

國立交通大學

資訊科學與工程研究所

碩士論文



Blueg: 一個新的建構在藍芽上類似 Blog 的 P2P
系統

Blueg: A Novel P2P Blog-like System via Bluetooth Devices

研究生：劉育嘉

指導教授：吳毅成 教授

中華民國 九十五年六月

Blueg: 一個新的建構在藍芽上類似 Blog 的 P2P 系統
Blueg: A Novel P2P Blog-like System via Bluetooth Devices

研究生：劉育嘉

Student : Yu-Chia Liu

指導教授：吳毅成

Advisor : Yi-Cheng Wu

國立交通大學

資訊科學與工程研究所



A Thesis

Submitted to Institute of Computer Science and Engineering
College of Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in
Computer Science

June 2006

Hsinchu, Taiwan, Republic of China

中華民國九十五年六月

Blueg：一個新的建構在藍芽上類似 Blog 的 P2P 系統

研究生：劉育嘉

指導教授：吳毅成

國立交通大學 資訊科學與工程研究所

摘要

隨著時代的潮流，網誌的蓬勃發展與網路相簿的流行，人們可以很容易地在網路上分享心得，記錄生活點滴，打開一條讓大家認識自我的管道。因此人們可以透過網路輕易地認識彼此，但是實際生活上可能仍與對方並不熟識，加上網路上認識的網友往往也給人相當虛幻的感覺。在日益忙碌的生活中，或許使得社交有所侷限，忽略了彼此，如果可以認識那些常常出現在周遭，彼此卻又不認識而心靈契合的人，將會是多麼美好的事。

由於行動裝置的普遍，本論文架構一系統於藍芽裝置上，透過藍芽裝置主動交換鄰近人們的個人資料，實際聚焦於身邊的每一個人，因此接收到的檔案皆具有區域性；本論文也探討如何以有效率的散播方式將個人資料傳播開來，這是一種新穎的人與人溝通方式。希冀此研究成果能夠對於行動運算領域有一定的貢獻。

Blueg: A Novel P2P Blog-like System via Bluetooth Devices

Student: Yu-Chia Liu

Advisor: Yi-Cheng Wu

Institute of Computer Science and Engineering

National Chiao Tung University

Abstract



Internet makes us know more people. Although we don't know most of them in the real life, we are familiar gradually with each other by chatting on the internet. As we live in the busy world, we perhaps had already lost our social contact. Some familiar strangers usually appear on our social network of physical spaces; however, it is a pity that we don't know each other. How wonderful if we know these familiar strangers further.

We use the mobile devices that via Bluetooth to develop a sharing system, which is called Blueg. It can automatically exchange and forward personal data with each other, furthermore, it also can combine the user's social life and the sphere of proximity. The introduce connection will be faster build by using Blueg system; therefore, we don't need to worry about missing a lady or gentleman who has appears around us.

誌謝

首先，要感謝的人是我的指導教授，吳毅成博士，在我寫這篇論文的期間，沒有他的辛勤指導與嚴格要求，我無法完成這篇水準之上的碩士論文研究，由於他的寶貴意見及指正，這篇論文才能夠順利的完成。

接下來要感謝的是博士班的徐健智、汪益賢、林秉宏等學長，這篇論文的許多設計和架構都是和他們一起不斷的討論、修正而完成的。同時，也要感謝實驗室的同學和所有的學弟妹們在我念研究所這段時間內所給予的幫助及鼓勵，也因為有他們的陪伴，讓我的生活充滿了歡樂。特別要感謝蔣承翰同學，多次的討論給予我不少啟發。

最後，要感謝的是我的父母、兄弟姐妹、女友雅閔，在我求學生涯上所給我的幫助及照顧，不管是否遇到挫折，不論成功或失敗，總是默默的支持我，給我最大的信心、動力。謹以此論文，獻給我最摯愛的家人。

目錄

摘要.....	i
誌謝.....	iii
目錄.....	iv
表目錄.....	vi
圖表目錄.....	vii
第一章、緒論.....	1
1.1 前言.....	1
1.2 研究動機與目的.....	2
1.3 論文章節說明.....	5
第二章、背景說明.....	6
2.1 智慧型手機概述.....	6
2.2 藍芽概述.....	7
2.3 相關系統.....	9
2.3.1 Kameleon.....	9
2.3.2 Miracle Lovers.....	10
2.3.3 Proxidating.....	10
2.3.4 Nokia Sensor.....	11
2.3.5 Social Serendipity.....	12
第三章、Blueg 系統架構設計.....	14
3.1 背景.....	14
3.1.1 藍芽協定堆疊.....	14
3.1.2 連結控制狀態.....	16
3.1.3 連線建立流程.....	17
3.1.3.1 藍芽裝置搜尋.....	18
3.1.3.2 服務搜尋.....	21
3.1.3.3 連結指定服務.....	23
3.1.4 建立連線時間.....	25

3.2 以嵌入訊息有效率建立連線機制	26
3.2.1 使用有限制的詢問存取碼.....	26
3.2.2 嵌入服務訊息.....	28
3.3 Store-and-Forward	30
3.3.1 Blueg 資料交換策略	31
3.3.2 SAF 散播演算法	32
3.4 相關議題	37
3.4.1 安全.....	37
第四章、實作與實驗數據分析.....	40
4.1 同步接收傳送	40
4.2 冗餘資料切截	42
4.3 實驗數據及特性分析	43
4.3.1 搜尋及建立連線時間與分析.....	43
4.3.2 散播演算法分析.....	46
第五章、結論與未來發展.....	50
參考文獻.....	51



表目錄

表 4-1、原藍芽搜尋及建立連線時間	44
表 4-2、改良後藍芽搜尋及建立連線時間	45
表 4-3、改良前後藍芽搜尋與建立連線時間分析	46
表 4-4、藍芽資料傳輸速率與資料傳輸總量	47
表 4-5、有無散播演算法散播效率比較 (N = 10).....	48
表 4-6、有無散播演算法散播效率比較 (N = 20).....	48
表 4-7、有無散播演算法節點活動狀態分析	49



圖表目錄

圖 1-1、BLUEG 生活圈	4
圖 1-2、STORE-AND-FORWARD 示意圖.....	4
圖 2-1、微網路.....	8
圖 2-2、分散網路.....	8
圖 2-3、KAMELEON 情境圖	9
圖 2-4、MIRACLE LOVERS 操作介面	10
圖 2-5、PROXIDATING 操作介面.....	11
圖 2-6、NOKIA SENSOR 操作介面.....	12
圖 2-7、SOCIAL SERENDIPITY 運作流程	12
圖 3-1、藍芽協定堆疊	15
圖 3-2、建立連線步驟狀態圖	18
圖 3-3、藍芽裝置搜尋（詢問、詢問掃描狀態）.....	19
圖 3-4、通用詢問存取碼與有限制的詢問存取碼回應示意圖	19
圖 3-5、藍芽裝置搜尋（詢問回應狀態）.....	20
圖 3-6、裝置的類別	20
圖 3-7、服務搜尋（呼叫、呼叫掃描狀態）.....	22
圖 3-8、服務搜尋（主裝置回應、從裝置回應狀態）.....	22
圖 3-9、服務搜尋（連線狀態）.....	23
圖 3-10、連結指定服務（呼叫、呼叫掃描狀態）.....	24
圖 3-11、連結指定服務（主裝置回應、從裝置回應狀態）.....	24
圖 3-12、連結指定服務（連線狀態）.....	25
圖 3-13、藍芽裝置建立連線所費時間	25
圖 3-14、改良後藍芽機置建立連線所費時間	26
圖 3-15、有限制的詢問存取碼回應示意圖	27
圖 3-16、手機於裝置類別的預設值	28
圖 3-17、嵌入服務訊息於裝置類別	29
圖 3-18、改良後快速建立連線示意圖	29
圖 3-19、STORE-AND-FORWARD 示意圖一（初始狀態）.....	30

圖 3-20、STORE-AND-FORWARD 示意圖二 (源頭與中介者相遇).....	30
圖 3-21、STORE-AND-FORWARD 示意圖二 (中介者與終點相遇).....	31
圖 3-22、重複傳送示意圖.....	31
圖 3-23、嵌入系統狀態於裝置類別.....	33
圖 3-24、散播演算法虛擬碼.....	34
圖 3-25、散播演算法示意圖一 (初始狀態).....	35
圖 3-26、散播演算法示意圖二 (A 中資料未同步, B 中資料同步)	35
圖 3-27、散播演算法示意圖二 (B、C 中資料皆同步).....	36
圖 3-28、無保護的資料竊改問題.....	37
圖 3-29、簽證示意圖.....	38
圖 3-30、接收者驗證憑證取到使用者公鑰.....	39
圖 3-31、接收者驗證結果.....	39
圖 4-1、ACTIVE OBJECT 運作示意圖.....	40
圖 4-2、同步收送模組的類別圖.....	41
圖 4-3、中途斷線切截資料處理.....	42
圖 4-4、資料傳輸封包格式.....	43



第一章、緒論

本章會說明在網路世界上與其他人交流、分享的行為，並提出本論文的動機與目的，如何將此行為聚焦於現實生的實體空間中，接著以情境切入整篇論文主軸。最後，會說明本論文的論文大綱。

1.1 前言

當網誌 (Blog) 與網路相簿[25]在近幾年發燒發熱的同時，行動網誌 (Moblog) [26]馬上被電信業者視為市場上的一塊大餅。現今所謂的 Moblog，是指透過行動電話利用 MMS 與 GPRS 服務，新增、刪除和修改個人 Blog，亦或是瀏覽其他人的 Blog，詳而觀之，可見 Moblog 本身跟 mobile 這個字詞並沒有直接的關係，因為在行動電話上還是如使用網際網路一般，將 Blog 存在遠端的伺服器供使用者操作。Moblog 的服務換言之只是提供另外一套網路系統與介面，再藉由行動電話達到隨處皆可存取 Blog 的功能。

使用手機瀏覽圖文並茂的 Moblog 看似方便，卻會嚴重地受到手機網路速度與穩定度的考驗，透過 GPRS 上網的 RTT (Run Trip Time) 往往可達 1~2 秒。除此之外，上網的價格也是相當重要的一環，無法平民化阻礙了 Moblog 的發展。

傳統的 Blog 必須透過第三管道來散播，如 BBS 或是交友網站等，流通的速度緩慢，如果自己不在第三管道上分享自己的 Blog，而他人要透過 Blog 入口網站看到自己的 Blog 也是微乎其微，在 Blog 與 Moblog 上認識的人們也給予人相當虛幻的感覺，造成很多人生活在不同的兩個世界中 -- 現實與網路。

近年來藍芽的普及使得各式各樣的應用得以發展[21][18]，以可攜式的行動裝置建立社群的觀念也漸漸受重視[14][24]，而行動裝置也已經具有足夠的能力提供點對點間的互動[2]；Blog 與 Moblog 的確造就了不少商機，也因此衍生出一些狀況，要如何建立一個可解決這些問題的系統，將成為本論文的目標。

1.2 研究動機與目的

首先，以下情境為現今網路上表現自我與分享的行為，述敘劇中主角布魯格所遇到的問題：

「唉，帳單又是三千多元，看來是該戒掉看 Moblog 的習慣了」布魯格暗自在心理嘀咕著。布魯格是個插畫家，靠的是接 case 過生活，每天又忙錢又不多，唯一讓他最快樂的時間是每天早上去星巴克用早餐時逛逛 Moblog。

『先生，您的 Latte 好了！』可愛的女服務生將咖啡遞了過來。為了想認識這個可愛的女服務生，布魯格每天早上都到星巴克報到。不過他實在太害羞了，一年來從沒跟她說過點餐以外的話。

「嗯…我這麼害羞大概一輩子找不到女朋友吧…」布魯格隨便找了個位置坐了下來，一邊輕啜著咖啡一邊玩弄著手機，又開始逛起了 Moblog。雖然貴，但是布魯格總是想要在上面多認識一點人，填補他小小的生活圈。不過網路終究是虛擬的，一年多來，布魯格靠著他雅痞風格的畫作，在網路上也闖出了一點名氣，但是他卻半個網友都沒見過。而手機上網的速度畢竟有限，布魯格通常利用下載網頁的時間瞄一瞄咖啡店裡的人——那些每天都見面卻又不認識的人，只能痴痴望著，沒有機會付出實際行動...

以上情境可以看到布魯格透過網路在虛擬世界中打出知名度，卻無緣認識現實生活中環繞在身旁的人，遇見心儀的女孩也缺乏跨出第一步的機會而躊躇不前。有鑒於以上各點問題，在此提出以藍芽裝置建立起 ad-hoc 網路的構想，並透過藍芽無線傳輸交換彼此的資料，達到交流的目的，稱之為 Blueg。有了 Blueg 系統之後我們預期布魯格的問題將會獲得改善：

大約一年半後，Blueg 流行起來。

「嗶～嗶～」手機再度傳來了接收到 Blueg 的聲音。在這涼爽的早晨，布魯格一如往常的在他用早餐的星巴克坐了下來，看了方才收到的相片，是一名穿著高中制服的長髮小女生（除了正推著門進來的短髮女孩穿著同樣制服外，似乎沒有其他候選人），Blueg 的鈴聲在這時響起卻是有點特殊。「大概是這小妹剛分散的同學吧！」布魯格心想。

一年多來，自從使用 Blueg 的系統後，布魯格認識了晨間會到星巴克來的人，這些人有上班族、有大學生，最令人玩味的不外乎是公車司機了，每次搭乘公車時，總是會從司機身上收到前班次乘客的 Blueg。原本布魯格最想認識的女店員，也是藉由 Blueg 才跟她熟識起來，雖然最後布魯格的告白失敗了，但兩個人還是變成了好朋友。也讓布魯格認識了許多朋友。

布魯格起身前往搭乘捷運的同時，「嗶～嗶～」手機不耐其煩的又響了起來，又有人加入這個大家族了...

上述情境中，布魯格透過 Blueg 系統與身邊的交際活絡起來，而 Blueg 取自於 Bluetooth 與 Blog，意指可以透過藍芽裝置[3]主動散播個人資料、相片等，相對的，收到 Blueg 的人皆近在咫尺，整個散播過程不需要用戶時時刻刻去操作，只要兩者相遇，收發流程會自動完成。可製造機會認識常常出現在身邊卻不認識的人 -- 熟悉的陌生人 (familiar stranger) [19]，因此本篇論文提供一個嶄新的人與人溝通模式，利用藍芽的短距通訊，將人們圈在一起，如圖 1-1 所示。

Blueg Social Network

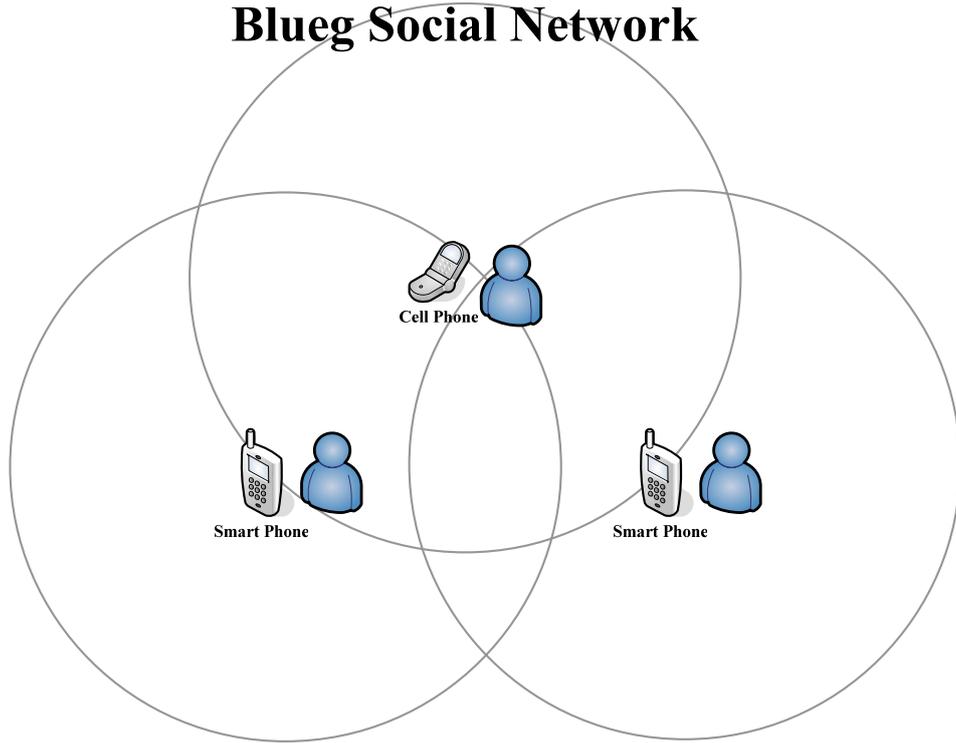


圖 1-1、Blueg 生活圈

拜科技的進步，行動裝置運算能力增強、儲存空間增大、內建數位相機都使得手機功能更加完備，本論文發展的系統除了連繫身邊人的通訊外，更進一步提出 Store-and-forward 的概念，如圖 1-2，使用者所散播出去的個人資料可透過接收者往外延伸，潛在性地增加能見度與擴散散播範圍。其間也必定會衍生一些問題，如有效率的散播演算法、竄改個人資料，以及其它相關技巧，將會在以下章節一一探討。

