

無線隨意網路中提供備援路徑的叢集繞送法

黃宗傳 黃繼玄

國立中山大學電機研究所

E-mail tch@mail.nsysu.edu.tw

摘要

在隨意式無線網路中，有良好效率的繞送協定一直是主要的研究議題。近年來，有許多研究提出階層式繞送法，利用建立骨幹架構來支援隨意式無線網路的封包繞送，尤其是大型無線網路。利用形成叢集的方式，可以在平面式的網路中建立階層式的架構，叢集內的叢集管理者具有較大的傳輸功率以傳送封包，利用叢集管理者當做骨幹節點可形成骨幹網路。骨幹網路繞送的好處是可以減少資料封包在整個網路中的轉送跳躍數，然而，過去的協定均著重在叢集架構上，在骨幹架構裡的容錯卻很少被考慮。本論文中，我們提出一個備援繞送的機制，藉由在資料封包的標頭資訊中包裹備援路徑，不需經過路徑重新搜尋的程序，便可以修復被損壞的路徑，達到路徑的可靠性。模擬分析的結果顯示，我們所提出的路徑備援機制繞送法，相對於其它一般的階層式繞送演算法，可以增加封包抵達率以及減少控制封包數。

關鍵詞：階層式繞送，骨幹網路，繞送協定

1. 簡介

隨意式無線網路的特色就像是一個高度變動的網路環境。網路中缺乏基地台的支援，受到限制的無線頻寬，以及節點能量的使用限制等等因素，讓無線網路中的繞送相對於其他傳統的網路環境，顯得更加複雜。在隨意式無線網路環境中，許多已提出的繞送協定可分成三個主要種類：Proactive、Reactive 和 Hybrid 繞送協定。

Proactive 繞送協定又稱為 table-driven。每個節點都維持一份繞送表格(route table)，其中記載網路中節點的資訊，但每隔一段時間，每個節點必須廣播路徑相關訊息，交換繞送表格，以維持最新的網路資訊，當有資料要傳送時，只要參考繞送表格即可找出適合的路徑。此類協定如:DSDV[2]。

Reactive 繞送協定又稱為 on-demand，只有在來源節點需要傳送資料的時候才開始發出詢問(query)封包尋找目的節點，接收到訊息的節點如果不是目的節點則把自己的 IP 位址加到訊息的 route_list 中，然後繼續把訊息廣播出去直到目的節點，如果接收到重複的訊息或者是自己已出現在 route_list 中，節點則會丟棄訊息，最後目的節點沿著 route_list 所記錄的節點，傳送回應(response)

封包回去給來源節點，表示路徑建立成功，這種尋找路徑的控制封包不論封包大小或數量通常比 proactive 方法中為了更新繞送表格的控制封包還要小，且節點不需要耗費資源去儲存繞送資訊，所以對網路造成較小的負擔。此類協定如:DSR[3]。

因為 Proactive 和 Reactive 二種繞送協定各有其優缺點，所以有人提出結合此二種型態的混合式(Hybrid)的繞送協定，每個節點記載局部區域內的資訊，區域內使用 Proactive 的方法查詢繞送表格作繞送，區域外使用 Reactive 的方法搜尋路徑。此類協定如:ZRP[7]。

若依照不同的網路組成的網路架構，繞送演算法也可以分為平面式或是階層式，平面式網路架構比較簡單，通常每個節點所扮演角色都一樣，所以不容易出現負載較重的節點，但由於可擴展性差，所以適合用在網路的規模較小時。階層式結構則是將整個網路的節點分成多個較小的子集合，俗稱叢集(cluster)，每個子集合只用一個或數個特定的節點與其他子集合的節點通訊，這些特定的節點可以相互連結來構成另一層網路，階層式架構可以將網路流量集中於這些特定節點，減少不適合的中間節點參加路徑繞送，這使得階層式路徑繞送演算法可適用於規模較大的網路。

