的相關protocol/framing functional blocks。

以 measurement policy functional block 而言，這個 functional block 必須決定在什麼時間點對那些目標基地台進行量測，以及怎麼進行量測。例如：量測的基地台順序是如何，量測的頻率是多少。Measurement policy functional block 直接使用底層提供的 measurement interface，並不需要知道底層如何運作。

Measurement interface底部的component就必須負責產生measurement使用的management frame，例如：probe request frame。並使用相關的protocol進行實際的measurement行為，例如：probe request/response exchanges。事實上，MLME (MAC Sublayer Management Entity) 產生包裝好measurement frame之外，最後還需要利用MAC Sublayer提供的SAP (Service Access Point)，見 Figure 2–16，讓MAC Sublayer內部功能方塊，以符合IEEE 802.11 標準[1] 的DCF相關傳送機制對目標基地台傳送measurement frame。

經由這整個流程的說明，我們可以看到「目標基地台的量測機制」的系統實作，會經由 measurement policy functional block, measurement protocol/framing functional block 以及 MAC Sublayer functional block 的垂直分工完成。而我們實作的部分，包括 measurement protocol/framing functional block 及 measurement policy functional block。

4.2 IEEE 802.11 無線網路卡硬體架構與系統介接

IEEE 802.11 無線網路提供OSI七層架構中的data link layer功能，負責相同link上資料的傳送。目前普遍的實作，依其將data link layer部分功能export到Host System的多寡，可約略分成「韌體為主的無線網路卡架構」及「作業系統端為主的無線網路卡架構」。

4.2.1 韌體為主的無線網路卡架構

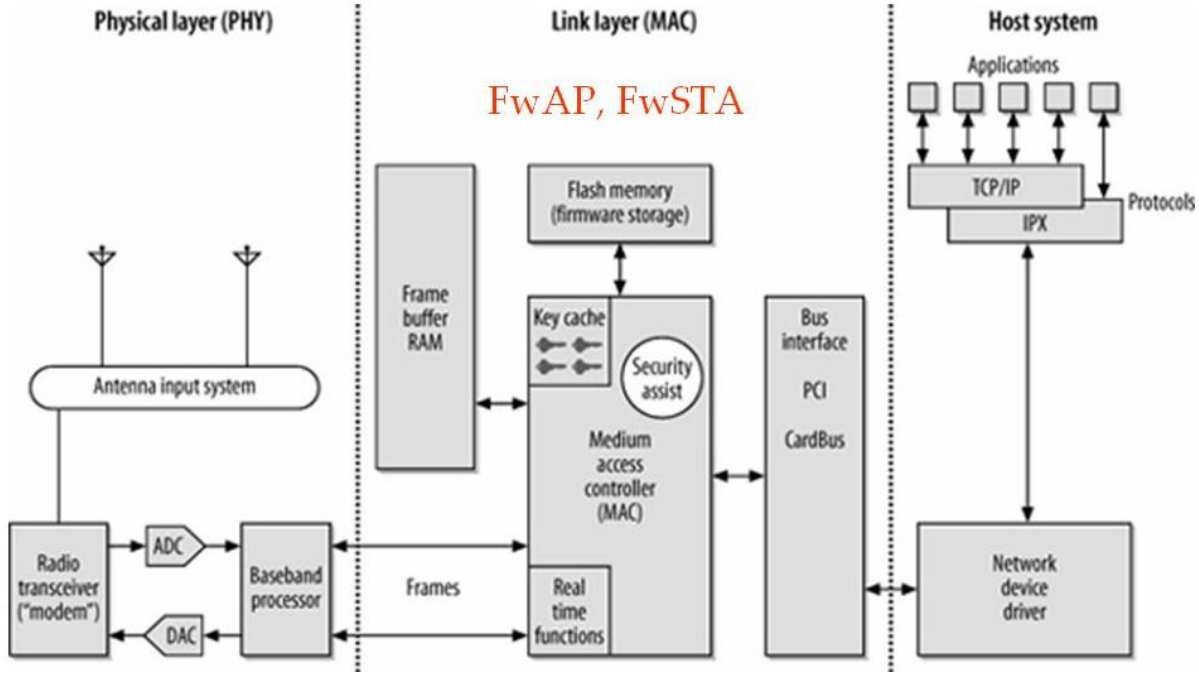


Figure 4–2 FwAP, FwSTA wireless card structure

(Source from [22])

如 Figure 4–2 所示，韌體為主的無線網路卡架構將IEEE 802.11 無線網路data link layer以及PHY layer功能方塊，實作在無線網路卡上，相關measurement policy functional block及protocol functional block也是在無線網路卡上的軟體實作。

這時候 Host System 端的 network device driver 僅提供簡單的 interface，包括傳送 data frame 的 SAP 及簡單的 management interface。Network device driver 可以視為一層薄薄的轉換介面 (adaptation layer)，在 Host System 端並沒有任何 data link layer 的 protocol state machine 存在。

由於在此韌體為主的無線網路卡架構 (也有稱為FullMAC implementation)，IEEE 802.11 無線網路的protocol state machine是實作在硬體本身，Host System端能做的控制僅止於簡單的management功能，並且規格隨不同廠商而有不同。例如：在Host System端只能控制使用者想要連接的SSID，無法指定BSSID連結到那一個特定基地台，也無法控制目標基地台搜索與量測的如何進行，只能下達沒有參數的Scan命令，目標基地台搜索及選擇，完全是由卡上的firmware決定。

這一類的IEEE 802.11 無線網路卡，我們可以稱呼為FwAP、FwSTA。其Host System Software、Device Driver、與無線網路卡本身的關係，在 802.11 basic reference model 下如 Figure 4–3 所示。

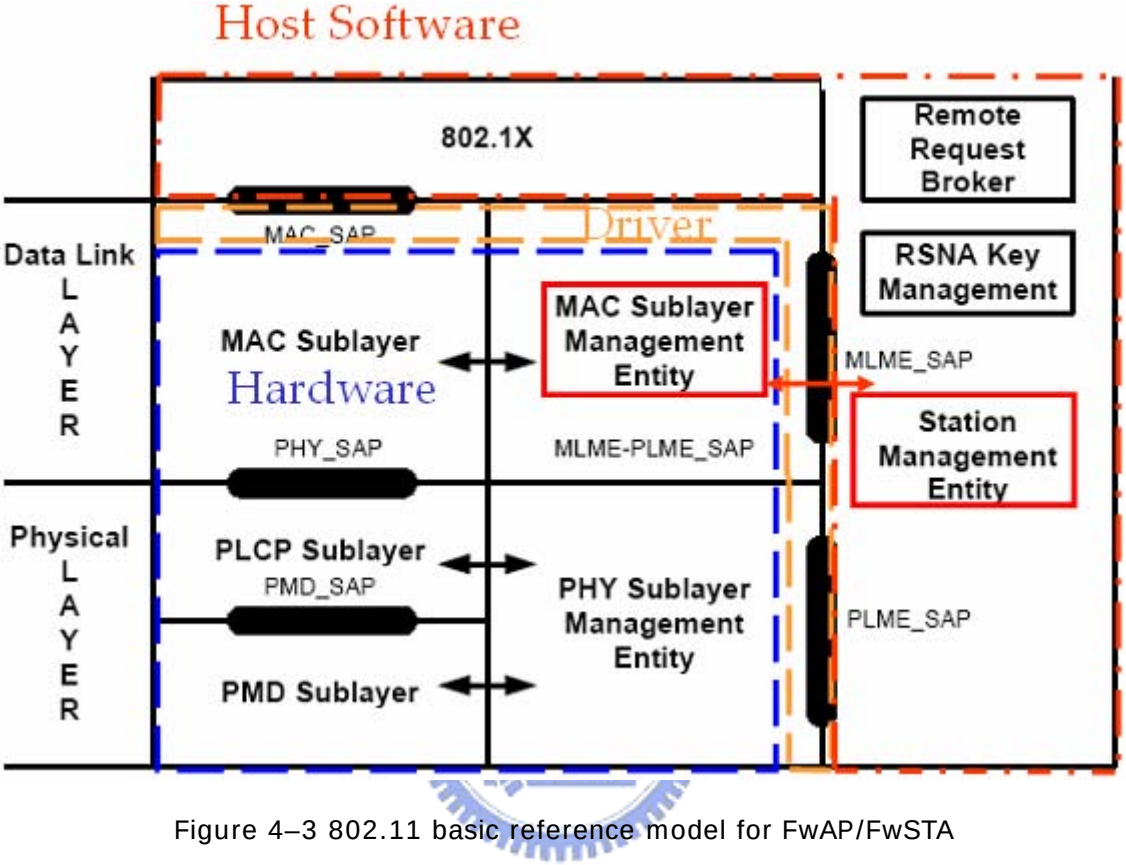


Figure 4–3 802.11 basic reference model for FwAP/FwSTA

(Adapted from [6])

4.2.2 作業系統端為主的無線網路卡架構

如 Figure 4–4 所示，作業系統端為主的無線網路卡架構將IEEE 802.11 無線網路 data link layer，部分實作在無線網路卡硬體內部、部分實作在Host System端。相關 policy functional block及protocol functional block是用Host System端上的軟體實作的，可視為network device driver層級的一部分。

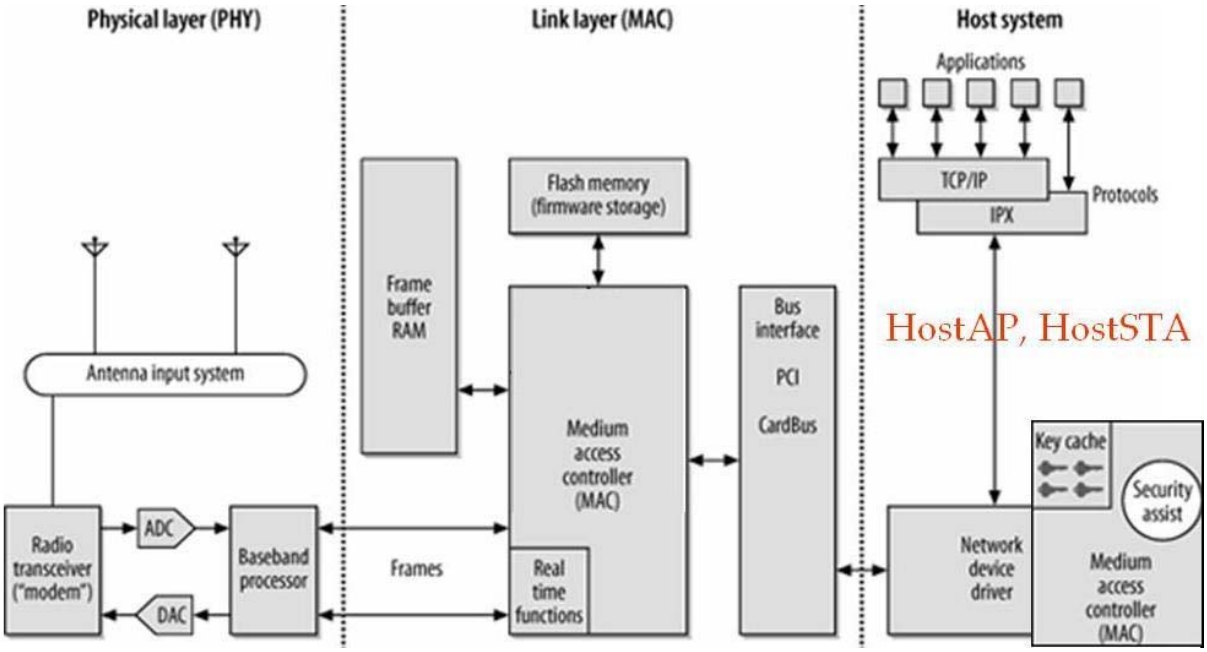


Figure 4–4 HostAP, HostSTA wireless card structure

(Adapted from [22])

這時候 Host System 端的 network device driver 則不再是提供簡單的 interface 而以，需要實作更多 IEEE 802.11 MAC 功能。除了需要提供傳送 data frame 的 SAP 之外，management interface 內部各類 protocol/framing functional block 也需要進行實作。Network device driver 這時候不僅僅是一層轉換介面，data link layer 的 protocol state machine 在這個模式下是實作在 Host System 端的 device driver。

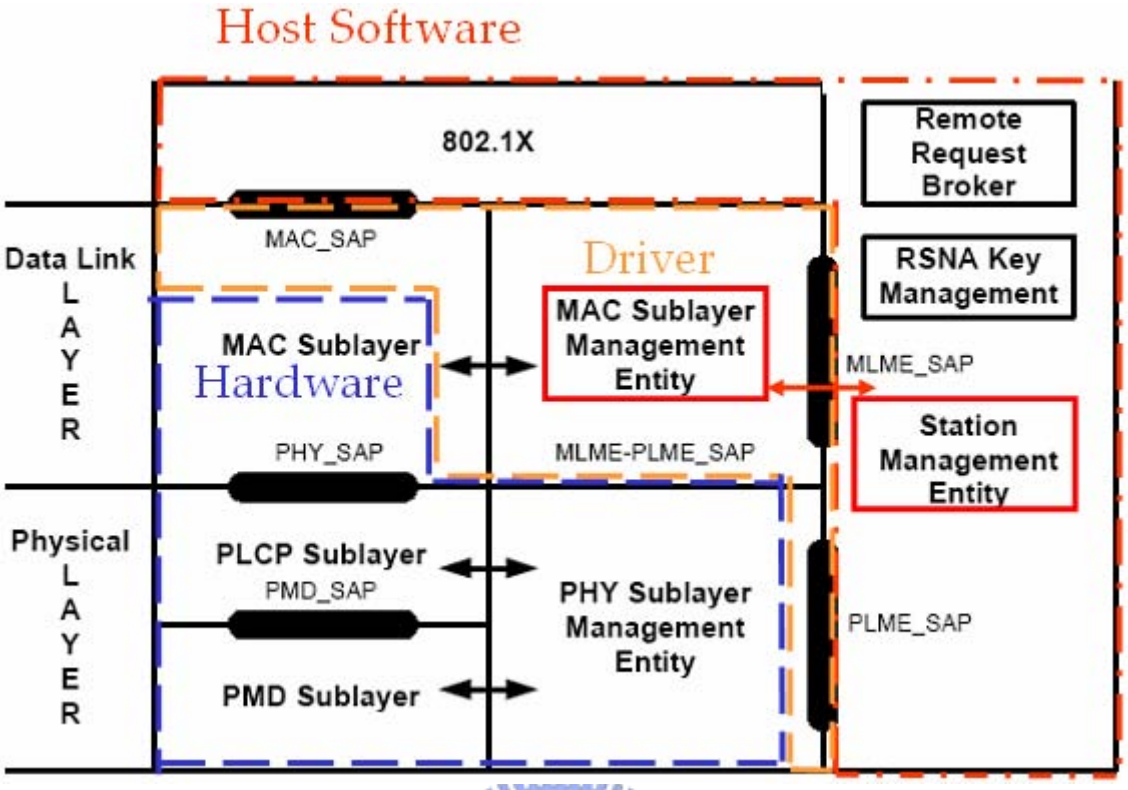


Figure 4–5 802.11 basic reference model for HostAP/HostSTA

(Adapted from [6])

