

# 行政院國家科學委員會專題研究計畫 成果報告

## 子計畫二：多階層隨意網路上具位置知覺的繞徑協定設計及 實作(2/2)

計畫類別：整合型計畫

計畫編號：NSC93-2219-E-032-002-

執行期間：93年08月01日至94年07月31日

執行單位：淡江大學資訊工程學系

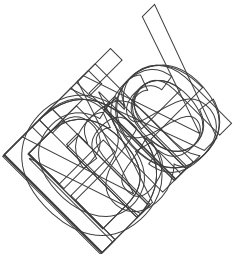
計畫主持人：張志勇

共同主持人：陳裕賢

報告類型：完整報告

報告附件：出席國際會議研究心得報告及發表論文

處理方式：本計畫可公開查詢



中 華 民 國 94 年 9 月 16 日

# 行政院國家科學委員會專題研究計畫成果報告

## 子計畫二：多階層隨意網路上具位置知覺的繞徑協定設計及實作(2/2)

計畫編號：NSC 93-2219-E-032-002

執行期限：93年8月1日至94年7月31日

主持人：張志勇 教授淡江大學資訊工程學系

共同主持人：陳裕賢 教授 中正大學資訊工程系

### 一、中英文摘要

本整合型計畫之多階層行動隨意網路是由Bluetooth模組、符合Wi-Fi標準之無線網路設備、及Cellular Systems (如GSM、GPRS、WCDMA/Cdma2000等)多套通訊系統所整合而成，其主要之目的是要實現一個多階層之行動隨意網路(Multi-Tier Mobile Ad Hoc Network)，使用者可不受時空限制，在這多階網路中享有整合性的通訊服務。本計畫為一個二年期的子計畫，其主要目的在研發Network Layer的通訊協定，透過不同的通訊媒介，在多階無線網路的環境中自動進行無線網路資源管理、頻道安排、通訊路徑建立、群播、廣播、資料收集與特定地區廣播等多樣化的通訊服務，供使用者以最經濟且穩定的方式進行通訊。在第二年的計畫中，我們主要的工作在理論部份為(1)研發Bluetooth Scatternet網路中Efficient Multi-Channel Management通訊協定(2)研發802.11 WLAN網路中Efficient Multi-Channel MAC protocol with Directional Antenna。在實作部份為(1) GPRS 通訊模組的實作與操控(2) Bluetooth、802.11與 GPRS 跨階切換模組的實作。

關鍵字：Multi-tier Network、Bluetooth、802.11、GPRS、Protocol、Routing、Multicasting、Geocasting。

#### 英文摘要

Recently, the applications of Irda, Bluetooth, 802.11-based Wireless LAN, and GPRS cellular system have been widely used in communication network. The advances of computer technology and the population of wireless equipment have promoted the quality of our daily life. The trend of recent communication technology is how to make good use of wireless equipments for constructing an ubiquitous communication environment. This project is an integrated project of Multi-Tier Ad Hoc Network that consists of Bluetooth, 802.11, GPRS radio systems is considered in this project. The goal of the integrated project is to develop wireless technologies for providing users with high-performance communication services in the Multi-Tier wireless network. This sub-project is 2-year sub-project that mainly focuses on developing communication protocols for providing users with unicast, multicast, broadcast, information collection, and geocast services in Multi-Tier wireless network. In this year, the theoretical achievements include (1) Efficient Multi-Channel Management Protocol (2) An Efficient Multi-Channel MAC protocol for 802.11 Wireless LAN with Directional Antenna. In addition, the implementation achievements include (1) Implementation and Control of GPRS Communication Module (2) Implementation of Bluetooth, 802.11, and GPRS Three-Tier Seamless Handoff Module.

Key word: Multi-tier Network、Bluetooth、802.11、GPRS、Protocol、Routing、Multicasting、Geocasting。

### 二、緣由與目的

近年來，科技的進步，使得無線通訊蓬勃發展，也增進人類的生活品質。多種無線通訊技術的提出，讓人們可以依照不同環境、因素選擇適當的通訊技術，達到不同溝通的樂趣。然而，我們發現無線存取系統依其傳輸環境、行動率之不同，會有不同之最佳設計、傳輸速率與品質。因此，若能夠混合多種不同通訊技術，將可使人類的通訊世界更加人性化及便利。為達到多種不同性質混合成的通訊環境，我們提出一個多階隨連即連網路(Multi-tier Ad Hoc Network)架構。此架構乃集Bluetooth、IEEE 802.11、GPRS三種技術之優勢所建立而成。

Bluetooth是一種短距離、低成本的無線通訊技術，所建構的Piconet是以快速跳頻的方式在ISM 2.4GHz頻帶的79個頻道中快速變換頻道，以避免來自共用頻帶中其它訊號的干擾，跳頻的序列主要以Master的48-bit BD\_ADDR及其Clock來決定，同一Piconet中的成員均以Master的跳頻序列為標準，跟隨Master以每秒跳頻1600次的速度變換於79頻道中，其傳輸量可達1MB/秒，傳輸範圍可達10至100公尺。

IEEE 802.11是由美國電子電機工程協會(Institute of Electrical and Electronics Engineers)在1997年7月提出的無線區域網路之標準，主要是針對網路的實體層(Physical Layer; PHY)與媒體存取控制層(Media Access Control Layer; MAC)進行規定，使用於2.4 GHz ISM頻帶，在媒體存取控制層則改良IEEE 802.3所採用的CSMA/CD(Carrier-Sense Multiple Access/Collision Detection)技術成CSMA/CA(Carrier-Sense Multiple Access/Collision Avoidance)，以減少資料在傳輸碰撞的機會。在IEEE 802.11標準的規範中，將無線區域網路的架構分為兩種－無基礎建設(Ad Hoc)及具基礎建設(Infrastructure)之無線區域網路。然而，隨著WLAN需求的增加與應用的擴大，IEEE於1999年完成了兩種5.5、11Mbps傳輸速度的技術規格標準－IEEE 802.11b，其中實體層包括跳頻(Frequency Hopping)與直接序列(Direct Sequence)技術。繼802.11b推出之後，IEEE相關小組仍積極研發高速的WLAN標準，IEEE 802.11a即為其產物之一。其採用OFDM(Orthogonal Frequency Division Multiplexing)的調頻技術，由於OFDM的運作需要消耗較大的頻寬，因此，不適用在擁擠且可用頻寬較小的2.4GHz頻帶，而改運作在5GHz頻帶上，傳輸速率可從6Mbps達到54Mbps。

整合封包無線服務(General Purpose Radio Service；

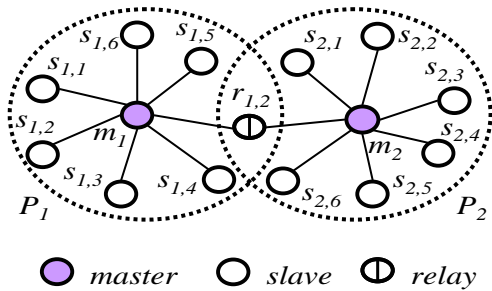
GPRS)，GPRS的標準是由ETSI提出，網路架構是利用現有的GSM網路再增加GPRS模組，而模組中的數據交換節點（SGSN、GGSN）則具有封包處理的能力，使得資料能以成串的封包進行傳送，因此，可有效地利用有限頻寬連結網際網路分享網際網路上的資源。

本計畫為一個二年期的子計畫，其主要目的在研發 Network Layer 的通訊協定，透過不同的通訊媒介，在多階無線網路的環境中自動進行無線網路資源管理、頻道安排、通訊路徑建立、群播、廣播、資料收集與特定地區廣播等多樣化的通訊服務，供使用者以最經濟且穩定的方式進行通訊。

### 三、今年度理論部份的成果

#### (1) Bluetooth Multi-Channel Management Protocol

這個通訊協定架構在已形成好的scatternet網路拓撲中，並且假設一個 piconet 最多只有兩個 relay 及相鄰的兩個 piconet 間僅存在一個 relay (如圖一)，以及一個 piconet 的所有 devices 彼此均在通訊範圍內，也就是當一個 device broadcast packet 時，piconet 內每個 device 都可以收到 packet。另外，為了節省 device 的電量消耗，device 的 power saving mode 採用 Hold mode，因此 device 的週期是固定的且進入省電模式的 device 不需要放棄其 AM\_ADDR，以便其可快速切換於 active 及 power saving mode 間。當 device 沒有要傳送資料及資料傳送完畢，就進入 sleep 狀態，達到省電效果。