本論文提出一種在隨意式無線網路中提供備援路徑的叢集繞送法(Cluster-based Routing with a Backup Route Protocol, CRBR)，不只提供了骨幹架構網路的好處，藉由長距離連結可以讓兩個相距很遠的節點之間減少轉傳次數，更可以減少傳送的延遲時間，使得傳送更穩定。另外也提出了維護機制來修復在繞送時產生的破損路徑並且不用再次執行路徑搜尋的程序，對於面臨網路拓撲改變時，能夠提供穩定的網路效能。

論文的其餘組成架構為：第二章節說明和本文有關的觀念與相關的研究。第三章節是系統架構與運作，詳細描述我們提出的想法及一些例子，分別以叢集的形成與維持，骨幹繞送，路徑維護機制等部份來做討論。第四章節規劃模擬環境、參數設定並且加以分析和討論模擬的結果。最後，第五章節則為本篇論文的綜合結論。

2. 相關研究

有許多的研究指出平面式的隨意無線行動網路具有的缺點[4]、[6]。每當網路拓撲擴大，節點數增多時執行效能容易降低，造成這種效能下降的主

要因素有更新繞送表格或尋找傳送路徑而產生過多的控制封包造成網路負載。還有就是多重跳躍路徑不易維持傳輸品質的問題和節點的移動性問題，在需要即時傳輸如視訊會議的應用上往往很注重傳輸品質。而多重跳躍路徑也就是需要轉傳多次的路徑容易因為中間轉傳節點的移動或關機，造成單一連結失敗而使得整條路徑不能使用需要再找新的傳輸路徑。

如果隨意式無線網路擴展到大範圍時，建立一個階層式網路相對於平面式網路繞送時所需耗費的處理負載 (overhead) 和儲存空間需求，是相對較小的[5]。然而過去的研究中，階層式架構形成的骨幹網路，繞送時所產生的錯誤容忍度、路徑維護卻是很少被考慮到。

3. 提供備援路徑的叢集繞送法

在本節將介紹在隨意式無線網路中，如何執行本論文所提出的 CRBR 演算法，以下將分成幾個部分，分別敘述叢集架構、骨幹網路的繞送和備援路徑機制。另外在設計演算法時，首先，我們假定每個節點的基本傳輸功率 (basic-power) 是相等的，以及在節點之間的連結是雙向的。基於這項假定，在通訊範圍內的任兩節點是彼此可以互相直接通訊。

3.1 叢集架構

最小 ID 叢集形成演算法或最高連結度叢集形成演算法，由於簡單快速，因此是最被廣泛使用的叢集形成演算法。在此我們採用最小 ID 叢集形成演算法，以便在無組織架構的隨意式無線網路中，形成 one-hop 式的叢集架構。

3.1.1 定義符號

首先定義下面的符號來描述叢集形成的演算法：

- *node_id* : 每個節點擁有的唯一 ID
- *cluster_id* : 每個節點所屬的叢集管理者的 ID, 如果本身就是叢集管理者, 則此欄位為自己的 ID
- *Address* : 每個節點擁有的位址
- *H* : 節點是叢集管理者的狀態
- *M* : 節點是叢集管理者底下成員的狀態
- *U* : 節點是尚未被歸類的普通節點的狀態
- *HELLO* 訊息: 每個節點週期廣播這種訊息讓鄰居節點知道自己的存在。此訊息裡包括 neighbor table 和 neighboring clusterhead table。如圖 1 表示所形成的叢集架構, 表格 1 為圖 1 內節點 7 的 neighbor table 例子以及表格 2 是圖 1 內節點 7 的 neighboring clusterhead table。

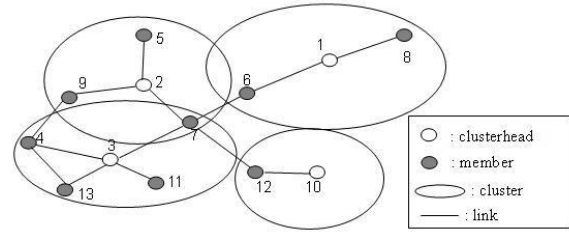


圖 1、one-hop 式的叢集架構

表格 1、節點 7 的 neighbor table

Neighbor id	Cluster id	Neighbor node address	Role
6	1	Node 6's address	M
2	2	Node 2's address	H
3	3	Node 3's address	H
12	10	Node 12's address	M

