

國立交通大學

資訊工程系所

碩士論文

IEEE 802.11 無線網路動態頻道分配策略



Dynamic Channel Allocation Schemes for
IEEE 802.11 Wireless LAN

研究生：戴郁芬

指導教授：曾煜棋 教授

中華民國九十四年八月

IEEE 802.11 無線網路動態頻道分配策略

Dynamic Channel Allocation Schemes for IEEE 802.11 Wireless

LAN

研究生：戴郁芬

Student : Yu-Fen Tai

指導教授：曾煜棋 教授

Advisor : Yu-Chee Tseng

國立交通大學

資訊工程學系



Submitted to Department of Computer Science and Information Engineering

College of Electrical Engineering and Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science and Information Engineering

August 2005

Hsinchu, Taiwan, Republic of China

中華民國九十四年八月

IEEE 802.11 無線網路動態頻道分配策略

學生：戴郁芬

指導教授：曾煜棋 教授

國立交通大學資訊工程系(研究所)碩士班



隨著無線技術的日新月異，無線網路已經成為佈置區域網路的普遍選擇。通常為了使整棟建築佈滿無線訊號，網路管理員會選擇將無線基地台緊密佈置。但是另一方面在 IEEE 802.11 基礎架構模式(Infrastructure mode) [1]下，使用同一頻道的相鄰無線基地台可能會發生訊號干擾的情形。由於頻道數目的限制以及使用者會到處移動的關係，靜態的頻道規劃並不足以解決問題。如何動態且適當的調整無線基地台的頻道以增進整個網路的效能，是我們想要討論的主題。

在本論文中，我們提出兩個中央控制的演算法，根據無線基地台的使用情況來動態調整它的頻道。我們實作了其中一個演算法，並且透過實驗來比較本論文的方法及其他方法的效能優劣。實驗結果顯示，我們的方法的確增進了網路的效能，並且比其他方法表現更好。

關鍵字: IEEE 802.11，基礎架構模式，無線網路，頻道分配，干擾


Dynamic Channel Allocation Schemes for IEEE 802.11 Wireless LAN

Student : Yu-Fen Tai

Advisors: Prof. Yu-Chee Tseng

Institute of Computer Science and Information Engineering
National Chiao-Tung University

ABSTRACT



With the rapid evolution of wireless technology, WLAN gradually becomes a common solution of setting up a local area network. In order to cover the whole building with wireless signals, the network administrators often choose to deploy the access points (APs) densely. On the other hand, under the IEEE 802.11 infrastructure mode, there may be interference between the nearby access points if they use the same channel. Static channel assignment is not good enough because of the limitation of channels and the mobility of users. How to change the channels of access points dynamically to speed up the throughput of the whole network is the main issue that we want to discuss.

In this thesis, we propose two centralized schemes to assign the channels of access points dynamically according to the AP utilizations. We also implement one of our schemes and compare the performance of it with other methods by experiments. The experiment result shows that our methods indeed increase the network throughput a lot and achieve better performance than others.

Keyword: IEEE 802.11, infrastructure mode, WLAN, channel assignment, interference

Acknowledgements

My advisor, prof. Yu-Chee Tseng, is the first one I would like to express my gratitude to. With the wonderful research conditions he provided and his attentive instructions, I came to discover the pleasure of research. Discussing with him benefited me in many ways. I would also like to thank all HSCC members for helping me to do the experiments and providing me some generous advices.



Contents

摘要	i
Abstract	ii
Acknowledgements	iii
Contents	iv
List of Figures	vi
List of Tables	viii
一、 簡介	1
1.1 關於 802.11	1
1.2 頻道與傳輸速率	1
1.3 802.11 的兩種模式	2
1.4 AP 頻道切換問題	3
二、 相關研究	4
2.1 Ad Hoc mode 下的相關研究	4
2.2 Infrastructure mode 下的相關研究	4
三、 動態頻道分配策略	6
3.1 動機及問題定義	6
3.2 系統基本原理與架構	9
3.3 名詞定義	10
3.4 演算法流程	13
3.4.1 資料結構	13
3.4.2 AP 端流程	13
3.4.3 Server 端流程策略一：動態頻道切換演算法	14
3.4.4 Server 端流程策略二：考量 Neighbor's Neighbor	16
四、 實驗結果	18
4.1 實驗一	18
4.2 實驗二	23
4.3 實驗三	29

五、結論與未來工作	35
參考文獻	36
自傳	38



List of Figure

圖 1.1	802.11 頻道示意圖	1
圖 1.2	Infrastructure Mode 示意圖	2
圖 3.1	兩 AP 之間的距離大於 3R(傳輸距離)	6
圖 3.2	2R<distance(Ap1,Ap2)<=3R 兩 AP 之間的距離在 2R~3R(傳輸距離)之間	7
圖 3.3	R<distance(Ap1,Ap2)<=2R 兩 AP 之間的距離在 R~2R(傳輸距離)之間	7
圖 3.4	distance(Ap1,Ap2)<=R 兩 AP 之間的距離小於 R(傳輸距離)	8
圖 3.5(a)	STA1 連接 AP1 , 傳輸量 500KB/S STA2 連接 AP2 , 傳輸量 500KB/S	9
圖 3.5(b)	STA1 移動到 AP3 , 傳輸量 500KB/S STA2 連接 AP2 , 傳輸量 500KB/S	9
圖 3.6	系統架構示意圖	10
圖 3.7	AP1 上只有一個 Client , 流量為 600KB/S。AP2 上則有兩個 Client , STA2 的流量是 200KB/S , STA3 的流量也是 200KB/S	11
圖 3.8	AP1 上只有一個 Client , 流量為 600KB/S。AP2 上則有兩個 Client , STA2 的流量是 100KB/S , STA3 的流量也是 100KB/S	12
圖 3.9	AP 端演算法流程圖	14
圖 3.10	Server 端演算法流程圖	15
圖 4.1	實驗一 動態頻道切換 示意圖	18
圖 4.2	實驗一 動態頻道切換 Client A 實測流量圖 (1)	19
圖 4.3	實驗一 動態頻道切換 Client B 實測流量圖 (1)	19
圖 4.4	實驗一 動態頻道切換 Client A 實測流量圖 (2)	20
圖 4.5	實驗一 動態頻道切換 Client B 實測流量圖 (2)	20
圖 4.6	實驗一 不切換頻道 示意圖	21
圖 4.7	實驗一 不切換頻道 Client A 實測流量圖	22
圖 4.8	實驗一 不切換頻道 Client B 實測流量圖	22
圖 4.9	實驗二 動態頻道切換 示意圖	23
圖 4.10	實驗二 動態頻道切換 Client C 實測流量圖	23
圖 4.11	實驗二 動態頻道切換 Client B 實測流量圖	24
圖 4.12	實驗二 動態頻道切換 Client A 實測流量圖(斷線)	25
圖 4.13	實驗二 動態頻道切換 Client A 實測流量圖(成功)	25
圖 4.14	實驗二 不切換頻道 示意圖	26
圖 4.15	實驗二 不切換頻道 Client B 實測流量圖	27
圖 4.16	實驗二 不切換頻道 Client A 實測流量圖	27

圖 4.17	實驗二 根據 AP 上的 client 數切換 示意圖	28
圖 4.18	實驗三 動態頻道切換 示意圖	29
圖 4.19	實驗三 動態頻道切換 Client A 實測流量圖	30
圖 4.20	實驗三 動態頻道切換 Client B 實測流量圖	30
圖 4.21	實驗三 動態頻道切換 Client A 實測流量圖	31
圖 4.22	實驗三 動態頻道切換 Client B 實測流量圖	31
圖 4.23	實驗三 根據 AP 的 Throughput 切換頻道 示意圖	32
圖 4.24	實驗三 根據 AP 的 Throughput 切換頻道 Client A 實測流量圖	33
圖 4.25	實驗三 不切換頻道 示意圖	34



List of Tables

Table 4.1	實驗一 不切換頻道 流量統計表	21
Table 4.2	實驗二 動態頻道切換 流量統計表	24
Table 4.3	實驗二 不切換頻道流量統計表	26
Table 4.4	實驗二 根據 AP 上的 client 數切換 流量統計表	28
Table 4.5	實驗三 動態頻道切換 流量統計表	29
Table 4.6	實驗三根據 AP 的 Throughput 切換頻道 流量統計表	32
Table 4.7	實驗三 不切換頻道流量統計表	34
Table 5.1	總結 效能比較表	35



第一章 簡介

1.1 關於 802.11

1985 年，美國聯邦通訊委員會 (FCC ; Federal Communications Commission) 決定開放三個 ISM 頻帶 (Industrial Scientific Medical bands)，即 902~928MHz，2.4~2.483GHz、5.725~5.875GHz 等三個頻帶。由於 ISM band 是不需要申請便能使用的頻帶，因此此一動作讓許多無線通訊產品的發展更加蓬勃。到了 90 年代初，使用 ISM 頻帶的通訊產品紛紛出現在市場上，為了使各產品之間能夠互相通訊，標準的制訂就成了重要的工作，而後便有 IEEE 802.11 [1]無線區域網路 (wireless LAN) 的標準產生。

美國電器電子技術協會(IEEE)於 1997 年所制定的 IEEE 802.11 [1]無線網路標準，主要是要讓各種符合該標準的無線網路設備利用 2.4GHz 頻帶傳遞無線電訊號，以達到交換資訊之目的。IEEE 802.11 [1]標準採用 2.4GHz 頻帶為共用頻道，802.11b [3]及 802.11g [4]標準也是採用 2.4GHz 的 ISM band，而 802.11a [2]標準則是用 5GHz 的 ISM band。



1.2 頻道與傳輸速率

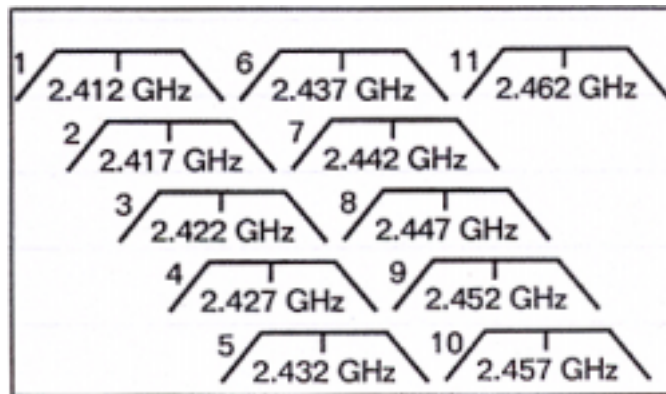


圖1.1 802.11頻道示意圖

從圖1.1可以看出，802.11b標準訂定不重疊的頻道有三個，分別是1、6、11三個頻道。802.11g在頻道的使用上跟802.11b相同。802.11b訊號速率最低為1 Mbps，最高則能達到11 Mbps；802.11a標準和802.11g最高提供54 Mbps訊號速率，比802.11b的訊號速率快五倍。

不過實際上，傳送資料並不能達到理想中的訊號速率。因為協定中包含一些資料格式，傳送資料都必須符合這些格式才能溝通，因此造成了負擔。也就是除原始資料外的一些添加資訊，例如協定標頭 (protocol header)、訊框時距 (Frame Interval times)、錯誤與流程控制(integrity)、確認 (acknowledgment)等，這些都造成了額外負擔，使得實際能達到的傳輸速率，以802.11b為例約只有6Mbps，而非訊號速率的11Mbps；802.11a或802.11g裝置的實際傳輸期望值也在30 Mbps左右，而不是54 Mbps的訊號速率。

1.3 802.11 的兩種模式

在 IEEE 802.11 中，制訂了兩種不同類型的無線區域網路模式：

模式 1: Ad Hoc Mode

Ad Hoc Mode 主要指的便是沒有固定基礎架構的無線網路，能夠隨時隨地架設起來，讓多人即時通訊，一旦大家離開，網路便又消失，任兩工作站之間都可直接通訊，適合用於臨時會議、戰場或者山區等不適合佈置基礎架構的地方。

模式 2: Infrastructure Mode

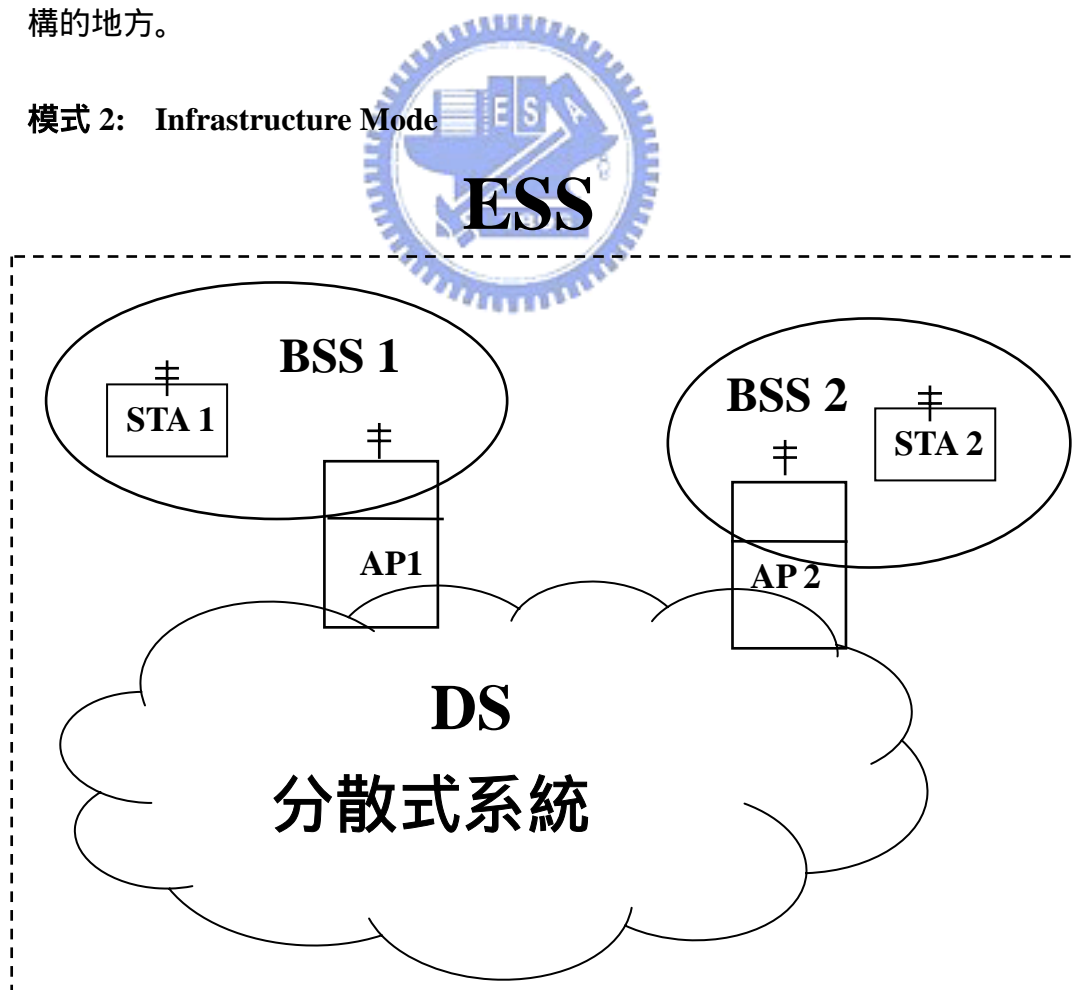


圖 1.2 Infrastructure Mode 示意圖

如圖 1.2 所示，所謂的 Infrastructure Mode 通常指的工作站(mobile station)可經過單一無線連接(Single hop)聯繫上一個現存的有線網路。在這種網路中，工作站所連結上的有線網路的特殊節點，稱作無線網路基地台 (AP)。AP 和工作站便是組成一個 Infrastructure Mode 無線網路的基本單位，稱作 basic service set(BSS)。不同的 BSS 透過 distribution system (DS)互相連結，每個工作站都可以透過 AP 來存取 DS，以目前的佈置來說，DS 多半是一個現存的有線網路，而整個系統包括 DS 和所有的 BSS 就成為一個 extended service set (ESS)。