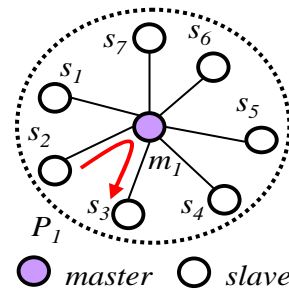


圖一：部份已形成的 Piconet topology 示意圖。

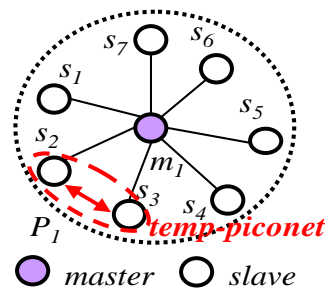
當 sender 或 receiver 使用 single-slot packet 傳送資料時，sender 和 receiver 會跳到 hopping sequence 的某個特定 channel 並且停留一個 time slot，當此次封包傳送完畢，sender 和 receiver 會依照 hopping sequence 跳到下一個 channel 繼續資料傳送，所以我們可以將 single-slot packet 視為擁有 single-channel resource 並且在這個 channel 進行資料傳送；而 sender 或 receiver 在使用 multi-slot packet 傳遞資料時，sender 和 receiver 也會跳到 hopping sequence 的某個特定 channel 並且停留 3 或 5 個 time slot，直到封包傳送結束，sender 和 receiver 會從現在的 channel 跳到 hopping sequence 中接下來的第 3 或 5 個 channel 繼續資料傳送，使得 hopping sequence 中的某些 channel 無 device 使用，因此我們將 multi-slot packet 視為有 multi-channel resource 並且多個 channel 中只有第 1 個 channel 有 device 進行通訊。所以當使用 multi-slot packet 或 single-slot packet 進行資料傳送時，在 hopping sequence 中皆只有一個 channel 被使用，並且在單一時間點上只有一對 device pair 進行通訊，我們將此稱為 single-channel 環境。另外，當資料傳輸是使用 multi-slot packet 時，因為 multi-slot packet 具有 multi-channel resource，如果 multi-slot packet 的 channel 資源有兩個以上 channel 被 device 使用時，在單一時間點上，可以有多對 device pair 平行傳輸，我們將此稱為 multi-channel 環

境。在封包長度為 5 個 time slot 的 packet 中，只允許同時最多 4 對不同 device pairs 平行通訊，因為在 piconet 中，master 加上最多七個 active slaves 只可以有不同的四對 device pairs 同時通訊，以避免有 co-channel 情況發生；在封包長度為 3 個 time-slot 的 packet 中，因為只有三個 channels 可供通訊，所以同時最多允許不同 3 對 device pairs 平行通訊。在 multi-channel 環境中，由於進行通訊的 device pairs 是平行傳送資料，通訊的 device pairs 是不可以有交集的，因為 sender 和 receiver 必須要跳到同一個 channel 進行資料傳送，直到資料傳送結束，所以一個 device 在單一時間點上只能待在某個特定 channel，也就是說，進行通訊的 device pair 是獨立的，以避免通訊失敗的問題。

我們將針對 piconet 頻寬使用的情形，舉例討論其所可能產生的問題，以及使用本通訊協定所提出的 multi-channel 管理協定後，其問題的改善程度，以達到充分使用頻寬傳送資料，並讓無通訊需求或可能造成過度競爭的 device 進入省電狀態的目的；另一個例子為簡述在 scatternet 中，relay 如何利用所研發的協定，有效率的切換於兩個 piconets 之間，以達到快速轉送資料的目的。



(a) Piconet  $P_1$  運作在無 time-slot leasing 機制的環境。



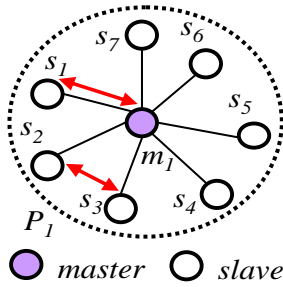
(b) Piconet  $P_1$  中，使用 time-slot leasing 機制的情况。

圖二：Piconet  $P_1$ ，有無使用 time-slot leasing 機制之比較情形。

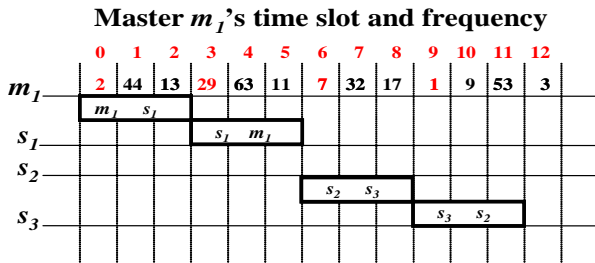
以下針對 piconet 的環境，說明 piconet 運作環境在無 time-slot leasing 機制和有 time-slot leasing 機制的差別。在 piconet 環境中，如圖二 (a) 所示，當 piconet 是運作在無 time-slot leasing 機制的環境下，s2 和 s3 要進行通訊時，是需要藉由 m1 來代傳資料，也就是 s2 需要先將資料傳送給 m1，m1 再將資料傳給 s3，此種 2-hop 的資料代轉方法會使得 m1 消耗電量及資料的 delay time 增長，整個 piconet 的 throughputs 減低。但是，當有 time-slot leasing 機制運作在 piconet 中，如圖二 (b) 所示，此時 s2 和 s3 欲進行通訊，是不需要 m1 幫忙代傳資料，在利用現有的 time-slot leasing 機制，s2 提出通訊的需求給 m1，經由 m1 的安排，m1 借用偶數 time slot 給 s2，使得欲通訊的 s2 和 s3 形成一個 temp-piconet，此 temp-piconet 的時間和頻率是同步於原本的 piconet。在 temp-piconet 中，s2 和 s3 可以直接進行資料傳送，使得只需要 1-hop 的傳輸次

數，如此可使資料的delay time減少及節省m1的電量，並且提高整個piconet的throughput。

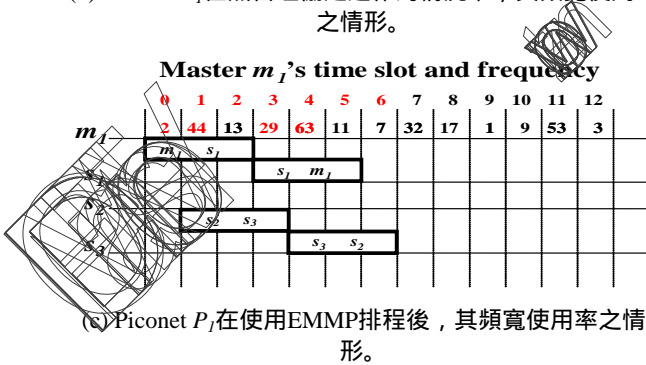
再來，針對有time-slot leasing機制運作的piconet，描述device在single-channel環境傳送資料時，可能產生的問題，及device在使用我們所提出的multi-channel管理協定後，其問題改善的情形。在single-channel的環境中，piconet P1如圖三(a)所示，有兩對device pair(device pair(s1,m1)、device pair(s2,s3))欲通訊，在有time-slot leasing機制下，device pair(s2,s3)可以直接通訊，但是在單一時間點上只有一對device pair傳送及回傳資料，如圖三(b)所示，所有的sender及receiver在傳送資料皆使用DH3 packet，但是在無任何管理協定的運作下，我們可以發現，使用multi-slot packet進行資料的傳輸，會導致頻寬的使用率降低及資料delay time增加。在圖三(b)中，兩對device pair的資料傳送，總共使用了4個channels(2,29,7,1)及12個time-slots，然而每次傳送DH3 packet 會造成2個channel沒有被使用，導致總共有8個channels(44,13,63,11,32,17,9,53)無device在傳送資料，造成頻寬的浪費，並且必須等待device pair(s1,m1)通訊完成後，另一對device pair(s2,s3)才可以開始進行資料傳輸，也就是在單一時間點上只允許一對device pair通訊，如此使得資料的delay time增長。