由於在作業系統端為主的無線網路卡架構下 (也有稱為SoftMAC implementation)，IEEE 802.11 無線網路的Protocol State Machine及Management Entity是實作在Host System內，開發者便能更有彈性去實作及提供不同的management 進行模式。例如：在HostAP driver (見4.4.1) 當卡上的firmware運作在HostAP模式及 Madwifi driver (見4.4.2) 對IEEE 802.11 WLAN的實作，都是屬於這種架構。

這一類的IEEE 802.11 無線網路卡，我們可以稱呼為HostAP、HostSTA。其Host System Software、Device Driver、與無線網路卡本身的關係，在 802.11 basic reference model 下如 Figure 4–5 所示。

4.3 Linux 作業系統的網路子系統

我們使用的目標平台是Linux作業系統，Fedora Core 5 Distribution，及Linux 2.6.16 版本的kernel。至於IEEE 802.11 無線網路的data link layer則是使用Prism 2 chipset的無線網路卡及使用HostAP driver (見4.4.1) 來實作我們設計的無縫換手機制。為了了解我們實作的範圍，首先我們先看看Linux作業系統的網路子系統是怎麼運作的。

4.3.1 The Whole Picture

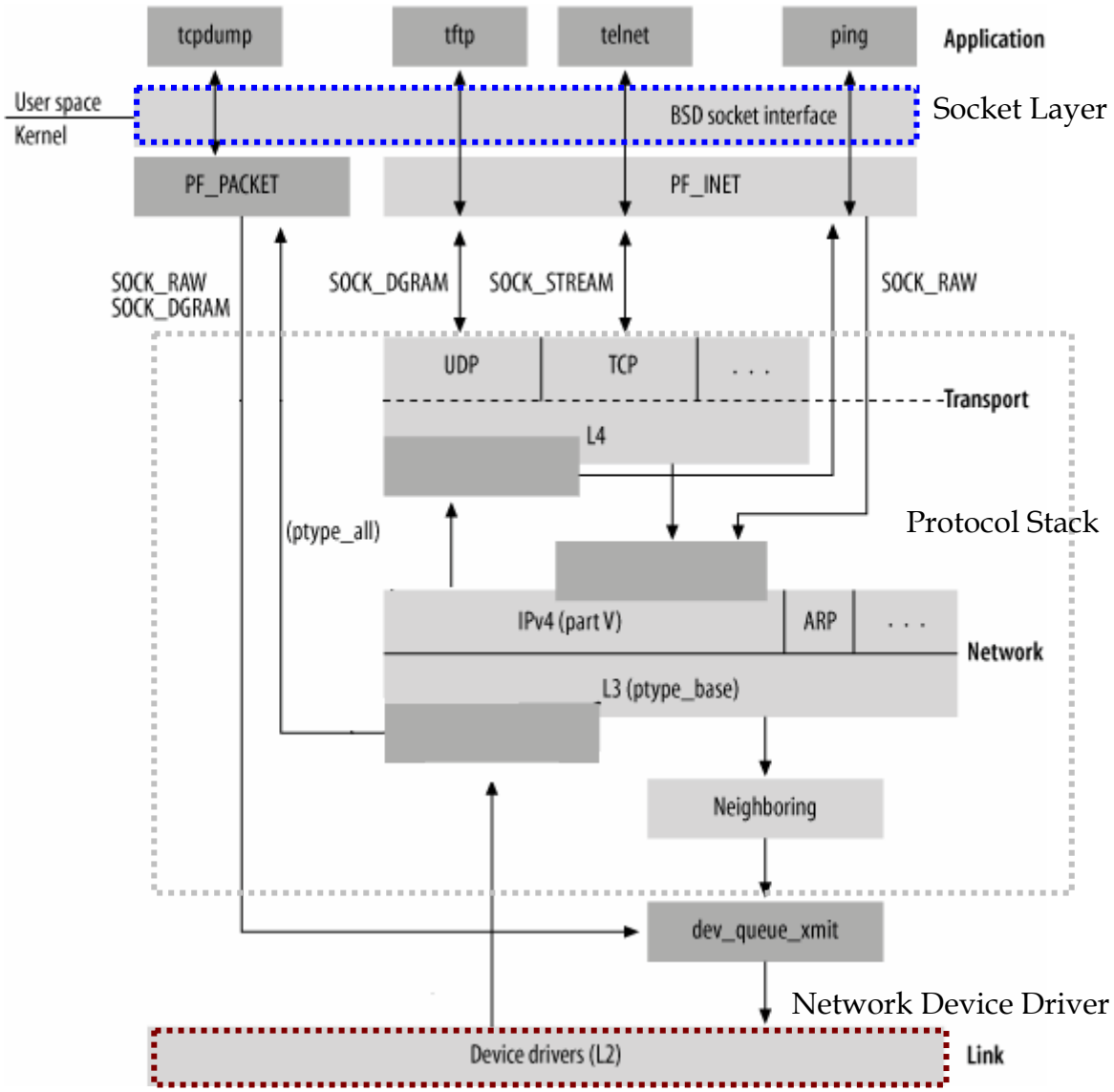


Figure 4–6 The whole picture of Linux networking framework

(Adapted from [24])

如 Figure 4-6 所示，Linux networking framework可以分成三大塊：Socket Layer、Protocol Stack、Network Device Driver。其中Socket Layer及Network Device Driver都有socket buffer[19] queues供封包中繼暫存；而Protocol Stack只是一堆Protocol-Specific用來供包裝及卸除各類封包header所使用的functions及相關Protocol Stack內部運作及處理的functions。

4.3.1.1 Socket Layer

在 Linux 作業系統下，Socket Interface 不但是 transport layer 與其它在 transport layer 以上的 protocols 的介面，同時也是 user space 跟 kernel space 之間的介面之一。事實上在 Linux 作業系統的實作，Socket API 眾多的 interfaces 是利用一個稱為 sys_socketcall 的系統呼叫所實作的。

Socket Layer 提供了一個與底層協定不相關的標準介面，供 user space 的 applications 對於 peer entities 進行傳送與接收資料。由於 Socket Layer 的與底層協定的不相關性，使得 peer entities 可以是同一台電腦上的 applications，別台電腦上的 applications，甚至是 kernel space 的 components。

值得一提的是，在 Linux 作業系統下 user space 的程式要對 network device 進行管理及設定的介面，最常使用的就是使用 ioctl 透過以 socket 當作 fd 的方式進入 kernel space。

4.3.1.2 Protocol Stack

Protocol Stack 的部分，可以說就是 Linux networking framework 中使用的 library functions。當使用者初始化好 Socket 之後，就已經決定好這個 Socket 底部使用的 network layer protocol 及 transport layer protocol 相關的 functions。

而 Linux 作業系統在 layer 與 layer 間，例如 L2→L3 及 L3→L4，也提供相關 protocol handler 註冊的機制，使得從網路卡收到的封包，可以經過一連串註冊好的 protocol stack handler 處理完之後才送到目標的 Socket Layer 暫存，等待 user space 的程式存取。最常使用的也是最著名的 protocol stack 當屬 TCP/IP protocol suites。

---

[19] Packet 在 Linux kernel 內是以 Socket buffer 資料結構呈現。

4.3.1.3 Network device driver

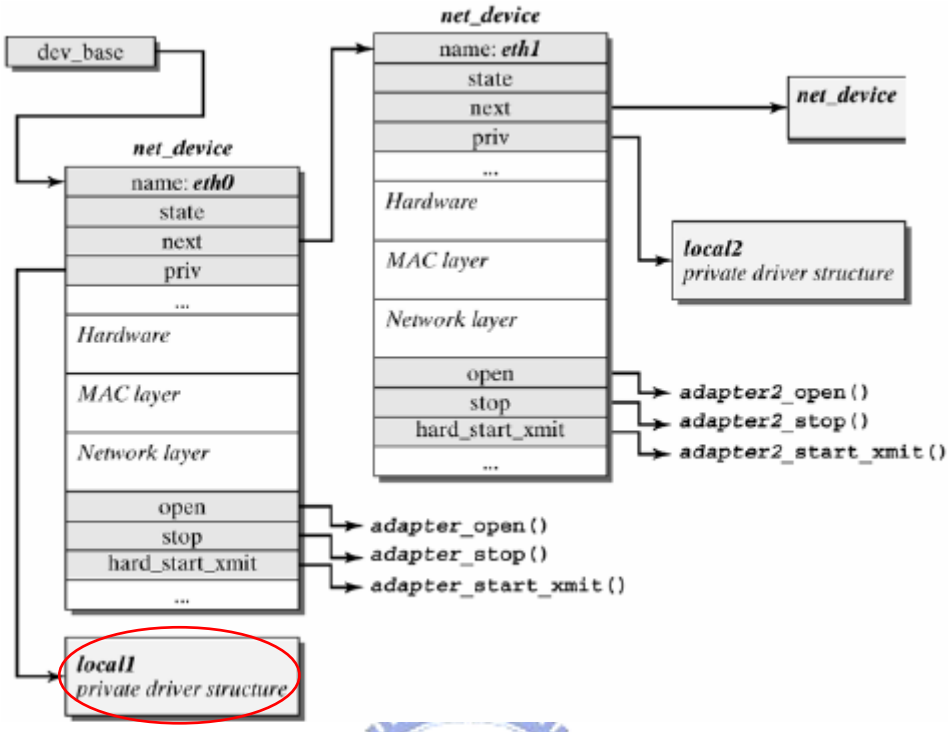


Figure 4–7 Network devices management in Linux kernel

(Source from [25])

如 Figure 4–7 所示，Linux kernel對network device提供了一個一般化的管理介面，讓不同性質的network device經由struct net_device資料結構初始與註冊，提供Linux kernel以device-independent的方式，對於該network device進行初始化、設定、管理、及傳送封包進行device-specific呼叫。此外，network device driver的內部實作需要用到的資料結構，通常會利用 Figure 4–7 紅色圈起來的priv欄位自行定義使用及存放。

每一個 network device 都必須提供一個 struct net_device 的實體、並且也是經由這個資料結構的註冊，將 network device 連接上 Linux networking framework。

以傳送端而言，封包經過 protocol stack 相關 functions 的處理後，即 enqueue 到目標 network device 的 socket buffer queues，等待接下來 network device 的傳送處理；以接收端的角度來看，封包從網路卡接收到之後，會依據 L2→L3 protocol stack handler 的註冊資訊，由相關的 protocol stack 往上進行處理，最後送到 Socket Layer 暫存。

我們實作的無縫換手機制是屬於 data link layer 的範圍，在 Linux 作業系統就是屬於 network device driver 的部分，因此我們的實作，都圍繞在 network device driver 的範圍，沒有動到 Linux networking framework 的實它部分。

4.3.2 Wireless Extension

IEEE 802.11 interfaces在已連結基地台、正常運作的情況下，和Ethernet interfaces幾乎沒有什麼不同。但是由於底層是使用radio technology進行資料的傳輸與接送，因此interface需要提供許多management相關的設定動作，而這些設定的介面，就是在Figure 2–16 802.11 basic reference model中的MAC Sublayer Management Entity所提供的MLME_SAP介面。

Linux作業系統為了讓不同廠牌的IEEE 802.11 無線網路卡能夠使用相同的設定介面，在Linux kernel中制定了一套MLME_SAP的adaptation layer介面，稱為Wireless Extension[21]。我們在 4.3.1.1 最後有提到使用ioctl設定network device的介面，Wireless Extension就是這個介面對於WLAN network device的擴充。

Jean Tourrilhes[21] 提供了一套根據Wireless Extension實作的user space configuration utilities稱為Wireless Tools。Wireless Extension API可以幫助應用程式的開發者，使用相同的device-independent API控制並設定不同的無線網路卡。

4.3.3 Linux Kernel Threading

Linux networking framework 運作的 philosophy，就筆者的眼光來看，就是

1. 在不同 layers 間提供相關 manipulating functions 的註冊機制。接著
2. 由某種機制產生 kernel execution path，順著規劃好的路徑，一路執行相關註冊的 functions，直到該 execution path 結束。若
3. 中途暫時無法繼續執行，則將封包 enqueue 到某些 layer 提供的中繼站 socket buffer queues 中，之後會由其它 kernel execution path 接著繼續處理。

Kernel execution path 依該 execution path 是否擁有「被暫時停止，之後再從停止處繼續執行」的能力，分成二大類。第一類是具有 process context 的 kernel thread；第二類是使用 interrupt context 的 ISR。

Linux 作業系統內可以被 manage 的 execution path，在 kernel space 下都具有一組 kernel thread、kernel stack，user space 的程式除了擁有一組 kernel thread、kernel stack 之外，另外還擁有一組 user thread 和 user stack。User space 的程式在使用 socket interface 傳送資料的時候，會利用自己的 kernel thread 執行相關 protocol stack 的傳送端 function，直到把封包 enqueue 到 network device queue 為止。

Thread 在執行的過程中可以被暫停下來，並且使用獨立的空間，因而不會受到其它 execution path 的影響。雖然 Thread 可以被 manage，對於 programming 而言比較容易，但正因為需要 manage，所以 thread 使用了額外 manage 需要的資源，kernel 切換到某一個 thread 時，會需要相關的轉換，稱為 context switch。

當kernel space內處理networking的事情變的比較複雜、或是為了將該負責的entity獨立出來，我們會使用kernel thread來實作。Linux kernel對此提供的facility稱為workqueue。我們實作的Handover Management Entity (見4.5.2) 就是用kernel thread來實作的。

Linux 作業系統內不能被停下的 execution path 為使用 interrupt context 的 ISR，這一類的 execution path 只能勇往直前，直到 path 結束。根據 trigger 的來源，又分為 Hardware ISR 和 Software ISR。

在 Linux networking framework 中，用到的 Hardware ISR 主要是 network device driver 註冊的 ISR，從事作業系統和硬體間搬動資料的相關動作，並且驅動特別指定的 software ISR，執行所謂 Bottom Half 的工作。

Software ISR 是由 Hardware ISR 執行完後，在打開硬體 interrupt 的功能下接著執行的 ISR。由於 Hardware ISR 可以中斷 Software ISR，將 ISR 再區分成兩類的主要目的是讓硬體有更高處理的 throughput。Software ISR 在 Linux kernel 提供的 facility 出現的型式有 softIRQ、Tasklet、Timer，大部分 networking 的 protocol stack 下的 functions 都是由 Software ISR 所執行。

