

混合式藍芽散網建構演算法

MixBlue Scatternet Formation of Bluetooth Networks

¹ 陳欣煒、¹ 盧永仁、¹ 劉如生、² 何應魁

¹ 元智大學資訊工程研究所

csrobinl@saturn.yzu.edu.tw

² 中正理工學院資訊科學系

ykho@ccit.edu.tw

摘要

藍芽規格定義了三種無線電傳輸功率，而目前各類文獻的探討均以單一類型的裝置為基礎，隨著產品的發展，所以我們可以預見不同功率的裝置在一起運作的情形出現。本篇論文將利用藍芽裝置傳輸距離不同的特性，提出一個混合兩種藍芽等級裝置的散網建構演算法 MixBlue，建構整合 Class 1 和 Class 3 所形成的散網 (Scatternet)，並加入微網 (Piconet) 合併程序，來降低環境中的微網數。此外，提出保留一個 Active Slave 的概念，可以讓微網選擇更適當的裝置當作 Slave。本演算法可以降低微網數目，提高裝置連通率，並建立較小 hop count 的傳輸路徑。

關鍵詞：藍芽，微網，散網，混合，功率。

Abstract

According to the specifications, bluetooth system defines three types of devices (Class 1, 2 & 3) with different radio powers. However, at present, all the related literatures of bluetooth discuss only one kind of Class 3 devices. As new products (Class 1 devices) have been developed, we may anticipate the coming wireless environments with Class 1 & 3 mixed products connected to each other. Based on the different transmission ranges of Class 1 and 3 devices, this paper proposes an efficient scatternet formation algorithm named MixBlue. Basically, we will construct a scatternet including the two different types of devices and design a merging mechanism for piconets to reduce the total number of piconets. Besides, in order to select a more suitable slave for piconet, we will propose a way to reserve one active slave for later utilization. Our methods will reduce the total number of piconets,

increase the device reachability and set up a routing path with smaller number of hop counts quite efficiently.

Keywords : Bluetooth , Scatternet , Relay , MixBlue , Power.

一、簡介

藍芽 (Bluetooth) 技術的出現，提供 10 公尺到 100 公尺以內、無方向性短距離無線傳輸的另一個選擇。這項訴求低價、低耗電量的無線傳輸技術，主要是用於取代各裝置間的纜線，使用者無須做複雜的設定，就可以輕易連接各裝置。該特性在不久的將來當會被廣泛的加以運用，從大型家電用品、各種行動設備到一枝筆都可能內嵌藍芽的晶片，以達到更多更方便的生活應用。

微網 (Piconet) 是藍芽裝置聯繫通訊的最小單位，由一個 Master 和最多七個 Slave 所構成，而不同的微網都有其不同的跳頻順序。根據規格書的定義，藍芽裝置可以擔任單一微網的 Master 或是 Slave，也可以是一個微網的 Master 及另外一個微網的 Slave，或是參與多個微網擔任 Slave 的角色，而參與兩個以上微網的裝置稱之為 Relay。兩個以上的微網，透過 Relay 的聯繫就形成所謂的散網 (Scatternet)。

藍芽裝置要建立微網進而形成散網，需要經過幾個程序。當一個裝置要探索在其所在的地區中所有可能的藍芽裝置時，所使用的程序就是 Inquiry。在這段期間，執行 Inquiry Scan 程序的裝置，收到有效的 Inquiry 訊息，就會以 FHS 封包回應。Inquiry 和 Inquiry Scan 利用特定的跳頻序列 (快的 Inquiry 和慢的 Inquiry Scan)，這是設計來減少在達到跳頻同步前所需花費的時間。執行 Inquiry 的裝置從 FHS 封包得知回應裝置的位址等相關基本資

訊，為了建立連線，進入 Page 程序，此裝置即擔任此微網的 Master。而回應 FHS 封包的裝置進入 Page Scan 的狀態，等待被連結成為 Slave。如果已知欲連結裝置的位址資訊，是可以不進行 Inquiry 程序就直接進入 Page 程序和執行 Page Scan 程序的裝置進行連結。

依照定義我們可以將藍芽所形成的散網(Scatternet)視為一個隨意型無線網路(Ad hoc Wireless Networks)的特例。以藍芽為主題的文獻相當多，概略可以分為硬體類與軟體協定類。其中，軟體協定類的主題有探討 Master 的選擇、置換問題，有藍芽散網的架構形成問題，在散網中選擇傳輸路徑問題，還有 Relay 的運作排程...等問題。