(a) Piconet  $P_1$ 中，有多對device pair欲通訊。



(b) Piconet  $P_1$ 在無管理協定運作的情况下，其頻寬使用率之情形。



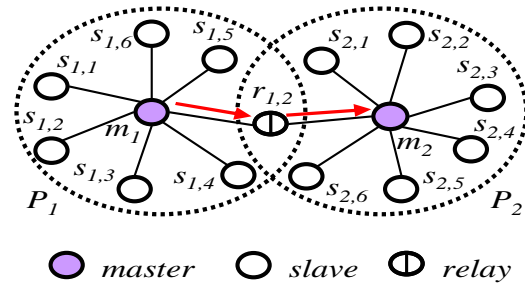
(c) Piconet  $P_1$ 在使用EMMP排程後，其頻寬使用率之情形。

圖三： Piconet  $P_1$ 未使用EMMP及使用EMMP 排程後頻寬使用率之比較情形。

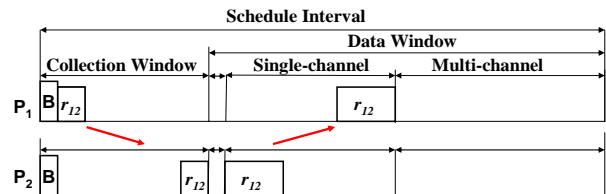
此時，若使用我們所研發的協定，master會根據各

device的頻寬需求，分配到適當的channel通訊，以營造出一個良好的通訊環境，達到最佳的頻寬使用狀況。如圖三(c)所示，所有欲通訊的device pair藉由multi-channel環境及規畫良好的排程，使得在單一時間點上可以有對device pair進行通訊，並且同時最多可以有四對device pairs在進行資料傳送，達到充分利用頻寬、提升throughput。在圖三(c)中，device pair(s1,m1)和device pair(s2,s3)的資料傳送總共只使用4個channels(2,44,29,63)及7個slots，以及只有3個channels(13,11,7)被浪費。相對於在single-channel環境通訊，具有管理協定的multi-channel環境，可以增加24%的channel utilization和節省58%的Slots，且當有愈多device pairs在同時傳送資料時，channel-utilization可愈提高。

經由以上的敘述，本通訊協定可以針對piconet的環境，利用multi-slot packet及time-slot leasing機制，建構出一個multi-channel的通訊環境，並且提出一個最佳的multi-channel管理協定，藉由master有效的排程，使得單一時間點上有對device pairs通訊，達到充分使用頻寬進行資料傳送，以提升系統performance，並且讓無通訊需求或可能造成過度競爭的devices進入省電狀態。



(a) Relay無效率地轉送相鄰兩piconet封包所產生延遲現象。



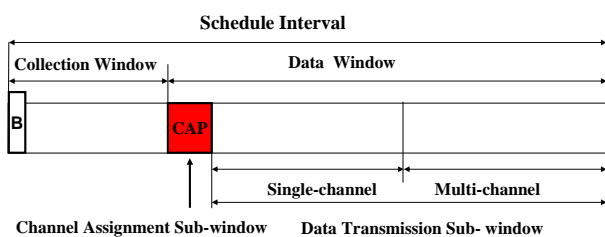
(b) Relay經EMMP特殊的安排後，迅速轉送相鄰兩piconet的封包之示意圖。

圖四： Relay未使用EMMP及使用EMMP管理後轉送相鄰兩piconet封包之比較情形。

以下我們將針對scatternet的環境，簡述relay在無規劃的情形下，轉送相鄰兩piconets封包所造成的data transmission delay，相對於使用本通訊協定所提出的機制後，relay迅速轉送相鄰兩piconets間資料的情形。在scatternet的環境中，如圖四(a)所示，r12為piconet P1及P2的relay，device m1及m2分別為piconet P1及P2的master。在scatternet中，piconet之間的device要傳送資料時，需要藉由relay幫忙代傳資料，也就是sender必須先將資料傳送給relay，再由relay傳送給receiver，但由於使用relay幫忙代傳資料，因而增加資料在相鄰兩piconet之間轉送的delay time。以圖四(a)為例，此時r12是參與piconet P1的資料傳輸，當piconet P1的m1要傳送資料給piconet P2的m2時，m1需先將資料傳送給r12，r12並回送一個null packet給m1，等到r12接收m1的資料結束後，r12將切換到piconet P2參加P2的資料傳送，但r12必須等待m2的polling才可以將資料傳送給m2。由於 r12必須接收完

m1所傳送的資料，然後r12從piconet P1切換到piconet P2是需要轉換的時間並且r12要等待m2的polling才能轉送資料給m2，所以此種的資料代傳方式會增加相鄰兩piconets間資料轉送的delay time。此時，若使用本通訊協定所研發的協定後，如圖四(b)所示，r12會先參與P1的Collection Window，將欲通訊的訊息傳送給m1，傳送完畢之後，再切換到P2的Collection Window並傳送欲通訊的訊息給m2，所以r12可以適當的切換於兩個 piconets 之間，參加每個 piconet 的 Collection Window，並將欲通訊的訊息傳送給每個master。在Data Window開始時，r12將繼續參與P2的Data Window，以減少切換piconet的次數，並接收或傳送跟r12相關的資料，在完成這些跟r12相關的資料傳輸後，r12再切換到P1的Data Window，停留在P1的Data Window接收或傳送跟r12相關的資料。所以，r12能適當地切換於兩個piconets之間的Data Window並且進行資料代傳，以減少資料在相鄰兩piconets之間傳送的delay time。

本通訊協定的主要目的是發展一有效率的Multi-channel Management Protocol，使其能適用於piconet、scatternet。因此，經由以上例子的問題描述及協定運作的結果得知：我們所研發的協定仍然依照Bluetooth的規定，並與其相容。本通訊協定中並提出在每個piconet都有其各自的Schedule Interval，且在scatternet中所有piconet的Schedule Interval都會經由relay的協調，使得每個piconet的Schedule Interval皆相差在一個time slot之間，以避免relay無法參加兩個piconet的Collection Window。一個Schedule Interval主要分成Collection Window以及Data Window兩大時段，如圖五所示，其中Data Window又分成兩個Sub-window：Channel Assignment Sub-window以及Data Transmission Sub-window，在Data Transmission Sub-window中又分成Single-channel和Multi-channel兩個環境的transmission時段，在進行Data Transmission Sub-window時會先進行single-channel環境時段再進行multi-channel環境時段，使得relay可以快速代傳資料。因此，本通訊協定所提出的EMMP主要區分為下列三個Phases：



圖五：EMMP中，Schedule Interval的各時段分配。

- Phase I : Collection Phase.** 此 Phase 發生於 Collection Window 時段中，主要是 master 搜集欲通訊者的資訊。首先由 master 發 beacon 通知 piconet 內所有的 slaves，公告此次 Schedule Interval 中可以傳送資料的 slaves，並且 assign 一個 time slot 給每一個可以傳送資料的 slaves，獲得 time slot 的 slave 如果欲通訊，會在其 time-slot 發送 通訊需求的 packet 給 master，master 收到訊息後會將欲通訊的 device pair 之 bandwidth requirement 紀錄於相關的 matrix 中，並針對這些 device 的角色，給予一個權重來讓 master 區分那些 device pairs 擁有較高的通訊優先權，如此的做法，主要是希望 master 透過所紀錄的資訊，在 Phase II 中挑選出適合通訊的 device pairs。

- Phase II : Channel Assignment Phase.** 此 Phase 發生在 Channel Assignment Sub-window 時段中，主要是由 master 根據在 Phase I 中所紀錄的資訊，挑選並計算適合通訊的 device pairs，並依照各 device pair 的資料傳輸類型，將適合 single-channel 環境的 device pairs 挑選到 single-queue 中及適合 multi-channel 環境的 device pairs 挑選到 multi-queue 中，讓 device pairs 分配到適當的 channel，以避免 co-channel 的情況發生。最後 master 將分配 channel 的結果以廣播方式通知 piconet 中每個 device，而所有 devices 在收到這個廣播封包後，就準備執行 Phase III 的動作。
- Phase III : Data Transmission Phase.** 此 Phase 發生於 Data Transmission Sub-window 時段中，所有 devices 在收到 master 的廣播封包後，會依照排程的順序，先進行 single-channel 環境的資料傳輸，再進行 multi-channel 環境的資料傳輸。被允許通訊的 device pairs 會根據封包內容切換到指定的 channel 通訊；未被允許通訊的 device 會在 Data Transmission Sub-window 時段中進入省電狀態，如此的做法，可以讓不適合通訊的 device 省電，更可以為進行通訊的 device 營造一個良好的通訊環境，減少不必要的干擾。