表格 2、節點 7 的 neighboring clusterhead table

Neighbor cluster id	Gateway node id	Neighbor cluster address
1	6	Node 1's address
2	—	Node 2's address
10	12	Node 10's address

3.1.2 叢集形成與維護

網路初始時，拓撲上的每個節點週期性的廣播 HELLO 訊息，藉以收集附近鄰居節點的資訊。故節點一開始的狀態為 O。當節點收到 HELLO 訊息後，將收到的 HELLO 訊息加入 neighbor table，並判斷自己是否可以成為叢集管理者，節點狀態為 H，或是叢集成員，節點狀態為 M；若無收到 HELLO 訊息，經過一段時間後，則宣告自己成為叢集管理者。而所建立的 neighbor table 經由節點身分不同的篩選，可以建立出 neighboring clusterhead table，neighboring clusterhead table 是紀錄相鄰叢集的資訊表格，可以用來幫助骨幹網路的繞送。

而在叢集架構維護方面，因為 *cluster_id* 是以叢集管理者的 ID 來加以分辨的，當叢集管理者的身分變動時，連帶地 *cluster_id* 也將受到改變，而影響到叢集架構的穩定性。因此為了減少叢集的變動率，我們採用的是 Least Cluster Change[2]，以維護叢集的穩定性。

3.2 繞送機制

我們提出一種利用加大傳送功率的方式來減少路徑上所需要的跳躍數，並達成縮短延遲時間，另一方面，利用在搜尋階段所獲得的路徑資訊，提出路徑維護的機制，這機制的好處在於，損壞的路徑可以在區域上達成修復，而不用回到來源節點執行重新搜尋路徑的程序，以下將詳細敘述 CRBR 的繞送方式。

3.2.1 路徑搜尋

CRBR 所採用的路徑搜尋方式與 DSR 的路徑搜尋原則相同，不同的是，CRBR 採用的是有方向性的搜尋路徑，而非 DSR 的廣播、漫播式的傳送路徑尋找封包 (Route Request, RREQ)，這樣的好處在於可以減少控制封包的無謂的傳送。在路徑搜尋階段，RREQ 總是依循下列的方式傳送：

Source->Clusterhead->Gateway->Clusterhead->Gateway->...->Destination

當 RREQ 到達目的節點時，就會有該路徑上所有節點的資訊。目的節點在眾多 RREQ 封包裡選出一條最佳路徑，根據 Route record 傳送一個路徑回覆封包 (Route reply, RREP) 回到來源節點。

3.2.2 骨幹繞送

由於在路徑搜尋階段，使用的是 source routing 的方式，當資料封包被傳送的時候，封包的標頭內會紀錄路徑上所有經過的節點位址資訊，如圖 2。而骨幹繞送的目的是在於使用叢集管理著所形成的骨幹網路來傳輸資料封包，利用標頭內已知的資訊，改變原有應傳遞的路徑。這樣的好處在於可以縮短來源節點與目的節點兩端間路徑上的跳躍數，達成縮短延遲時間。

Option Type	Option Data Length	Identification	Index	Address [1]	Address [2]	Address [3]	Address [4]	...
-------------	--------------------	----------------	-------	-------------	-------------	-------------	-------------	-----

圖 2、資料封包的標頭格式

當來源節點收到 RREQ 後，由於在搜尋階段方向性的搜尋路徑，因此整條路徑上的節點身分，不是叢集管理者的話，即是 gateway。而一條 multi-hop 的路徑缺點是容易導致路徑斷裂或引起封包遺失，進而浪費頻寬，甚至網路的效能滑落，因此一條路徑應盡可能的縮短路徑的跳躍數。在本篇論文中，我們利用加大傳送功率的方式來傳送資料封包，讓資料不再依循 clusterhead-gateway 交替的方式傳遞封包，而是直接利用叢集管理者間因加大傳送功率後形成的骨幹網路來傳遞封包。

