

行政院國家科學委員會專題研究計畫 成果報告

藍芽 MAC 排程機制之研究

計畫類別：個別型計畫

計畫編號：NSC92-2213-E-216-004-

執行期間：92年08月01日至93年07月31日

執行單位：中華大學資訊工程學系

計畫主持人：嚴力行

計畫參與人員：蔡銘洲、劉立群、曾友稜、楊平源、鄭仰民、林千惠

報告類型：精簡報告

處理方式：本計畫可公開查詢

中 華 民 國 93 年 11 月 4 日

## 摘要

Bluetooth 是以跳頻展頻為實體層傳輸技術的短距離無線網路，同時支援語音及資料傳輸。在 MAC 層頻寬分配方面是以 Polling 方式為基礎，由單一的 Master 裝置動態分配傳送資料封包的時槽給其他所有的 Slave 裝置。標準的做法中，Master 會以 Round-Robin 的方式對每一個 Slave 進行 Polling，而陸續也有許多學者提出各式的排程機制，分為預視與非預視兩大類。目前所見機制的效能評估，普遍建立在所有 Slave 送給 Master 的交通流量是一致的 (Uniform)，而且 Master 與 Slave 之間的交通流量是對稱的 (Symmetric) 兩項假設上。在我們觀察發現，當這兩項假設不成立時，現有機制的效能表現會有很大差異。在本計畫中，我們設計出一種新的非預視排程機制 RR-FIFO，並探討與現有 Bluetooth 排程機制 RR 與 ERR 於各種交通流量模式（一致、非一致、對稱、非對稱）下的效能（頻道使用率、封包延遲以及佇列長度）。我們的實驗結果發現，就封包延遲而言，ERR 除了在一致、對稱的 heavy load 時較 PRR 差外，其餘狀況均勝過 PRR。RR-FIFO 在對稱狀況下，效能與 ERR 在伯仲之間；在非對稱狀況下，如 Master 的封包到達率較高時，則效能較 ERR 好，反之則不然。就平均佇列長度而言，無論在任何情況下 RR-FIFO 始終有最小的 Master 佇列長度。

**關鍵詞：** Bluetooth, Wireless Network, Round-Robin, MAC, Scheduling.

## Abstract

Bluetooth is a short-range TDD (Time Division Duplex) wireless network that supports both circuit- and packet-oriented applications. A piconet is composed of a device configured as master and at most seven other devices acting as slaves. At Medium Access Control (MAC) layer, the master can select a slave to send a data packet and until then, the slave is not allowed to transmit. Round-Robin (RR) and Exhaustive Round-Robin (ERR) are two elementary MAC scheduling schemes that are both simple and efficient. As a result of this research project, we propose RR-FIFO, a simple MAC scheduling scheme that has the same advantages as RR and ERR. RR-FIFO acts as RR if the master's queue is empty and starts transmitting packets in first-come-first-serve order otherwise. The simulation results show that RR-FIFO's performance in terms of packet delay and queue length is comparable with those of RR and ERR.