由以上的敘述，已簡略的說明我們所提出的EMMP各Phase的工作目標及理念。由於在Phase I中，master可在Collection Window時段中，紀錄及分析每對欲通訊的Device pair之封包傳輸類型及頻寬需求，因此，master可以在Phase II中作有效率的通訊排程。而在Phase III中，允許通訊的device pair會根據master在Phase II所廣播的封包資訊，切換至適當的channel通訊，被禁止通訊的device pair則會進入省電狀態，以減少對進行通訊的device pair不必要的干擾，並達到省電的功能。

因此，依照各Phase的運作，我們針對在multi-channel環境中，所產生的通訊問題給予適當的解決，並利用通訊的優先權及頻道的排程，將可使relay順利參加兩piconets間的Collection Window及Data Window，至於其相關的運作細節我們將在下一章中描述。如此，本通訊協定將可以達到營造一個良好的通訊環境、channel utilization高、device省電、減少相鄰兩piconets間的Data transmission delay time等目標。

## (2) Multi-Channel MAC protocol for WLAN with Directional Antenna

在802.11的研究方面，我們考慮以Multi-channel為環境，且在配有方向性天線的硬體設備下，設計一個更有效率的802.11 MAC協定。透過Control Channel中RTS/CTS封包的交換，我們所研發的協定以Host ID所產生的Channel Hopping Sequence作為變換頻道的依據，以達到不需溝通便能快速變換頻道的目的，解決兩對通訊的主機在Data Channel產生封包衝撞時的窘境。此外，透過方向性天線的狀態管理，本論文所研發的協定將具有使Multi-channel能充分被利用、維持公平性、增加空間利用率等優點，並將802.11在Priority控制上的優點予以延伸到Multi-channel with Directional Antenna的環境。透過實驗數據顯示，我們所研發的協定大幅提升了空間上的利用率，並且維持公平性及具有較高的bandwidth utilization與throughput等優點，Wireless LAN中的主機將可因此而提升其資料傳輸的效能。

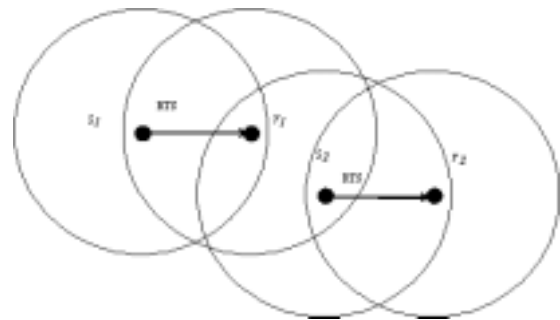
我們所研發的協定乃運作在具有一個control channel與

n-1個 data channel 的 Multi-channel 通訊環境，現今 MICA2 Mote 已可提供多個 channel，例如，MPR500CA 在 916MHz 下可提供至少 100 個 channel。在硬體設備上，每個 device 皆配有 180° 方向性天線，增加無線網路環境中空間上的利用率。而在本論文中，不考慮 device 移動的情況，每個 device 皆為靜態且所使用的 180° 方向性天線可選擇以 omni-direction 接收資料，若不同方向同時接收到不同的訊號時，則會發生封包 collision，此外，此 180° 方向性天線不論是接收或傳送資料，若有需要，均可將某方向 block 住，使此 block 住的方向無法傳送或接收資料，只能以另一個 180° 方向接收或傳送資料。

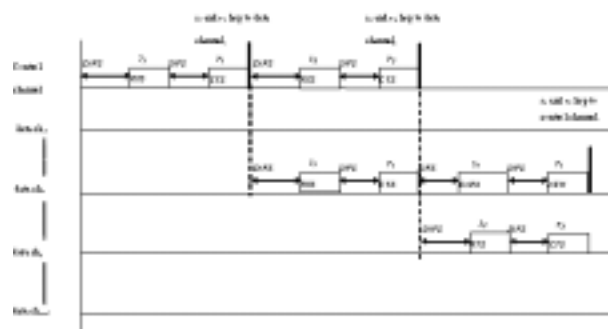
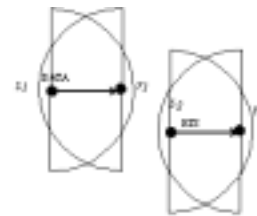
以下我們以圖六為例來說明本論文大致的做法及觀念。如圖六(a)中所示，欲通訊的 devices 均須先在 control channel 中，利用 IEEE 802.11 DCF 協定競爭通訊的機會，sender 以 omni-directional 的方式傳送 RTS 封包，receiver 以 directional 的方式傳送 CTS 封包以競爭通訊的機會，其中 RTS 及 CTS 分別含有 sender 及 receiver 之 48-bit MAC address，例如有兩對 devices，s1 欲與 r1 通訊，s2 欲與 r2 通訊，若此兩對欲通訊的 pairs 分別先後競爭到通訊的機會，則先競爭到通訊機會的 pair(s1, r1) 會利用其 sender 與 receiver 彼此的 48 bits ID 產生一 data channel hopping sequence，根據此 hopping sequence，sender 與 receiver 不需再做額外的溝通，便具跳頻共識，跳頻至 data channel i 中進行資料傳輸。如圖六(b)中所示，pair(s1, r1) 再以 directional antenna 競爭在此 data channel 中資料傳輸的機會，並以 directional antenna 作 data packet 的傳輸；而 device s2 與 device r2 也根據其 hopping sequence hop 至 data channel j 中，此時 device s1 與 device r1 在 data channel i 中以 directional antenna 競爭到通訊機會並傳送 data packet，device s2 與 device r2 同樣地在 data channel j 中競爭可在此 data channel 內通訊的機會。由於 pair(s1, r1) 及 pair(s2, r2) 其 MAC address 不同，因此可降低這兩對 devices 選擇同一 data channel 的機會。再者，又因 hopping sequence 為 sender 及 receiver 之 MAC address 產生，通訊雙方均具有此資訊，因此可達到不需溝通，即可共同以相同的 hopping sequence 具共識地選擇一恰當的 channel 通訊。爾後，通訊的 pairs 在 data channel 中通訊結束時，則立刻切換回 control channel。在此例中，僅兩 pairs 欲進行通訊，當欲進行通訊的 pair 數增多時，sender 與 receiver 依 channel hopping sequence 所跳頻的 data channel，可能有其他對主機正在通訊，其情況將較此例複雜；如圖六(c)所示，若正好有第三對欲互相通訊的 pair，device s3 與 r3，其在 control channel 已競爭到互相通訊的權利後，隨即依據彼此的 48 bit IDs 產生一 hopping sequence，若 device s3 與 r3 依據 hopping sequence 跳至 data channel i 中，但 device r3 carrier sense 到 s1 正在傳送 data packet 給 r1，而 r3 判斷出 s1 與 s3 皆在其左邊，此時 r3 以其方向性天線的左邊傳送封包並不會影響 s1 的傳輸與 r1 的接收。則 r3 立即以左邊的 directional antenna 傳送一個 Warning packet 給 s3，並跳至 hopping sequence 中的下一個 channel，假設為 channel k，s3 收到此 Warning packet 後也隨即跳至 channel k 中，若 s3 與 r3 carrier sense 一段 DIFS 時間後，皆無 devices 在此空間中通訊，則 s3 與 r3 即可在 channel k 中通訊。如此，使原本已經在通訊的 pairs 可以不受受到干擾，而且使後來欲通訊的 pair 可以跳至其他 channel 中競爭通訊的機會，有效的擴充 Multi-channel 與 directional antenna 在空間與時間上的優點。本論文在下個章節中將敘述我們所研發的協定如何利用 Multi-channel 及方向性天線開發空間及時間維度上的利用率，並且有效的避開封包衝撞的問題。

