

南 華 大 學

資訊管理研究所

碩士論文

一個新叢集協定基於有效能源為無線隨意網路

A New Cluster Protocol Based on Efficient-Energy  
for Wireless Ad Hoc Network



研 究 生：李岳勳

指 導 教 授：吳光閔 博士

中 華 民 國 九 十 六 年 六 月

# 南 華 大 學

資訊管理學系

碩 士 學 位 論 文

一個新叢集協定基於有效能源為無線隨意網路

研究生：李岳鈞

經考試合格特此證明

口試委員：蔡德諤

孫思強

吳光閔

\_\_\_\_\_

\_\_\_\_\_

指導教授：吳光閔

系主任(所長)：吳光閔

口試日期：中華民國 96 年 6 月 20 日

## 誌 謝

這兩年的碩士班生涯轉眼間就過去了，在研究所的生活可以說是既忙碌又精彩，讓我學習到許多的專業知識與經驗。首先要感謝的是我的指導教授吳光閔博士，在撰寫論文期間，歷經了文獻收集、研究方法及數據分析等過程中，給了我很多在研究方面的教導。所以在此獻上最高的敬意與謝意，感謝吳教授在這兩年對我的諄諄教誨。學生最大的收穫並不在於研究之成果，而是在學習及研究的過程中所得到的感想與人生啟示。同時也謝謝口試委員蔡德謙教授與謝昆霖教授對本論文所給予寶貴的建議與不同的思考方向，使得本論文能夠更嚴謹與完善，謹向兩位教授致上最誠摯的謝意。

當然也要感謝實驗室的士颺學長與同學小賴、雅君、珊珊、佻紘及學弟妹小豐、詠生、淳淳，在這段難熬的時光裡大家彼此勉勵，終於完成碩士論文。特別的是五、六月這段在實驗室熬夜的日子裡，我、佻紘和詠生都與泡麵、罐頭及優酪乳為伍，這段時光絕對是我人生中最難忘的經驗之一。

最後感謝我的父母親，辛苦勤勞的工作及無悔的付出並在背後默默支持，還有大姐、二姐、三姐以及所有好朋友的支持與鼓勵，使我能夠順利的完成我的碩士學位。沒有各位的相助，也就沒有今天的我。在此僅將論文之成果貢獻給大家，榮耀與各位同分享，感謝大家陪我走過這一段精彩的日子。

# 一個新叢集協定基於有效能源為無線隨意網路

學生：李岳勳

指導教授：吳光閔

南 華 大 學 資 訊 管 理 學 系 碩 士 班

## 摘 要

行動隨意網路 (Mobile Ad hoc Network, MANET) 是一種無固定網路架構的，且是暫時性網路。在 MANET 網路環境下，每個行動主機可以隨意移動，並且具有發送和接收訊號的裝置。來源節點傳送資訊到目的地節點時，必須透過 hop to hop 方式。在有限的能源與頻寬，如何節省能源或減少網路負載是非常重要的。

有許多叢集演算法 (Cluster algorithm) 被提出來研究，但在這些叢集演算法裡，卻沒有考量行動主機之能源因素 (Energy factor)。這些叢集演算法可能造成叢集改變增加，使得 MANET 網路效能下降。本文中我們提出一個新的叢集演算法，將考量能源消耗與鄰居數量之因素。這二個因素在建立叢集時是非常重要的，因為這可以決定叢集標頭角色。後來將利用叢集內節點數量與權重值因素維護叢集標頭。

模擬結果顯示，本文中所題出的新叢集協定基於有效能源演算法，在 MANET 網路可以延長 40% 至 88% 的平均網路壽命。MANET 網路整體的效能提升，使得 MANET 網路達到較佳的叢集效能及減少能源快速消耗。

**關鍵字：**叢集演算法、有效能源、路由協定

# **A New Cluster Protocol Based on Efficient-Energy for Wireless Ad Hoc Network**

Student : Yueh-Hsun Li

Advisors : Dr. Guang-Ming Wu

Department of Information Management  
Nan-Hua University

## **Abstract**

Mobile Ad hoc Network is Uninfrastructure and Temporary. In the environment of MANET, every mobile node can move and have devices of delivery and receive. The information delivered in MANET from source node to destination node through by hop to hop. It is very important that how to save power or reduce the loading of network causes limited power supply and bandwidth.

A number of proposed Cluster Algorithms have been explained. But these Cluster Algorithms, Actually has not considered energy of factor the mobile host. These Cluster the algorithms possibly to create Cluster the change to increase, causes the MANET network performance reduction. For this reason, we proposed a new cluster Algorithm will consider the energy consumption and the number of neighbor factor to solve the problems above. These two factors are important, because is the use cluster head in the decision. Employ number and weigh value of node of cluster factor to maintain cluster head. Furthermore, we also propose routing scheme used in CPEE.

The simulation results show that our scheme is more efficiently on Network lifetime than HCC scheme.

**Keywords: Clustering Algorithm, Efficient-Energy, Route Protocol**

# 目 錄

書名頁.....	i
博碩士論文授權書.....	ii
著作財產權同意書.....	iii
論文指導教授推薦書.....	iv
論文口試合格證明.....	v
誌謝.....	vi
中文摘要.....	vii
英文摘要.....	viii
目 錄.....	ix
表目錄.....	xi
圖目錄.....	xii
第一章 緒論.....	1
1.1 研究背景.....	2
1.2 相關研究.....	8
1.2.1 傳統繞徑協定.....	8
1.2.1.1 DSDV Protocol.....	9
1.2.1.2 AODV Protocol.....	10
1.2.1.3 ZRP Protocol.....	11
1.2.2 Cluster Based Routing Protocol.....	13
1.2.2.1 最低識別碼叢集法.....	15
1.2.2.2 高連接度叢集法.....	16
1.2.2.3 最少叢集變化演算法.....	18
1.2.2.4 權重叢集法.....	18
1.2.2.5 K-hop 叢集法.....	19
1.3 A New Cluster Protocol Based on Efficient-Energy (CPEE).....	19
1.4 章節架構.....	21
第二章 問題描述.....	22
第三章 研究方法.....	25
3.1 叢集路由協定基於有效能源.....	25
3.2 叢集管理.....	31
3.2.1 選擇閘道節點與連接鄰近叢集節點方法.....	32
3.2.2 內部傳輸與外部傳輸方法.....	36
3.2.3 叢集維護與拓樸改變方法.....	37
3.3 叢集路由協定.....	40
第四章 模擬實驗.....	45
4.1 模擬環境.....	45

4.2 模擬結果.....	46
第五章 結論.....	52
參考文獻.....	53

## 表 目 錄

表 3-1、成立叢集時節點 5 建立的鄰居表.....	31
表 3-2、節點 9 成為閘道節點時，廣播給叢集標頭的訊息.....	36
表 3-3、節點 6 成為閘道節點時，廣播給叢集標的鄰近叢集資訊.....	36
表 3-4、沒被覆蓋的區域由節點 9 傳輸.....	36
表 4-1、模擬環境的參數表.....	45
表 4-2、70 個節點下平均 Network Lifetime 之效能分析 ( $\alpha=0.95$ ) .....	47
表 4-3、70 個節點下平均 Network Lifetime 之效能分析 ( $\alpha=0.65$ ) .....	48
表 4-4、70 個節點下平均 Network Lifetime 之效能分析 ( $\alpha=0.05$ ) .....	48
表 4-5、移動速率 40 km/hr 下平均 Network Lifetime 之效能分析 ( $\alpha=0.95$ ) .....	49
表 4-6、移動速率 40 km/hr 下平均 Network Lifetime 之效能分析 ( $\alpha=0.65$ ) .....	50
表 4-7、移動速率 40 km/hr 下平均 Network Lifetime 之效能分析 ( $\alpha=0.05$ ) .....	51

## 圖目錄

圖 1-1、有基礎架構的無線網路環境.....	2
圖 1-2、無基礎架構的無線網路或 MANET 環境.....	4
圖 1-3、隱藏節點的問題.....	5
圖 1-4、Overhear 的問題.....	6
圖 1-5、AODV 協定.....	11
圖 1-6、ZRP 1-hops 區域示意圖.....	12
圖 1-7、網路封包氾濫.....	14
圖 1-8、叢集架構.....	15
圖 1-9、最低識別碼叢集法.....	16
圖 1-10、高連結度叢集法.....	17
圖 1-11、有效能源叢集演算法.....	20
圖 2-1、每個節點計算權重值.....	24
圖 2-2、選擇叢集標頭 (1-hop).....	24
圖 3-1、建立叢集演算法.....	30
圖 3-2、2-hop 叢集環境.....	31
圖 3-3、選擇閘道節點演算法.....	34
圖 3-4、叢集標頭選擇閘道節點.....	35
圖 3-5、叢集標頭改變.....	38
圖 3-6、解除叢集標頭演算法.....	39
圖 3-7、假設成員移動方向.....	40
圖 3-8、路徑發掘.....	43
圖 3-9、路徑維護.....	44
圖 4-1、70 個節點下平均 Network Lifetime 之效能分析.....	47
圖 4-2、移動速率 40 km/hr 下平均 Network Lifetime 之效能分析.....	49

# 第一章 緒論

民眾對網際網路(Internet)的依賴日深，在家上網的應用行為中，仍是以「瀏覽資訊」及「收發 E-Mail」最為普遍。但近幾年來，網際網路帶給國人的附加價值越來越多，並且越來越貼近國人日常生活上的需求，而一些網路應用服務的使用人口，在近幾年來皆呈現大幅度的成長。其中，以「網路電話」、「線上轉帳或信用卡交易」及「即時短訊 (Instant message, IM)」的成長最為驚人。這顯示出越來越多人透過網際網路進行人際溝通，這不僅具有即時性和便利性，更有節費效益；而透過網路進行各項金融交易，更節省了跑銀行、跑便利商店繳款時所耗費的時間成本。在醫療、娛樂、生活、食衣住行，都帶給我們許多方便。

隨著科技一直不斷進步行動隨意網路 (Mobile Ad hoc Network, MANET) 無線通訊科技迅速發展，使得行動通訊已成為電腦科技領域中備受矚目的一環。MANET 是一種完全經由無線連結的行動節點 (Mobile node) 所建構而成的網路架構，具有動態連結、頻寬限制及電力限制等特性[23]。網路裡的行動成員則可能有個人數位助理 (Personal Digital Assistant, PDA)、平板電腦 (Table PC)、行動電話 (Mobile phone)、筆記型電腦 (Notebook, NB) 或任何具有無線通訊能力的設備所組成。遊走各個地方開會或是工作，而不受限於特定空間地點，為使用者帶來很大的方便與機動性。而通訊與網路技術的持續發展，不僅縮短了世界的距離，也使其成為現代人日常生活不可或缺的重要部分。無線網路的主要架構可以分為兩大部份：有基礎架構的無線網路 (Infrastructure Network) 和無基礎架構的無線網路 (Non-infrastructure Network)，在下一節做詳細介紹。