1.4 AP 頻道切換問題

從前面的敘述我們可以知道，所有的 mobile station 都是透過無線連接 AP 存取 DS。通常 AP 會有一個 IEEE 802.11 LAN 的無線介面及一個 IEEE 802.3 LAN 的有線介面，而 IEEE 802.11b 實際上的傳輸速率約莫只在 6Mbps，也就是說所有連接同一 AP 的 mobile station 都必須分享這 6Mbps 的頻寬，隨著連接的 mobile station 越多能使用的頻寬就越少。

為了增加網路容量(network capacity)，提供更多的頻寬，設置多個 AP 是很自然的想法，並且由於無線網路訊號不穩定，常常會因為距離及環境的關係而有訊號衰減的問題，為了讓所有地方都能收到訊號沒有死角，佈置多個覆蓋範圍有重疊的 AP 亦是很常見的一個設置。而相鄰的 AP 如果使用同一頻道便會出現訊號互相干擾的問題，從而減低了傳輸的速率，但是 802.11b 裡面不重疊的頻道只有三個，隨著 mobile station 的移動或者斷訊又重連，所有 AP 上面的 mobile station 個數及流量是會不斷改變的，要如何動態的切換所有 AP 使用的頻道，在有限的三個頻道裡面選擇最恰當的一個，使整個網路的頻寬需求能盡量被滿足，提昇整體網路的吞吐量(throughput)，是本論文要探討的課題。接下來第二章我們將介紹相關研究，第三章則會更清楚的定義論文所要解決的問題，包括介紹整個系統的架構、定義名詞及說明演算法的流程，第四章則是我們所設計的三個實驗的結果及討論，最後提出結論和未來方向。

第二章 相關研究

關於頻道分配的問題，其實在 cellular network 上有許多相關的研究，研究如何規劃基地台的頻道以提升整體網路的效能，然而由於 cellular network 和 IEEE 802.11 無線網路有許多不同的地方，IEEE 802.11 無線網路上的流量變化要來的更大，使得 cellular network 上的研究並不能直接套用到 IEEE 802.11 無線網路環境裡面。而針對 IEEE 802.11 無線網路的相關研究，我們將之分成兩類，第一類為 Ad Hoc mode 下的研究，第二類則是針對 Infrastructure mode 的探討。

2.1 Ad Hoc mode 下的相關研究

在 Ad Hoc mode 下，每一對工作站的通訊都需要使用一個頻道，在其通訊範圍內如果有另一對工作站使用相同頻道通訊，便會對其造成干擾，由於頻道是有限的，而且工作站會任意移動，如何讓每對工作站都能找到最好的頻道來通訊，便是相關研究的課題。相關研究[9][10][11][12][13]主要集中在更改 MAC 層的協定，使其能支援多個頻道，並且企圖利用多個頻道來避免鄰近工作站彼此之間的干擾，為每個傳輸選取最好的頻道。例如文獻[11]提出了一個多頻道 CSMA 協定，採用柔性的頻道保留策略，假設每個工作站都能同時監聽所有頻道，當有封包要傳送時，工作站會選取目前沒人使用的頻道來傳送，如果有多個可用頻道，則優先選用上次傳輸成功的頻道。

參考文獻[14]則提出了一個動態頻道安排協定，由於工作站是會移動的，當其移動到另一對使用相同頻道的工作站附近時，兩對工作站的傳輸便會互相干擾，此協定希望能夠在有效率且低成本的考量下，將工作站快速更換到適當的頻道繼續通訊，靜態來看這類似一個著色問題，只是因為移動的因素，有重新分配頻道的需求。文獻[15]的環境則是工作站配合無線網路基地台組成一個網狀的網路架構，形成一個骨幹使得獲得網路服務的區域能增加，其提出一個中央控制的演算法來分配頻道及頻寬，特色在於每個節點皆配備多個無線網路介面卡並使用不同的頻道，並且強調其協定並不需要更改 MAC 層。

2.2 Infrastructure mode 下的相關研究

在無線網路 Infrastructure mode 下，通常客戶端是根據無線網路基地台的訊號強度(RSSI)來選擇無線網路基地台，所以可能很多使用者都連上同一個無線網路基地台，但是有些無線網路基地台卻沒有使用者利用。造成負載量的不平均也造成無線網路基地台頻寬的浪費。

因此有一些研究是針對這點作改善，像是文獻[7]便提出了一個叫做 DLBA 的演算法，不僅考慮了 RSSI 也考慮了每個無線網路基地台上的負載量，作出動態平衡，在模擬結果中顯示此演算法能夠將客戶端均勻的分散到每個無線網路基地台上，甚至也能夠使客戶端與其使用的無線網路基地台間的訊號強度(RSSI)的平均值最大化。

文獻[8]則提出了將無線網路上動態負載平衡的方法分成三個層面：

- (1) 無線網路基地台頻道自動選擇層級：目標在於使無線網路基地台能夠盡量分散到可使用的頻道去，將干擾減到最低。
- (2) 客戶端決定加入層級：由客戶端選擇要連接哪個無線網路基地台
- (3) 連線觀察層級：藉由觀察連線狀況，決定客戶端何時要離開無線網路基地台，漫遊要被執行。

文獻[5]則比前面的方法多考量了客戶端存取無線網路基地台頻寬的情況，設計了一套衡量負載量的標準，利用無線網路基地台被其上工作站存取的情形來推算該基地台的負載量，並配合訊號強度，來決定工作站該使用哪一個無線網路基地台，並且由基地台指示工作站進行連結。文獻[6]則是由無線網路基地台主動要求其某些工作站回報各頻道的狀況，以 RSSI 及 PER 等因素來決定要不要更換頻道，再由基地台通知與其連接的所有工作站一起更換頻道。

以上這些演算法，多半是站在客戶端的角度上，利用指揮工作站連結不同無線網路基地台來達到分攤負載量到不同無線網路基地台的效果，較少研究是從無線網路基地台的角度出發，透過動態切換頻道來達到負載量平衡及提高整體網路效能的目的。

第三章 動態頻道分配策略

3.1 動機及問題定義

因為無線網路訊號不穩定，容易受到溫度、溼度、障礙物、訊號干擾等問題影響，所以通常一環境下要佈置無線網路環境，會選擇佈置多個訊號範圍有重疊 AP 的方式來確保在每個地方訊號都有一定強度，避免訊號死角問題發生。但是這種佈置方式也會帶來訊號互相干擾的問題，雖說可以將鄰近 AP 設定在不同頻道上以避免干擾，但是往往鄰近的 AP 數量會多過於可用的不重疊頻道，因此我們想要在不改變客戶端的情況下，提出並且實作一個 AP 端的頻道切換策略，使得 AP 能根據整個網路的流量來決定自己該採用的頻道，讓整體網路 throughput 能夠提昇。

首先，我們來看兩 AP 之間隨著距離的不同，所可能導致的訊號干擾問題，我們將干擾情形分成以下四種，STA 代表與 AP 連結的客戶端(client)：

干擾情形 1：

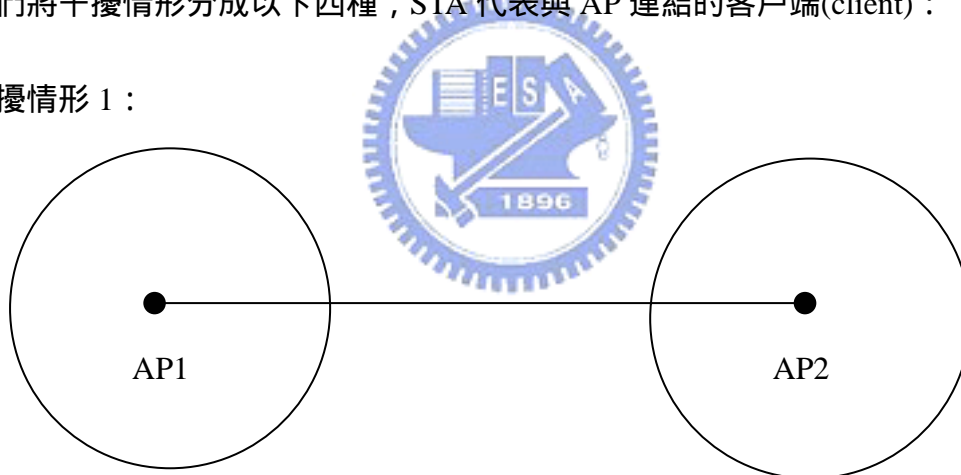


圖 3.1 兩 AP 之間的距離大於 3R(傳輸距離)

Interference (Client, Client) : none

Interference (Client, AP) : none

Interference (AP, AP) : none

如圖 3.1 所示，第一種情形是兩 AP 之間的距離大於三個傳輸半徑，在這種情況下，不論是 AP1 跟 AP2 之間或者是 AP 上的 Client 對另一方 AP，又或者兩方 AP 上的 Client 彼此之間，都沒有訊號干擾問題。

干擾情形 2 :

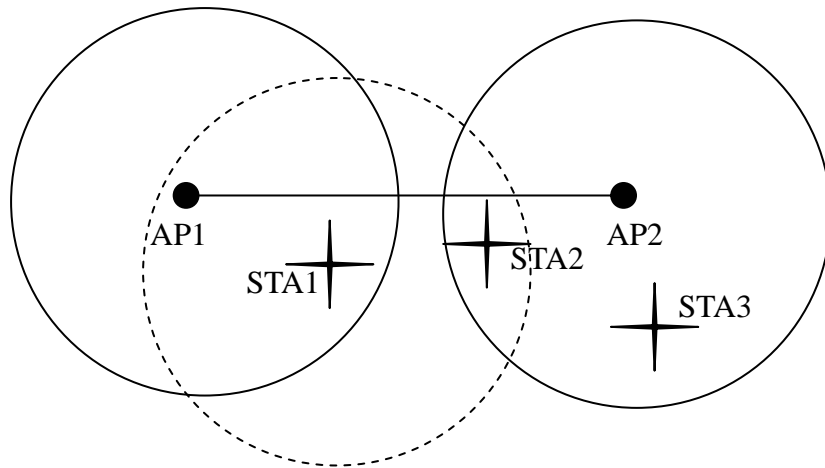


圖 3.2 $2R < \text{distance}(\text{Ap1}, \text{Ap2}) \leq 3R$
兩 AP 之間的距離在 $2R \sim 3R$ (傳輸距離)之間

Interference (Client, Client) : maybe

Interference (Client, AP) : none

Interference (AP, AP) : none

從圖 3.2 的例子，我們可以看到 STA1 跟 AP1 通訊的時候，也會影響到在 AP2 的 STA2，但是 AP2 跟 STA3 的通訊是沒有問題的。也就是這種距離下的 AP，兩方的 Client 有可能會干擾，但是這種干擾在密佈 AP 的環境下，相較於後面兩種狀況是比較不嚴重的，因此在本論文中我們主要關注後面兩種干擾對 AP 傳輸的影響。

干擾情形 3 :

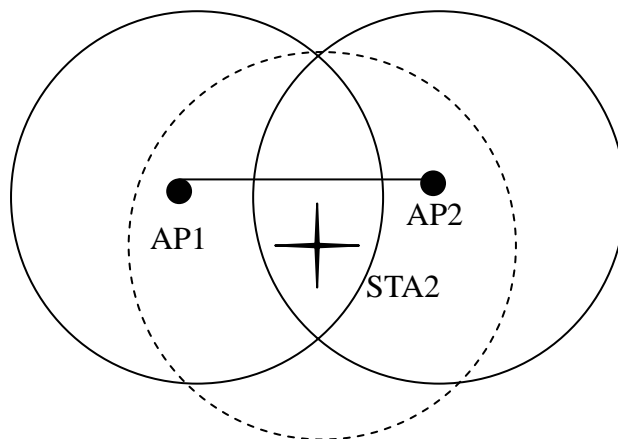


圖 3.3 $R < \text{distance}(\text{Ap1}, \text{Ap2}) \leq 2R$
兩 AP 之間的距離在 $R \sim 2R$ (傳輸距離)之間

Interference (Client, Client) : maybe

Interference (Client, AP) : maybe

Interference (AP, AP) : none

從圖 3.3 的例子，我們可以看到，即使 AP1 跟 AP2 彼此聽不到對方的訊號，但是 AP 在與其上的 Client 通訊的時候是可能會干擾到對方 AP 的。就像 STA2 在跟 AP2 通訊時，會影響到 AP1，造成 AP1 上所有的 client 端都無法進行傳輸，引起嚴重的訊號干擾問題。

干擾情形 4：

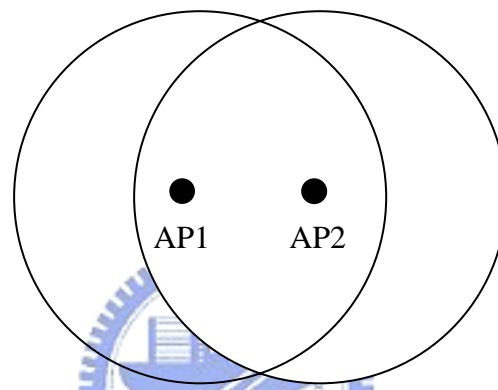


圖 3.4 $\text{distance}(\text{Ap1}, \text{Ap2}) \leq R$
兩 AP 之間的距離小於 R(傳輸距離)

Interference (Client, Client) : maybe

Interference (Client, AP) : maybe

Interference (AP, AP) : YES!

圖3.4的例子是兩個AP使用同一頻道的話互相可以聽到對方的訊號，也就是不管什麼時候，只要任一方在進行傳輸，另一個AP都不能進行通訊，使得兩個AP所能達到的最大效能實際上只相當於一個AP，也就因他們不能同時存取同一個頻道，所以是四種情形中訊號干擾最嚴重，最需要使用不同頻道的狀況。

在上述的狀況中，我們主要關注 3 和 4 這兩種情況，希望透過 AP 的頻道切換，避免這兩種情況的干擾。但是考量到各 AP 上頭的 Client 是會不斷移動的，也就是傳輸量的改變，讓我們覺得僅僅是靜態規劃每個 AP 使用的頻道是不足夠的，必須要把 AP 上面的負載(Load)考慮進來，隨著每個 AP 及其鄰近 AP 的 Load 不斷改變去動態調整整個網路中每個 AP 要使用的頻道。

下面是一個簡單的例子，用來實際說明動態調整頻道的必要性：

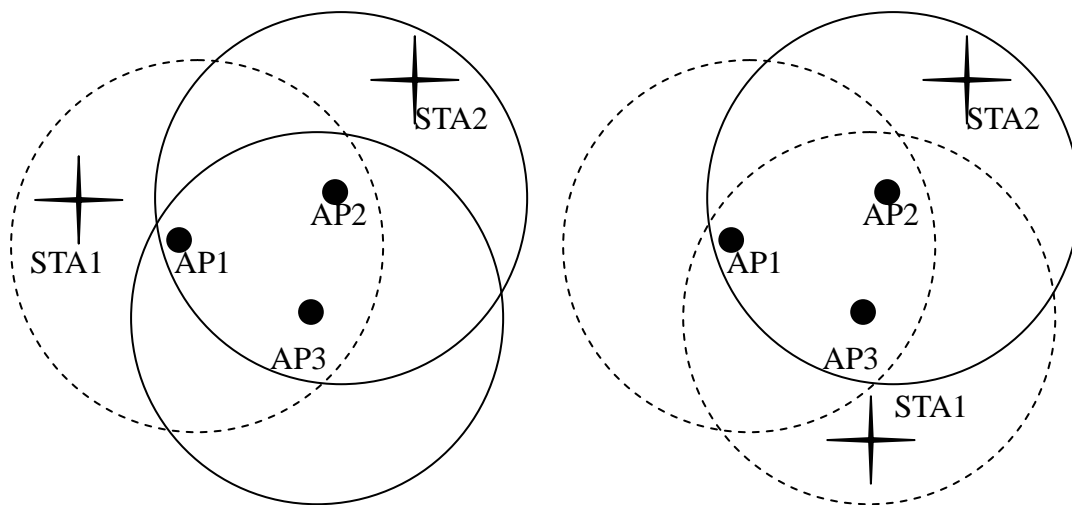


圖 3.5(a)