我們所研發的協定主要分成兩個 Phases，意即為

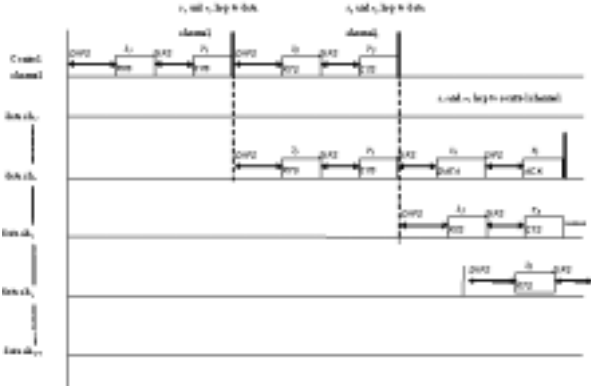
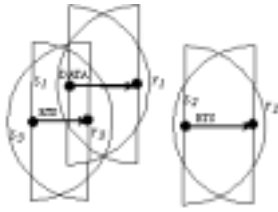
Negotiation Phase 與 Communication Phase。Negotiation Phase 為兩欲互相通訊的 devices 在 control channel 或 data channel 中，彼此交換 RTS/CTS packet 以競爭彼此通訊的機會；在 control channel 競爭通訊機會成功的 pair(s<sub>i</sub>, r<sub>i</sub>)，則變換到 data channel 中，利用 data packet 交換資料，此階段稱之為 Communication Phase。最後，當欲通訊的主機數量遠超過 data channel 數量時，我們所設計的 Block Phase 便啟動以避免主機因一再變換頻道所增加之成本。



(a) 每對欲通訊者在 control channel 中以 omni-directional antenna 競爭彼此通訊的機會



(b) 利用 hopping sequence 使多對欲通訊者至不同的 data channel 中通訊，開發頻道的利用率。



(c)有多對欲互相通訊的pairs跳頻至同一data channel時，雖然directional antenna開發了空間利用率，但仍有interference的情形。

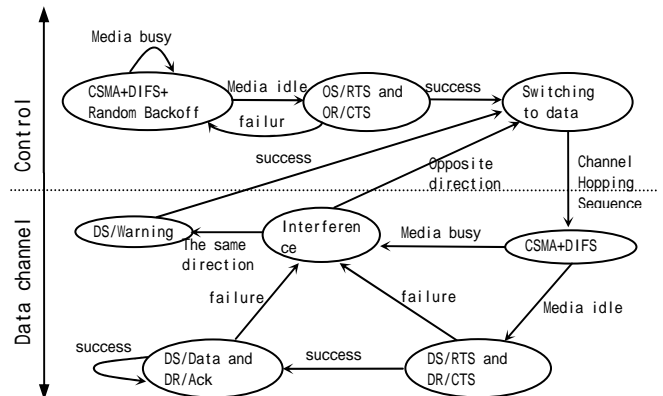
圖六：我們所研發的MAC protocol，利用Multi-channel與directional antenna開發空間與時間的利用率

以下，我們列出本協定之State Diagram 及其Pseudo Code。首先，在Normal phase中，如圖七(a)所示，role為sender的devices會先在control channel中依IEEE 802.11 MAC協定carrier sense一段DIFS與random backoff時間後，若Media仍為idle時，隨即以Omni-directional antenna傳送RTS packet給sender，再以omni-directional antenna接收其receiver所傳送的CTS packet，若接收CTS packet失敗，則又回到carrier sense的狀態重新競爭，若成功接收CTS packet後，則sender以其本身與receiver的48 bit IDs產生一data channel hopping sequence，並依此CHS在data channel中跳頻，sender與其receiver跳頻至data channel中後，sender在data channel中carrier sense一段DIFS時間後，若medium仍為idle時，則sender以directional antenna之其receiver的方向傳送RTS packet給其receiver，並同樣以其directional antenna之receiver方向接收CTS packet，若成功接收CTS packet後，則sender再以directional antenna的receiver方向傳送Data packet與接收Ack packet，若在接收CTS packet與Ack packet失敗時，則sender會以directional antenna之receiver方向傳送Warning packet給其receiver，通知其receiver一起跳頻至另一個data channel中重新競爭通訊的機會。

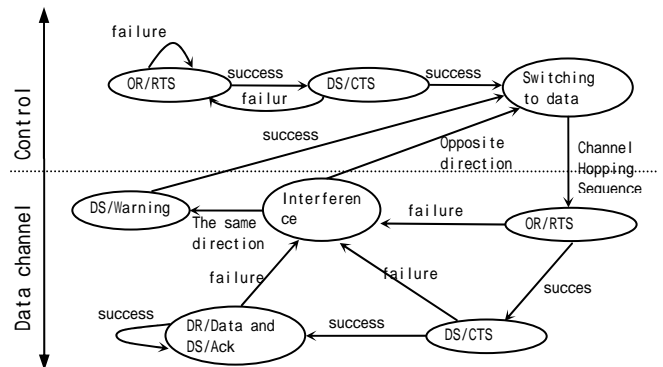
而如圖七(b)中所示，role為receiver的device在Normal phase中，receiver會先在control channel中以omni-directional antenna接收其sender所傳送的RTS packet，成功接收RTS packet後，receiver會以directional antenna之sender方向傳送CTS packet給其sender，然後依receiver與sender本身之48 bit IDs所產生的data channel hopping sequence跳頻至data channel中，當receiver跳頻至data channel中後，receiver以omni-directional antenna接收sender所傳送的RTS packet，成功接收RTS packet後，再隨即以directional antenna之sender方向傳送CTS packet，待傳送CTS packet成功後，則同樣以directional antenna之sender方向接收Data packet與傳送Ack packet，若receiver接收RTS packet與Data packet、傳送CTS

packet與Ack packet任一個state失敗時，則receiver以其directional antenna之sender方向傳送Warning packet，通知sender與其一起跳頻至下一data channel中重新競爭。

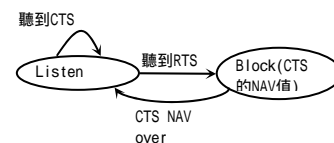
而在Normal phase中，其他非sender或receiver的devices其state diagram如圖七(c)所示，other devices平時皆在control channel中Listen，若收到RTS packet時，則將其directional antenna之收到RTS packet方向Block住，其duration為待CTS packet傳送完後，若other devices是接收到CTS packet時，則繼續在control channel中Listen。



(a)Normal phase中，sender的state diagram。



(b)Normal phase中，receiver的state diagram。



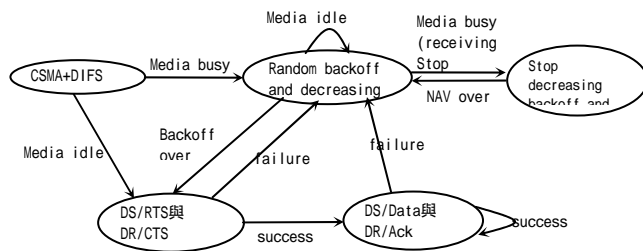
(c)Normal phase中，other devices在control channel內的state diagram。

圖七本論文所提出的protocol，其devices在Normal phase的state diagram。

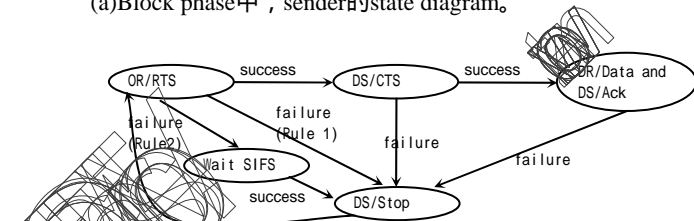
最後，當網路中欲通訊對的數目遠大於data channel的數目，導致每個data channel內的密度皆很高，使欲通訊對不停地switch data channel，當欲通訊對依據其data channel hopping sequence跳頻過所有的data channels而仍競爭不到通訊的機會時，則此欲通訊對將進入Block phase狀態，進入Block phase的pairs將跳回其data channel hopping sequence的第一個data channel中，並留在此data channel內競爭通訊的機會，以下我們將敘述Block phase的state diagram，將device的role分為sender與receiver兩方面來討論，如圖八(a)中所示，role為sender的device會在其data channel hopping

sequence的第一個data channel內carrier sense一段DIFS的時間，若此時Media仍為idle時，則sender以directional antenna之其receiver方向傳送RTS packet給receiver，然後同樣以directional antenna之其receiver方向接收receiver所傳送的CTS packet，反之，若當Media為busy時，則sender會random backoff一段時間，並當sender carrier sense media為idle時，則decreasing the backoff time，若sender carrier sense到Media為busy、或是收到其receiver所傳送的Stop packet時，則sender將停止decreasing backoff time且設定等待的NAV值，直到此NAV值結束後再繼續decreasing backoff time，而當sender遞減完其backoff的時間後，sender同樣再以directional antenna之其receiver方向傳送RTS packet與接收CTS packet，若傳送RTS packet或接收CTS packet失敗，則在進入random backoff且decreasing backoff time的狀態；若當sender皆成功地傳送RTS packet與接收CTS packet後，則sender將同樣以其directional antenna之receiver方向傳送Data packet與接收Ack packet，若有一者失敗，則再進入random backoff狀態重新競爭。