## 1.1 研究背景

在 Infrastructure Network 裡可以區分為兩種角色，一種是基地台 (Base-Station, BS) 或擷取點 (Access-Point, AP)。AP 本身具有橋接器的功能，依照各廠牌的不同再加上一些額外的服務項目，其發射功率所及的區域便形成了一個基本服務區。另一種是用戶端 (Client) 包含：PDA、Table PC、NB 等，基地台或擷取點為資料交換的重要角色，因為它負責在無線環境下所有用戶端之間的訊息傳遞以及廣播資料項目給管轄範圍內的用戶端；而基地台與基地台之間是透過骨幹網路或有線網路來互相傳遞訊息，但是在基地台廣播覆蓋範圍下的用戶端是可以透過無線電波與伺服器 (Server) 進行連線 (如圖 1-1)。

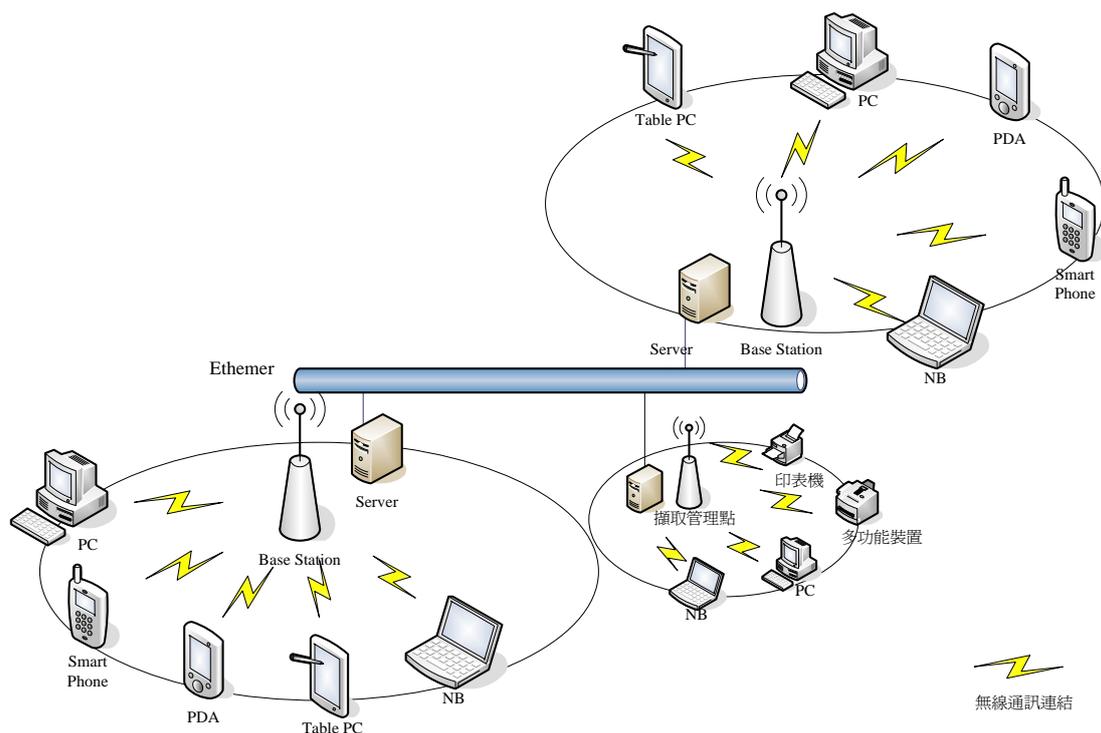


圖 1-1、有基礎架構的無線網路環境

而 MANET 網路是屬於 Non-infrastructure Network 架構，所以每個行動主機必須扮演基地台的角色。MANET 網路是一種方便且彈性

高的技術，透過其 Multihop 的無線架構（如圖 1-2），讓行動主機（Mobile host，又稱節點）透過與鄰近節點做訊息交換，如此就可以不用依靠固定的基地台就能夠彼此通訊。

由於 MANET 網路是透過無線的網路通道，在使用上具備移動性的特性，因此在無線網路的環境中，每個行動主機不需要在固定的位置，由於節點在實體上是屬於可移動的，因此邏輯上網路拓樸（Topology）也有可能隨之改變。雖然無線網路具有比有線網路更佳的機動性。又由於 MANET 網路是缺少中央控管的運作模式，所以所有的節點都必須另外再具備路由器（Route）的功能。除了運行一般通訊協定（Protocol）之外，還必須運作包括路由尋找、路由維護、封包轉送等常見的路由操作。

因此 MANET 網路環境可適用於災區緊急救援、戰區通訊及小型集會等無基地台支援的通訊環境。但是卻存在一些先天上的限制，例如有限的頻寬、有限的能源（Limited energy）壽命及有限的記憶體容量等，這些問題可能造成無線網路載體效能降低。加上行動主機本身隨意移動的特性，造成資料傳送的困難，因此在通訊的即時性及網路資源的平衡性上很難兩全。如何在有限的資源環境下，達到即時性通訊，效能與可靠度良好，並減少資料遺失的叢集協定，便是一個值得研究的議題。

行動主機大多以電池作為主要能源來源，在使用行動主機時，使用者經常要為了能源電量（Coulometer）而困擾，就像是一般筆記型電腦在使用中不充電的情形下，往往無法使用超過 4-5 個小時，便需要更換電池或是充電，其他的行動主機如：PDA、Smart Phone 等，然而科技發展能持續較久的能源，但能源的消耗仍然令人困擾；因此電池持續力往往直接影響行動主機在網路上的存活時間（Lifetime）。

由於 MANET 網路的特性之一是 Multihop 資料傳送，行動主機若是因能源耗盡，除了影響資料傳送外，嚴重者甚至會使得網路連通失效，造成資訊無法傳遞。

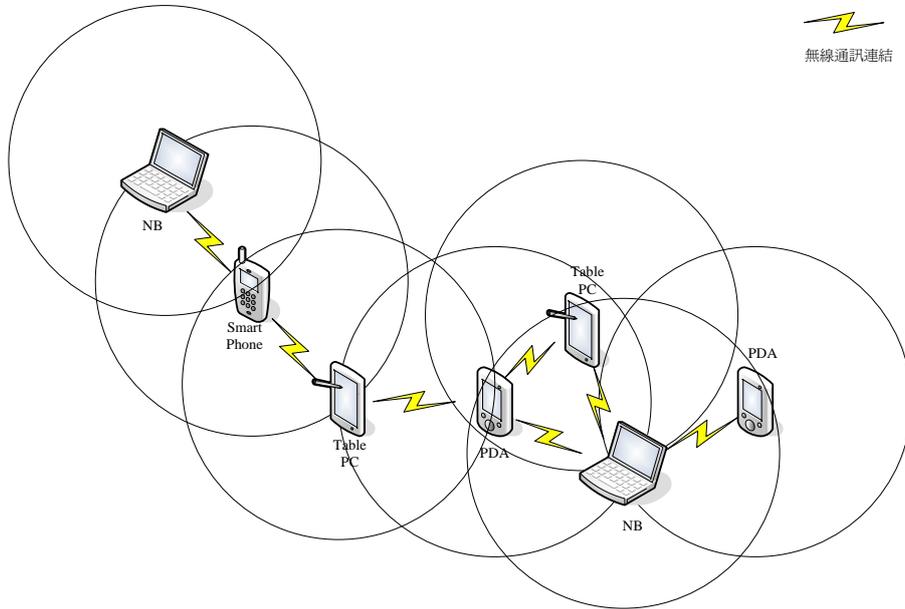


圖 1-2、無基礎架構的無線網路或 MANET 環境

我們將 MANET 網路的特性及問題整理如下[42]：

(1) 網路具有自動組織能力：

MANET 網路可在任何環境、時間及地點完成小型網路建構，而不需要任何現有的基礎網路建設。

(2) 網路拓樸具動態改變：

在 MANET 網路環境中重要的優勢是機動性，但由於行動主機的任意移動，加上無線發送裝置功率的變化、訊號相互干擾、地形等等各式各樣的影響，使得網路拓樸隨時都會發生變化。造成追蹤行動主機時候的困難，也額外的耗用一些資源，如：無線電干擾、無意聽到（Overhear）的問題。

當然我們不太可能自行隨意使用某些無線電頻率，因此常常

在使用 MANET 網路時會面臨到無線電波的干擾問題，例如：IEEE 802.11b 或 802.11g、藍芽、微波爐或是其他的無線設備都是操作在 2.4GHz 附近的頻帶，可想而知當這些東西都在工作時，無線電上的干擾勢必會很嚴重，這都會使得我們的網路受到干擾或是被癱瘓掉。另外，並不是只有非節點的裝置會干擾到網路的運作，節點間也常常會彼此干擾，如圖 1-3 中的隱藏節點（Hidden terminal）問題，當節點 a 正廣播訊息時，節點 c 也在廣播訊息，則無線訊號可能會彼此干擾，使得無法接收到訊息，這也都會影響 MANET 網路的效能。