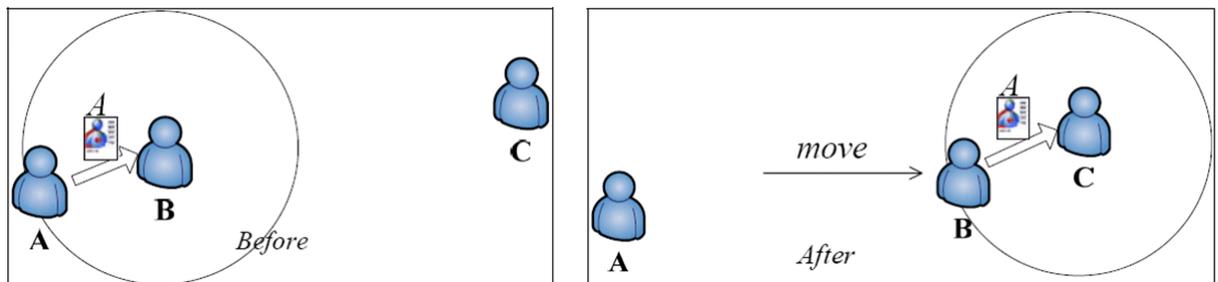


圖 1-2、Store-and-forward 示意圖

1.3 論文章節說明

本論文之章節架構如下：

1. 第二章簡介藍芽傳輸技術與智慧型手機，並比較相關系統與研究。
2. 第三章說明系統架構及設計。
3. 第四章說明實作的方法和在實作上碰到的問題，提出實驗數據並分析其結果。
4. 最後，第五章為結論和未來展望。



第二章、背景說明

本章介紹開發 Blueg 系統的相關背景。針對現有的智慧型手機平台與藍芽無線傳輸技術做一簡介，並且提出過去所發表的相關系統和研究，與之做比較。

2.1 智慧型手機概述

隨著行動裝置各方面上的進步，搭載百萬畫素相機、內建藍芽無線傳輸已經成為基本配備，手機也幾乎成為人生必需品。從智慧型手機的銷售量看出，其族群正在快速成長中，並為大眾所接受與使用。

目前最常見的智慧型手機作業系統有 Symbian [23]、Linux [15] 以及 Windows Mobile [16]。本論文採用 Symbian 作業系統作為 Blueg 系統的實作平台。Symbian 是由易利信、松下、摩托羅拉、諾基亞和 Psion 在 1998 共同出資組成的公司，在此之前 Symbian 是 Psion 的軟體部門。Symbian 成立後，陸續和一些廠商簽訂合約，這些公司包括日本電信業者 NTT DoCoMo、昇陽 Sun Microsystems 等。

Symbian 和其它作業系統的主要不同點在於，一般系統僅是以個人數位助理為出發考量來設計，而 Symbian 在開始設計時，就定位為無線資訊設備的即時作業系統。所以當 Linux 和 Windows Mobile 要發展無線通訊應用服務，而開始努力研究如何和無線手持式裝置整合時，Symbian 就已經和無線通訊作好緊密的結合。再加上 Symbian 在市場上行之有年，技術成熟度遠遠領先其它作業系統，其平台的穩定性也是大眾所稱道的。

2.2 藍芽概述

藍芽技術是一種無線連結技術，也是通行全球的無線連結標準，為一種低成本、低功率、短距的無線電 (radio) 技術，讓不同產品 (例如：印表機、行動電話、PDA、Notebook) 於短距離進行資料傳輸及溝通[5][6][7]。

藍芽裝置運作在 2.4MHz 頻帶上，這是在全球都可用、免費的工業、科技及醫學 (ISM - Industrial, Scientific and Medical) 頻帶範圍之內。主要是運用跳頻展頻技術 (FHSS - Frequency Hopping Spread Spectrum)，由這樣的方式，藍芽裝置使用了整個 ISM 可用的頻帶，並且在某一個頻道傳送資料失敗時，重送的資料會在另一個不同的頻道上傳送。

藍芽裝置可以運作在二種模式下：主裝置 (Master) 或從裝置 (Slave)。兩個模式的分界並沒有一定的規則，一個藍芽裝置可以同時扮演兩者，或是在兩者之間切換。一般而言發出連線要求的一端是 Master，而接收連線要求的一端是 Slave。在 Master 模式下可以設定跳頻序列。Slave 必須遵守 Master 決定的跳頻序列，經由 Master 跳頻序列將時間及頻率同步到與 Master 相同。

每一個藍芽裝置都有一個唯一的藍芽裝置位址 (BD_ADDR) 和時脈 (clock)。藍芽規格中的基頻 (Baseband) 部分描述如何由藍芽裝置位址及時脈計算出跳頻序列的演算法。除此之外，本論文也採用唯一的藍芽裝置位址做為辨別不同行動裝置的依據。

一群 Slave 和一個共同的 Master 一起運作即所謂的微網路 (piconet)，如圖 2-1，所有在同一微網路上的裝置有著和 Master 一樣的跳頻序列及時脈。規格中限定在微網路中的 Slave 數目最多為七個，若需要較大含括範圍則可透過連結微網路來達到，參考圖 2-2。

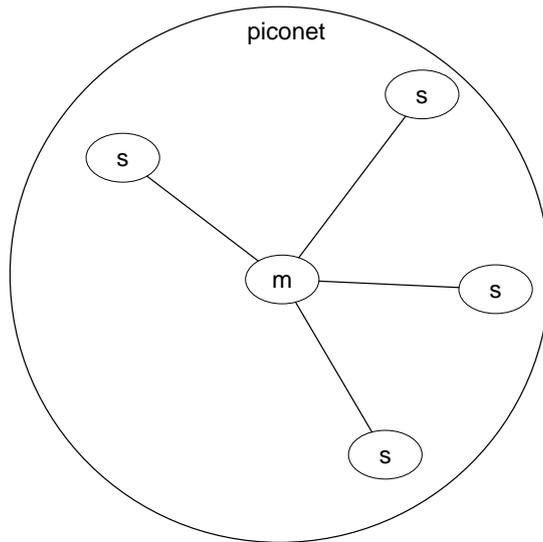


圖 2-1、微網路

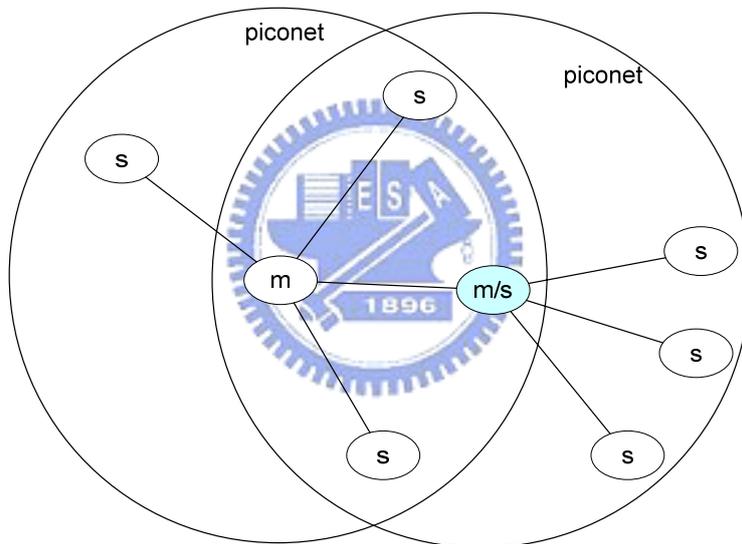


圖 2-2、分散網路

藍芽並不像任何有線的網路，任何與之通訊的裝置並不需要實體上用纜線連接起來；事實上可能也不知道正在和那些裝置溝通以及那些裝置具有那一些功能。為了達到這個目的，藍芽提供詢問 (Inquiry) 與呼叫 (Paging) 的機制和服務發現協定 (SDP - Service Discovery Protocol)，用來詢問鄰近的藍芽裝置是否提供指定的服務。

2.3 相關系統

本節將介紹過去曾經發表的相關系統與研究，分別說明其使用條件和用途，再分析其優缺點。

2.3.1 Kameleon

Kameleon [12] 架設藍芽發射器於商店、廣告看板、人潮擁擠的地方，透過內建藍芽的智慧型手機，使用者可以從各據點的藍芽發射器獲得商品折價訊息，電影預告片的連結，並透過 GPRS 下載電影預告片觀看，Kameleon 為一應用藍芽的例子，但由於其主要是靠著定點的藍芽發射器做廣告行銷[1]，與本論文方向有所差異，在此提出 Kameleon 僅因它是一個透過藍芽散播資訊的一個應用，參考圖 2-3。



圖 2-3、Kameleon 情境圖

2.3.2 Miracle Lovers

KDDI 在 2001 年 6 月上市配備藍芽功能的 cdmaOne 手機 C413S。Miracle Lovers [13] 即為配備在 C413S 上的軟體。Miracle Lovers 除了在等朋友赴約時用來尋找朋友位置的朋友雷達以及預測與對方姻緣的姻緣占卜外，還有從參加遊戲中的人中尋找夢中情人的尋找命中人等功能。其使用上主要的缺失為使用者在尋找他人必須親自操作，該軟體並不會有智慧地自行主動尋找感興趣的對象，也因此造成使用上的不方便，參考圖 2-4。



圖 2-4、Miracle Lovers 操作介面

2.3.3 Proxidating

利用藍芽交友的軟體 Proxidating [20]，只要把手機的藍芽啟

動，並事前輸入自己的個人資料、性別與要尋找對象的性別，當合適對象出現在周遭時，手機便會響起，並顯示對方的長相，如果中意對方即可上前交談見面，參考圖 2-5。



圖 2-5、Proxidating 操作介面

Proxidating 在使用上仍然需要使用者親自操作，每一次手機響起，通知使用者附近有合適的對象後，必須再一次啟動搜尋功能才能進行下一次配對，而對象的個人資料也僅是一時的，於下次搜尋前會被清除。缺乏資料管理，也無有效率的散播機制。

2.3.4 Nokia Sensor

Nokia Sensor [17]運用了藍芽無線技術，當手機「感應」到其它配備 Nokia Sensor 的手機進入接收範圍，即可以閱讀他們的資料及向他們發送訊息，如圖 2-6。比起 Proxidating，Nokia Sensor 進一步對接收到的資料做管理，好友名單上的人鄰近時也會主動發出提醒提示聲，但 Nokia Sensor 所接收與發送的資料也僅限於曾經接觸過的人們，並無有效率的散播機制。



圖 2-6、Nokia Sensor 操作介面

2.3.5 Social Serendipity

Social Serendipity [8] 利用藍芽手機的藍芽裝置位址與使用者簡介資料庫，來認識身旁的人。此系統一旦啟動即進入背景執行，主動搜尋附近是否有其它藍芽手機，根據所搜尋到的藍芽裝置位址透過 GPRS 連結到系統架設的使用者簡介資料庫，下載所搜尋到的使用者簡介。因為藍芽裝置位址是唯一的，因此在使用者簡介資料庫中，每一個使用者對應到一個藍芽裝置位址，也是基於這個緣故，才能經由藍芽裝置位址下載指定使用者的個人資料，參考圖 2-7。

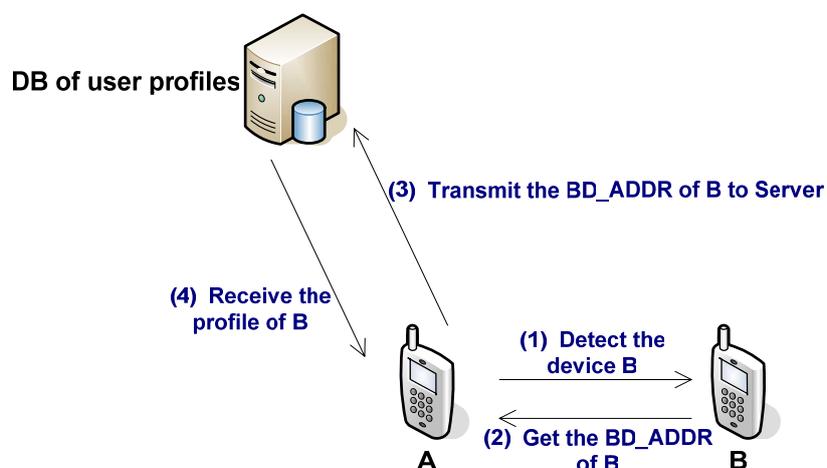


圖 2-7、Social Serendipity 運作流程

此系統利用藍芽搜尋功能，並連上資料庫下載個人簡介來認識身旁直接接觸的人，或利用在會議中，只要到每個人身旁走動即可做自我介紹。

但考量 GPRS 花費與社交範圍，卻有其不足的地方，因為僅靠著搜尋藍芽裝置位址，所能得到的資訊僅限於直接接觸的人，個人資料無法往外延伸。而每下載一筆個人簡介都需額外負擔 GPRS 的費用，也造成使用上的不便。



第三章、Blueg 系統架構設計

本章將介紹 Blueg 系統架構設計，對藍芽溝通協定與建立連線的前置作業與細節做詳細說明，接下來介紹本論文利用溝通協定中的那些技巧使得建立連線比原本藍芽一般建立連線還快速，緊接著是 Store-and-forward 的機制，最後探討由 Store-and-forward 而衍生的相關議題。

3.1 背景

在章節 2.2 中有提到藍芽裝置透過無線連結可以形成微網路，而主裝置 (Master) 與從裝置 (Slave) 要如何建立連線即為本章節之重點，並針對建立藍芽連線所需的背景知識做一介紹。

3.1.1 藍芽協定堆疊

藍芽規格定義多項協定，有些屬於核心協定，建議必須實作，部分屬於選擇性協定，實作部分的多寡使得藍芽定堆疊有些許不同，一般而言，大部分藍芽裝置除了核心協定外，也都支援電纜替代協定 (RFCOMM)。

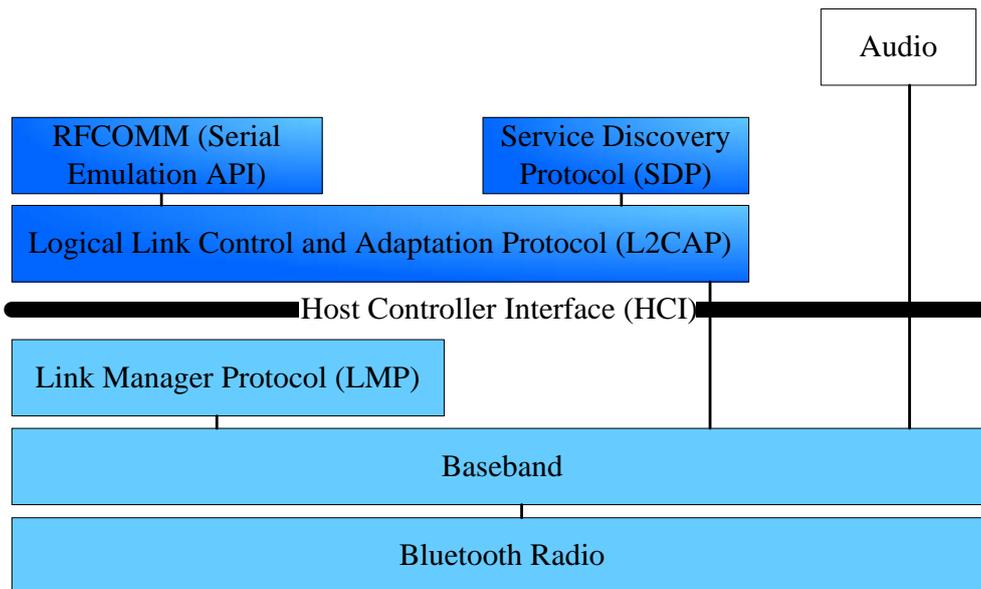


圖 3-1、藍芽協定堆疊

上圖 3-1 為絕大部分藍芽裝置所支援，接下來將會針對圖中核心協議以及電纜替代協定，分別介紹如下[7]：

- **基頻協定 (Baseband)**
基頻和連結控制層確保微網路內各藍芽設備之間由射頻構成的實體連結。基頻是負責頻道的編碼與解碼和低層的時間控制，並管理連線單一資料封包傳送的部分。
- **連結管理協定 (LMP - Link Manager Protocol)**
連結管理協定負責藍芽各設備間連結的建立與設置。它控制微網路的管理（建立與破壞連結）、連結設定、安全功能。
- **邏輯連結控制和調節協定 (L2CAP - Logical Link Control and Adaptation Protocol)**
邏輯連結控制和調節協定為上層提供服務。向上層提供連接導向和無連接的資料服務時，採用了多工傳輸、分段和重組技術。允許高層協定以 64k 位元組收發資料分組。

- 服務搜尋協定 (SDP - Service Discovery Protocol)
服務項目在藍芽技術框架中有著極為重要的作用，使用 SDP 可以查詢到設備資訊和服務類型，從而在藍芽設備間建立相對應的連接。在一般的區域網路下，一但發現有一條連線連接到印表機，它將會有幾個月甚至幾年都是留在同一地方而不會改變。藍芽的設計則是允許移動到一個區域並且不需先行安裝設定就可找到一台印表機。SDP 想要找到一個能列印的裝置，或者瀏覽在此區域中其它藍芽服務裝置的範圍，找到可提供列印的裝置。
- 電纜替代協定 (RFCOMM - Serial Emulation API)
RFCOMM 是基於 ETSI 07.10 標準的序列模擬協定。RS-232 序列埠有九條用來傳送資料和信號的電路線，RFCOMM 可以模擬序列電纜線設定和 RS-232 序列埠的狀態。



