

# 國立交通大學

電機資訊學院 資訊學程

碩士論文

動態選取基地台策略於 IEEE 802.11 無線網路

Dynamic Access Point Selection Strategies for IEEE 802.11 Wireless  
Networks



研究生：阮騰輝

指導教授：曾煜棋 教授

中華民國九十三年六月

動態選取基地台策略於 IEEE 802.11 無線網路

**Dynamic Access Point Selection Strategies for IEEE 802.11 Wireless  
Networks**

研究生：阮騰輝

Student : Teng-Huei Juan

指導教授：曾煜棋

Advisor : Yu-Chee Tseng



A Thesis

Submitted to Degree Program of Electrical Engineering Computer Science  
College of Electrical Engineering and Computer Science  
National Chiao Tung University  
in Partial Fulfillment of the Requirements  
for the Degree of  
Master of Science  
in  
Computer Science  
June 2004  
Hsinchu, Taiwan, Republic of China

中華民國九十三年六月

# 動態選取基地台策略於 IEEE 802.11 無線網路

學生：阮騰輝

指導教授：曾煜棋 教授

國立交通大學電機資訊學院 資訊學程（研究所）碩士班

## 摘 要

在行動主機的普及化下，帶動整個無線網路的蓬勃發展，尤以熱點(Hot Spot)擴增最為明顯，設置密集度也逐漸增高，使我們隨時隨地皆可使用網際網路上的服務。而使用者在熱點是透過基地台上網來接收或傳送資料，此種無線網路傳輸資料方法，稱為基礎架構模式(Infrastructure mode)，使用者能夠獲得較佳的上網服務品質，不只在於基地台是否能夠提供較佳的頻寬，而且所選的頻道干擾也須要少，因此如何選擇一個提供較好的服務品質的基地台，便是本論文所要研究的問題。

在本篇論文中，我們分析影響網路負載的各項因素，提出且實作兩個動態選擇基地台的策略，一是由基地台廣播負載之策略，另一是探測訊框策略，利用訊框傳送與回覆時間來判斷網路的負載，使得不論在何種狀況，都能夠更精確的選擇服務品質較佳的基地台。此外，並將此兩種策略的實作透過測試來證明可以獲得較佳的效能。

# Dynamic Access Point Selection Strategies for IEEE 802.11 Wireless Networks

Student : Teng-Huei Juan

Advisors : Prof. Yu-Chee Tseng

Degree Program of Electrical Engineering Computer Science  
National Chiao Tung University

## ABSTRACT

Wireless communications have advanced rapidly in recent years. In the hot-spot areas, we have seen dense deployment of access points. Because of that we can get internet service everywhere. Users can connect to networks and send/receive easily. This transmission method is called infrastructure mode. A user will prefer to connect to an AP with higher bandwidth and less interference. This thesis is to address how a wireless station chooses an AP that can provide better service.

In the thesis, we analyze the factors which affect the networks load. We will propose and implement two approaches to dynamically selecting access points based on our analysis. The first one is to broadcast load information by APs. The second one is to transmit detection frames to APs by stations so as to check the network loads of APs. The response time from APs will used to make the selection. The latter approach can select better APs under almost all conditions. Finally, we verify the effectiveness of our schemes by showing real experimental results.

## 誌 謝

感謝指導老師曾煜棋教授的長時間細心的指導，這篇論文才得以順利完成，期間的辛苦與壓力超乎我預期的想像，一方面有工作上的負擔以及課業上的繁重，常因自我期許較高以及各方面皆要兼顧下增加了不少壓力，也謝謝老師的包容與體諒。

感謝林致宇學長在論文研究過程中提供了不少寶貴的意見及見解，以及在論文的最後階段裡，在百忙中還辛苦撥出時間來幫忙校正。

感謝口試指導委員簡榮宏教授、陳志成教授、林亭佑學姊的指導與意見，使得本篇論文更臻完善。

感謝金麗科技支持在職進修的員工以及實驗設備的支援，使我能擁有充分的資源。



回首這兩年的研究生生涯，能順利得走過來，都要感謝我親愛的妻子蕙雯，承擔起繁重的家務，且不斷的給予我鼓勵與支持，使得在就讀專班期間無後顧之憂，能專心的完成學業。

# 目 錄

中文提要	.....	iii
英文提要	.....	iv
誌謝	.....	v
目錄	.....	vi
表目錄	.....	viii
圖目錄	.....	ix
一、	簡介.....	1
1.1	無線網路現況.....	1
1.2	802.11 通訊協定.....	1
1.3	影響基地台服務品質的相關問題.....	4
1.3.1	選擇基地台方式的問題.....	4
1.3.2	基地台間干擾的問題.....	4
1.3.3	基地台負載變化的問題.....	5
1.4	本篇論文結構.....	6
二、	相關研究.....	7
2.1	描述.....	7
2.2	比較.....	8
三、	廣播負載策略.....	11
3.1	動機.....	11
3.2	描述.....	11
3.3	架構.....	13
3.3.1	基地台的實作策略流程.....	13
3.3.2	工作站的實作策略流程.....	15
四、	探測訊框策略.....	17
4.1	動機.....	17
4.2	描述.....	18
4.3	策略.....	18
五、	實驗.....	21
5.1	描述.....	21
5.2	工具.....	21
5.2.1	linux-wlan-ng 驅動程式.....	21
5.2.2	金麗科技開發用電路板.....	22
5.2.3	PCATTCP 產生 TCP 流量的工具.....	23
5.3	FTP 下載檔案的測試環境.....	24
5.4	FTP 下載檔案的測試結果.....	25

5.5	探測訊框策略PING的準確度測試環境.....	26
5.6	探測訊框策略PING的準確度測試結果.....	27
六、	結論.....	30
參考文獻	.....	31
附錄一	.....	32
附錄二	.....	35



## 表目錄

表 2.1	考量網路負載的因數比較表·····	9
表 2.2	架構流程比較表·····	9
表 2.3	實作方面的問題比較表·····	10
表 3.1	Ideal 和 In Practice Throughput 比較表	12
表 5.1	FTP 下載檔案的平均值·····	26
表 5.2	探測時間的平均值·····	29



## 圖目錄

圖 1.1(a)	無基礎架構的無線區域網路	2
圖 1.1(b)	有基礎架構的無線區域網路	2
圖 1.2(a)	被動掃瞄	3
圖 1.2(b)	主動掃瞄	3
圖 1.3	選擇基地台方式的問題	4
圖 1.4	基地台間干擾的問題	5
圖 1.5	基地台負載變化的問題	5
圖 2.1	基地台的傳輸速率問題	8
圖 3.1	基地台負載計算示意圖	12
圖 3.2	廣播負載策略的架構	14
圖 3.3	自行定義的負載訊框	14
圖 3.4	工作站的實作策略流程	15
圖 3.5	基地台與工作站架構分層機制圖	16
圖 4.1	DCF 狀態圖	18
圖 4.2	工作站策略流程圖	20
圖 5.1	PCATTCP 流量狀態圖	24
圖 5.2	FTP 下載檔案的測試環境	25
圖 5.3	FTP 下載檔案的測試結果	26
圖 5.4	探測訊框策略的測試環境	27
圖 5.5	探測時間分布圖 (a)(b)(c)(d)(e)	28

# 第一章、簡介

## 1.1 無線網路現況

無線網路媒介無所不在，使用者不須到處拉線，因此擴充十分容易及具有彈性，使得人們漸漸汰換舊有的有線傳輸，行動主機的普及化帶動整個無線網路的需求，尤以熱點(Hot Spot)擴增最為明顯，熱點設置已由學校、機場、咖啡館、速食店以及旅館漸漸漫延至許多公共場所，使我們隨時隨地皆可使用網際網路上的服務。

IEEE 802.11 根據無線區域網路協定的功能需求，擬訂了一套無線區域網路系統的基本架構。IEEE 802.11 將最低的功能頻寬訂為 1Mbps，這對於一般性的操作，像檔案傳輸、程式載入、交易處理等，是絕對必要的。對於需要傳輸即時資料的應用軟體，像數位式聲音、影像等，IEEE 802.11 也提供了時限性 (time bounded) 的服務。另外，IEEE 802.11 也定義了包括財務、辦公室、學校以及工業大樓等各種環境中的可靠操作需求。此外，還規範了行動式的電腦系統至少必須支援每小時幾哩的行人速度。由於近年來無線網路傳輸速度呈現跳躍式的演進，從 2Mbps, 11Mbps, 54Mbps, 甚至現今已有廠商以自定規格號稱可達 108Mbps，姑且不論實際是否能達到此速度，但因無線網路速度演進之快，自然產生新的應用，手機、喇叭、DVD、及儲存媒體都已加入無線網路功能，新的應用以及創意不斷產生，在這蓬勃發展的無線網路，新的需求也不斷產生，除了下一代的傳輸規格，網路安全，QOS 新規格也正如火如茶被提出討論與修訂中。

以下章節我們將陸續介紹 802.11 無線區域網路基本架構及通訊協定，並分析現今規格所產生的問題。

## 1.2 802.11 通訊協定

IEEE 802.11 制訂出兩種不同類型的無線區域網路基本架構：1. 有基礎架構的無線區域網路 (Infrastructure Wireless LAN)，2. 無基礎架構的無線區域網路 (Ad Hoc Wireless LAN)

### 1. 有基礎架構的無線區域網路

所謂的基礎架構通常指的就是一個現存的有線網路分散式系統 (wired distribution system)，如圖 1.1(b)，在這種網路架構中，會存在一種特別的節點，稱作擷取點 (access points)，這個擷取點的功能就是要將一個或多個的無線區域網路和現存的有線網路分散系統相聯結，以提供某個無線區域網路中的工作站，能和較遠距離的另一個無線區域網路的工作站通訊，另一方面也促使無線區域網路中的工作站，能擷取有線分散式系統中的網路資源。這一類型的無線網路通訊範圍，通常是以同一棟建築物出現，例如，商店、醫院、或是同一棟樓層。

## 2.無基礎架構的無線區域網路

無基礎架構的無線區域網路主要是要提供不限量的用戶，能即時架設起無線通信網路，在這種架構中，通常任二個用戶間都可直接通訊，這一類的無線網路架構在會議室裡經常用得上，如圖 1.1(a)。

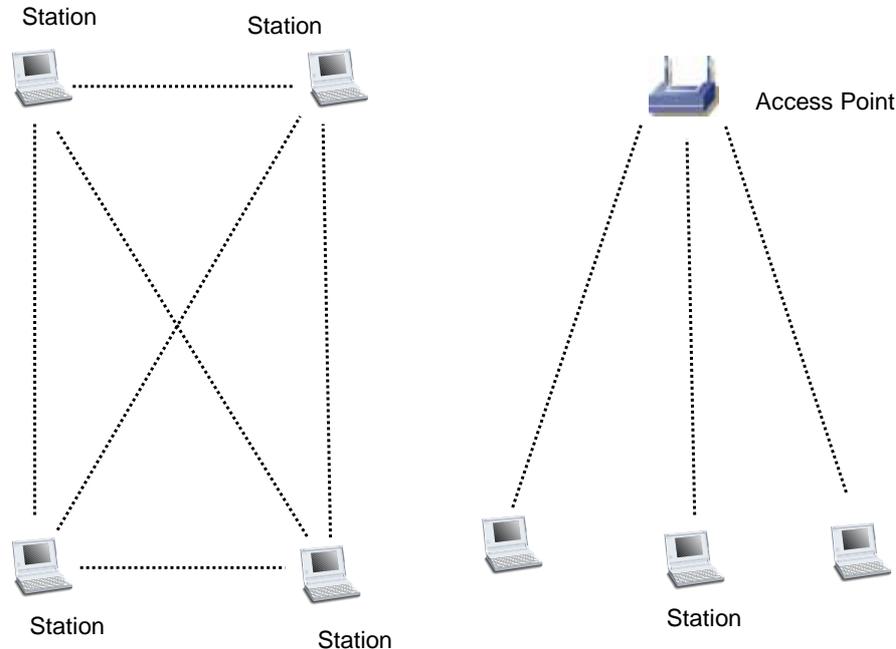


圖 1.1(a)：無基礎架構的無線區域網路 圖 1.1(b)：有基礎架構的無線區域網路

802.11 基本通訊協定為 CSMA/CA (Carrier Sense Multiple Access With Collision Avoidance)。如果同時有二個或三個以上的工作站同時傳送訊框將造成碰撞，發生碰撞的訊框視為無效並丟棄。802.11 也提供兩種傳送方式，分散式協調功能 (Distributed Coordination Function, DCF) 使用 CSMA/CA，適合傳輸非即時資訊，非即時傳輸使用之頻寬不保證公平分配，在 DCF 部份由於工作站利用 CSMA/CA 通訊協定來互相競爭傳送訊框的機會，並沒有輪流傳送的特性，因此每個工作站實際使用的頻寬可能不同。而集中式協調功能 (Point Coordination Function, PCF)，由網路協調者 (Point Coordinator) 掌控並且以輪詢 (Polling) 的方式安排工作站傳送訊框的時機及順序，由於工作站傳送的時間可事先安排，因此可提供保證傳送延遲的服務。