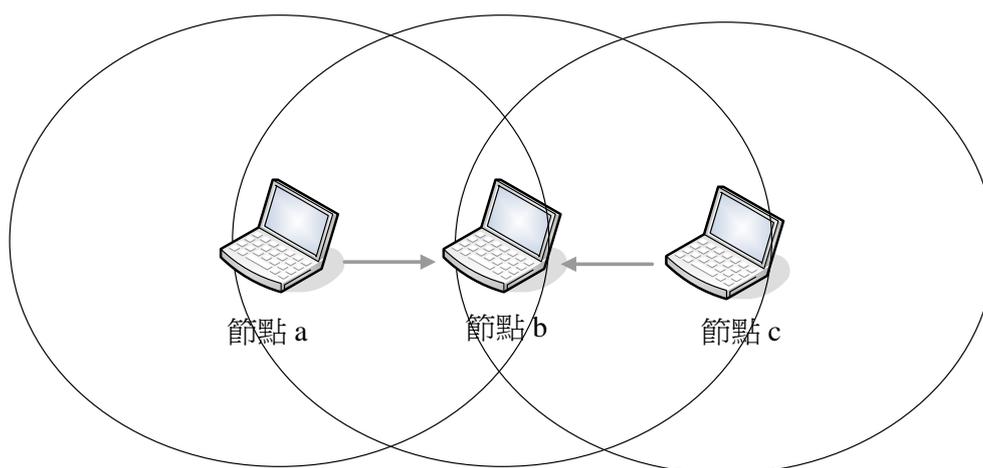


圖 1-3、隱藏節點的問題

當一個節點在傳遞封包給下一個結點時，由於其傳輸範圍內可能有不只一個的節點，這時候這些鄰近的節點可能會 Overhear 節點正在傳送的封包，如圖 1-4 中的行動主節點 a 要傳送資料給節點 b 時，節點 c 與節點 d 兩個節點也都會接收到這筆資料，此時對於節點 b 以外的節點都會消耗電力來接收這個與其無關的封包，這很明顯的是一個浪費的行為，尤其當節點在拓撲上呈現

不均勻分佈時，便很容易發生這樣的情況。

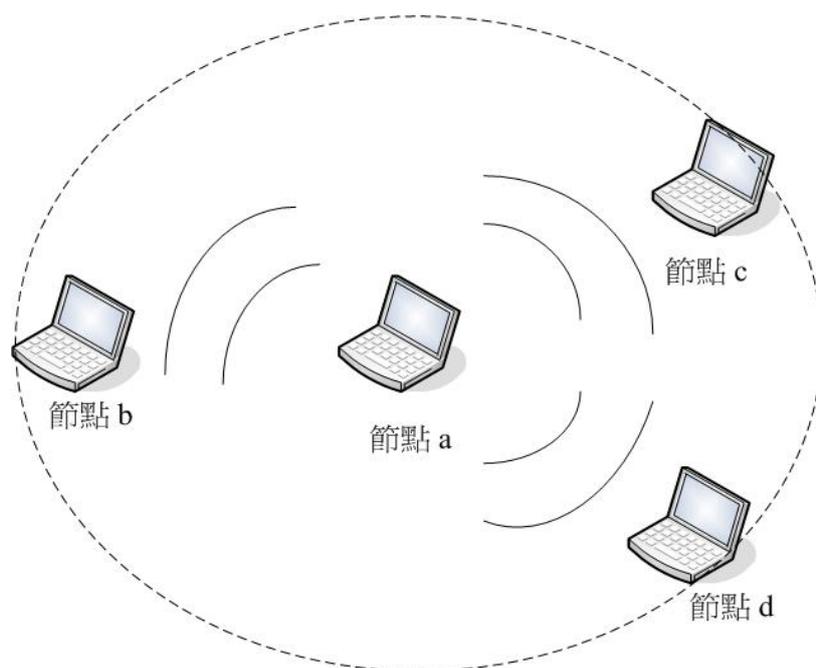


圖 1-4、Overhear 的問題

(3) 存活時間較固定網路短及電池壽命：

由於 MANET 網路往往都是為了特定原因而臨時建構出來的，因此使用結束後網路環境就會消失，相較於固定網路來說，MANET 網路的生存時間相對比較短。

在電池壽命的限制問題，為了享受無線通訊帶來的便利性，無線通訊設備所使用的電源自然非固定式的插在電源座上，所依靠的是越來越精小的電池。電池的持續力對於無線設備來說是相當重要的命脈，無論功能多強大的通訊設備，都將在電池用盡的那一刻失去效用，因此如何讓電池的使用時間延長，如何減低通訊設備的電池耗用，便是相當重要的一個關鍵。所以，許多學者在實用的研究方面，考量的便是如何增加電池的持續力[23] 和減少通訊設備的耗電量，或是改善網路架構，利用最少的通訊訊號，達到最好的通訊效果，其中，省電或休眠模式便是降低耗電

量的有效方法，另外開發更持久的儲電元素或是透過大自然接收電力，例如太陽能，也能有效的加強電池的持續力。

(4) 安全性較差：

由於 MANET 網路是使用無線的方式所佈建，相較於有線網路，在先天上就比較容易遭受到竊聽、入侵等等的攻擊。現階段的無線網路通訊，對於訊息傳遞上安全的防護是相當薄弱的。由於無線網路的傳遞，是透過電波作為媒介，只要能在傳輸範圍內接收到發射的電波，便可以擷取到網路的資源。因此，訊息的傳遞是需要適當的編碼及加密的動作。資訊安全的相關研究，在有線網路的環境上實行已久，在無線網路的領域上更是需要更多的研究。資訊的保護嚴謹與否，將決定帶來的傷害有多大，因此我們更應該謹慎以對。

(5) 網路頻寬有限：

目前 MANET 網路的傳輸速度約在 11~54Mbps，無線網路本身的物理特性包括訊號衰減、碰撞或雜訊干擾等等原因，使得 MANET 網路實際傳輸的頻寬相較於有線網路（100Mbps）會來得更小。所以如何有效的利用這些有限的頻寬也是十分重要的問題。

(6) 行動主機設計的侷限性：

MANET 網路的行動主機本身體積較小、CPU 處理能力較低及電力，導致處理器的功能受到限制，使得在設計上比較困難。所以一般設計在 MANET 網路的演算法，通常以簡單有效為主，避免複雜的運算，以免因為這些運算拖累了整體的速度。

雖然資訊科技在硬體技術進步的速度十分迅速，在記憶體的空間上有大幅度的提升，但因體積的限制，儲存的空間依然無法

滿足過多的需求，所以要讓節點有效的利用這些記憶體空間，提升記憶體的效能，也是一個必須要考量到的問題。

#### (7) 無中心性：

MANET 網路中所有的節點地位平等，彼此透過各個演算法來協調彼此行為。

由於 MANET 網路這些先天上的限制，過去許多學者在有線環境討論的繞徑演算法便無法適用於這樣的環境中，例如大量消耗電力、浪費頻寬及記憶體等，都是目前 MANET 網路在研究上較受人矚目的議題。

## 1.2 相關研究

在這個章節，我們將介紹無線網路的一些相關研究，並且說明現有叢集演算法及繞徑演算法的一些問題。

### 1.2.1 傳統繞徑協定

在 MANET 網路通訊中拓樸資訊卻因為行動主機的移動而不斷的在變化，追蹤這些行動主機移動，有三個方法，第一種稱為繞徑表導向方式（Table-driven 或是 Proactive）的方法：如 DSDV（Destination-Sequenced Distance-Vector）[9,17]，每個行動主機間隔固定一段時間就會發送訊號，而整個網路上所有的行動主機就必須更新它們的路徑資料表。行動主機透過不斷的交流，而獲取最新的網路拓樸資訊，這種方法會造成行動主機負擔過重、網路頻寬擁塞及大量記憶體空間浪費，但這種方法的好處就是每個行動主機所記錄的路徑資

料都是正確且最新的，當有訊息需要傳遞路徑時，便可以馬上從記憶體中尋找，立刻建立連線。另一種方法稱為需求導向（On-demand 或是 Reactive）方法：如 AODV（Ad hoc On-demand Distance Vector）[21,27,30]，就是當節點需要路徑的時候才開始向外去尋找，正由於這個特性，將會有延遲的現象，缺乏即時性，不過，這種方法的優點就是不必一直更新記憶體內的繞徑表，也不需要太多的記憶體來儲存繞徑資訊。最後則是混合式(Hybrid):如 ZRP（Zone Routing Protocol）[6,14,15,21,36,37,38]，就是混合繞徑表導向與需求導向方法[41]。

#### 1.2.1.1 DSDV Protocol

每個行動節點間隔固定一段時間就會發送一些路徑相關資訊，各個行動節點就依據蒐集進來的資訊去改變自己的繞徑表（Routing table），可稱為繞徑表導向協定（Proactive routing protocol）。如 DSDV[9,30]、WRP[30]等，根據建立的繞徑表可以讓每個送出去的封包立刻得知到達目的地（Destination）的路徑，不會有任何的延遲。雖然路徑的錯誤率會降低，且沒有建立繞徑的延遲時間，但是在網路拓撲快速變化的環境下，這種協定必須週期性的去廣播訊息，所以相當浪費無線網路的頻寬與行動節點的電源，但是如果降低廣播所造成大量頻寬的消耗，就要拉長每次廣播的間隔時間，這又將會造成繞徑表不能正確反應網路拓撲的變化，所以只適合在 MANET 無線網路節點數少及網路拓撲變化不大時。

DSDV（Destination-Sequenced Distance-Vector Routing）[9,17]是基於傳統 Bellman-Ford 繞徑選擇演算法所改良而發展出來的一個以繞徑表為基礎的通訊協定。每一個行動節點必須儲存一個繞徑表，其中紀錄所有與該節點可能進行連結的節點及距離，繞徑表內的每筆紀

錄同時也包含了一個連續編號 (Sequence number)，這是用來判斷是否有些路徑比較老舊，以避免繞徑迴圈 (Routing loop) 的產生。

DSDV 基本上和 Internet 上的 Distance-Vector Routing 相同，只是多了目的端連續編號 (Destination sequence number) 的紀錄，使得 Distance-Vector Routing 更能符合 MANET 這種動態網路所需，此外，當網路拓撲變動比較不頻繁時，並不需要將繞徑表的所有資料進行交換。DSDV 在每個節點內再加了一個表格，用來記錄其繞徑表從上次交換至今所更改的部分，如果更改很多，就進行全部資料的交換，如果改變很少，就只針對改變部分交換。

#### 1.2.1.2 AODV Protocol

當行動節點需要傳送資料時，卻無法尋獲目的地的路徑，才會開始去運作的繞徑協定，也就是需求導向繞徑協定，這類協定的最大好處就是頻寬的使用量較小，只有當一個行動節點欲傳送封包，卻未能從繞徑表找到路徑時起用，因此平均延遲時間較長。例如 Dynamic Source Routing (DSR) [17,20, 21,30]、Ad hoc On-demand Distance Vector Routing (AODV) [21,27,30] 等就屬於這一類，又稱為需求導向繞徑協定 (On-demand routing protocol -Reactive)。

AODV [21,27,30] 利用 distance-vector 的概念，AODV 並不隨時維持一個繞徑表，而是當一個節點需要和另一個節點溝通時才以 On-demand 的方式建立繞徑表。