而在Block phase中，當role為receiver的device會在其data channel hopping sequence中的第一個data channel內，以omni-directional antenna接收其sender所傳送的RTS packet，若成功地接收到sender的RTS packet後，則receiver會以其directional antenna之sender的方向傳送CTS packet給sender，然後receiver會block住directional antenna之非sender位置的方向，只以directional antenna之sender方向接收Data packet與傳送Ack packet，此外，若receiver在接收RTS packet與Data packet，或傳送CTS packet與Ack packet，受到干擾發生collision時，receiver會立刻以其directional antenna之sender方向傳送Stop packet，通知其sender 停止decreasing backoff and waiting，且設定並等待NAV值結束，而receiver則回到以omni-directional antenna接收其sender的RTS packet狀態。



(a)Block phase中，sender的state diagram.



(b)Block phase中，receiver在data channel的state diagram.

圖八：本協定之Block phase的state diagram.

以下，為我們所提出的protocol之pseudo-code，其中我們採用與IEEE 802.11相同的defer timer，如SIFS、DIFS、PIFS...等，此外，亦採用IEEE 802.11所定義的backoff timer、contention window (CW)、network allocation vector (NAV)。

```

Procedure MAIN ( )
Begin
  Set each of the antenna state as IDLE
  Initialize NAV for the directional antennas
  Initialize channel to control channel
  Do
  If (packet p is detected) then
  If (packet p is detected in buffer) then Call procedure Sender (p)
  ElseIf (destination is it self) then Call procedure Receiver (p)
  Else Set antenna NAV
  End If
  Loop
  End
  
```

```

Procedure Sender (packet p)
Begin
  If (antenna state is IDLE) then
  If (After deferring for a DIFS antenna state is IDLE)
  then Call OmniSend_packet (RTS packet)
  Else Reset Defer Timer and Start Backoff Timer ( )
  End If
  If (Type of OmniReceive_packet (r) is CTS) then
  Set antenna to the first channel of hopping sequence
  Do (antenna state is IDLE)
  If (After deferring for a DIFS antenna state is IDLE) then
  Call DirSend_packet (RTS)
  Set CTS Timeout Timer ( )
  If (Type of DirReceive_packet (r) is CTS) then
  Call DirSend_packet (p)
  ElseIf (CTS timer is expired) then
  DirSend_packet (Warning)
  If (Hopping Sequence is not end) then
  Hopping to next channel
  Else
  Hopping to first channel and waiting to content to transmit
  /* Block Phase */
  End If
  If (Type of DirReceive_packet (r) is ACK) then
  Reset contention window
  Start Backoff Timer ( )
  Set antenna IDLE
  Hopping to control channel
  Resume to transmit next packet
  Else
  DirSend_packet (Warning)
  If (Hopping Sequence is not end) then
  Hopping to next channel
  Else
  Hopping to first channel and waiting to content to transmit
  /* Block Phase */
  End if
  Else
  DirSend_packet (Warning)
  if (Hopping Sequence is not end) then
  Hopping to next channel
  Else
  Hopping to first channel and waiting to content to transmit
  /* Block Phase */
  End If
  Loop
  If (Hopping Sequence is not end) then
  Hopping to next channel
  Else
  Hopping to first channel and waiting to content to transmit
  /* Block Phase */
  Else Error_handler ( )
  End
  
```



```

Procedure Receiver (packet p)
Begin
  DirSend_packet (CTS)
Set antenna to first channel
  Set RTS Timeout Timer ( )
  Do (Timer is not expired)
    If (Type of OmniReceive_packet (r) is RTS) then
      DirSend_packet (CTS)
    If (Type of DirReceive_packet (r) is DATA then
      DirSend_packet (ACK)
      Reset contention window
  Start Backoff Timer ( )
  Set antenna IDLE
  Hopping to control channel
  Resume to transmit next packet
Else
  DirSend_packet (Warning)
if (Hopping Sequence is not end) then
  Hopping to next channel
Else
  Hopping to first channel and waiting to receive RTS /* Block
  Phase */
  End If
Else
  DirSend_packet (Warning)
if (Hopping Sequence is not end) then
  Hopping to next channel
Else
  Hopping to first channel and waiting to content to receive RTS
  /* Block Phase */
  End If
  Loop
  If (Hopping Sequence is not end) then
  Hopping to next channel
  Else
  Hopping to first channel and waiting to receive RTS /* Block
  Phase */
  End If
End

```

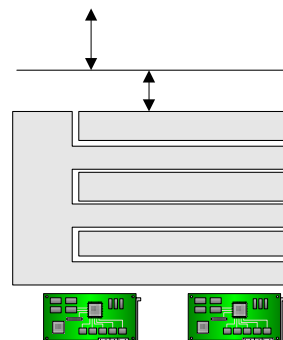
#### 四、今年度實作部份的成果

##### (A) 實作之軟體元件與基本概念

在介紹今年的成果之前，先說明兩項實作時的重要概念與系統架構。

##### (1) 控制 802.11 底層傳輸的相關軟體元件

我們的系統主要是發展在 Microsoft Windows 的環境上，並以 Microsoft Visual Studio .NET 作為開發之工具，此開發工具是易於使用的視窗開發介面，擁有龐大而完備的函數庫 (Library)，和詳細的線上文件 (Online Document) [1][3][4][5]。在整個實作的過程中，因需要能夠直接控制 802.11 無線網路卡和 GPRS 數據卡，在此簡單介紹於 Microsoft Windows 中，控制 802.11 Wireless Device 所需要瞭解的軟體元件，GPRS 的控制將於下一節中介紹。

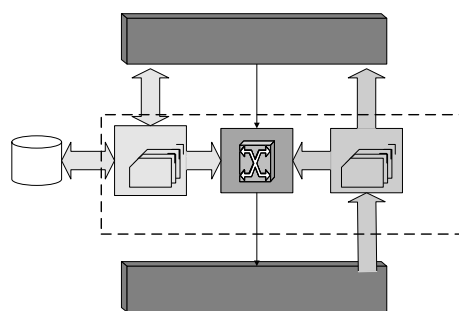


圖九：NDIS 函數庫架構圖

圖九是有關 802.11 無線網路卡控制的 NDIS 函數架構圖，NDIS 全名為 Network Driver Interface Specification，為 Microsoft 所訂定，其中包含了 Miniport Driver、Intermediate Driver 和 Protocol Driver 介面的標準、Hook Function 的定義和資源使用規範。由於 NDIS Driver 為 Kernel Mode Driver，所以不當的程序或命令，可能導致作業系統不穩定、資料流失，嚴重可能會導致系統損壞 (Crash)。

每一個 Miniport Driver 皆擁有自己一份 Manage Information Base (MIB)，MIB 中包含目前的設定參數和統計參數，MIB 中的這些參數 (Element)，NDIS 稱它為 Object。NDIS 定義了兩個函數 - QueryInformationHandler 和 SetInformationHandler，來讀取和設定管理這些 MIB。NDIS 亦定義了 Object Identifier (簡稱 OID)，來提供 Windows 系統本身對特定的 MIB 參數作讀取和寫入的動作。舉例說明，Windows 系統就是透過這些 OID 來取得網路卡目前的連線狀況。在此以實作 802.11 控制模組的方式，透過對這些 MIB 參數的設定，便可控制整個 802.11 無線網路卡。

802.11 無線網路控制模組除了提供控制 802.11 無線網路卡的方法，也紀錄和管理 802.11 Wireless 連線設定。在此提出了最佳連線品質方法 (Best Quality Connection)。在 Best Quality Connection (BQC) 的方法中，由 Parameter Manager、Site Survey Information Manager 和 Arbiter 等三個模組所組成 (圖十)。