**Key words:** Bluetooth, Wireless Network, Round-Robin, MAC, Scheduling.

## 報告內容

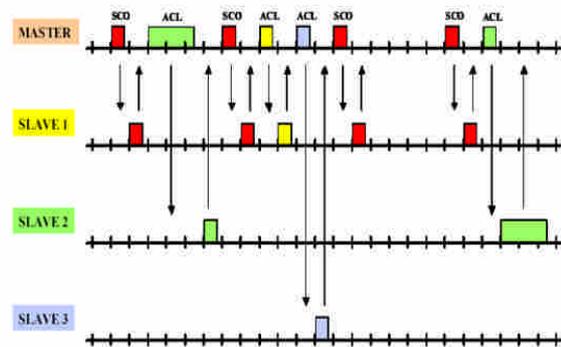
### 一.前言

近年來，由於網際網路服務的多樣化和行動計算需求的普及，無線通訊漸漸在現代科技生活中扮演重要的角色。現有無線通訊網路中，除了商業性服務的長距離無線電信系統（如 GSM、GPRS）以外，還有運作於 2.4 GHz ISM (The Industrial, Scientific, and Medical band; The ISM band) 頻段的短距離無線通訊網路。其中後者又可概分為無線區域網路 (Wireless Local Area Network; WLAN) 和無線個人網路 (Wireless Personal Area Networking; WPAN) 兩類 [2, 3]。其中 WLAN 系統以 IEEE 802.11b/a 為代表，實體層傳輸技術主要是直接序列展頻 (Direct-Sequence Spread Spectrum)，多應用在網際網路的無線擷取服務。另外 WPAN 則以 Bluetooth 為代表，以跳頻展頻 (Frequency Hopping Spread Spectrum) 為實體層傳輸技術 [4]，以取代近距離同質／異質性週邊的纜線連接為主要目的，亦可藉由跨越 Bluetooth 配備週邊，擷取遠距離的資料及語音服務。一群 Bluetooth 裝置亦可形成無線隨意網路 (Ad hoc networks)，彼此互相進行通訊 [5]。

Bluetooth 網路的最小運作單位稱為 Piconet，任意兩個 Bluetooth 裝置之間，在達成互相通訊之前，都必須藉由形成一個 Piconet 或是加入已形成 Piconet，才能進行通訊。一個 Piconet 由 2 個到 8 個不等的 Bluetooth 裝置（配備有 Bluetooth 傳輸功能的周邊設備，如手機、PDA、印表機等）組成。Piconet 的形成是由一個扮演 Master 角色的 Bluetooth 裝置發起，由它負責召集在其通訊距離內的其它 Bluetooth 裝置（最多可到 7 個）形成，而被召集的 Bluetooth 裝置稱為 Slave。Master 須使所有 Slave 的時脈與其同步，以便協調一致的跳頻序列。Master 也負責分配 Slave 對頻寬的使用權，並轉送 Slave 與 Slave 之間的資料封包。

一個 Piconet 內中所有的 Bluetooth 裝置共享約 1Mbps 的傳輸頻寬。頻寬分配的

細節如下所述。Master 依據其運作時脈，將傳輸時間切割成長度為 0.625 ms 的時槽 (Time Slots)，作為其頻寬分配與跳頻序列的基本單位。Master 以 TDD (Time Division Duplex) 機制，切割上行 (Slave 至 Master) 與下行 (Master 至 Slave) 資料流量。依據傳送資料型態的要求不同，在 Master 與 Slave 之間可建立兩種連結 (Link)，一種是 Synchronous Connection-Oriented (SCO) 連結，另一種則是 Asynchronous Connection-Less (ACL) 連結。



圖一：Bluetooth 時槽配置範例

SCO 連結是全雙工 (Full Duplex) 的連結，主要用於需要較高服務品質的語音資料傳送。為了保證頻寬，Master 於分配時槽時，須預留 SCO 連結所需之雙工時槽 (Duplex Slots)，也就是同時要預留雙向的時槽。其具體做法是每隔固定數目的時槽，就必須分配時槽給特定的 SCO 連結使用。在圖一中的時槽分配範例，每隔六個時槽即預留了兩個時槽給 Master 與 Slave 1 的 SCO 連結。

ACL 連結主要用於傳送資料封包，是由 Master 動態分配時槽的連結。Bluetooth 定義了數種 ACL 資料封包格式，分別需要佔用一、三、或五個時槽的傳送時間。Master 可在偶數時槽傳送資料封包給某個 Slave，而此 Slave 接收封包後，如有要傳送给 Master 的封包，可在緊接著的那個奇數時槽傳送（如圖一所示）。如該 Slave 沒有資料要傳送给 Master，也須在緊接著的時槽回傳一個沒有任何實際資料的 Null Packet，造成頻寬的浪費。此外，如果 Master 未傳送任何封包給某個 Slave，該

Slave 即無機會傳送資料給 Master。因此 Master 即使沒有資料要傳送，也可能會傳送沒有任何實際有用資料的 Polling Packet 給某個 Slave，以讓該 Slave 有機會傳送其封包。Polling Packet 的傳送也是造成頻寬浪費的另一個原因。