當一個節點想要傳送資料給網路中的另一個節點時，首先要廣播一個 Route Request (RREQ) 封包，RREQ 裡紀錄了這是哪一個來源端 (Source) 所發出，是要用來找尋哪一個目的端節點的。RREQ 在網路中是一種 flooding 的傳遞方式，會一直傳遞直到被目的端收到，當然，一個節點只能對同一個 RREQ 處理一次，AODV 的繞徑封包

也與 DSDV 一樣也紀錄了目的端連續編號，可以避免 Routing loop 的發生，同時也可以確保紀錄的繞徑表儲存了最新的網路拓樸。理論上所有來源端與目的端之間的節點都會被 RREQ 所經過，也會暫時性記下關於 RREQ 行經路徑的 last hop 資訊，當目的端收到來自不同地方的 RREQ 時，會選擇一條最短路徑，並往來源端方向送出 Route Reply (RREP) (如圖 1-5)。

隨著 RREP 的經過，沿路上的節點也會將這條路徑的有關資訊紀錄下來，當 RREP 被送到一開始送出 RREQ 的來源端時，這條從來源端到目的端的路徑就算被建立起來了，此後來源端都可以利用這條路徑送封包給目的端。(如圖 1-5)

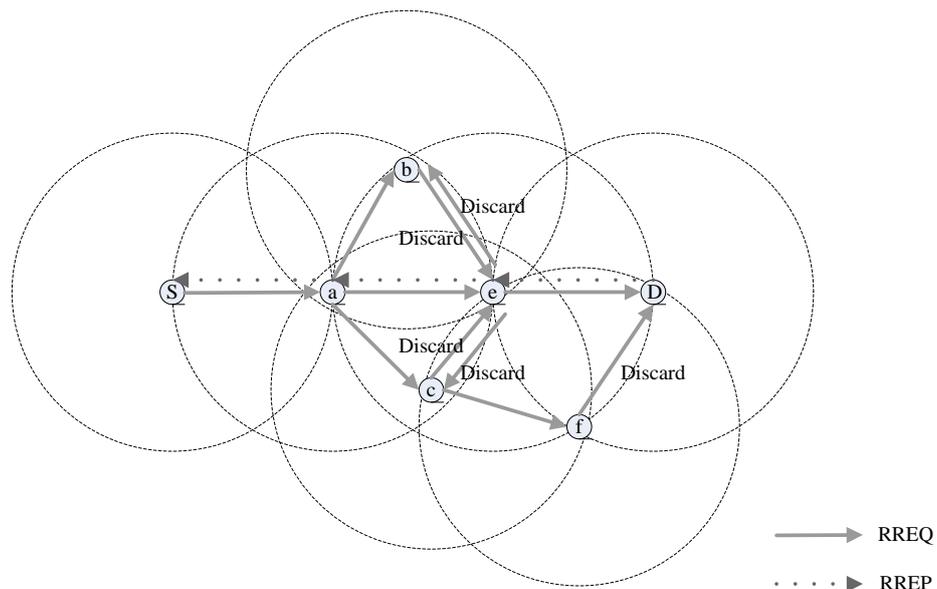


圖 1-5、AODV 協定

### 1.2.1.3 ZRP Protocol

混合 (Hybrid) 導向繞徑協定是結合前面繞徑表導向及需求導向的繞徑方法，運用 MANET 網路的特性，在群集內及群集間使用不同的繞徑策略，各取其優點，沒有過大的繞徑流量控制、延遲時間及路

徑錯誤率，適合 MANET 網路在節點數過多及網路拓撲變化大時使用。

ZRP (Zone Routing Protocol) [6,14,15,21,36,37,38] 結合了繞徑表導向方式與需求導向兩種路徑建立方式，讓網路裡的一個節點隨時紀錄附近節點的繞徑資訊，另一方面如果要和較遠的節點連結時，只讓網路裡少數節點參與繞徑。當一個節點要和鄰近節點溝通時，則可以透過紀錄立刻得到路徑資訊，而當紀錄中沒有該節點的資訊時，則僅有部分節點參與繞徑，可減少網路資源的浪費。ZRP 定義一個節點 S 鄰近幾個 hop 距離以內的所有其他節點，成為 S 的繞徑區域 (Routing zone)，由於每個節點都有自己不同的相鄰節點，因此在 ZRP 裡，整個網路就是許多重疊的繞徑區域。在同一個區域 (Zone) 裡的繞徑稱為區域內繞徑 (Intrazone routing)，屬於一種 Distance vector routing，主要是同一區域的節點交換 Distance vector，且所有 Distance vector 都被限制在區域的範圍內 (如圖 1-6)。

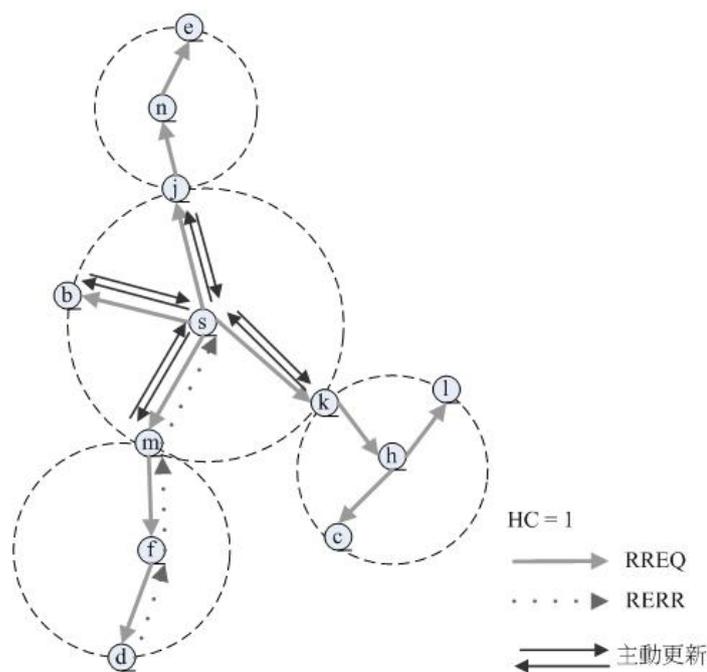


圖 1-6、ZRP 1-hops 區域示意圖

當一個來源端要尋找一個不在其繞徑區域內的目的端時，來源端利用 Broadcasting，也就是將繞徑查詢 (Route query) 送給在來源端的繞徑區域的邊界節點，透過邊界節點檢查所查詢 (Query) 的目的端是否在其繞徑區域內，如果沒有，就繼續進行 Broadcasting，如果有，就可以回應 (Reply)，表示已經找到可以連到目的端的路徑。這樣的過程跨越了多個區域，因此稱為區域間繞徑 (Interzone routing)。

### 1.2.2 Cluster Based Routing Protocol

在 MANET 網路中，由於行動主機任意移動的特性，加上沒有基地台的支援，所以行動主機本身就扮演一個移動的基地台角色，因此必須紀錄一些網路拓樸資訊來維持路徑的暢通。網路上將多個特性相同的節點劃分在一起，即是所謂的 Cluster，並從叢集中選出叢集標頭，由它來負責管理叢集裡的成員和交換資訊，並記錄整個叢集的繞徑資訊。對叢集標頭來說，必須儲存叢集內的所有資訊，但對於非叢集標頭的節點而言，就不必記錄這些資訊，因此，整個網路所承載的總資料量就大幅的減少。

叢集的繞徑方法，分為叢集內及叢集外，當訊息的傳遞是在叢集內發生時，叢集內的成員只要發送訊息給叢集標頭，則叢集標頭便會將路徑傳遞予它，反應速度快。當所要傳遞的節點不在叢集內時，亦透過叢集標頭向外廣播，尋獲路徑後再傳遞給叢集成員。雖然叢集的運用具有相當多的優勢，但還是有一些缺點，在叢集的架構下，由於叢集標頭負責較多的工作，需要管理叢集內的節點並且維護繞徑資訊，因此負擔較重，也需要運用更大的記憶體空間及電力。

叢集架構的特性具有防止網路封包氾濫的功能 (如圖 1-7)，屬於叢集內的封包不會氾濫到整個網路中，所有傳遞的網路資料，都由所

屬叢集的叢集領導者來記錄，就像把叢集標頭當成有線基地台，因此整個網路所需要紀錄的資料量便可以大幅減少。如圖 1-8 中將網路拓樸劃分成數個叢集，每個叢集選出叢集標頭，賦予它維護叢集成員及叢集繞徑資訊的責任，其餘非叢集標頭節點只記錄鄰居節點的資訊，繞徑表只要傳給叢集標頭，不用再傳給其他節點，如此一來就可以減少傳輸繞徑表時所造成的資料氾濫。叢集架構的另一個好處，就是叢集內的拓樸改變時，只有叢集標頭需要更改繞徑表的記錄，而其他叢集及節點則不受影響。