STA1 連接 AP1, 傳輸量 500KB/S
STA2 連接 AP2, 傳輸量 500KB/S

圖 3.5(b)

STA1 移動到 AP3, 傳輸量 500KB/S
STA2 連接 AP2, 傳輸量 500KB/S

為了舉例方便，我們假設可用的不重疊頻道只有兩個，頻道 1 和頻道 11。上圖是三個互相能聽到對方的 AP，也就是如果使用同一頻道，他們的效能只等同於一個 AP，所有 client 必須分享 6Mbps 的頻寬，是屬於情況(4)也就是干擾最嚴重的狀況。

在圖 3.5(a)的時候，AP1 上的傳輸量是 500KB/S，AP2 也是 500KB/S，因此 AP1 和 AP2 分別用頻道 1 和頻道 11 是最好的選擇一開始。但是隨著 STA1 移動到 AP3 上，AP1 的流量下降，反而 AP3 的流量上升到 500KB/S，此時 AP3 轉而使用與 AP1 一樣的頻道 11 會是好的選擇。也就是隨著 AP 上的 client 數以及流量來動態切換頻道是有其必要性，比單純靜態規劃頻道來的更能防止干擾。

3.2 系統基本原理與架構

在整個系統架構中，主要分成 Server 端與 AP 端：

(1)AP 端

- 1.負責收集與其連接的 Client 的流量資訊，將之回報給 Server 端
- 2.當接受到 Server 端的命令時，依照命令切換到適當的頻道

(2)Server 端

定時收集所有 AP 的流量資訊，依照演算法計算出目前最需要切換的 AP 及該切換到哪個頻道，並下命令給該 AP。

我們的演算法中，主要會面臨的問題就是該切換哪個 AP？以及該切換到哪個頻道？兩個問題，主要基本的想法就是希望能夠盡量將鄰近的 AP 切換到不同頻道上，並且盡量選擇 Load 小的頻道來使用，另外優先選擇目前對最多人造成干擾的 AP 來切換。在 3.3 中我們將討論怎麼定義 Load 及定義一些基本名詞，接著 3.4 將詳細描述 Server 及 AP 兩端的演算法流程。

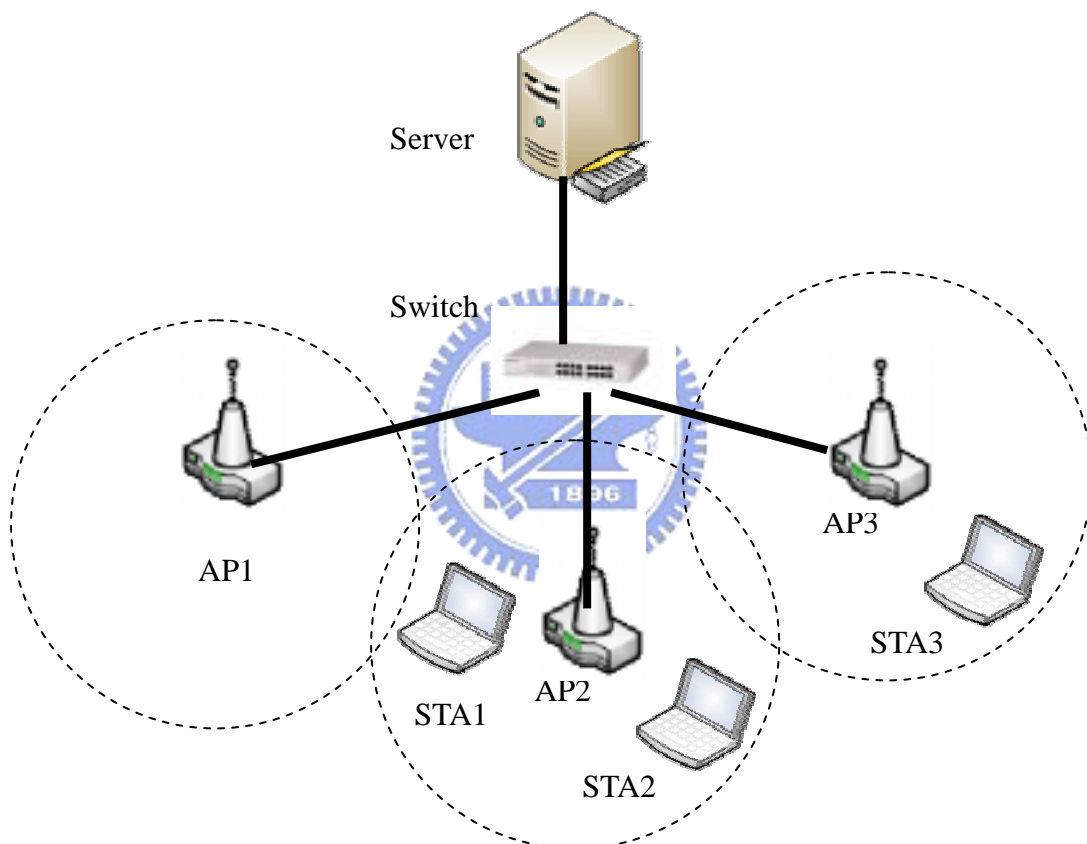


圖 3.6 系統架構示意圖

3.3 名詞定義

在這個小節中，我們把演算法中會用到的名詞一一作定義和解釋如下：

1. **Neighbor**：當 A 和 B 兩 AP 使用同一頻道，而 A 聽的到 B 的 Beacon 的時候，我們稱 B 是 A 的 Neighbor。
2. **Load**：此為我們判斷一個 AP 負載重不重的一個依據 參考[5]中對於 Load 的計算方法，我們定義一個 AP 的 Load 如下：

For every client i,
 Active number(i) = (actual bandwidth(i) / P)
 Or 1 if (actual bandwidth(i) / P) > 1
 P = Max Throughput / number of clients
 Load = sum of Active number(i)

在上面這個式子中，P 的計算方式是以 AP 最大 Throughput 除以 AP 上的 client 數，代表的是每個 client 應該分配到的平均頻寬。而 actual bandwidth(i) 則是 Client(i) 實際上所用掉的頻寬，此項數值可以經由讀取 AP 無線網路卡的數值計算出來。Active number(i) 這項數值則由 client 實際上用掉的頻寬除以分配的平均頻寬得到，代表 client(i) 對於頻寬需求的潛力，如果數值超過 1 以 1 代表。最後，我們將一個 AP 的 Load 定義成其上所有 client 的 Active number 之總和，該數值不一定是整數，通常會是浮點數。

接著我們舉兩個例子來說明，經過上述方法定義出來的 Load 比起其他簡單的計算方式更能代表一個 AP 上的流量負載(以下兩個例子為了舉例方便假設 AP 的最大 Throughput 是 600KB/s)。

(1) Active number 和 Throughput 兩個計算方法的比較

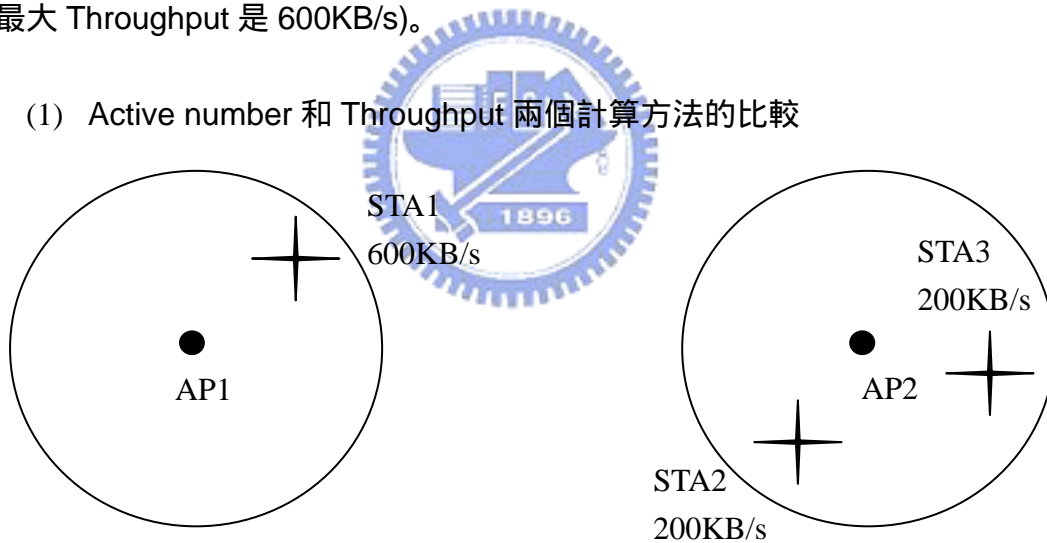


圖 3.7 AP1 上只有一個 Client, 流量為 600KB/S, AP2 上則有兩個 Client, STA2 的流量是 200KB/S, STA3 的流量也是 200KB/S

AP1 : 600k
 P=600k/1=600k
 Load=600/600=1

AP2 : 200k+200k=400k
 P=600k/2=300k
 Load= 200/300+200/300=4/3

上面這個例子中,如果單純以 AP 上面的總流量(Throughput)當 Load 來看,因為 AP2 上兩個 Client 的流量加總還比 AP1 的小,所以此例中,會以為 AP2 所在的頻道負載比較小,實際上假設將第三 AP 設定在跟 AP2 同一頻道上,因為多人競爭的關係,平均能得到的頻寬反而會不如 AP1,在 AP2 所在頻道上只能分到 200KB/S,而選擇 AP1 所在的頻道反而能得到 300KB/S,在此例中,我們的計算方法也指出 AP2 的 Load 較大,所以應該選擇 AP1 的頻道。

(2) Active number 和 Client Number 兩個計算方法的比較

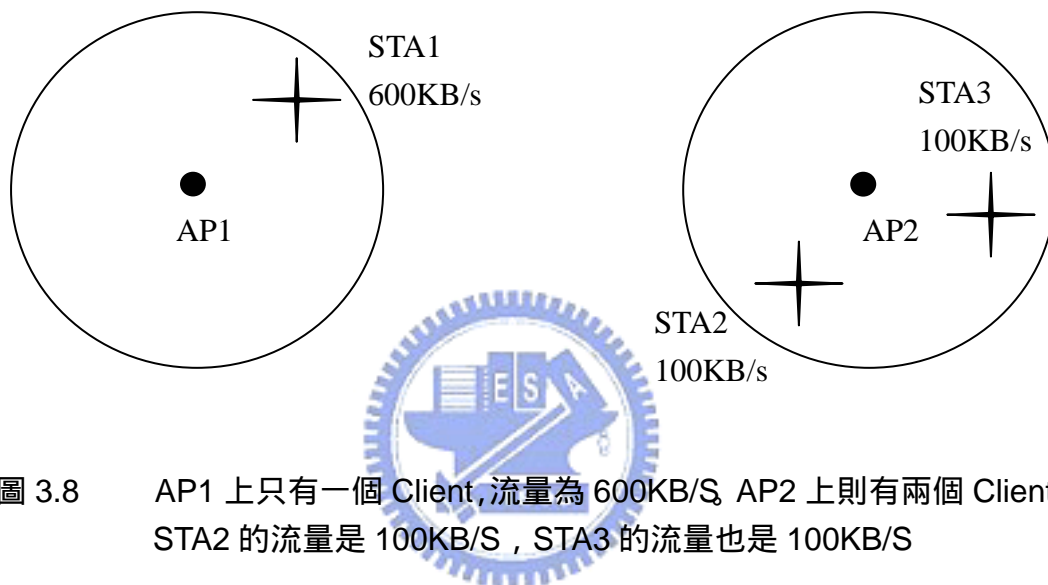


圖 3.8 AP1 上只有一個 Client,流量為 600KB/S, AP2 上則有兩個 Client, STA2 的流量是 100KB/S, STA3 的流量也是 100KB/S

AP1 : 600k	AP2 : 100k+100k=200k
Client number: 1	Client number: 2
$P=600k/1=600k$	$P=600k/2=300k$
$Load=600/600=1$	$Load= 100/300+100/300=2/3$

經過上面的例子後,這次我們看單純以 AP 上面的 Client 數當 Load 能否正確反映出頻道的擁擠性。這次因為 AP2 上有兩個 Client,所以明顯此例中,會以為 AP1 所在的頻道負載比較小,但實際上假設將第三 AP 設定在跟 AP1 同一頻道上,因為流量太大的關係,平均能得到的頻寬反而會不如 AP2,在 AP1 所在頻道上只能分到 300KB/S,而選擇 AP2 所在的頻道反而能得到 400KB/S,在此例中,我們的計算方法亦正確的指出 AP1 的 Load 較大,所以應該選擇 AP2 所在的頻道。

3. Neighbor's Load : 將鄰近 AP(Neighbor)的 Load 加總,即為某 AP 的 NL(Neighbor's Load),我們利用 NL 來判斷一個頻道的擁擠程度,也就是

當鄰近越多 Load 大的 AP 使用此一頻道，該頻道的 NL 變會越大，我們應避免將 AP 切換到這種頻道上。

NL1:將目前位於 channel 1 的 Neighbor 的 Load 加總起來得到的值

NL6:將目前位於 channel 6 的 Neighbor 的 Load 加總起來得到的值

NL11:將目前位於 channel 11 的 Neighbor 的 Load 加總起來得到的值

NL \odot =將目前跟本 AP 所處頻道相同之 Neighbor 的 Load 加總起來

- 4. Neighbor's Neighbor** : 如果 A、B、C 三 AP 位於同一頻道時，A 能夠聽到 B 的 Beacon，B 能夠聽到 C 的 Beacon，而 A 聽不到 C 的 Beacon，則我們稱 C 是 A 的 NN(Neighbor's Neighbor)。

3.4 演算法流程

3.4.1 資料結構

這個小節中我們說明在 AP 和 Server 之間所傳遞訊息的資料結構。

- (1) Server傳給AP的控制訊息一共有兩種：

- 1.Get_Load_Cmd : 代表要求AP回報其上的流量訊息給Server
- 2.Change_Channel_Cmd : 代表要求AP切換到某一頻道

- (2) AP傳給Server的資料訊息一共包含以下幾個資料項目：

- 1.AP_MAC_Addr : 此AP的MAC address，用以辨別AP的身分
- 2.Nb_MAC_Addr : AP所有Neighbor的地址，方便Server建立網路拓樸。
- 3.Load : 依照前面所提的Active Number加總方式算出來的Load
- 4.Neighbor number : AP 所擁有的 Neighbor 個數
- 5.Channel : AP 所在的頻道

3.4.2 AP 端流程

AP 端在我們的演算法中是屬於比較被動的一方，隨著 Server 端傳來的控制訊息而動作。如果收到 Get_Load_Cmd 就執行 Transfer Message Back，如果收到 Change_Channel_Cmd 就執行 Change Channel。

- (1) Transfer Message Back : 這個步驟按照前面說的辦法計算 AP 的 Load，並且回傳 AP 的相關資訊，放在資料訊息中給 Server。