由於在藍芽規格書中並無明確定義藍芽微網(Piconet)，散網(Scatternet)的形成，所以大家只須遵守基本的規定限制，便可以依照自定的演算法去形成想要的散網型態，但不同的架構也有不同的假設。現有的建構演算法，概略可以分兩類，一種是有一個 Coordinator，它知道範圍裡所有裝置的資訊，由其來控制分配散網的形成，或是由此 Coordinator 來發起散網的建構。像是 BlueTree[12]、BlueRing[8]、BTC[9]...等，都是屬於此類。另外一類，就是不沒有 Coordinator，所有的裝置節點完全分散式的運作。本篇論文屬於這一類。

許多的散網建構演算法都有一些特殊的假設限制，例如：假設所有裝置都在彼此的通訊範圍中，或是限制裝置的數目...等。下面僅舉出幾個與本論文有相關的文獻摘要。

(一) 相關研究

1. 微網合併

在文獻[7]中，提出在所有裝置節點在彼此通訊範圍假設下，一個散網建構的演算法，提到了 Piconet 的合併問題，當 Piconet 與 Piconet(或是一個孤立節點)連接時，將會執行合併的動作，如果數目未超過 Piconet 的上限，則使其合併為單一的 Piconet。如果超過，則形成 Scatternet(一個滿的 Piconet 連結一個未滿的 Piconet)，其中未滿的 Piconet 繼續與別人執行合併的動作。

2. A Scatternet operation protocol

文獻[10]中，沒有限制節點必須在彼此通訊範圍裡，這樣的環境架構與本篇論文相似。開始時，所有的裝置執行 Neighbor Discovery Protocol 來偵測所有周圍的裝置，

每個節點將經過隨機時間的 Page Scan 後，才會進入 Page 狀態。一但加入 Piconet 成為 Slave 後不再動作，而 Master 則繼續執行 Page 看看還有沒有裝置可以加入。

如果 Master 滿了或是一直沒有裝置加入，便通知自己的所有 Slave 成員，該 Piconet 的所有資訊。此時，Slave 可以進入 Page Scan 狀態，如果沒有其他的 Piconet 可以加入，就再進入 Page 狀態。

當 Master 建立完成 Piconet，經過一段時間，開始執行"非連結偵測"，可以偵測到某一個 Slave 的鄰居節點是非連結的。並選擇最少數目的 Slave 能涵蓋這些非連結的裝置來建立 Piconet。

並在最後提出一個散網調適協定(Scatternet Adjustment Protocol)，內容是 Slave 加入 Piconet 後仍然執行 Inquiry Scan 一段時間，而 Master 在未滿的狀況下週期執行 Inquiry。另外，當裝置節點脫離了原 Piconet 成為孤立點時，若以前它是 Master 則執行 Inquiry 來建立 Piconet，相反的則執行 Inquiry Scan 來等待加入別的 Piconet。

3. Blue-star

在文獻[13]中，沒有限制裝置節點的數目，也沒有限制節點的行動性，並且是完全分散式處理，這樣的環境架構與本篇論文的環境架構相符，下面簡述此演算法。

開始所有的節點均為孤立點，經過第一階段(即 Inquiry、Inquiry Scan、Page、Page Scan 的連結程序)，加入 Piconet 後，Slave 只能執行 Inquiry Scan 等待加入其他的 Piconet。

Master 則週期進入 Inquiry 直到 Piconet 滿，除此之外，Master 也會週期進入 Inquiry Scan。這樣的方式，可能產生兩個 Piconet 的 Master 分別進行 Inquiry 和 Inquiry Scan 而連接的狀態，此時，執行 Inquiry Scan 的 Master 將切斷所有與其 Slave 的連線。

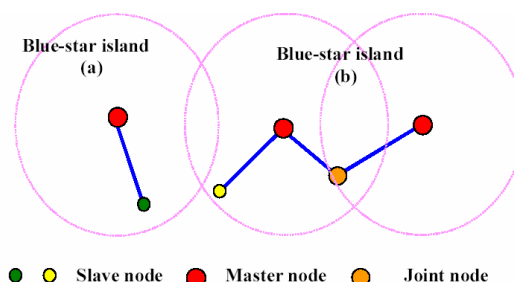


圖 1 Blue-star island

可相互連結的微網所形成散網，此文獻稱為 Blue-star island。上圖 1，顯示兩個孤立 "Blue-star island"的問題，該篇提出 Routing Trigger 的概念，當 Slave 接收到 Trigger 便會去執行 Inquiry 的動作，去尋找它傳輸範圍中，是否有其他的 Slave，若有則建立新的 Piconet 使其連通。