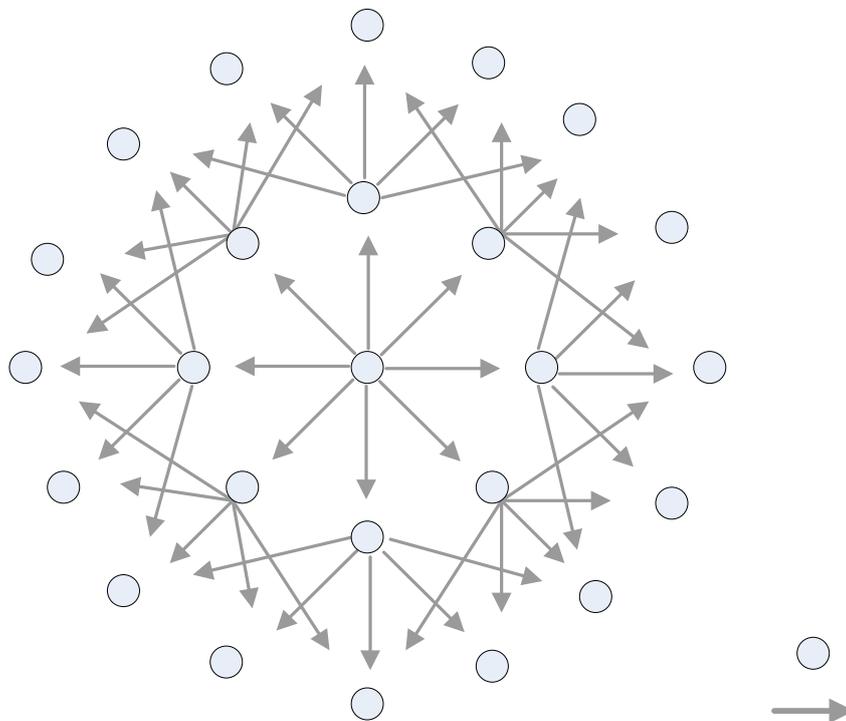


圖1-7、網路封包氾濫

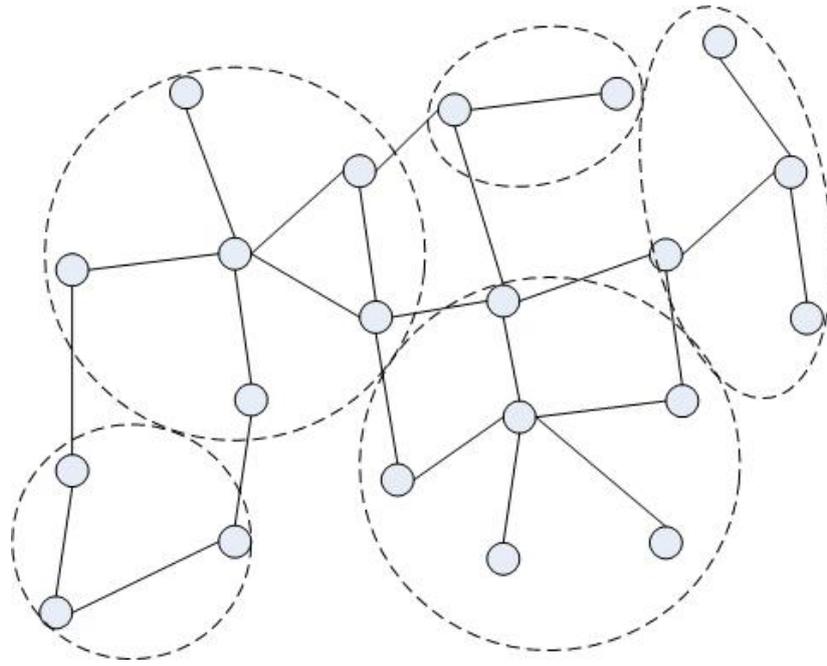


圖1-8、叢集架構

在叢集的劃分方法上，主要有最低識別碼叢集法 (Lowest ID clustering, LIDC) [2,8,23,28,31,40]、高連結度叢集法(Highest degree or High connectivity clustering, HCC)[12]以及權重叢集法[10]，上述的方法所建構出的叢集架構主要都是 1-hop 的叢集結構，也就是叢集標頭只負責管理一個跳躍節點 (1-hop) 之內的成員 (如圖 1-8)。在網路規模較大的情況下，1-hop 的叢集架構，其叢集的數目可能變多而使得叢集的優點無法呈現，因此有一些學者提供以 k-hop [8,39] 的方式來架構叢集的演算法，以應付較大網路的需求，並提供調整的彈性。下面將分別說明這些方法的差異性[41]。

#### 1.2.2.1 最低識別碼叢集法

由於每一個行動主機都有一個固定唯一的識別碼 ID，最低識別碼叢集法就是利用這種特性來劃分叢集，以 ID 的數字最小的節點當

叢集標頭，每個節點都和自己相鄰的節點比較 ID 值，如果四周相鄰節點的 ID 都比自己大，則自己就成為叢集標頭，其相鄰點就成為所屬的成員。利用節點 ID 的編號來找出叢集標頭這個方法的好處就是既簡單且快速，但缺點就是劃分來的叢集沒有條理可言，對於網路拓樸改變時的調適性不佳，而且極容易產生叢集個數很多的情況。如圖 1-9 中節點 1 ID 最小，管轄節點 2、9，節點 3 管轄節點 8、10，節點 4 管轄節點 7，節點 5 管轄節點 6。

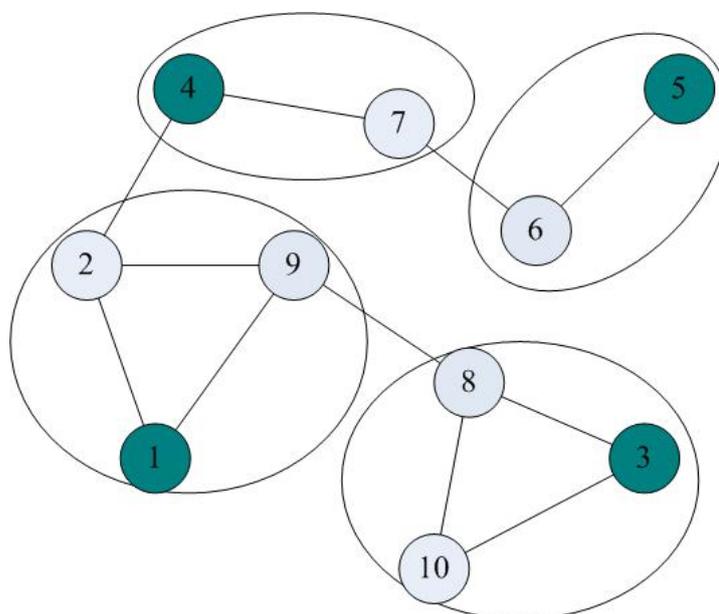


圖 1-9、最低識別碼叢集法

#### 1.2.2.2 高連接度叢集法

由於最低識別碼叢集法所劃分出來的叢集較無意義，所以有學者提出了以節點連結度 (Degree) 的多寡來決定叢集領導者的方式。在這個高連結度叢集法演算法[12]中，節點是利用自己的連結度 (如圖 1-10 中節點旁的鄰居數量) 來作為競爭叢集管理者的依據，利用這種劃分的方式是想最大化叢集管理者管理的節點數，因此叢集的個數也會隨之相對減少，就可以使得整個網路維護的資訊量降低。但是由於

叢集管理者是利用連結度去競選出來的，所以當有節點消失或是移動，造成拓樸改變時，常常會造成整個網路必須重新進行叢集的形成，也就是網路的穩定性不高。而且雖然這個演算法希望最大化叢集管理的節點數，但卻也容易形成（如圖 1-10）右下邊那種只有一個節點的叢集，反而使得整體效率降低。

如圖 1-10 中節點 9 的連結度都為 4，管轄節點 2、4、7、12，節點 8 的連結度為 3 管轄節點 1、10、11，節點 6 的連結度為 3，管轄節點 3、5。同理。高連結度叢集法相較於最低識別碼叢集法是更佳的方法，因為增加叢集領導者管理的節點數，則叢集個數就會減少，叢集減少則整個網路維護的資訊量也會降低，而且對於網路拓樸改變時有更多的彈性，不會因為突然出現 ID 較小的節點而使得叢集領導者變換身分。

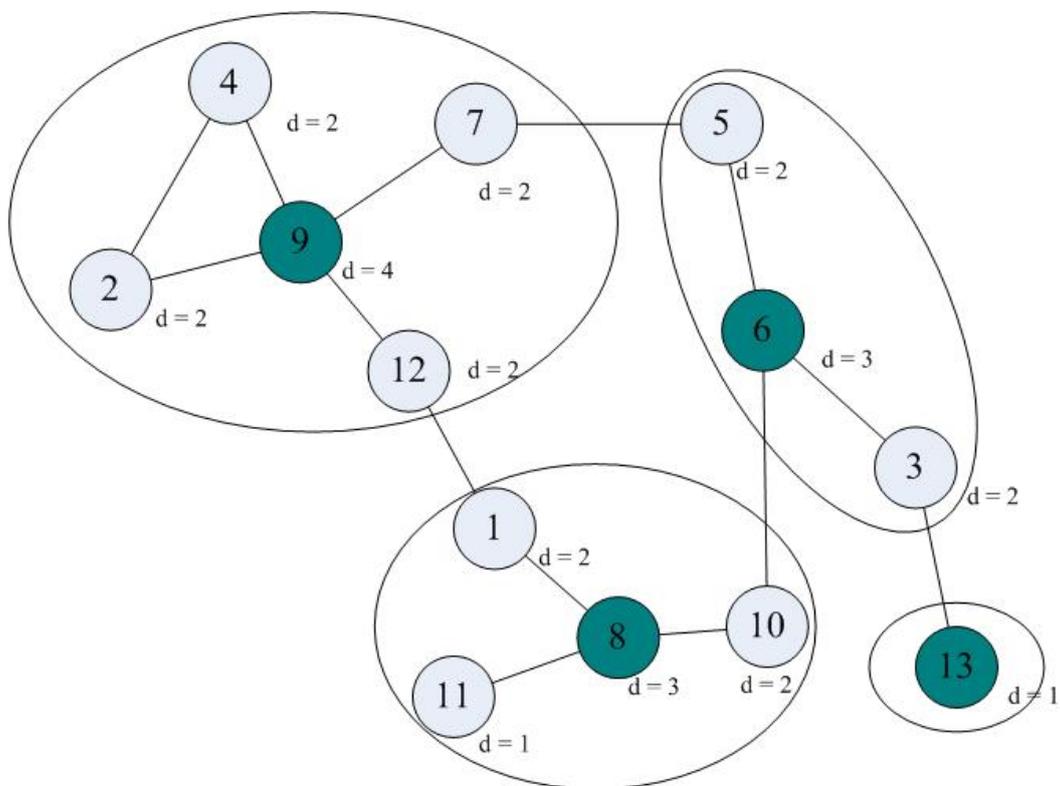


圖 1-10、高連結度叢集法

### 1.2.2.3 最少叢集變化演算法

無論是最小識別碼叢集或是高連結度叢集方法，都有一個嚴重的問題，也就是缺乏叢集領導者退休的機制，也就是說，隨著時間的變動，叢集領導者的數目將會一直上升，且無法下降，因此必須在系統執行一段時間之後，再重新執行叢集演算法。這樣的作法將使得網路上所有的節點狀態發生改變而使得效能下降，因此 Ching-Chuan Chiang 及 Mario Gerla 提出了最少叢集變化法 (Least Cluster Change Algorithm, LCC) [8]，這個方法是使用在叢集的維護部分，叢集的產生部分可以使用任何一種方式來產生。LCC 演算法能使最小識別碼和高連結度叢集法減低其叢集變動率，其詳細的規則如下：

- Step1. 初始狀態時，可使用最小識別碼或高連結度方法來架構叢集。
- Step2. 當一個非叢集領導者  $i$  移動到叢集  $j$  的範圍內時，叢集領導者  $i$  或  $j$  並不需變動。
- Step3. 當一個非叢集領導者  $i$  移出了自己的叢集外，且無法加入別的叢集時，則自己形成叢集領導者。
- Step4. 當叢集領導者  $i$  移入叢集  $j$  時， $i$  將挑戰  $j$  的地位， $i$  和  $j$  將有一者會退休成為一般點，其中挑戰方式可依據最小識別碼或高連結度任一方式。
- Step5. 當某叢集領導者退休後，其成員將依據最小識別碼或高連結度方式重新加入其他叢集。