3.1.2 連結控制狀態

在任何時候，一個藍芽裝置會處於許多不同狀態中的其中一個狀態，在詳細解說之前，先對主要的狀態作定義[7]：

- 詢問 (Inquiry)
詢問是一個程序，當一個裝置要搜尋所在地區中所有可用的藍芽裝置時，所使用的程序。在詢問程序的這段期間，被詢問的裝置會以 FHS (Frequency Hop Synchronization) 封包來回應給詢問者，用來記錄被要求建立連線的必要資訊。
- 詢問掃描 (Inquiry Scan)
詢問掃描是另一半的詢問程序。他們對使用通用詢問存取碼 (GIAC - General Inquiry Access Code) 或有限制的詢問存

取碼 (LIAC - Limited Inquiry Access Code) 封包監聽較長的一段時間。當他們接收到有效的詢問訊息就會進入詢問回應 (Inquiry Response) 狀態，以 FHS 封包回應。

- 呼叫 (Page)

為了建立連線，變成 Master 的裝置被應用程式命令來進行呼叫程序。Master 首先進入呼叫狀態，在這個狀態中，會直接傳送呼叫訊息到他所打算要成為 Slave 的裝置。Slave 確認呼叫訊息，而 Master 進入主裝置回應 (Master Response) 狀態，並且以其 FHS 封包回應。

- 呼叫掃描 (Page Scan)

一旦在呼叫掃描狀態的裝置成功地接收呼叫封包後，它就會進入從裝置回應 (Slave Response) 狀態，在此狀態中會確認該封包，並等待 FHS。

- 連線 (Connection)

在進入連線狀態之前，Slave 會與 Master 的時脈做同步，移到 Master 的跳頻頻率和時序序列。Master 傳送 POLL 封包來確認連結已經成功建立起來，然後 Slave 必須以任意型態的封包來回應，通常是 NULL 封包。

3.1.3 連線建立流程

使用藍芽裝置連結到指定服務正常分為三個步驟，如圖 3-2：

1. 搜尋藍芽裝置。
2. 建立連線透過服務搜尋協定 (SDP) 確定該藍芽裝置提供指定服務。

3. 經由 SDP 所提供的訊息，建立連線連結到指定服務。

本篇論文觀察到建立連線的過程中，可透過嵌入訊息的方式，有效節省連線時間，在此之前，需要了解每個步驟流程與其封包格式，分述如下。

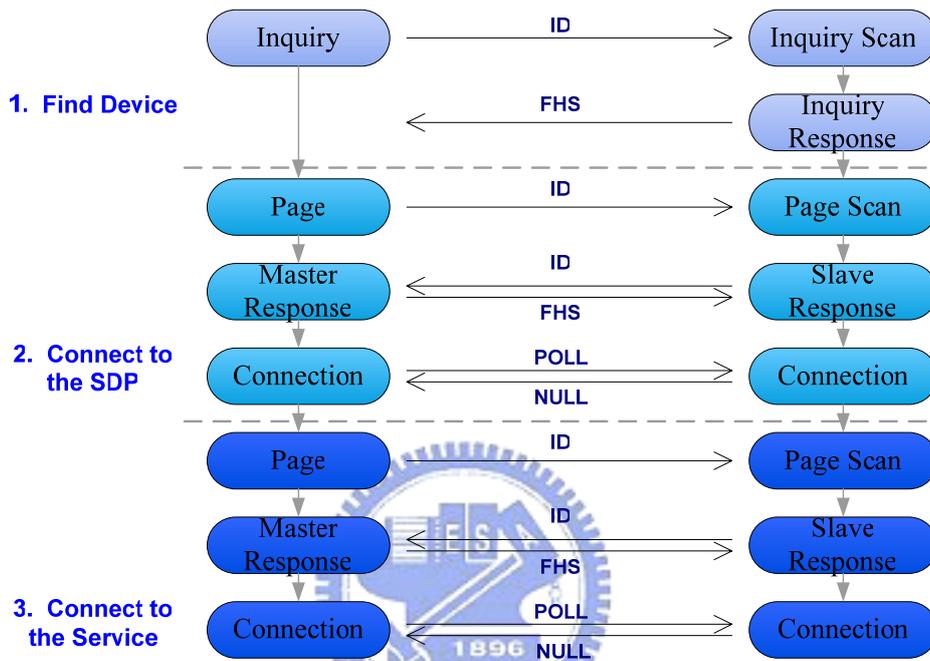


圖 3-2、建立連線步驟狀態圖

3.1.3.1 藍芽裝置搜尋

當一個藍芽裝置要搜尋所在地區中所有可用的藍芽裝置時，會進行詢問 (Inquiry) 狀態，不斷地在各頻率中傳送 ID 封包，ID 封包中含有詢問存取碼 (IAC - Inquiry Access Code)，透過詢問存取碼來指定期望何種可發現模式 (discoverable mode) 裝置會回應，如圖 3-3。

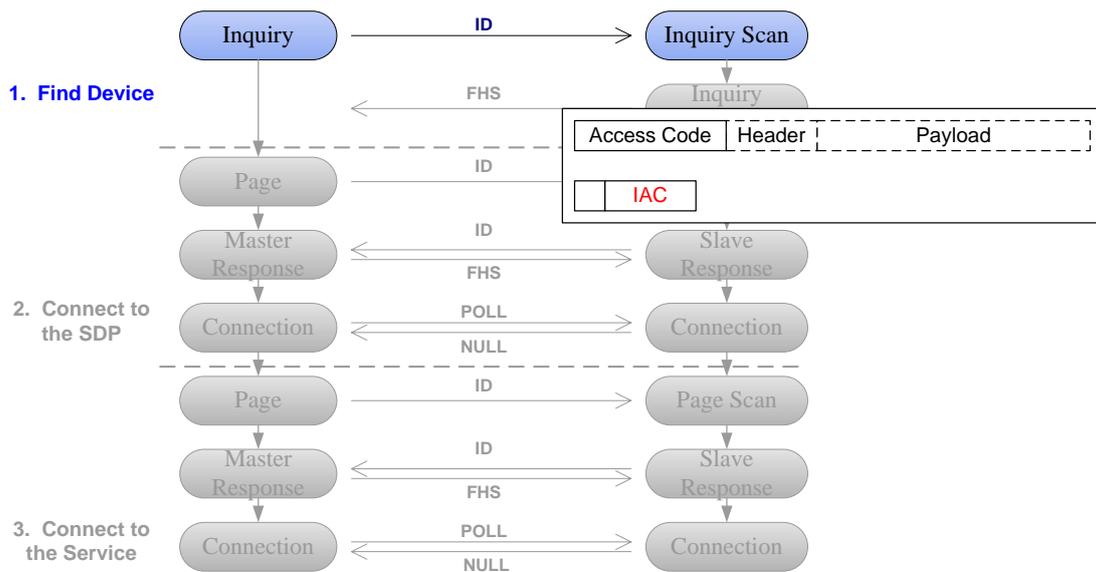


圖 3-3、藍芽裝置搜尋（詢問、詢問掃描狀態）

目前已制定的藍芽可發現模式有兩種，通用可發現模式（General discoverable mode）與有限制的可發現模式（Limited discoverable mode）。

詢問存取碼也有兩種，通用詢問存取碼（GIAC）與有限制的詢問存取碼（LIAC），在 GIAC 的詢問程序下，通用與有限制可發現模式的藍芽裝置都會回應，在 LIAC 的詢問程序下，僅有限制的可發現模式藍芽裝置會回應，如圖 3-4 [6]。

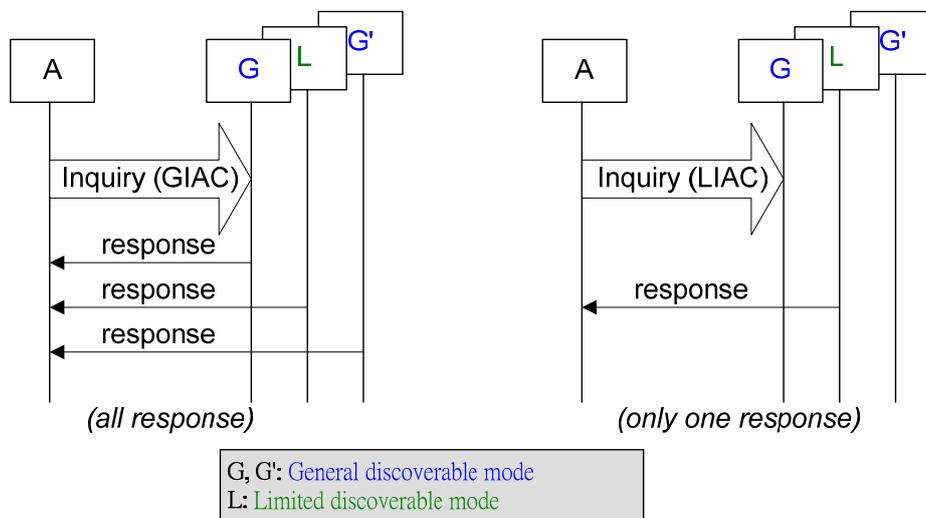


圖 3-4、通用詢問存取碼與有限制的詢問存取碼回應示意圖

被搜尋的藍芽裝置收到 ID 封包後，隨即在一個亂數時間延遲後回應 FHS 封包，此 FHS 封包含有此藍芽裝置位址 (BD_ADDR) 與裝置的類別 (CoD - Class of Device) [4]，如圖 3-5。

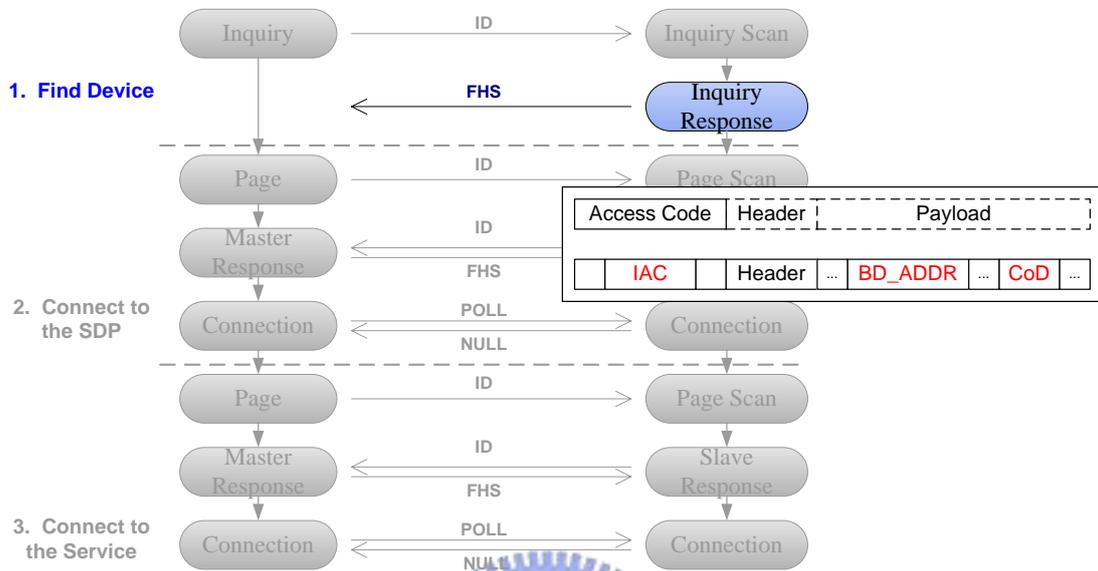


圖 3-5、藍芽裝置搜尋 (詢問回應狀態)

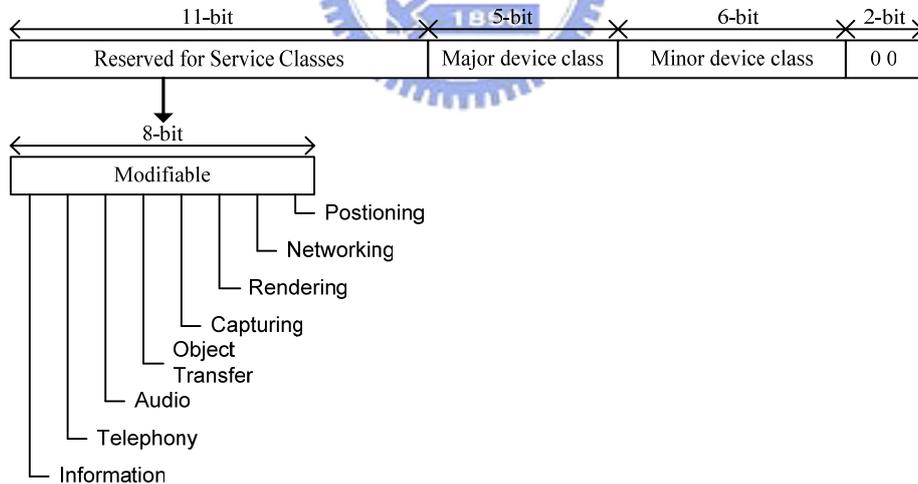


圖 3-6、裝置的類別

藍芽裝置位址為唯一的藍芽硬體位址，裝置的類別包含了該藍芽裝置等級，這個類別由主要 (Major) 及次要 (Minor) 部份所組成。以手機上的藍芽裝置為例，主要裝置類別 (Major device class) 告

知已找到一個電話，次要裝置類別 (Minor device class) 告知這是一個手機。除此之外，也提供較高層的服務類別 (Reserved for Service Classes)，告知此藍芽具有何服務的能力，實際上，服務類別欄位是可被應用程式修改的，參考上圖 3-6。

在詢問的過程中，所在地區中的藍芽裝置會一一回應，而回應的 FHS 封包具有建立連線所需的必要資訊，將會在建立連線時使用到。

3.1.3.2 服務搜尋

取得可連線的藍芽裝置之後，首先需進行服務搜尋，查詢該藍芽裝置是否提供所需服務，而服務搜尋這個動作將會在兩藍芽裝置間建立連線，並透過 SDP 協定經由應用程式進行服務屬性比對。

建立連線時會進行呼叫 (Page) 程序，直接傳送呼叫訊息的 ID 封包給被呼叫的藍芽裝置，封包中含有裝置存取碼 (DAC - Device Access Code)，裝置存取碼是由被呼叫的藍芽裝置位址 (BD_ADDR) 計算而來，如圖 3-7。初始連線的裝置定義為主裝置 (Master)，而此圖中左邊的狀態可視為主裝置的狀態，圖中右邊狀態可視為從裝置 (Slave) 的狀態。

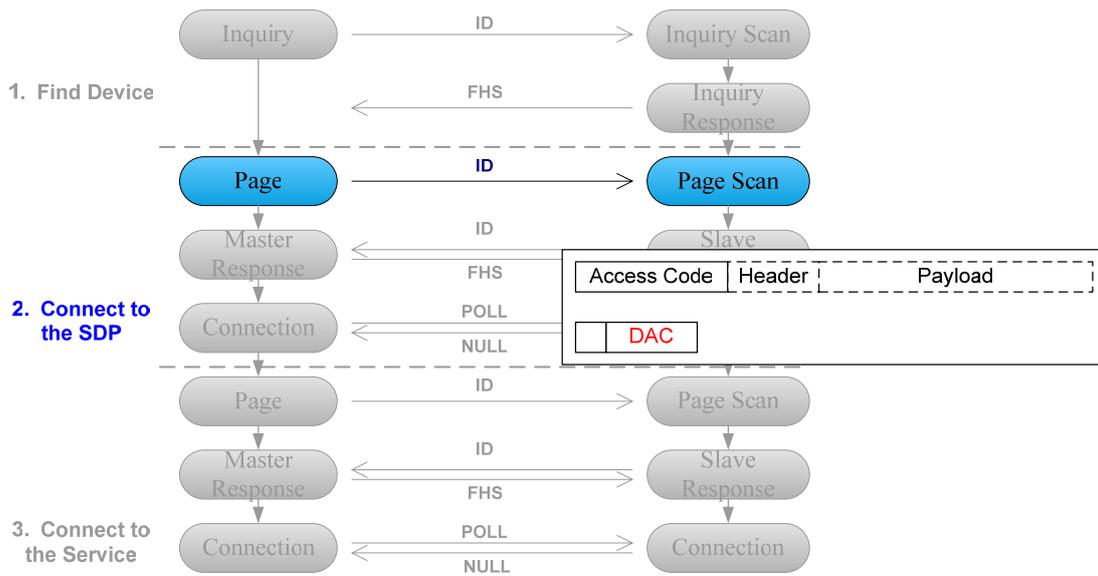


圖 3-7、服務搜尋（呼叫、呼叫掃描狀態）

Slave 接收到呼叫訊息的 ID 封包後，隨即進入從裝置回應（Slave Response）狀態，並回應相同的 ID 封包。在 Master 收到 Slave 的回應後，即傳送 FHS 封包給 Slave，FHS 封包中具有 Master 的位址、時脈訊息，Slave 透過此訊息來與 Master 同步，如圖 3-8。