接下來我們討論到在骨幹網路中應該使用何種不同的功率，首先我們根據下列的公式 (1)，套用訊號強度與距離平方成反比，可以將功率單位轉換成以距離單位型式來表示：

$$Pr(d) = \frac{PtGtGr\lambda^2}{(4\pi)^2 d^2 L} \Rightarrow d^2 \propto \frac{1}{Pt} \dots\dots(1)$$

在平面階層中節點使用的功率所能傳送的最遠距離是 R，而在骨幹階層中節點使用較強大的功

率，能傳送的距離也相對較遠，但是由於隨意式節點的能量來源常常是像電池這種有限制的供應，基於能有效率的使用頻寬和現有的能源，所以我們目標是將兩個相鄰的叢集能相互溝通，相鄰即表示中間沒有經過任何其它的叢集，在考慮 distributed gateway 問題後 (圖 3)，發現兩個相鄰的叢集管理者之間最大的跳躍數是 3 hops，而 1 hop 的最大距離為 R，所以兩個相鄰的叢集管理者距離最遠是 3R，因此我們取 high-power 作為骨幹節點的傳輸功率最遠可傳到的範圍，其最遠傳送範圍 3R。

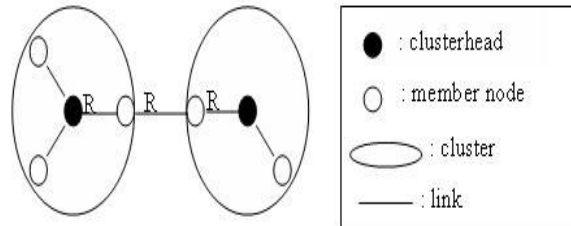


圖 3、兩個相鄰的叢集管理者之間最遠距離是 3R

解決傳送功率大小的問題後，接下來討論叢集管理者如何正確的傳送資料封包給下一骨幹節點 (叢集管理者) 或是目的節點。當某一骨幹節點收到資料封包後，在資料封包的封包標頭裡可以獲得路徑上所有節點的位址資訊，首先將次於自身節點位址與自身節點內所擁有的 neighboring clusterhead table 的位址欄位，比對位址資訊，如果位址相符，即可確定此位址是下一骨幹節點位址，便將資料封包傳送給它；如果是不相符，即再由資料封包的封包標頭裡，提取再次一節點的位址資訊，比對位址資訊，直到找到下一骨幹節點位址或是目的節點位址。獲得位址資訊後，便以 high-power 的方式傳送資料封包，如下一骨幹節點正確未回應 ack 封包，即進入路徑救援機制。

如此一來，在 CRBR 的骨幹繞送階段，原本由路徑搜尋階段所獲得而來的封包路徑，路徑上的所有 gateway 節點將不再收到資料封包。資料封包在骨幹繞送階段，將依循著下列的方式傳送：

Source ->Clusterhead1 ->Clusterhead2 -> ... -> Clusterheadn ->Destination

例如在圖 4 中，當來源節點 a 要將資料傳送給目的節點 p，節點 a 首先發送 RREQ 來搜尋到節點 p 的路徑。當節點 a 獲得到節點 p 的路徑，<a-b-d-e-h-k-m-o-n-p>。在骨幹繞送階段，此路徑成為較短的骨幹路徑，<a-b-e-k-n-p>。其中的原本路徑上的 gateway 節點，<d、h、m、o>，則不需要再轉送封包。兩相比較下，可以容易的得知在來源節點與目的節點間，路徑上的跳躍數變少，資料封包可以沿著骨幹路徑，傳送的更快。