#### 1.2.2.4 權重叢集法

由上述的叢集演算法可知，叢集演算法最重要的考量參數，就是在於選取叢集領導者時所考量的因素。因此，若將此考量因素視為一權重值，將所欲參考的因素以數值加權的方式呈現，便可形成不同的叢集演算法架構，這便是所謂的權重叢集法（Weighted Clustering Algorithm）[16]。例如前面所提的最低識別碼叢集法，就是以節點的ID為主要權重考慮參數所形成的。而高連結度叢集法則是將節點的連結度視為主要權重。所以我們可以將所需考慮的因素參數化，將其視為一權重函數，便可透過加權不同考慮因素而產生叢集演算法，Mainak Chatterjee 等人所提出的 On-demand 權重叢集演算法[16]，便是以這樣的前提為考量所設計的。此叢集演算法所考量的因素包含了點的連結度、傳輸半徑、移動特性以及電池能量等，將以上的因素視為一函數運算比較後，便可得鄰居中最適合擔任叢集領導者的點。

#### 1.2.2.5 k-hop叢集法

在 k-hop 叢集演算法(K-hop Clustering Algorithm)部分，Zygmunt J. Hass, Marc R. Pearlman [14,15,39] 和 Dongkyun Kim, Seokjae Ha [3] 所使用的叢集演算法十分接近，其作法是將每一點皆視為叢集領導者，讓網路中每個點都必須紀錄其周圍 k-hop 之內的資訊，此叢集的重疊度相當高，而 Taek Jin Kwon [23] 所使用的方式則是將原本的最低識別碼及高連結度叢集法延伸至 k-hop 架構以適應較大型的網路環境。

### 1.3 A New Cluster Protocol Based on Efficient-Energy (CPEE)

本文提出以叢集路由協定 (Cluster route protocol) 為基礎的有效能源 (Efficient-Energy)。我們將每個節點的能源和鄰近節點數量作為選擇叢集標頭 (Cluster Head) 的因素，利用權重 (Weigh) 機制來建立叢集，要成為叢集標頭必須是在 2-hop 的 Weigh 為最大 (圖 1-11)。後來將利用叢集內所以成員的權重值，來計算維護標頭叢集機制。當叢集標頭的權重值低於叢集內平均權重值時，叢集標頭必須解除他的身份。這樣的機制會讓叢集標頭壽命延長，使得無線網路不會被分割。在這樣的無線網路環境，可以提升網路效能。

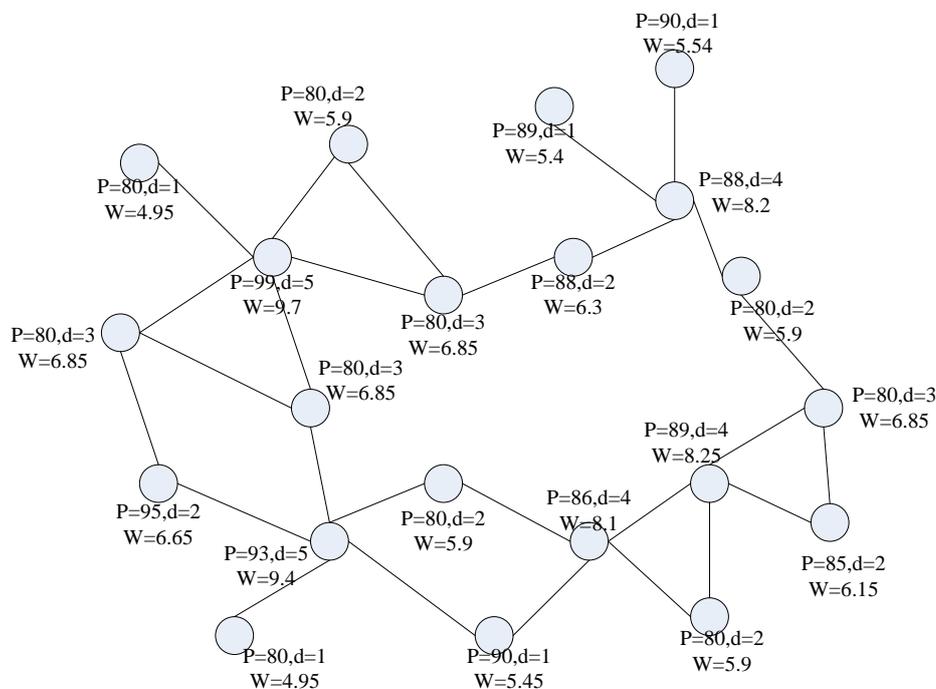


圖 1-11、有效能源叢集演算法

模擬實驗結果，本文中所題出的新叢集協定基於有效能源演算法，在 MANET 網路可以延長 40% 至 88% 的平均網路壽命。MANET 網路整體的效能提升，使得 MANET 網路達到較佳的叢集效能及減少能源快速消耗。

## 1.4 論文架構

剩下有關本篇論文的部份說明如下：第二章節將說明我們所提出的 CPEE 所要解決的叢集問題；第三章節會詳細的介紹我們所提出的 CPEE 演算法及維護叢集方法；第四章節會呈現模擬實驗的結果及分析圖表；最後的章節，對我們本篇研究的結論。



## 第二章 問題描述

由於 MANET 網路裡每個行動主機具有可移動的特性，以及有限的無線廣播涵蓋範圍和有限的電力等因素，可能會因為行動主機的移動造成網路拓樸的改變，使得原本建立好的叢集發生叢集成員改變和路徑發生斷裂的情形。MANET 網路在沒有像是基地台的集中式管理中心的環境，並且無法知道整個網路拓樸的關係，通常繞徑協定所使用的方式此以全面性的氾濫式廣播技術（如圖 1-7），將 RREQ 封包傳遞出去，此種方式能夠將所有可能通訊的路徑建立出來，但只會選擇一條是最快到達目的地節點的路徑來回傳 RREP 封包；但這樣的方式，極度的可能造成頻寬的浪費甚至造成網路的癱瘓，所以有學者提出許多相關的方法來解決這樣的問題。

首先，我們假設 MANET 的網路環境用  $G=(V, E)$  符號來表示，其中  $V$  代表網路上所有節點的集合，而每一個節點  $i (v_i)$  的廣播範圍大小為  $l_i$ ，在模擬實驗的部份我們假設所有的  $v_i$  都是公平的，所以  $L$  都設定為相同大小； $E$  表示節點與節點之間的連線數。如果  $v_i$  在  $v_j$  的傳輸範圍內，則存在一個邊緣  $e = (v_i, v_j) \in E, v_i, v_j \in V$ 。在  $G$  圖裡所有的連接邊緣是雙向 (bi-directional)。例如， $v_i$  在  $v_j$  的傳輸範圍內，亦則  $v_j$  也在  $v_i$  的傳輸範圍內，二者可以互相連接通訊。我們假設這個網路是彼此有互相連接的狀態。如果被分割，則視為一個獨立的網路。在這個網路裡廣播封包的長度是固定的。對於網路上每一個節點  $i (v_i)$  的鄰居節點集合可以用  $N(v_i) = \{v_j | (v_j, v_i) \in E\}$  來表示，節點  $i (v_i)$  的鄰居節點的鄰居集合可以用  $Thops(v_i) = \{v_k | (v_k, v_j) \in E, v_j \in N(v_i)\}$  來表示。而 Cluster

內成員的集合可以用  $|C_i| = \{N(v_i) \cup Thops(v_i)\}$  來表示，則  $v_i \in C_i$ 。在 Route Discovery 過程中的所有轉送 RREQ 封包的節點可以用  $F(v_s, v_d) \subseteq V$  來代表。

在 MANET 的網路環境裡，並沒有基地台的存在，它是透過節點之間的連線 ( $e$ ) 做為訊息傳遞的方式，也就是網路任意一個節點  $i$  ( $v_i$ ) 必需透過它的  $N(v_i)$  以及其他的  $F(v_s, v_d)$  來傳遞控制封包或是資料封包，所以節點之間必須互助合作，尋找與目的地節點 ( $v_d$ ) 之間的資料傳輸路徑。即使先前的 Cluster 協定已經改善了氾濫式廣播的問題，但是在選擇叢集時，仍然有更好的方式存在。透過本章節的說明，我們將描述一般叢集協定下的問題，並針這些問題來進行改善。

本論文裡，我們提議一個在 MANET 網路下，叢集協定基於有效能源方法。透過使用這個演算法，每個節點會廣播一個 Hello Message 封包。當鄰居回傳此訊息時及可知道鄰居數量時，則利用鄰居數量與他自己的能源來計算權重值 (如圖 2-1)。當權重值愈大，表示在無線網路裡，可以提供愈多鄰居與傳輸時間。在這樣的方式，可以減少能源消耗與網路頻寬浪費。後來在將每個節點的權重值廣播給鄰居節點。每個節點會決定他們自己的身分為叢集標頭或叢集標頭的成員。在叢集建立前節點的權重值是大於 2-hop 的鄰近節點時，則成為叢集標頭，反之則加入叢集裡，成為叢集標頭的成員 (如圖 2-2)。

在叢集標頭維護時，叢集標頭必須利用他的能源值來決定是否解除叢集標頭的身分，是依據 *removeCH\_threshold* 計算機制來維護叢集標頭。當叢集標頭的權重值低於叢集內平均權重值時，叢集標頭必須解除他的身份。這樣的機制是可以延長無線網路壽命，而不會讓無線網路分割。在這樣的無線網路環境，以提升無線網路效能。