(2) Change Channel：將此 AP 的頻道調整到 Server 指定的頻道。

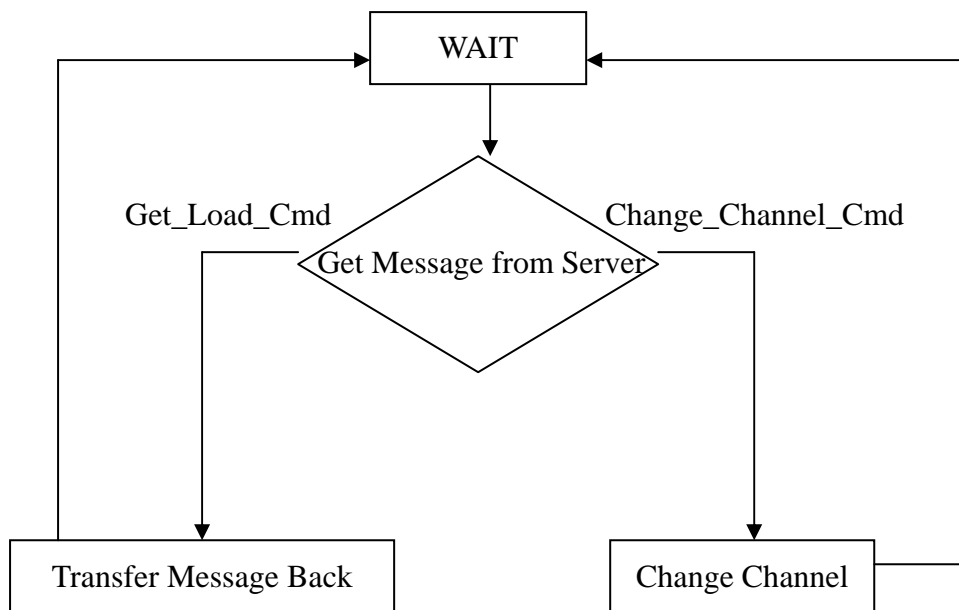


圖 3.9 AP 端演算法流程圖

3.4.3 Server 端流程策略一：動態頻道切換演算法

```
WHILE()  
{  
    sleep();  
    Get the Load of every AP  
    Function();  
}
```

sleep()：等待一段時間，主要是AP剛切換頻道後，其上的client不見得會隨之而切換，在頻道切換過程中各鄰近AP也會造成干擾，所以這段等待的時間是要讓因為切換AP頻道造成的影響穩定後，才再次收集AP上的流量資訊，否則若收集的流量資訊不穩定，通常會造成頻道不斷作錯誤的切換。

Get the Load of every AP：向每個AP發出Get_Load_Cmd()，並收集回傳的資訊建立網路基本拓樸，及各AP的資訊表。

Function()：主要演算法所在副程式，負責決定哪個AP需要優先切換頻道以及切換到哪一個頻道，並且向該AP發出Change_Channel_Cmd指示切換頻道。


```

Function()
{
  A : APs which has not been picked
  A<- all Aps
  Calculate NL © (AP) of every AP
  While (A!=empty)
  {
    J = MAX{NL© (AP) in A }
    if NL© (J) is not the minimum among NL1(J), NL6(J), and NL11(J)
    {
      Change J to channel n ( NLn(J) is the minimum among three channel)
      Calculate NL © (AP) of every AP
    }
    A=A-J
  }
}

```

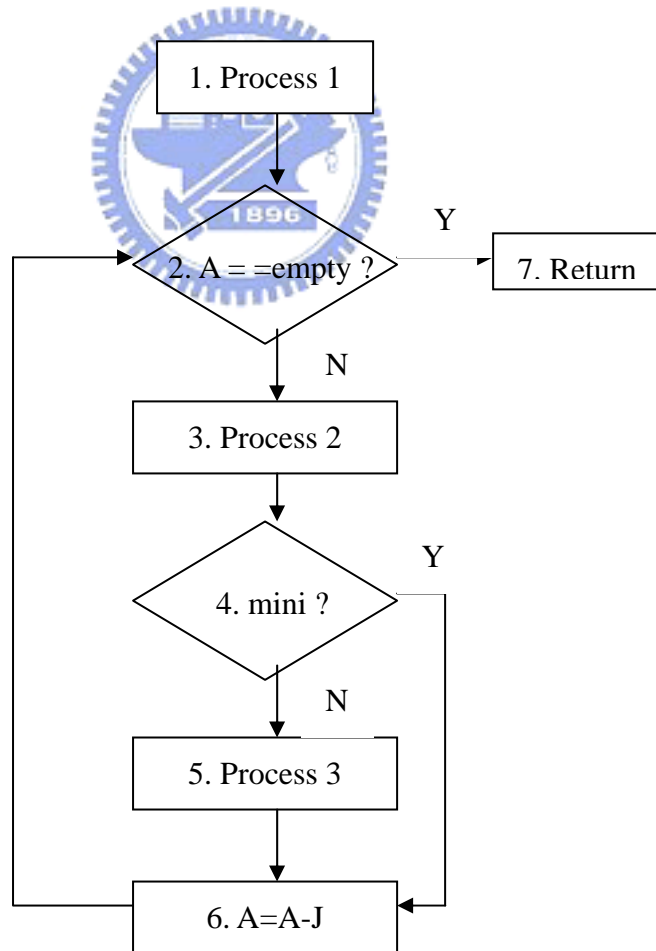


圖 3.10 Server 端演算法流程圖

(1) Process 1：集合 A 是由所有還沒被挑選過的 AP 所形成，一開始所有的 AP 都在此集合中，Server 並且逐一計算每個 AP 的 NL \odot 。

(2)此步驟主要辨識集合 A 是否為空集合？

(3) Process 2：在集合 A 中挑選 NL \odot 值最大的 AP，稱其為 J。NL \odot 值最大代表此 AP 目前所在的頻道上鄰近有最多 Load，也就是被其所干擾的人會很多，我們優先選擇這樣的 AP，嘗試將其切換到較為寬鬆的頻道上，希望可以先解決最擁擠的地方。

(4) 此步驟主要是比較 AP J 的 NL \odot 值是否為 NL1、NL6、NL11 中最小的一個，也就是企圖分辨此 AP 所在的頻道是不是已經是最不擁擠的一個。

(5) Process 3：將 AP J 切換到 NL 值最小的頻道上去，也就是將 J 切到最寬鬆的頻道上。並且重新計算所有 AP 的 NL \odot 。

(6) 此步驟主要是將 J 從集合 A 中剔除，代表已經考慮過它。

(7)結束 Function()副程式，Return。

3.4.4 Server 端流程策略二：考量 Neighbor's Neighbor

前面的演算法中，我們計算 NL 這個數據的時候，只考慮了 Neighbor 的 Load，也就是假定只有 Neighbor 的傳輸會對 AP 造成影響，實際上，從 3.1 節中討論的 AP 距離和干擾的關係所列出的四個情況可得知，不僅僅只是 Neighbor 會對 AP 造成影響，兩 AP 之間的距離在 $R \sim 2R$ (傳輸距離)之間時，即使聽不到對方 AP，Client 跟 AP 在進行傳輸時候也有可能對對方 AP 造成嚴重的干擾，如圖 3.3 所示。

因此本節將進一步將這種干擾的考量納入演算法中，也就是考慮 Neighbor's Neighbor(NN)的 Load 對 AP 所造成的影響。基本上新的演算法類似原演算法，需要修改的地方，大致集中在 Server 端作改變，更動如下：

- 1.在Server端針對每個AP都必須新增一個叫做probability table的資料結構，此結構中針對AP的每一個NN都賦予一個機率值P， $0 \leq P \leq 1$ ，初始值為0。由於NN並不是一定會對AP造成干擾，此機率值主要是用來猜測這個NN對AP會造成多大的影響。

2. Server端演算法更動 Neighbor's Load的定義：

$$NL1^* = NL1 + \text{sum of (Load of NNi in channel 1 * P(NNi))}$$

$$NL6^* = NL6 + \text{sum of (Load of NNi in channel 6 * P(NNi))}$$

$$NL11^* = NL11 + \text{sum of (Load of NNi in channel 11 * P(NNi))}$$

也就是新的 NL 是舊的 NL 加上一個值所形成，這個值是把每個 NN 的 Load 乘上其機率值 P 再加總。

3. AP 端傳送到 Server 的資料訊息必須新增一個 Throughput 的欄位，回報 Server 此刻 AP 上的總流量。

4. 當我們將 AP 從頻道 A 切換到頻道 B 後，發現 AP 所回傳的 Throughput 反而降低而非提高的時候，代表我們可能低估了頻道 B 中的 NN 對 AP 所造成的影響，或者高估了頻道 A 中的 NN 造成的干擾，因此我們這時候將所有位於頻道 B 之 NN 的機率值 P 設定為原本的 P 加上 0.1，所有位於頻道 A 之 NN 的機率值 P 設定為原本的 P 減掉 0.1，藉以猜測 NN 對此 AP 所造成的影響。



第四章 實驗結果

在實驗的部分，我們主要設計了三個實驗，來測試我們提出來的策略—動態頻道切換演算法的效果，並且將之與另外三種簡單的演算法作比較，分別為

- (1) AP 不切換頻道
- (2) 根據 AP 的 Throughput 切換，選擇總流量較低者
- (3) 根據 AP 上的 client 數切換，選擇較少 client 使用的頻道。

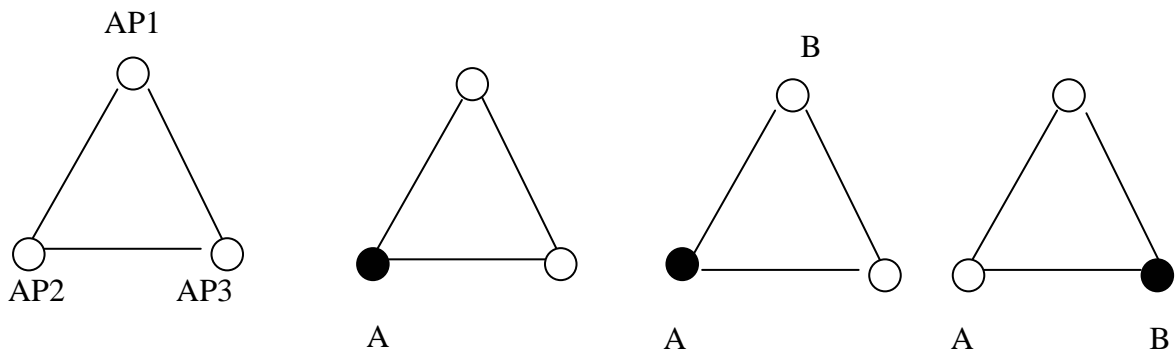
也就是我們實作了不同的演算法，並且觀察在同一個實驗中使用不同演算法所帶來的結果，比較他們的優缺點及能提高多少網路 Throughput。為了簡化實驗時所需要的 AP 個數，我們三個實驗都假設可用的不重疊頻道只有兩個，分別為頻道 1 及頻道 11。

實驗中利用 Intersil Prism2 晶片的無線網卡及 notebook 和有線網路，架設了三個 Host AP，此三個 AP 擺設的位置約相距 2 公尺，彼此都能聽到對方的 BEACON，其餘四台 notebook 則使用普通無線網卡，扮演到處移動的 client。實驗示意圖中的圈圈，代表 AP，白色代表使用頻道 1，黑色代表使用頻道 11，兩圈圈若有直線連接，代表此兩 AP 是 Neighbor，ABCD 則代表到處遊走的 Client。

4.1 實驗一

此實驗主要用來比較動態頻道切換以及不切換頻道兩種演算法的優劣。

(1) 使用動態頻道切換演算法



此實驗中，Client A 和 Client B 分別開一個 TCP 連結，A 固定不動連接 AP2，隨著 B 的走動，AP 會動態轉換頻道，使得 A 和 B 始終保持使用不同的頻道，因此即使三個 AP 的訊號範圍彼此覆蓋，A 和 B 仍能夠避免競爭同一頻道。

下面是各client的平均傳輸速率：

Client A 平均傳輸速率：617 KB/S

Client B 平均傳輸速率：632 KB/S

網路流量總和：1249 KB/S

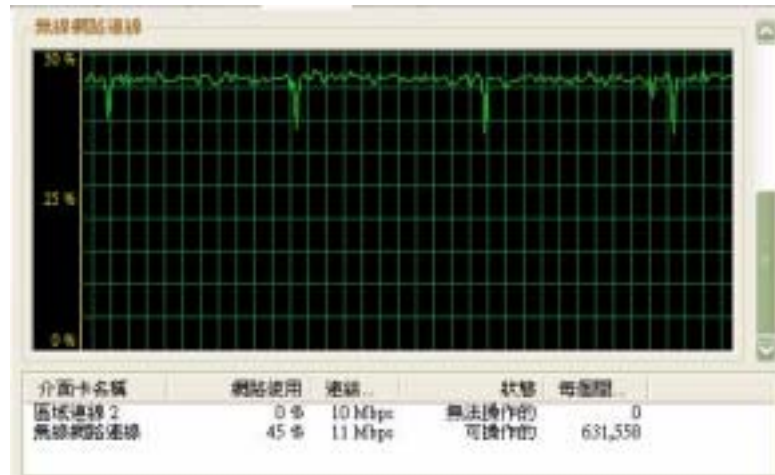


圖 4.2 實驗一 動態頻道切換 Client A 實測流量圖 (1)

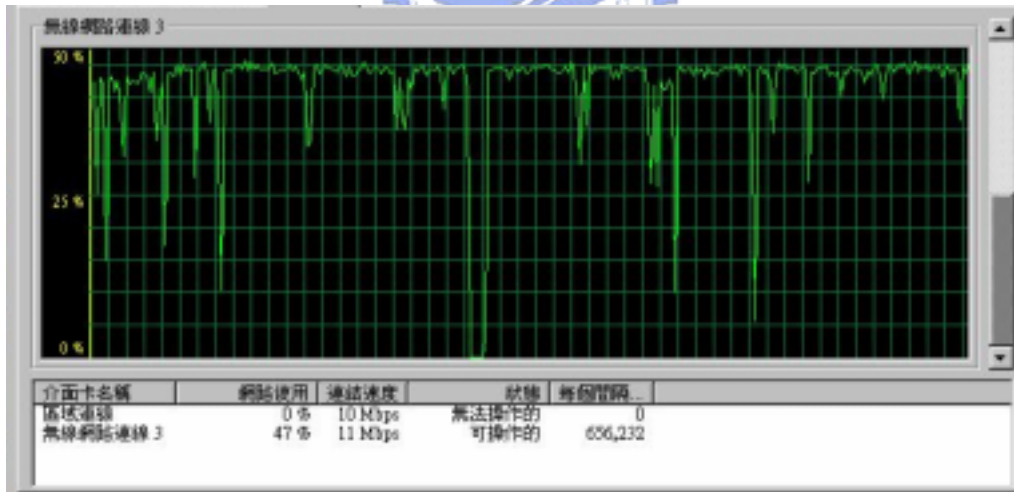


圖 4.3 實驗一 動態頻道切換 Client B 實測流量圖 (1)

從上面實際上測試的數據圖，可以看出流量有時候會出現不甚穩定的情況，因為實驗時有些時候會有其他 AP 的訊號干擾，造成訊號的不穩定。但是平均來說 Client A 及 Client B 都維持有將近 6Mbps 的流量，也就是因為動態頻道切換使得他們始終能維持在不同的頻道上，充分利用各自 AP 的流量，即使他們所在 AP 的位置相當接近。

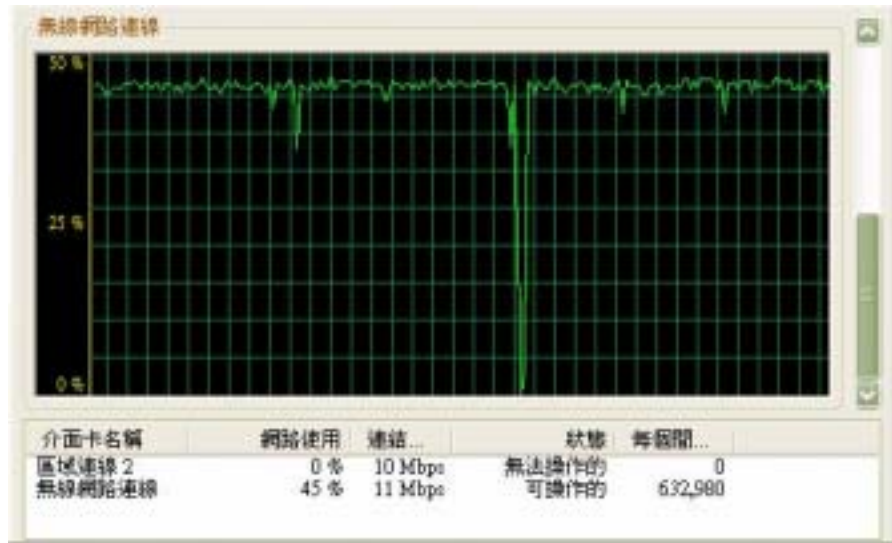


圖 4.4 實驗一 動態頻道切換 Client A 實測流量圖 (2)