前述題及在無線網路可以透過基地台可以讓使用者連線到既有的有線網路，此種通訊模式稱為基礎架構模式，如何選擇一個讓使用者得到最大頻寬，及最佳服務品質的基地台便是此篇論文所要探討的問題。但在選擇加入任何相容網路之前，工作站必須先經過一番辨識工作。於所在區域辨識現有網路的程序稱為掃瞄 (scanning)，掃瞄的型態有兩種，一為被動掃瞄，一為主動掃瞄。

被動掃瞄 (Passive Scanning) 可以節省電池電力，因為不需傳送任何訊號。如圖 1.2(b) 在被動掃瞄，工作站會針對頻道表 (Channel List) 所列的各個頻道加以掃瞄，並靜候 Beacon 訊框的到來。所收到的任何訊框都會被暫存起來，以便取出傳送這些

訊框的 BSS 相關資料。在被动扫描过程，工作站会在频道间不断切换 并且会从所收的 Beacon 讯息记录资讯。Beacon 讯框在设计上是为了让工作站得知加入某个基本服务组合(Basic Service Set 简称 BSS)所需之参数，以便进行通讯。

在主动扫描(Active Scanning)，工作站扮演比较积极的角色。如图 1.2(b)，在每个频道上，工作站都会发出探查要求讯框(Probe Request)讯框，请求某个特定网路发出回应。主动扫描係主动试图寻找网路，而不是听候网路宣告本身的存在。使用主动扫描的工作站会以下列程序扫描频道表所列频道：

1. 跳至某个频道，然后等候来讯显示，或等到计时器逾时，如果在這個频道收得到讯框，就证明该频道有人使用，因此可以加以探测。计时器是用来防止某个閒置频道让整个程序停摆：工作站不会一直听候讯框到来。
2. 利用基本的 DCF 存取程序取得媒介使用权，然后送出一个 Probe Request 讯框。
3. 至少等候一段最短的频道时间(Minimum Channel Time)。
  - (1)如果媒介并不忙碌，表示没有网路存在。因此可以跳至下个频道。
  - (2)如果在這段最短的频道时间内，媒介非常忙碌，就继续等候一段时间，直到最长的频道时间(Maximum Channel Time)，然后发出探查答覆讯框(Probe Response)讯框。当网路收到搜寻所属延伸服务组合的探查要求讯框(Probe Request)，就会发出探查答覆讯框。

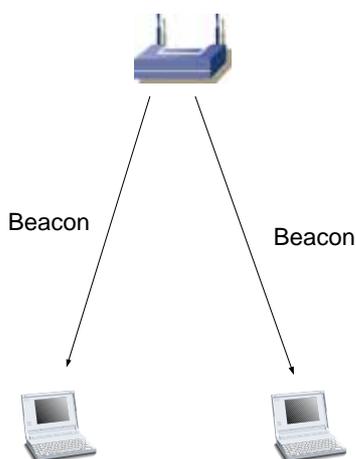


圖 1.2(a)：被动扫描

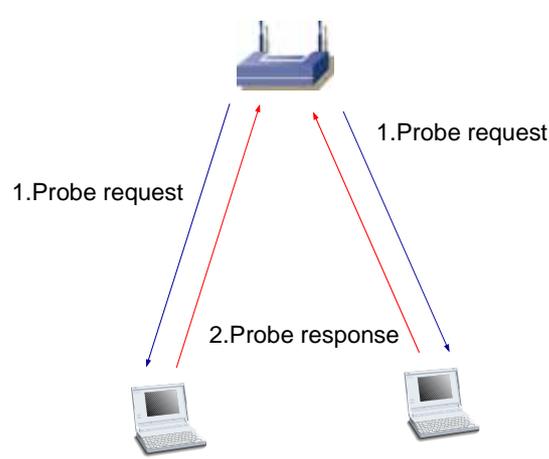


圖 1.2(b)：主动扫描

扫描结束后会产生一份扫描报告。这份报告列出了该次扫描所发现的所有 BSS 及相關参数。进行扫描的工作站可以利用这些完整的参数，加入(Join)其所发现的任何网路。

### 1.3 影響基地台服務品質的相關問題

在基礎架構模式，使用者要經由聯結基地台得到網路服務，而較佳的網路服務往往取決於基地台是否可以提供較大頻寬予以工作站，但在現實環境裡，使用者常面臨多個基地台，無法判別聯結至哪個基地台可以獲得最佳服務品質，此外 IEEE 802.11 並無規範選擇基地台的策略，使得工作站往往要面臨種種以下問題。

#### 1.3.1 選擇基地台方式的問題：

以現今市面上的產品，工作站聯結基地台都是依據信號強度(RSSI)或者是接受到 Beacon 訊框的先後順序，而開始做加入的動作，但這會造成有時大多數的工作站聯結到少數的基地台，如圖 1.3 所示，所有工作站皆若只依據信號強度聯結到基地台 AP1，容易因競爭而產生碰撞，因此整體網路的吞吐量會下降，而基地台也不能獲得更有效率應用。

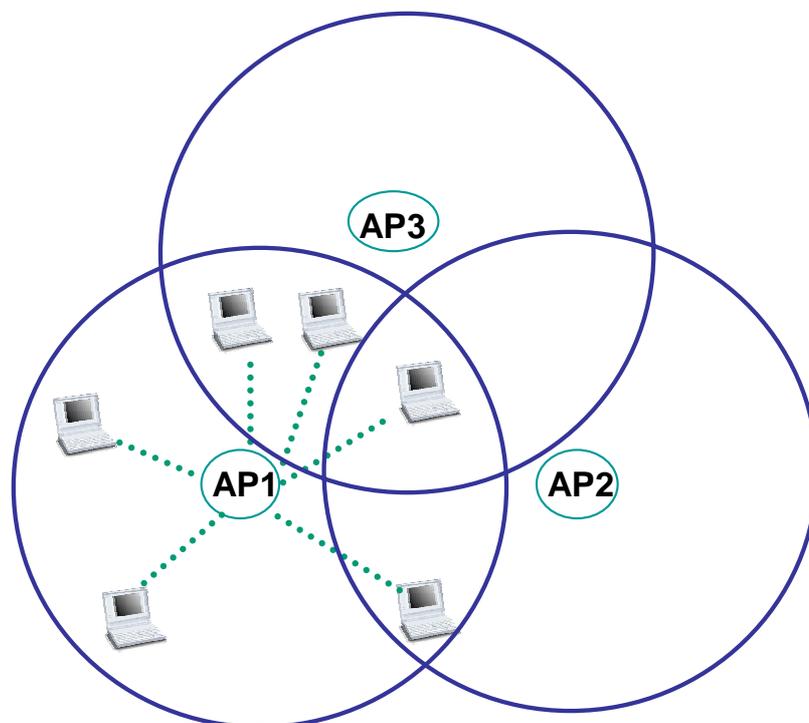


圖 1.3：選擇基地台方式的問題

#### 1.3.2 基地台間干擾的問題：

基地台部署的位置也會產生干擾的問題，進而影響網路的吞吐量或服務品質，雖然基地台部署問題並不在本篇文章討論的範圍內，但如何使工作站在這情況下可以正確選擇較佳的基地台，減低頻道訊號相互干擾。如圖 1.4，基地台 AP2 並無任何工作站聯結，但基地台 AP1 和基地台 AP2 共用同一頻道，會有訊號干擾問題，因此工作站 STA1 聯結到基地台 AP2 未必是較佳選擇。

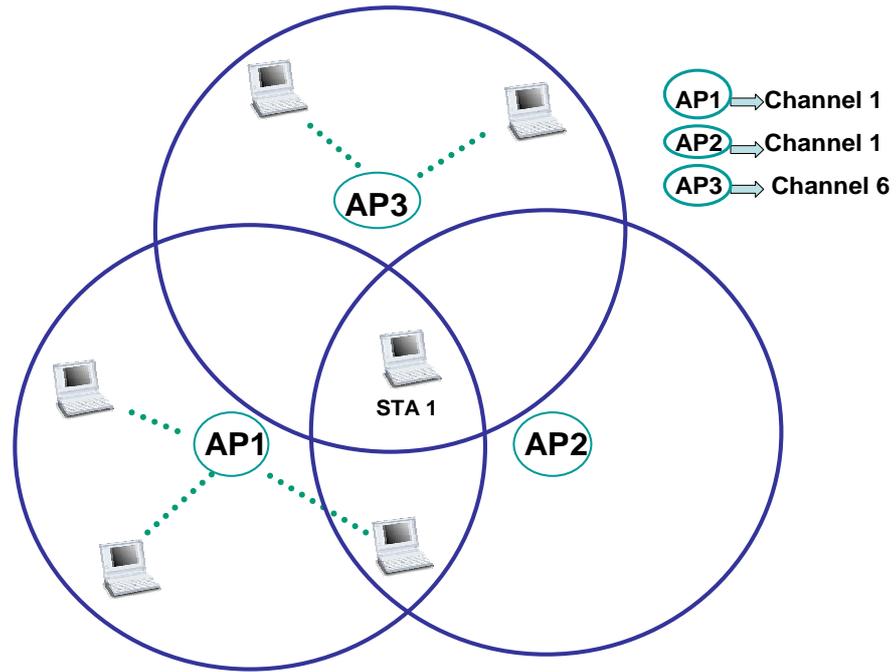


圖 1.4：基地台間干擾的問題

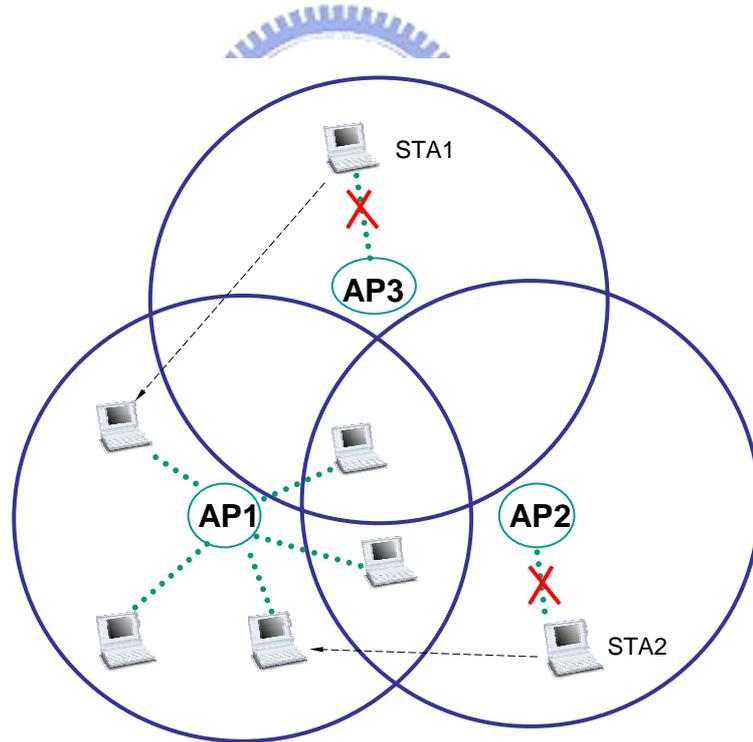


圖 1.5：基地台負載變化的問題

### 1.3.3 基地台負載變化的問題：

基地台在不同時間，有不同數量工作站聯結上來，或是工作站移動而離開目前基地台的服務範圍，而造成 BBS 轉換，網路的負載皆會產生變化，如圖 1.5，工作站 STA1 和工作站 STA2 離開原本基地台服務區域而移動到基地台 AP1 的區域，造成整體網路