許多文獻的探討都要求在彼此通訊範圍中，甚至加上其他的假設，只有文獻[10][13]和本論文的環境相符，沒有特殊的限制，完全分散式的執行。文獻[10]中大部分的節點幾乎都是一起一個一個步驟執行，這樣對於分散式環境並不合理。文獻[13]中的方式較符合分散式環境的運作。另外形成過多的 Piconet 也將造成傳輸時效率的問題，[7]是所有文獻中少數提到 Piconet 的合併，隨然它的環境架構與本篇論文不同，但仍極具參考價值。

本論文將 Piconet 合併的概念用在沒有特殊環境架構的完全分散式環境，這與文獻[10][13]有很大的不同。此外，目前所有的文獻均探討同一等級的裝置之散網的架構，本論文將提出混合 Class1 與 Class3 不同等級的裝置的散網建構方式，下面將詳細說明建構的原則與方式。

二、研究方法

(一) MixBlue 建構演算法

本論文主要探討 Class3(最大通訊範圍 10m)和 Class1(最大通訊範圍 100m)兩種裝置在混合出現在同一區域的微網建構演算法。

1. 建構原則

本建構演算法是屬於分散式的做法，每一個裝置遵守一定的原則與方法運作，下面將列出建構原則並說明訂立的原因。

- 一個 Class3 的裝置最多參與兩個 Piconet (Class1 則最多參與三個，其中一個是 Class3 為 Master 的 Piconet)。避免太多的 inter-Piconet 的負載都落在少數的 Relay 上，同時也降低在多個 Piconet 間轉換頻道的負擔。
- 兩個 Piconet 間最多只能有一個 Relay。Master 可以節省 AM_ADDR 分配給其他裝置使用，從另一個角度看，讓 Piconet 可以去連接更多的其他的 Piconet。
- Class1 和 Class3 的關係：Class1 的裝置只能為 Class3 裝置的 Slave。Class1 可以為

Class1 的 Master，卻只能是 Class3 的 Slave，主要的考量是兩者傳輸距離不同，互通的問題，同時也能發揮 Class1 長距離傳輸的特性

- 一個 Class3 裝置為 Master 的 Piconet 只能有一個 Class1 的 Slave
- 每個 Piconet 以容納 6 個 Slave 為上限。保留一個 Active Slave 可以讓 Master 自由運用，例如可以選擇更好更適當的 Slave...等。
- 盡可能讓所有 Piconet 至少有一個 Relay。一個 Piconet 若沒有任何的 Relay 會造成孤立，其他裝置無法跟該 Piconet 聯繫。

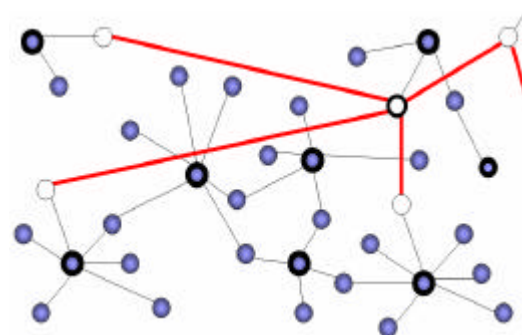


圖 2 MixBlue 概念架構

2. Class1 裝置與 Class3 裝置的關係

由於兩種不同等級的裝置通訊範圍不同，所以我們必須限制若 Class1 與 Class3 在同一個 Piconet 中，Class1 裝置只能為 Class3 裝置的 Slave，如此才能保證可以雙方向通訊。

基於此原則，Class3 裝置啟動後，可以隨機進入 Inquiry 和 Inquiry Scan，然後執行 Page 與 Page Scan 來形成 Piconet。而 Class1 的裝置啟動後只能執行 Inquiry Scan 來等待加入 Class3 為 Master 的 Piconet，直到加入後或是逾時才能開始執行 Inquiry 和 Inquiry Scan 來建構 Class1 的 Scatternet。

3. Class3 程序

我們規定，裝置一旦形成 Piconet 後，Master 在尚未滿的狀況下只能繼續執行 Inquiry 來尋找其他 Slave 的加入，而 Slave 只能執行 Inquiry Scan 來看看是否有其他的 Piconet 可以加入。又 Slave 要參與第兩個 Piconet 時，必須通知其原所屬 Piconet 的 Master 此項資訊來避免兩個 Piconet 共用兩個 Relay 的狀況。

上圖 3 顯示，A, B 兩裝置分別隨機進入

Inquiry (I) 和 Inquiry Scan (IS) 且此時在彼此通訊範圍，所以經由 Page 和 Page Scan 程序，形成一個以 A 為 Master 而 B 為 Slave 的微網 (Piconet)，裝置 A 繼續執行 Inquiry 而裝置 B 則執行 Inquiry Scan。