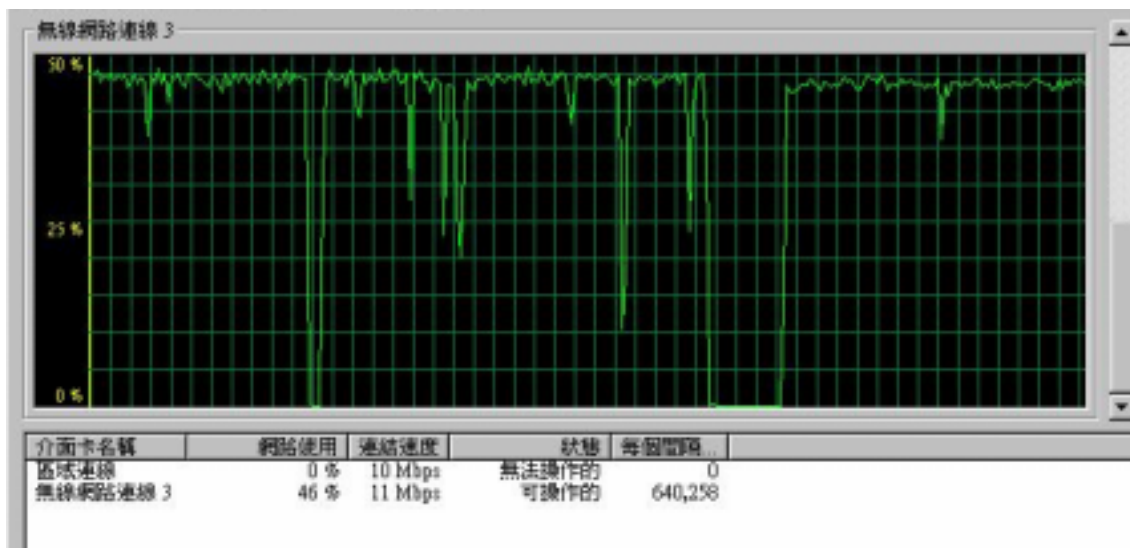


圖 4.5 實驗一 動態頻道切換 Client B 實測流量圖(2)

另外，AP 轉換頻道的時候，Client 需要一段時間才能跟著轉換頻道，如圖 4.4 便有 Client A 轉換頻道時候流量突然大量下降的痕跡，圖 4.5 一小段流量為 0 的時間則是 Client B 從 AP1 轉到 AP3 的時候所造成的。

(2)使用 AP 不切換頻道演算法

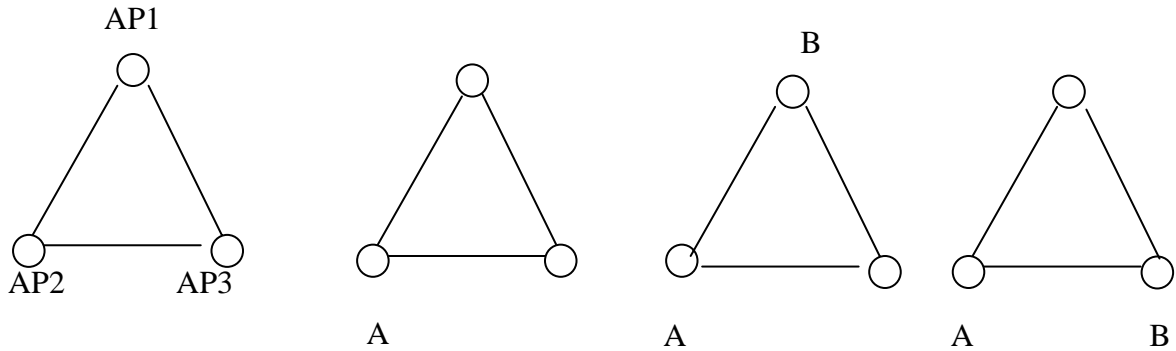


圖 4.6 實驗一 不切換頻道 示意圖

此方法中，和前一個方法相同，Client A和Client B分別開一個TCP 連結，各自連接一個AP，A固定不動，而B則一開始使用AP1接著轉到AP3上面去，不同的是這次所有AP都使用同一個頻道，從實驗結果可以看到Client A和Client B所能得到的流量總合平均只有660.45KB/S，遠不如使用動態頻道切換演算法，因為此三個AP傳輸範圍互相覆蓋，不切換頻道代表這三個AP上的Client都必須分享總共相當於一個AP的頻寬，使得效能大降。下面是各client的平均傳輸速率：

Client A 平均傳輸速率：318.18 KB/S

Client B 平均傳輸速率：342.27 KB/S

網路流量總和：660.45 KB/S

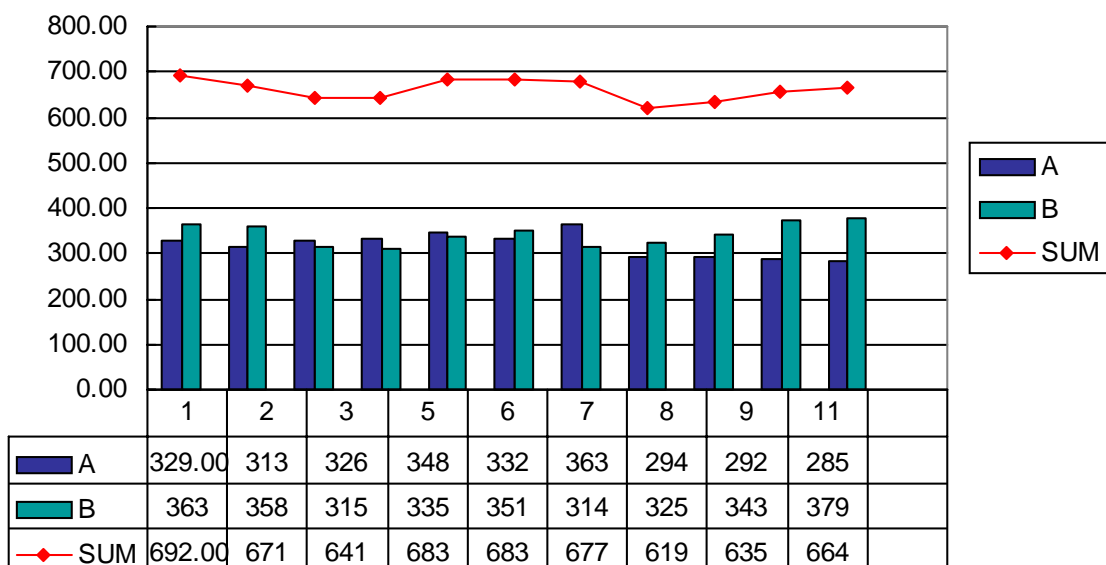


Table 4.1 實驗一 不切換頻道 流量統計表

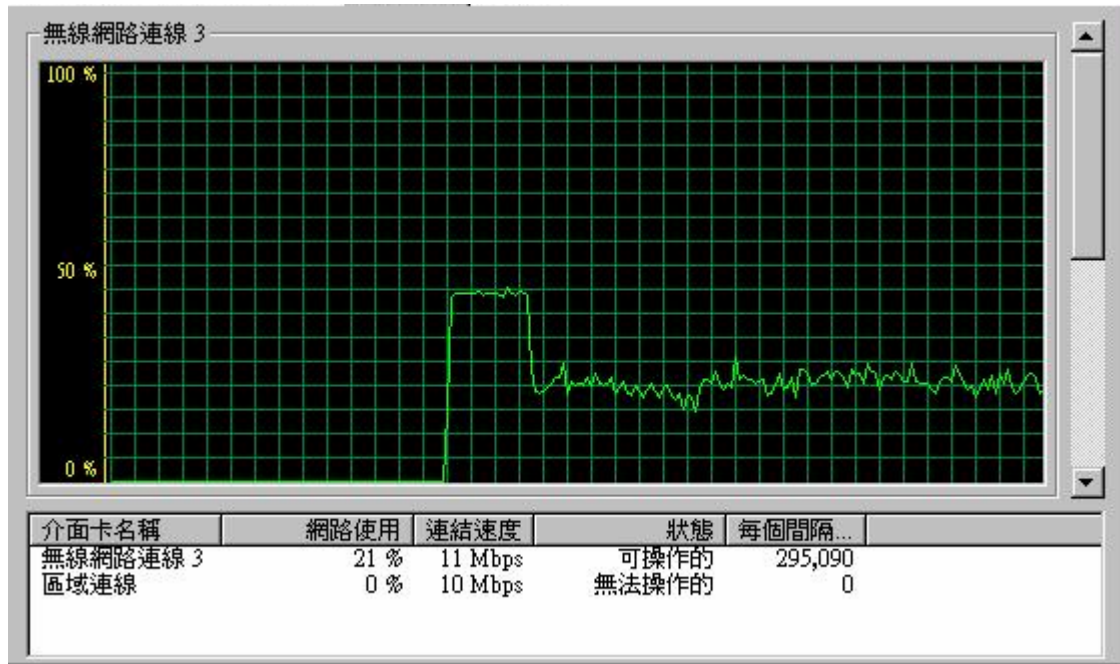


圖 4.7 實驗一 不切換頻道 Client A 實測流量圖

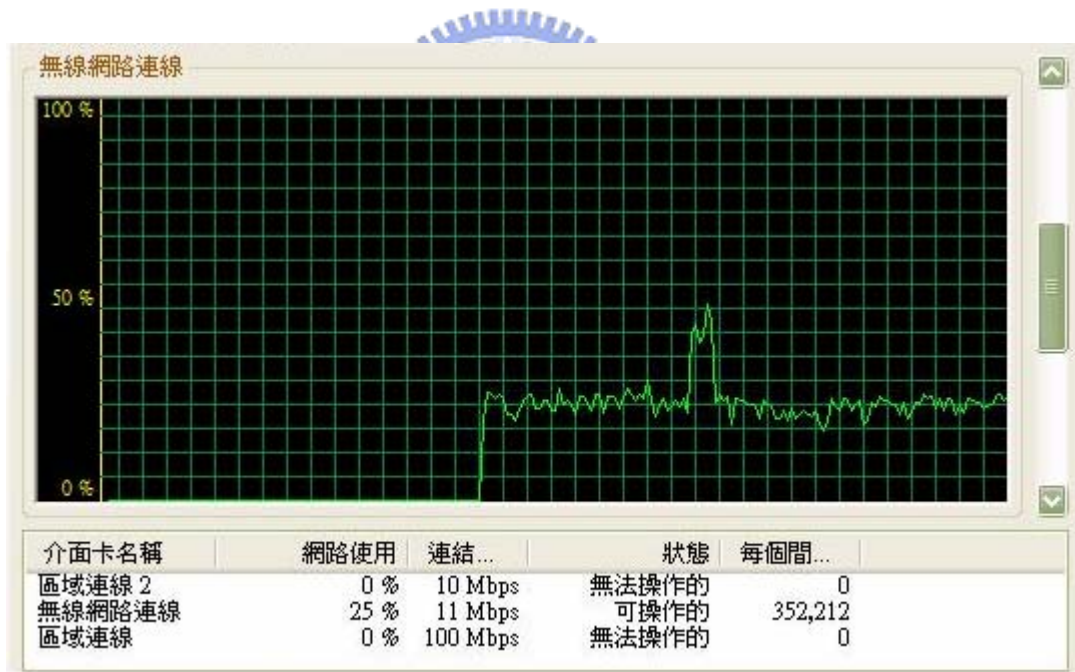


圖 4.8 實驗一 不切換頻道 Client B 實測流量圖

從實測圖上可以看到，原本 Client A 的流量有六百多 KB/S，因為三個 AP 都使用同一頻道，所以隨著 Client B 也開始連接 AP，Client A 的流量即下降到三百多 KB/S，此後兩個 Client 都維持在三百多 KB/S，流量總合約只有動態切換演算法的一半。

4.2 實驗二

此實驗比較動態頻道切換、不切換頻道及根據 Client 數切換三種演算法的優劣。

(1) 使用動態頻道切換演算法

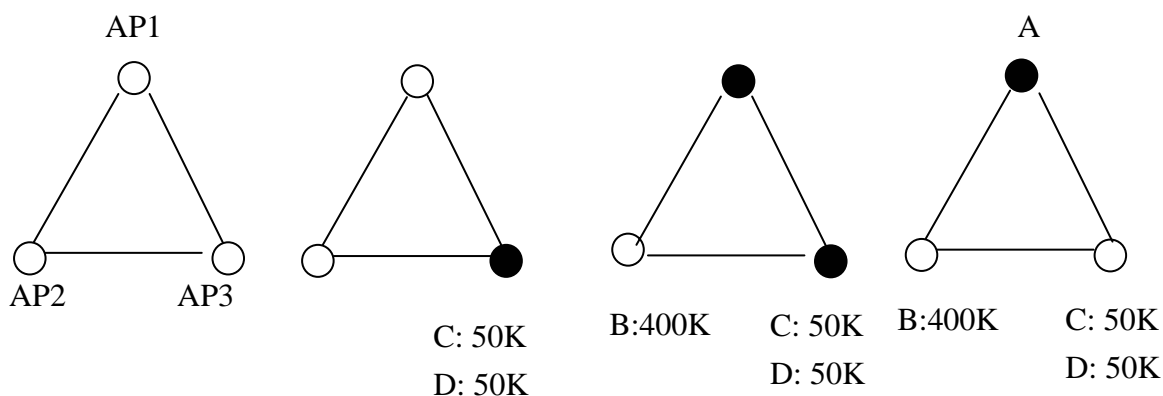


圖 4.9 實驗二 動態頻道切換 示意圖

此實驗中,C和D分別灌入50KB/S的流量在AP3上,而B則在AP2灌入400KB/S的流量,隨著A進入AP1且流量越來越大,AP2和AP3會轉而使用同一頻道。下面是各client的平均傳輸速率:

- A 平均傳輸速率 : 575 KB/S
- B 平均傳輸速率 : 400 KB/S
- C 平均傳輸速率 : 50 KB/S
- D 平均傳輸速率 : 50 KB/S
- 網路流量總和 : 1075 KB/S

