

# A GPS-based Quorum Hybrid Routing Algorithm in Ad hoc Wireless Network

\*I-Shyan Hwang, \* Chih-Kang Chien and \*\*Chiang-Ying Wang

\*Department of Computer Engineering and Science

Yuan-Ze University, Chung-Li, Taiwan, 32026

\*\*Department of Information Management

Transworld Institute of Technology, Douliu, Taiwan 640

E-mail: \*ishwang@saturn.yzu.edu.tw, \*kinghong@accton.com.w, \*\*ann@tit.edu.tw

## 摘要

本文提出一個在隨意式無線網路(Ad Hoc Wireless Networks)中全新的繞送演算法,名為以全球定位系統為基礎的集合式繞送演算法(GPS-based Quorum Hybrid Routing Algorithm),或簡稱 GPS-QHRA。在這個演算法中,我們把網路分割成若干個六角形區域(zone)。利用全球定位系統(GPS),使得網路中的每一個節點都知道它自己在那一個區域中,如此非常容易掌握節點的方位,因此特別適用於隨意式無線網路的環境。接下來,我們在每個區域中挑選出一個特別的資料庫節點(Location Database Node),用來模擬在蜂巢式系統中 HLR 的功能。每個資料庫節點負責收集它管區之內節點的繞送資訊。在 GPS-QHRA 中,數個資料庫組成一個有規則集合(Uniform Quorum System)。這些有規則集合兩兩相交且相異,當來源節點有封包需要傳送時,會先查詢它所屬的資料庫節點,之後該資料庫節點會選出一個有規則集合,再根據那個有規則集合中的資料庫節點來查詢。由於有規則集合兩兩相交,因此我們便能用局部(Local)資訊,來獲得整體(Global)網路中各節點的相對位置資訊。由模擬結果證明,本文所提出的演算法能夠改善同樣是利用 GPS 技術的區域式階層鍊結繞送(ZHLS, Zone Hierarchical Link State)演算法。ZHLS 演算法的區域是以正方形劃分,而我們所提的演算法則是以六角形劃分,因此區域跟區域之間可以從 4 個方向傳輸改為 6 個方向傳輸,從而在繞送時可減少封包的數量和縮短路徑的長度(number of hops);同時,在 ZHLS 演算法中並沒有叢集標頭(Cluster Head)的觀念,故每個節點都要維護繞送資訊。在 GPS-QHRA 中,利用資料庫節點作中央管理,負責維護每個區域(Zone)之間的繞送資訊,因此可減少多餘封包的產生,從而減少網路負載。且當來源節點要傳送封包給在區域之外的目標節點時,在 ZHLS 演算法中,必須發出一個位置搜尋封包廣播給網路中所有其他的區域來找出目標節點的位置。而 GPS-QHRA 由於具備有規則集合的觀念,因此只需廣播給有規則集合中的資料庫節點即可。此外,我們還證明了 GPS-QHRA 所產生的網路負載、平均路徑長度都較 ZHLS 為佳。

關鍵字: 隨意式網路(Ad Hoc Wireless Network), 繞送演算法(Routing Algorithm), 全球定位系統為基礎的集合式繞送演算法(GPS-QHRA), 有規則集合式系統(Uniform Quorum System)

## 1. 簡介

隨意式無線網路(Ad Hoc Wireless Network),由許多可移動的節點或路由器(Router)所組成,其優點為可快速的建立網路連線,省略佈線的麻煩,因此在沒有佈線的區域或災難環境中十分有用。雖然隨意式無線網路可以快速建立一個網路通訊系統,但由於節點可以到處移動,而且沒有像傳統無線網路中基地台(Base Station)或行動服務中心(Mobile service center)這樣的機制,因此沒有固定的網路拓樸(Topology)和中央控制,在傳送資料時的路徑選擇便不能直接套用例如像最短路徑等的方法來傳輸,必須有一套專有的繞送演算法(Routing algorithm)[1]來達成。例如可以架構來區分的階層式(Hierarchical)與平面式(Flat)兩種[2]。階層式即是把節點分成一個個的叢集(cluster),每個叢集再透過一些特定的節點,叫做叢集標頭(Cluster Head)來管理自己所屬的叢集;而平面式則將網路上每個節點都視為相同,並沒有階層的區分,研究[3]發現利用階層式的方法會有更好的效能。而就選擇路徑的演算法,我們可大致分為 Table Driven 和 On-Demand 兩種。所謂的 Table-driven 繞送演算法,就是網路中的每一節點定時的利用廣播(broadcast)來收集節點和節點間的路徑資訊,要求每個節點要有一個以上的表格來儲存繞送資訊,而且要在網路的拓樸改變時隨時更新這些資訊。如 DSDV [4]、CGSR [5], ..等等。On-Demand 則是需要傳送時才開始找尋路徑,當來源節點需要傳送資料時,才會送出一個路徑搜尋封包(routing discovery packet)來建立繞送路徑,如 AODV [6]、TORA [7]、Communication-Aware based algorithm [8], ..等等。這兩種方法各有優、缺點。Table Driven 的優點是當需要傳輸時,便可馬上開始快速的傳輸,但其缺點是必須定期更新繞送表,而要付出使網路的流量較大的代價。而 On-Demand 則只有在通訊時才把路徑找尋出來,因此可節省定期更新所付出的網路資源,但是每一次傳送資

料時，都必須重新搜尋路徑。也因為如此，近期有一些 Hybrid 的演算法提出，如[9]中的 ZHLS (Zone Hierarchical Link State)演算法。ZHLS 是一應用全球定位系統 (Global Positioning System(GPS)) [10]、[11]的繞送演算法。它把網路分為若干個正方形區域(zone)，在區域之內使用 Table-driven，而在區域之外的節點則使用 On-Demand。這類 Hybrid 的演算法，能對 On-demand 和 Table driven 演算法的優、缺點作一擇中的方案，在整體效能上能獲得改善。另外，在隨意式無線網路的相關研究中，還提出了一種名叫有規則集合式系統(UQS, Uniform Quorum System)[12]，其觀念是動態地找適當節點來當 HLR。但是 UQS 並沒有使用 GPS 技術的協助，因此在存取或更新資料庫時，對於節點的方位路徑，並不能作完善的掌握，因而產生多餘的傳送封包。在[9]中的 ZHLS 演算法中，是一個應用 GPS 的繞送演算法。在這個演算法中，雖然是階層式的架構，但是並沒有叢集標頭，每個節點都要有二個 Table，一個是記錄和自己同區域的其他節點的資訊，另外一個是區域和區域之間的表格。當節點在自己的區域之內時，便利用第一個表格的資訊把封包傳輸出去，而在自己區域之外時，則要 On-Demand 的廣播給所有其他的區域，找出目標節點的所在區域位置，再把封包傳輸給該區域。但由於此演算法並沒有集合的觀念，因此其所需傳輸的封包一定比有規則集合架構上來得多。因此，本文利用 GPS (Global Positioning System)的技術和有規則集合系統(quorum system)的特性，提出一個全新的演算法，稱之為 GPS-based Quorum Hybrid Routing Algorithm (GPS-QHRA)。這個演算法主要有三部分，一個是 GPS-based 繞送演算法，第二個個是維護當資料庫節點移動時，必須找別的節點來取代其地位的移動性管理策略 (mobility management scheme)。第三個是在一些特殊狀況下的容錯機制。本文的架構如下，第一章的緒論中，介紹隨意式無線網路的架構、研究重點、研究動機和目的。第二章相關研究，描述一些已有的繞送演算法，分別介紹 Table-driven, On-Demand, Hybrid 和有規則集合式系統的演算法。第三章為本文提出的隨意式網路之 GPS 基礎集合式繞送演算法 (GPS-Hybrid Quorum Routing Algorithm(GPS-HQRA))，描述此演算法的環境、原理、以及繞送原則。第四章則為程式模擬，利用路徑長度(Path Length)，移動性的影響(mobility effect)，和找尋路徑的負載(Location Search Overhead)等參數，來與 ZHLS 作效能上的比較，從而證明本文的演算法較 ZHLS 佳。最後是本文的結論與未來展望。