的負載不平衡，但現今市面上的基地台和工作站並不能週期性偵測和調整負載，使得工作站 STA3 和 STA4 沒有機會選擇較佳的基地台，我們也必須解決此問題，讓網路整體頻寬不論何時使用率皆維持最大值。

造成這些問題最主要原因是目前 IEEE 802.11 並無規範一方式可以偵測網路的狀態，及回報通知基地台或工作站，然後作相對應的處理。綜觀上述的問題，我們假設要解決的問題是架構在基礎建設網路裡，使用 DCF 傳輸模式，使得工作站可以在不同的情況下動態選擇基地台，且不論何時都可以獲得最佳傳輸效能，進而使網路整體使用率維持最大，因此，我們提出兩個策略來解決這些問題，一是廣播負載策略，一是探測訊框策略。

#### 1.4 本篇論文結構

論文第二章將對相關研究來作討論，並與我們提出的策略作各方面的比較，第三章詳述廣播負載策略的方法，雖然亦無法解決 1.3.2 的問題，但對於影響網路負載平衡考量更為周全，在選擇基地台時加上考慮傳送資料時的傳輸速率，並將此策略實作在現有產品上。第四章則詳述探測訊框策略，此方法可解決前述所有的問題，在實作上並不需修改工作站的驅動程式。第五章對這兩個策略作實作的測試，並分析其結果，第六章則作一個總結。



## 第二章 相關研究

### 2.1 描述

市面上的產品選擇基地台所採用的策略我們稱之為傳統方法，傳統方法並無提供動態選擇基地台或頻道的策略，因為此方法一旦聯結到基地台，除非斷線，否則將不會改變所選擇的基地台。傳統方法選擇基地台的方式是工作站先掃描所有頻道，根據信號強度(RSSI)聯結基地台，最主要是因為無線網路硬體或軟體會依訊號強度決定工作站傳輸速率，因為訊號強度低代表距離較遠，為了保有一定的傳輸品質，會以較低的速率傳輸。

在[1]中，此篇文章要解決網路負載平衡的問題，而訊號強度和聯結基地台的工作站個數，這二項變數設定為選擇基地台所考慮的因數，使所有工作站的平均訊號強度是最大值為其目標。

在[2]中，也是要解決網路負載平衡的問題，最主要分為三個部分：

- 1.將基地台頻道作最佳化的設定配置，而不會產生訊號干擾。
- 2.工作站根據訊號強度和工作站個數量選擇聯結基地台。
- 3.基地台會週期性的更新負載的狀態且廣播通知工作站，工作站也會週期性的監督基地台負載狀態，若基地台的負載超出上限值的狀況下會找新的基地台聯結。

在[1]及[2]的方法中，只依據工作站個數和訊號強度選擇基地台是不夠正確的，而沒有考慮到工作站是需透過競爭才可取得傳送訊框權利的特性，且工作站並非總是在傳送資料，因此若只依工作站個數來判斷並不精準，而應以真正參與傳送競爭的工作站個數作為考量。

在[3]中，此方法延伸[2]的概念，並加入實際運作工作站(Number of Active Station)的因數使其能得知目前有幾個工作站在競爭，而更準確的判斷網路負載的狀態，來選擇基地台。此篇論文仍然有下列問題無法解決：

- 1.前述[1]、[2]及此篇論文皆是要解決基地台負載平衡的問題，因此對其基本假設都是基地台間不會互相干擾，在這個假設前提下，再來訂定工作站選擇基地台的策略，但在現實的環境中，多個基地台的環境中仍是無法避免彼此間的干擾，導致工作站錯選了服務品質差的基地台。
- 2.其次，此篇並無考慮到聯結基地台的傳輸速率問題，如圖 2.1，若可以將基地台的負載以數據化表示，如基地台 AP1 的負載為 32%，基地台 AP3 的負載為 2%，理論上應聯結基地台 AP3，但工作站因對基地台 AP3 訊號較弱，而導致傳送速

率較慢，因此選擇基地台 AP3 並非是最好的抉擇，所以選擇基地台的方法也應要把傳送速率考慮進去。亦即工作站聯結到基地台雖然預期可獲得的吞吐量較高，但受限於傳送速率，無法以最大的傳輸率傳輸，因此對工作站而言，如何選擇較佳的基地台，傳輸速率是一個重要的因數。

3.此篇實作的方面需要新增類似 Inter Access Point Protocol (IAPP) 協定，即是基地台與基地台間可以互換訊息的協定，若要實作而小幅度修改現有的產品較為困難，雖然此篇論文並無實作，但已把整體運作的策略步驟及訊框欄位定義詳細描述，並以 NCTUns 作模擬實驗。

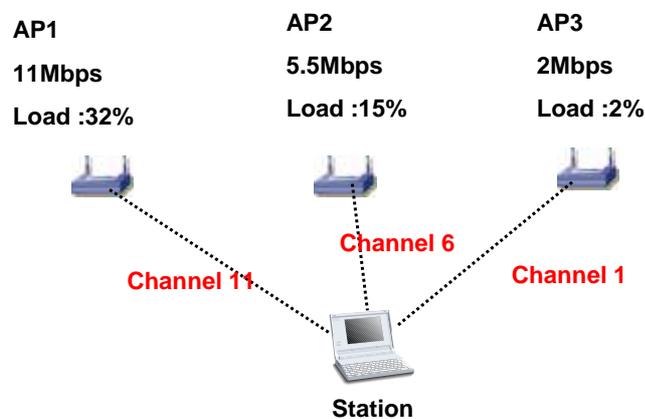


圖 2.1：基地台的傳輸速率問題

在 [4] 中，此篇文章的概念，是依據訊號強度和傳送訊框錯誤率 (Packet Error Rate, 簡稱 PER) 及載波偵測/淨空頻道評估 (Carrier Sense/Clear Channel Assessment, 簡稱 CS/CCA) 來指定基地台的頻道，其中 CS/CCA 我們可以把它視為基地台負載的反比，而訊框錯誤率所表示的涵義應可視作訊號間碰撞干擾的程度，也可能是基地台與基地台間使用相同頻道造成訊號互相干擾，雖然此篇文章是指定及改變基地台的頻道的策略來達成網路有效率的應用，但基地台頻道的變動，工作站的頻道也相對的改變，且基地台是由收集所有工作站回報的網路狀況資訊而做抉擇，因此我們也仍可視為是另一種工作站選擇基地台的策略，但相同地，與 [2] 一樣無法得知傳送資料時，有幾個工作站正在參與競爭。

## 2.2 比較

綜合上節的相關研究，以下我們將上述選擇基地台的各種方法與我們提出的策略，分別依“考量網路負載的因數”，“架構流程”，“實作方面的問題”來作比較。

1. 考量網路負載的因數:綜合前述的所有問題，可以歸納出影響網路負載的因數有四項，分別為，訊號干擾，實際運作的工作站個數，基地台的負載，和傳輸速率，因此我們將選擇基地台各種策略，和對考量哪些影響網路負載的因數作比較，從表 2.1 可以看到廣播負載策略選擇基地台時考慮了三項因素，而探測訊框策略則涵蓋所有要素。

表 2.1 考量網路負載的因數比較表

	訊號干擾	實際運作的工作站個數	基地台的負載	傳輸速率
傳統方法				V
[1]				V
[2]				V
[3]		V	V	
[4]	V		V	
廣播負載策略		V	V	V
探測訊框策略	V	V	V	V

2. 架構流程:要解決選擇基地台的問題，以其架構來區分依其流程順序有三大步驟，(1)誰來作負載測量報告，(2)誰將負載結果收集，(3)誰作選擇或交換基地台的決定。從表 2.2 可以看到[3]、[4]都是由基地台決定作出選擇基地台或指定頻道，廣播負載策略則與[1]、[2]相同，都是由基地台來測量或計算負載資訊，再通知工作站，而探測訊框策略三項步驟都是由工作站自己獨立完成。

表 2.2 架構流程比較表

	誰來作負載測量報告	誰將負載結果收集	誰作選擇或交換基地台的決定
傳統方法	無	無	工作站
[1]	基地台	工作站	工作站
[2]	基地台	工作站	工作站
[3]	基地台	基地台	基地台
[4]	工作站	基地台	基地台
廣播負載策略	基地台	工作站	工作站
探測訊框策略	工作站	工作站	工作站

3. 以實作方面的問題來比較:有三點來作探討(1)實作的程度，(2)是否更動 IEEE 802.11 協定，(3)修改工作站或基地台軟硬體的程度。其中(2)是否更動 IEEE

802.11 協定此項，因大部分的策略都須要在訊框中特別定義一欄位來代表負載資訊，因此都需要更動 IEEE 802.11 協定，另外(3)修改工作站或基地台軟硬體的程度此項，我們比較若要將其策略實作時，是否一定要更動無線網路晶片的硬體設計，或是只要驅動程式修改即可，畢竟晶片設計更動的成本要比軟體大的多，從表 2.3 可以看到由於[4]的策略是依 CS/CCA 及 PER 選擇頻道，因此須要硬體支援，其餘只須修改驅動程式就可達成，而探測訊框策略透過系統命令程式即可達成實作的目的。

表 2.3 實作方面的問題比較表

	實作的程度	是否更動 IEEE 802.11 協定	修改工作站或基地台軟硬體的程度
[1]	模擬	是	驅動程式
[2]	無	是	驅動程式
[3]	模擬	是	驅動程式
[4]	無	是	硬體
廣播負載策略	實作	是	驅動程式
探測訊框策略	實作	否	無



## 第三章 廣播負載策略

### 3.1 動機

在傳統方法中，工作站皆依據信號強度聯結基地台，使得大多數工作站聯結至相同基地台，導致競爭加劇，訊框容易產生碰撞，網路也不能獲得更有效率應用，因此我們提出一策略希望能綜合前述相關研究，可以實作並解決選擇基地台的問題，首先實作的架構與[2]相似，希望在 Beacon 或 Probe Response 訊框中加入負載資訊的欄位，使得工作站可以隨時得知基地台的狀況，進而判斷是否須要更換基地台，但以目前市面上的無線網路晶片若只更改驅動程式是很難作的到，因此我們作了一些 802.11 協定的更動，以及修改驅動程式來達成此目標，其次，我們也希望基地台的廣播資訊包含負載狀態以及實際運作的工作站數量，其中負載狀態若是可以與[4] CS/CCA 相同，精準測出媒介的狀況是最佳選擇，但實作方面硬體並無法更動來支援，因此參考[3]的策略並作定義上的修改，把負載的資訊與實際運作的工作站數量以運算公式表達，最後，將傳送速率列入考量，讓選擇基地台的策略趨於完整。

### 3.2 描述

此策略對於選擇基地台涵蓋了三個影響網路負載的因數：

#### 1. 基地台的負載

我們希望能把負載的定義能在驅動程式裡以數值化表示，並利用公式計算出來，所以將負載定義改為資料傳送與接收時間佔據總時間比例的概念，如圖 3.1 所示，在時間間隔  $t$  的時間中，我們假設最大可以允許  $T_m$  的時間來傳送或接收資料，但若算出實際上傳收訊框的時間，就必須將所有個別傳送訊框長度 ( $tx\_len$ ) 除以個別傳送資料速率 ( $tx\_rate$ )，而得到個別傳送時間，接收的訊框亦同，可以得到個別接收時間，最後將個別傳送及接收時間加總，除以最大傳送時間  $T_m$ ，最後得到的值即為單位時間內媒體介質忙碌時間的比例，我們定義此為基地台的負載 (Load)。在計算傳送時間的動作過程是在驅動程式系統中，將要預備傳送的資料長度除以對此工作站的傳送速率，但傳送時需要等待的競爭或重傳動作，是由硬體機制完成，因此這部分的時間是無法估算的，基地台接收的訊框也是無法得知競爭所需的時間及因傳送訊框失敗而需重傳的時間，因此此部分的計算會有些許的誤差。

#### 2. 實際運作工作站的數量

我們以基地台接收或傳送訊框的工作站，認定是加入參與競爭的工作站，因此統計在這段時間間隔  $T$  裡有幾個不同的工作站發出訊框和接收訊框，便視為實際運作的工作站數量。