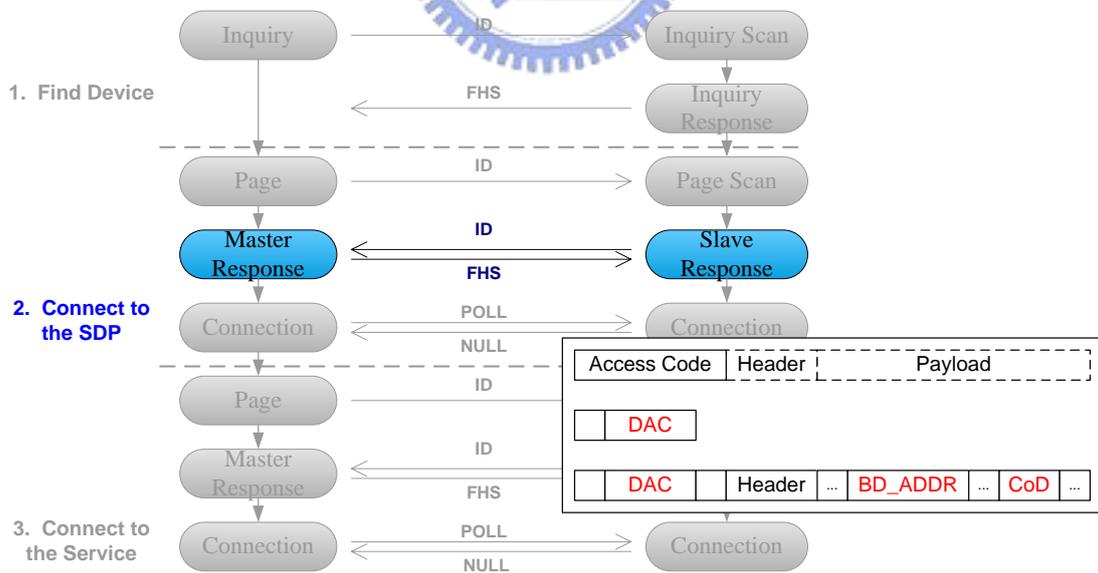


圖 3-8、服務搜尋（主裝置回應、從裝置回應狀態）

隨後兩藍芽裝置即進入連線狀態，主裝置傳送 POLL 封包來確認連結已經成功建立起來，從裝置以 NULL 封包回應，如圖 3-9。

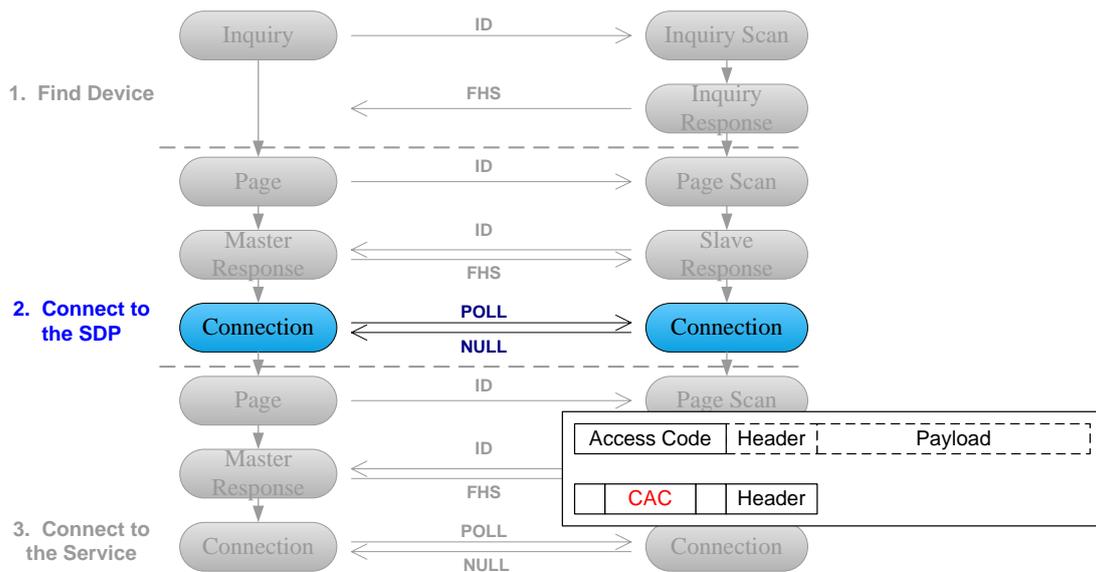


圖 3-9、服務搜尋（連線狀態）

接下來應用程式即可利用 SDP 來查詢對方是否具有指定的服務，若比對服務屬性結果確定有該服務，則根據查詢的結果再進行一次連線連結到指定服務。



3.1.3.3 連結指定服務

根據服務搜尋的結果，可得知指定服務是否存在與相關連結訊息，透過這些資訊再建立一次連線，即可使用該服務，其連線過程與前一小節節連線過程一樣，僅以圖 3-10、圖 3-11、圖 3-12 示意，不再多加詳述。

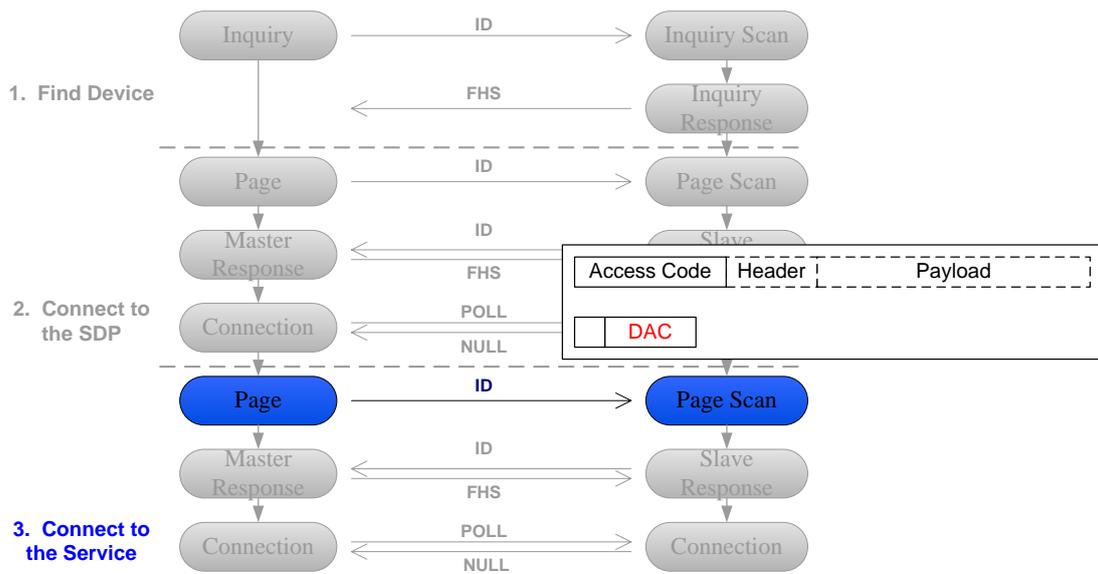


圖 3-10、連結指定服務（呼叫、呼叫掃描狀態）

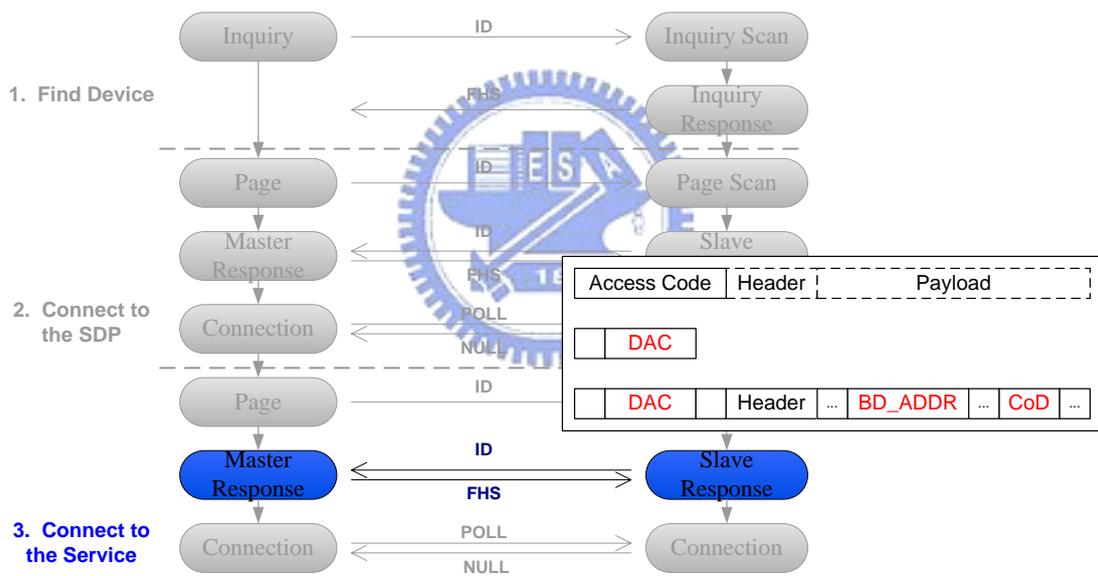


圖 3-11、連結指定服務（主裝置回應、從裝置回應狀態）

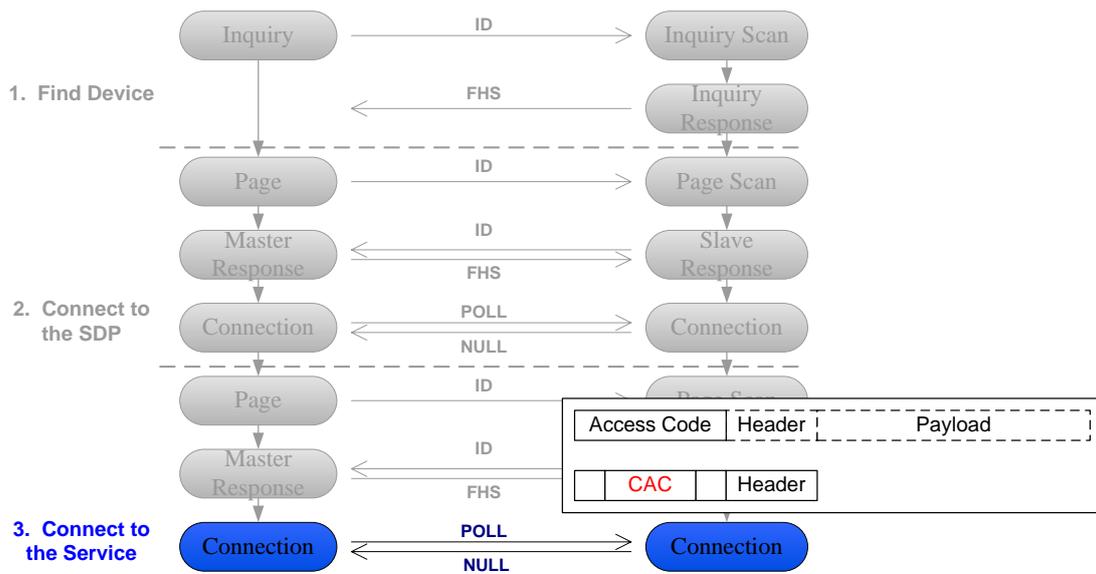


圖 3-12、連結指定服務（連線狀態）

3.1.4 建立連線時間



從搜尋藍芽裝置到最後建立連線到服務的時間分三階段，如圖 3-13，其中藍芽搜尋用去 2.3 秒，服務搜尋用去 1.9 秒，為瓶頸所在。

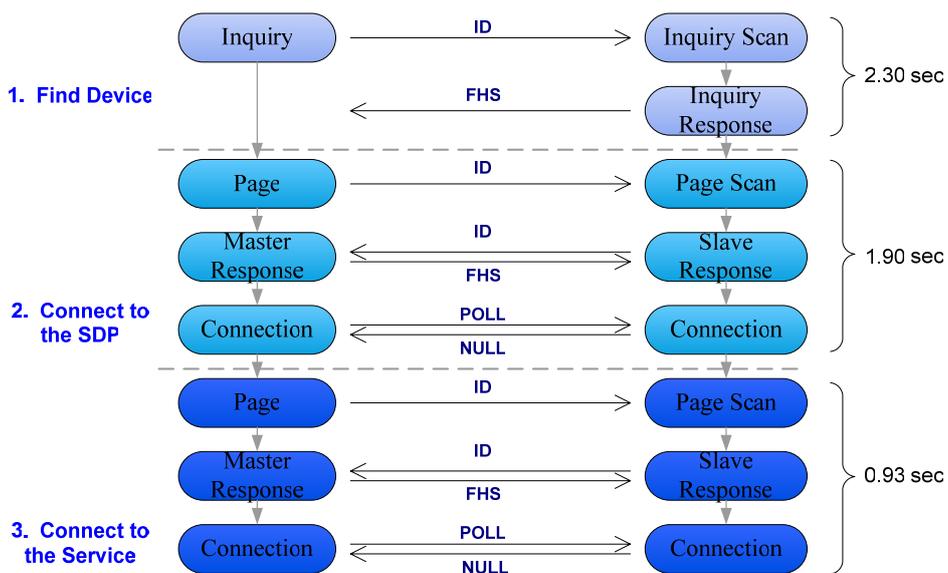


圖 3-13、藍芽裝置建立連線所費時間

為了改善此情形，我們透過嵌入訊息的方式，使得藍芽裝置搜尋時間縮短，更可將第二個步驟服務搜尋省略掉，如圖 3-14，細節將在下一章節詳細敘述。

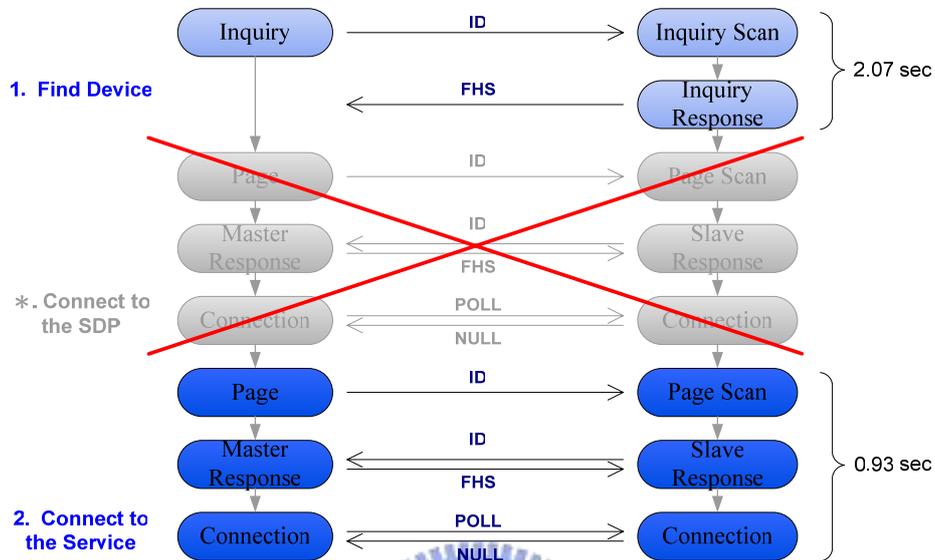


圖 3-14、改良後藍芽機置建立連線所費時間

3.2 以嵌入訊息有效率建立連線機制

從建立連線流程的觀察中，藍芽搜尋與服務搜尋佔去絕大部分的時間，為了節省建立連線所花費的時間，我們提出使用有限制的詢問存取碼 (LIAC) 縮短藍芽回應時間，並利用嵌入服務訊息於 FHS 回應封包，判斷是否省略服務搜尋步驟。

3.2.1 使用有限制的詢問存取碼

在 3.1.3.1 章節中提及，被搜尋的藍芽裝置接收到 ID 封包後，

會在一個亂數時間延遲後回應 FHS 封包，在藍芽規格中定義，使用 LIAC 允許有更短的亂數時間延遲，也就是說使用 LIAC 之後，接收到 LIAC 的藍芽裝置將在更短的時間內回應，並提供建立連線的必要資訊。

通用詢問存取碼 (GIAC) 與有限制的詢問存取碼 (LIAC) 主要差別在於回應裝置的模式，GIAC 允許所有可發現模式的藍芽裝置回應，LIAC 只允許有限制的可發現模式 (Limited discoverable mode) 的藍芽裝置回應，參考圖 3-15。一個藍芽裝置在設定成有限制可發現模式並不影響其正常運作，因為 LIAC 為 GIAC 的子集合，此模式下仍會對 GIAC 詢問有所回應[6]。