## 2. 相關研究

在隨意式無線網路網路中，在結構上分為階層式與平面式兩種。所謂階層式，即把網路的拓

樸分成若干個叢集，在叢集中以叢集標頭來管理，而平面式則每個節點都可視為相同的功能，很多的研究中證實階層式的管理有較佳的效能。而在繞送的原則上分為三種，一種是 Table Driven，第二種是 On-Demand，而最後一種則是結合以上兩種的 Hybrid。現有屬於 Table driven 的繞送演算法有很多如叢集標頭閘道分送 (Clusterhead Gateway Switch Routing(CGSR))演算法[5]，它為一階層式的繞送演算法，選擇一些特定的節點來做叢集標頭，負責記錄它叢集之內成員的資訊，而不用每個節點都記錄繞送表 (Routing Table)，如此便能減少網路中的負載。但卻造成選取叢集標頭或叢集標頭的變更時的系統負載。CGSR 為了減少在選取標頭時的負擔，它使用了一個叫最少叢集改變演算法 (Least Cluster Change clustering algorithm (LCC))，利用 LCC，叢集標頭只有在兩個叢集標頭需要通訊時或其中一個節點離開叢集標頭的範圍時才使用，再重新安排叢集標頭。當一個節點要傳送資料時，它會根據它的繞送表。先把封包傳到叢集標頭，之後叢集標頭便看看目的節點是否在其叢集中，若不是，便傳到一些閘道節點(Gateway node)。屬於 Source-initiated On-Demand 的繞送演算法也有很多，如：節點互動式繞送演算法 (Communication-Aware Mobile hosts Algorithm) [8]。在一般的繞送演算法中，通常是找一最短路徑。但是其缺點是本來存在的路徑可能因節點的移動而消失，而使整個系統因處理這些問題而使負載加大。節點互動式繞送演算法，是一個建立在節點間互動的演算法。節點互動式繞送演算法的基本原理是每個節點必須知道其他節點的動向，而透過這些動向的資訊，我們的演算法便能找出最穩定的路徑來傳送資料。而不是找最短的距離。這樣在傳送時便會大大減輕連接失敗(link fault)的情況發生，其中最主要是用到親屬度 (affinity)的參數來判定。Table-Driven 在維護繞送表方面，無論是否需要用到該路徑，每一節點或在叢集標頭節點都需要一個完整的繞送表。而 On-Demand 則只有需求在繞送時才需要繞送表的資訊。在繞送架構方面，On-Demand 大都是平面型態，而 Table-Driven 則會有階層式的結構。表 1 為 On-Demand 與 Table-Driven 的比較。從表 1 中我們可以看到，On-Demand 與 Table-Driven 在繞送上各有其優、缺點。然而，區域式階層鍊結繞送 (ZHLS)[9]演算法，為一利用 GPS 技術的 Hybrid 演算法，因為他結合了 On-Demand 與 Table-Driven 兩種繞送的特性，所以 ZHLS 才被稱為 hybrid 演算法。不像 Table-Driven 演算法，每個節點都要記錄到網路中所有其他節點的路徑，ZHLS 只需記錄區內節點的路徑和一到區外的繞送表即可，可以減少網路封包的流量。不同於 On-Demand 的是，不是傳送給所有節點都要發出路徑搜尋封包找出路徑，若節點在自己的區域內即不用，可加快繞送的速

度。這就是所謂擇中的原則，從[14]中有它們之間的比較表，整體而言，其效能整體上比起前兩者有較佳的表現。

參數	On-Demand	Table Driven
繞送資訊的維護	有需要時才要	必須常常維護
繞送架構	平面式	大部分階層式
定期更新路徑	不需要	需要
QoS	需然大部分使用最短路徑但很小能提供 QoS，	使用最短路徑來支援 QoS
網路流量	隨著 mobility 而增加	比 On-Demand 來得大
如何處理 mobility	利用 route discovery	通知其他節點來維護定期的繞送表

表 1: On-Demand 與 Table-Driven 比較

有規則集合式系統，就是在隨意式無線網路中，挑選出一些特定的節點(通常挑選叢集節點或骨幹節點)，來模擬在一般如 GSM 等中的 Home Location Register(HLR)的功能。在 GSM 中，HLR 的主要功能是用來記錄有那些節點(Mobile)在這個基地台中註冊，當一個節點要和另一個節點通訊時，首先要發射訊號給它的基地台，接下來再透過 HLR 來看看要通訊的節點是在那一個位置，並發出通訊要求給該節點，之後再能再進一步的建立通訊。在 Ad hoc 無線網路環境中，並沒有基地台等固接式的機制，而且節點是會移動的，因此我們必須動態的挑選一些節點來模擬 HLR 的功能，而由於節點會移動的，所以當 HLR 節點移開時，我們必須挑選其他節點來繼承它的功能，

### 3. GPS-QHRA

本章介紹我們新提出的 GPS-QHRA 演算法，此演算法結合了 GPS 技術與 UQS 系統觀念。利用 GPS 技術，我們可以把網路地圖分割成許多的區域，在每個區域裡，動態挑選一些節點來模擬 UQS 中方位資料庫的功能，剛好符合 UQS 的特性，而且由於有了 GPS 的協助，我們更容易掌握節點的動向，當一節點離開自己所屬的區域時，我們只需發射訊號給 GPS 衛星，在跟區域地圖比對後，馬上便可知道節點已離開原來所屬的區域。因此，在方位資料庫節點快要離開時，我們可提前的挑選一個候選人節點來繼承方位資料庫節點，使其方位資料庫在不可使用狀態(inaccessible)的機率降至最低，對於整個 UQS 系統觀念的建構，十分容易且簡單的幫助。和 ZHLS 相比，同樣是階層式的結構，這裡卻有叢集標頭節點(方位資料庫節點)，因此不用每個節點都記錄同樣的繞送資訊，減少網路上重覆的資訊。而