Network device driver 最常使用的 software ISR 當屬 Linux kernel 所提供的 Tasklet 和 Timer facility，這兩者都可以 interrupt 一般的 threads，使用時機的差別在於有無指定的目標時間，若需要在指定時間點執行 software ISR 則會 schedule Timer，若沒有指定的時間則會 schedule Tasklet。

4.4 IEEE 802.11 無線網路卡硬體與驅動程式相關資源

4.4.1 HostAP driver

HostAP driver [19] 是使用Intersil's Prism 2/2.5/3 chipset的IEEE 802.11 無線網路卡在Linux作業系統下發展的driver。HostAP driver支援firmware運作在所謂的Host AP模式，此時Host System端則需要提供其它相關的功能，例如：802.11 management functions。因此使用Intersil's Prism2/2.5/3 chipset的IEEE 802.11 無線網路卡配合Host System端的HostAP driver，可以提供基地台的功能。除此之外，HostAP driver也支援firmware運作在STA的模式 (BSS or IBSS)。

Intersil 對 Prism2/2.5/3 chipset 的 IEEE 802.11 無線網路卡提供的 station firmware 支援所謂的 Host AP 模式，在 Host AP 模式下，firmware 會處理 MAC Sublayer 內具 time critical 的工作，例如 beacon 的發送還有收接 frame 時 Ack 的回應，至於 MAC Sublayer 其它部分及 MAC Sublayer Management Entity 則留給 Host System 實作。

當Prism2/2.5/3 chipset的IEEE 802.11 無線網路卡的station firmware運作在Host AP模式時，firmware與driver間的分工是屬於 4.2.2 所介紹這一類的架構；至於運作在STA模式 (BSS/IBSS)，則firmware與driver間的分工是屬於 4.2.1 所介紹這一類的架構。

4.4.2 Madwifi driver

Madwifi driver [20] 是使用Atheros chipset的IEEE 802.11 無線網路卡在Linux作業系統下發展的driver，配合firmware提供的功能，Madwifi driver支援STA (BSS/IBSS) 模式、及AP模式的運作。不管是STA模式或AP模式，IEEE 802.11 protocol都是實作在driver的層次，與device硬體本身無關。

Madwifi driver 分成兩部分：一部分是 Open Source 對於 IEEE 802.11 protocol 及 MAC Sublayer Management Entity 的實作；一部分是所謂 Hardware Access Layer (HAL) 的 closed-source library。Madwifi driver 採取部分 Open Source、部分 closed-source 作法，仍源自於 Atheros chipsets 無線網路卡硬體提供 configure 項目的彈性，使得 Host System 端可能將 RF component 運作在合法的 Frequency band 之外，為了符合 FCC 的規範，Madwifi driver 將可能違反規定的實作設定部分採取 closed-source 的方式處理。

Atheros chipset的IEEE 802.11 無線網路卡的firmware與driver間的分工是 4.2.2 所介紹這一類的架構。

4.5 無縫換手機制之實作與系統架構

我們使用 Prism 2 chipset 的無線網路卡及 HostAP driver 實作我們設計的無縫換手機制。

由於當Prism2/2.5/3 chipset的IEEE 802.11 無線網路卡的station firmware運作在STA模式時，其運作架構是屬於 4.2.1 所提的架構；不過Prism 2 chipset的station firmware在STA模式下有提供更多的功能，依「目標基地台的搜索決策」與「目標基地台的選擇」對Host System開放的程度分成三類，如 Table 4–1 所示：

Table 4–1 Three levels of support for Intersil's station firmware operating in STA mode

	Scan Decision Making	**Join AP Decision Making**
Mode 0	Firmware	Firmware
Mode 1	Firmware	Host System
Mode 2	Host System	Host System

雖然 Prism 2 chipset 的 WLAN card 在 STA 模式下有提供 Mode 2 讓 Host System 能有更大的自由度選擇目標基地台連結及進行目標基地台搜索的時機，但是沒有辦法讓 Host System 自行實作目標基地台搜索與量測進行的方式，並不符合我們的需求。因此，我們是使用 Prism 2 chipset 的 WLAN card 運作在 HostAP 模式下的情況，來實作我們設計的無縫換手機制。

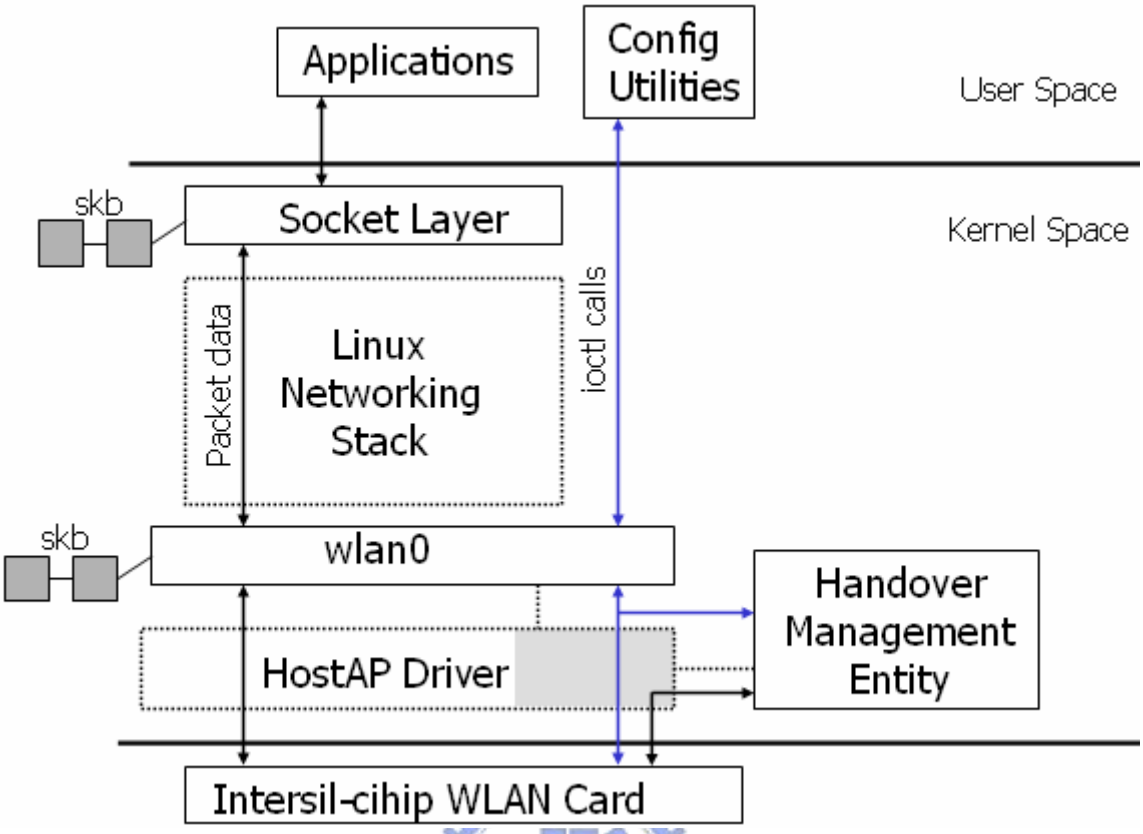


Figure 4–8 Architecture of our seamless handover scheme implementation

4.5.1 Partial MAC Sublayer Functionalities & MLME Component

由於HostAP driver對於Prism 2 chipset的WLAN card運作在HostAP模式只實作了基地台的功能，並沒有實作STA的功能，因此我們自己也必須實作Prism 2 chipset的WLAN card運作在HostAP模式下，Driver要提供的相關功能，這部分的實作會大量使用到 4.3.3 所提 kernel提供的Tasklet和Timer facility。如 Figure 4–8 所示，HostAP Driver內部灰色的部分代表我們對HostAP Driver修改的部分，其中包括：

1. Figure 2–16 802.11 basic reference model 中的MAC Sublayer對於STA使用的802.11 protocol state machine
2. 目標基地台搜索與量測需要的 measurement protocol/framing functional block
3. MAC Sublayer Management Entity、MLME_SAP[20]相關介面與功能

[20] 我們實作了 Figure 3–4 中的 MLME-SCAN.request() 大部分可供指定參數的功能。

4.5.2 Handover Management Entity

此外我們在Linux kernel內新增一Handover Management Entity，如 Figure 4–8 所示，負責目標基地台的搜索與量測的決策與換手的決策。這部分是利用 4.3.3 所提 kernel提供workqueue facility實作。

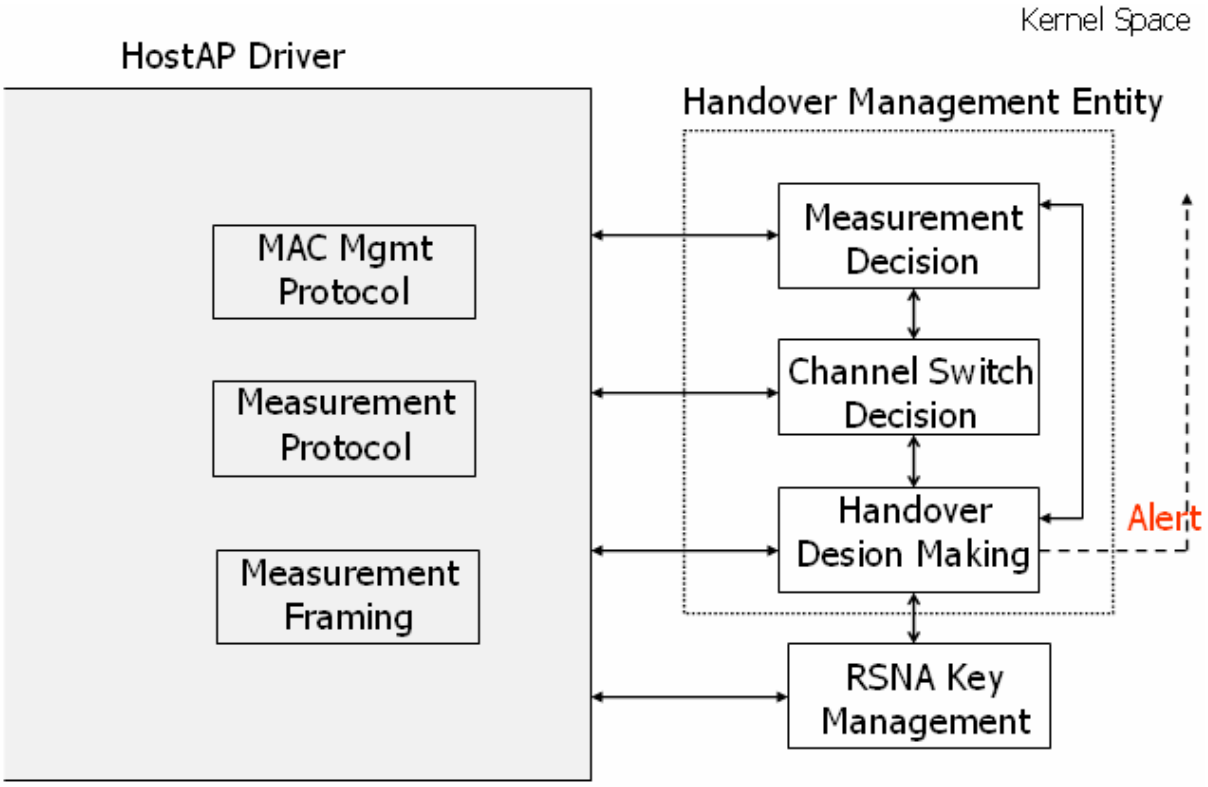


Figure 4–9 Detail components of our handover mechanism implementation

至於實作的元件細節，可以參考 Figure 4–9。Handover Management Entity內部包括measurement decision functional block、channel switch decision functional block、及handover decision making functional block，Figure 3–20 的state machine 就是實作在Handover Management Entity之內。

Handover Management Entity根據服務基地台的訊號強度以及目標基地台的量測結果，進行如 Figure 3–20 設計的狀態圖，其Handover相關的決策，有時也會驅動往上層 Mobility Management Entity 的 Altert 驅動信號，或者是對 RSNA Key Management下達相關IEEE 802.11i增強安全協定標準[3] 802.1X Pre-authentication的動作或是 802.11r基地台快速換手程序標準[6] 的Fast Transition Authentication Sequences動作的要求。

值得一提的是，Handover Management Entity和HostAP driver之間的溝通，我們大部分採用Wireless Extension (見4.3.2) 所定義的API，因此Handover Management Entity也可以實作在user space。不過由於「目標基地台量測」機制會接收許多量測的Beacon/Probe Response frames，為了減少messages transfer及context switch overhead和避免scheduling priority的問題，我們選擇利用kernel space下單獨的kernel thread實作。

第五章　效能評估

本章節對於我們所設計與實作的無縫換手機制進行相關評估，主要進行的方式是以系統端 (Host System) 對於無線網路卡設備使用的觀點與層面來看待這些動作背後所需的花費。我們沒有使用 Sniff 攫取 frames 的方式分析效能的原因，主要在於我們認為光看攫取到的 frames 只能捕捉到這些動作執行的瞬間，不足以代表系統內部軟體元件能夠得到的執行效能。因此我們接下來的評估，都是以在系統核心內部加入偵測點的方式量測每一個動作從初始到結束得到結果所需等待的時間。

首先，我們在 2.4 有提到SME透過MLME-SAP介面向MLME呼叫相關的MAC management function的動作，並且在 4.2 也粗略介紹了 Host System 與網路卡設備介接的關係，大致可分為韌體為主及作業系統端為主兩種架構，其中這兩種架構在 Figure 2–16 802.11 basic reference model 中，host software、device drive、和網路卡的關係分別可由 Figure 4–2 及 Figure 4–4 約略表示。

5.1 單一動作所需的花費

在 4.2.1 談到的韌體為主的無線網路卡，其提供的介面所能供給SME執行management function的能力 (capability)，通常隨著不同的廠商實作而有不同。我們以使用HostAP driver的Prism 2 Chipset無線網路卡運作在STA模式為例，最早的MLME-SCAN SAP只能用來呼叫一個完整的active scan動作而以，這個動作從初始到整個結束由網路卡回傳結果，大約需要 1000 ms；後來修改過後的firmware，將這個MLME-SCAN SAP開放成可以指定SSID及可指定搜尋的channels，我們實驗結果量得在此種模式下，單獨搜索一個channel從Host System初始到firmware回應結果，需要花費約 75 ms的時間。

我們使用相同的硬體設備，但是將無線網路卡運作在 HostAP 模式，而我們自行在 Host System 實作 STA 的功能與其 802.11 protocol state machine。在這樣的模式下，搜索一個 channel 所需的時間為 2 * Channel_Switch_Time + Broadcast_Frame_Transmit_Time + PowerSave_Overhead + ProbeWait_Time。