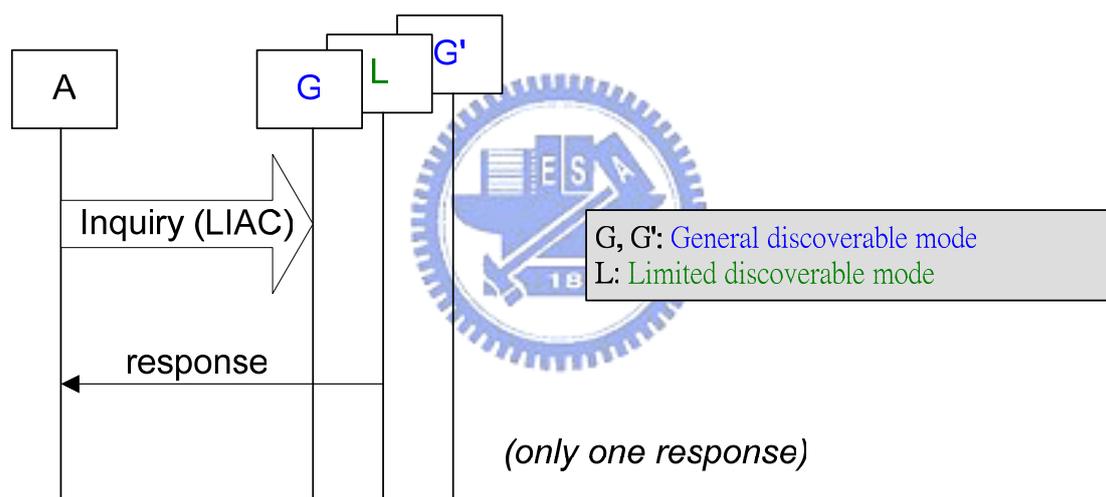


圖 3-15、有限制的詢問存取碼回應示意圖

因為絕大部分的藍芽裝置都設定在通用可發現模式 (General discoverable mode)，所以使用 LIAC 除了在藍芽裝置搜尋上會加快之外，也潛在性的過濾了其它設備 (例如：藍芽耳機、藍芽滑鼠)，減少其它一般性藍芽設備所造成的干擾。

3.2.2 嵌入服務訊息

在前述章節提到藍芽裝置被搜尋到時，會回應 FHS 封包，封包中含有建立連線的訊息，其中裝置的類別 (CoD) 訊息可用來辨別此藍芽裝置的主要類別 (例如：電腦、電話、週邊設備) 與次要類別 (例如：桌上型電腦、筆記型電腦)，參考圖 3-16，其中服務類別左方 8 個位元可經由應用程式修改。

圖中可觀察到主要類別的值等於 0x02，代表此裝置為電話，次要類別的值等於 0x03，代表此裝置為智慧型手機，而服務類別的左邊第一個位元定義為網頁伺服器等服務，智慧型手機一般不提供此服務，此值為 0，類別的定義請參照[4]。

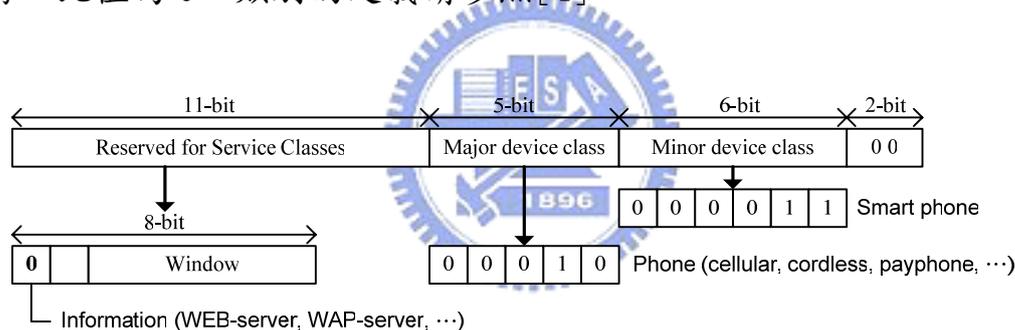


圖 3-16、手機於裝置類別的預設值

基於以上觀察，將 CoD[23] 的值設定為 1 代表此手機具有 Blueg 系統服務，如圖 3-17。因此是否提供指定服務的訊息即嵌入屬於建立連線第一個步驟的 FHS 封包內，而後當每一個藍芽裝置回應時，即可利用 CoD 的主要類別、次要類別與 CoD[23] 來辨別該裝置是否具有 Blueg 服務，並可省略第二個步驟的服務搜尋時間。

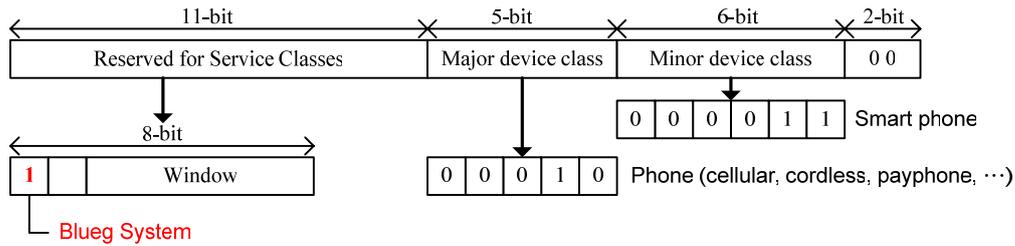


圖 3-17、嵌入服務訊息於裝置類別

下圖 3-18 為使用 LIAC 並修改 CoD 來達到快速連線的流程。首先透過 LIAC 縮短藍芽回應時間，並於回應的 FHS 封包內嵌入 Blueg 服務訊息，再經由簡單的比對確認 Blueg 服務的存在，直接跳過服務搜尋步驟，建立連線至 Blueg 服務。原先服務搜尋所花費的時間所佔比例約五分之二，如今將之省略，顯然有助於縮短藍芽裝置間連線的建立。

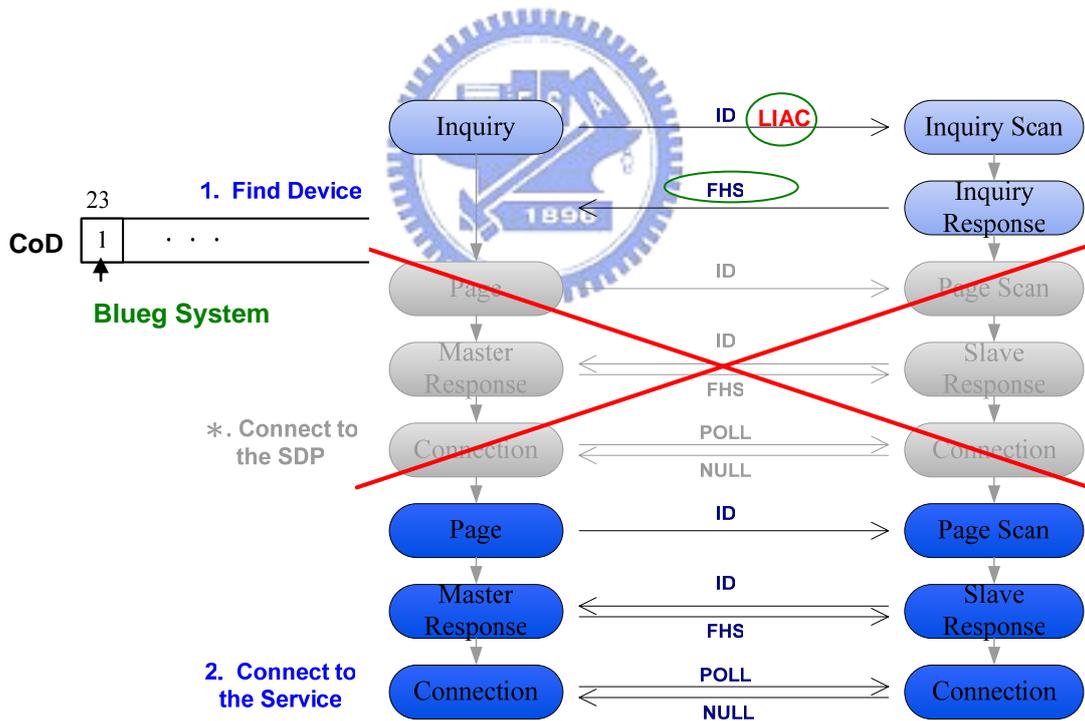


圖 3-18、改良後快速建立連線示意圖

3.3 Store-and-Forward

提出 Store-and-forward 的概念，是為了讓個人資料的散播不僅只於限制直接接觸的對象，而能進一步散播出去，以同心圓的方式一層層擴散到周遭的每一個人（例如：公車站牌等待的人群、同一間咖啡廳的人們、捷運車廂的群體），圖 3-19、圖 3-20、圖 3-21 為 Store-and-forward 的展示圖。小華遇見小王時，將自己的個人資料傳送給小王，而後小王遇見小明，將小王以及剛從小華那邊所接收到的個人資料一併傳送給小明，最後，即使小明沒有與小華直接接觸，也擁有小華的個人資料。



圖 3-19、Store-and-forward 示意圖一（初始狀態）

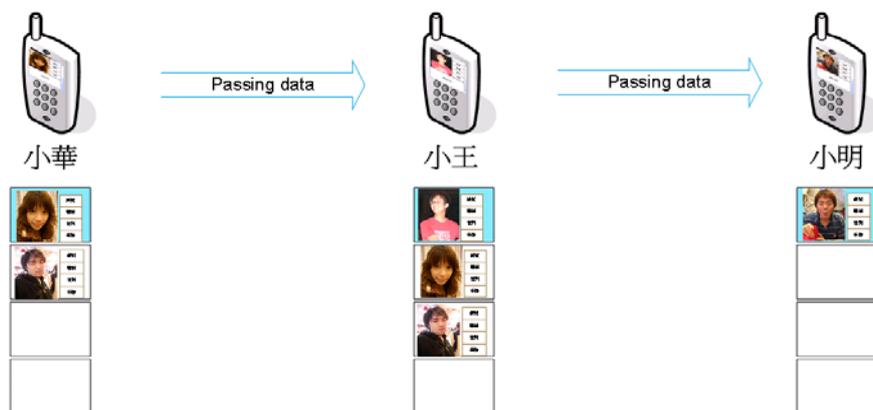


圖 3-20、Store-and-forward 示意圖二（源頭與中介者相遇）

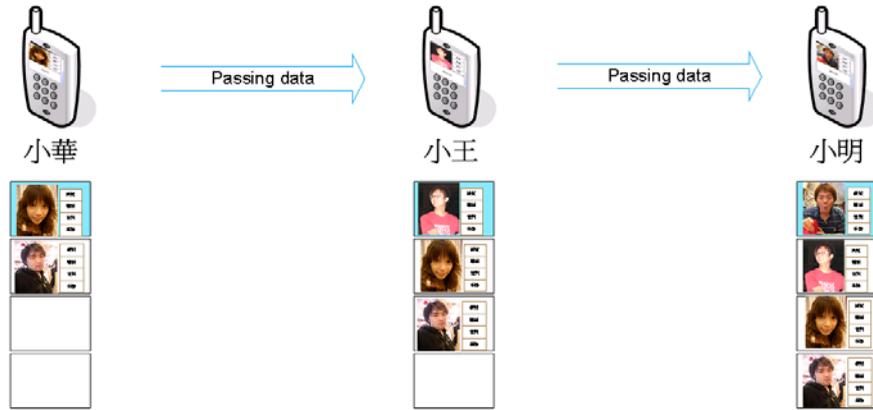


圖 3-21、Store-and-forward 示意圖二（中介者與終點相遇）

若僅有 Store-and-forward 必定會產生一些問題，如圖 3-22，小華遇上小明，將傳送重複的個人資料。有鑑於此，因此本論文嵌入額外的 6-bit 的狀態改變值於 FHS 封包內，有效率的控制個人資料的散播。

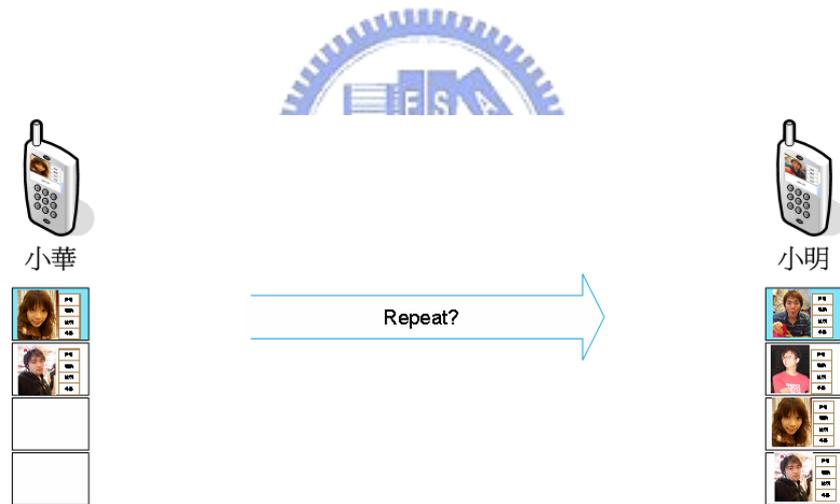


圖 3-22、重複傳送示意圖

3.3.1 Blueg 資料交換策略

在 Blueg 系統中，資料交換行為發生在兩裝置建立連線之後，為了確定交換的資訊是對方所沒有的，資料交換分二步驟進行，首先交

換彼此各自擁有那些使用者的基本資料，再根據此訊息判斷與對方交換何資訊。

在有效的利用藍芽頻寬的條件之下，並不希望重複傳送對方已經擁有的使用者資料，因此在真正交換資料之前，以簡短的訊息告知對方自己目前所擁有的以及期望收到的使用者資料，顯得更加重要。在設計上的必要訊息為使用者帳號與時間戳記，以使用者帳號做為雜湊表 (Hash Table) 的索引鍵，判斷是否擁有該份檔案；若曾經接收過同一使用者的資料，以時間戳記為數值，辨別資料是否更新。

有了這些訊息之後，Blueg 系統便可以進一步比對，以使用者帳號當索引鍵，判斷本地端擁有那幾份使用者資料是對方所沒有的，若雙方皆具有同一使用者帳號的資料，則利用時間戳記辨別資料是否有更新，經由這些訊息的判斷，再交換彼此缺少的使用者資料。

透過分析，傳送一組帳號與時間戳記需要 20 位元組，傳送 100 份僅需 2 KB，而任何一筆多傳送的重複使用者資料必大於 2 KB；除了可以有效率的使用藍芽頻寬外，亦可以減少重複資料氾濫問題。

3.3.2 SAF 散播演算法

3.2 章節中提到用 CoD[23] 來得知該藍芽裝置是否具有 Blueg 系統，在此用同一手法，將 CoD[21-16] 指定為 6-bit 的狀態改變值 (Window)，如圖 3-23。在 Blueg 運作當中，第一次使用時，預設初始值等於 1，只有在接收到新的使用者資料，或者改變本地端的使用者資料，其值才會累計加一 (非本地端更新累計運算必需在兩端點正常結束連線條件下才會更新)。如 3.2 章節中所述，CoD 訊息夾帶在藍芽 FHS 封包中，在藍芽搜尋階段中會回應 FHS 封包得以獲取建立連線的必要資訊，使用狀態改變值可以在尚未建立連線前，獲知對方的

狀態是否與本地端所記錄的相符，若不相符，則可能是對方在距離上一次連線的期間內，曾經接收到新的使用者資料，亦或者是對方修改了自己的基本資料。因此透過嵌入 6-bit 的狀態改變值於 FHS 封包內，可以在第一時間把關，在對方本身狀態沒有變更的情形下，不盲目建立連線，也不交換彼此擁有何使用者資料的訊息，可避免不必要的額外負擔。

本論文提出的散播演算法分連線前與連線後兩部分，在藍芽搜尋階段可透過狀態改變值避免建立無謂的連線，若有必要建立連線，則可進一步利用狀態改變值，圖 3-24 為散播演算法的虛擬碼。建立連線前的藍芽搜尋階段會取得嵌入 FHS 封包內的狀態改變值，透過狀態改變值來判斷對方有無新進使用者資料，再決定是否建立連線。若狀態改變值與本地端所記錄的不一樣，則會建立連線，開始交換彼此所缺少的使用者資料，3.3.1 章節提及資料交換的流程，在使用者資料交換前先行交換彼此所擁有的使用者資料為何，再根據這些事先傳送的訊息來決定那些使用者資料可能是對方所缺少的；在交換完使用者資料後，會得知相異使用者資料是否已全部接收，如果原本儲存在遠端的使用者資料已被全部接收，在此稱為已與遠端同步。並且會在正常結束連線後，記錄適當的遠端狀態改變值，以便下次欲與同一裝置建立連線前的依據，也會更新自己的狀態改變值。

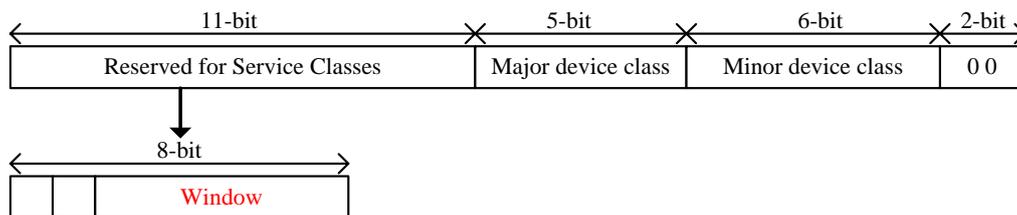


圖 3-23、嵌入系統狀態於裝置類別

```

Forwarding Algorithm
oldWindow = GetWindow(newBT.iBD_ADDR);
if oldWindow == newBT.iWindow
    return;

Connect(newBT.iBD_ADDR);

Exchange(bSyn);
if (bSyn)
    Update(newBT, newBT.iWindow + 1);
else
    Update(newBT, newBT.iWindow - 1);
CountWindow();