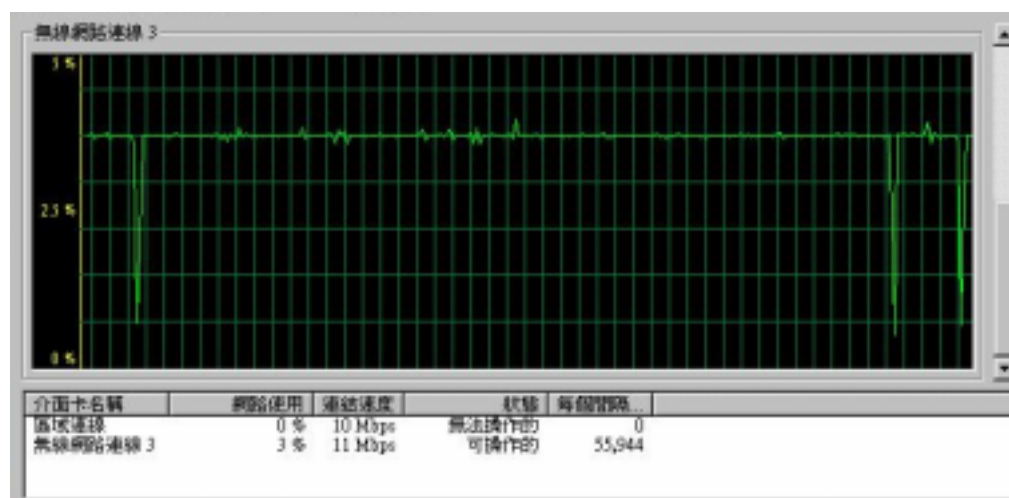


圖 4.10 實驗二 動態頻道切換 Client C 實測流量圖

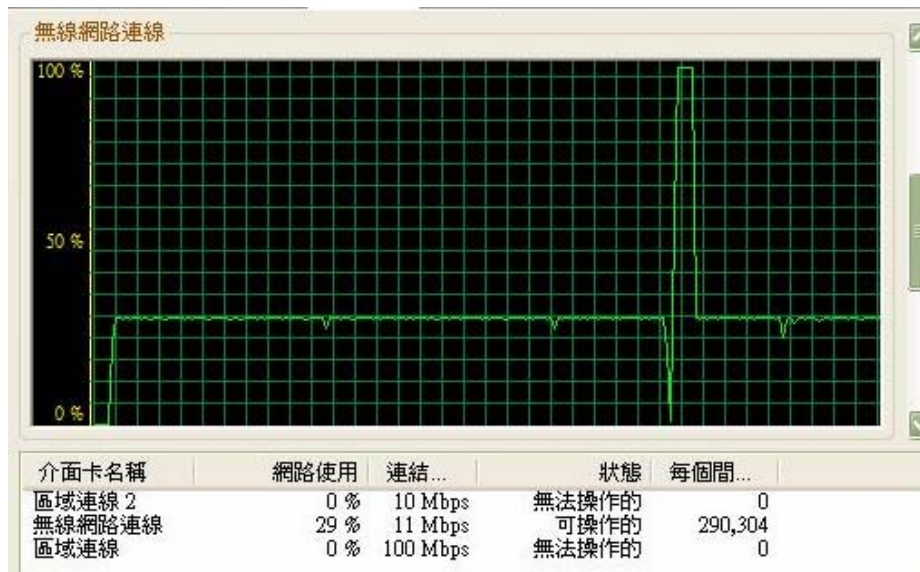


圖 4.11 實驗二 動態頻道切換 Client B 實測流量圖

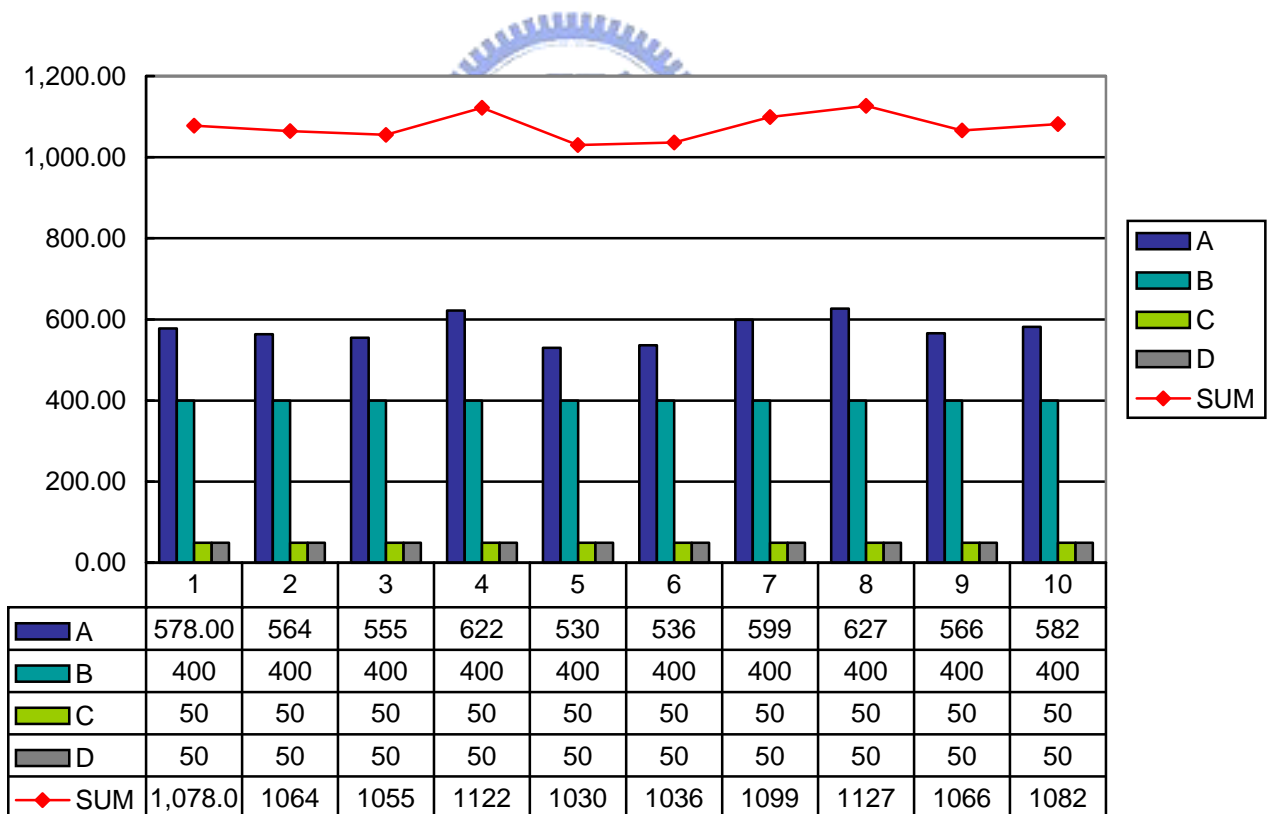


Table 4.2 實驗二 動態頻道切換 流量統計表

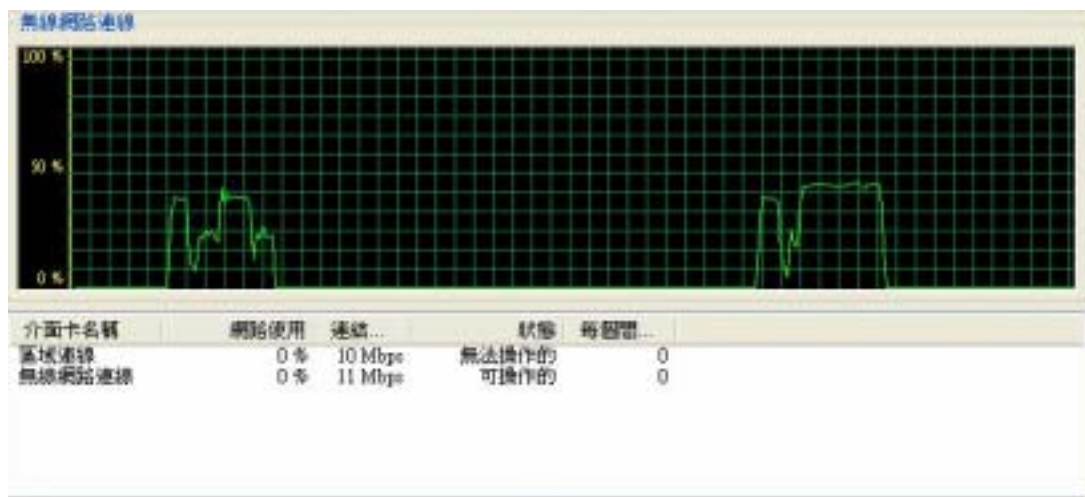


圖 4.12 實驗二 動態頻道切換 Client A 實測流量圖(斷線)

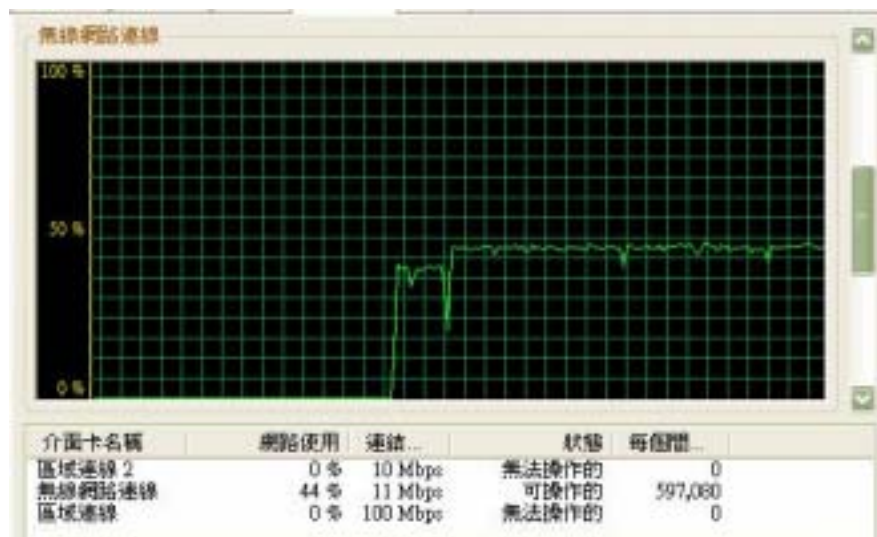


圖 4.13 實驗二 動態頻道切換 Client A 實測流量圖(成功)

從上面的圖表可以看到，Client C 及 Client B 分別穩定維持在 50KB/S 和 400KB/S，而圖 4.12 則顯示了 Client A 不斷的连接上 AP 又斷線，這是因為一開始程式所設定的檢查 AP 流量時間間隔太短，而對流量波動過於敏感，造成頻道不斷錯誤切換而讓其上的 client 斷訊，加長檢查流量時間間隔後，便如圖 4.13 能夠正常運作。這個現象除了讓我們知道演算法中 Sleep() 的時間間隔不宜太短外，也同時顯示了由 AP 端主動切換頻道的缺點，因為缺乏通知 Client 跟著切換頻道的機制，所以不能保證所有 Client 都能跟上 AP 切換頻道的腳步。從實際上的實驗得知，以現在 Client 端的機制，在沒有明顯位置移動下，通常會在 AP 切換頻道後的幾秒後重新搜尋到原 AP 現在所使用的頻道而重新連接上。

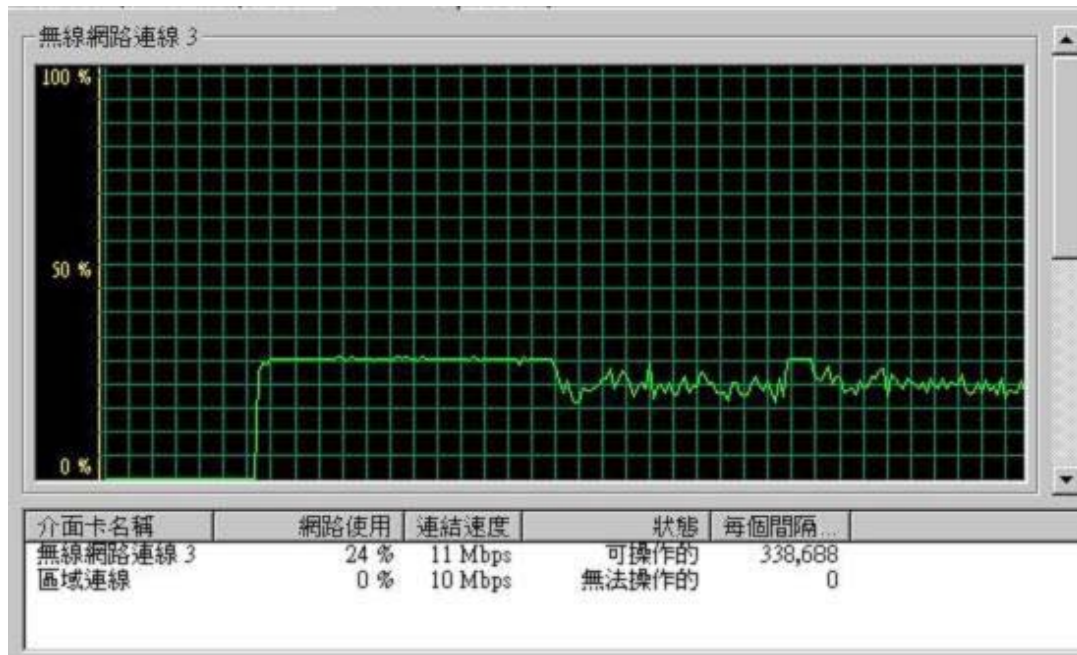


圖 4.15 實驗二 不切換頻道 Client B 實測流量圖

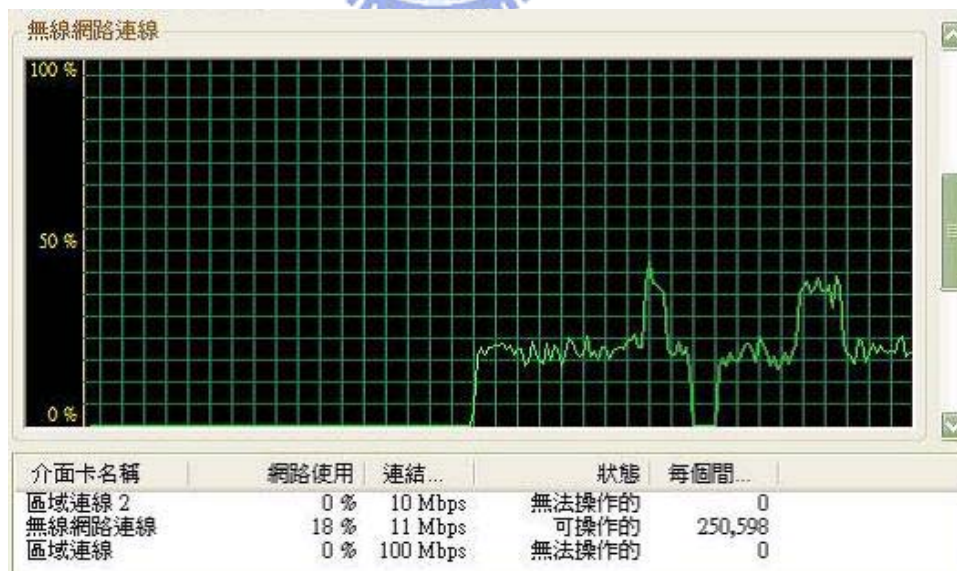


圖 4.16 實驗二 不切換頻道 Client A 實測流量圖

(3) 使用根據 AP 上的 client 數切換演算法

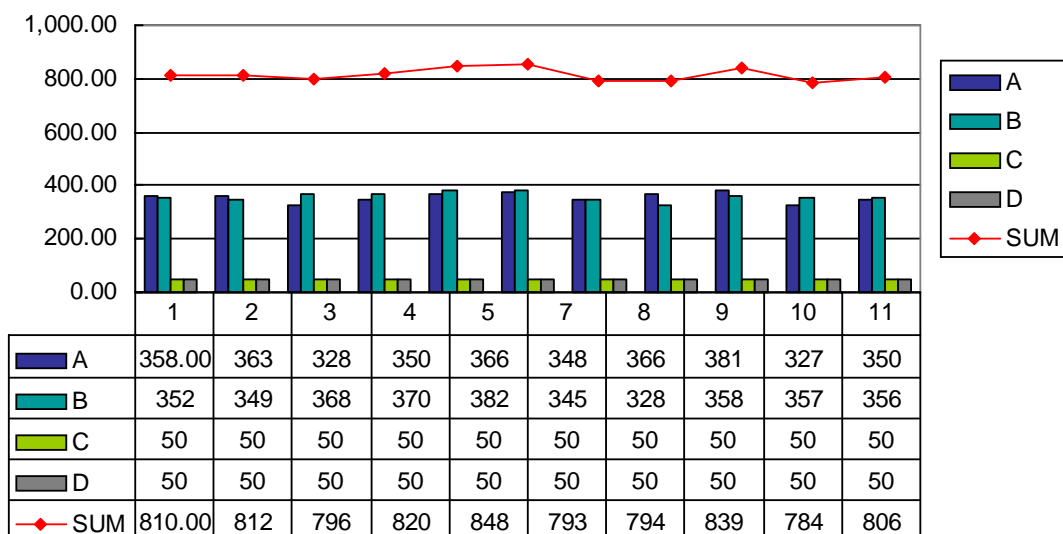


Table4.4 實驗二 根據 AP 上的 client 數切換 流量統計表

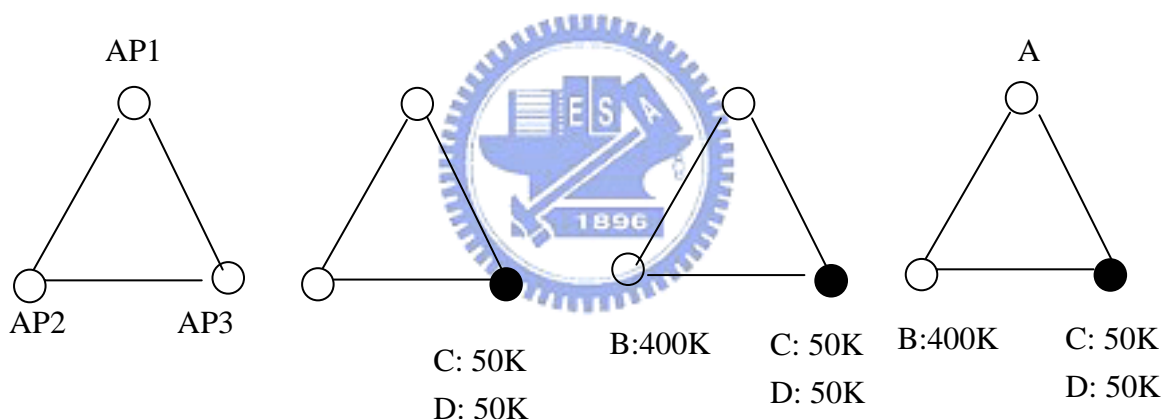


圖 4.17 實驗二 根據 AP 上的 client 數切換 示意圖

和前一個方法的環境相同，C和D分別灌入50KB/S的流量在AP3上，而B則在AP2灌入400KB/S的流量，但是這個方法會選擇Client數較少的頻道，因此AP1和AP2使用同一頻道，而AP3使用另一頻道，結果網路流量總合約只有814.27 KB/S，只比不切換頻道好一些，因此可見單是以Client數來判斷頻道的擁擠程度是不足夠的。下面是各client的平均傳輸速率：