當工作站選擇基地台時，除了參考基地台負載外，還必須考慮有多個工作站在競爭的情況，因此我們要計算出每個工作站預期可分得的時間，以及空閒媒介的時間，這兩種時間取較大值者，舉例來說，我們將時間切成若干個時間間隔  $T$ ，在這段  $T$  的時間中，假設最大可以允許  $T_m$  的時間來傳送或接收資料， $T$  和  $T_m$  的關係可以由表 3.1 推算得知， $T_m$  大約占  $T$  的 50%，若有  $N$  個工作站，因為競爭的緣故，預期每個工作站可分得  $T_m/N$  的時間，但實際頻道可能只有  $T_c$  的時間在傳送資料，而剩餘的  $(T_m - T_c)$  並沒有資料傳送，因此真正要傳送資料的工作站是可以分配取得的時間為  $\{T_m/N, T_m - T_c\}$  這兩者較大值來傳送或接收資料。

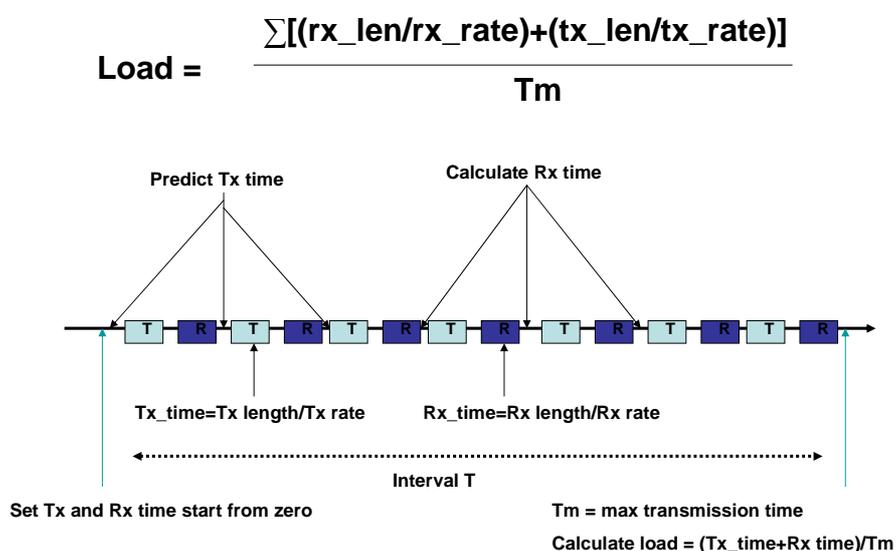


圖 3.1：基地台負載計算示意圖

表 3.1：Ideal Throughput 和 In Practice Throughput 比較表

	Ideal Max Throughput	In Practice Max Throughput	In Practice / Ideal
802.11b	11Mbps	6Mbps	54%
802.11g	54Mbps	26Mbps	48%

### 3. 連結時傳送的速率

當工作站選擇基地台時，還必需考慮連結此基地台的資料傳送速率。我們定義前述  $\{T_m/N, T_m - T_c\}$  這兩者較大值的時間為  $T_s$ ，而連結的基地台傳送速率是  $R$ ，因此計算出此基地台可以提供的吞吐量是  $T_s * R$ ，使用相同的方法，計算此環境

所有基地台所提供的吞吐量大小，就可以選擇到較好基地台作聯結。

我們將工作站選擇基地台的公式列於下：

*Max\_Time* : 基地台在時間間隔中提供最大的傳輸時間

*Num\_Active\_STA* : 基地台統計實際運作工作站的數量

*Exp\_Time* : 平均工作站可以分配到的時間

*Comsume\_Time* : 基地台的負載

*Remain\_Time* : 空閒媒介的時間

*Trasmit\_Rate* : 對此基地台傳送資料的速率

*Throughput* : 對此基地台可以獲得的吞吐量

工作站收到基地台負載的資訊，根據以下步驟計算應選擇那個基地台聯結

計算現在聯結基地台 AP1 預期可以獲得的吞吐量，

$$Exp\_Time1 = Max\_Time1 / Num\_Active\_STA1$$

$$Remain\_Time1 = Max\_Time1 - Comsume\_Time1$$

$$Throughput1 = Trasmit\_Rate1 \times Max \{Exp\_Time1, Remain\_Time1\}$$

計算若聯結到其它基地台 AP2 可以獲得的吞吐量，

$$Exp\_Time2 = Max\_Time2 / Num\_Active\_STA2 + 1$$

$$Remain\_Time2 = Max\_Time2 - Comsume\_Time2$$

$$Throughput2 = Trasmit\_Rate2 \times Max \{Exp\_time2, Remain\_time2\}$$

選擇可以獲得較佳吞吐量的基地台。

$$Choice \ the \ AP \ is \ Max \ {Throughput1, Throught2\}$$

### 3.3 架構

此策略在架構上與[3]有所不同，圖 3.2(b)為[3]的架構，基地台與基地台間要互相交換負載資訊，收集相連基地台的資訊，並計算自己的負載是否超出臨介上限值，再通知工作站切換基地台，而我們提出的策略在架構上與[2]的方法較為相似，如圖 3.2(a)，基地台定期廣播及更新基地台的負載資訊，而工作站定期掃描及收集所有頻道裡基地台的資訊，再判斷決定聯結到那個基地台，以下討論在此架構下基地台與工作站實作時的策略流程。

#### 3.3.1 基地台的實作策略流程

- (1) 基地台每一秒鐘皆更新資訊，更新的資訊包含實際運作工作站的數量及基地台的負載，前面提及基地台的負載是計算所有個別傳送及接收訊框長度除以個別傳送速率，所得到的時間加總，實際運作工作站的數量則是統計傳送和接收到幾個不同的工作站發出的訊框。
- (2) 基地台收到聯結(Association)或再聯結訊框(Re-Association)，則發出廣播負載訊框，如圖 3.3 為自行定義之負載訊框，我們定義一個訊框型態欄位(Type)為

10b，代表是資料訊框，而次型態欄位(Subtype)為 1111b，在 IEEE 802.11 協定中 Type 是 00b 且 Subtype 是 1111b 目前在 IEEE 802.11 中定義為保留，因此需要修改工作站驅動程式辨識此特殊訊框。定義為資料型態訊框的目的，是因為工作站實體層接收後會傳送到驅動程式中，而不會自行將此訊框丟棄或自行處理。在資料欄位中，填入負載的相關資訊，這些資訊有基地台 BSSID、頻道號碼、基地台在時間間隔中提供最大的傳輸時間、基地台的負載、實際運作工作站的數量，以及傳送速率。廣播負載訊框的用意是希望在相同頻道中的工作站可以接收到此負載訊框，更新最新基地台的狀況。