```

圖 3-24、散播演算法虛擬碼

交換使用者資料後，每一個藍芽裝置會維護一個狀態表對應於所接收到的使用者資料狀態值。下一範例為散播演算法一例，圖 3-25 為 A 手機與 B 手機相遇前的各項設定。圖 3-26 為 A 手機與 B 手機交換個人資料後更新的值，圖中 A 手機沒有收到 B 手機中所有的個人資料，因此 A 手機的更新表中 Device B 所對應的值是 B 手機的舊值減一 (12)；而 B 手機有收到 A 手機中所有的個人資料，因此 B 手機的更新表中 Device A 所對應的值是 A 手機的新值 (24)。圖 3-27 為 B 手機與 C 手機交換個人資料結果，圖中顯示兩者皆已同步，所以更新表中的值皆為新值。

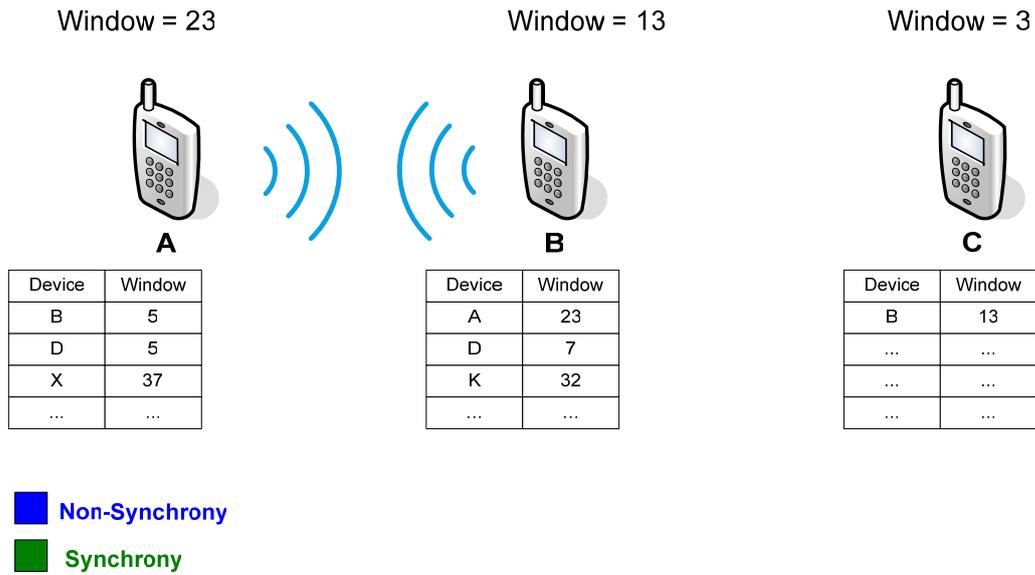


圖 3-25、散播演算法示意圖一（初始狀態）

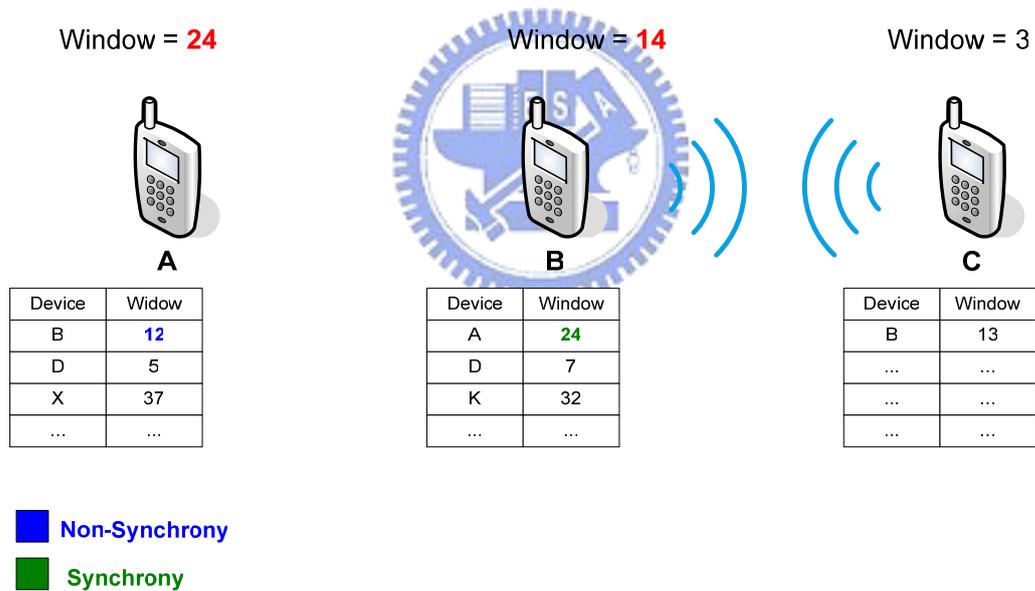
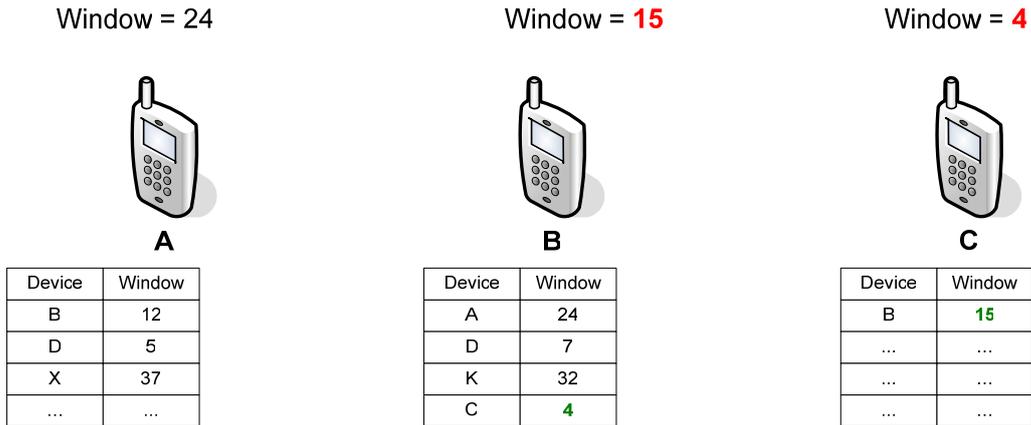


圖 3-26、散播演算法示意圖二（A 中資料未同步，B 中資料同步）



- Non-Synchrony
- Synchrony

圖 3-27、散播演算法示意圖二 (B、C 中資料皆同步)

此演算法仍存在一些問題，6-bit 的值最大為 31，更新 32 次後即繞回原值，若與另一藍芽裝置離上次交換期間，剛好改變 32 次狀態，則會發生誤判，因為此情形發生的機率低，並且只要再改變一次狀態即可正常運作，尚在容忍範圍內。狀態改變值都是在正常結束連線下做更新，假使網路環境不穩定或其它外在因素影響，可能會造成兩方已經同步，卻誤判成為同步的問題，在我們的設計之下，只要狀態更新值不同就會進一步連線交換資料，此種誤判情形造成的額外負擔並不會比沒使用狀態改變值的方法還差，因為沒有使用狀態改變值的系統本身就會對每一次接觸建立連線，然後再經由比對彼此資料來決定要傳送何使用者資料給對方。

另一個問題是使用者可能對接收到的檔案做刪除的動作，因為此演算法通常作用在短暫停留某一區域做有效率的散播，盡可能的散播絕大多數的個人資料（例如：捷運車廂、咖啡廳、書店），將來會針對此議題做更深入的研究，如何更有效率的散播資料。

3.4 相關議題

在以往的相關研究中並無在安全性問題上多加著墨，在此提出安全性問題仍因 Store-and-forward 機制下使得必須重新正視此問題。Store-and-forward 機制下，個人資料將可能透過中間人協助散播，首當其衝的就是資料竊改問題。

3.4.1 安全

在點對點的交換機制下，資料不會經由第三者傳送出去，可是 Store-and-Forward 的機制下，為了將使用者資料盡量散播，資料會透過第三者傳送，在沒有任何保護下，A 的資料透過 B 傳送給 C 時，很容易被 B 竊改而 C 不會發覺，這個問題會嚴重影響資料的正確性，如圖 3-28。

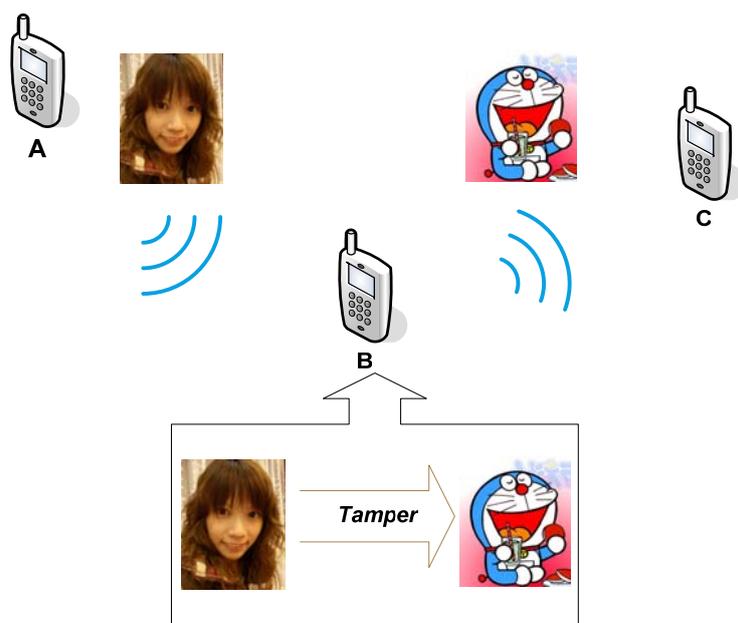


圖 3-28、無保護的資料竊改問題

本篇論文針對竄改與認證所提出的解法，是以現存多年的安全機制。使用者以自己的公鑰和帳號等基本資料經第三方的私鑰做簽證，如圖 3-29，並將此憑證隨個人檔案散播的過程中一併傳送出去，因為每個使用者皆具有第三方的公鑰，因此很容易可以解開憑證取得該使用者所散播的公鑰。透過使用者的公鑰即可驗證資料的正確性 [27]。

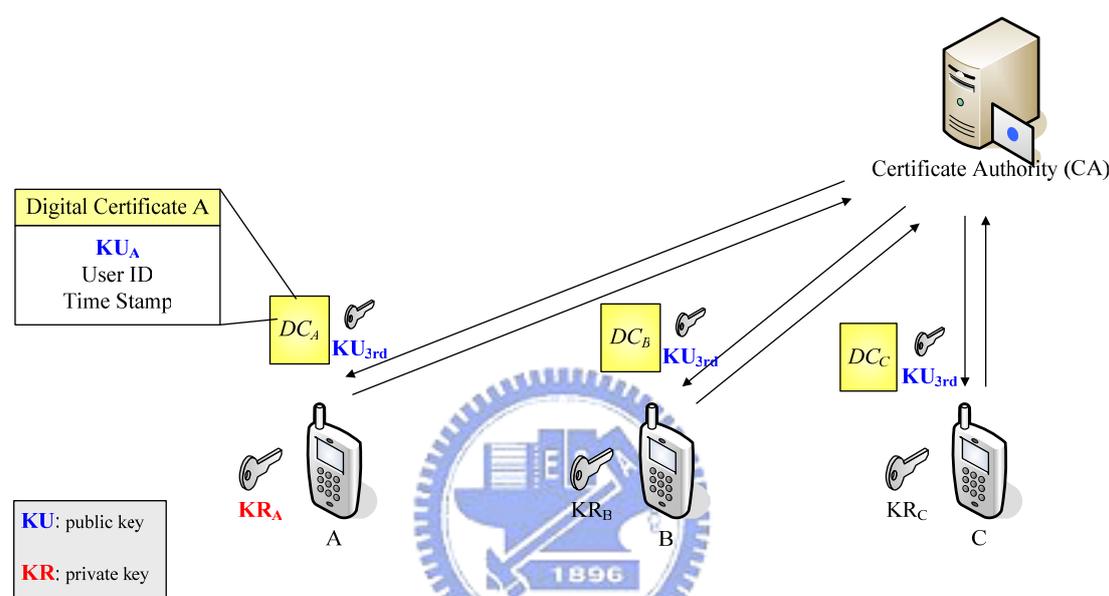


圖 3-29、簽證示意圖

下圖 3-30、圖 3-31 為加解密機制套用在 Blueg 系統上的示意圖，將已更改的資料 D2、D3 做 MD5 簽章 (Signature)，再以 Alice 的私鑰對簽章做加密取得訊息驗證碼 (MAC)。最後將憑證、個人資料與 MAC 散播出去。在此先對個人資料做簽章的目的是，減少以私鑰做加密的負擔。接收者 Bob 驗證資料的方式，先以第三方的公鑰對憑證做驗證，取得 Alice 的公鑰，再以 Alice 的公鑰與個人資料的簽章、訊息驗證碼做驗證。細節可參考 [27]。

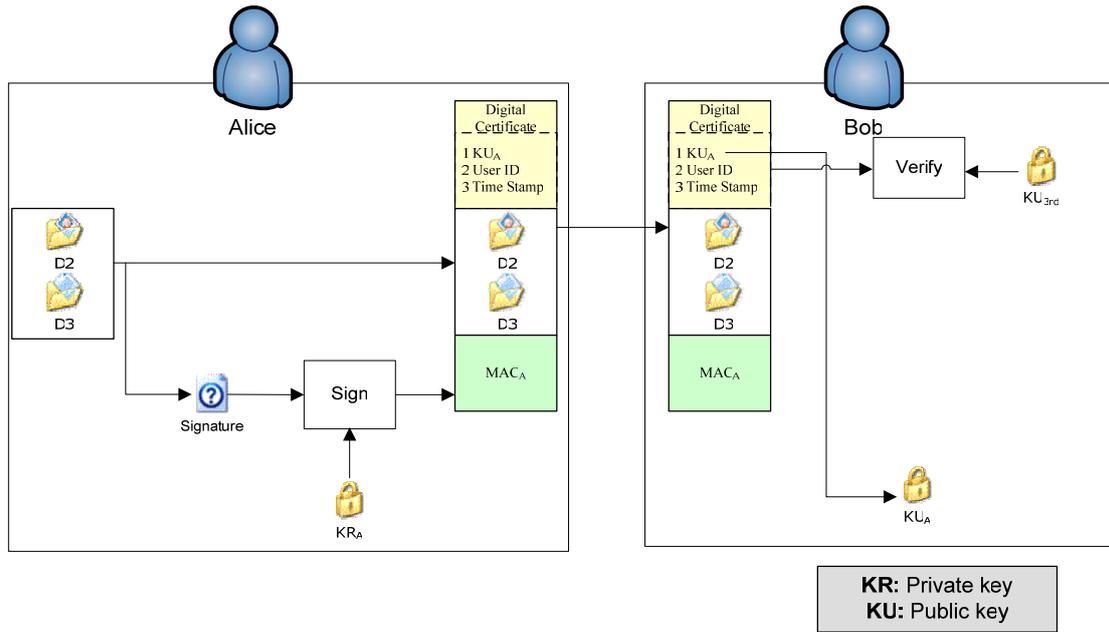


圖 3-30、接收者驗證憑證取到使用者公鑰

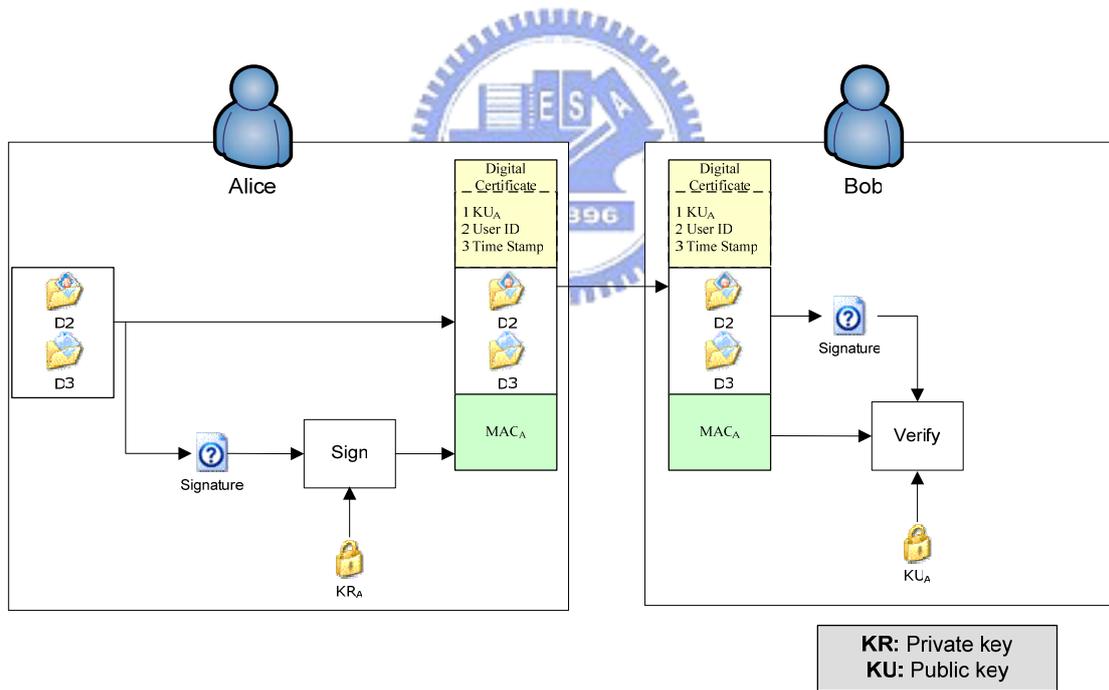


圖 3-31、接收者驗證結果

第四章、實作與實驗數據分析

本章節將實作上所遇到的問題與解決方法分敘如下，首先是在 Symbian 平台上實作同步收送的問題，接下來是中途斷線的處理。最後對實驗數據做分析。

4.1 同步接收傳送

Symbian 提供一種有別於其它作業系統的服務，稱非同步服務 (Asynchronous Services)，特性為服務要求送出後，經過一段時間才會完成，但在服務要求發出到完成的這段時間內，不能送出第二個服務要求[22][9][10]。在 Symbian 下提供別於執行緒 (Thread) 的多工環境，稱為 Active Object (AO)，AO 為事件驅動 (event-driven) 的。每一個 AO 同一時間只能送出一個服務要求，如圖 4-1。