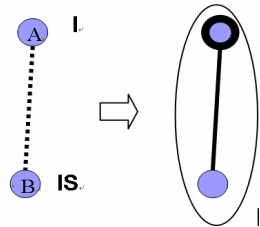


圖 3 形成 Piconet (I)

下圖 4 顯示另一種常見的狀況，原 Piconet 有三個裝置，一個 Master，兩個 Slave (一個為 Class1，另一個為 Class3)，Master 執行 Inquiry 而外面的裝置正在執行 Inquiry Scan，又此時在彼此通訊範圍中，所以經由 Page 連結程序，形成一個四個裝置的 Piconet (1 個 Master + 3 個 Slave)。

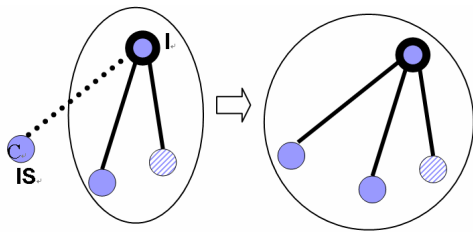


圖 4 形成 Piconet (II)

4. 合併程序

當兩個 Piconet 透過 Relay 聯繫，可以調整合併 Piconet，來降低 Piconet 的數目，執行時先比較兩個 Piconet 的數目，由少的加入多的。下圖 5 舉一個通例來作合併的說明。

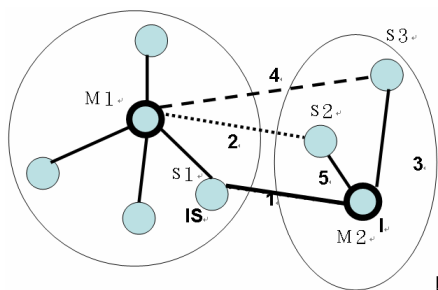


圖 5 合併程序 (I)

圖中的虛線表示尚未連結的狀態，M 2

執行 Inquiry，S1 執行 Inquiry Scan，經確認 S1 不是重複的 Relay，所以 S1 成為 M1 和 M2 的 Relay，兩個 Piconet 可以透過此點做聯繫，發現兩者均尚未滿 6，所以準備執行合併的動作。

首先比較兩個 Piconet 的成員數目，M1 Slave 的數目 $S(M1)=4 > S(M2)=2$ ，所以準備由 M2 選擇成員加入 M1 直到該 Piconet 滿 6 (若兩個 Piconet 的 Slave 數目相同，則比較 Master 的 BD_ADDR)。M2 經由 Relay S1 通知 M1，分別告知 S2 和 S3 的資訊，此時 M2 也通知 S2 和 S3 執行 Page Scan 等待加入 M1，由於 M1 知道 S2 和 S3 的資訊，所以可以直接執行 Page 的動作，而系統允許 Class 3 裝置在執行合併程序時，可以暫時同時參與第 3 個 Piconet (Class 1 裝置則可參與第 4 個 Piconet)。

當連線 2 建立完成，M2 切斷與 S2 的連線 5。另一個狀況，連線 4 可能因為 M1 和 S3 不在彼此通訊範圍內，所以無法完成連線轉移，而當 M2 Piconet 尚有 Slave 未轉移成功，則 M2 本身將不會加入 M1 的 Piconet。

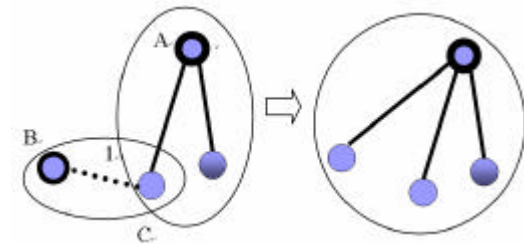


圖 6 合併程序 (II)

合併程序執行到最後，會遇到圖 6 中的狀況，即 B Piconet 內除了擔任 Relay 的 C 外，所有的 Slave 都轉移成功了，剩下 Master 本身 B 準備合併。在此狀況下，B 透過 C 傳送本身的相關資訊給 A，然後自己進入 Page Scan 的狀態，等待 A 的 Page，如果連結成功，B 將解除與 C 之間的連線 1。

5. Class1 程序

如前所述，Class1 裝置與 Class3 裝置的關係只有一種，即 Class1 裝置為 Class3 裝置的 Master。所以 Class1 的裝置啟動後只能執行 Inquiry Scan 來等待加入 Class3 為 Master 的 Piconet，直到加入後或是逾時才能開始執行 Inquiry 和 Inquiry Scan 來建構 Class1 的 Scatternet。又為避免 Class1 裝置參與過多的 Piconet，降低傳輸效能，規定 Class1 只能參