且對於方位資料庫節點，由於只需記錄自己區域內的繞送資訊和有規則集合中其他節點的位置，因此也不會造成方位資料庫過多的負載。另外，在節點需要傳送資料給一在目前區域以外的節點時，由於有有規則集合的觀念，因此並不需廣播給所有其他的區域，只需局部廣播(multi-cast)給有規則集合中的其他方位資料庫節點即可，也可減少多餘的封包在網路上充斥著。

#### 3-1 環境參數

GPS-QHRA 假設在這個隨意式無線網路環境中，有下列預設條件。

- (1) 每一個節點都有唯一的 ID。
- (2) 每一節點都有一 GPS 的接收器與發射器，當其發射訊號給 GPS 衛星後，在得到 GPS 衛星的回應後，便可得到一個二維座標，接著便可從區域地圖(Zone map)中得到自己所在的區域位置。
- (3) 每一個節點都有一同步時鐘，用來分辨封包的先後。
- (4) 每一個節點的發射功率是一樣的，即若節點 A 能收到節點 B 的訊號，節點 B 也能收到節點 A 的訊號。且不考慮訊號干擾而造成的訊號衰減(Fading)問題。
- (5) 有一個有效的頻寬管理的方法，因此不須考量碰撞或頻寬分配的問題。
- (6) 假設在網路中的節點有足夠的多，因此在任一時刻在一個區域中完全沒有節點的可能性很低。但若真的發生，就要啟動 GPS-QHRA 容錯法則。

#### 3-2 GPS-QHRA 原理介紹

在開始介紹 GPS-QHRA 演算法的繞送原則之前，我們必須先定義區域地圖的設計，區域地圖對應表(Zone Map table)，以及有規則集合參數的定義等。

##### ● 區域地圖(Zone Map)的設計

首先，我們先把整個地圖分割成一個個的區域(Zone)，至於區域數的多寡，即為設計 UQS 時(n, q, k, m, r)中的 n 參數，或由網路中節點的多寡，傳輸半徑等不同的因素來設計。由於在這個 GPS-QHRA 中我們給予每一個區域一個方位資料庫，因此，必須限制一個區域不能太大，以避免方位資料庫離開其管轄區域之內的節點太遠，這裡的假設大約是一個傳輸直徑左右。如圖 1 所示，而真正的大小可藉由模擬結果來獲得。另外，和 ZHLS 不同的是，我們把區域劃分為正六角形，在[13]的研究中有證明，六角形的分割方式比正方形的分割方式好。像和蜂巢式(cellular)一樣，而不是正方形，如圖 2，把區域分割為六角形的好處是，我們在通訊時不只可以朝四方傳輸，而是可以朝六個方向來傳輸，這樣可避免在選擇路徑上多繞一些不必要的節點。在決定完區

域的多寡後，我們便把網路中的每一個座標，定義好一個區域，然後存放成一個區域地圖對應表 (Zone Map Table)。以便當節點向 GPS 衛星得到一個座標後，再對應區域地圖對應表，馬上知道自己目前在那一個區域中。另外，圖中的虛線部分，定義為方位資料庫危險區域，當方位資料庫節點移動到此區域時，代表隨時有可能離開這個區域，因此必須重新挑選別的節點當方位資料庫，後面章節會詳盡的說明其做法如何。而我們必須要把圖中虛線部分建構成表格，供方位資料庫查詢，以便知道它是否已經移動到此一區域。

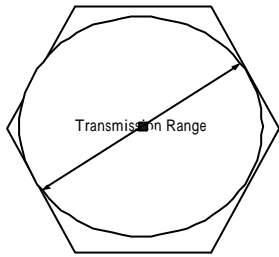


圖 1: GPS-QHRA 區域  
之內的傳輸直徑

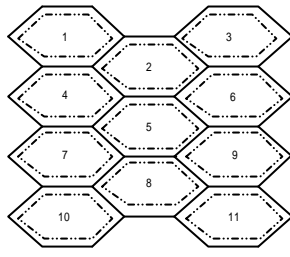


圖 2: GPS-QHRA  
區域地圖

- 有規則集合的定義

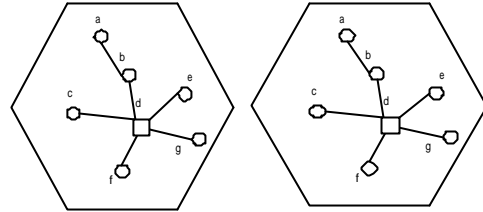
我們利用模擬結果來決定網路節點數多寡以及地圖大小，從而得到最佳的有規則集合參數，後就可決定集合的大小及相交性。

### 3-3 GPS-QHRA 協定

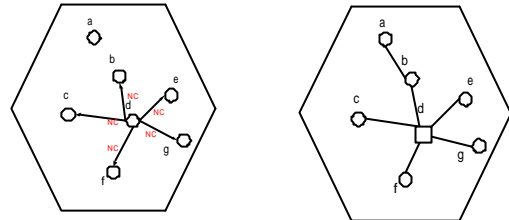
- 挑選方位資料庫

我們利用叢集(cluster)選擇標頭的概念為每個區域選出一個節點來充當方位資料庫，本文利用最大連結度的方法，因為由[15]可知選擇最大連結度的效能比選擇最小編號好。因方位資料庫是用來記錄其他節點的位置，連接度最高的節點便能以最小的 Hop 數，把其區域間的所有其他節點都記錄下來，可以減少路段繞送的距離。網路上的每一個節點都要先廣播一個**鄰居搜尋封包** (Neighbor Found packet(NF) )<node ID, Zone ID> 給它所有的鄰居。以便知道它自己的連接度(或鄰居數)，圖 3(a)中，以 d 節點為例，分別廣播給節點 b, c, e, f, g。接著，當一節點收到一個 NF 封包時，便回應一個<node ID, Zone ID> 給發給它 NG 封包的節點，如圖 3(b)所示，節點 b, c, e, f, g 回應給節點 d，這時節點 d 便知道它的連接度為何了，在這例中連接度為 5。在知道了自己的連接度後，便會再發出一個**連接度通知封包** NC(Neighbor Connectively packet)給它的鄰居，告知它鄰居此節點的連接度為何，如圖 3(c)所示。之後，此一節點便會啟動一個計時器(Counter)，等待它的鄰居把 NC 封包傳給它，以便知道其他節點的連接度。在所有節點都執行完同一步驟後，最後每個區域之間便能選出其連接度最高的節點來當方位資料庫了，如圖 3(d)所示。節點 d

的連接度最高，因此被挑選為方位資料庫節點，以正方形表示之。但請注意，由於在危險區域內的節點不能成為方位資料庫節點，因為在此區域中的節點隨時會離開，因此必須避免選在此區域中的節點為方位資料庫節點。我們可以把在危險區域中節點的連接度設為 0 即可避免挑選出這種節點。