其中Channel_Switch_Time為硬體設備本身的限制，在我們的實作平台測量結果需要 10 ms，Broadcast_Frame_Transmit_Time需要 2-3 ms，PowerSave_Overhead約需 2-3 ms，而ProbeWait_Time則是可以由呼叫端指定，若無指定時實作預設為 10 ms。我們將這三類active scan function執行所需的時間，整理成 Table 5–1。

Table 5–1 Needed time to perform active scan function

Type	one channel	total 11 channels
Old STA mode + HostAP STA driver		1000 ms
New STA mode + HostAP STA driver	75 ms	700 ms
HostAP mode + our STA driver	34 ms	250 ms[21]

我們從上面的數據可以看出，Channel_Switch_Time佔具搜索一個channel動作大部的時間（20/34 = 58.8 %），然而不同的硬體的Channel_Switch_Time也不同，I. Ramani and S. Savage[13] 提到Atheros 5212-based Chipset的無線網路卡 (4.4.2) 的Channel_Switch_Time僅需要為 5 ms，假設其它組成不變，搜索一個channel則需要 24 ms的時間，Channel_Switch_Time仍佔具可觀的時間比例 (10/24 = 41.6 %)。從實務的觀點來看，Channel_Switch_Time是以每個channel為基礎增加，並且也佔搜索動作不少比例的時間，這是除了SyncScan[13] 外，大部分研究報告所沒有考慮的因素。

接著我們以在我們實作平台 (Prism 2 Chipset NIC + our STA driver) 所測量到的數據，製表 Table 5–2 來比較我們基地台搜索與量測機制所用到各別單一動作的花費。

Table 5–2 Cost for individual AP measurement operation

Operation Type	ChSw	PS op	Extra traffic	Wait time	Total cost
Active Scan one channel	2 * 10 ms	2-3 ms	Yes, 2-3 ms	ProbeWait, 10ms	34 ms
Unicast Probe one AP	2 * 10 ms	2-3 ms	Yes, 2-3 ms	0 ms	24 ms
Non-AP-Aligned SyncScan for one AP	2 * 10 ms	2-3 ms	No	0-5 ms	24 ms

ChSw: Channel Switch Time, **PS op**: PowerSave operation

[21] 250 ms 而不是 330 ms 的原因在於，Full Scanning 需要的 Channel Switch 一共是 11+1 = 12 個，而不是 2 * 11 = 22 個。

Table 5–2 中的三種量測動作比較值得注意的差異點為 1. 是否會產生額外的traffice以及 2. 等待量測frames的時間。對於active scan以及Unicast Probe而言，都會產生額外solicitation request/reponse的traffic，並且active scan產生的response會比Unicast Probe產生的response還要多；而Non-AP-Aligned SyncScan由於是量測基地台周期性傳送的beacon frame，並不會產生額外的traffic。

接下來要討論的是等待量測 frame 的時間。Active scan 和 Non-AP-Aligned SyncScan 都需要等待基地台發送的 probe response 及 beacon frame，active scan 需要等待的原因，在於 STA 必須等待目標 channel 下的基地台都回覆 probe response frame，而 Non-AP-Aligned SyncScan 則是等待基地台在 TBTT 傳送的 beacon 藉以量測。相對於這兩種需要等待的量測動作，STA 使用 Unicast Probe 則不需要等待。一來是若量測的目標基地台不存在於 STA 通訊的範圍內，則 STA 在傳送 unicast probe request 時便無法收到 positive ack，便可以得知不須要對此基地台進行量測的結果；再者是 STA 只需要收到唯一一個量測目標基地台的 probe response frame 即可完成量測動作，並且因 STA 和基地台彼此共用相同的 Wireless Medium，訊息傳送的時間很短幾乎可以忽略。

5.2 Non-AP-Aligned SyncScan 的正確性及有效性

我們在 3.1.1 談到根據IEEE 802.11 標準[1] 11.1.2.1 規定，基地台必須在自己維護用來BSS timing synchronization用的clock time時間點為 0, T, 2T …等TBTT發送beacon，其中T為基地台的beacon interval。這些資訊會由beacon和probe response frame攜帶，因此STA可以透過與基地台的第一次接觸後，即可得知基地台發送beacon的時間表。雖然標準有規定，但實際的實作是否遵循？以及在Host System端是否有辦法有效地使用這些資訊？根據我們的評估，答案是肯定的！

由於 beacon 和 probe response frame 攜帶 timestamp 以及 beacon interval 的資訊，當 Host System 端收到網路卡往上送的 frame 時，Host System 即可利用 Host System 端以 millisecond 解析度的 clock 推算並計錄基地台會在 Host System 端的什麼時間點發送 beacon frame。我們從實作在 Host System 端的 STA driver 所產生的 logging messages 的結果可以看出，我們的確可以從 beacon 和 probe response frame 推算出基地台傳送 beacon 的時間，並且在該時間收到該目標基地台的 beacon frame，藉以進行目標基地台的量測的動作。

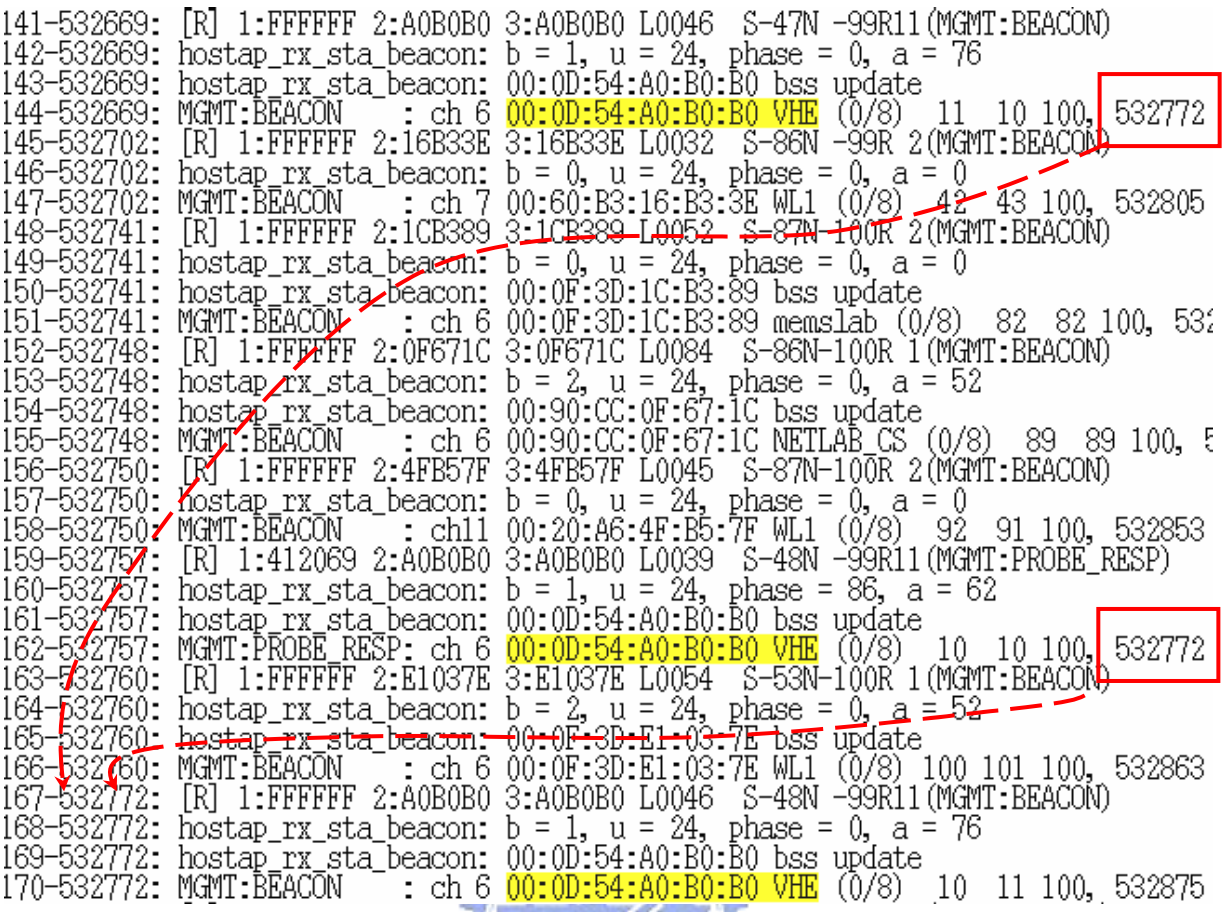


Figure 5–1 Logging messages showing the beacon prediction from the previous received beacon and probe response frames

Figure 5-1 顯示了我們從beacon和probe response frame所攜帶的timestamp及beacon interval推算出基地台下一個beacon傳送的時間。在這個scenario下，無線網路卡並沒有切換channel，工作在channel #6 下，當STA在編號 144，時間 532669 ms收到一個VHE的beacon frame，推算出下一個beacon傳送的時間為本地端 532772 ms；另外STA在編號 166，時間 532757 收到由VHE發出的probe response frame，亦推算出基地台下一個發送的beacon frame的時間點為本地端 532772 ms。

我們所觀察到的基地台，大部分都符合IEEE 802.11 標準[1] 的規範，在基地台自己用來維護BSS timing synchronization使用的clock time時間點 0, T, 2T …等TBTT發送beacon。在論文定稿之前，我們尚未發現到沒有按照標準實作的基地台；即使如此，對於那些beacon發送時序沒有按照標準實作的基地台，我們仍然可以使用Unicast Probe在得知目標基地台存在後對其進行量測，使用Unicast Probe方式進行量測，亦會比原始使用的active scan來得有效率。

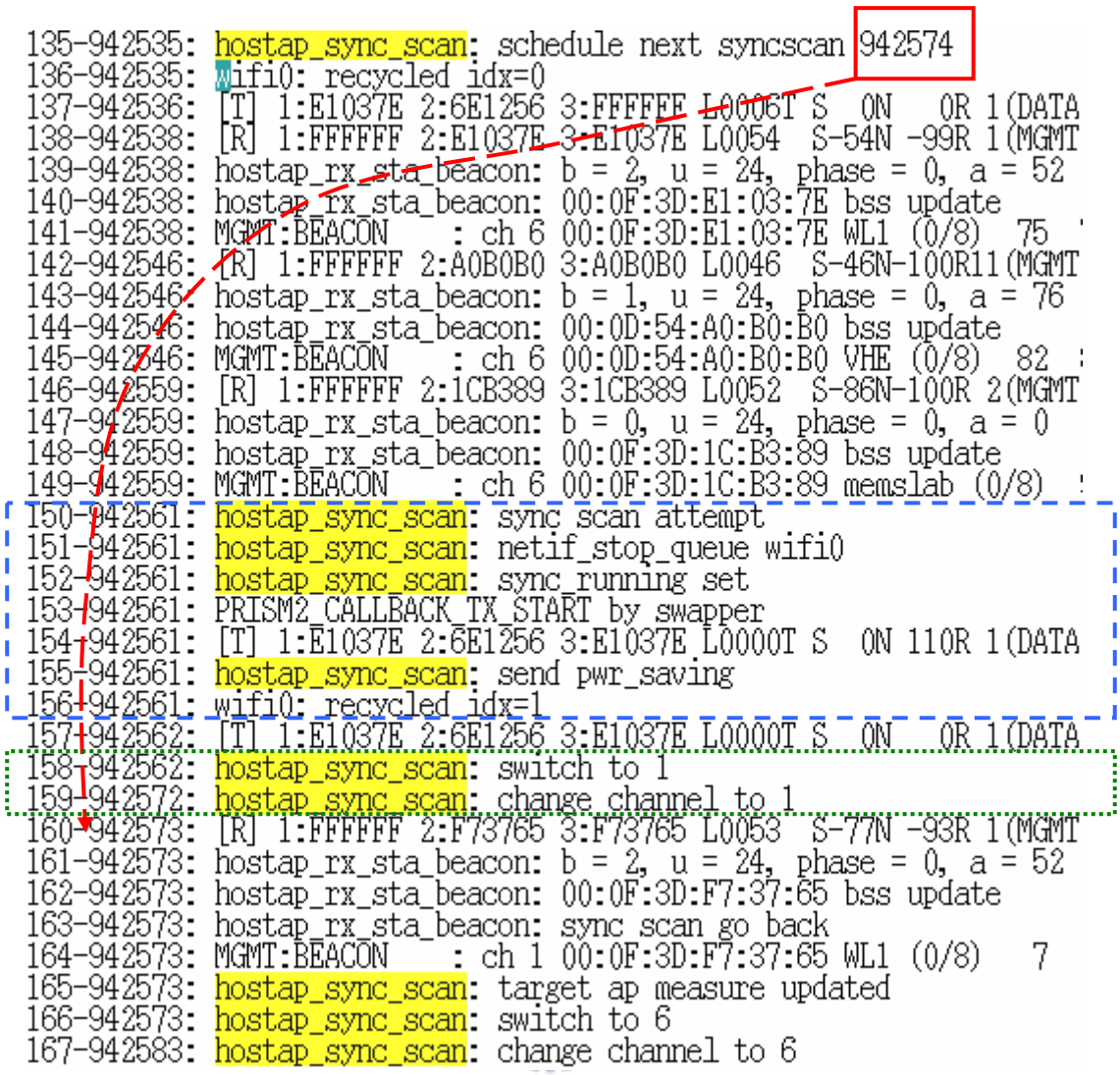


Figure 5–2 Logging messages showing the internal Non-AP-Aligned SyncScan operation

接下來我們要展示我們自己實作的STA driver進行SyncScan對目標基地台量測的流程。如 Figure 5–2 所示，STA在編號 135、時間點 942535 ms排入一個目標為 942574 ms出現在channel #1 的beacon。為了執行這個目標基地台的量測動作，STA在編號 150、時間點 942561 開始初始，用 2 ms執行了PowerSave的動作，並且在 942562 ms開始切換channel、942572 ms完成切換到channel #1 後，無線網路卡的tranceiver開始正常運作，並且在 942573 ms收到量測目標基地台發送出來的beacon frame。

從上述的流程以及輸出的 logging messages 可看出，STA 由 beacon 和 probe response frame 中所演算出來的基地台 beacon 發送時間，可會有 1-2 ms 的誤差，並且 beacon frame 可能會因 channel busy 而延後發送。我們在 Host System 端的設計，必需考量到這些因素，提早並且多等待幾個 ms 以確保有比較大的機會聽到 beacon。

(a)