與一個 Class3 為 Master 的 Piconet,及最多兩個 Class1 為 Master 的 Piconet。

而 Class1 等級的 Scatternet 建構及合併方式與 Class3 類似,不同的地方是加上利用訊號強度指標來判斷距離,使 Class1 的 Piconet 涵蓋範圍較大。下面將說明 Class1 運作程序與 Class3 不同的地方,以下未說明等級的裝置,均視為 Class1 裝置加入 Class 1 為 Master 的 Piconet 後,Slave 必須得到 Master 許可才能執行 Inquiry Scan 的動作,而 Master 的判斷標準是依照訊號強度的指標是否超過一個門檻值(Threshold)來判斷,因為距離過短的裝置形成 Piconet 將無法運用其遠距離傳輸的特性。Slave 在得到 Master 的許可便可執行 Inquiry Scan 等待加入另一個 Piconet。

圖 7 中的虛線表示開始時尚未連接,圖中的裝置均為 Class1,此 Piconet 已經有 6 個 Slave(其中 S1 與 S3 未達門檻值,所以不能執行 Inquiry Scan),而 M 因為有保留第七個 Active Slave,所以週期性進入 Inquiry 看看有沒有更好的 Slave 人選。

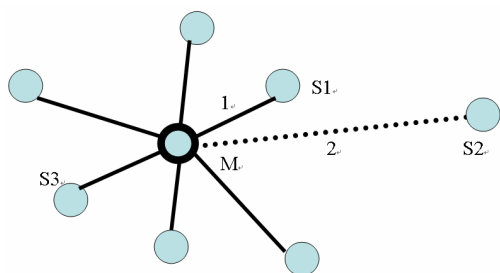


圖 7 使用第七個 Active Slave

由訊號強度的指標(排除雜訊干擾等因素),推得 S2 的距離較遠且其為 Relay,所以 M 選擇一個低於 Threshold 且其不為所參與 Class 3 為 Master 的 Piconet 之唯一 Relay 的裝置做置換。此外,當存在 M Piconet 中的 Slave 雖然其超過 Threshold 但它除了參與此一 Piconet 外,沒有加入其他的 Piconet, M 也可以選擇其為 S2 的置換對象。有此例中選擇將 S1 做置換,切斷連線 1。

6. 孤立微網問題

在這類分散式進行的演算法中,又沒有規定在彼此通訊範圍內的限制,所以產生孤立微網(isolated Piconet)是有可能的,所謂的孤立 Piconet 就是該 Piconet 中所有的成員都只有參與此一 Piconet,以下將說明兩種可能的狀況。

第一種狀況,我們借用圖 4-6 做說明,

M 原本六個 Slave 因為某原因皆無法加入其他 Piconet 成為 Relay(可能是周圍的 Piconet 都已經滿了),所以包括 M 在內的七個裝置將為孤立的 Piconet,而我們希望每一個 Piconet 至少有一個 Relay,雖然這樣並不保證一定可以連通,但可以把孤立的狀況降低,提高連通率,此時, M 就可以利用還有一個 Active Slave 的容量來為 Piconet 加入 Relay。假設圖中的 S2 是某一 Piconet 的 Slave,正在執行 Inquiry Scan,而 M 為了尋找 Relay,將週期進入 Inquiry,只要 M 與 S2 此時在彼此通訊範圍中,就可以讓 S2 加入 M 的 Piconet,解決孤立 Piconet 的問題。除了上述兩項使用外,預留的第七個 Active Slave 還可以做其他的特殊使用。

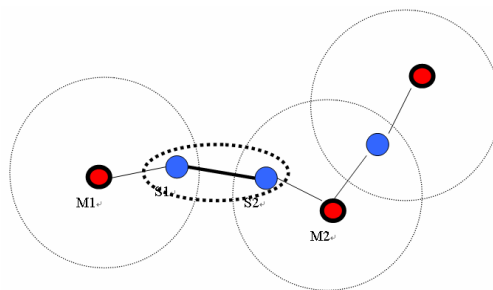


圖 8 Routing Trigger

圖 8 表示另外一種可能發生的孤立的狀況,依據本演算法的定義,裝置開始時隨機執行 Inquiry 和 Inquiry Scan,加入 Piconet 後,限制 Slave 只能做 Inquiry Scan, Master 只能做 Inquiry,所以雖然 S1 和 S2 在彼此的通訊範圍中,但 S1 與 S2 分別不在 M2 與 M1 的通訊範圍中,便有可能造成上圖中的左邊孤立的 Piconet。本問題的解決方式,在欲建立傳輸路徑時,當 Master 接收到 RREQ 封包,Master 允許其非 Relay 的 Slave 進行 Inquiry Scan 一固定時間。圖例中 M2 收到 Scatternet 的傳來的 RREQ, M2 允許 S2 去執行 Inquiry,此時 S1 和 S2 就可以建立連線形成 Piconet,藉由兩個 S1 與 S2 傳送資訊到 M1。此種方式,文獻[13]中,稱為"Routing Trigger"。