(a) 節點 d 廣播 NF 封包 (b) 當收到 NF 封包時，  
鄰居 Reply 給節點 d



(c)當收到所有鄰居回 (d)最後節點 D 會被挑為  
應後，再把一 NC 封包 方位資料庫節點  
<d node, d conn#>廣播  
給它的鄰居

圖 3: GPS-QHRA 挑選方位資料庫方法

GPS-Database Formation Algorithm 的演算法如下:

#### GPS-DFA

For each node x in the network, broadcast a NF packet <nodeID, zoneID> to its neighbors

If x's node Location is within the Database Danger Region (DDR), then

Set conn#=0

Once receive all reply, then broadcast a NC <nodeID, conn#> to its neighbors,

count=conn#; maxconn#= conn#

databaseID = x's nodeID

DO While count! =0

Wait for any neighbor NC packet;  
count=count-1;

if neighbor's maxconn# > x's  
maxconn# and neighbor  
zoneID=x's ZoneID then  
maxconn#= neighbor's  
maxconn#

databaseID= neighbor's ID

end if

LOOP

End if  
Node x with database = databaseID

● **GPS-QHRA 繞送表格的產生**

在我們的 GPS-QHRA 演算法中，共需要 3 種繞送表格。第一種是區內繞送表，第二種是局部區外繞送表，不同於 ZHLS 的是，這個繞送表只需記錄其有規則集合中的區域便足夠了。而第三種則是有規則集合中的資料庫節點成員位置表。(1)區內繞送表的產生：當跑完 GPS-DFA 後，此時在每一個區域內都已經有一方位資料庫節點了，如圖 4 所示。這時方位資料庫節點要開始收集它區域之內其他節點的繞送資訊了，同樣的，可利用類似 GPS-DFA 中廣播的方式，方位資料庫節點便能知道它的區域中有那些節點。以及它和那些區域有連接。由於我們之前在規劃一個區域的大小時，是大約在一個傳輸直徑左右，因此我們可以以下面的定義，很容易的把繞送表格建立。同一個區域之內，共有三種類型的節點，(i)方位資料庫節點，(ii)直接節點(Direct Node): 指可直接與方位資料庫通訊的節點，(iii)間接節點(In-Direct Node): 指不能直接與方位資料庫通訊的節點。在圖 5 中，節點 e 為方位資料庫，節點 b、c、f、g、h 為其一個 Hop 可到達的節點，我們稱之為直接節點 而節點 a 跟 d, 在節點 e 的傳輸範圍之外，因此，必須透過其他節點才能到達節點 e。此種節點稱為間接節點。方位資料庫節點會定期的監視它管區節點的動向。方位資料庫建立繞送表格的程序如下，首先，它會先把所有的直接節點找出，接下來，它會把它的直接節點清單廣播給它的直接節點，之後，直接節點會再發出一個間接節點搜尋封包(Indirect Node probe packet)INP 去尋找此區域內的間接節點，那些不包含直接節點清單中的節點便是間接節點了，這時找到間接節點會再傳送間接節點資訊給方位資料庫，由於我們之前規定一個區域大小為一個傳輸直徑，因此，最多只能兩個 Hop，故一定可以把所有節點都能包含到。當然，有可能會有兩個直接節點同時找到同一個間接節點，此時，方位資料庫節點便可根據那一個直接節點較接近該間接節點來決定如何建立繞送表(距離可由 GPS 座標算出)。以此同時，也順便會把閘節點(Gateway node)同時記錄，以便在區外繞送表中使用。由圖 5 中產生的區內繞送表如表 2。

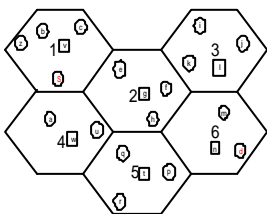


圖 4: GPS-QHRA 整體拓樸

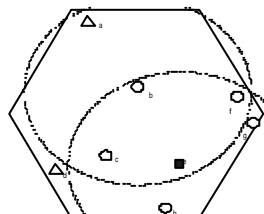


圖 5: GPS-QHRA 區域內拓樸

B	B	Direct
C	C	Direct
D	C	Indirect
F	F	Direct
G	G	Direct
H	H	Direct

表 2: 圖 5 的區內繞送表

目標區域	下一個區域	下一個節點
2	2	e
3	2	e

表 3: 從圖 4 可能建立的局部區外繞送表

(2)區外繞送表的產生：類似 ZHLS 的做法，這裡我們也要記錄一個局部區外繞送表，記錄到其他集合中的區域的繞送表。由於在找尋區內繞送表的同時，我們有把閘節點記錄下來，因此方位資料庫知道它與那些鄰近的區域有交接，且可以知道透過那些節點來傳送。這時，閘節點負責把自己和那些區域有連接局部廣播(multi-cast)給有規則集合中的資料庫節點，最後以最短路徑產生的區外繞送表如表 3 所示。這裡和 ZHLS 不同的是我們只需記錄有規則集合中的區域就可以了。(3)節點位置表：最後一個我們雖要記錄的表格是有規則集合中其他方位資料庫間的其他節點。以圖 5 中的有規則集合為例，假設我們定義的有規則集合為{(1, 2, 3), (1, 4, 5), (2, 4, 6), (3, 5, 6)}。在建立完區域之內的繞送表後，我們會把自己區域內有那些節點的資訊，傳送給一個有規則集合中的所有資料庫，假設選出的是(1, 2, 3)。則方位資料庫 1 利用局部繞送表局部廣播 (multi-cast) 給方位資料庫 2 和方位資料庫 3 它區內節點的資訊，在所有方位資料庫都做完了以上步驟後，在方位資料庫 1 中，便有可能記錄了如表 3 的節點位置資訊。

Zone	Node Resident
1	b, c, v, s, z
2	
3	I, j, k, l

表 3: 從圖 5 所建立的路徑資訊表

● **GPS-QHRA 繞送法則**

在建立完所有的繞送表以後，任一來源節點便有足夠的資訊把封包繞送到它的目標節點了，我們分兩個情況討論。情況 1 是當來源節點跟目標節點在同一個區域時的情況，此時採用 Table-Driven 的方法。每一來源節點在繞送前，會先跟它所屬的方位資料庫節點查詢，若發現目標節點就在自己的區域中，它就會從方位資料庫節點的繞送表中找出路徑，並把封包傳送出去。情況 2 是當來源節點跟目標節點在不同區域時的情況，因來源節點在繞送前，會先跟它的方位資料庫節點查詢，若發現目標節點不在同一區域時，此時，它會選擇其中一個有規則集合，如[1, 2, 3]。利用局部區外繞送表，分