圖十：802.11 控制模組方塊圖

Parameter Manager 和 Site Survey Information Manager 皆由 LinkList 所串連起來的資料鏈，分別擁有著連線設定參數和 Site Survey 的資料。

Parameter Manager 儲存著連線時所需的設定參數，使用者透過輸入視窗將已知的連線設定一一輸入並存檔，以供後續參考使用。

Site Survey Information Manager 每 0.5 秒呼叫 WlAdapter Class 所提供函數，由 WlAdapter 所宣告的物件上，取得一

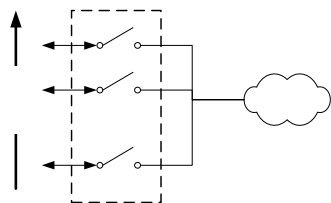
份Site Survey Information 清單，這份清單是由無線網卡驅動程式(Driver)所建立的，清單內容為無線網卡所能夠感應(Sense)到的存取點(AP)和無基礎建設(Ad Hoc)通訊。Site Survey Information Manager隨時保持最新的清單(List)以提供Arbiter參考。

Arbiter決定觸發連線的時機和選擇訊號品質最佳的連線組合。Arbiter由Site Survey Information Manager取得Site Survey Information List，和由Parameter Manager取得Parameter List，將二者以BSSID、模式和加密設定作配對，並以訊號強度作排序，以決定何為最佳連線品質設定參數。Arbiter決定觸發連線的時機有二，一為上層應用程式要求建立連線時，二為當連線訊號過低時。

(2)多階層之行動隨意網路切換原則

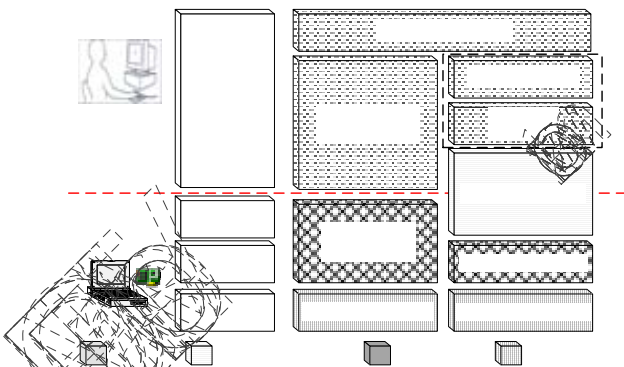
多階層之行動隨意網路切換媒體的設計概念有下列三個重點：

- 系統中，所有階層的無線網路，依連線速度、成本，決定各異質性無線網路的連線優先權，以速度越快、成本越低的連線的優先權越高。
- 要求擁有較高優先權的介面建立連線。若無法建立連線，則以次高優先權的介面建立連線，以此類推。
- 若目前連線並非最高優先權介面，定期嘗試使用更高優先權的介面建立連線。若成功建立較高優先權連線時，則中止目前連線。(如圖十一)



圖十一：決定連線介面之優先權

(B) 系統架構與功能



圖十二：802.11和GPRS控制模組架構

802.11和GPRS控制模組使用了很多Microsoft Windows的函數，且Microsoft Windows對於Kernel Mode和User Mode的程式有不同的規範，圖十二表示802.11和GPRS制模組在Microsoft Windows的地位及相關的架構。今年我們所實作的跨階系統架構與其功能大致可區分為

(1)GPRS 通訊模組的實作與操控

(2) Bluetooth、802.11與 GPRS 跨階切換模組的實作

等兩大部份。以下我們分別敘述這兩大部份的系統架構與功能。以下分別敘述這兩大工作內容。

(1)GPRS 通訊模組的實作與操控

GPRS 通訊模組的操控在Microsoft Windows中是以Remote Access Service (RAS)函數組完成，其中有可顯示RAS通用對話盒(RAS Common Dialog Boxes)、管理遠端存取連線和遠端存取元件(Device)和處理、建立電話簿等功能。透過RAS函數可直接找到指定的撥號元件(Dial-up Device)和對撥號元件下達撥號連線或中止連線的命令，分述如下。

撥號連線是使用RasDial這個函數，呼叫這個函數時需使用一組名為RASDIALPARAMS的參數，其中包含了要撥的電話號碼、認證要用的使用者名稱及密碼。如果撥號成功時GPRS的連線同時被建立，可以取得一個RAS connection handle的數值，RAS connection handle是使用RAS函數的識別數值，不同的連線擁有不同的數值，在整個RAS通訊期間必需一直記錄這個數值。

要中斷連線是使用RasHangUP這個函數，這個函數只有一個參數，是要被中斷連線的RAS connection handle。呼叫這個函數後GPRS連線即會被中斷。

想要得到GPRS連線的狀態可以呼叫RasGetConnectStatus，同樣的這裏需要傳入RAS connection handle讓系統知道要傳回那一個連線的狀態，傳回的狀態有connected和disconnected兩種。

(2)Bluetooth、802.11與 GPRS 跨階切換模組的實作

Bluetooth、802.11與GPRS跨階切換模組可分為四個狀態

- Disconnected
- Bluetooth connected
- 802.11 connected
- GPRS connected

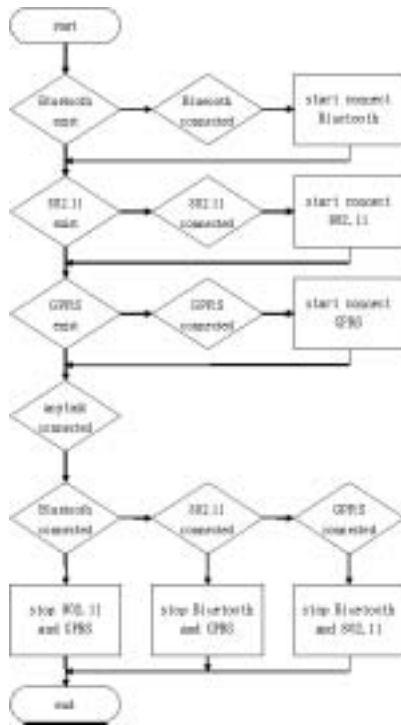
Disconnected是Bluetooth、802.11與GPRS跨階切換模組的初始狀態，這個狀態下要選擇一個適合的媒體建立連線，在其它的三個狀態下發生無法預期的事件時也會回到這個狀態重新開始。

Bluetooth connected是模組處於使用Bluetooth連線，這個時候模組要每隔十秒鐘監測連線的品質，並依據連線品質起動或中止另位兩個媒體的連線。

802.11 connected是模組處於使用802.11連線，其餘的動作和Bluetooth connected相同。

GPRS connected是模組處於使GPRS連線，其餘的動作和Bluetooth connected相同。

每個狀態都有不同的演算法，以達到任何時候至少有一個連線的目的，下面分別介紹各個演算法。



圖十三：disconnected狀態流程圖

● Disconnected 狀態的處理

Disconnected狀態的處理如圖十三的流程圖所示。因為建立連線需要花費一點時間，而每次建立連線的時間都不一，所以我們無法知道連線建立完成的正確時間，且那一個媒體是可完成連線建立我們亦無法得知，於是一開始時分別嘗試使用三種媒體建立連線，直到有一個以上的連線建立完成時，再依照先Bluetooth，其次802.11，最後GPRS的順序來保留一個連線，其餘的連線予以中斷。

● Bluetooth connected 狀態的處理

Bluetooth connected狀態的處理如圖十四的流程圖所示。為了能隨時得知連線的品質，要每隔十秒鐘執這個流程圖一次。這個流程圖主要是依照Bluetooth連線的品質決定是否要切換至其它的連線，如果連線的品質是低於RSSI lower時，要開始建立其它的連線，當連線的品質低於RSSI lowest時要完成切換的動作。如果802.11和GPRS都完成連線時保留802.11的連線中斷GPRS的連線。