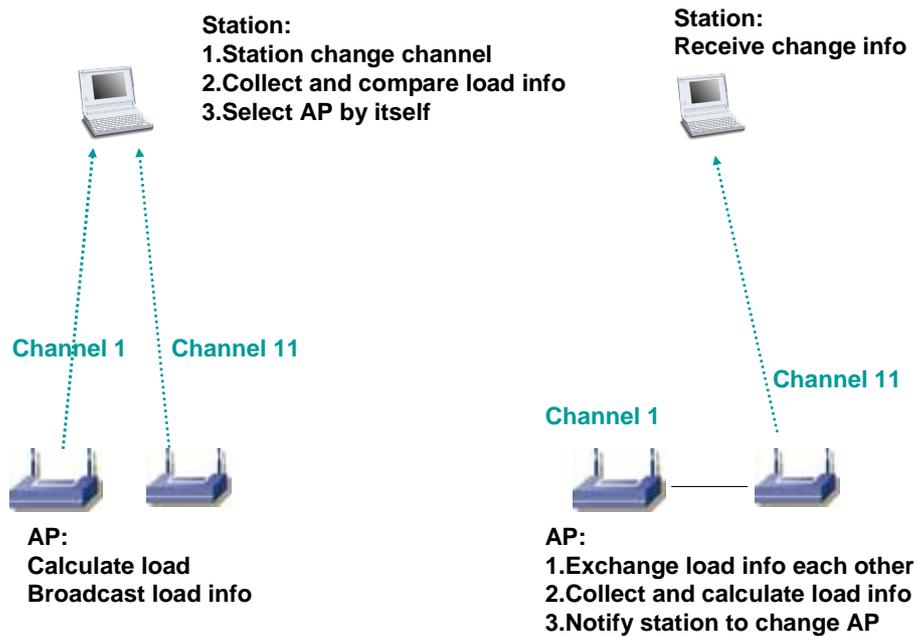


圖 3.2(a)：廣播負載策略的架構

圖 3.2(b)：策略[3]的架構

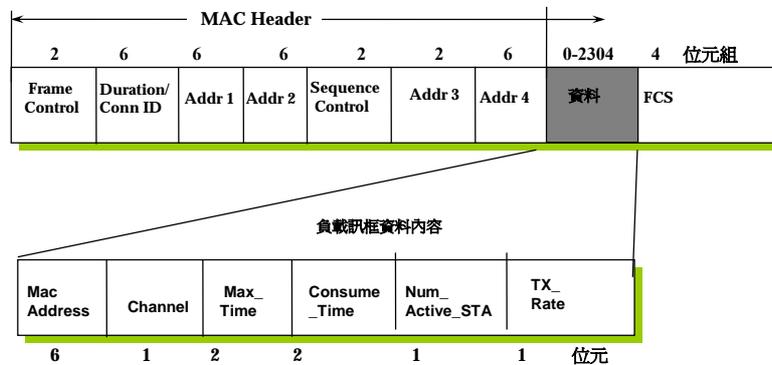


圖 3.3：自行定義的負載訊框

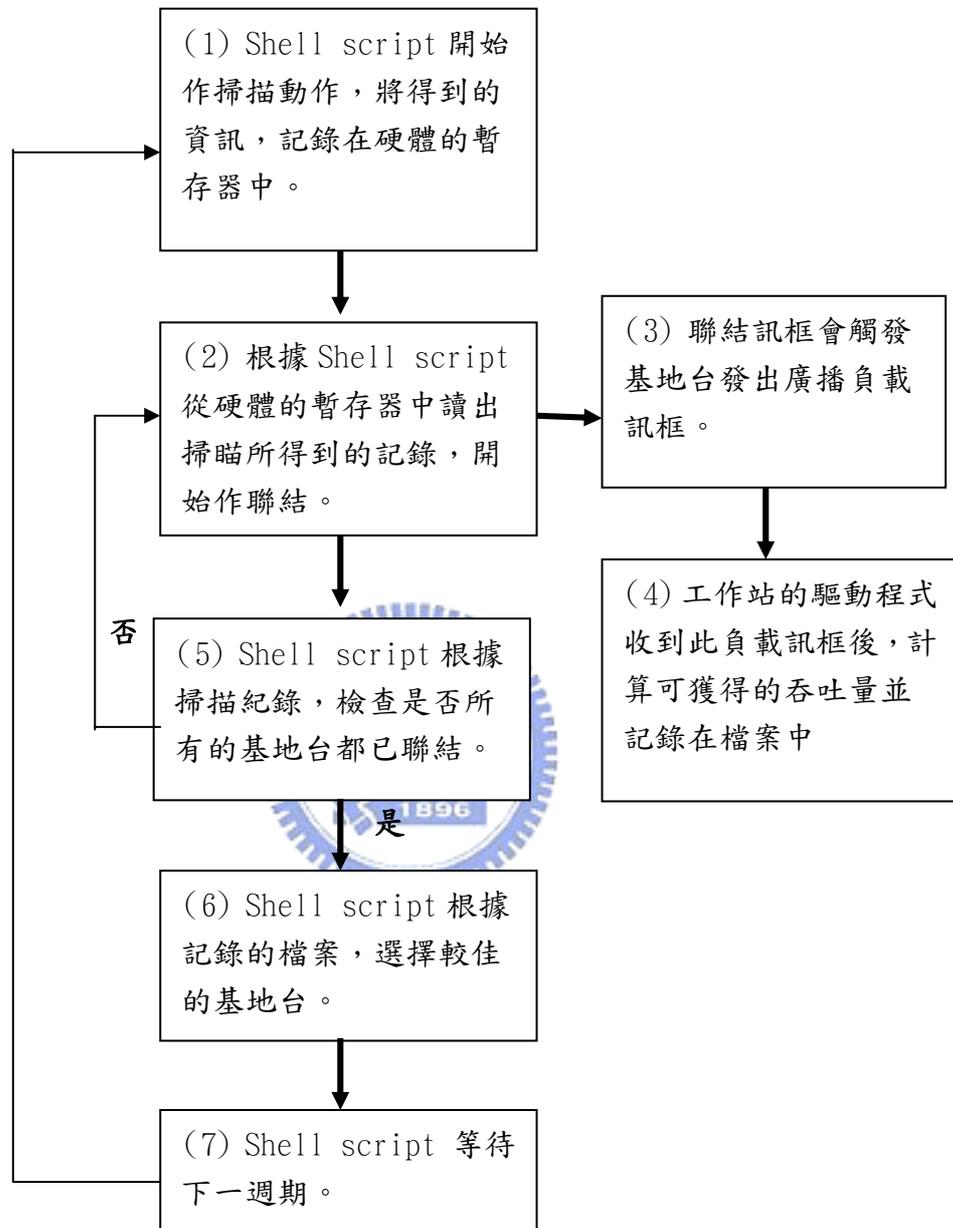


圖 3.4：工作站的實作策略流程

### 3.3.2 工作站的實作策略流程

工作站在執行 shell script 之前，須先載入無線網路卡驅動程式。圖 3.4 為工作站的實作策略流程，以下依序說明每一步驟。

1. Shell script 開始作掃描動作，將得到的資訊，記錄在硬體的暫存器中。
2. 根據 Shell script 從硬體的暫存器中讀出掃描所得到的記錄，開始作聯結 (Association)。

3. 聯結訊框會觸發基地台發出廣播負載訊框。
4. 工作站的驅動程式收到此負載訊框後，計算可獲得的吞吐量並將相關資訊記錄在檔案中。
5. Shell script 則繼續根據掃瞄記錄，檢查是否所有的基地台都已聯結過了，若是則繼續步驟(6)，若非則回到步驟(2)。
6. Shell script 根據記錄的檔案資訊，選擇較佳的基地台。
7. Shell script 等待下一週期再回到步驟(1)。

歸納前述基地台與工作站的實作策略，將基地台與工作站以架構分層機制圖來說明，如圖 3.5 所示，我們必須修改基地台的韌體，使得基地台可以定期更新及計算負載的資料，並於收到聯結訊框時廣播負載訊框，而工作站方面，在作業系統執行一 Shell Script 應用程式，可以定期掃瞄頻道和聯結基地台，另外再修改驅動程式得到新的負載資訊，並記錄在檔案中。

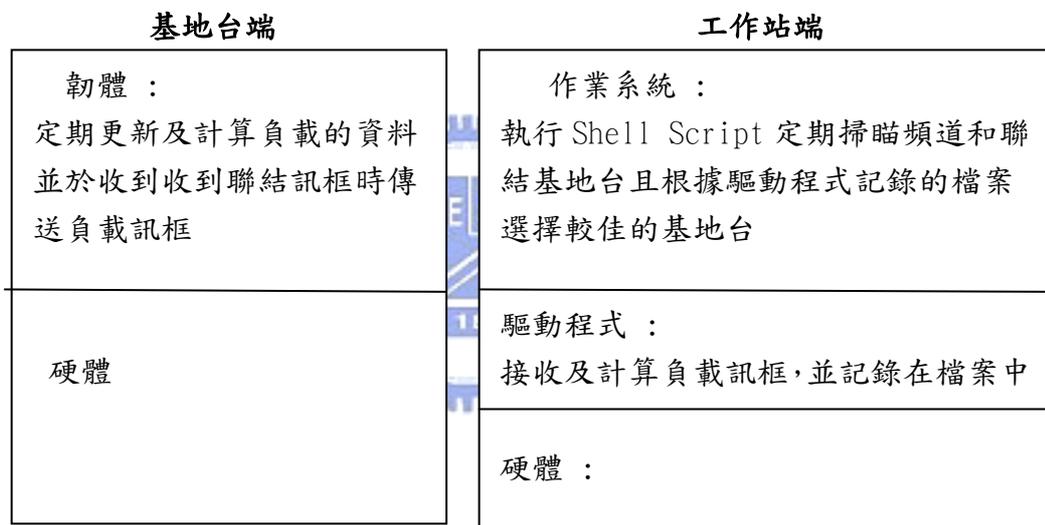


圖 3.5：基地台與工作站架構分層機制圖

## 第四章 探測訊框策略

### 4.1 動機

在上一章的廣播負載策略仍無法解決基地台間訊號干擾，而導致工作站錯選基地台的問題，為了解決此一問題，我們從 IEEE802.11 DCF 傳輸模式的定義來探討，而實作方面，並不希望更動基地台或工作站的驅動程式，而是可以找出一簡單方式將此意義表達出來，若市場或廠商可以接受，甚而將此策略實作在硬體晶片中。

在所有使用 DCF 的傳輸模式，無論是工作站或基地台在試圖傳送任何資料之前，都必須檢視媒介是否處於閒置狀態，若處於忙碌狀態，工作站或基地台皆必須延遲存取，並利用指數型遞延 (orderly exponential backoff) 演算法來避免碰撞發生。我們將整個 DCF 的流程由狀態圖圖 4.1 來表示，當有訊框需要傳送時，狀態圖從『IDLE State』進入到『Packet Arrival State』，若媒介處於閒置狀態便會進入到『Backoff State』，再經過指數型遞延演算法所得出的時間後，若媒介仍是空閒，就可以進入到『Transmit State』將訊框傳送出去，若媒介仍是忙碌則回到『NAV State』，直到媒介空閒時再度進入『Backoff State』。

由此可知，在 IEEE 802.11 協定，傳送訊框是需要透過競爭，且媒介空閒時間越長，訊框愈有機會傳出，換句話說，愈少工作站競爭，碰撞次數便會減少，指數型遞延演算法所得出的時間愈短，而媒介愈不忙碌及干擾愈小，代表可以隨時進入『Backoff State』，傳送時間相對的也變短，但若發生碰撞或在訊號干擾的狀況下，訊框會產生 CRC 錯誤，傳送端因而沒有收到回應訊息，亦即表示此訊框並沒有成功送達，需重新傳送訊框，而錯誤復原 (error recovery) 屬於傳送端的責任，必須負責重送作業，直到傳送成功為止，而整個狀態圖也需從『Transmit State』再重新進入到『Packet Arrival State』，因此我們將探測訊框策略的網路負載定義為傳送訊框時間加上回應訊框時間，可以代表整個網路的狀態。

接著將此策略與廣播負載策略對於影響網路負載的四項因素作比較，是否有滿足考量到所有的要素，第一，實際運作的工作站，此策略也是需要透過競爭才能傳送訊框，因此對於指數型遞延演算法及等待媒介閒置狀態的時間上已滿足此要素，第二，基地台的負載，即是等待媒介閒置狀態的時間，第三，傳送速率，此策略已考慮到回應訊框的時間，也已滿足此要素。因此廣播負載策略可以解決的問題，此策略也同樣可以達成，另外廣播負載策略對於基地台間互相訊號干擾的問題並無法解決，而在此策略中，若傳送訊框時因干擾而產生錯誤，會啟動重送機制，依據此策略的網路負載定義，偵測的回應時間也會變長，代表訊號干擾的嚴重程度增加，由以上的分析可以得知，此策略已滿足所有選擇基地台的要素，因此可以解決前述的相關所有問題。

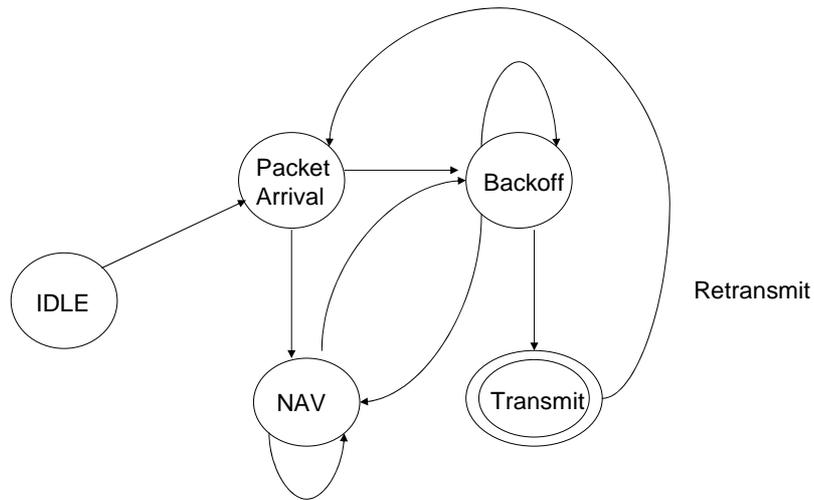


圖 4.1：DCF 狀態圖

## 4.2 描述

根據 3.1 的定義，在理想狀態下，若是要得知此時網路狀態，就從 MAC 主動發出一探測訊框，並開始計算時間，直到收到回應訊框，這段時間的長度便可作為測量網路狀態，但目前無線網路晶片並無法作到，其原因為假設工作站使用管理訊框 (Management Frame) 中的 Probe Request 訊框作為一探測訊框至基地台，基地台成功收到訊框後會回覆 Probe Response 訊框，當工作站收到 Probe Response 訊框時，大部份的無線網路晶片在實體層接收時並不會通知驅動程式，即使可以將回覆訊框通知驅動程式，仍須修改驅動程式來計算時間。

因此若要將此方法精準的實作出來，只有更改硬體設計，但我們提出一個簡單方法模擬此一精神，且不須要修改工作站的驅動程式亦可達到相同目的地，我們利用 Shell Script 檔案下達 ICMP Echo Request 封包來達成相同的目的地，在作業系統下使用工具命令 PING 與基地台相連的任一電腦或工作站，等到收到 ICMP Echo Reply 封包回覆時，記錄這段時間的長度，來測量網路負載的狀態。

## 4.3 策略

工作站執行探測訊框的策略以圖 4.2 來說明：

- (1) 工作站執行探測訊框策略的 Shell Script (附錄一)，在 Shell Script 中發出三次 PING Request 封包，目的地為已知 IP 位址的電腦或工作站，如匣道器 (Gateway)，並記錄平均回應時間。

- (2) 假如平均回應時間小於我們所設定網路負載過重的上限時間，則執行步驟(8)，代表網路負載未過重，並不需要浪費系統資源作探測，因此繼續進行步驟(3)。
- (3) Shell Script 開始作掃瞄動作，將得到的資訊，記錄在硬體的暫存器中。
- (4) Shell Script 從硬體的暫存器中讀出掃瞄所得到的記錄，依序選擇基地台作聯結(Association)。
- (5) 與步驟(1)相同測試網路狀態，把平均回應時間乘於 0.3，而前次對此基地台的回應時間乘於 0.7，將兩者相加當成此次的回應時間，並將此時間及基地台的 BSSID 和頻道等記錄在檔案中。平均回應時間作加權的用意在於預防 PING 的時間誤差較大，一方面也考慮網路的使用行為模式。
- (6) 假如我們測試了所有基地台，則進行步驟(7)，否則進行步驟(4)，測試下一個基地台。
- (7) 依照檔案記錄中的回應時間作排序，並選擇最佳基地台加入。
- (8) 等待週期性的時間，之後回步驟(1)重新啟動此流程。