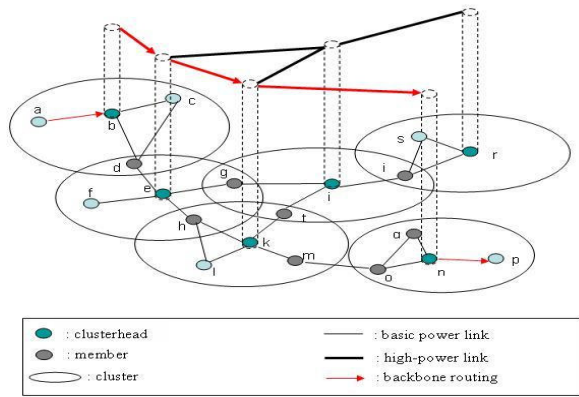


圖 4、骨幹繞送

3.2.3 備援路徑機制

因為骨幹節點（即叢集管理者）擁有較大的傳輸範圍，骨幹節點外的能量消耗也相對較大，將會導致它們被迫關機或者消失在網路拓撲之中。另外，當一個傳送中的骨幹節點發現它與它的下一骨幹節點間的連結斷裂，它將發送路徑失誤封包（Route Error, RERR）回到來源節點，通知來源節點此路徑斷裂。如上所述的關機，連結斷裂等狀況，這些情況都會危害到路徑的存在。因此本論文所提的 CRBR 演算法也提出了路徑維護機制，來修復這些遭損毀的路徑。

當一個骨幹節點發現連結斷裂，在發送 RERR 封包之後，路徑維護機制將利用資料封包的標頭資訊，這些標頭資訊是來自路徑搜尋階段所獲取的網路資訊，不須另外建立或者搜尋，再搭配上自身 HELLO 訊息內的資訊來拯救路徑。以下圖 5 將詳細敘述路徑救援的方式。

- BCH: The Broken ClusterHead.
 - PCH: The Present ClusterHead. It is a clusterhead before BCH in the source route.
 - NCH: The Next ClusterHead. It is a clusterhead after BCH in the source route.
 - FBG (Front Backup Gateway): The gateway between PCH and BCH.
 - RBG (Rear Backup Gateway): The gateway between BCH and NCH.
1. When PCH detects that BCH has disappeared from the network, it transmits the data packet to FBG
 2. FBG checks the packet header and forwards the data packet to RBG using high-power.
 3. RBG receives the packet, it gets the next address from the packet header and checks the neighboring clusterhead table to decide whether the next node is NCH. If so, RBG transmits the packet to the NCH. Otherwise the next node address is checked by repeating the process until a clusterhead is found or the destination is reached. When a clusterhead is found, the data packets will be forwarded along the original backbone route from now on.

圖 5、路徑備援機制

例如在圖 6 之中，有一條骨幹路徑 $\langle a-b-e-k-n-p \rangle$ 正在傳送資料，然而在路徑上的節點 e 移動或消失了，節點 b 也發現到與節點 e 之間的 b-e 連結斷裂，資料無法順利傳送，這時便啟動了備援路徑機制。

節點 b 不再使用骨幹路徑，而依資料封包的標頭資訊，傳送資料封包給下一節點，也就是節點 d。節點 d 先查看資料封包裡的標頭資訊，得知原本的下一節點是節點 e；但節點 e 已消失，這時節點 d 便在標頭資訊裡取得節點 e 的下一位址，也就是節點 h。節點 d 將資料封包傳送給節點 h，就像骨幹節點一般地以 high-power 的方式傳送資料封包。當資料封包抵達節點 h 後，節點 h 也從標頭資訊裡獲取下一節點位址，並將此一節點位址與自身的 neighboring clusterhead table 比對位址欄位，以便找出一叢集管理者，也就是節點 k。如果節點 k 與節點 h 相距只有一個 hop，則以 basic-power 傳送資料封包，否則以 high-power 的方式傳送。當叢集管理者節點 k 收到資料封包後，也代表回到了骨幹路徑上，之後便以骨幹繞送的方式傳送資料封包，直到封包抵達目的節點。

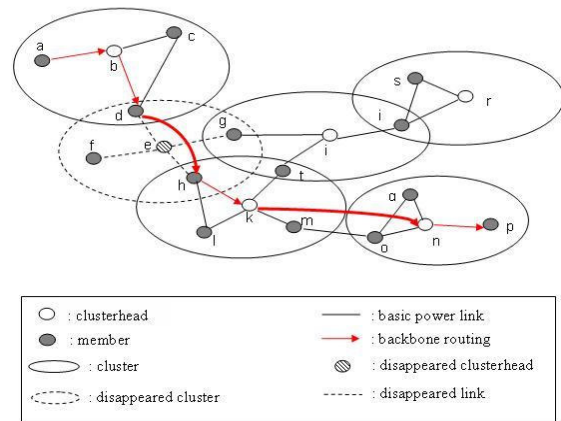


圖 6、路徑備援的例子

備援路徑機制的好處有兩點：

- [1] 對於任何一對來源節點與目的節點間的骨幹路徑，總是存在一條備援路徑來抵擋路徑失誤。
- [2] 由骨幹路徑轉換到備援路徑上的時間小於重新建立端點與端點間的路徑的時間。