由上述描述可知，Bluetooth ACL 的 MAC 機制乃是以 Polling 為基礎。在此 Polling 系統中，每位顧客（即 Slave）都有一 Slave-to-Master Queue，存放欲送至 Master 的資料封包。服務提供者（Server。在此即 Master）也為每個 Slave 準備一個對應的 Master-to-Slave Queue，存放欲送至各個 Slave 的資料封包。Master 藉由拜訪每一個佇列的方式來服務排隊等候的顧客。而其拜訪佇列的順序規則，即 Polling 機制的規則集合 (Rules Set)，決定 Master 何時該從一個佇列的拜訪轉到其他的佇列 [6]。顯而易見地，此規則將會影響 Polling 機制的效能，如服務延遲 (Delay)、公平性 (Fairness) 等 [6, 7]。在 Bluetooth 的 Polling 機制中，尚有一項特別的效能考慮因素，即頻寬的有效利用率 (Utilization)。如前所述，Polling Packet 與 Null Packet 都沒有實際有用的資料。所以載送此兩類封包的時槽都被浪費掉了。

依照 Bluetooth 規格書 [1] 中所訂定最基本的作法，Master 會以 Round-Robin 的方式對每一個 Slave 進行 Polling。此種方式在網路負載較輕時，會產生許多 Polling Packet 和 Null Packet，使得整體的頻寬有效利用率相當差，也造成 Master 和 Slave 的不必要的電力耗損。而在網路負載高時，因為仍要固定對資料封量包較低 Slave 傳送 Polling Packet 及 Null Packet，會造成其他需要傳送的資料封包有較高的 Packet Delay [8]。

為了解決 Round-Robin Polling 機制效率不佳以及所造成的封包延遲的問題，已有許多學者陸續提出各式的排程 (Scheduling) 機制 [8]-[12]。這些做法有些宣稱有較高的產出量 [8,10,11]，有些則宣稱公平性較佳 [8,9,12]。不過我們觀察

到，這些所謂效能上的增進，普遍建立在一個不太實際的運作環境上，即假設所有 Slave 送給 Master 的交通流量是一致的 (Uniform)。直覺上，如果每個 Slave 送給 Master 的交通流量並非一致時，這些排程機制的效能表現應該會有差異。事實上，我們初步進行的模擬實驗結果也驗證了這樣的猜想。

## 二. 研究目的

本研究計畫的目的，在於探討現有 Bluetooth 排程機制於各種交通流量模式（一致、非一致、對稱、非對稱）下的效能，針對所得結果提出合理解釋，並設計出在各種交通流量模式下皆有不錯效能的非預視排程方法。此處所要探討的效能包含下列各項：

1. **封包延遲 (Packet Delay)**：為封包到達佇列至離開佇列之間所花的時間。需要量測的數值包含平均封包延遲與最大封包延遲，分別表示排程機制在此項目的平均值與最差值 [11]。我們認為這項效能應該是評估排程機制優劣的最重要考量因素。
2. **頻道利用率 (Channel Utilization)**：為實際傳輸資料時槽數目與最大可用時槽數目的比值。如前所述，排程機制額外送出的 Polling Packet 與 Null Packet 將會降低此項目的數值。若不計通道傳輸可能發生的錯誤狀況，此數據可直接類比資料產出量 (Throughput)。
3. **佇列長度 (Queue Length)**：佇列長度定義為佇列中封包的數量，我們測量佇列的平均長度 (Average Queue Length)。

## 三. 文獻探討

根據是否假設 Master 知道 Slave 的佇列狀

態，我們可將現有的 Bluetooth MAC 排程機制分為預視排程 (Preview-Scheduler) 與非預視排程 (Non-Preview-Scheduler) 兩大類。預視排程假設 Master 於排程時已預知 Slaves 佇列的目前狀態及即將傳送訊息的資訊 [13]；非預視排程則無此假設。

非預視排程符合 Bluetooth 系統在真實環境下的運作，但因 Master 僅能根據現有 Master-to-Slave 的狀態進行排程工作，無法考慮到 Slave-to-Master 的狀態，因此造成 Slave-to-Master 的封包延遲較為嚴重。預視排程需要利用額外的時槽來傳送 Signal 訊息封包，方能達成預視的效果，造成流量額外負擔 (Traffic Overhead)。不過預視排程因擁有全部的資訊，在產出量、封包延遲、以及公平性各方面的效能評估，通常也比非預視排程的來的較好，但預視排程只可視為理想化的排程方式。非預視的排程方式中，大致會呈現兩種情形：