本演算法使用第七個 Active Slave 來尋找 Relay,在加上 Routing Trigger 的方式,來解決可能造成的孤立 Piconet 的問題。

三、模擬與分析

(一) MixBlue 與 BlueStar 裝置連通率的比較

模擬範圍設定為 100m*100m。因為 BlueStar 的演算法,僅適用於所有裝置為單

一種類的狀況(Class 3)，與本演算法 MixBlue (可以混合 Class1 與 Class 3)不同，所以模擬時設定所有的裝置均為 Class 3 的單一種類的裝置，模擬時間為 30 個單位時間。

而兩種演算法在執行 Routing Trigger 後，連通率將大幅提高，這是可以預期的，所以設定兩種演算法在均未執行 Routing Trigger 時，做連通率的比較。圖 9 中，X 軸為模擬時的 Class 3 裝置節點數目。而 Y 軸 device reachability 代表的意思為任何兩點可以透過散網(Scatternet)的繞送達到聯繫通訊的比例，其計算方式為

$$\frac{\text{實際可以連通的link數}}{\text{所有裝置的link數}}$$

分母中的所有裝置 link 數即為 $C_2^{\text{裝置數目}}$ 。

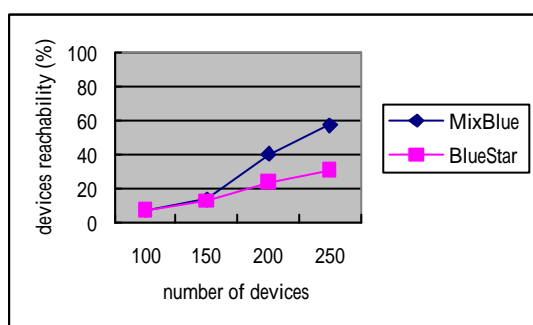


圖 9 MixBlue 與 BlueStar 的連通率比較

圖中顯示，在 250 個裝置時，MixBlue 的連通率接近 60%，比 BlueStar 的約 30% 提高了約一倍的連通率。這是因為 BlueStar 為了減少 Piconet 的數目，允許 Piconet 中的 Master 執行 Inquiry Scan 來加入其他的 Piconet，此時該 Master 會解散其原本的 Piconet，此舉雖然可以降低 Piconet 的數目，但這對於連通率會產生影響。此外，BlueStar 對於 Scatternet 的建構時間也將需要較常的時間，因為被解散的 Piconet 的 Slave 成員，都將重新執行 Inquiry 或是 Inquiry Scan 的程序，來形成 Piconet，進而形成 Scatternet。

而本演算法 MixBlue 的 Merge 程序，除了可以降低 Piconet 數外，對於連通率的提升也有幫助，因為在執行 Merge 程序時，裝置的轉移是透過 Relay 來得知欲合併裝置的資訊，所以只需執行 Page，所以在整個 Scatternet 的形成上，MixBlue 也將比 BlueStar 有效率。

此外，在 MixBlue 中，Class 3 為 Master 的 Piconet 保留了一個 Active Slave，來選擇已加入其他 Piconet 的裝置成為其第七個 Slave，來降低形成孤立微網(isolated Piconet)的機會。所以 MixBlue 有高於 BlueStar 的連

通率是合理的。

(二) 使用 Class 1 裝置對於 link 數的影響

本演算法 MixBlue 是為了讓不同通訊範圍的兩類裝置(Class1 和 Class 3)在一起運作，並且發揮 Class3 的較長傳輸距離的特性。所以在比較有無 Class1 裝置的參與對於 link 數的影響，模擬範圍為 150m*150m，模擬 30 個單位時間，裝置總數目為 500 個，其中全為 Class3 的裝置，和有 50 個 Class1 的裝置做比較。