BSSID	ch	Sig	Noi	age	Phr	int	cnt	sct	uct	ESSID
00:0F:3D:E1:03:7E	06	-70	-100	00041	78	100	001	000	000	WL1
00:20:A6:4F:B5:7F	11	-50	-100	00534	97	100	002	080	020	WL1
00:0F:3D:F7:37:65	01	-74	-91	00429	99	100	002	084	015	WL1
00:60:B3:16:B3:3E	07	-91	-99	34410	20	100	002	097	002	WL1
00:60:B3:16:68:5C	01	-86	-90	46984	32	100	002	000	000	WL1

(b)

BSSID	ch	Sig	Noi	age	Phr	int	cnt	sct	uct	ESSID
00:0F:3D:E1:03:7E	06	-53	-99	00000	96	100	002	000	000	WL1
00:20:A6:4F:B5:7F	11	-57	-99	01006	12	100	003	078	021	WL1
00:0F:3D:F7:37:65	01	-75	-100	00901	14	100	003	088	011	WL1
00:60:B3:16:B3:3E	07	-91	-100	25866	40	100	002	096	003	WL1
00:60:B3:16:68:5C	01	-85	-91	27537	54	100	002	000	000	WL1

Figure 5–3 Logging messages showing the phase drift phenomenon

另外我們在實驗中也發現到，雖然理論上STA只要和目標基地台一次接觸後，即可得到目標基地台發送beacon的時間表，在我們的實作中，此時間表資訊是以本地端時間的Phase值所代表，如 Figure 5–3 中的Phr欄所示，但是這個Phase值會隨著時間推進而有drift的現象。這是因為STA與不同的基地台間使用的clock長度不一致所致，不過相近的beacon frames間仍然具有前者推算出後者的特性。對於這種phase drift現象，我們採取維護最新Phase information的策略來解決這一項議題。另外，I. Ramani and S. Savage 所提出的SyncScan[13] 中，也有提到基地台需要定期做global timing synchronization的必要，以達到global view timing information的一致性。我們的Non-AP-Aligned SyncScan不會有這樣子的麻煩，若失去timing synchronization馬上利用Unicast Probe的動作重新synchronized即可。

另外，我們亦針對STA在進行目標基地台持續性量測的同時，統計其所用到Non-AP-Aligned SyncScan以及Unicast Probe的次數比例，如 Figure 5–3 中的sct (Syncscan CounT) 及uct (Unicast probe CounT) 所示。由此可以看出在我們的實作裡，有將近八成左右目標基地台的量測，都是使用不會產生額外traffic的Non-AP-Aligned SyncScan量測方法。

5.3 網路傳輸 jitter 的影響

為了模擬 VoIP 這一類 real-time service 以及評估我們所提出的目標基地台搜索與量測機制所用到的 Non-AP-Aligned SyncScan 對此類網路流量所造成的 jitter 效應，我們在每秒執行二次 Non-AP-Aligned SyncScan 的 STA 與在 wired 端固定的 Server 間持續傳送與一般 VoIP 網路封包等同大小的 UDP traffic。

在上述的scenario下，我們藉由改變來源端UDP traffic產生的周期 (分別以 20 ms 與 50 ms傳送)，並且在接收端統計並記錄所收到的封包，其到達終點端的間隔時間 (Inter-Arrival Time, IAT) 的分佈。我們把IAT = 20 ms與IAT = 50 ms這兩種情況產生終端封包間隔時間的結果，以累積比例函數的方式製圖成 Figure 5–4 與 Figure 5–5供讀者參考。

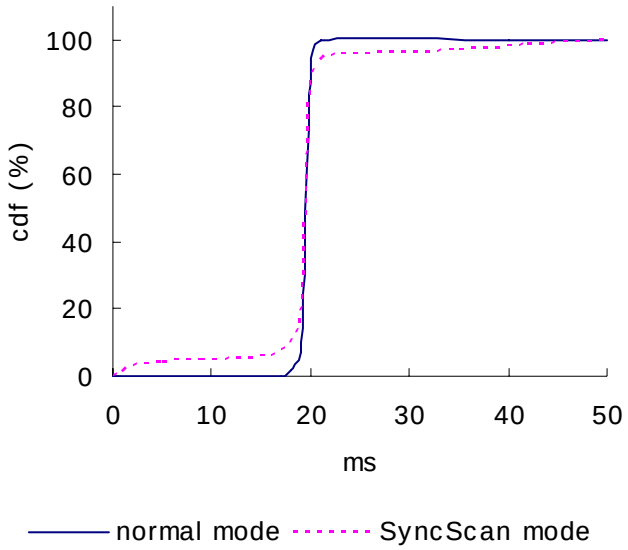


Figure 5–4 Cdf of packets with IAT = 20 ms

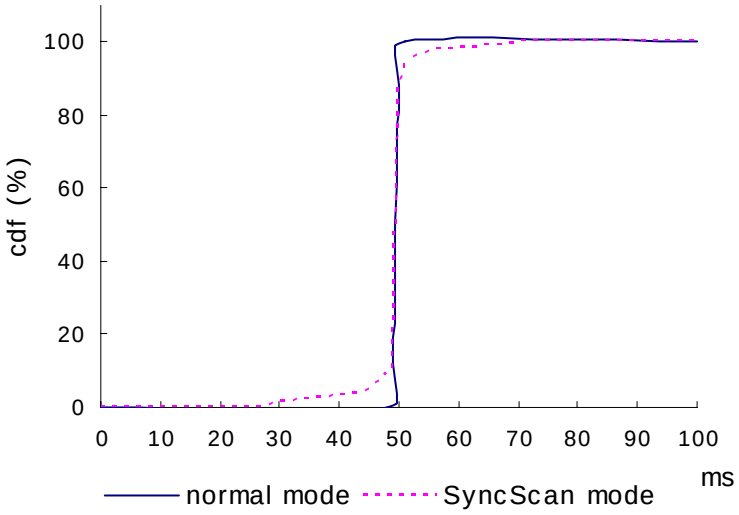


Figure 5–5 Cdf of packets with IAT = 50 ms

Figure 5–4 與 Figure 5–5 中的藍色實線部分，代表STA於不執行Non-AP-Aligned SyncScan，終端接收到的封包到達間隔時間的結果；而粉紅色虛線代表的是STA在有執行每秒二次Non-AP-Aligned SyncScan情況下，在終端接收到的封包到達間隔時間的結果。

從 Figure 5–4 與 Figure 5–5 的結果可以看出，STA執行Non-AP-Aligned SyncScan，會對正常傳送的封包增加些微的jitter效應。增加的部分主要是發生在STA freeze正常資料傳送，切換channel進行目標基地台量測的動作與正常封包傳送撞期的情況。這時候封包會被queue在STA或是基地台 (因traffic的流向而有不同)，使得封包在傳輸過程中停留較久的時間，造成額外的jitter效應。比較有趣的是，我們在 5.1 提到每一個Non-AP-Aligned SyncScan動作需要約 24 ms的時間，也著實反應在我們量測的數據中。Figure 5–4 與 Figure 5–5 中因Non-AP-Aligned SyncScan而額外增加或減少IAT的instance，主要都分佈在約正負 20 ms以內。

5.4 頻寬使用的影響

我們在這一節則是針對Non-AP-Aligned SyncScan對於正常封包傳送造成bandwidth使用的影響進行評估。評估的方式，是以在STA與wired 端的server間使用TCP傳輸資料，記錄每一秒鐘達到的瞬間傳輸速度來進行。我們一共進行 5 種cases的評估，以STA每秒鐘進行 0, 1, 2, 3, 4 次Non-AP-Aligned SyncScan做為分別。其所量測到的TCP瞬間傳輸速率如 Figure 5–6 所示。

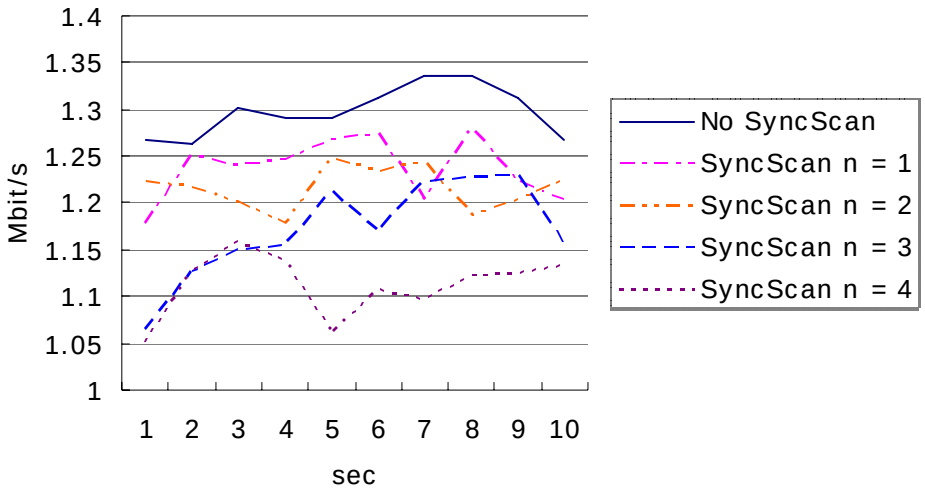


Figure 5–6 Instantaneous bandwidth binned at 1 sec interval

從 Figure 5–6 的結果可以看出，TCP傳輸能使用到的頻寬，隨著每秒鐘進行的Non-AP-Aligned SyncScan的次數增加而微幅下降，這是因為每秒鐘執行Non-AP-Aligned SyncScan的次數愈多，STA能用在正常傳輸封包的時間也會隨之減少。由於每秒鐘執行Non-AP-Aligned SyncScan的次數是根據STA所得到的local topology的端點個數而定；在同一平面下，相鄰的基地台個數並不會無限制的擴大，即使在無地形障礙的環境下，最多也只需 6 個基地台即可將各個方向行進可能涵蓋住 (參考Figure 2–1)。

我們認為藉由犧牲些微的頻寬使用而換得在換手前足夠的目標基地台量測資訊是值得並且可行的。再者在我們提出的無縫換手設計中，STA 只有在服務基地台的訊號強度低於某預先設定的臨界值時才會開始進行目標基地台的量測動作，在某種程度上也兼顧到了頻寬使用與目標基地台量測需求這兩者間的平衡。

5.5 對 Handover Decision 的影響

這一部分的評估，我們是以 STA 換手前後量得 Serving AP 與 Target AP 的訊號強度的時間進展，以及 STA 換手決策的時間點做為比較。比較的兩個 cases，一為普遍實作的換手設計以及我們實作的無縫換手設計。前者是使用 Prism 2 chipset 無線網路卡並且 firmware 是運作在 STA 模式下，使用 HostAP drive 提供的 STA driver 的情況；後者則是 Prism 2 chipset 無線網路卡運作在 HostAP 模式，並且使用我們所實作的 STA driver。

我們量測統計的結果和I. Ramani and S. Savage在其SyncScan[13] 這篇paper所做相同的評估得到的結果相近，如 Figure 2–14 所示。由於我們所設計的無縫換手機制，在換手前對於換手前後的基地台使用Non-AP-Aligned SyncScan的方式進行持續性的量測，因此STA可以有效利用 2.2 所提及的Treshold-based Handover Algorithm執行換手的決策。目前普遍實作的換手設計，在換手前沒有量測目標基地台的訊號強度，直到與服務基地台的通訊受到威脅才開始換手的動作，其換手決策的時間點較為延後，並且換手前經歷的服務基地台訊號強度比我們所實作的無縫換手所經歷到的來得低，封包傳送也較容易出錯。


E
S
A
1896

第六章　結論與未來工作

6.1 結論

我們的研究結果顯示，目前市面上常見實作的 WLAN 換手機制的效能，尚未能達到 VoIP 這類即時性應用服務的需求的主要原因，在於**「目標基地台搜索與量測」機制和「換手程序」互動時機上的不搭配**，尤其在相同無線網路下的基地台與基地台間換手過程特別明顯，足以嚴重到影響 VoIP 的服務品質。

我們所提出的改善之道，便是針對「目標基地台搜索與量測」機制的重新設計，對於**「初次進入的連結程序」**與**「相同無線網路下基地台間的換手程序」**這兩種情況做出區別。對於前者的情況，市面上一般實作的換手機制足夠使用，即先進行完整的搜索，再選出最適合的基地台進行連結；至於後者，我們的設計讓STA至少可以利用 Background Further Observed Scanning 的方式，先找出可能的目標基地台集合，即服務基地台的Local Topology，接著利用 Non-AP-Aligned SyncScan 的機制持續對這些目標基地台進行量測與維護。

簡言之，我們設計並實作了一套適合 IEEE 802.11 無線網路下使用的無縫換手機制，藉由改善「目標基地台搜索與量測」的機制，縮短 STA 換手過程所花的時間，進而改善 STA 因換手的過程無法進行通訊時間過長所造成 real-time 應用程式服務品質下降的問題。

我們設計與實作的「目標基地台搜索與量測」機制，可以提供 STA 持續性的目標基地台量測資訊，使得換手決策方塊能夠有更充足的資訊進行換手時機的決策，達到更好品質的換手過程。

我們所設計實作的無縫換手機制，完全符合 IEEE 802.11 的標準，不需要額外的資源部署，只需要對 STA 端的 device driver 進行小幅度修改，這對已經佈署使用中的無線網路是比較可行的。經過實測評估，我們設計的無縫換手機制在換手過程中的花費不多並且由於是分散進行，不會影響正常資料的傳送與接收。

當使用者無法在相同的無線網路下經由換手動作繼續使用時，我們所設計實作的無縫換手機制能夠提早通知 STA 的上層的 Mobility Management Entity 早一步進行 Vertical Handover 或相關的準備。

6.2 未來工作

我們所設計實作的 IEEE 802.11 無線網路下的無縫換手機制，目前僅在沒有加強安全協定及沒有服務品質保證的環境下進行評估。在使用加強安全協定與具服務品質保證的 IEEE 802.11 無線網路下，STA 在換手過程需要額外執行初始驗證金鑰 PMK 與 QoS 資源部署的動作，這些動作都會進一步造成額外的換手延遲。

IEEE基於有快速換手需求的應用服務，提出的IEEE 802.11r基地台快速換手程序標準[6] 中所提供的Pre-Reservation Mechanism，可以減少換手程序commit operations所需的messages exchange到最理想的情況下，僅需兩個messages exchange，然而這個快速換手程序，需要STA在執行commit operations前透過DS執行相關的預備部署動作，才有辦法達成。

我們所設計實作的 IEEE 802.11 無線網路下的無縫換手機制的架構，已經具備提供這些預備動作背後所需要的事件驅動的介面；針對這一部分的議題，我們還可以進一步進行實際系統運作相關的探索與評估。

參 考 文 獻

[1] IEEE 802.11 WG, Part 11, "IEEE Standard 802.11-1999: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification," 1999.