第一種排程不考慮 Master 與 Slave 之間資料封包的傳送狀態。Master 依循一個既定的 Polling 模式，不論系統的交通流量高低，都依照此種模式來運作，如

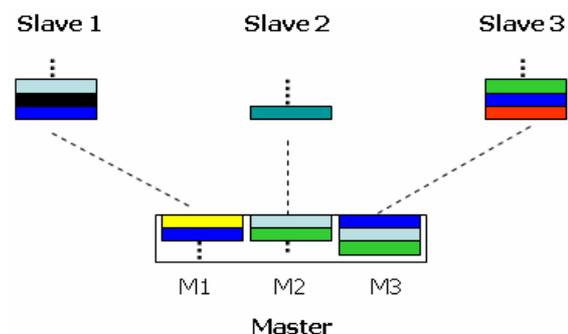
- ◆ PRR (Pure Round-Robin)[1][5-9]：Master 依循一個固定的順序，以資料封包或 Poll packet 輪流 polling 各個 Slave。被 polling 到的 Slave 一次只可以回傳一個資料封包。即使 Slave 沒有資料封包要傳送，仍然要回傳一個 Null Packet。
- ◆ ERR (Exhaustive Round-Robin)[5][6]：Master 仍依照一固定的順序輪流 Polling 各 Slave。與 PRR 不同的是 Master 可以持續 Polling 一個 Slave，而被 Polling 到的 Slave 也可以在回傳資料封包時，夾帶要求的訊息，要求 Master 繼續 Polling 此 Slave，直到 Master 與此 Slave 的傳送佇列清空為止。在這過程中即使 Master 已沒有資料封包要傳送給此 Slave，Slave 仍可以要求 Master 繼

續以傳送 Polling Packet 的方式 Polling 它。

第二種排程是 Master 根據自己目前所擁有的資訊進行排程。由於 Master 在進行排程時，只有 Master 單方面的資訊，使得 Master 與 Slave 的效能表現呈現獨立且有明顯的落差，例如：FIFO、LQFS[14]、MERR[14]，Master 的效能表現從 Load 高到 Load 低都相當好，而 Slave 在 Load 低的時候表現則相當差，到了 Load 高時，才漸漸轉好，但也可能存在少數幾個 Slave，遲遲無法被排程到而一直相當差的情況發生。造成 Slave 的排程效能大大降低。

#### 四. 研究方法

我們以模擬實驗的方式進行效能評估，其實驗架構如圖 2 所示。我們假設 Piconet 中 Slave 的個數為 7 個。每個 Slave 各自維護自己的佇列(queue)，存放即將傳送給 Master 的(上行)封包，稱為 Slave-to-Master queue(或 Slave queue)。而 Master 也為每個 Slave 準備一個對應的佇列，存放下行封包，稱為 Master-to-Slave queue(或 Master queue)。我們設定 Master 無法事先預知 Slave queue 的資訊，只能依據 Master 單方面的資訊進行排程。此設定符合實際的運作狀況。



圖二. Slave-to-Master & Master-to-Slave queue 封包類型 (Type) 的位元傳遞時間的一個時槽(slot)的 DH1 packet、Poll packet、或 Null packet。Slave-to-Master 與

Master-to-Slave 的 DH1 封包均以波氏程序 (Poisson process) 產生。其它實驗參數如下：

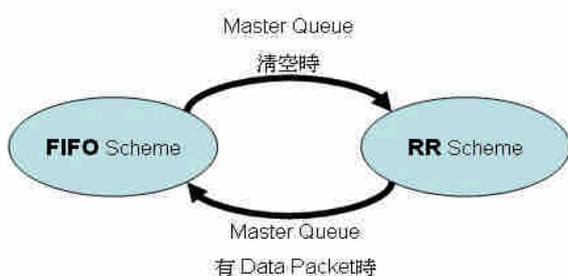
$\lambda$ : 在一個 Slot Pair (2 個時槽) 時間中的封包產生率。其值從 0.1 至 0.9 packet/slot-pair 變化。