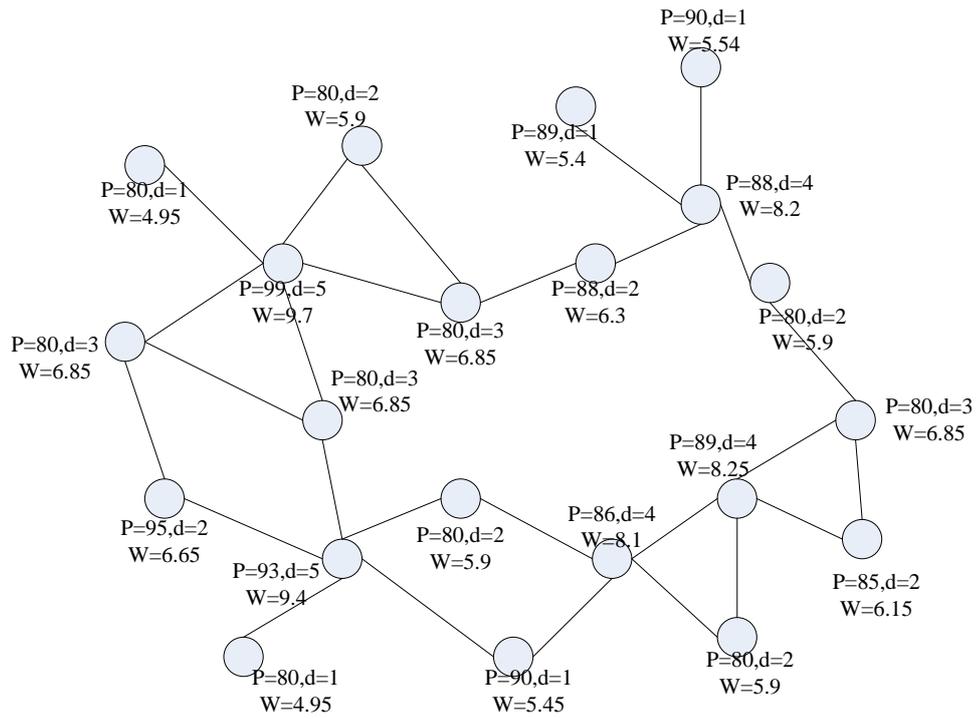


圖 2-1、每個節點計算權重值

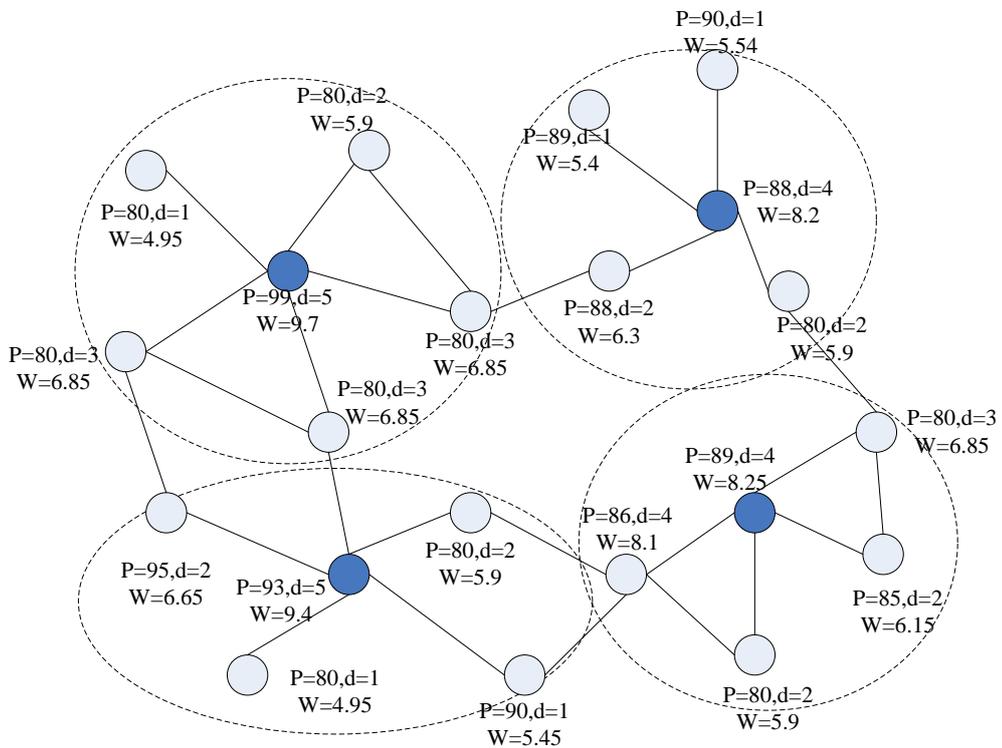


圖 2-2、選擇叢集標頭 (1-hop)

## 第三章 研究方法

我們提議這個演算法，在高密度的網路使用 2-hops 叢集技術，可以有效的減少多餘的封包或不需要的封包。而在此演算法裡加入能源因素考量，使得整個網路不會被分割 (Separation)，且讓網路和每個節點的存活時間延長。在路由協定方面採用多路徑 (Multitpath) 方式，路由方式有主要路由 (Main route) 和候選路由 (Candidate route)，提供路徑斷裂時，可以快速啟動備用路徑。以上幾個優勢可提升網路整體的效率。

### 3.1 叢集路由協定基於有效能源

我們先定義演算法中用到的叢集狀態，再來說明詳細的步驟：

- 叢集標頭 (Cluster head)：叢集內的領導者，負責管理自己叢集內的節點。
- 閘道節點 (Gateway node)：在叢集的架構中，閘道節點是相當重要的一個節點，負責與其他叢集溝通通訊之用。
- 連接鄰近叢集節點 (Joint adjacent cluster node)：若節點可連接到其他叢集的成員，則該節點便為叢集之連接鄰近叢集節點。
- 叢集成員 (Cluster member)：在叢集內為一般成員。

#### 3.1.1 建立叢集演算法

起初網路內節點呈隨機排列，而網路裡的每個節點開始的狀態 (State) 設定為 Unassigned。其二、執行有效能源叢集演算法程序，

依照以下步驟執行。

**Step 1.** 每個節點會廣播一個 Hello Message 封包給他的鄰近節點，這 Hello Message 包含：節點 ID，當鄰居回傳此訊息時及可知道鄰居數量。這些接收到 Hello Message 封包的接收人，可利用接收到的訊息來更新他們的資訊如：鄰居數量 (Number of neighbor)。

**Step 2.** 假設已知鄰居數量，則可以計算權重值 (公式 1)。  $D_i$  :  $v_i$  的鄰居數量。  $P_i$  :  $v_i$  的能源電量 (百分比)。此權重值在叢集標頭成立前每個節點會計算好，使用下面公式：

$$\text{Weigh\_value}_i = \alpha * D_i + (1 - \alpha) * P_i \quad (\text{公式 1})$$

其中

$\text{Weigh\_value}_i$  : 每個  $v$  的權重值

$\alpha$  :  $0 \leq \alpha \leq 1$ , (分配比率值)

$D_i$  :  $v_i$  的鄰居數量

$P_i$  :  $v_i$  的能源 (百分比)

計算權重值所須要的因素包含鄰近節點的數量和節點能源電量 (Energy coulometer of node)，這是選擇叢集標頭是很重要的因素。這代表的是鄰近節點的數量和節點能源電量愈大，則可連接的鄰居愈多和在網路存活時間愈長，所以這二個因素愈大愈好。每個節點會互相競爭成為叢集標頭，成為叢集內的管理者。

**Step 3.** 計算完成後每個節點會將自己的權重值廣播給他的鄰居

節點。相對的，自己也會接收到鄰居節點廣播的權重值。

**Step 4.** 互相比較權重值，當他的權重值比鄰居節點的權重值大時，鄰居節點會儲存最大權重值的 ID，並且幫忙轉播 (Retransmission) 給其他與鄰居節點相鄰的節點。當在 2-hop 的範圍內具有最大的權重值，就能成為叢集標頭。

**Step 5.** 選出叢集標頭時，叢集標頭會廣播資訊給 1-hop 和 2-hop 的節點，邀請鄰近節點加入此叢集。如果 1-hop 和 2-hop 的節點已加入其他叢集時，則 1-hop 和 2-hop 的節點會廣播一個已加入另一叢集封包，且拒絕加入其他叢集。鄰近節點加入叢集之後，便成為此叢集成員。在叢集標頭的 1-hop 節點稱為內部成員 (Internal member) 和 2-hop 節點稱為邊界成員 (Bound member)，決定他們在叢集裡的狀態。每個叢集成員會廣播自己的資訊給叢集標頭，叢集標頭會建立維護表 (Maintain table) 資訊，存放此叢集標頭內的每個成員資訊 (如表 3-1)。當叢集建立完成時，必須計算  $removeCH\_threshold$ ， $C_k$ ：叢集內節點數量。 $\sum_{i=1}^n Weigh\_Value_i$ ：叢集內  $v$  的權重值，存於叢集標頭內的維護表資訊， $\sum_{i=1}^n Weigh\_Value_i \in C_k$ 。此  $removeCH\_threshold$  在叢集標頭成立時已計算好，使用下面公式 2：

$$removeCH\_threshold = \frac{\sum_{i=1}^n Weigh\_Value_i}{C_k} \quad (\text{公式 2})$$

其中

*removeCH\_threshold* : 解除叢集標頭的門檻值

$\sum_{i=1}^n Weigh\_Value_i$  : 叢集內所有節點的權重值

$C_k$  : 叢集內節點數量

並且邊緣成員 (Border member) 會再次廣播給沒加入叢集的節點，請他們在去找權重值是大的，去加入其他的叢集。邊緣成員就是較接近叢集邊緣附近的節點，並且邊緣成員也有可能連接到鄰近的叢集。詳細建立叢集演算法如圖 3-1。

舉例如圖 3-2、假設以節點 5 廣播一個 Hello Message 封包，鄰近節點收到 Hello Message 封包會回傳訊息給節點 5。接收到鄰近節點回傳訊息後，節點 5 也就知道他的鄰居數量。相對的，每個節點都會知道他們自己的鄰居數量。節點 5 則計算他的權重值 (公式 1)。計算完後則將權重值廣播給鄰居節點 3、4、7、8。節點 5 也會接收到鄰居節點 3、4、7、8 所廣播的權重值。那麼節點 3、4、7、8 的權重值都小於節點 5 的權重值，則節點 3、4、7、8 幫忙轉播節點 5 的權重值給其他鄰居節點 1、2、6、9、10。假設鄰居節點 1、2、6、9、10 的權重值也小於節點 5。節點 5 的權重值是在 2-hop 內為最大，所以節點 5 成為叢集標頭。

當節點 5 成為叢集標頭時，他會廣播訊息給他的鄰居節點 (1-hop) 及其鄰居節點 (2-hop)，邀請他們加入節點 5 的叢集。鄰近節點加入叢集之後，便成為此叢集成員。在叢集標頭的 1-hop 的節

點 3、4、7、8 稱為內部成員和 2-hop 的節點 1、2、6、9、10 稱為邊界成員，決定他們在叢集裡的狀態。每個叢集成員加入時，會廣播自己的 ID、Hop count 及 power 資訊給叢集標頭。叢集標頭會建立維護表資訊，存放此叢集內成員的資訊（如表 3-1）；如果節點 14 為叢集標頭時，節點 14 也會邀請節點 9 加入節點 14 叢集。但節點 9 已加入節點 5 為叢集標頭的叢集，則節點 9 會廣播一個已加入其他叢集封包，且拒絕加入節點 14 為叢集標頭的叢集。當建立叢集表後，必須計算 *removeCH\_threshold*，並且邊緣成員（節點 1、2、6、9、10）會在廣播給沒加入叢集的節點，請他們在去找權重值是大，去加入其他的叢集。詳細解除叢集標頭演算法如下。

當解除叢集標頭的身分時，他的權重值是小於 *removeCH\_threshold*（詳細說明在往後描述 3.2.3 叢集標頭維護）。

### Alg.1 Cluster Algorithm

$v_i$  :  $i$  th node

$n_i$  :  $i$  th neighbor

$v\_Weigh_i$  :  $i$  th Weigh value of node

Neighbor\_Weigh $_i$  :  $i$  th Weigh value of neighbor

=====  
**Step 1.**Each node broadcast Hello Message packet(Node ID).