Destination	Next Node	Remark
A	B	Indirect

別去這三個方位資料庫節點去查詢目標節點所在的區域，由於有規則集合兩兩相交的特性，因此必定有一區域知道目標節點所在的區域。在知道了區域後，首先把封包繞送到該區域的方位資料庫中，再透過方位資料庫節點的繞送表，便能把完整的路徑找出，此時再把封包送出。在這個演算法中，由於有規則集合的觀念，因此不像 ZHLS 中必須廣播給所有其他的區域，才能找到目標節點所在的區域，因而比 ZHLS 更有效率。所以我們採用 On-Demand 的方式，因此本演算法是一個 Hybrid 的演算法。

● **GPS-QHRA 位置更新法則**

在隨意式無線網路中，節點是會隨時移動的，因此我們必須定時的檢查繞送表，才能確保其資訊是最新的。在 GPS-QHRA 中，定義了兩種位置更新法則，分別是定時更新與當傳送時更新。(i)定時更新：由於節點會隨時移動，所以資料庫節點會定時檢查它所屬區域內的節點與其繞送路徑，同樣也會定時的通知集合中的方位資料庫它區域內有那些成員。(ii)傳送時更新：當節點要傳送封包時，首先藉由 GPS 檢查自己是否已離開本來的區域，若該節點發現自己已離開原來所屬的區域，它必須要先跟新的方位資料庫節點註冊，然後方位資料庫節點再把新的成員路徑資訊，記錄在其繞送表中，並把新的成員位置，選出一個有規則集合，通知其他的方位資料庫節點。

● **方位資料庫節點重新挑選法則**

由於方位資料庫節點也會移動，因此，若一方位資料庫節點移動，而且離開其區域時，此時一個區域內便沒有方位資料庫節點，這樣我們的演算法便會出現問題，會產生該區無人管理的情況出現。因此，為了避免這種情況出現，我們之前在區域地圖中所定義的方位資料庫危險區域便能用上了，當一方位資料庫節點移動到該區域的危險區域時，此時，重選法則便會啟動，我們必須重新挑選一個繼承人，把該方位資料庫節點有的資訊傳給它，如圖 6 所示，方位資料庫節點為節點 D。在定期更新法則中，當一方位資料庫節點在定期檢查時發現它已經到達了方位資料庫危險區域，這時它會發出一個方位資料庫選擇封包(Database Choose Packet)通知它區域內的所有節點。在經過一段時間後，此區域內的所有節點便會 Reset，重新啟動挑選方位資料庫的程序。這個程序和 DFA 是一樣的，在新的方位資料庫節點選出後，新的方位資料庫會去通知舊的方位資料庫，把它之前的有規則集合資訊告訴它，至於區域內的繞送資訊則必須重新計算，在計算完後再把成員資訊局部廣播給它有規則集合中的會員。

● **GPS-QHRA 容錯法則**

在某些情況上，某些節點找不到它的資料庫節點，但其實它的資料庫節點是存在的。例如當節點 a 離開了其資料庫節點所能包含的傳輸範圍，我們定義這種節點為游離節點。由於游離節點找不到資料庫，因此它跟網路孤立了，即使有路徑存在，卻不能跟別的節點通訊。當游離節點需要傳輸時，發現找不到任何資料庫，在經過一段時間後，便會啟動

容錯法則，這時游離節點會重新啟動 DFA 挑選資料庫節點。如圖 7 所示，節點 b 會被挑為新的資料庫。並通知其他有規則集合，做法和之前資料庫節點一樣，在這一時刻，區域一會同時存在二個資料庫節點，不過這對整個演算法的運作並沒有影響。在後來若因節點的移動性，使新與舊的資料庫再次連接，此時當它們發現有二個資料庫時，便會從中選一個當資料庫，選擇法則以連接度判斷。雖然在此時刻，在一區域存在二個資料庫時，可能導致封包流量的增加。但是由於網路節點若夠多，我們假設出現這種例外的出現機率不高，因此對整個 GPS-QHRA 效能上的影響並不會很大。

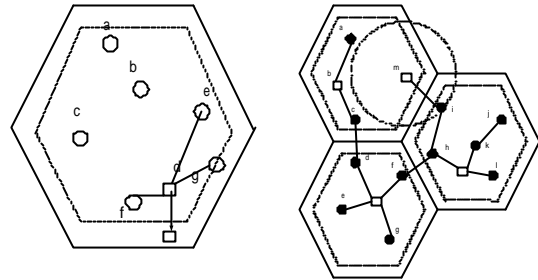


圖 6: GPS-QHRA 資料 庫節點重選法則 圖 7: 節點 b 成為新的 資料庫

**四、 模擬評估**

本章將 GPS-QHRA 與階層式，Hybrid 的 ZHLS 繞送演算法作比較，之所以選擇與此演算法作比較，是因為它同樣是利用了 GPS 的技術觀念，在網路的環境可以說是一樣的。在 ZHLS 中，區域是以正方形劃分，而我們所提的演算法則是以六角形劃分，在[13]中有證明六角形的方式的路徑長度會較短，從而獲得較佳的效能，而我們也將會在模擬中針對平均路徑長度、LSP 封包產生數量、移動性影響、搜尋路徑負載、封包漏失機率分別做討論。

**4-1 模擬環境**

本文模擬 ZHLS 與 GPS-QHRA 的效能，至於網路環境，為了配合傳輸半徑的大小使得傳輸直徑的大小是接近一個區域的大小，在 GPS-QHRA 中，我們分別設計了二個模擬環境，分別是傳輸範圍為 200 以上的網路環境，共有 6 個區域的圖 8 以及專為傳輸範圍為 100 以上，共 25 個區域的圖 9。