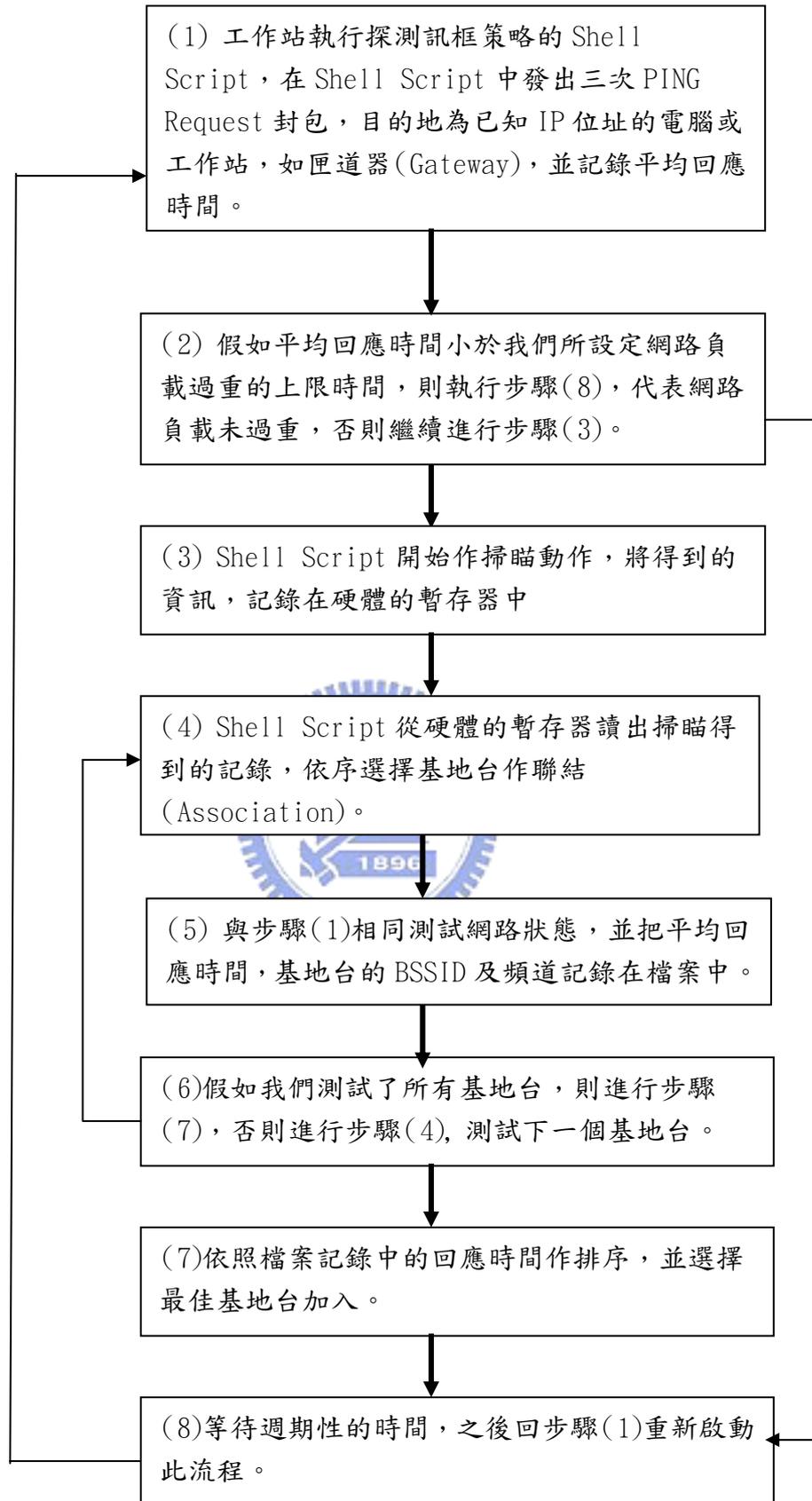


圖 4.2：工作站策略流程圖

## 第五章 實驗

### 5.1 描述

在本章中將對廣播負載策略及探測訊框策略的實作作一些實驗，此實驗的目的在於驗證我們提出的兩個策略相較於傳統方法選擇基地台時更能增進傳輸效能，我們模擬真實網路狀況，並以 FTP 下載相同大小檔案所需的時間來作比較，在有兩台基地台且網路負載會隨時變動的情況下，此兩種策略是否能在較短時間將檔案下載完成。另外我們再針對探測訊框策略作一些實驗，分析及說明 ICMP Echo Request 所測得的時間與網路負載的相關性及準確性。以下章節會針對實驗的工具以及環境作一些介紹，並對結果加以討論。

### 5.2 工具

#### 5.2.1 linux-wlan-ng 驅動程式：

因為廣播負載策略需更動無線網路卡的驅動程式，由於 Intersil 無線網路卡晶片在 Linux 上開放驅動程式原始碼且支援介面管理工具，因此我們選擇 Intersil 無線網路卡晶片在 Linux 上實作，我們在待測的工作站上安裝 Linux Redhat8.0，以及從以下網址下載 Intersil 無線網路卡晶片的支援程式套件，

*ftp://ftp.linux-wlan.org/pub/linux-wlan-ng/linux-wlan-ng-0.2.0.tar.gz*

此套件採用 PRISM-2 晶片組網路卡的系統核心模組驅動程式，它實作了 Intersil HFA384x 晶片組的硬體介面、硬體中斷服務，允許使用者層次的工具軟體從硬體取得統計與計次資料，除了驅動程式外，還包含了 Wlanctl-ng 元件，這是用來控制目前執行組態以及改變硬體狀態的管理工具，它可以透過 `Ioctl` 命令控制驅動程式的內部初始轉態。

Wlanctl-ng 命令大致上分為三類，以 `Dogllreq_` 為首的命令，如 `Dotllreq_Mibset`，主要針對 802.11 所規範的資料結構，第二類命令以 `P2req_` 為首，主要針對 PRISM-2 晶片組，第三類命令以 `Inxreq_` 為首，主要針對 Linux 驅動程式。由於目前此管理工具並未完全支援所有指令，因此只能使用現有的命令完成一些簡單的 Shell Script 檔案，如 "Scan"，"Scan\_Result"，"Join" 等，最後再將這些功能組合而成完整的 Shell Script 檔案執行動態選擇基地台的策略。以下介紹這些我們建立的一些 Shell Script 的命令。

`.\Join :`

*wlanctl-ng wlan0 p2req\_join bssid=\$1 ssid=\$2 channel=\$3 authtype=opensystem*

第一個引數為 wlan0，是 Linux 給予此無線網路卡的介面代號，第二個引數為

p2req\_join, 是命令工作站作聯結的動作, 第三個引數為 BSSID, 為基地台實體位址, 第四個引數為 SSID, 驅動程式才能夠辨識所欲加入網路的 Beacon 訊框, 第五個引數為 Channel, 指定此基地台所在的頻道, 最後一個引數為 authtype, 為身份認證類型 (authtype), 在此實作中指定為開放性認證 (opensystem), 執行此一命令後, 工作站就會加入指定的基地台。

.\Scan :

```
wlanctl-ng wlan0 dot11req_scan bsstype=infrastructure  
bssid=00:00:00:00:00:00 ssid= scantype=passive probedelay=0  
channellist=01:02:03:04:05:06:07:08:09:0A:0B:0C:00:00 minchanneltime=200  
maxchanneltime=250
```

第一個引數為 wlan0, 是 Linux 給予此無線網路卡的介面代號, 第二個引數為 dot11req\_scan, 是命令工作站作掃瞄的動作, 第三個引數為網路類型 (bsstype), 在此指定為基礎架構型網路 (infrastructure), 第四個引數為 BSSID, 為基地台實體位址, 在這並不指定, 第五個引數為 SSID, 在這也不指定, 第五個引數為掃瞄型態 (scantype), 在此指定為被動掃瞄 (passive), 第六個引數為頻道列表 (channellist), 列出所要掃瞄的頻道, 最後兩個引數為 authtype, 掃瞄每個特定頻道時, 所使用的最小與最大的時間量, 執行此一命令後, 工作站就會開始掃瞄頻道列表存在的基地台, 並將結果記錄在硬體暫存器。

\Scan\_result :

```
wlanctl-ng wlan0 dot11req_scan_results bssindex=$1
```

第一個引數為 wlan0, 是 Linux 給予此無線網路卡的介面代號, 第二個引數為 dot11req\_scan\_results, 是命令工作站從硬體暫存器讀出掃瞄的結果, 第三個引數為暫存器指標的索引, 由索引可以找到記錄在暫存器中基地台的資料。執行此一命令後, 工作站就會開始讀出暫存器裡存放基地台的資料, 並將結果列出。

### 5.2.2 金麗科技開發用電路板

此電路板是採用金麗科技 (RDC) R2020 晶片組成, 且有一 PCMCIA 插卡式介面, 在這電路板上的韌體已有 RDC 發展的無線基地台的嵌入式系統程式, 支援 Intersil 無線網路卡驅動程式, 因此我們將此電路板外插 Intersil 無線網路卡來作為實作的無線基地台, 但為了符合我們實作的策略仍須要修改系統的原始碼, 在基地台端需要修改驅動程式中的三個副程式及新增一個時間處理事件, 要修改的程式簡述於下 :

接收副程式 (RX):

計算接收訊框的長度除以接收時的傳送速率, 可以得到此訊框的傳送時間, 並累加傳送時間。若此接收訊框的來源位址為已聯結之工作站位址, 則將實際運作的工作站數量加一, 但不計算已重覆記錄之工作站。

傳送副程式(TX):

計算傳送訊框的長度除以傳送時的傳送速率，可以得到此訊框的傳送時間，並累加傳送時間。若此傳送訊框的目的位址為已聯結之工作站位址，則將實際運作的工作站數量加一，但不計算已重覆記錄之工作站。

聯結副程式(Association):

基地台在接收到聯結需求訊框(Association Request)後會送出聯結回覆訊框(Association Response)，在送出聯結回覆訊框後馬上接著將負載訊框也傳送出去，負載訊框的目的地位址填入 FF:FF:FF:FF:FF:FF，此負載訊框為廣播訊框，而資料內容則填入圖 3.3 負載資料結構的相關資訊。

時間處理事件(Timer Task):

時間處理事件會週期性計算出基地台的負載並更新負載的資訊，且將基地台相關負載資訊以及實際運作工作站數量填入在負載資料結構內，之後再將接收時間及傳送時間歸零，實際運作的工作站的個數也歸零，使得接收程式和傳送程式得以重新累計新的負載資訊。

### 5.2.3 PCATTCP 產生 TCP 流量的工具

利用 PCATTCP 軟體可產生 TCP 的流量，但我們希望模擬製造出網路流量會隨著時間改變且呈現不同的變化，所以修改此 PCATTCP 的原始碼，新增一參數(-J)，藉著控制傳送封包數量和傳送封包間隔，產生固定三種吞吐量 1. 0.3Mbps 2. 1.3Mbps 3. 2.3Mbps，各別執行 2 分鐘，再依設定的隨機陣列 Rand\_Tbl(n)= { ( 1, 2, 3 ), ( 1, 3, 2 ), ( 1, 3, 2 ), ( 1, 3, 2 ), ( 1, 3, 2 ), ( 1, 3, 2 ) }，隨機選擇一組陣列流量，如此一來在固定一段時間內所產生出的總流量，會比完全以隨機方式產生出的流量固定，亦即差異值會在一定的範圍內，才不會使每次測試結果產生較大的誤差，另外若不同的基地台流量差異值大，也可以讓工作站根據負載的資訊，更容易依負載的大小選擇基地台。

假設 PCATTCP 流量產生器在兩個基地台上產生負載，在理想狀態下，希望能產生如圖 5.1 的流量狀態，使得兩個基地台在每個時間點負載差異皆很大，但事實上在實作時，難以將時間控制同步，而達到此理想狀態，但只要時間稍微拉長，差異的總時間相對也變長，累積的結果也就對於我們的實作影響就不會太大。

接下來的實驗將會使用以下的三個批次檔，並可以在 Windows 下執行，PCATTCP 參數如附錄二所示

在接收端編輯 RX.bat :

```
pcattcp -R
```

(-R)的參數，代表是接收者

在傳送端編輯 TX.bat:

```
:begin  
pcattcp -i 192.168.30.12 -t -m 2 -x -J  
goto begin
```

第一個引數為(-i)後面指定接收者 IP 位置，第二個引數(-t)代表是傳送者，第三個引數(-m)指定 TCP 連線個數(Multi Client Connect)，可利用此項調整整體吞吐量大小，第四個引數(-x)資料傳完後會停止，第五個引數(-J)依據隨機流量陣列(Rand\_tbt)隨機選擇一組傳送。

在傳送端編輯 TX1.bat:

```
pcattcp -i 10.0.0.5 -t -c -j [interval]
```

第一個引數為(-i)後面指定接收者 IP 位置，第二個引數(-t)代表是傳送者，第三個引數(-c)是連續傳送不會停止，第四個引數(-j)重送封包的間隔時間，可利用此項調整整體吞吐量大小。