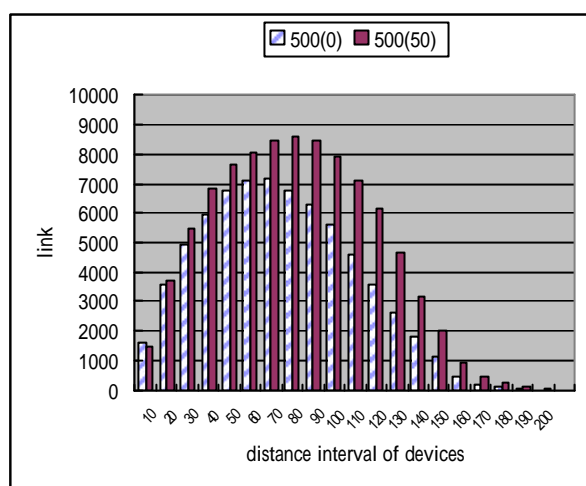


圖 10 使用 Class 1 裝置對於 link 數的影響

圖 10 的 X 軸為距離的間隔區間，以 10m 為一個區間，Y 軸為 link 數，任一長條表示任何兩裝置的距離在此區間的可連結數。圖中顯示在距離 10m 以內，沒有 Class 1 參與的 link 數多於有 50 個 Class 1 裝置的 link 數，但超過 10m 的區間，有 Class 1 參與的 link 數均多於無 Class 1 參與的。甚至到了最後，只有 Class1 參與的才可以連通。

可以推知，當範圍更大時，有 Class 1 參與的，連通會越多。

(三) 使用 Class 1 裝置對於通訊路徑 hop count 的影響

通訊範圍不同的兩類裝置 Class 1 (100m)、Class 3 (10m) 根據 MixBlue 演算法，在同一範圍中運作，Class1 可以發揮其長距離傳輸的特性，讓通訊路徑的 hop count 減少。

圖 11 中，顯示在 150m*150m 的範圍中，模擬 30 個單位時間。總裝置數目為 500 個，其中一類全為 Class 3 的裝置，另外一類含有 50 個 Class1 的裝置做為比較的對象。

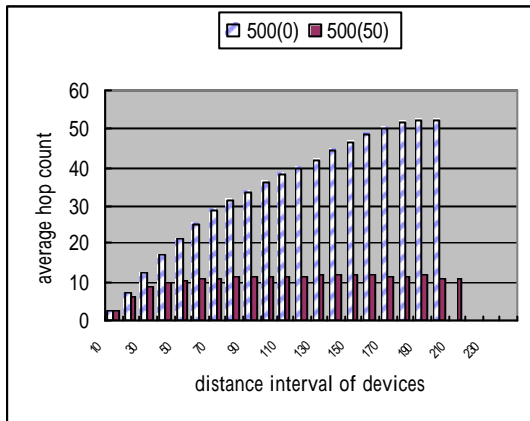


圖 11 使用 Class 1 裝置對於通訊路徑 hop count 的影響

由於直接計算所有的通訊路徑的平均 hop count 並不客觀，所以在 X 軸上區分以 10m 的間隔區間，Y 軸為路徑的 hop count，每一個長條表示所有直線距離落在該距離區間的兩個裝置，它們可通訊路徑的平均 hop count。

圖表中顯示，無論通訊的兩個裝置的距離為何，有 50 個 Class 1 參與的，其通訊路徑 hop count 遠低於沒有 Class 1 參與的。除此之外，可以明顯發現，兩個裝置距離超過 50m，有 Class 1 參與的，hop count 均維持在 10 個 hop 的左右，表示通訊範圍 100m 的 Class 1 所形成的散網(Scatternet)，可以當做通訊傳輸的骨幹，大幅降低傳輸的 hop count，提升傳輸的效率。

(四) 合併對 Piconet 數目的影響

藍芽裝置想要進行溝通就必須加入微網(Piconet)，所以微網是裝置溝通的最小單位，而散網(Scatternet)就是跨微網突破傳輸範圍限制傳輸的關鍵。執行合併是為了減少所形成的 Piconet 數目，此舉讓通訊傳輸更有效率。以下模擬範圍 200m*200m，模擬時間為 30 個單位時間。

圖 12 中的 X 軸表示裝置的數目，括弧中的數字代表包含多少個 Class 1 的裝置；Y 軸是 Piconet 的數目，以長條圖對應表示。MixBlue 所執行的 Merge 程序，不會降低連通率，卻可讓 Piconet 的總數下降，使傳輸路徑的 hop count 減少，以增進傳輸的效率。

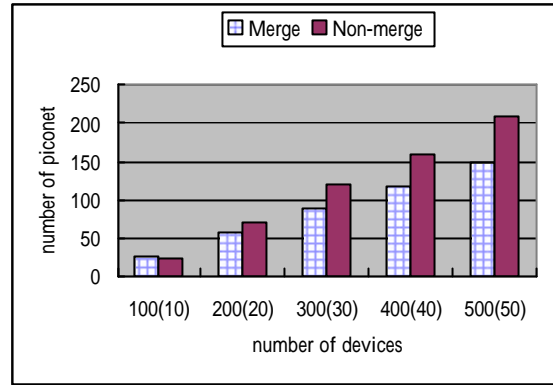


圖 12 合併對 Piconet 數目的影響

四、結論

藍芽規格書中，定義了 Class 1、Class 2、Class 3 三種不同等級的裝置，初期的藍芽產品都以製造 Class 3 的產品為主，現在 Class 1 的產品已經出現在市面上，而且有越來越普及的趨勢。而其相關的文獻探討都是以單一等級的裝置來發展，本篇論文所提出的 MixBlue 藍芽散網建構演算法，是在無特殊的環境假設下(沒有限制所有裝置在彼此通訊範圍，沒有限制裝置的個數...等等)，混合 Class 1 和 Class 3 兩種等級的裝置，建構出藍芽散網(Scatternet)。