[2] IEEE 802.11 WG, Part 11, "IEEE Standard 802.11F-2003: IEEE Trial-Use Recommended Practice for Multi-Vendor Access Point Interoperability via an Inter-Access Point Protocol across Distribution Systems Supporting IEEE 802.11 Operation," 2003.

[3] IEEE 802.11 WG, Part 11, "IEEE Standard 802.11i™-2004: Amendment 6: Medium Access Control (MAC) Security Enhancements," July 2004.

[4] IEEE 802.11 WG, Part 11, "IEEE Standard 802.11e™-2005: Amendment 7: Medium Access Control (MAC) Quality of Service (QoS) Enhancements," July 2005.

[5] IEEE 802.11 WG, Part 11, "IEEE Draft Standard 802.11k/D3.0: Amendment 9: Radio Resource Measurement," Oct. 2005.

[6] IEEE 802.11 WG, Part 11, "IEEE Draft Standard 802.11r/D1.0: Amendment 10: Fast BSS Transition," Nov. 2005.

[7] A. Mishra, M. Shin, and W. A. Arbaugh, "An Empirical Analysis of the IEEE 802.11 MAC Layer Handoff Process," ACM SIGCOMM Computer Communications Review, Vol. 33, No. 2, pp. 93-102, Apr. 2003.

[8] H. Velayos and G. Karlsson, "Techniques to Reduce IEEE 802.11b MAC Layer Handover Time," KunglTekniska Hogskolen, Stockholm, Sweden, Tech. Rep. TRITA-IMIT-LCN R 03:02, ISSN 1651-7717, ISRN KTH/IMIT/LCN/R-03/02－SE, Apr. 2003.

[9] M. Shin, A. Mishra, and W. A. Arbaugh, "Improving the Latency of 802.11 Handoffs using Neighbor Graphs," in Proceedings of the ACM MobiSys Conference, June 2004.

[10] K. Kwon and C. Lee, "A Fast Handoff Algorithm using Intelligent Channel Scan for IEEE 802.11 WLANs," The 6th International Conference on Advanced Communication Technology, Vol. 1, pp. 46-50, 2004.

[11] S. Shin, A. S. Rawat, and H. Schulzrinne, "Reducing MAC Layer Handoff Latency in IEEE 802.11 Wireless LANs," in Proceeding of ACM MobiWac 2004, Oct. 2004.

[12] N. Mustafa, W. Mahmood, A. Chaudhry, and M. Ibrahim, "Pre-Scanning and Dynamic Caching for Fast Handoff at MAC Layer in IEEE 802.11 Wireless LANs," Mobile Adhoc and Sensor Systems (MASS) Conference, IEEE International Conference on, Nov. 2005.

[13] I. Ramani and S. Savage, "SyncScan: Practical Fast Handoff for 802.11 Infrastructure Networks," IEEE INFOCOM, Mar. 2005.

[14] C. C. Tseng, K. H. Chi, M. D. Hsieh, and H. H. Chang, "Location-based Fast Handoff for 802.11 Networks," IEEE Communication Letters, Vol. 9, No. 4, pp. 304-306, Apr. 2005.

[15] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based User Location and Tracking System," IEEE INFOCOM, Mar. 2000.

[16] A. Mishra, M. Shin, and W. A. Arbaugh, "Context Caching using Neighbor Graphs for Fast Handoffs in a Wireless Network," IEEE INFOCOM, Mar. 2004.

[17] A. Mishra, M. Shin, N. L. Petroni, Jr., T. C. Clancy, and W. A. Arbaugh, "Proactive Key Distribution using Neighbor Graphs," IEEE Wireless Communications Magazine, Vol. 11, pp. 26-36, Feb. 2004.

[18] H. H. Duong, A. Dadej, and S. Gordon, "Proactive Context Transfer in WLAN-based Access Networks," Proceedings of the Second International Workshop on Wireless Mobile Applications and Service on WLAN Hotspots (WMASH) 2004, Oct. 2004.

[19] J. Malinen, "HostAP Driver for Intersil Prism 2/2.5/3," http://hostap.epitest.fi/, 2006.

[20] "MadWifi (*Multiband Atheros Driver for Wireless Fidelity)* Project – a Linux kernel driver for Wireless LAN chipsets from Atheros," http://madwifi.org/, 2006.

[21] J. Tourrilhes, "Linux Wireless Extensions and Tools," http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html

[22] M. Gast, "802.11 Wireless Networks: The Definitive Guide, 2nd Edition," O'Reilly Media, Inc., Apr. 2005.

[23] P. Roshan and J. Leary, "802.11 Wireless LAN Fundamentals: A Practical guide to understanding, designing, and operating 802.11 WLANs," Cisco Press, Dec. 2003.

[24] C. Benvenuti, "Understanding Linux Network Internals," O'Reilly Media, Inc., Dec. 2005.

[25] K. Wehrle, F. Pählke, H. Ritter, D. Müller, and M. Bechler, "The Linux® Networking Architecture: Design and Implementation of Network Protocols in the Linux Kernel," Prentice Hall, Aug. 2004.

附錄A　Fast BSS Transition

IEEE在完成IEEE 802.11i增強安全協定標準[3] 與IEEE 802.11e服務品質保證規格標準[4] 後，針對需要快速換手需求的應用（如VoIP），草擬製定IEEE 802.11r基地台快速換手程序標準[6]。會有這樣子的狀況產生，主要的原因是IEEE 802.11i增強安全協定標準[3] 與IEEE 802.11e服務品質保證規格標準[4] 的制定，是以盡量保留原有架構的模式，而以類似patch的方式進行標準的修正改良，並且由於彼此針對各自的主題獨立進行，導致了換手程序中相關的messages exchange沒有最佳化的情況產生。

對IEEE 802.11i增強安全協定標準[3] 而言，原始link layer的安全協定所使用的messages exchange程序是保留下來不動的，但使用open system沒有認證效果的link layer authentication。而真正進行authentication的部分，則採取接下來新增的 4-Way Handshake，「**驗證**」經由IEEE 802.1X以User為基礎的認證程序後產生的PMK。對IEEE 802.11e服務品質保證規格標準[4] 而言，QoS negotiation過程使用的messages exchange，僅需要建構在STA和基地台握有PMK安全金鑰的前提下即可，但由於這些QoS negotiation使用的messages exchange在現有的標準規格裡，必須由STA和目標基地台使用management action frame直接傳送，因此STA必須等到連結上目標基地台且完成PMK佈署才能開始進行negotiation的動作。

就筆者的眼光看來，IEEE 802.11r基地台快速換手程序標準[6] 的內涵乃針對IEEE 802.11i增強安全協定標準[3] 與IEEE 802.11e服務品質保證規格標準[4] 後STA所需進行的一連串messages exchange，在不失安全層級的前提下，做訊息傳遞最佳化的工作。主要的設計有四項：

1. 將原先 STA 對每個基地台都至少必須執行一次的 802.1X 認證程序，縮減為 STA 對同一個 Mobility Domain 下的基地台群，只需執行一次 802.1X 認證程序即可。

2. 擴充link layer authentication/(re-)association messages exchange、新增Fast Transition Action Frames，以及產生新的驗證messages exchange，稱為Fast Transition Authentication Sequences[22]。

3. 支援 STA 查詢目標基地台的相關資源，以提供換手決策所需的資訊。

4. 支援 STA 在 Commit Phase 之前，預先佈署必要的安全協定及 QoS 需求使用的 context。

A.1 Three-Level Key Hierarchy

為了將原先STA在每個基地台至少必須執行一次的 802.1X認證，縮減成STA對同一個Mobility Domain下的基地台群只需執行一次 802.1X認證，IEEE 802.11r基地台快速換手程序標準[6] 設計了相對於IEEE 802.11i增強安全協定標準[3] 二階層的key hierarchy架構的三階層key hierarchy架構。相關key的產生者和保有者及傳遞過程如 Figure A–1 所示：

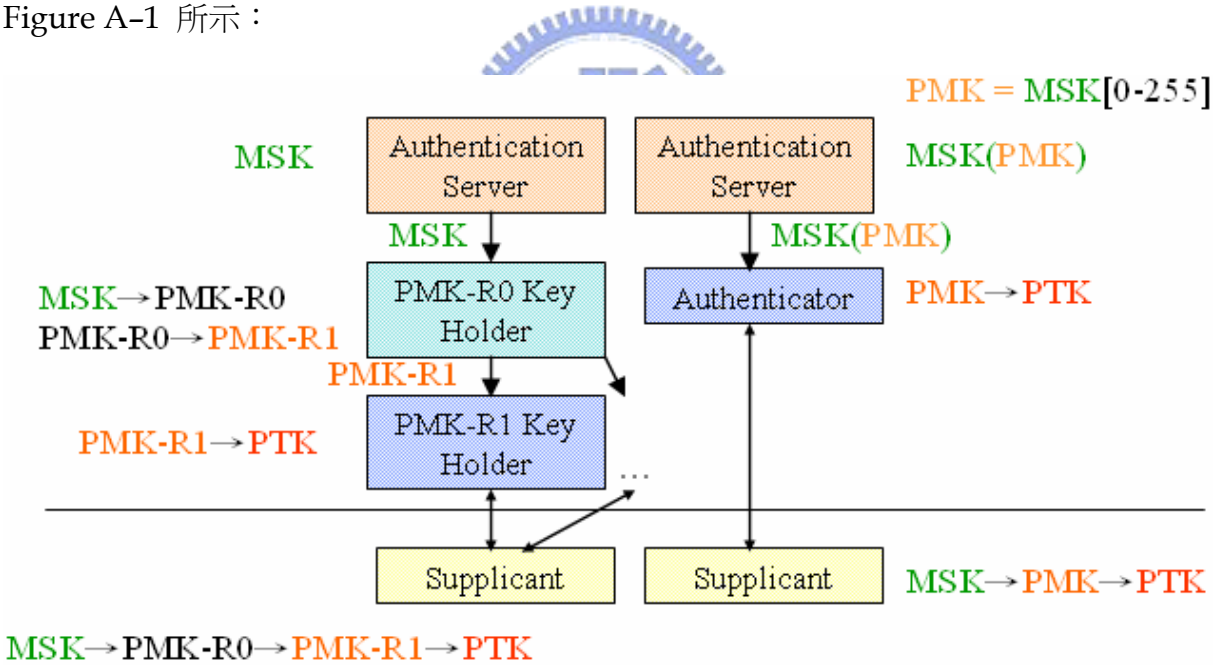


Figure A–1 Fast Transition key hierarchy and 802.11i key hierarchy

[22] FT Auth. Seq. 和 4-Way Handshake 一樣是四個訊息的交換程序，用來驗證彼此的 PMK 安全金鑰；不同的是，FT Auth. Seq. 是由 STA 而非基地台驅動，並且 FT Auth. Seq. 這四個訊息可以更動在 MAC Sublayer 的 802.11 protocol state 的狀態。簡單的說，FT Auth. Seq. 的其中一項動作，就是合併作用在 MAC Sublayer 及 RSNA Key Management Entity 的訊息交換，達到訊息傳遞最佳化的目標。 (參考 Figure 2–16 802.11 basic reference model)

IEEE 802.11i增強安全協定標準[3] 下的key hierarchy主要有三類角色。Authentication Server和Supplicant是執行 802.1X以User為基礎的兩端個體，而Authenticator和Supplicant則是執行 4-Way Handshake驗證PMK的兩端個體。Authenticator 的 PMK 金鑰是從 Authentication Server 經由安全網路通道在Authentication Server和Supplicant完成 802.1X認證程序後取得。

IEEE 802.11r基地台快速換手程序標準[6] 下key hierarchy則分成四類角色。Authentication Server和Supplicant仍然是執行 802.1X以User為基礎的兩端個體，而PMK-R1 Key Holder和Supplicant則是執行 4-Way Handshake或Fast Transition Authentication Sequence驗證PMK-R1 的兩端個體。PMK-R1 Key Holder的PMK-R1是從同Mobility Domain下的PMK-R0 Key Holder經由安全網路通道取得，取得的時機只要在PMK-R1 Key Holder需要使用到PMK-R1 之前即可。PMK-R0 Key Holder在Authentication Server和Supplicant完成 802.1X認證程序後，可以從Authentication Server經安全網路通道得到PMK-R0，並且負責在接下來PMK-R0 的有效時間內，依據在同Mobility Domain下不同的PMK-R1 Key Holder代號，由PMK-R0 再產生相對應的PMK-R1。

在基地台快速換手程序下，我們可以歸納下列結論：

1. 同 Mobility Domain 下，STA 只需要執行一次完整的 802.1X 認證程序。
2. PMK-R1 Key Holder 就是原始的 Authenticator 角色。
3. PMK在基地台快速換手程序下，分成PMK-R0 及PMK-R1 兩階層，分別由PMK-R0 Key Holder及PMK-R1 Key Holder擁有。PMK-R0 Key Holder負責所有PMK-R1 的產生，因此也取代了IEEE 802.11i增強安全協定標準[3] 下由Authentication Server產生PMK供基地台進行進一步驗證的角色。

A.2 First Contact

當STA第一次進入無線網路的某一個Mobility Domain的時候，STA必須按照原始IEEE 802.11i增強安全協定標準[3] 的安全協定程序進行如 Figure A–2 中一連串的messages exchange。其中包括legacy authentication、association、802.1X認證程序、及由Authenticator發動的 4-Way Handshake (紅線實線方框) 等。因此STA在換手程序中的Commit Phase (藍色虛線方框) 所需進行的messages exchange和IEEE 802.11i增強安全協定標準[3] 下的安全協定程序情況一樣。