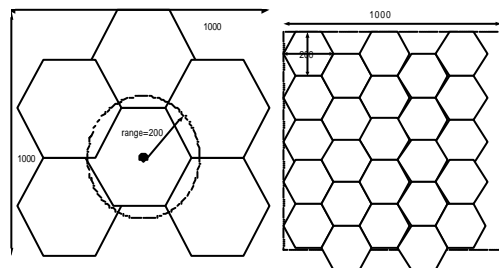


圖 8: GPS-QHRA 的模 擬環境 圖 9: GPS-QHRA 網 路環境

**4-2 模擬參數**

平均路徑長度:我們選擇以 50、100、150、200、250、500 個節點，傳輸半徑 100、200、250 和 300 來比較

利用 GPS-QHRA 和 ZHLS 中平均路徑的長度，在 GPS-QHRA 中我們用了二種環境來比較，分別是圖 8 和圖 9 的環境，一個是分成 6 個區域的，另外一個是分成 25 個區域。我們選定一個節點，然後分別對網路的其他節點作路徑的搜尋，我們模擬了 2000 個單位時間，扣除了漏失的路徑，將得出的路徑長度作平均值，如圖 10 的結果。從圖中可以發現，節點數為 50 時，在同樣的傳輸範圍下，所得出的平均路徑長度較長，而在節點數為 100、150、200、250、500 時，得出的平均路徑沒有太大的差別，那是由於當節點數較少時，所能選擇的路徑不多，因此在選擇最短路徑時因節點數不多而要選出較長的路徑，因此得出的平均路徑會較長。而在節點數為 150 到 500 時，由於網路中的節點數已較多，因此對於整體得出的平均路徑不會有太大的影響。另外，我們可以在圖中發現，在傳輸範圍變大時，所得出的路徑會明顯的減少，那是由於傳輸範圍愈大，所能選擇的路徑愈多，因此平均路徑會縮短。另外一個值得我們留意的是，在使用 GPS-QHRA 環境(25 個區域)會比使用 GPS-QHRA 環境(6 個區域)好一些，那是由於把網路細分後，得出的路徑一定比大部的分解來得好，不過在後面我們將會看到，把區域分得愈多，其產生的 Node LSP 也愈多。從各方面的數據顯示下，GPS-QHRA 所得到的平均路徑都為 ZHLS 來得短。另外，從上一節中我們可以清楚的看到，利用我們的 GPS-QHRA 會得出較短的平均路徑長度，然而，為了認證 GPS-QHRA 之所以會得出較短的路徑是和六角形有關，在這節中，我們分別模擬 GPS-QHRA、ZHLS 和以六角形切割區域的 ZHLS，在不同節點，不同傳輸範圍時路徑長度的差別。從圖 11 中我們可以發現，利用六角形切割區域的 ZHLS，所得出的平均路徑長度和 GPS-QHRA 所得出的差不多，因此我們可以證明出，我們的 GPS-QHRA 之所以會得出較短的路徑長度，是因為利用六角形切割的關係，和 Quorum 的觀念並沒有什麼影響，不過，後面我們將會看到，利用 Quorum 的觀念，雖然對路徑長度沒什麼影響，卻可以減少系統的負載。



圖 10: GPS-QHRA 和 ZHLS 平均路徑的比較

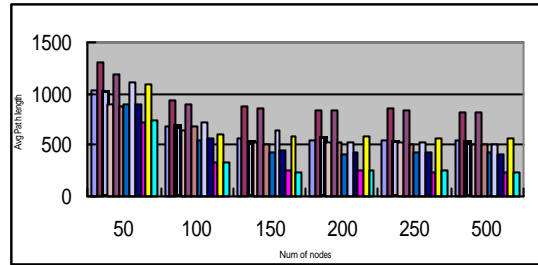


圖 11: 六角形對路徑長度的影響

● **LSP 封包數量的比較:**在 GPS-QHRA 和 ZHLS 的演算法中，為了能找到傳送的路徑，如前面章節所說，分別是利用二種封包來達成，Zone LSP 和 Node LSP 在此節中我們分別以節點數 50、100、150、200、250 和 500 來模擬 GPS-QHRA 和 ZHLS 所產生的 LSP 數目。在 ZHLS 中，共有 6 個區域，由於在某一區域內，每個節點都會對在同區域內的其他節點廣播其 Node LSP，因此在節點很平均的分配之下，所預測產生的 Node LSP 為  $(\text{節點數目}^2) / \text{區域數目}$  而在 GPS-QHRA 中，我們有二個模擬環境，分別是環境 a(6 個區域)和環境 b(25 個區域)，在這二個環境中，只有資料庫節點需要產生 Node LSP，我們利用如 3-3-2 節直接節點和間接節點的方式，來產生 Node LSP，並在區內的 Node LSP 產生完後，透過 Gateway 節點去通知其集合中的資料庫節點它的 Node LSP，整體上，其產生的 Node LSP 會明顯的比 ZHLS 所產生的來得小 而在 Zone LSP 方面，以 GPS-QHRA 環境(a)和 ZHLS 來言，其所產生的 Zone LSP 並沒有太大的差別，不過，從表 4 我們可以看出，切割成 25 個區域的 GPS-QHRA 環境的 Zone LSP 較多，且在節點數為 50 和 100 時，其總 LSP 數甚至比 ZHLS 來得多，不過，在節點數變多時，ZHLS 的 Node-LSP 成長較快，因 ZHLS 的 Node LSP 是以  $(\text{節點數目}^2) / (\text{區域個數})$  來成長，在全部的 LSP 數量的比較下，GPS-QHRA 明顯的產生較小的 LSP 下表 4 比較出 ZHLS 和 GPS-QHRA 在不同節點下所產生的 LSP 數量 而圖 12 則是以其整體的 LSP 數量來作比較

No of nodes	ZHLS			GPS-QHRA (a)			GPS-QHRA (b)		
	Zone LSP	Node LSP	Total LSP	Zone LSP	Node LSP	Total LSP	Zone LSP	Node LSP	Total LSP
50	298	423	721	287	276	563	1198	298	1496
100	643	1819	2462	632	598	1230	2432	1021	3453
150	1135	3998	5133	1209	1327	2536	3672	1624	5296
200	1366	6989	8385	1450	2454	3904	4968	3212	8180
250	1731	10621	12352	1722	3906	5628	6309	4320	10629
500	3023	41732	44764	3012	4327	7339	12831	7212	20043

表 4 ZHLS 和 GPS-QHRA 產生的 LSP 的比較

#### 4-2-3 移動性的影響

在這節中，我們模擬當節點移動時，其對網路所產生的負載，我們以移動速率為 1、5、10、20、40 單位/cycle，分別以節點數為 50、100、150、200、250 和 500 節點，隨機分佈在此區域中來模擬。我們假設節點是以一種固定的方式行走，再經過一段 x 時間後，再隨機的選出一個角度，然後再走 y 距離，之後再重覆上面步驟，而當節點走到網路的邊緣時，則會再隨機的轉另外一個方向移動。我們以 10 個 cycle 來模擬，看看在移動前跟移動後的變化，以 Zone Lsp 為例，即移動後網路拓樸的變化數量，在表 5 中我們可以看到，當節點數變大時，其拓樸的改變則是變小，這是由於節點數變多時，網路的區外拓樸，即 zone LSP，不會有太大的改變。而以 Node LSP 來觀察，則節點數變多時，其區內的拓樸，即 Node LSP 也跟著變多。最後，我們依比例算出其改

變的百分比，再依每個節點因為此改變而所需產生的額外 LSP 算出，最後我們以圖 12 把其整體的負載作比較，在 6 個區域的 GPS-QHRA 環境中，不論節點的多寡，都明顯的比 ZHLS 來得好，而 25 個區域的 GPS-QHRA 環境中，在節點數為 150 以上時，也漸漸的比 ZHLS 好，在整體上，我們可看出 GPS-QHRA 有較佳的表現

接著我們再比較不同的移動速率，ZHLS 和 GPS-QHRA 效能上的比較，從圖 13 到圖 17，可明顯的看出，切成 6 個區域的 GPS-QHRA 環境負載最小，接著是 25 個區域的 GPS-QHRA 環境，最後才是 ZHLS，而當移動速率很小時，其負載也很小，三種演算法的差異不大，不過，在高速的環境中，ZHLS 在節點數增加時的成長很大，因此我們得出了在速度變快且節點數變多時，我們所提出的演算法在移動的影響上有較好的抗衡性的結論。