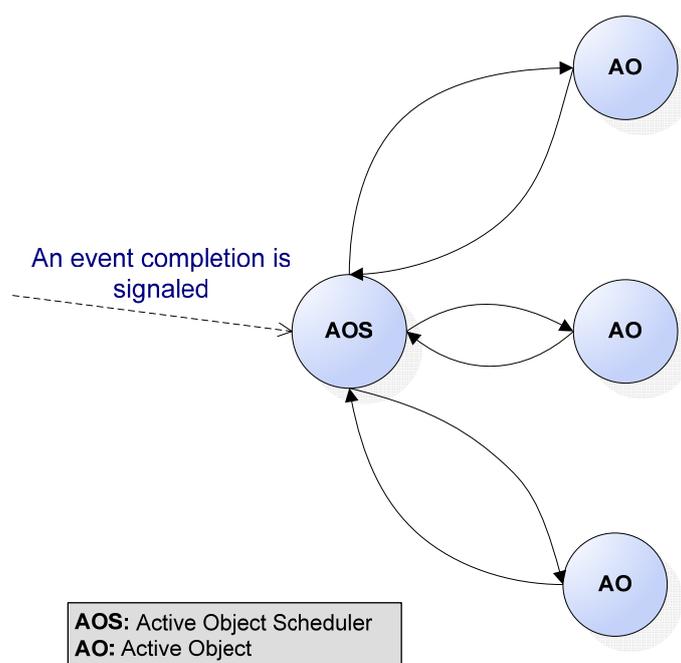


圖 4-1、Active Object 運作示意圖

本篇論文實作在 Symbian 作業系統上，在同步收送上的實作有些許挑戰。為了達到同步收送，個別實作 Writer 與 Reader 元件，Writer 將上層欲傳送出去的封包放進佇列中，一一處理；Reader 則每讀取一個完整封包，再將它分派給指定的 AO。以此將傳送與接收切割開來，達到同步收送目的。圖 4-2 為同步收送類別圖[22][11]，CMajorServer 與 MajorClient 透過 CAcceptor 與 Connector 建立連線，任何傳送與接收的封包都會透過 CAcceptor 與 Connector 轉送到 CWriter 與 CReader 做寫出與讀取的動作，CReader 需自行實作切割封包的功能，並在切割一個完整的封包後，送封包轉送到指定的 Component 做處理，因為非同步服務的特性，無法在接收封包同時傳送封包，反之亦然，因此實作 CDeputy 元件，在建立連線時當作 CMajorServer 或 MajorClient 的分身，一方做傳送檔案的工作，一方做接收檔案的工作。

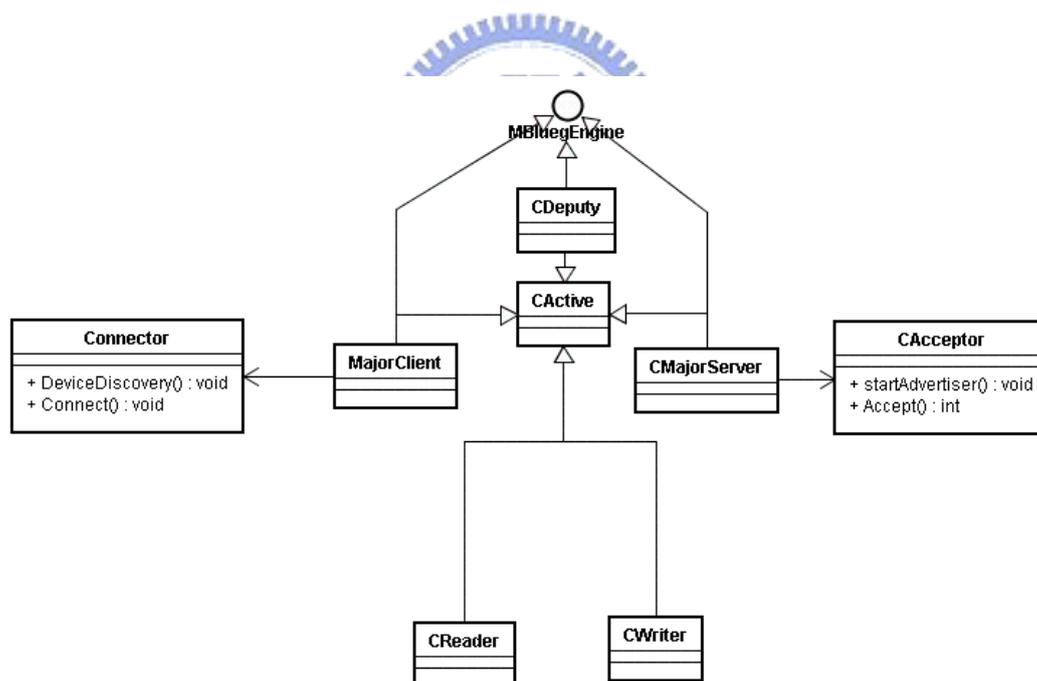


圖 4-2、同步收送模組的類別圖

4.2 冗餘資料切截

由於本論文檔案傳送結構與實作同步收送原因，當個人資料傳送中途斷線，仍能保留部分已傳送完成的資料，如圖 4-3。圖 4-4 為檔案封包傳送格式，以此封包傳送格式的設計，可以在不正常中斷連線時保留已接收的資料，而前一章節 4.1 中提及負責接收檔案的元件為 CReader，因此在 CReader 中特別實作中途斷線的處理，一但產生斷線，會進行處理將有效檔案收集，並往上通知 CAcceptor 或 Connector，進行冗餘資料切截，最後再把切截後的檔案往上通知 MajorClient 與 CMajorServer 做處理儲存。

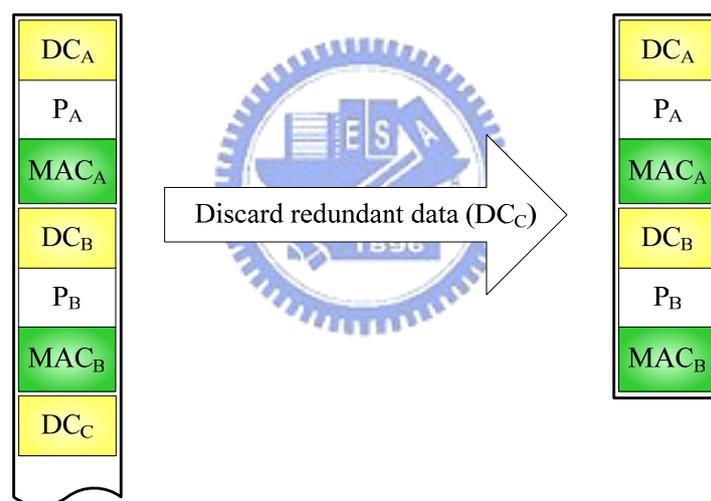


圖 4-3、中途斷線切截資料處理

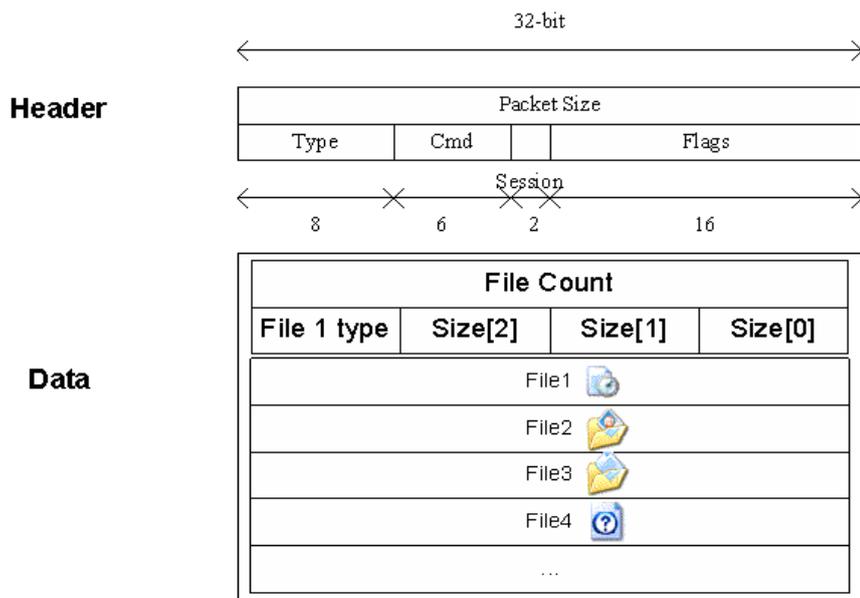


圖 4-4、資料傳輸封包格式

4.3 實驗數據及特性分析



在本論文當中，使用 Symbian 平台手機 Nokia N70，做了數個實驗觀察手機藍芽的特性，藍芽裝置搜尋、服務搜尋與建立連線的時間，傳輸速度與時間的影響，以及改良前後的比較。並比較有無散播演算法對資料散佈的影響，影響的因子及結果分析。

4.3.1 搜尋及建立連線時間與分析

本實驗的目的在觀察連線時間的特性，是否有規律性，並將舊有藍芽連線時間與本篇論文所提出的加速連線機制的連線時間做比較。

下表 4-1 為舊有藍芽連線時間的一次結果，橫軸為每單次連線，縱軸為以秒為單位的時間。可以發現相鄰的連線測試上，其時間有時快有時慢，花費時間範圍介於 1.93 秒~7.47 秒，搜尋藍芽的時間平均為 2.30 秒，建立連線平均為 2.83 秒，平均總共需 5.13 秒。

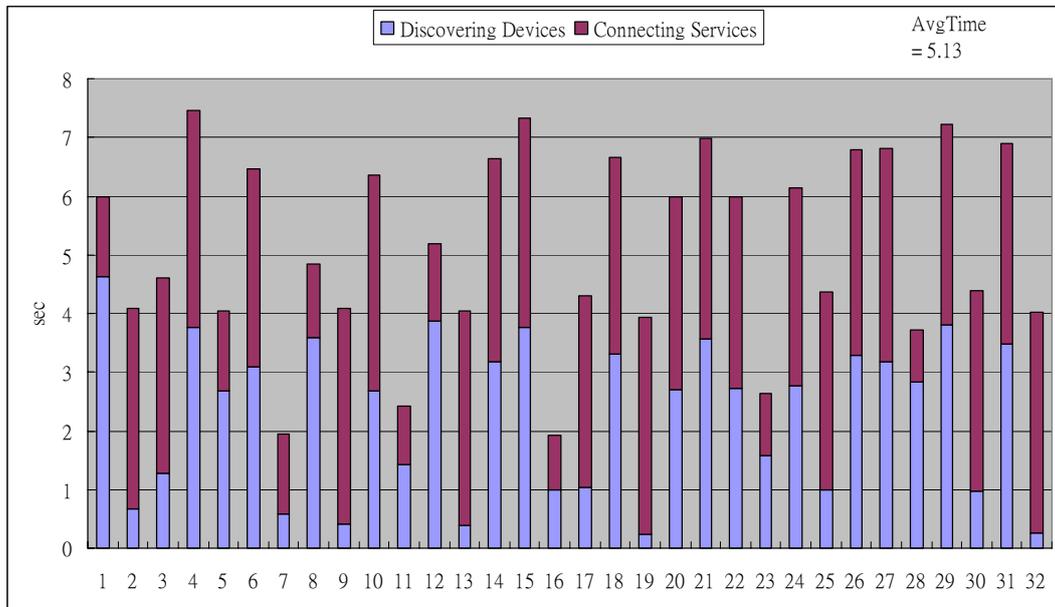


表 4-1、原藍芽搜尋及建立連線時間

下表 4-2 為加入加速連線機制的連線時間結果，橫軸為每單次連線，縱軸為以秒為單位的時間，花費時間範圍介於 0.98 秒~5.84 秒，搜尋藍芽的時間平均為 2.07 秒，建立連線平均為 0.94 秒，平均共需 3.01 秒。

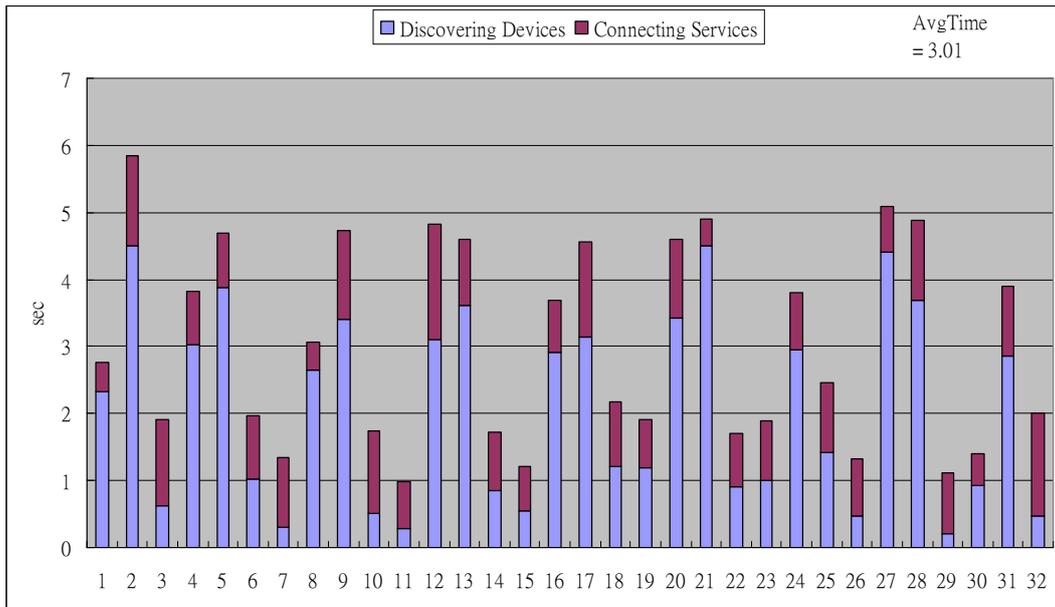


表 4-2、改良後藍芽搜尋及建立連線時間

將兩實驗的結果做一比較，改良前與改良後，如下表 4-3，差中改良後的藍芽搜尋時間變快，是因為藍芽裝置接收到 ID 封包後，會在一個亂數時間延遲後回應 FHS 封包，在藍芽規格中定義，使用 LIAC 允許有更短的亂數時間延遲，在實驗中也可以看得出來，的確有所改善。

再觀察改良後的建立連線時間有顯著的改善，快了將近三倍，可見在 FHS 嵌入特定訊息簡化連線過程有所幫助，實驗最後結果顯示，改良後的結果比改良前快 1.7 倍。

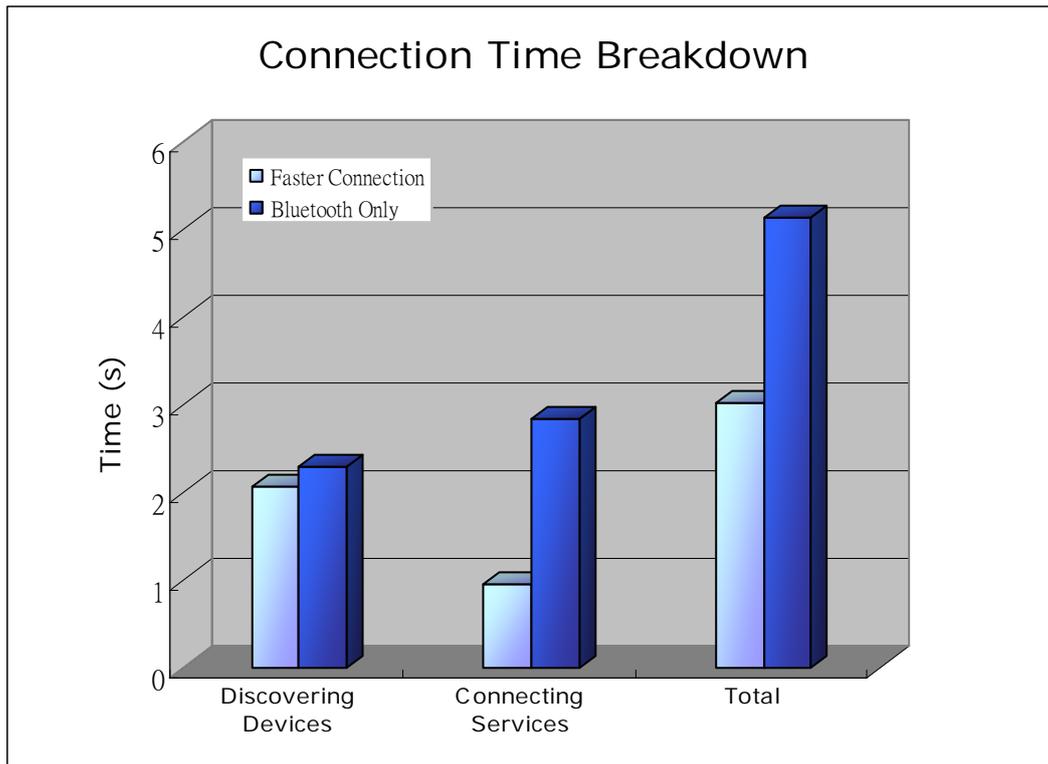


表 4-3、改良前後藍芽搜尋與建立連線時間分析



4.3.2 散播演算法分析

本實驗將比較有散播演算法與無散播演算法所得到的結果，並針對不同影響因素所造成的結果做分析。下表 4-4 為藍芽傳輸速度與時間的關係，其平均速度隨時間慢慢加快，約在第 6 秒與第 13 秒為一個階級，因此以下實驗模擬以不超過 5 秒的傳輸時間做為假設。