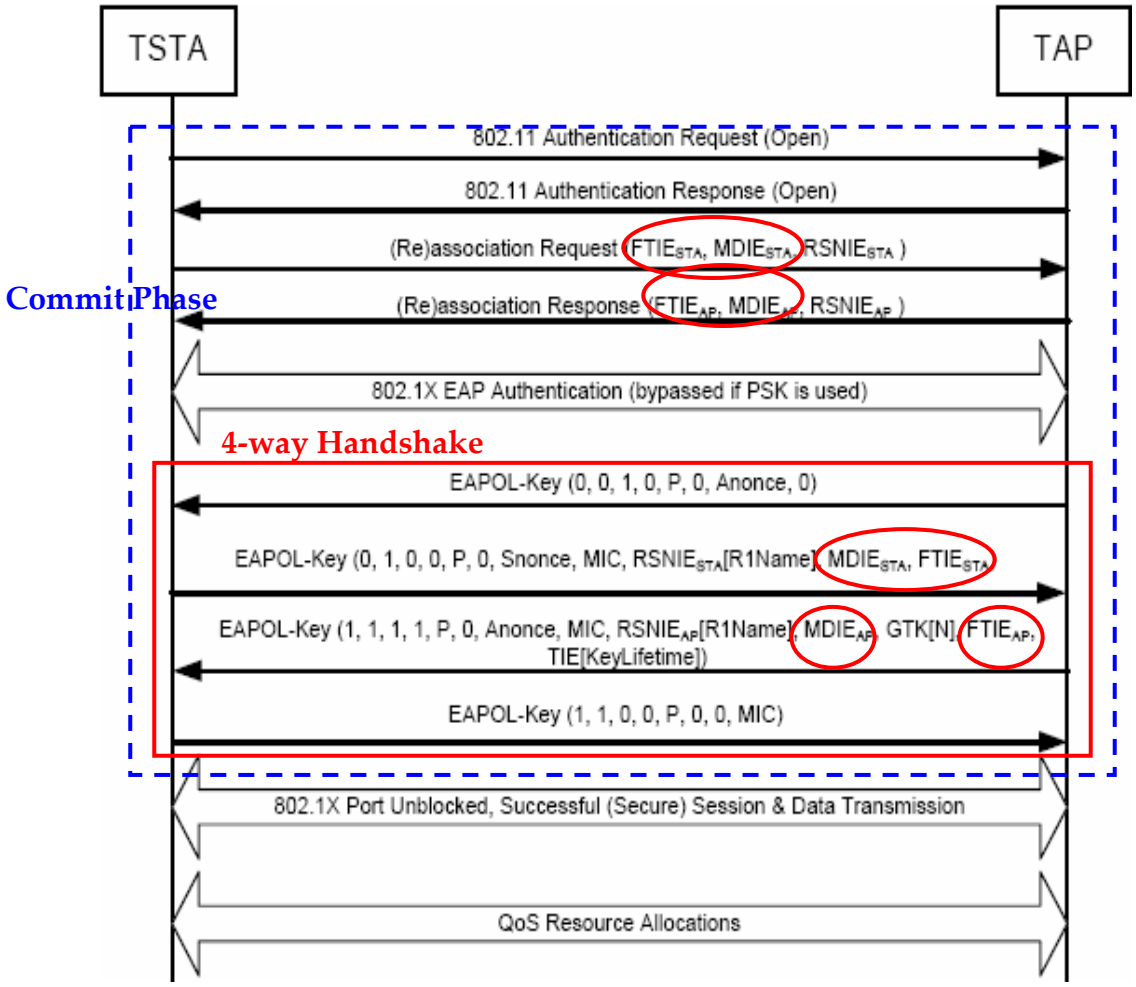


Figure A–2 First Contact sequences of messages exchange

(Source from [6])

幾個不同的地方有：在(Re-)association exchanges 和 4-Way Handshake 的 messages exchange 內，需要加上 STA 使用基地台快速換手程序的選項以資 Authenticator 識別。此時 Authenticator 不但成為 PMK-R0 Key Holder，同時也是 PMK-R1 Key Holder；而 4-Way Handshake 驗證的 PMK，使用的是基地台快速換手程序中的 PMK-R1。

A.3 Fast Transition Authentication Sequence

接下來在同一 Mobility Domain 下，STA 不需要再經由 802.1X 認證程序對目標基地台佈署 PMK；除此之外，換手程序中的「驗證動作」可以更有效率，並且可以經由 Wireless Medium 或是經由 Distribution System 預先佈署。

在IEEE 802.11r基地台快速換手程序標準[6]，稱上述的驗證程序為Fast Transition Authentication Sequences。和IEEE 802.11i增強安全協定標準[3] 定義的 4-Way Handshake相似，有 4 個messages exchange，然而是由STA端發起。這 4 個messages exchange依不同的情況 (見A.4 Base Mechanism、A.5 Pre-Reservation Mechanism) 可以由不同種類的message frames乘載。IEEE 802.11r基地台快速換手程序標準[6] 稱此 4 快速換手驗證訊息個別的名稱為：

1. **Fast Transition Request**: (從 STA→目標基地台)

   帶有 SNonce、RSNIE、FTIE、MDIE 等。其中 FTIE 內可以讓目標基地台 (PMK-R1 Key Holder) 得知 PMK-R0 Key Holder 的身份，如果目標基地台沒有 PMK-R1，需要向 PMK-R0 Key Holder 索取。此 message 沒有 MIC 保護。

2. **Fast Transition Response**: (從目標基地台→STA)

   帶有 ANonce、RSNIE、FTIE、MDIE、可能帶有 reassociation deadline Time Interval IE 及 key lifetime Time Interval IE。Reassociation deadline Time Interval IE 的目的是告知執行預先資源佈署的 STA 要在多久時間內完成 Re-association 的動作；另一種 key lifetime Time Interval IE 是告知金鑰的使用期限。此 message 沒有 MIC 保護。

3. **Fast Transition Confirm**: (從 STA→目標基地台)

   帶有 SNonce、RSNIE、FTIE、MDIE、RICIE。RICIE 可供 STA 攜帶資源佈署資訊 (Resource Information)。此 message 必須使用 MIC 保護。

4. **Fast Transition Ack**: (從目標基地台→STA)

   帶有 ANonce、RSNIE、FTIE、MDIE、可能帶有 reassociation deadline Time Interval IE 及 key lifetime Time Interval IE。Reassociation deadline Time Interval IE 的目的是告知執行預先資源佈署的 STA 要在多久時間內完成 Re-association 的動作；另一種 key lifetime Time Interval IE 是告知金鑰的使用期限。此 message 必須使用 MIC 保護。

A.4 Base Mechanism

IEEE 802.11r基地台快速換手程序標準[6] 提供兩種基地台快速換手程序，第一種為Base Mechanism。Base Mechanism基地台快速換手程序可以是經由Wireless Medium (over the Air)，或是經由DS Medium(over the DS)。Base Mechanism基地台快速換手程序可供

1. 不需要 QoS 佈署的 STA 或是
2. 不需要預先確立 QoS 佈署的 STA 使用。

Fast Transition Authentication Sequences在over the Air模式下，是內嵌於原始換手程序中的Authentication Request/Response、及(Re-)association Request/Response這四個IEEE 802.11 management frame之中；而在over the DS模式下，Fast Transition Request/Response是使用Fast Transition Action Frames經由正在使用中的基地台relay到目標基地台，Fast Transition Confirm/Ack則是內嵌在原始換手程序中的(Re-)association Request/Response這二個IEEE 802.11 management frames之中[23]。

[23] 使用 over the DS 模式的 STA，在切換 channel 與目標基地台進行換手程序時，不需要再傳送 Authentication Request/Response； 原本 802.11 protocol state machine (Figure A–3) 在其完成 Fast Transition Request/Response 即移往 State 2。

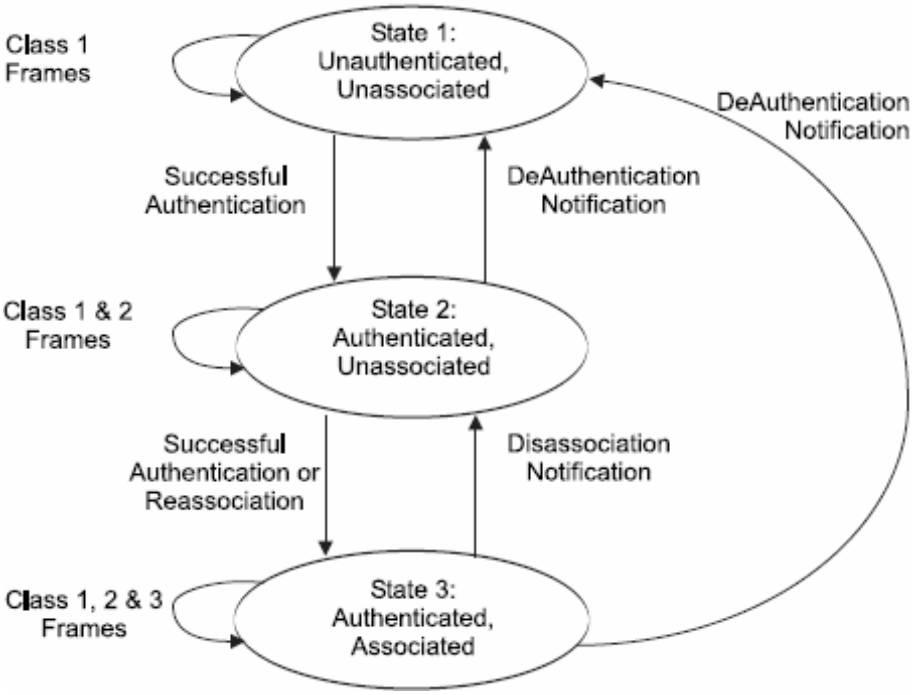


Figure A–3 802.11 protocol state machine

(Source from [1])

A.4.1 Over the Air

Base Mechanism在over the Air模式下，STA將原本換手程序中的Authentication Request/Response、(Re-)association Request/Response和 4-Way Handshake的messages合併。使得 Figure 2–16 中，MAC Sublayer內的 802.11 protocol state machine (Figure A–3) 和4-Way Handshake導致的RSNA state machine (處於 Figure 2–16 中 Station Management Entity內的RSNA Key Management元件之內) 同時進行。以達到縮減messages exchange的目標。

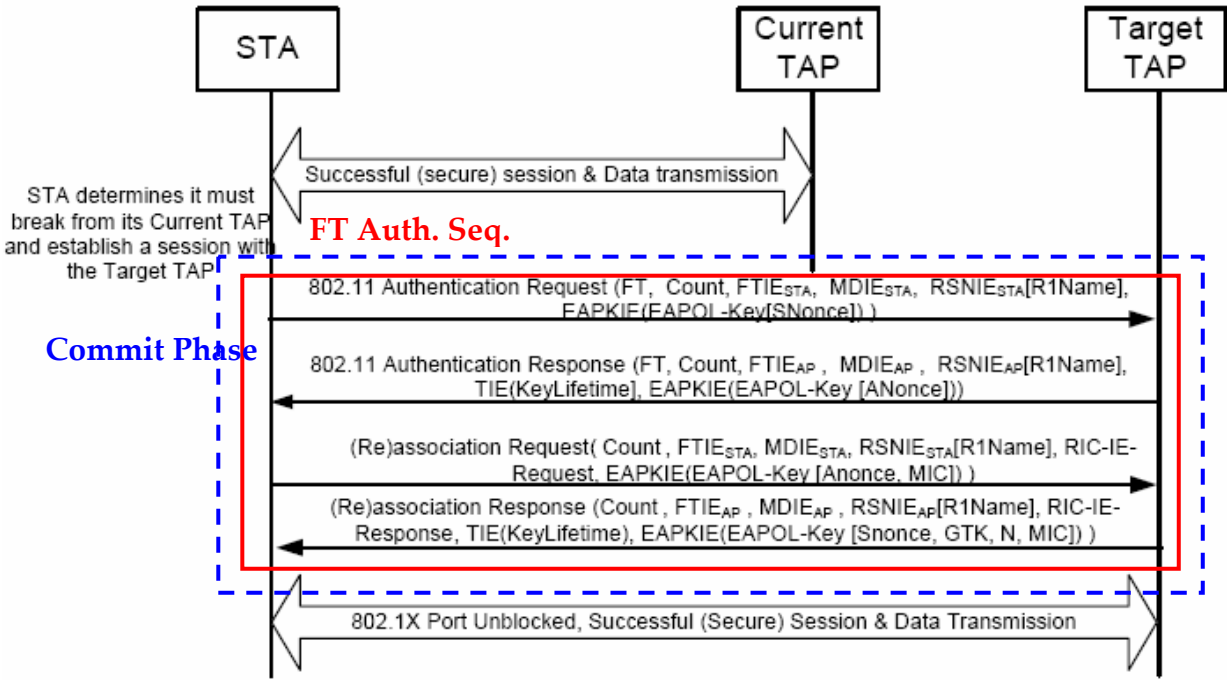


Figure A–4 Messages flows in Base Mechanism over the air

(Source from [6])

Base Mechanism在over the Air模式下，除了縮減Commit Phase所需的messages exchange，針對需要進行QoS佈署的STA，也提供了利用Fast Transition Confirm/Ack這兩個messages exchange可提供的一次溝通transaction的機會 (見 Figure A–4 中的RIC-IE-Request/Response IE)。如此對於需要進行QoS佈署的STA又進一步減少了二個QoS佈署的訊息量。

值得一提的是，Fast Transition Request和Response訊息之間的時間長度，可能會因目標基地台尚未擁有PMK-R1 而增加[24]。

[24] 這是因為目標基地台需要向 PMK-R0 Key Holder 索取屬於目標基地台的 PMK-R1。PMK-R1 Key Holder 向 PMK-R0 Key Holder 取得 PMK-R1 的機制，目前草擬標準沒有相關規定，不過可以使用類似 IEEE 802.11F [2] 制定的 IAPP CACHE-notify 機制。

A.4.2 Over the DS

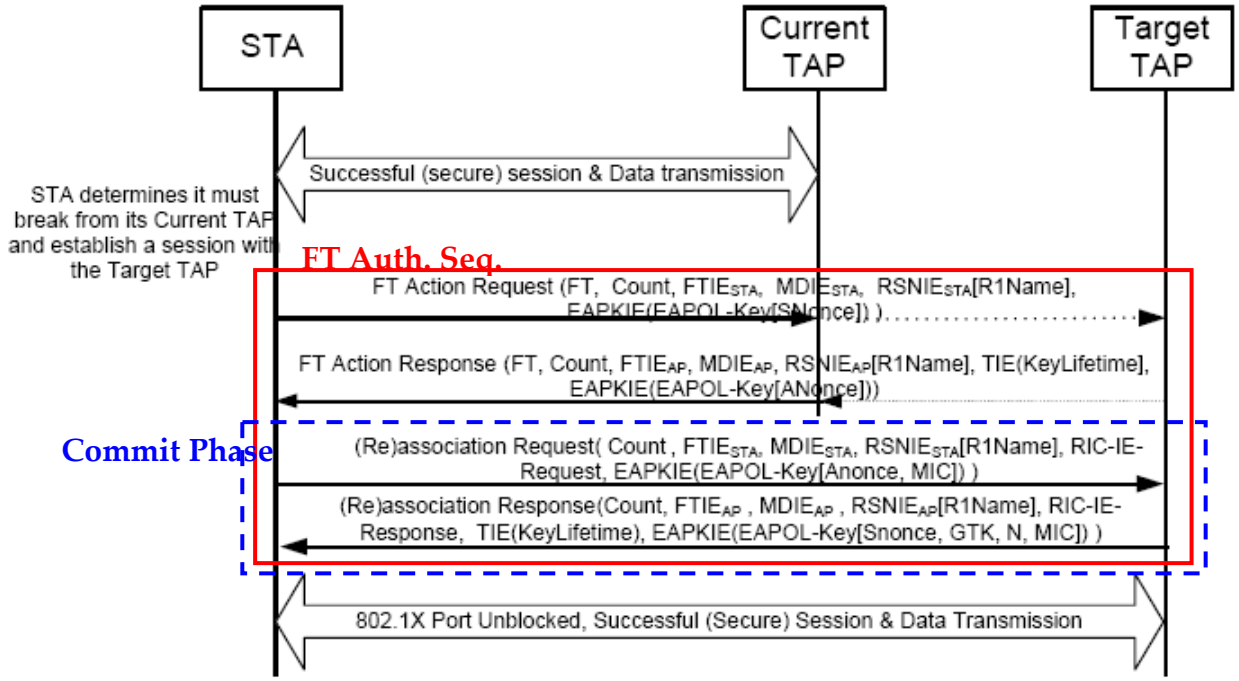


Figure A–5 Messages flows in Base Mechanism over the DS

(Source from [6])