A 平均傳輸速率：355.72 KB/S

B 平均傳輸速率：358.34 KB/S

C 平均傳輸速率：50 KB/S

D 平均傳輸速率：50 KB/S

網路流量總和：814.27 KB/S

4.3 實驗三

此實驗比較動態頻道切換 不切換頻道及根據 AP 的 Throughput 切換三種演算法的優劣。

(1)使用動態頻道切換演算法

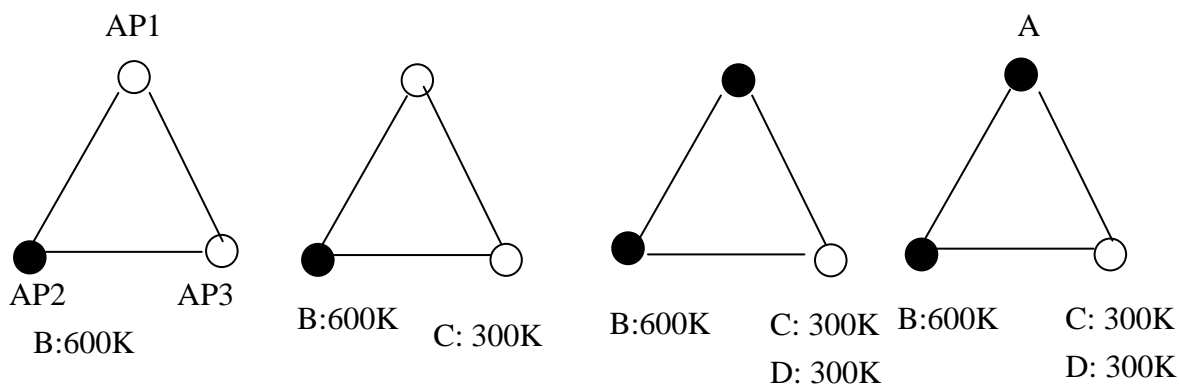


圖 4.18 實驗三 動態頻道切換 示意圖

此實驗中，分別在AP2灌入600KB/S的流量，及在AP3中兩Client各300KB/S的流量，AP1會自動切換頻道以得到最大頻寬。從實際上測試的流量統計表中，可以見到最後AP1會切換到與AP2同一頻道，兩者共享頻寬，網路流量總和大概平均維持在1330.58 KB/S。

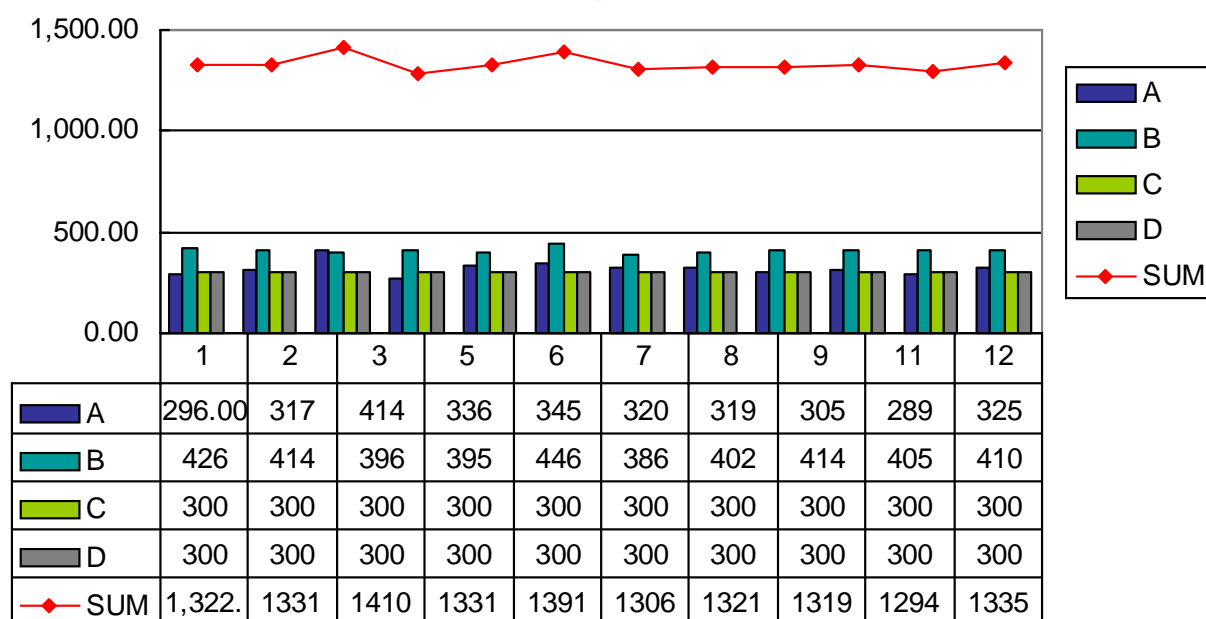


Table 4.5 實驗三 動態頻道切換 流量統計表

下面是各client的平均傳輸速率：

- A 平均傳輸速率：320.6 KB/S
 - B 平均傳輸速率：410 KB/S
 - C 平均傳輸速率：300 KB/S
 - D 平均傳輸速率：300 KB/S
- 網路流量總和：1330.58 KB/S