$\mu$ : 在一個 Slot Pair (2 個時槽) 時間中的封包服務率。其值固定為 1 packet/slot-pair。

我們以 VC++ 來撰寫排程模擬程式。實驗中的 Traffic Load 為在一個 Piconet 中 Master 和 Slave 的封包到達率 (Packet Arrival Rate) 與 Mater 和 Slave 的封包服務率 (Packet Service Rate) 的比值。

## 五. 結果與討論

我們設計出的排程方法 RR-FIFO，是在 FIFO (First-In-First-Out) 中加入 RR 的機制。在 FIFO 機制中，Master 的 FIFO 封包清空便停止，但 Slave-to-Master 的封包可能仍然在產生，這段時間對 Slave 的封包和 Piconet 的整體運作都是一種浪費，如圖三所示，所以我們在 Master 閒置時進行 RR (Round-Robin) 的排程方式，直到 Master-to-Slave 的 FIFO Queue 又有封包進來。



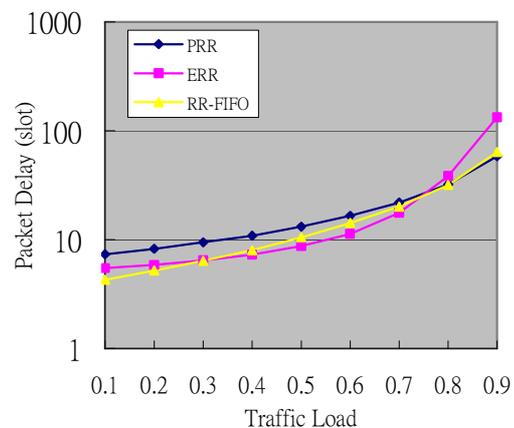
圖三. RR-FIFO 的狀態表示圖

當 Master Queue 清空而進入 RR 排程時，Master 會由上一次執行完 RR 的下一個 Slave 接著進行 Polling。如果每次 Master 閒置時即固定由第一個 Slave 開始執行 Polling，會造成排在前面順序的 Slave 被 Polling 頻率較高，而最後幾個 Slave 被 Polling 的頻率較低。這樣在某些 Traffic

Load 情況下，可能會導致排在後面的 Slave 產生 Starvation 的情形。

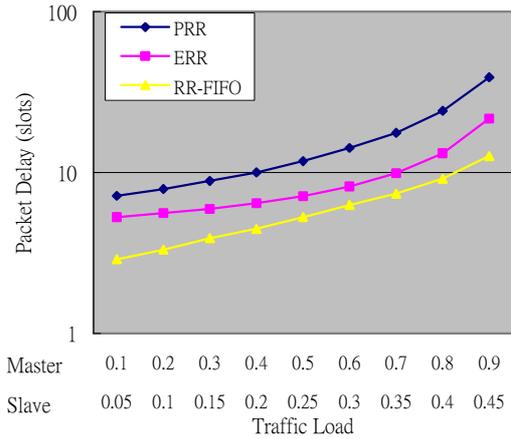
茲將我們的實驗結果概述如下。

圖四顯示 Master 與 Slave 封包產生率為對稱且一致下的封包延遲。可以看出 RR-FIFO 在 Traffic Load 小於 0.3 與大於 0.75 時表現較好。這是由於 RR-FIFO 在 Traffic Load 低時 FIFO 與 RR 的輪替，可以即時的將剛產生的 Master 與 Slave 的封包做傳送，而在 Traffic Load > 0.75 時，Master 執行 FIFO 的頻率較高，呈現 FIFO 在 Traffic Load 較高時對 Packet Delay 時，有較好表現的特性(由圖二)。而 ERR 整體表現都算不錯則，只有在 Traffic Load > 0.8 明顯較差。PRR 則是在 Traffic Load > 0.8，每一個 Slave 與 Master 都有較頻繁的封包傳送，相對也減少時槽的浪費，才有較好表現。RR-FIFO 在 Traffic Load 高時，仍然保持與 PRR 相當接近。