Base Mechanism在over the DS模式下，STA將 Figure 2–5 中原本換手程序Commit Phase中的Authentication request/Response省略，其在 Figure 2–16 中MAC Sublayer的protocol state machine transition (Figure A–3) 從State 1 到State 2 的轉移，是由經DS發送的Fast Transition Request/Response驅動的、而(Re-)association Request/Response則用來當作Fast Transition Confirm/Ack。

Base Mechanism在over the DS模式下，更進一步縮減Commit Phase messages exchange數量，Fast Transition Authentication Sequences (紅色實線方框) 的Fast Transition Request/Response在尚未切換channel前即可經由服務基地轉送。Commit Phase (藍色虛線方框) 因此只剩下導致 802.11 protocol state machine (Figure A–3) 從State 2 移動到State 3 的兩個 messages exchange: 即 (Re-)association Request/Response。

Base Mechanism在over the DS模式下，針對需要進行QoS佈署的STA，也提供了經由Fast Transition Confirm/Ack進行一次溝通transaction的機會 (Figure A–5 中的RIC-IE-Request/Response IE)。如此對於需要進行QoS佈署的STA又進一步減少了二個QoS佈署的訊息量。

A.5 Pre-Reservation Mechanism

IEEE 802.11r基地台快速換手程序標準[6] 提供的第二種程序為Pre-Reservation Mechanism。Pre-Reservation基地台快速換手程序可以是經由Wireless Medium (over the Air)，或是經由DS Medium (over the DS)。Pre-Reservation基地台快速換手程序適合需要預先確立QoS資源佈署的STA使用。

Fast Transition Authentication Sequences在over the Air模式下，是內嵌於原始換手程序中的Authentication Request/Response/Confirm/Ack這四個IEEE 802.11 management frames之中；而在over the DS模式下，Fast Transition Request/Response/ Confirm/Ack是使用Fast Transition Action Frames經由正在使用中的基地台relay到目標基地台。無論是Over the Air模式或是Over the DS模式，STA連結基地台最後的commit operations都需要交換類似Fast Transition Confirm/Ack帶有PMK-R1 驗證訊息及之前預先確立QoS佈署相關訊息[25] 的(Re-)association Request/Response。

[25] STA 可在 Commit Operations (藍色虛線方框) 完成前，更改之前已經預先佈署的服務品質保證的規格，例如 STA 決定降低需求，可以利用 (Re-)association Request/Response 的機會進行更改。

A.5.1 Over the Air

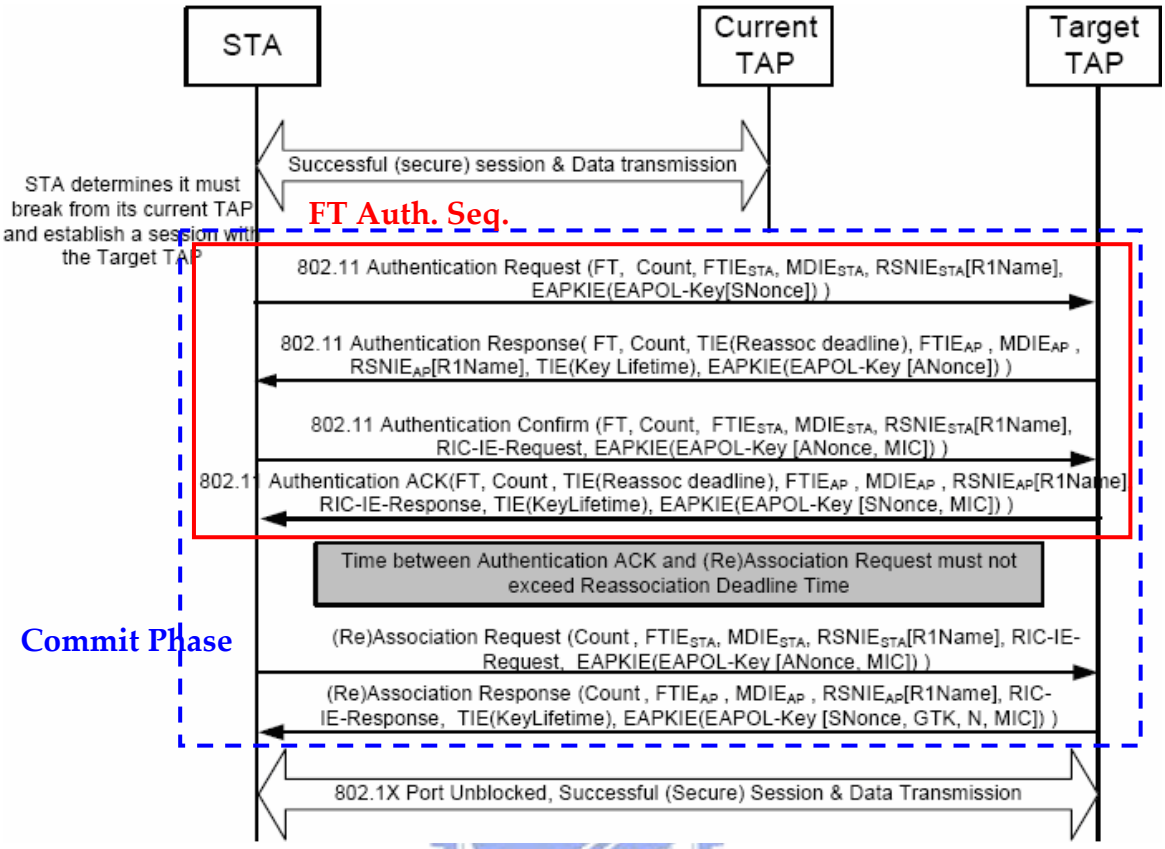


Figure A–6 Messages flows in Pre-Reservation Mechanism over the air

(Source from [6])

Pre-Reservation Mechanism在Over the Air模式下，STA使用原本換手程序中的Authentication Request/Response/Confirm/Ack[26] messages exchange和 4-Way Handshake messages exchanges合併。使得在 Figure 2–16 中MAC Sublayer的protocol state machine (Figure A–3) 和4-Way Handshake導致的RSNA state machine (處於Figure 2–16 中Station Management Entity內的RSNA Key Management元件)同時進行。以達到縮減messages exchange的目標。

[26] Authentication Request/Response/Confirm/Ack 四個 Management frames 的 link layer 認證 messages exchange 也使用在 WEP 認證；事實上，IEEE 802.11r 基地台快速換手程序標準[6] 針對 Fast Transition，在原本Open System和WEP這兩類的link layer認證種類之外，新增了第三種Fast Transition link layer 認證種類。

Pre-Reservation Mechanism在over the Air模式下，Commit Phase (藍色虛線方框)範圍，包含STA進行預先QoS佈署的Fast Transition Authentication Sequences (紅色實線方框)，也包含了最後真正導致 802.11 protocol state machine (Figure A–3) 從State 2 移動到State 3、帶有PMK-R1 驗證訊息及之前預先確立QoS佈署相關訊息的(Re-)association Request/Response。

STA在Authentication Confirm/Ack、(Re-)association Request/Response可以進行QoS佈署建立與修改 (見 Figure A–6 中的RIC-IE-Request/Response IE)。事實上，在Fast Transition Authentication Sequences (紅色實線方框) 和(Re-)association Request/Response之間，STA都可以和目標基地台進行QoS佈署規格再修改的動作。

Pre-Reservation Mechanism在over the Air模式下將Commit Phase (藍色虛線方框) 切成前半部分Fast Transition Authentication Sequences (紅色實線方框) 與後半部真正導致 802.11 protocol state machine (Figure A–3) 從State 2 移動到State 3 的(Re-)association Request/Response，可以讓STA在Commit Phase (藍色虛線方框) 前半部分，即得知目標基地台符不符合STA的QoS的需求，若不符合需求，STA可以不需要進行Commit Phase (藍色虛線方框) 的後半部，STA可立即和其它可能的目標基地台進行換手程序。

值得一提的是，前半部 Fast Transition Authentication Sequences (紅色實線方框)下，第二個、第四個 messages 內的 TIE(Reassociation deadline) 會規定後半部必須完成的時間底線，預先保留的網路資源超過時間立即失效。

A.5.2 Over the DS

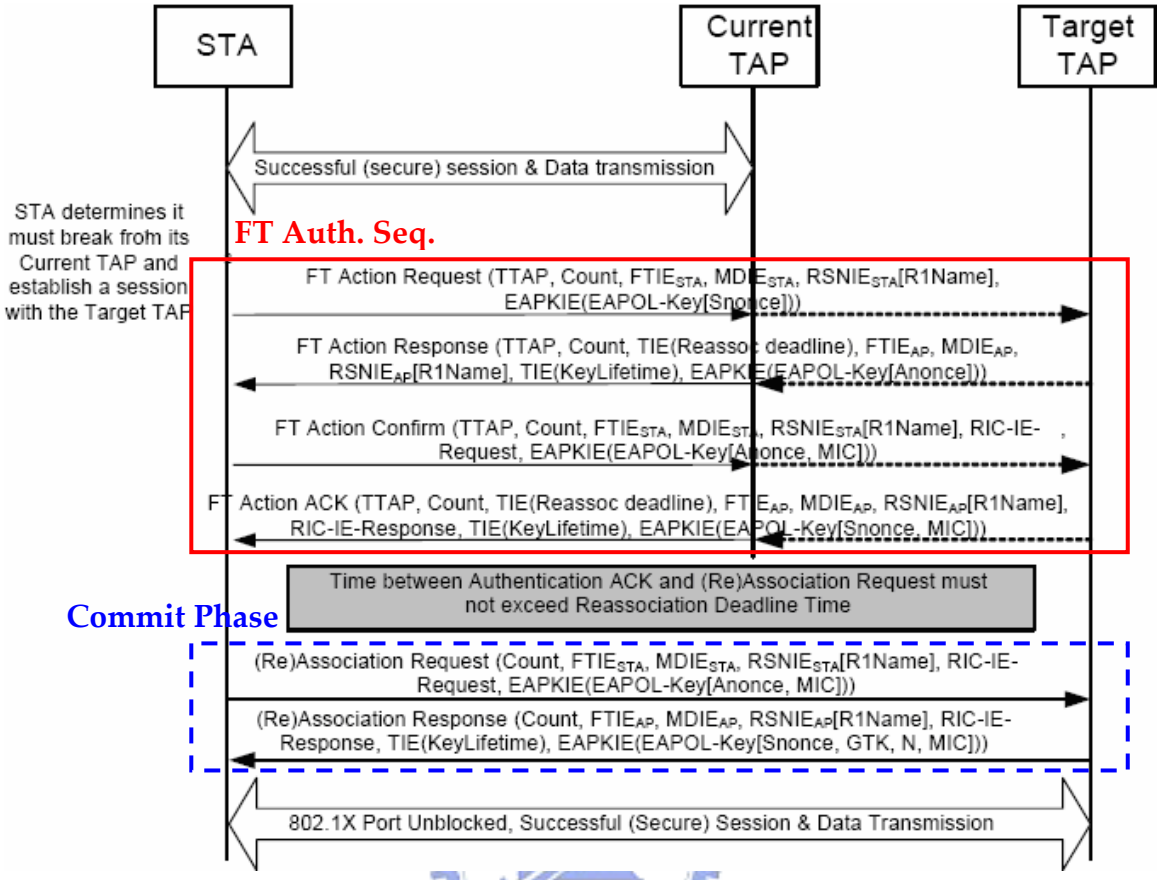


Figure A–7 Messages flows in Pre-Reservation Mechanism over the DS (Source from [6])

Pre-Reservation Mechanism在Over the DS模式下，STA將 Figure 2–5 中原本換手程序的Authentication Request/Response省略，其在 Figure 2–16 中MAC Sublayer的protocol state machine transition (Figure A–3) 從State 1 移往State 2 的移動，是由經DS發送的Fast Transition Request/Response/Confirm/Ack驅動的。

Pre-Reservation Mechanism在Over the DS模式下，Commit Phase (藍色虛線方框) 僅包含最後真正導致 802.11 protocol state machine (Figure A–3) 從State 2 移動到State 3、帶有PMK-R1 驗證訊息及之前預先確立QoS佈署相關訊息的 (Re-)association Request/Response。

STA在Fast Transition Confirm/Ack、和Commit Phase (藍色虛線方框) 內的(Re-)association Request/Response可以進行QoS佈署建立與修改 (見 Figure A–7 中的RIC-IE-Request/Response IE)。事實上，在Fast Transition Authentication Sequences (紅色實線方框) 和 Commit Phase (藍色虛線方框) 內的 (Re-)association Request/Response之間，STA都可以和目標基地台，利用Fast Transition Action Frame透過服務基地台轉達的方式，進行QoS佈署規格再修改的動作。

Pre-Reservation Mechanism在Over the DS模式，其前半部分Fast Transition Authentication Sequences (紅色實線方框) 在STA尚與舊基地台連結時即可執行，不會影響到其它資料封包的傳送、接收，Commit Phase (藍色虛線方框) 只有後半部真正導致802.11 protocol state machine (Figure A–3) 從State 2移動到State 3的兩個messages exchange: (Re-)association Request/Response。

Pre-Reservation Mechanism 在 Over the DS 模式可以讓 STA 在尚與舊基地台連結時，透過經由 DS 與目標基地台執行的 Fast Transition Authentication Sequences (紅色實線方框) 了解目標基地台符不符合 STA 的 QoS 的需求，若不符合，STA 可立即再利用經由 DS 的 Fast Transition Authentication Sequences (紅色實線方框)，對其它可能的目標基地台進行 QoS 的探詢與判斷。

值得一提的是，前半部 Fast Transition Authentication Sequences (紅色實線方框) 下，第二個、第四個 messages 內的 TIE (Reassociation deadline) 會規定後半部必須完成的時間底線，預先保留的網路資源超過時間立即失效。