4. 模擬結果與分析

本章節將對我們所提出的 CRBR 演算法與其他演算法，做效能上的比較與分析。我們則與 Dynamic Source Routing Protocol (DSR) 和 CRBR without backup route maintenance (即一般的階層式繞送協定)，記為 CRBR-NonBR，來作比較。接下來並將說明我們在模擬之前所提出的假設和模擬軟體 NS-2[8]裡的環境參數設定。最後，則是模擬實驗的結果數據和分析。

4.1 模擬環境假設與參數設定

在我們設計的模擬環境中，主要僅限於網路層 (network layer)，對於資料連結層 (data link layer) 的一些細節，如多重存取所造成的干擾 (multiple

access interference)、無線通訊的功率衰減、隱藏節點問題等，並不加以考慮，而實體層(physical layer)以及頻道資源之細節問題，也不在考慮之列。在本篇論文中，假設每個行動節點均擁有相同的電源能量和傳輸功率，只要兩個節點互相在傳輸範圍內時，雙方就可以成功的傳送資料；假定不會有單向連結的問題發生。連結的問題可以用現有的 IEEE 802.11 協定加以解決。

以下是模擬環境所設定的一些參數：

模擬空間：670 公尺 * 670 公尺
 節點個數：50、100 個
 傳輸半徑：100 公尺、300 公尺
 移動速度：0 ~ 20 公尺/秒
 封包大小：64 byte
 封包速率：4 packets / sec
 移動模式：Random waypoint
 模擬時間：600 秒
 Traffic source：Continuous bit rate (CBR)
 Data flow：10 data flows

其中 random waypoint 的移動方式，是指網路上的每個節點先隨機選取一個方向，然後以預先設定的速度移動到該點，移動完接著便休息一段時間 (pause time)，然後再隨機選取下一個方向，重複執行此移動模式。在 X 軸的 pause time，越往左靠，所代表的移動性越高，以 pause time=0 時，節點移動性最高，無時無刻不停地移動；越往右靠，所代表的移動性越低，以 pause time=600 時，節點移動性最低，節點是處於原地不動的狀態。

4.2 數據分析與討論

針對以下 3 個效能項目分析：

- [1] Packet delivery ratio: 節點傳送資料時會因為轉傳節點的移動而造成路徑斷裂，此時必須重新搜尋另外的傳送路徑，期間來源端還是會持續傳送資料封包，因而造成封包流失，此實驗是測量繞送期間資料封包的傳輸抵達率，詳細作法是在預先設定的環境下，將正確送達之資料封包總數除以發送之資料封包總數便得此值。
- [2] Average end-to-end delay of data packets: 包含所有可能引起延遲時間的事件，有路徑搜尋所耗用的時間、儲存在 queue 裡所耗用的時間、節點的處理時間、因 MAC 層引起的重傳延遲時間、傳播和發送的時間。在這分析中，唯有成功抵達的資料封包所產生的延遲時間才被計數。
- [3] Normalized control message overhead: 在這一數據分析中所探討的是為了成功繞送資料封包所產生的控制封包數，其中並不包括叢集行程與維護所產生的控制封包 (HELLO 訊息) 數量。計算的方法是所產生的控制封包總量除以成功抵達的資料封包總量。

首先，圖 7 和圖 8 顯示出移動性對封包抵達率的影響。我們發現不論在任何移動率下，階層式的 CRBR 都比平面式的 DSR 表現得好，這是因為 DSR 的封包路徑跳躍數比 CRBR 多，而產生較高的失誤率。而比較 CRBR 和 CRBR-NonBR，CRBR 總是比 CRBR-NonBR 突出，尤其在高移動性的情況下。這是因為 CRBR 具有路徑維護機制，可以提高封包抵達率。