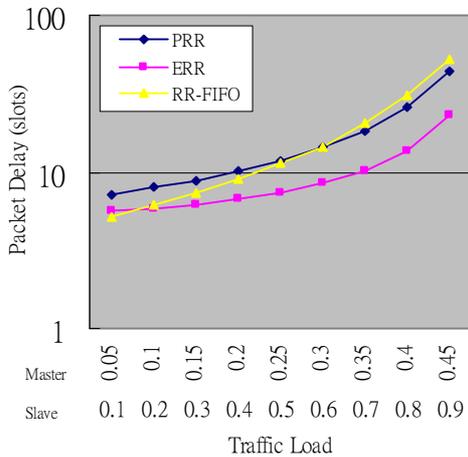


圖四. 對稱且一致型的 Packet Delay 情形

圖五顯示各個下行頻道封包產生率一致但上下行封包產生率非對稱 (Master : Slave = 2 : 1) 的封包延遲。RR-FIFO 整體表現最好。這是由於 RR-FIFO 優先處理 Master 封包，故在 Master Load 較高時，能使整體的封包延遲降到最低。而 ERR 也在一次清空 Master 和 Slave 才換下一個的特性下，明顯比 PRR 好。在模擬上 Slave Load > 0.45 時由於 Master 已超過負荷，所以不被考慮。



圖五. 非對稱、一致型的 Packet Delay 情形 (Master : Slave= 2 : 1)

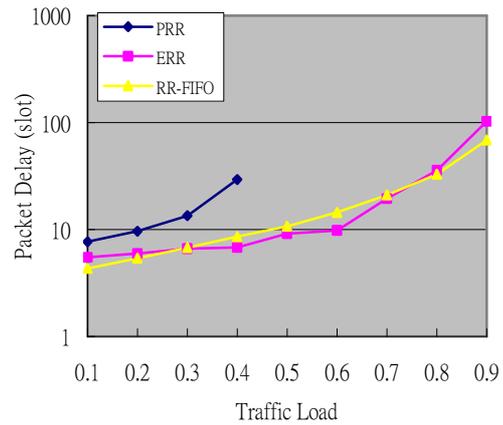


圖六. 非對稱、一致型的 Packet Delay 情形 (Master : Slave= 1 : 2)

圖六則是與圖五上下行封包產生率相反比例的非對稱型模式下，封包延遲的情況。此處 Slave 的 Traffic Load 較 Master 為高。RR-FIFO 在 Slave Load 小於 0.5 時的表現還算可以，但大於 0.5 以後就明顯較差。這是因為在 Slave Load 大於 0.5 時 Slave 的累積封包已越來越重，但 Master Load 仍然較輕。RR-FIFO 只能以不執行 FIFO 時的空閒時間來執行 RR，無法有效消耗 Slave 所累積的封包，造成整體封包延遲嚴重降低。

圖七顯示於對稱與非一致環境中的封包延遲。此處 ERR 與 RR-FIFO 大致與圖四對稱與一致環境下的結果相近。PRR 在 Traffic Load > 0.4 時，系統便呈現不穩定的狀態，第四對 Master 以及 Slave 的 Queue

有無限增長的情形產生。



圖七. 對稱、非一致型的 Packet Delay 情形 (MS1 : MS2 : MS3 : MS4 : MS5 : MS6 : MS7= 1:2:1:4:1:2:1)

#### 四. 結論與建議

在這個研究中，我們探討了現有藍芽的排程方式，預視排程與非預視排程，以及真實的藍芽運作模式的相關性。我們並設計出一個新的藍芽排程方式 RR-FIFO。在模擬實驗過程中，我們加入與以往不同的模擬模式，即非一致型與非對稱式的交通流量，試圖使模擬更接近真實網路中的各種不同交通流量情形。實驗模擬結果顯示，不同藍芽的排程方式，的確會對藍芽 Piconet 的運作效能造成顯著的影響。就封包延遲而言，ERR 除了在一致、對稱的 heavy load 時較 PRR 差外，其餘狀況均勝過 PRR。RR-FIFO 在對稱狀況下，效能與 ERR 在伯仲之間；在非對稱狀況下，如 Master 的封包到達率較高時，則效能較 ERR 好，反之則不然。就平均佇列長度而言，無論在任何情況下 RR-FIFO 始終有最小的 Master 佇列長度。

未來我們希望能在 Scatternet 環境下，對這些排程進行更廣泛而深入的效能探討，如時槽的使用率、延遲差異性、公平性等。我們希望這個研究成果能對藍芽無線網路的運作有直接的貢獻。