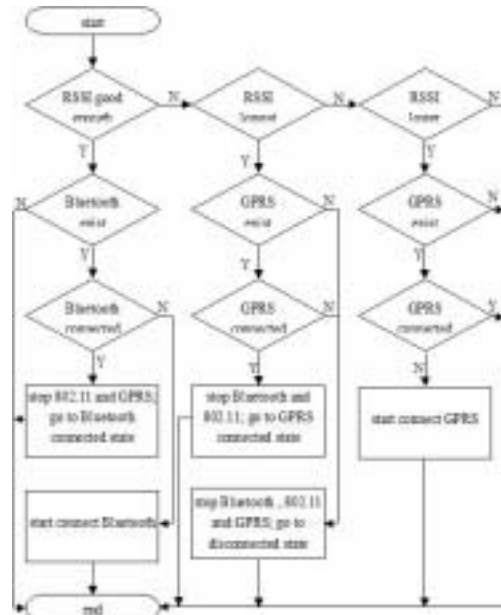
如果無法於連線品質低於RSSI lowest時完成切換的動作，或是因為無法預期的事件使得連線品質忽然降低至RSSI lowest以下，這時視為錯誤發生，重新啟動Bluetooth、802.11與GPRS跨階切換模組，模組狀態回到Disconnected。



圖十四：Bluetooth connected狀態流程圖

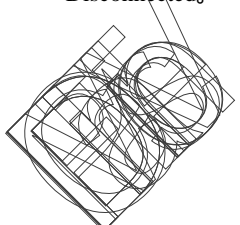
● 802.11 connected 狀態的處理

802.11 connected狀態的處理如圖十五的流程圖所示。為了能隨時得知連線的品質，要每隔十秒鐘執這個流程圖一次。這個流程圖主要是依照802.11連線的品質決定是否要切換至其它的連線。如果連線品質是高於RSSI good enough時，開始建立Bluetooth的連線，直到Bluetooth的連線建立完成中斷802.11連線，或連線品質低於RSSI good enough中止Bluetooth連線的建立。如果連線的品質是低於RSSI lower時，要開始建立GPRS連線，當連線的品質低於RSSI lowest時要完成切換的動作。



圖十五：802.11 connected狀態流程圖

和之前述過的一樣，如果無法於連線品質低於RSSI lowest時完成切換的動作，或是因為無法預期的事件使得連線品質忽然降低至RSSI lowest以下，這時視為錯誤發生，重新啟動Bluetooth、802.11與GPRS跨階切換模組，模組狀

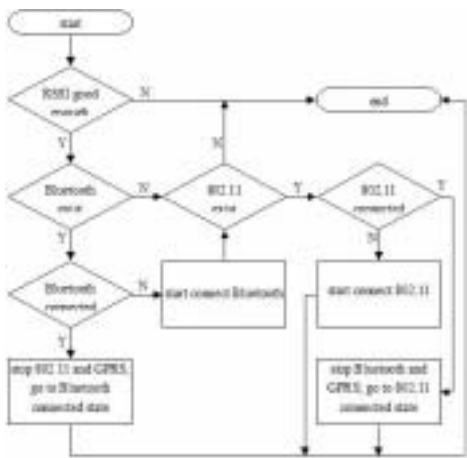


態回到Disconnected。

● GPRS connected狀態的處理

GPRS connected狀態的處理如圖十六的流程圖所示。為了能隨時得知連線的品質，要每隔十秒鐘執行這個流程圖一次。這個流程圖主要是依照GPRS連線的品質決定是否要切換至其它的連線。如果連線品質是高於RSSI good enough時，開始建立Bluetooth或802.11的連線，直到其中一個連線建立完成。如果兩個連線同時建立完成保留Bluetooth的連線。

再一次，如果無法於連線品質低於RSSI lowest時完成切換的動作，或是因為無法預期的事件使得連線品質忽然降低至RSSI lowest以下，這時視為錯誤發生，重新啟動Bluetooth、802.11與GPRS跨階切換模組，模組狀態回到Disconnected。



圖十六：GPRS connected狀態流程圖

## 五、附錄

目前我們完成了 (1) GPRS control DLL(2)Bluetooth, 802.11, and GPRS auto switch module，以下列出部份重要的程式碼。

(1)GPRS control DLL

```
class CGPRSCTL
{
public:
    GPRS_STATUS GetAdptName(CString* str);
    GPRS_STATUS QueryAdptStatus(CString str);
    GPRS_STATUS GprsTriggerConnect();
    GPRS_STATUS GprsTriggerDisconnect();
    .....
}
```

(2)Bluetooth, 802.11, and GPRS auto switch modul

```
class BWSwitch
{
public:
    void disconnected_process(void);
    void bluetooth_connected_process(void);
    void 802_11_connected_process(void);
    void gprs_connected_process(void);
    .....
}
```

## 六、文獻參考:

- [1] NDIS Miniport Drivers , URL: [http://msdn.microsoft.com/library/default.asp?url=/library/en-us/network/hh/network/102gen\\_24174df5-78af-48f3-8853-563c44c2e852.xml.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/network/hh/network/102gen_24174df5-78af-48f3-8853-563c44c2e852.xml.asp)
- [2] NDIS Object Identifiers , URL: [http://msdn.microsoft.com/library/default.asp?url=/library/en-us/network/hh/network/21oidovw\\_d55042e5-0b8a-4439-8ef2-be7331e98464.xml.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/network/hh/network/21oidovw_d55042e5-0b8a-4439-8ef2-be7331e98464.xml.asp)
- [3] CWinThread Class Members, URL: [http://msdn.microsoft.com/library/default.asp?url=/library/en-us/vcmfc98/html/\\_mfc\\_cwinthread\\_class\\_members.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/vcmfc98/html/_mfc_cwinthread_class_members.asp)
- [4] Remote Access Service, URL: [http://msdn.microsoft.com/library/default.asp?url=/library/en-us/ras/ras/ras\\_start\\_page.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/ras/ras/ras_start_page.asp)
- [5] 802.11 Wireless LAN Objects , URL: [http://msdn.microsoft.com/library/default.asp?url=/library/en-us/network/hh/network/217wirelessoid\\_bca9862e-feea-406f-b11d-ea01859bfd3.xml.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/network/hh/network/217wirelessoid_bca9862e-feea-406f-b11d-ea01859bfd3.xml.asp)

附件:

## 出國參加國際會議報告

本次赴舊金山參加『第 2004 年行動式系統、電子商務及代理人技術國際研討會議』國際會議 (The 2004 International workshop on Mobile Systems, E-commerce and Agent Technology (MSEAT ' 2004)), 也一併進行『第 2004 年分散式多媒體系統國際會議』國際會議(第 2004 International Conference on Distributed Multimedia Systems (DMS' 2004))。這個會議是在 San Francisco, California 舉行的, 從九月 8 號到九月 10 號共持續三天。討論主題很廣, 包括一、代理人(Agent)相關議題, 包含代理人結構(Agent architectures)、行動代理人 (Mobile Agent)、網路代理人 (Network agents)、多元代理人 (Multi-Agents) 等。二、行動式系統與電子商務 (Mobile Systems and E-Commerce) 議題, 包含行動電子商務 (Mobile commerce)、電子商務的代理人 (E-commerce interface agents)、電子商務的安全 (E-commerce security)、行動代理人的執行、推銷與處理程序 (Mobile agents for order processing, selling, transactions processing)、行動代理人的金融支付程序 (Mobile agents for payment, financial processing) 等。

在此次國際會議中本人也是其中一位議程委員, 分別參與了以下多個議題的討論, 主題分別有網路代理人、行動代理人、多元代理人、行動電子商務、電子商務的代理人、行動代理人的執行、推銷與處理程序, 因此, 每個時段均挑選以上相關之主題參與討論。在參與的過程中, 得知對行動代理人以及行動式系統與電子商務設計的技術與方法一再精進, 突破瓶頸深感科技研究的重要性。在本次會議中, 有多篇論文談及代理人的架構、多元代理人的最佳安排、行動代理人的執程序等, 而在另一方面, 對於代理人結合電子商務行動性設計的理論上, 也提供了行動 e 化代理主機的重要概念, 並針對代理主機資料的分配及服務, 給予一適當安排法則, 此外資料通訊的產生方法、資料通訊的效率提昇都提出一演算法來執行。

個人在會議進行期間亦發表論文。在發表完後, 共有三位研究學者提出有關無線網路應用在 Mobile Learning 的適用性、面臨困難及挑佔等問題, 我大約以五分鐘的時間為這些問題提出解釋與看法, 並均獲這三位外國學者的認同。經由實際的參與行動式代理人系統與電子商務的研討使我更積極規畫未來的研究的領域以符合潮流及實用性。

### 攜回資料名稱與及內容

- (1) 會議論文一本。
- (2) Call for paper 資料。