在建構的過程中，加入 Piconet 合併的機制，此機制不但使整體 Piconet 數目減少，並同時提升了裝置的連通率。合併機制再加上保留第七個 Active Slave 的做法，讓本演算法即使在沒有 Class 1 參與的狀況下，連通率仍然比 BlueStar 高出許多。在有 Class 1 裝置參與的情況下，連通率更將大幅提升。

合併機制讓整體 Piconet 數減少，再加上 Class 1 的參與，發揮長距離傳輸的特性，使得傳輸路徑的 hop count 數目減少，使得傳輸更有效率。

MixBlue 是目前唯一提出混合兩種等級裝置的散網(Scatternet)建構演算法。此外，論文中所提的 Merge 程序，運作在沒有限制所有裝置在彼此通訊範圍的情狀下亦是首見。未來可以針對其中所提及的保留第六個 Active Slave 方法做更多的應用，或是在 MixBlue 所形成的散網架構下，進行繞送路徑建立、多重路徑...等方面的研究。

五、參考文獻

- [1] Bluetooth SIG , "**Specification of the Bluetooth System , Version 1.1**" , <http://www.bluetooth.com/>
- [2] Jennifer Bray and Charles F Sturman , "**BLUETOOTH Connect Without Cables,**" Foreword by Joe Mendolia of CATC
- [3] P. Bhagwat and A.Segall, "**A routing vector method (RVM) for routing in Bluetooth scatternets,**" Mobile Multimedia Communications. (MoMuC '99) IEEE International Workshop, pp.375 -379, Nov. 1999.
- [4] D. Groten and J. R. Schmidt, "**Bluetooth-based mobile ad hoc networks: opportunities and challenges for a telecommunications operator,**" Vehicular Technology Conference. IEEE VTS 53rd, Vol 2, pp.1134 -1138, May 2001.
- [5] J. C. Haartsen, "**Bluetooth-ad-hoc networking in an uncoordinated environment,**" Acoustics, Speech, and Signal Processing. Proceedings. (ICASSP '01). IEEE International Conference. Vol. 4, pp. 2029 -2032, May 2001.
- [6] J. C. Haartsen, "**The Bluetooth radio system,**" Personal Communications, IEEE [see also IEEE Wireless Communications], Vol. 7, pp. 28 -36, Feb. 2000.
- [7] C. Law and L. Y. Siu, "**A Bluetooth Scatternet Formation Algorithm,**" Global Telecommunications Conference. GLOBECOM '01. IEEE, Vol. 5, pp. 2864-2869, Nov. 2001.
- [8] T. Y. Lin, Y. C. Tseng, K. M. Chang, C. L. Tu, "**Formation, routing, and maintenance protocols for the bluering scatternet of bluetooths,**" System Sciences. Proceedings of the 36th Annual Hawaii International Conference. Pp.313-322, Jan. 2003.
- [9] T. Salonidis, P. Bhagwat, L.Tassiulas, R. LaMaire, "**Distributed topology construction of Bluetooth personal area networks,**" INFOCOM. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, Vol. 3, pp.1577 -1586, April 2001.
- [10] T. Sato and K. Mase, "**A Scatternet Operation Protocol for Bluetooth Ad Hoc Networks,**" Wireless Personal Multimedia Communications. The 5th International Symposium, Vol. 1, pp. 223 -227, Oct. 2002.
- [11] R. Shorey and B. A. Miller, "**The Bluetooth technology: merits and limitations,**" Personal Wireless Communications. IEEE International Conference, pp. 80 -84, Dec. 2000.
- [12] G.. V. Zaruba, S. Basagni and I. Chlamtac, "**Bluetrees-scatternet formation to enable Bluetooth-based ad hoc networks,**" Communications. ICC IEEE International Conference, Vol. 1, pp. 273 -277, June 2001.
- [13] B. Zhen, J. Park and Y. Kim, "**Scatternet Formation of Bluetooth Ad Hoc networks,**" System Sciences. Proceedings of the 36th Annual Hawaii International Conference, pp. 312 -319, Jan. 2003.