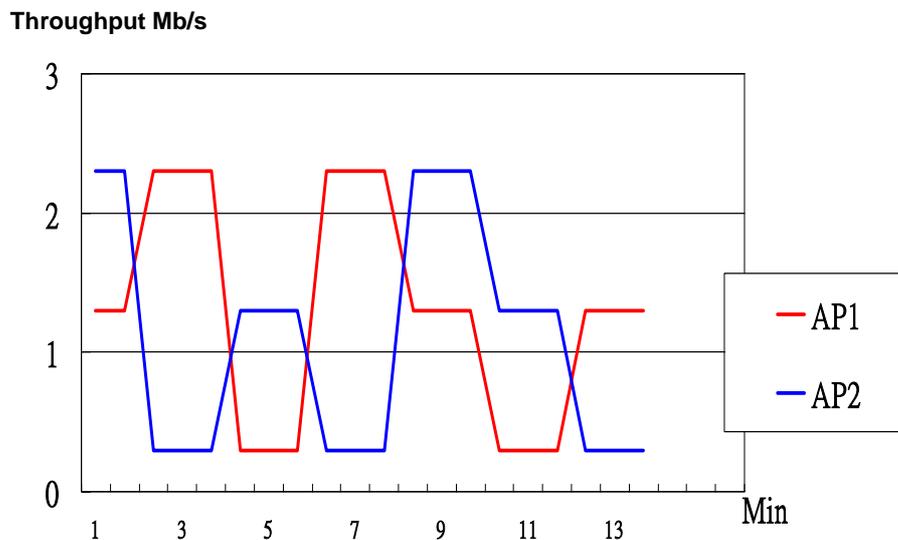


圖 5.1 : PCATTCP 流量狀態圖

### 5.3 FTP 下載檔案的測試環境

我們建構如圖 5.2 的環境從 FTP 伺服器下載 466MB 檔案，當下載完成時，比較各個策略平均的傳送速率。在實驗中都是使用 802.11b 的基地台與網路卡，基地台 AP1，工作站 STA1，工作站 STA2 使用頻道 1 傳送資料。基地台 AP2，工作站 STA3，工作站

STA4 使用頻道 11 來傳送資料。工作站 STA1, STA3 執行 TX.bat 傳送訊框。工作站 STA2, STA4 執行 RX.bat 接收訊框。執行 TX.bat 與 RX.bat 的目的在於製造網路流量，使得兩個不同頻道的基地台 AP1 和基地台 AP2 產生不同負載。而待測的工作站 STA5 將會執行各別策略的 Shell Script 檔案。

在測試廣播負載策略時，將兩塊 RDC 電路板作為基地台 AP1 和 AP2。在測試探測訊框策略時，使用市面上的 Cisco Aironet340 和 PCi 無線基地台作為 AP1 和 AP2。在測試傳統方法的策略時，我們將工作站 STA5 固定聯結至基地台 AP1 或 AP2 來測量檔案下載的傳輸速率。此外我們在省略測試廣播負載策略和探測訊框策略時每回合都需要作重新掃描的動作，原因有三，一是基地台數量為兩台並不會更動，二是 Wlanctl-ng 掃描的命令(dot11req\_scan)並非每次都可以準確偵測到兩台基地台，三是 Wlanctl-ng 與驅動程式的搭配仍有問題，尤其資料正在傳送時又啟動 Wlanctl-ng 命令，造成系統反應不正常，故我們只在第一次執行掃描後接著省略此步驟，我們期望在新版本的驅動程式會修正此問題。

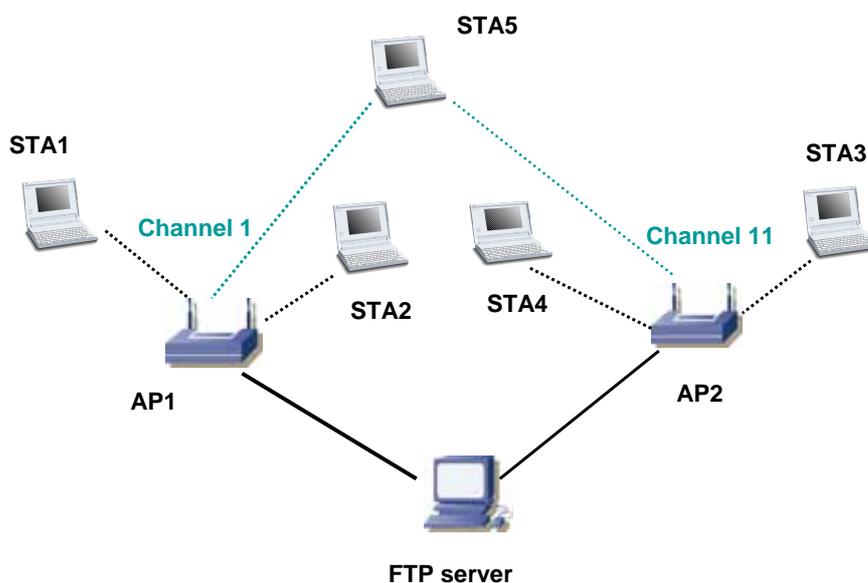


圖 5.2：FTP 下載檔案的測試環境

#### 5.4 FTP 下載檔案的測試結果

從圖 5.3 及表 5.1 的測試結果裡我們可看到廣播負載策略和探測訊框策略大約比傳統方法多獲得 10%~20% 的效能。雖然可以看到策略的效果，但在固定的測試環境，只能驗證基地台負載此項因素的影響，無法驗證其他影響選擇網路負載的因數，如工作站數目，傳送速率，基地台間干擾的問題等。此外多獲得的效能，並不如預期的多，其原因有，頻道須作切換且須重新聯結基地台所耗費的時間，而且當切換頻道時，會造成原有的基地台聯結斷線，TCP Congestion 機制便會啟動，使得 FTP 下載的吞吐

量會馬上下降，等到成功重新聯結至新的基地台後吞吐量才又上升，導致影響下載的傳送速度變慢，這個切換基地台的動作，稱之為 Handoff，此外，有時 Handoff 時間過久，導致 FTP 伺服器等待逾時，而中斷傳輸，使得要重新實驗，這是另一個問題，並不在本文中討論。而我們提出的兩個策略因使用的基地台不同而無法作比較，但從實驗過程中可以發現，使用探測訊框策略偶爾會選擇到負載較重的基地台，其原因可以從以下的實驗來解釋。

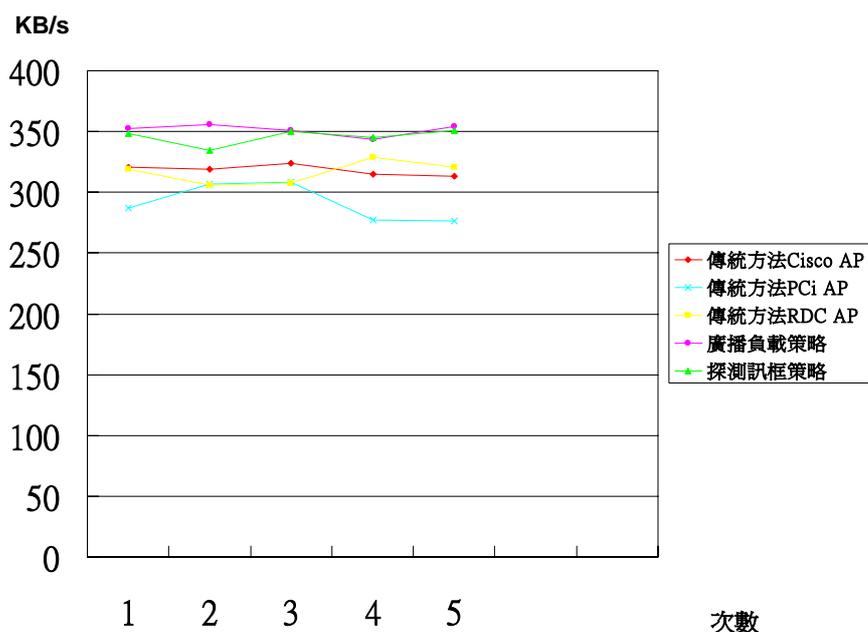


圖 5.3：FTP 下載檔案的測試結果

表 5.1：FTP 下載檔案的平均值

下載 466MB 檔案	平均吞吐量	minutes
傳統方法 Cisco AP	318.34KB/s	24.98
傳統方法 PCi AP	291.07KB/s	27.32
傳統方法 RDC AP	316.45KB/s	25.13
廣播負載策略	351.51KB/s	22.62
探測訊框策略	345.72KB/s	23.00

## 5.5 探測訊框策略 PING 的準確度測試環境

探測訊框策略主要是在網路應用層下達 PING Request 封包並開始計時，計算接收到 PING Reply 封包回覆時這段時間的長度，來測量網路負載的狀態，因此我們稍微修改探測訊框策略的 Shell Script 檔案，主要修改部分為每一次皆會作掃描 (scanning),且只做一次 PING 的動作，以下以此 Shell Script 做個簡單的實驗，分

析所測得的時間與網路負載的相關性與準確性。

我們建構如圖 5.4 的環境，從 PC 傳送資料到工作站 STA2，目的在於製造網路流量，工作站 STA2 在 Windows XP 下執行 RX.bat，PC 端執行 TX1.bat，TX1.bat 可以調整 PCATTCP 中(-j)的參數，(-j)的參數控制傳送每筆封包與封包間的時間，因此也造成不同的負載流量經無線路由器至工作站 STA2，工作站 STA1 則執行 PING 的命令，PING 的目的地為無線路由器來測量封包回覆的時間，亦即測量此時網路負載的狀態，但此環境只有一台基地台，並無其他訊號干擾，且所有裝置的位置皆擺放靠近，避免訊號強度衰減，造成傳送速率變慢，因此基地台訊號干擾與傳送速率這兩項影響網路負載因數不會改變，我們也控制了基地台的負載大小，只剩實際運作的工作站個數此項變異的因數，雖然在環境中有兩台工作站，但仍無法掌握實際運作的工作站個數，在這情況下我們調整基地台負載的大小觀察回覆時間與整體網路負載呈現什麼樣的關係。

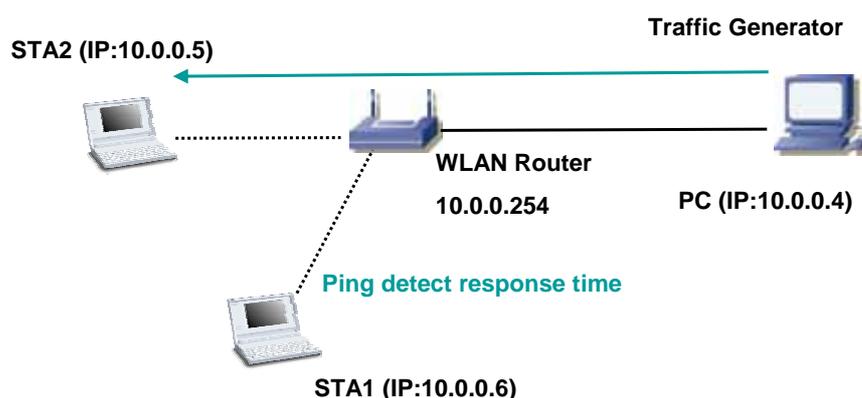


圖 5.4：探測訊框策略的測試環境

## 5.6 探測訊框策略 PING 的準確度測試結果

我們經由觀察工作站 STA2 在 Windows XP 中的網路流量圖可以得知目前的負載流量以百分比表示，當工作站 STA2 接收到最大的吞吐量為 11Mbps，即會顯示 100%，但在實際的 802.11b 情況下是不可能發生的。我們觀察到有以下以點現象：

1. 從圖 5.5(a)(b)(c)(d)(e) 中可以看到分別在五種流量時，各別作 120 次 PING 的偵測得到的回應時間分佈圖，當流量改變，測得的時間也會改變。
2. 在表 5.2 中，第一列為參數(-j)的數值，控制傳送每筆封包與封包間的時間時

間，數值愈小代表封包間隔較密集，基地台所承受的負載也愈大，可以從表 5.2 第二列中工作站 STA2 吞吐量得知，相對的在表 5.2 第三列可觀察到當基地台負載變大，平均的回應時間也變長。