# of ZHLS							
Zone			Node			Total	
# of LSP	% of Zones generating LSPs	Overhead	# of LSP	% of nodes generating LSPs	Overhead	Overhead	
50	4.6	76.7%	228	41	82%	346	<b>574</b>
100	2.8	46.7%	300	87.5	87.5%	1611	<b>1871</b>
150	1.2	20%	227	139.2	92.8%	3710	<b>3937</b>
200	0.6	10%	136	197.7	98.9%	7048	<b>7032</b>
250	0.4	6.7%	115	248.9	99.5%	10582	<b>10667</b>
500	0	0%	0	499.9	99.9%	41315	<b>41315</b>
# of GPS-QHRA (6 個區域)							
Zone			Node			Total	
# of LSP	% of Zones generating LSPs	Overhead	# of LSP	% of Zones generating LSPs	Overhead	overhead	
50	4.7	78%	223	31	62%	171	<b>394</b>
100	3.2	53.3%	334	82	82%	490	<b>824</b>
150	1.1	18%	217	132	88%	1167	<b>1384</b>
200	0.58	9%	1305	194	97%	2380	<b>3685</b>
250	0.3	5%	86	248	99.2%	3874	<b>3960</b>
500	0	0%	0	497	99.4%	7169	<b>7169</b>
# of GPS-QHRA (25 個區域)							
Zone			Node			Total	
# of LSP	% of Zones generating LSPs	Overhead	# of LSP	% of Zones generating LSPs	Overhead	overhead	
50	24.2	96.8%	1160	28	56%	167	<b>1327</b>
100	22.3	89.2%	2169	79	79%	807	<b>2976</b>
150	17	68%	2496	135	90%	1461	<b>3957</b>
200	14.2	56.8%	2821	195	97.5%	3132	<b>5953</b>
250	11.4	45.6%	2913	244	97.6%	4216	<b>7129</b>
500	6.3	25.2%	3233	491	98.2%	7082	<b>10315</b>

表 5: 移動性對 GPS-QHRA 和 ZHLS 的負載影響(Mobility=40units / cycle)

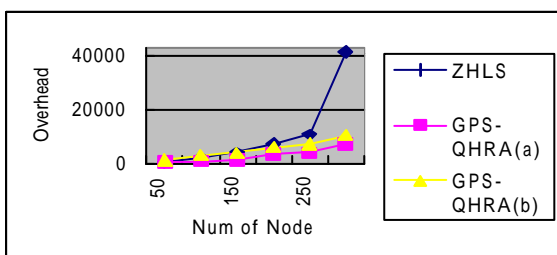


圖 13: 移動性對 GPS-QHRA 和 ZHLS 的負載影響 (mobility= 40 units/cycle)



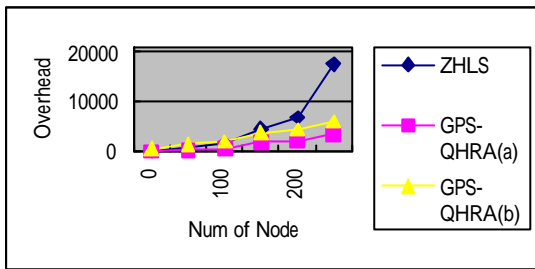


圖 14: 移動性對 GPS-QHRA 和 ZHLS 的負載影響 (mobility= 20 units/cycle)

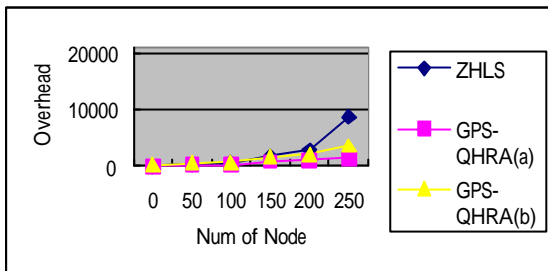


圖 15: 移動性對 GPS-QHRA 和 ZHLS 的負載影響 (mobility= 10 units/cycle)

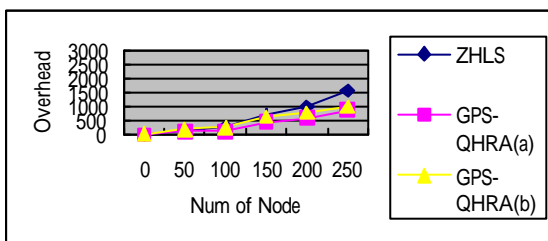


圖 16: 移動性對 GPS-QHRA 和 ZHLS 的負載影響 (mobility= 5 units/cycle)

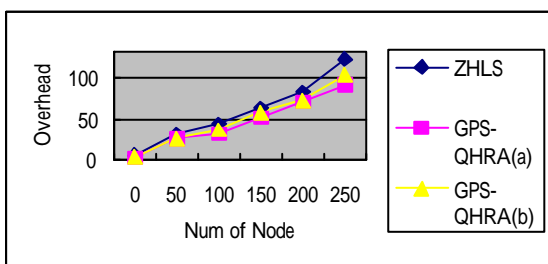


圖 17: 移動性對 GPS-QHRA 和 ZHLS 的負載影響 (mobility= 1 units/cycle)

- 搜尋路徑的負載:我們要模擬比較當一節點要找到它路徑所要產生的封包數量(圖 18)。我們以節點數為 50、100、150、200、250 和 500 來比較,找一固定的節點,來對其他的節點找出路徑,我們模擬 1000 個單位時間,最後把其平均值算出。在 ZHLS 中,當節點要傳送封包給區域外的節點時,需要廣播給所有區域,而

GPS-QHRA 中,我們使用二個模擬環境,分別是 6 個區域的 GPS-QHRA 和 25 個區域的 GPS-QHRA。在 GPS-QHRA 的環境中,由於只需廣播給部分的區域,即廣播給有規則集合中的資料庫節點即可。因此會得到較 ZHLS 較好的效能,而 6 個區域的 GPS-QHRA 和 25 個區域的 GPS-QHRA 比較,6 個區域的 GPS-QHRA 會比 25 個區域的 GPS-QHRA 來得好一些,那是因為在分成 25 個區域在搜尋時會比分成 6 個區域的多。當節點數由小變大時,我們發現負載變小。由於網路中的節點數夠多,因此其發生重送的機率變小,故負載也跟著變小,由模擬的結果中我們可以認證,利用集合的觀念,我們能減少網路對於路徑搜尋的負載