## 參考文獻

- [1] Bluetooth SIG, *Specification of the Bluetooth System - Version 1.1 B*, Feb. 2001.
- [2] J. Lansford, R. Nevo, A. Stephens, "Wi-Fi (802.11b) and Bluetooth: enabling coexistence," *IEEE Network*, 15(5): 20-27, Sep/Oct. 2001.
- [3] A. Kamerman, "Coexistence Between Bluetooth and IEEE 802.11 CCK Solutions to Avoid Mutual Interference", *IEEE 802.11-00/162*, July 2000.
- [4] J.C. Haartsen, "The Bluetooth radio system," *IEEE Personal Communications*, 7(1): 28-36, Feb. 2000.
- [5] G. V. Zaruba, S. Basagni, I. Chlamtac, "Bluetrees---Scatternet formation to enable Bluetooth-based ad hoc networks," in *Proc. IEEE International Conference on Communications (ICC 2001)*, pp. 273-277, 2001.
- [6] Z. Liu, P. Nain, D. Towsley, "On optimal polling policies," *Queueing Systems*, vol. 11, pp. 59-83, 1992.
- [7] O. Fabin, H. Levy, "Polling system optimization through dynamic routing policies," in *Proc. of IEEE INFOCOM '93*, vol. 1, pp. 194-200, 1993.
- [8] Jong Soo Oh, Kyun Hyon Tchah, Oh-Seok Kwon, Tae-Jin Lee, Yang-Ick Joo, Yongsuk Kim, "An efficient and QoS-aware scheduling policy for Bluetooth," in *Proc. IEEE 56th Vehicular Technology Conference (VTC 2002-Fall)*, vol. 4, pp. 2445-2450, 2002.
- [9] M. Kalia, D. Bansal, and R. Shorey, "Data scheduling and SAR for Bluetooth MAC," in *Proc. of IEEE 51st Vehicular Technology Conference (VTC 2000-Spring)*, vol. 2, pp. 716-720, Tokyo, Japan, May 2000.
- [10] M. Kalia, D. Bansal, and R. Shorey, "MAC scheduling and SAR policies for Bluetooth: A master driven TDD pico-cellular wireless system," in *Proc. Sixth IEEE International Workshop on Mobile Multimedia Communications (MOMUC'99)*, pp. 384-388, San Diego, USA, Nov. 1999.
- [11] A. Das, A. Ghose, A. Razdan, H. Saran and R. Shorey, "Enhancing performance of asynchronous data traffic over the Bluetooth wireless ad-hoc network," in *Proc. of IEEE INFOCOM'01*, pp.591-600, 2001.
- [12] S. Lu, V. Bharghavan and R. Srikant, "Fair scheduling in wireless packet networks," *ACM SIGCOMM'97*, August 1997.
- [13] A. Capone, M. Gerla and R. Kapoor, "Efficient polling schemes for Bluetooth picocells," in *Proc. of IEEE International Conference on Communications*, vol. 7, pp. 1990-1994, Helsinki, Finland, June. 2001.
- [14] Liron Har-Shai, Ronen Kofman, Gil Zussman and Adrian Segall, "Inter-piconet scheduling in Bluetooth scatternets," in *Proc. OPNETWORK 2002 Conference*, Aug. 2002.
- [15] 廖啟宏, 嚴力行, "藍芽 MAC 層排程機制效能之研究," 全國計算機會議 03, Dec. 2003, pp 900-907. (本計畫之論文發表)

## 計畫成果自評

本研究計畫內容與原計畫相符，不過並沒有達成完全達成原先預期目標。主要原因是現有 PRR 與 ERR 排程機制原先即有不錯效能表現，如要進一步改善，除非進行複雜精巧的設計。但 Bluetooth 為小型化無線通訊裝置，任何稍微複雜的排程機制在實務上均為不可行。我們不願我們的設計只有理論價值而毫無實用性，因此只能小幅度地改良原有的機制。所以最後成果在效能上的表現並非全面優於現有機制。

此研究成果具有一定之學術與應用價值，初步成果並已發表在全國計算機會議上（見參考文獻[15]）。但若要增加其學術價值，須對其效能表現進行理論上的分析，而非僅有實驗模擬結果。如有後續的理論分析結果，即適合在學術期刊上發表。在申請專利方面，目前評估並不適合。