3.圖 5.5(b)(c)(d)(e)都較圖 5.5(a)中的點較為稀疏，原因是工作站掃瞄時並沒有找到基地台，因此沒有把結果描繪上去。

4.圖 5.5(a)為負載是 0%的狀況下所測的時間，點的分布範圍密集，大多集中在 2ms 到 3ms，圖 5.5(d)(e)若以平均回應時間為基底相對於圖 5.5(b)(c)比較來的集中，這有可能是因為負載較重時，工作站 STA2 在大多數的時間都是扮演實際運作的工作站而加入競爭，因此偵測到的時間也較為固定，而圖 5.5(b)(c) 中工作站 STA2 在大多數時間並無加入競爭，因此偵測的時間長短範圍較大，因此這也說明當使用探測訊框策略時並非以一次測得的時間作決定，而是 PING 三次後得到的平均時間，再與前次測得的時間作加權運算以免誤判。

5.有時會產生較大的偵測回應時間，這個原因有可能是訊框產生碰撞，而造成重傳訊框。

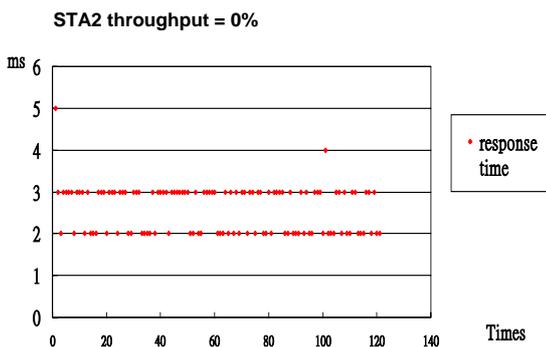


圖 5.5(a)：負載為 0%時  
探測時間分布圖

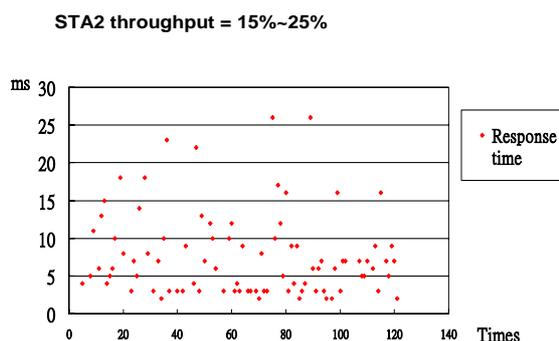


圖 5.5(b)：負載為 15%~25%時  
探測時間分布圖

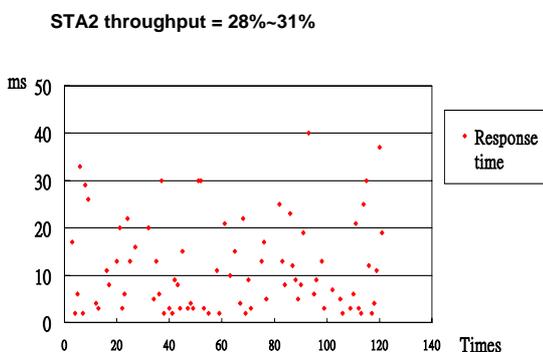


圖 5.5(c)：負載為 28%~31%時  
探測時間分布圖

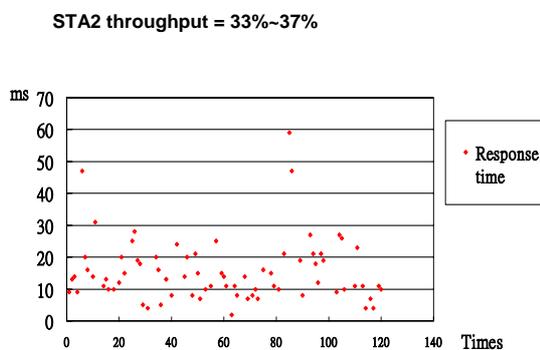


圖 5.5(d)：負載為 33%~37%時  
探測時間分布圖

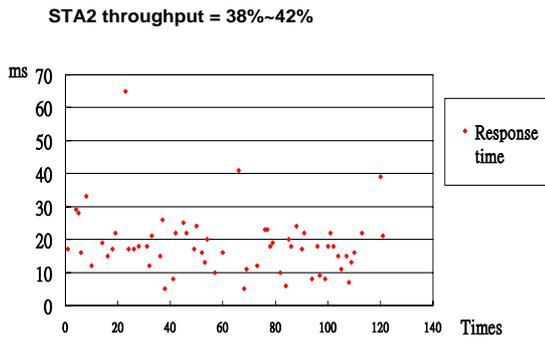


圖 5.5(e) : 負載為 38%~42%時探測時間分布圖

表 5.2 : 探測時間的平均值

Interval(-j)	NA	1800	1500	1000	750
STA2 throughput	0%	15%~25%	28%~31%	33%~37%	38%~42%
Average response time	2.59 ms	7.62 ms	11.76 ms	15.52 ms	18.45 ms

## 第六章 結論

我們提出兩種策略來解決動態選擇基地台的問題，雖然這兩種策略都有實作，但實作時，都必須要聯結基地台加入其 BBS，才可獲得相關的負載資訊，但以策略的精神卻是不須要此步驟，所以們認為若市場認同此種機制，則強烈建議可以實作在晶片中，一方面降低系統額外的負荷，也可大幅增進系統功能，雖然無法評論以晶片設計難易度，選擇那種策略來實現，但儘可能以含蓋較多影響網路負載因素來作考量，若是如此，我們已可想見未來的無線網路環境一定能夠更便利，更有效率。



## 參考文獻

- [1] Shiann-Tsong Sheu and Chih-Chiang Wu, “Dynamic Load Balance Algorithm (DLBA) for IEEE 802.11 Wireless LAN” , Tamkang Journal of Science and Engineering, vol1 2, No 1, pp.45-52 (1999).
- [2] I. Papanikos and M. Logothetis, “A Study on Dynamic Load Balance for IEEE 802.11b Wireless LAN” , Proc. 8<sup>th</sup> International Conference on Advances in Communication & Control, COMCON 8, Rethymna, Crete, June, 2001.
- [3] Chih-Liang Chou, and Shie-Yuan Wang, “Dynamic Load Balancing for IEEE 802.11 Wireless Networks” , Master thesis, National Chiao Tung University, Hsinchu, Taiwan, 2002.
- [4] Gerard Cervelló, Sunghyum Choi, Stefan Mangold\*, and Amjad Soomro, “Dynamic Channel Selection (DCS) Scheme for 802.11” , IEEE 802.11-00/195r2, September 2000.
- [5] Shiann-Tsong Sheu, Yihjia Tsai and Jenhui Chen, “MR<sup>2</sup>RP: The Multi-Rate and Multi-Range Routing Protocol for IEEE 802.11 Ad Hoc Wireless Networks” . Wireless Networks 9(2), 165-177 (2003)
- [6] Matthew S.Gast 著，802.11 無線網路技術通論，黃裕彰譯，美商歐來禮，台北，民國九十二年。
- [7] IEEE Std 802.11 Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, June (1997).

## 附錄一

```
echo "pgdet [router IP] [essid]"
ifconfig eth0 down
# init
echo > load.txt
count=0
count_rst=0
old_bssid=00:00:00:00:00:00
now_bssid=00:00:00:00:00:00
num=0

#start
while true; do
    if [ $num != 2 ]; then
        num=$(./foundap)
        echo "Found $num Access Point"
    else
        break
    fi
    sleep 5
done
while true; do
    echo $(date +%X%N)
    # arp -d $1
    i=0
    ap_num=0

    echo ===== scan =====
    #echo ===== Only list ESSID = $2 =====
    for i in 0 1 2 3 4 5 6 7 8 9 10 11 12
    do
        if [ $num -gt $i ]
        then
            ssid=$(./scan_result $i | grep -m1 " ssid" |cut -c9-) # grep the ssid
            bssid=$(./scan_result $i | grep -m1 bssid |cut -c9-) # grep the bssid
            chanl=$(./scan_result $i | grep -m1 chan |cut -c13-) # grep the channel
            ssid=${ssid%"} # delete the last character
        fi
        #echo "ssid=$ssid"
    done
done
```



```

#echo "para2=$2"
rsptime=$(ping $1 -l1 -c3 -w5 | grep -r avg | cut -f5 -d'/' | cut -f1 -d'.')

echo "now AP response time=$rsptime ms"
echo =====
if [ $rsptime -lt 25 ]; then
    echo "keep this AP...don't change"
    break
fi

if [ $ssid = $2 ]; then

    ap_num=$((ap_num+1))
    ./join $bssid $ssid $chanl >> load.txt          # associated every AP
    now_bssid=$bssid

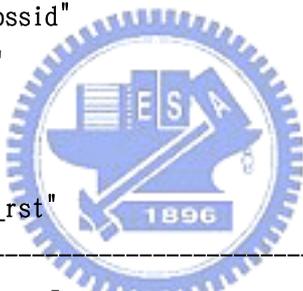
    echo " ssid=$ssid"
    echo " bssid=$bssid"
    echo " chanl=$chanl"
#    ping $1 -c1 -w2 -q >> load.txt
    rsptime=$(ping $1 -l1 -c3 -w5 | grep -r avg | cut -f5 -d'/' | cut -f1 -d'.')
    sleep 5
    if [ -z $rsptime ]; then
        rsptime=99
        echo " response time=timeout"
        count_rst=$((count_rst+1))
        ./reset
        num=$(./foundap)
        break
    else
        if [ $rsptime -gt 99 ]; then
            rsptime=99
        fi
        echo " response time=$rsptime ms"
        if [ $rsptime -lt 25 ]; then
            echo "$bssid $ssid $chanl $rsptime" >> ap_new.txt
            break
        fi
    fi
fi

```

```

#      echo "*****ARP TABLE*****"
#      arp -i wlan0
#      arp -d $1
      echo "$bssid $ssid $chanl $rsptime" >> ap_new.txt
      echo -----
      fi
    else
      break
    fi
done
if [ $ap_num != 0 ]; then
  echo "-----join-----"
  sort_list=$(./sortrsp $ap_num)
  bssid=$(echo $sort_list | cut -f1 -d' ')
  ssid=$(echo $sort_list | cut -f2 -d' ')
  chanl=$(echo $sort_list | cut -f3 -d' ')
#   echo "now_bssid=$now_bssid"
  echo "new_bssid=$bssid"
  echo "ssid=$ssid"
  echo "chanl=$chanl"
  echo "count_rst=$count_rst"
  echo "-----"
  if [ $now_bssid != $bssid ]; then
#    echo "----- change...-----"
    ./join $bssid $ssid $chanl >> load.txt      # associated good AP
    now_bssid=$bssid
  fi
  rm ap_new.txt
fi
sleep 45
done

```



## 附錄二

```
Usage: pcattcp -t [-options] -i host [ < in ]
       pcattcp -r [-options] > out]
Common options:
-l ### length of bufs read from or written to network (default 8192)
-u      use UDP instead of TCP
-p ### port number to send to or listen at (default 5001)
-s      toggle sinkmode (enabled by default)
       sinkmode enabled:
       -t: source (transmit) fabricated pattern
       -r: sink (discard) all received data
       sinkmode disabled:
       -t: reads data to be transmitted from stdin
       -r: writes received data to stdout
-A      align the start of buffers to this modulus (default 16384)
-O      start buffers at this offset from the modulus (default 0)
-v      verbose: print more statistics
-m ### multi client connect. ## the client number. default 1
-g ### a thread repeat ## times. default 1
-x      broken connect but just return ,don't esc.
-e ### the interval between threads (ms) default 100ms
-j ### the interval between each buffer send. default 0
-d      set SO_DEBUG socket option
-b ### set socket buffer size (if supported)
-L X    LogFile name. default ttp.log .
-f X    format for rate: k,K = kilo(bit,byte); m,M = mega; g,G = giga
-c      -t: send continuously
       -r: accept multiple connections sequentially
-R      concurrent TCP/UDP multithreaded receiver
Options specific to -t:
-n ### number of source bufs written to network (default 2048)
-D      don't buffer TCP writes (sets TCP_NODELAY socket option)
Options specific to -r:
-B      for -s, only output full blocks as specified by -l (for TAR)
-I      "touch": access each byte as it's read
```