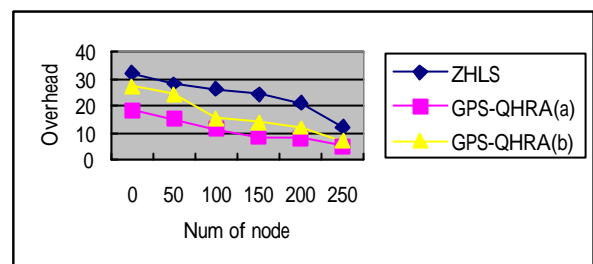


圖 18: 搜尋路徑的影響

- 漏失機率(Dropping Probability)的比較:我們比較節點數為 50、100、150、200、250 和 500 和傳輸範圍為 100、200 和 250 下,分別在 ZHLS、GPS-QHRA 網路環境下漏失機率的比較。以一固定的節點,隨機的對其他節點作路徑的搜尋,若無法正確的找出一條路徑,我們視該次路徑漏失掉,假設在 2000 次的路徑搜尋中漏失掉 L 次,我們的漏失機率即為  $L/2000$ ,路徑的漏失有幾種可能會發生,第一種是當節點數太小時,網路中沒有足夠的節點讓來源節點傳輸到目的節點。第二種可能是傳輸範圍的影響,從圖 19 中我們可以明顯的看到,傳輸範圍為 200 和 250 時,ZHLS 在各個節點時漏失率都為 0,而傳輸範圍為 100 時,才發生一部分的漏失率,因此所得出傳輸範圍愈小時,其發生漏失的機率愈大。第三種可能是出現在 GPS-QHRA 中,假設在某一時刻中,從資料庫節點中查不到目的節點的位置,或剛好查到錯誤的資訊,則有可能發生漏失,從圖可以看出,在 GPS-QHRA 中使用傳輸範圍 100 時,其漏失率相當嚴重,不過,當傳輸範圍回到 200 時,則有很明顯的改善,那是由於參數設計錯誤所致,從模擬的結果中可以看出,在漏失機率方面,ZHLS 比 GPS-QHRA 有較好的效能,不過,當節點數夠多或傳輸範圍夠大時,則沒有多大的影響,因此,只要我們好好設計 GPS-QHRA 中的參數,對整體的效能不會有太大的影響。

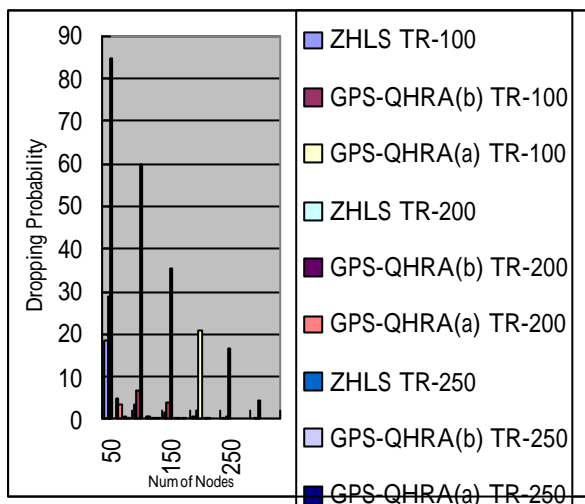


圖 19: Dropping Probability 的比較

## 五、結論

隨意式無線網路在近年來開始有很多繞送演算法的研究。由於在無線網路中，節點會隨意移動，當拓撲(topology)改變時我們要如何來找一新的繞送路徑來傳送。因此我們提出了一個階層式、hybrid 的繞送演算法。這個演算法結合了 GPS 方位偵測(Location Tracking)的技術，以及有規則集合系統的觀念。這兩種技術對於節點會隨時移動的隨意式無線網路環境十分適用。從模擬結果我們可以知道，GPS-QHRA 利用方位資料庫節點用來作一個區域之內的中央控制，因此，每個區域之內只需要一個節點來記錄繞送表格與作定期的更新資訊，如此可改善同樣是利用 GPS 技術的階層式繞送演算法-ZHLS，因此能得到較短的路徑、以及可證明在高度移動的環境中有較好的效能。另外 GPS-QHR 只需局部廣播給有規則集中的方位資料庫節點也可有效降低系統負載。在漏失率方面，由於 ZHLS 付出了較多網路負載的代價，因此其在漏失率中較 GPS-QHRA 為佳，但其中所付出的代價是資料庫節點會造成較大的負載，而且當資料庫節點發生錯誤時，會使漏失率變大，從而影響到系統的效能，這中間必須作一考量(Trade-off)。因此未來的研究可朝 (1)服務品質(QoS)的方向作努力。改善封包常常在節點改變或拓撲改變時而流以造成系統延遲增加的問題。(2)有限頻寬分配的問題，在本篇論文中假設已有一套妥善的頻寬管理方法，然而並沒有真正去探討，因此，此類問題也是值得去探討的。

## 參考文獻

[1] Royer, E.M.; Chai-Keong Toh, "A Review of Current Routing Protocols For Wireless Ad Hoc Networks", IEEE Personal Communications, Volume: 6 2, April 1999,

pp.46-55.

[2] D.Baker et al., "Flat Vs. Hierarchical Network Control Architecture" ARO/DARPA Wksp. Mobile Ad-Hoc Networking; Mar. 1997.

[3] L.Kleinrock and F.Kamoun, "Hierarchical Routing for Large Networks: Performance Evaluation and Optimization" Computer Networks, Vol.1, No.3, 1977, pp.155-174.

[4] C. E. Perkins and P.Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers" Comp. Commun.Rev. Oct.1994, pp.234-244.

[5] C-C, Chiang. "Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel" Proc.IEEE SICON '97, APR.1997. Pp.197-211.

[6] C.E. Perkins and E.M. Royer, "Ad-hoc On-Demand Distance Vector Routing" Proc. 2<sup>nd</sup> IEEE Wksp. Mobile Comp. Sys and Apps, Feb.1999, pp.90-100.

[7] V.D. park and M.S. Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks" Proc. INFOCOM '97, Apr 1997.

[8] Paul, K.; Bandyopadhyay, S.; Mukherjee; Saha, D."Communication-aware mobile hosts in ad-hoc wireless network" Personal Wireless Communication, 1999 IEEE International Conference on, 1999,pp.83-87.

[9] Joa-Ng. M., I-Tai Lu "A peer-to-peer zone-based two-level link state routing for mobile ad hoc networks" Selected Areas in Communications, IEEE Journal on Volume: 178,Aug. 1999, pp.1415-1425.

[10] E.D. Kaplan (Editor), Understanding The GPS: Principles and Applications Artech House, Boston, MA, Feb 1996.

[11] Moore, T. "An introduction to the global positioning system and its applications, Developments in the Use of Global Positioning Systems, 1994, pp.1/1-1/6.

[12] A. Dey, Theory of Block Designs, 1986.

[13] Zhenyu Tang, Garcia-Luna-Aceves, J.J., "collision-avoidance transmission scheduling for ad-hoc networks" Communications, 2000. ICC 2000. 2000 IEEE International Conference on Volume: 3, 2000, PP. 1788-1794, vol. 3.

[14] I-Shyan Hwang, Chiung-Ying Wang "Dynamic Allocation of clustering Technique in Ad Hoc Wireless Network", M.S. of the Department of Computer Engineering and Science Yuan-Ze University.