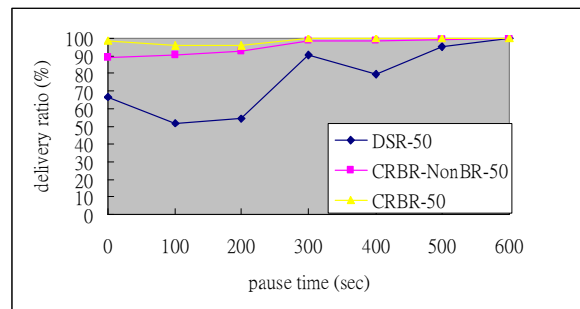


圖 7、50 個節點的封包抵達率

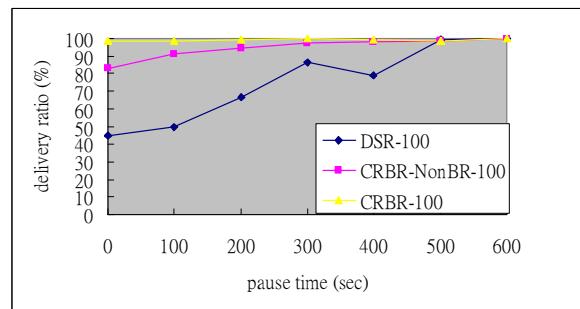


圖 8、100 個節點的封包抵達率

圖 9 和圖 10 中顯示的是移動性對兩端點間延遲時間的影響。平面式的 DSR 由於封包抵達率較低，路徑重建次數高，RREQ 和 RREP 的封包多，影響到資料封包在 queue (採用的是 priority queue) 裡的等待時間，造成在移動性越高的情況下，兩端間的延遲時間快速升高。而階層式的 CRBR，有穩定的封包抵達率，因而延遲時間穩定上升；但略遜於 CRBR-NonBR，因為 CRBR-NonBR 的路徑跳躍數略小於 CRBR。但 CRBR 的封包成功抵達總數比 CRBR-NonBR 多，如表格 3。

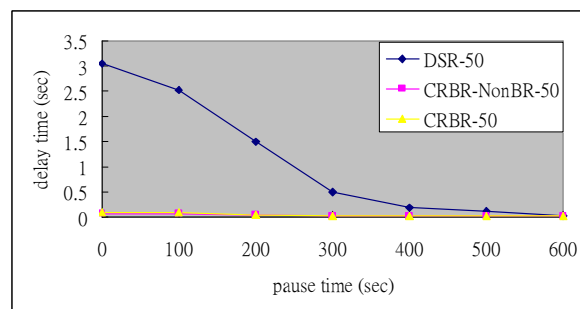


圖 9、50 個節點的 E2E delay

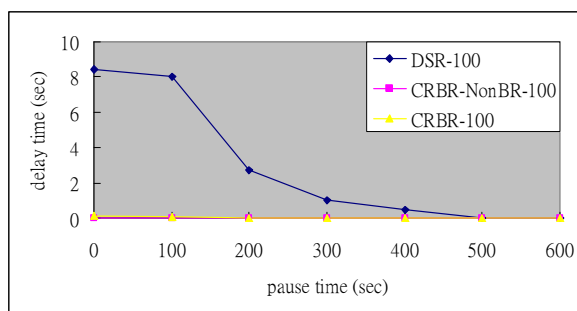


圖 10、100 個節點的 E2E delay

表格 3、資料封包抵達總數

	NonBR-50	CRBR-50	NonBR-100	CRBR-100
0	18040	20645	17137	21001
100	17511	18809	18351	20281
200	17959	18649	19006	20197
300	19733	19964	19494	20024
400	19780	19998	19507	19979
500	19831	19953	19776	19985
600	19946	19947	19922	19922