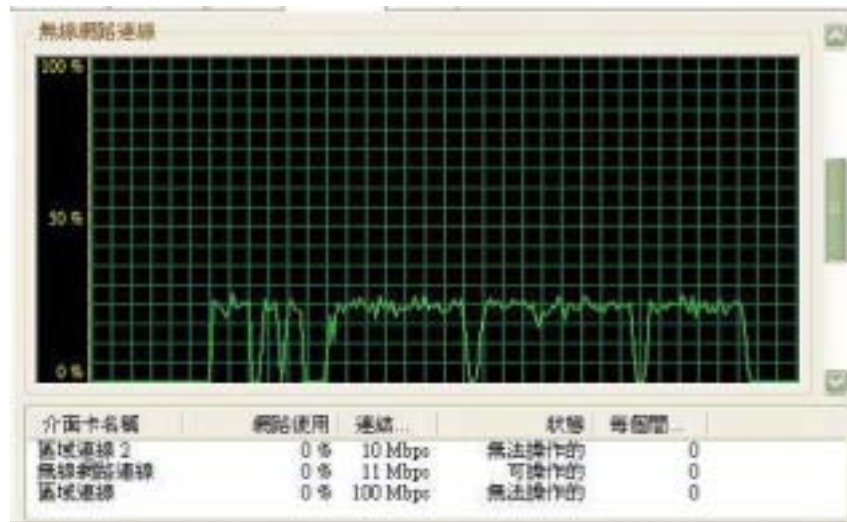


圖 4.19 實驗三 動態頻道切換 Client A 實測流量圖

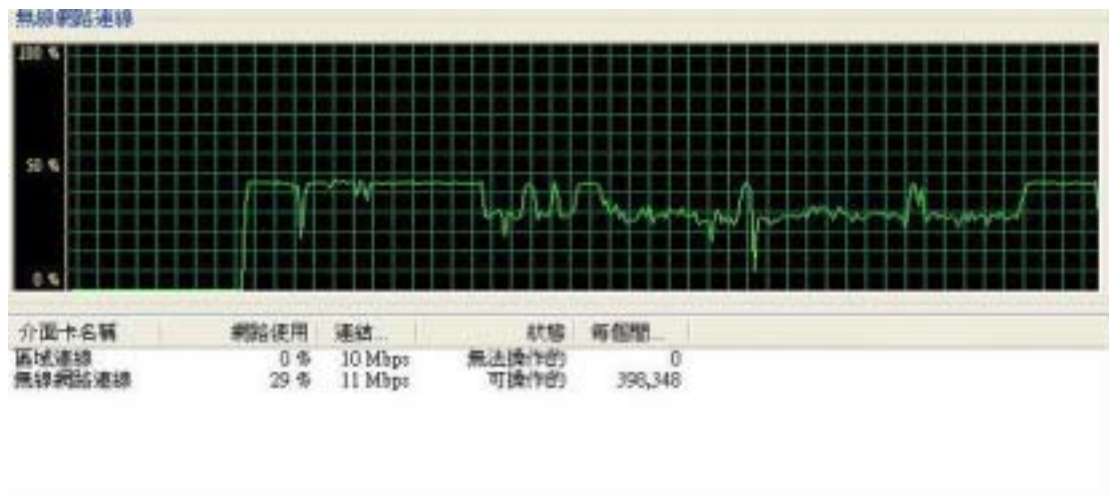


圖 4.20 實驗三 動態頻道切換 Client B 實測流量圖

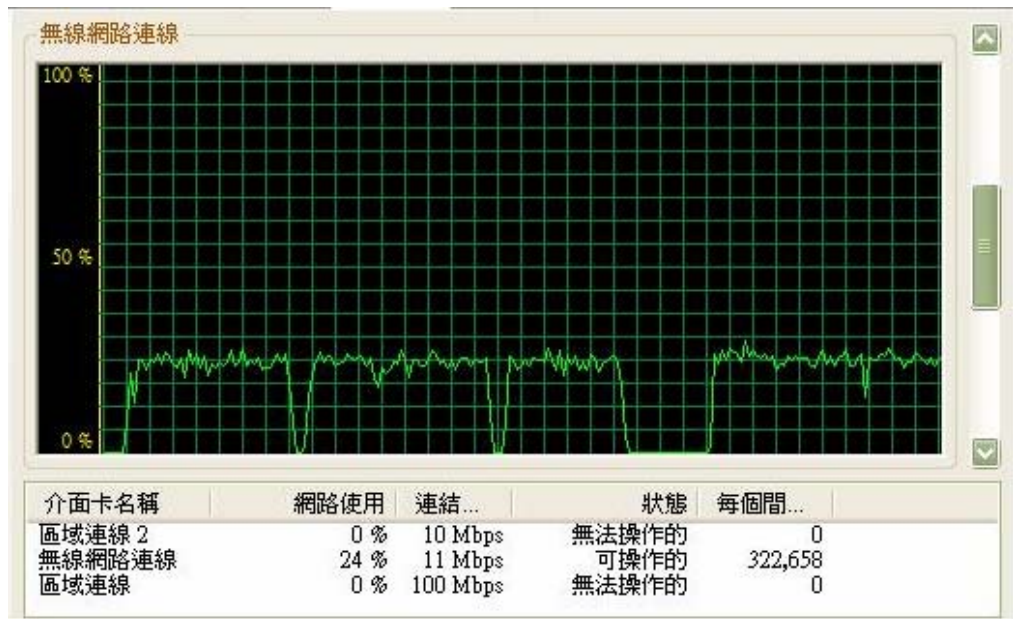


圖 4.21 實驗三 動態頻道切換 Client A 實測流量圖

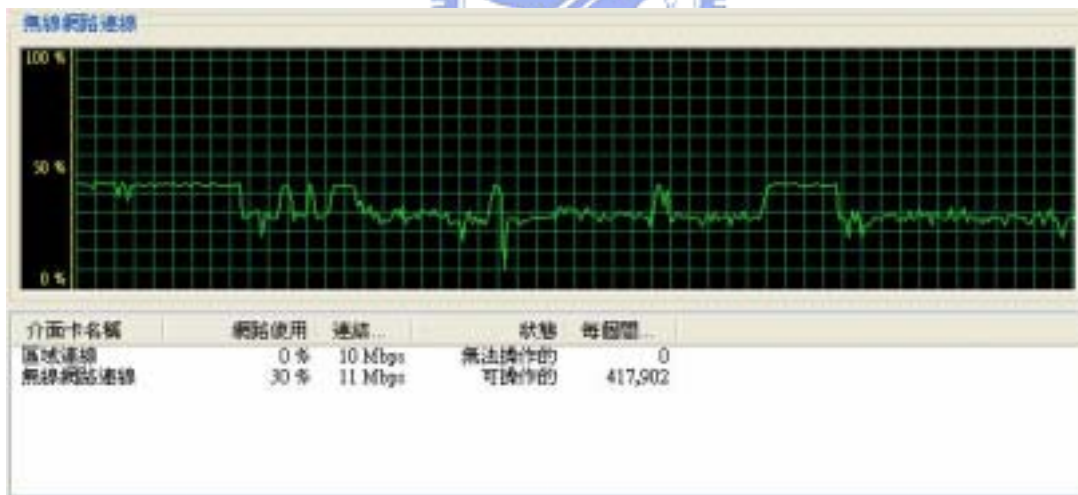


圖 4.22 實驗三 動態頻道切換 Client B 實測流量圖

(2) 使用根據 AP 的 Throughput 切換演算法

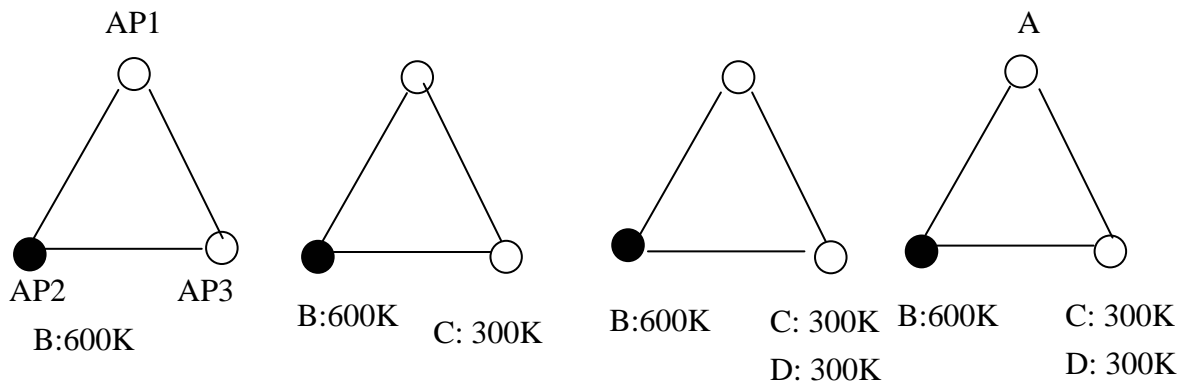


圖 4.23 實驗三 根據 AP 的 Throughput 切換頻道 示意圖

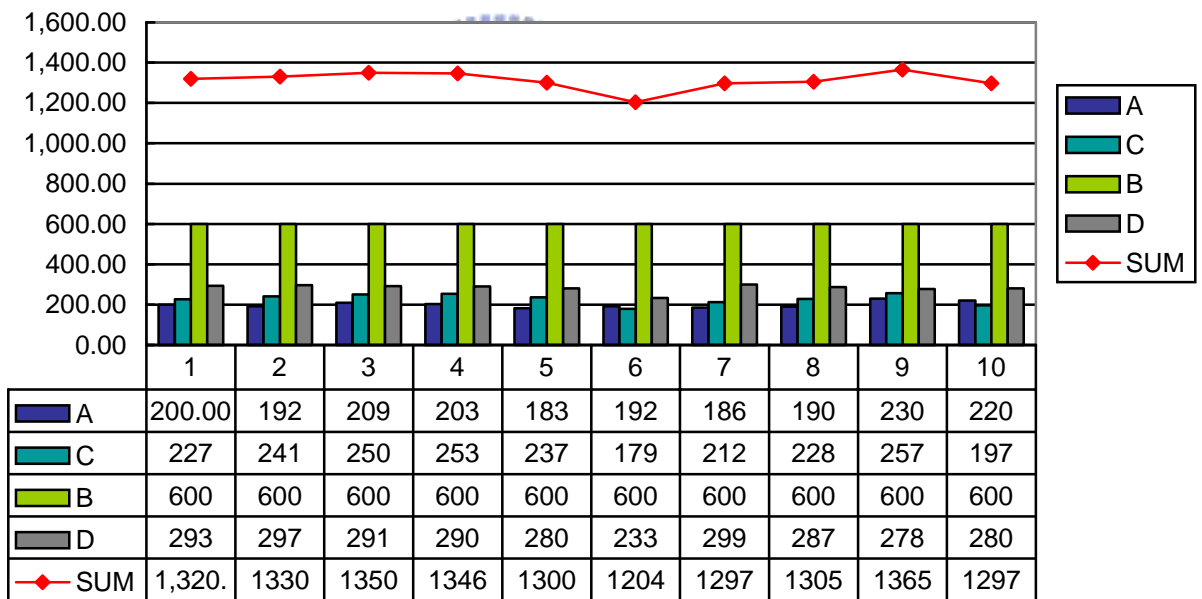


Table 4.6 實驗三根據 AP 的 Throughput 切換頻道 流量統計表

下面是各client的平均傳輸速率：

A 平均傳輸速率：200.5 KB/S

B 平均傳輸速率：600 KB/S

C 平均傳輸速率：228.1 KB/S

D 平均傳輸速率：282.8 KB/S

網路流量總和：1311.4 KB/S

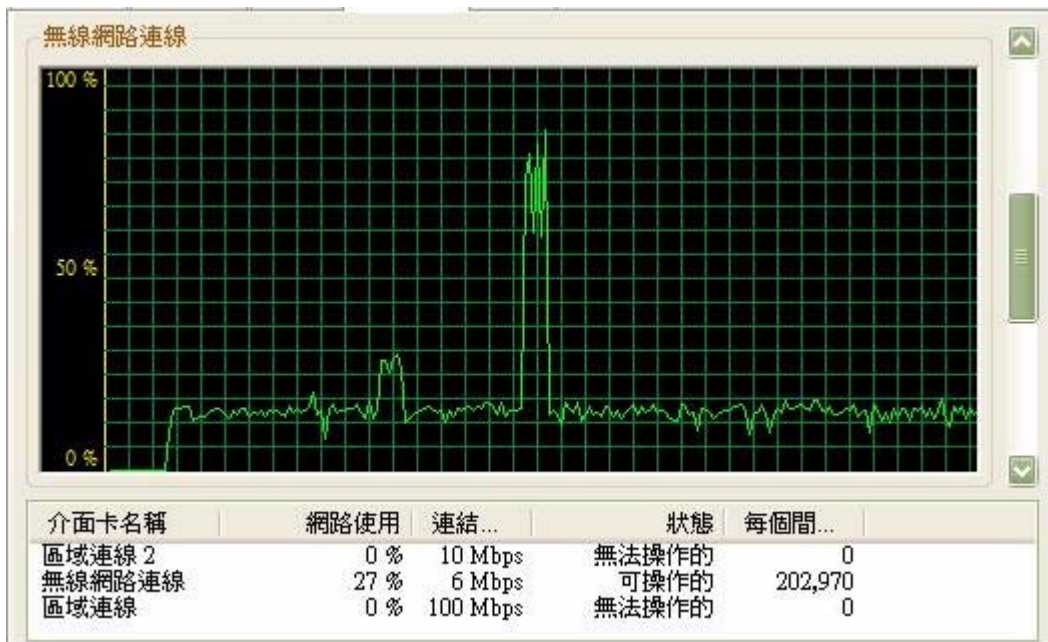


圖 4.24 實驗三 根據 AP 的 Throughput 切換頻道 Client A 實測流量圖

此方法中，與前面相同環境，分別在 AP2 灌入 600KB/S 的流量，及在 AP3 中兩 Client 各 300KB/S 的流量，但是 AP 是以 Throughput 為考量來切換頻道，最後的結果是 AP1 和 AP3 使用同一頻道。從流量統計圖中，可以看出，根據 AP 的 Throughput 切換頻道這個方法，所得到的網路流量總和平均有 1311.4 KB/S，雖然只比動態切換頻道的方法少一些，但是各 AP 所得到的頻寬卻相差一到兩倍，沒有動態切換頻道分配的均勻。因為只考量 Throughput 選擇頻道，可能選到許多 Client 所使用的頻道，因而造成必須跟很多使用者競爭頻寬，而另一頻道卻由一人獨享所有頻寬的不平均情形。

(3) 使用不切換頻道演算法

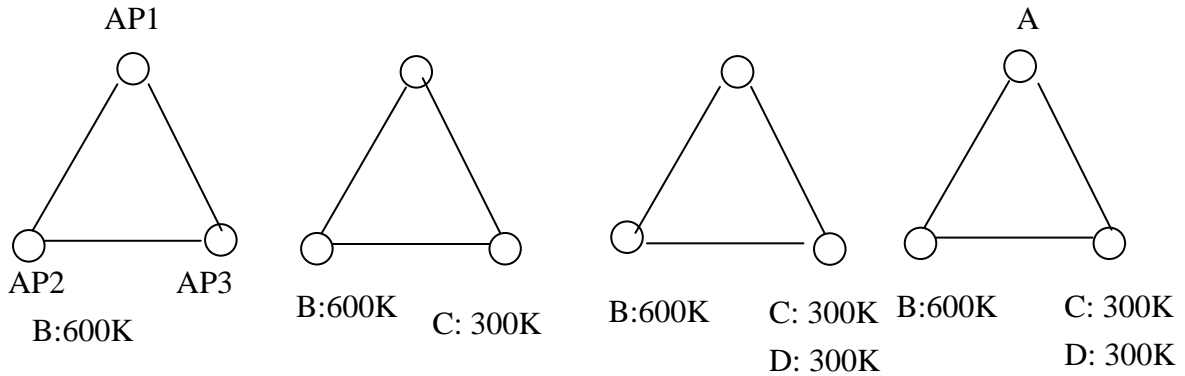


圖 4.25 實驗三 不切換頻道 示意圖

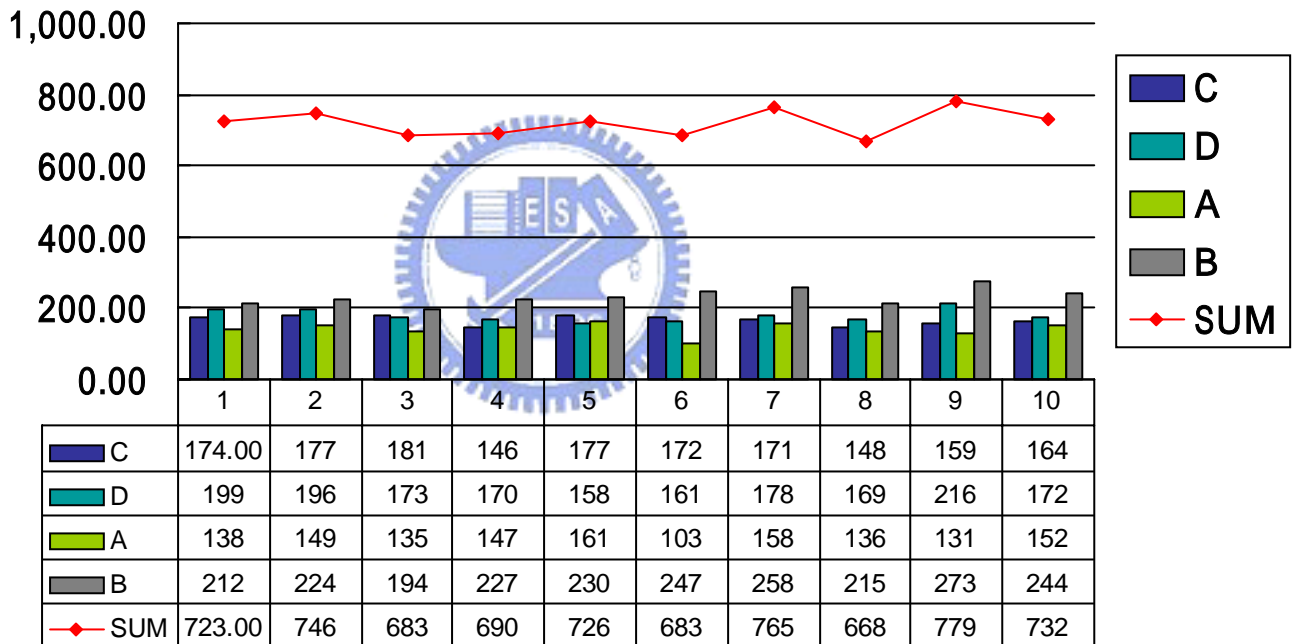


Table 4.7 實驗三 不切換頻道流量統計表

此方法中，與前面相同環境，分別在AP2灌入600KB/S的流量，及在AP3中兩Client各300KB/S的流量，但是AP全部使用同一頻道，結果Client A的平均傳輸速率為141 KB/S，Client B的平均傳輸速率為232.4 KB/S，Client C的平均傳輸速率為166.9 KB/S，Client D的平均傳輸速率為179.2 KB/S，而網路流量總和平均只有719.5 KB/S，遠不如前面兩個方法。

第五章 結論與未來工作

網路總流量 (KB/S)	策略一 動態頻道 切換	不切換	根據 client 數 切換	根據 Throughput 切換
實驗一	1249	660.45		
實驗二	1075	711.8	814.27	
實驗三	1330.58	719.5		1311.4

Table 5.1 總結 效能比較表

實驗一裡面，我們主要比較 AP 動態切換頻道和 AP 完全不切換頻道兩個方法，結果後者的總流量約只有前者的一半，而在後面兩個實驗中都可以看到，不切換頻道都是效能最差的。而實驗二中，將只考量 Client 數的方法加入比較，從結果可以看到在某些情況下其效能只比不切換頻道好一點點，遠遜於動態頻道切換的方法，因此僅考量 Client 數作頻道切換也是不足夠的。而實驗三則主要是比較根據 AP 的 Throughput 來作切換，雖然在一些情況下其所得到的總流量只比動態切換小一些，但是各個 AP 所分配到的頻寬沒有動態切換來的平均。從上面三個實驗，可以看到，我們的策略一動態頻道切換演算法，能夠隨著 Client 的移動以及 AP 上的流量變化而動態切換 AP 到適當的頻道，有效提昇整體網路的效能，其他三個演算法都有其較不適用的狀況，然而在三個實驗裡面，我們的方法普遍都能有效且平均的分配可用的頻寬。另外，演算法策略二以策略一為基礎，進一步考量 Neighbor's neighbor 的因素，但因為需要大型的實驗才能看出此因素對效能的影響，所以此次並沒有實驗評比策略二跟其他演算法的效能。

在未來改進方面，因為本演算法只更動 AP 端及後端 Server，在使用者裝置上並沒有任何更動，所以難免會發生 AP 更換 channel 但是其上的 Client 並沒有隨之更換，造成斷線的問題，如果能透過 Client 端程式的更改，提供 AP 端更多資訊，並且建立 AP 跟 Client 端雙向溝通的機制，則後端 Server 程式便可以取得更多有利資訊有助於判斷 AP 的頻道轉換，也能夠更明確的指定 Client 端連接哪個 AP，使得整個網路的頻寬分配更有效率。

參考文獻

- [1] IEEE Std 802.11 Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, June (1997).
- [2] IEEE Std 802.11a Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, September (1999).
- [3] IEEE Std 802.11b Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, November (2001).
- [4] IEEE Std 802.11g Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, June (2003).
- [5] 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.
- [6] Gerard G. Cervello, Sunghyun Choi, Stefan Mangold, and Amjad Soomro, "Dynamic Channel Selection (DCS) Scheme for 802.11," IEEE 802.11-00/195r2, September 2000.
- [7] Shiann-Tsong Sheu and Chih-Chiang Wu, "Dynamic Load Balance Algorithm (DLBA) for IEEE 802.11 Wireless Lan", Tamkang Journal of Science and Engineering, vol 2, No 1, pp.45-52(1999).
- [8] I. Papanikos and M. Logothetis, "A Study on Dynamic Load Balance for IEEE 802.11b Wireless LAN", Proc. 8th International Conference on Advances in Communication & Control, COMCON 8, Rethymna, Crete, June, 2001.
- [9] S.-L. Wu , C.-Y Lin , Y.-C. Tseng , J.-P. Sheu, A New Multi-Channel MAC Protocol with On-Demand Channel Assignment for Multi-Hop Mobile Ad Hoc Networks, Proceedings of the 2000 International Symposium on Parallel Architectures, Algorithms and Networks (ISPAN '00), p.232, December 07-07, 2000

- [10] W. Hung, K. Law and A. Leon-Garcia, "A Dynamic Multi-Channel MAC for Ad Hoc LAN," in Proc. of 21st Biennial Symposium on Communications, April 2002.
- [11] A. Nasipuri, J. Zhuang and S. R. Das, "A Multichannel CSMA MAC Protocol for Multihop Wireless Networks," in Proc. of IEEE Wireless Communications and Networking Conference (WCNC), September 1999.
- [12] J. So, N. Vaidya; ``Multi-Channel MAC for Ad Hoc Networks: Handling Multi-Channel Hidden Terminals Using A Single Transceiver"; Proc. ACM MobiHoc 2004
- [13] R. Garces and J.J. Garcia-Luna-Aceves; ``Collision Avoidance and Resolution Multiple Access for Multichannel Wireless Networks"; Proc. Infocom 2000.
- [14] C. Y. Chang, P. C. Huang, C. T. Chang, and Yuh-Shyan Chen, "Dynamic Channel Assignment and Reassignment for Exploiting Channel Reuse Opportunities in Ad Hoc Wireless Networks", IEICE Transactions on Communications, Vol.E86-B, No.4, pp. 1234-1246, April 2003. (SCI, EI)
- [15] Ashish Raniwala, Kartik Gopalan, Tzi-cker Chiueh, "Centralized Channel Assignment and Routing Algorithms for Multi-Channel Wireless Mesh Networks", ACM SIGMOBILE Mobile Computing and Communications, Volume 8 , Issue 2 ,April, 2004

Curriculum Vita

Yu-Fen Tai (yftai@csie.nctu.edu.tw) received her B.S. degree in Computer Science and Information Engineering from the National Cheng Kung University, Taiwan, in 2002. Her current research interests include wireless network, sensor network and Bluetooth.