經過實際測試，發現手機藍芽的無線傳輸範圍半徑達 26.2 公尺；公車大小為長 8 公尺，寬 4 公尺；捷運車廂長 23.4 公尺，寬 3 公尺；一般咖啡廳內部空間通常也在這個範圍內，在此本實驗假設節點散播在長 25 公尺，寬 25 公尺的區域內，約 190 坪的區域範圍。

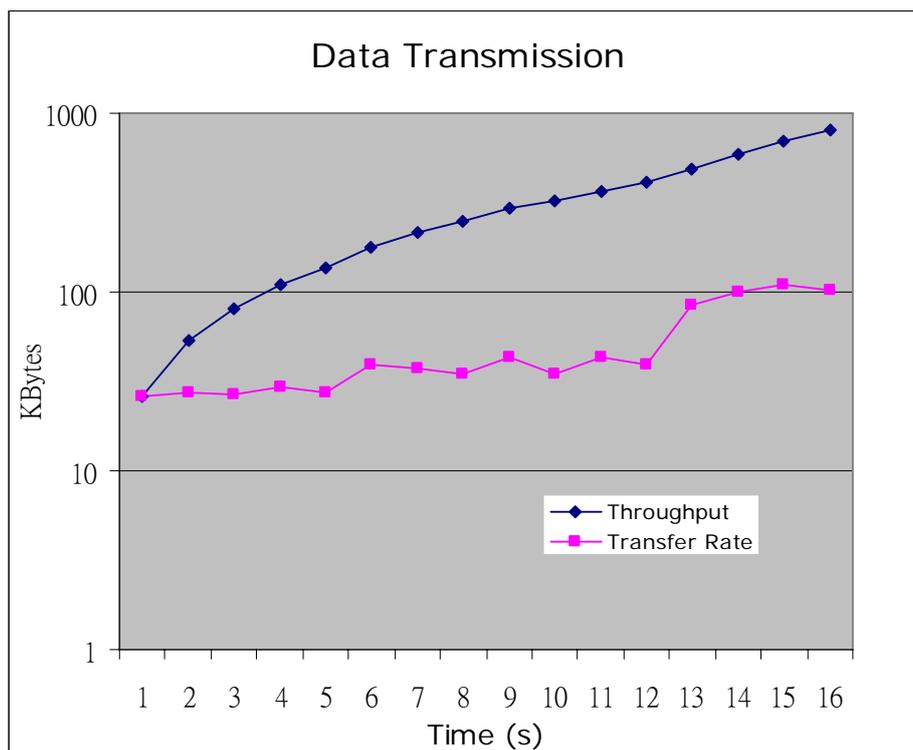


表 4-4、藍芽資料傳輸速率與資料傳輸總量

下表 4-5、表 4-6 為比較同區域 N 個節點所擁有的個人資料 (personal profiles) 筆數，橫軸為每節點擁有的資料筆數，縱軸為以分鐘為單位的時間。時間計量方式為每一個節點皆收到其餘節點的資料為止。得到的結果可看出在有散播演算法的花費時間較少，且節點數為 10 時，不超過 3 分鐘。同樣地，表 4-6 中，節點數為 20 時，也顯示出有散播演算法的花費時間較少。

在表中可以得到，單一節點所攜帶的資料筆數愈多，所需時間也呈線性成長。

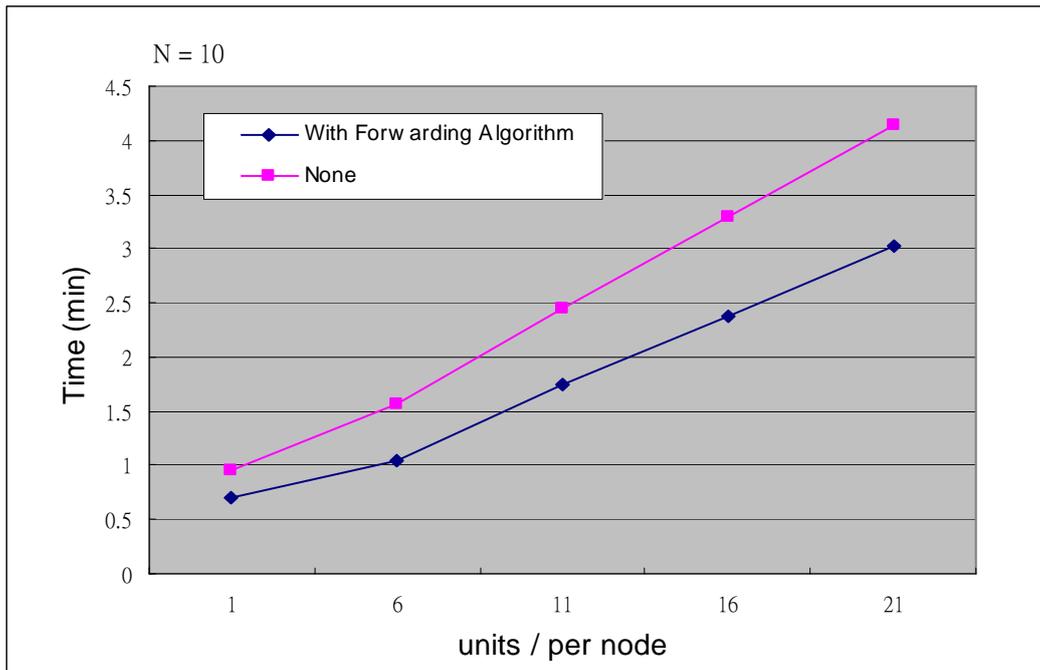


表 4-5、有無散播演算法散播效率比較 (N = 10)

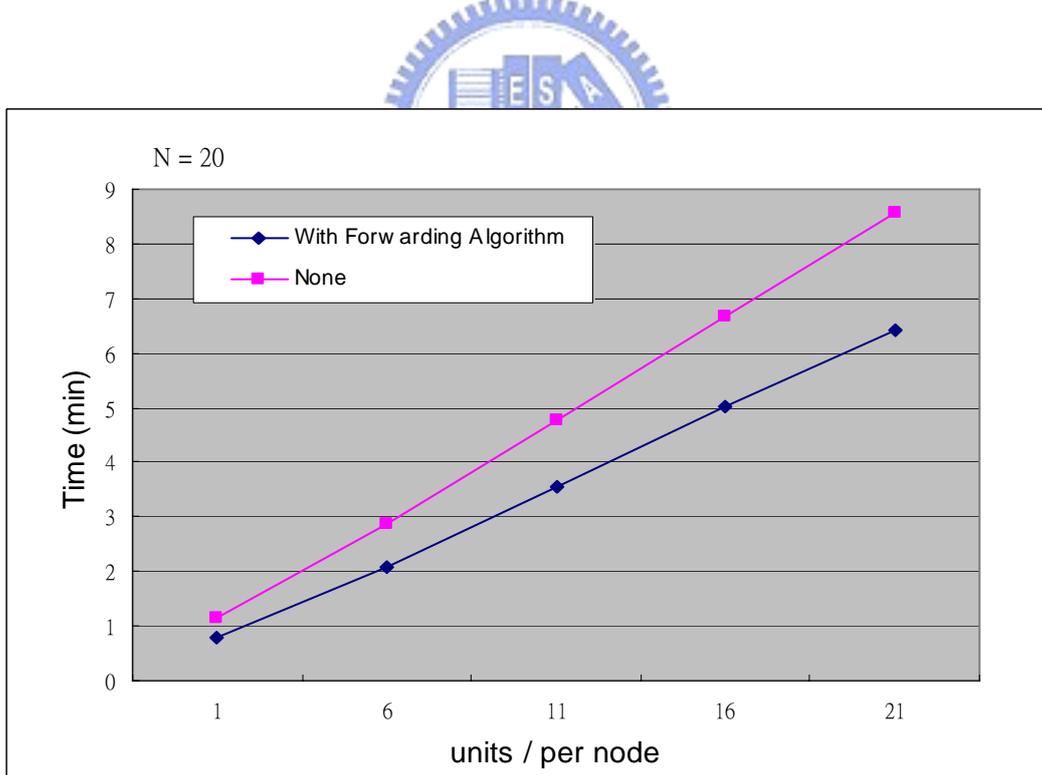


表 4-6、有無散播演算法散播效率比較 (N = 20)

由於散播演算法在搜尋藍芽裝置時，就會經由 FHS 封包得到該藍芽裝置內的個人資料是否有更改，若是資料沒更改就不會連線至該節

點進行傳輸。因此無散播演算法的節點在結束後可看出仍然會連上其它節點，有散播演算法的節點則保持在穩定的狀態下。表 4-7 為同一區域散播資料結束後，所有節點的活動情形，橫軸為以分為單位的時間，縱軸為建立連線時間所佔百分比。

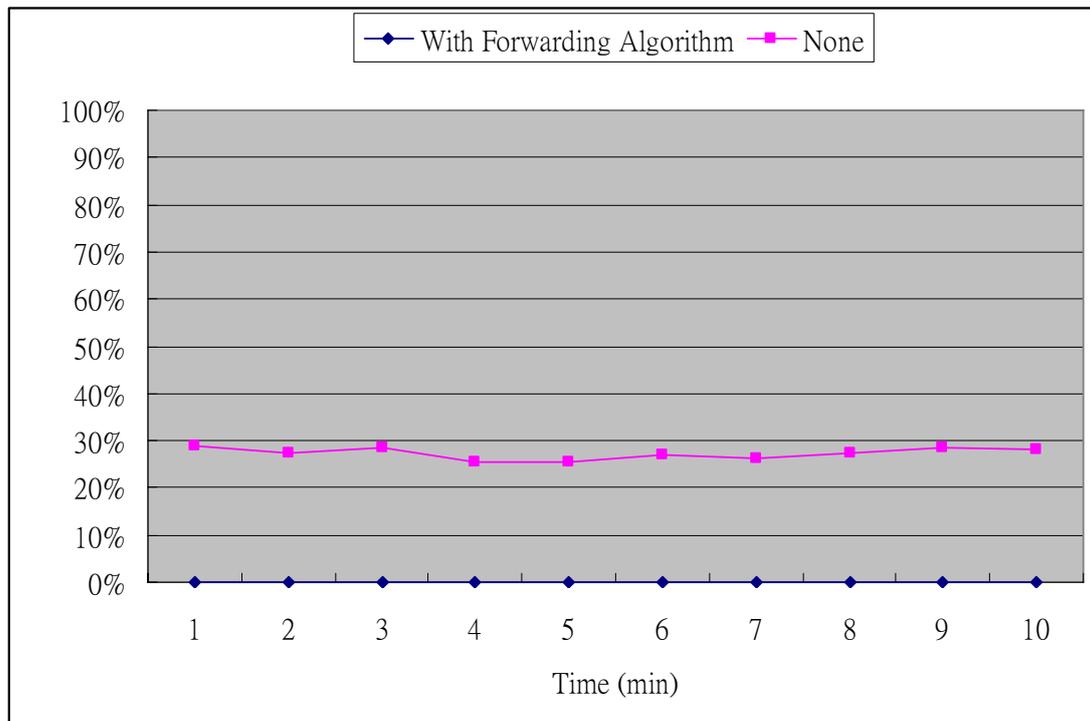


表 4-7、有無散播演算法節點活動狀態分析

第五章、結論與未來發展

在本論文當中，實作了一個建構在藍芽裝置上的行動運算系統。這個系統主要是用來做區域性的資訊散播。此系統包含三個部分：

- 以嵌入訊息快速建立連線
透過有限制的詢問存取碼與內嵌於回應封包內的服務訊息，跳過服務搜尋來節省建立連線所需要的時間負擔。
- Store-and-forward 與散播演算法
提出擴大人與人接觸的溝通方式，並在回應封包內嵌個人資料同步訊息，有效加快散播速度，並減少在區域性資料平衡之後建立連線。
- 防竄改之安全性問題
由於個人資料經轉送會有竄改之虞，因此以非對稱性公私鑰以及訊息驗證碼的方式來檢驗資料是否遭竄改。

本論文提出以藍芽無線傳輸為媒介往外散播個人訊息的新溝通模式，並提出 Store-and-forward 的機制來擴大散播範圍，為了讓訊息不會無止盡的散播，內嵌資料狀態值於藍芽回應封包內，建立散播演算法來有效率的散播與收斂。也考慮到在 Store-and-forward 之下所衍生出來的資料竄改問題。希望能進一步改良散播演算法，並以此架構為基礎，衍生更多的應用，提供給大家做參考。

參考文獻

- [1] Lauri Aalto, Nicklas Göthlin, Jani Korhonen, Timo Ojala, "Bluetooth and WAP Push Based Location-Aware Mobile Advertising System." *Proc. 2nd ACM MOBISYS*, Boston, MA, June, 2004.
- [2] Russell Beale, "Supporting Social Interaction with Smart Phones," *IEEE Pervasive Computing*, Special Issue: The Smart Phone, April-June 2005, pp. 35-41.
- [3] Bluetooth SIG, <http://www.bluetooth.com/>, 2006.
- [4] Bluetooth, "Assigned Numbers - Bluetooth Baseband", <https://www.bluetooth.org/foundry/assignnumb/document/baseband/>, 2006.
- [5] Bluetooth SIG, "Specification of the Bluetooth System. Version 1.1," 2001.
- [6] Bluetooth SIG, "Specification of the Bluetooth System. Version 1.2," 2003.
- [7] Jennifer Bray, Charles Sturman, Bluetooth: Connect Without Cables, Prentice Hall PTR, Dec 2000.
- [8] Nathan Eagle, Alex Pentland, "Social Serendipity: Mobilizing Social Software," *IEEE Pervasive Computing*, Special Issue: The Smart Phone, April-June 2005, pp. 28-34.
- [9] Richard Harrison, Symbian OS C++ for Mobile Phones, Volume 1, Symbian Press, April 2003.
- [10] Richard Harrison, Symbian OS C++ for Mobile Phones, Volume 2, Symbian Press, Aug 2004.
- [11] Michael J Jipping, Symbian OS Communications Programming, Symbian Press, June 2002.
- [12] Kameleon Technology, "Kameleon," available from <http://www.kameleon-europe.com/>, 2006.
- [13] KDDI, "Miracle Lovers," available form

- <http://pcweb.mycom.co.jp/articles/2001/06/28/c413s/002.html>, 2001.
- [14] Gerd Kortuem, Zary Segall, "Wearable Communities: Augmenting Social Networks with Wearable Computers," *IEEE Pervasive Computing*, Vol. 2, No 1, January-March 2003, pp. 71-78.
- [15] Linux, available from <http://www.linux.com/>, 2006.
- [16] Microsoft, "Windows Mobile,"
<http://www.microsoft.com/windowsmobile/>, 2006.
- [17] Nokia, "Nokia Sensor," available from <http://www.nokia.com/sensor/>, 2006.
- [18] Timo Ojala, Jani Korhonen, Tiia Sutinen, "Mobile Kärpät – A Case Study in Wireless Personal Area Networking." in *ACM International Conference Proceeding Series, Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia*, 2004.
- [19] Eric Paulos, Elizabeth Goodman, "The Familiar Stranger: Anxiety, Comfort, and Play in Public Places," *Proc. Human Factors in Computing Systems (CHI 2004)*, ACM Press, 2004, pp. 223–230.
- [20] Proxidating, "Proxidating," available from <http://www.proxidating.com/>, 2005.
- [21] David Scott, Richard Sharp, Anil Madhavapeddy, Eben Upton, "Using Visual Tags to Bypass Bluetooth Device Discovery," in *ACM SIGMOBILE Mobile Computing and Communication*, Vol.9, Issue 1, Jan, 2005.
- [22] Jo Stichbury, Symbian OS Explained, Symbian Press, Oct 2004.
- [23] Symbian, "Symbian OS," available from <http://www.symbian.com/>, 2006.
- [24] Michael Terry, Kathy Ryall, "Social Net: Using Patterns of Physical Proximity over Time to Infer Shared Interests," *Proc. 2002 Conf. Human Factors in Computing Systems (CHI 2002)*, ACM Press, New York, 2002, pp. 816–817.
- [25] 無名小站, "無名小站," available from <http://www.wretch.cc>, 2006.
- [26] 樂多, "樂多行動日誌," available form

<http://moblog.roodo.com/info/index.htm>, 2006.

- [27] 賴溪松, 韓亮, 張真誠, 近代密碼學及其應用, 二版, 松崗出版社, 1998.