最後則是移動性對控制封包數的影響，如圖 11 和圖 12。可以看出所有測試的繞送法，控制封包數都隨著移動性升高而上升。在這種情況下，路徑搜尋的次數成為控制封包數的主要因素。DSR 的失誤率高，因此所需要的繞送控制封包數相對也高。而具有路徑維護機制的 CRBR，不需要路徑重新搜尋，即可維護兩端間的路徑，所以控制封包數略小於 CRBR-NonBR。

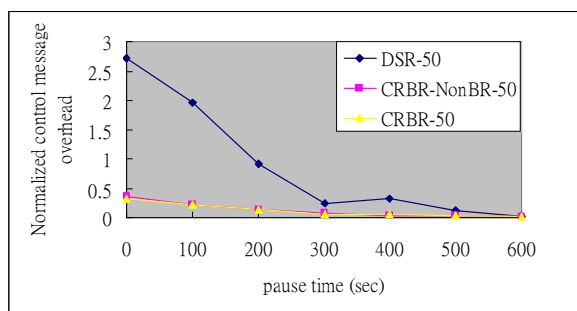


圖 11、50 個節點的控制封包數

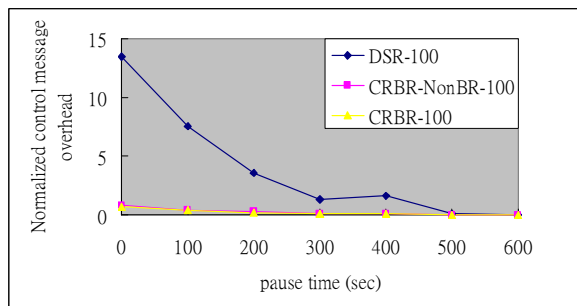


圖 12、100 個節點的控制封包數

5. 結論

在這一篇文章中，我們主要的議題在於討論在階層式隨意無線網路中建立備援路徑。典型的例子中，有數百個節點分布在一個大範圍的區域裡，在來源節點和目的節點的跳躍數相對的也會變大，而階層式的架構能減少到遙遠目的節點距離的跳躍數。用較短的骨幹網路路徑代替較長的平面架構的多重跳躍數路徑。然而，任何一個在節點間的連結斷裂都會引起資料傳輸的錯誤。我們所提出的 CRBR 演算法可以減少資料傳輸路徑的跳躍數，並且可以修復已損壞的路徑。實驗結果證明我們提出的 CRBR 演算法可以穩定網路面臨網路拓撲變動時的效能，提升傳送速度和傳送的成功率。

參考文獻

- [1] C. C. Chiang, H. K. Wu, W. Liu, and M. Gerla, "Routing in Clustered Multihop, Mobile Wireless Network with Fading Channel", in Proceedings of IEEE Singapore International Conference on Networks, April, 1997, Singapore, pp. 197 - 211.
- [2] C. E. Perkins, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," Computer Communications Review, ACM SIGCOMM, Vol. 24, No. 4, October 1994, pp. 234 - 244.
- [3] J. Broch, D. B. Johnson, and D. A. Maltz, "The dynamic source routing protocol for mobile ad hoc networks", In Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom), November 1996, Rye, New York, pp. 153 - 181.
- [4] P. Gupta, and P. R. Kumar, "The Capacity of Wireless Networks", IEEE Transactions on Information Theory, Vol. 46, No. 2, March 2000, pp. 388 - 404.
- [5] S. Banerjee and S. Khuller, "A Clustering Scheme for Hierarchical Control in Multi-hop Wireless Networks", in Proceedings of the IEEE International Conference on Computer Communication (INFOCOM), April 2001, Alaska, USA, pp. 1028 - 1037.
- [6] S. R. Das, C. E. Perkins and E. M. Royer, "Performance Comparison of Two On-demand Routing Protocols", in Proceedings of the IEEE International Conference on Computer Communication (INFOCOM), March 2000, Tel Aviv, Israel, pp. 3 - 12.
- [7] Z. J. Haas, "A new routing protocol for the reconfigurable wireless networks", in Proceedings of IEEE International Conference on Universal Personal Communications, October 1997, San Diego, pp. 562 - 566.
- [8] NS - Network Simulator : <http://www.isi.edu/nsnam/ns/>