**Step 2.**Received Hello Message packet, then compute its  $Weigh\_value_i$  user (公式 1).

**Step 3.**After they computation completes. Once more broadcast its  $Weigh\_value_i$  for neighbor node, and its received neighbor node  $Weigh\_value_i$ .

**Step 4.**Each other compare they  $Weigh\_value_i$ .

```
If (Neighbor_Weigh $_i$  < v_Weigh $_i$ ){  
  n $_i$  retransmission its neighbors for v $_i$   
}  
Else{  
  v $_i$  retransmission its neighbors for n $_i$   
}
```

**Step 5.**When its  $Weigh\_value_i$  is Max, then neighbor broadcast for neighbor.

```
If (v_Weigh $_i$  > 1-hop Neighbor_Weigh $_i$ )&&  
  (v_Weigh $_i$  > 2-hop Neighbor_Weigh $_i$ ){  
  v $_i$  = Cluster Head
```

Invitation neighbor node of 1-hop and 2-hop increases to cluster inside

```
If (neighbor node is member of other cluster){  
  broadcast a refuse invitation.  
}
```

```
Else{  
  Internal member = 1-hop  
  Bound member = 2-hop  
  1-hop and 2-hop node broadcast information for Cluster Head,  
  Information contain "Node ID", "Power value", "Hop count".  
  Establish Cluster Table for Cluster Head,  
  then computation  $removeCH\_threshold$  .  
  And Border member broadcast for other neighbor of unjoint cluster  
}
```

```
}  
Else Re-broadcast Max of  $Weigh\_value_i$  for its neighbors
```

圖 3-1、建立叢集演算法

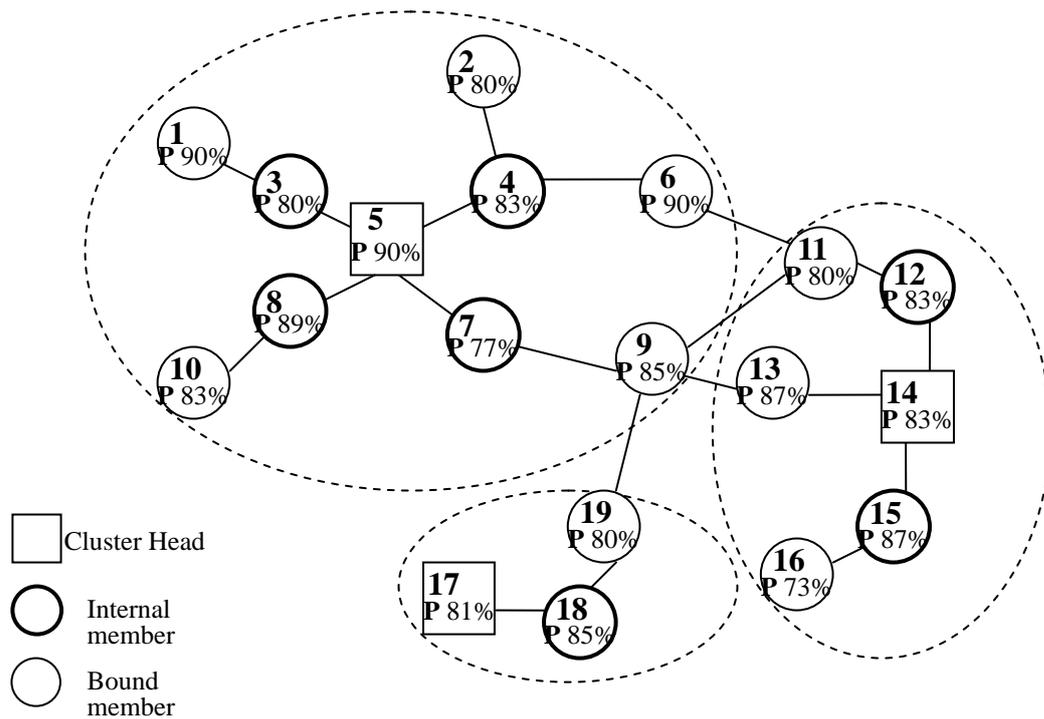


圖 3-2、2-hop 叢集環境

表 3-1、成立叢集時節點 5 建立的鄰居表

Neighbor node's ID	Neighbor ID
3 (80%)	1 (90%),5 (90%)
4 (83%)	2 (80%),5 (90%),6 (90%)
7 (77%)	5 (90%),9 (85%)
8 (89%)	5 (90%),10 (83%)

### 3.2 叢集管理

建立叢集後，如何管理叢集是相當重要。在這裡說明如何讓網路的效能提升。無線網路裡使用相同的叢集演算法，但使用不同的選擇開道節點、傳輸方法及叢集維護。這也會影響整個網路的效能。所以

叢集管理是相當重要的，叢集管理包含選擇閘道節點方法、叢集之間的傳輸方法及叢集如何維護。詳細考量叢集與叢集之間的每一個處理流程，將整個無線網路效能提升到最佳狀態。

### 3.2.1 選擇閘道節點與連接鄰近叢集節點方法

在叢集裡要選擇閘道節點是相當重要的。邊緣成員是最有可能成為閘道節點，因為他們比較接近鄰近叢集。當邊緣成員有連接到另一個叢集時，我們稱此成員為連接鄰近叢集節點。而這些連接鄰近叢集節點會被叢集標頭選為閘道節點，但要成為閘道節點是有二種類型。其一、在一個叢集內，只有一個節點可以連接到另一個叢集時，則此節點為閘道節點。其二、在一個叢集內，有二個或二個以上節點可連接到另一個叢集，則此叢集標頭會選擇邊緣成員權重值（公式 3）為最大的成為閘道節點。成為閘道節點時，必須廣播他的鄰近叢集節點資訊給叢集標頭。 $LTACDeg_i$ ：邊緣成員  $v_i$  的連接鄰近叢集成員數量 (Number of Link To Member of Adjacent Cluster)。 $P_i$ ：邊緣成員  $v_i$  的能源電量 (百分比)。此邊緣成員權重值 ( $Border\_Weigh\_value$  ;  $BWV$ )，是當邊緣成員有接收到鄰近叢集廣播的訊息時，邊緣成員會請求叢集標頭，由叢集標頭來選擇  $BWV$  為大的成為閘道節點，使用下面公式：

$$BWV = \alpha * LTACDeg_i + (1 - \alpha) * P_i \quad (\text{公式 3})$$

其中

$BWV$ ：邊緣成員的權重值

$\alpha$ ： $0 \leq \alpha \leq 1$ , (分配比率值)

$LTACDeg_i$  : 連接鄰近叢集節點數量

$P_i$  : 節點的能源(百分比)

**Step 1.** 邊緣成員有接收到鄰近叢集廣播的訊息時，則此節點為連接鄰近叢集節點，並且會廣播一個請求訊息給叢集標頭。連接鄰近叢集節點在廣播請求訊息前，必須先計算好他的邊緣成員權重值。此邊緣成員權重值是包含他自己的能源與鄰近叢集數量的因素來計算此權重值。每個連接鄰近叢集節點廣播請求訊息時，此邊緣成員權重值會一起廣播給叢集標頭。

**Step 2.** 當叢集標頭接收到請求訊息時，會決定閘道節點。如果叢集標頭只接收到一個連接鄰近叢集節點的請求訊息時，則此節點為一個閘道節點。如果叢集標頭有接收到二個或二個以上的連接鄰近叢集節點的請求訊息時，則選擇邊緣成員權重值為最大的，則可成為閘道節點。成為閘道節點時，必須廣播他的鄰近叢集節點資訊給叢集標頭。

**Step 3.** 如果二個閘道節點有覆蓋相同區域時，則  $BWV$  小的閘道節點必須在他的鄰居表內有覆蓋相同區域的節點停止廣播訊息。除非他是可以連接到不同叢集區域，則可對另一區域廣播訊息。

**Step 4.** 如果有新的叢集接近時，則重新執行 **Step 1.**。詳細選擇閘道節點演算法如圖 3-3。

這個選擇閘道節點的方法，在 MANET 網路環境下，可以減少許多叢集往外廣播封包，顯然的控制封包會降低。在模擬實驗時，在閘道節點方面，我們都採用相同的選擇閘道節點方法。

### Alg.2 Choose Gateway Node Algorithm

**Step 1.** Border member has received Broadcast of adjacent Cluster.

Border member becomes joint adjacent cluster node,  
and broadcast request for Cluster Head

**Step 2.** When Cluster Head received request of border member,

then Cluster Head choice Max  $BWV$  becomes Gateway node.

**If**(Cluster Head Received of Border member the request  $\geq 1$ ) {

**If**(Number of Cluster received request  $\geq 2$ ) {

Cluster Head choice Max  $BWV$  of border member

}

Border member = Gateway node

then broadcast adjacent cluster information for Cluster Head.

Information contain "Neighbor ID", "Neighbor of Cluster Head".

}

**Step 3.** If two gateway have recover area, gateway the  $BWV$  is smaller ,

then table have recover the deletes.

**Step 4.** If has the new Cluster to approach, then return **Setup 1**.

圖 3-3、選擇閘道節點演算法

舉例如圖 3-4、當鄰近的叢集接近他們（節點 6, 9）的傳輸範圍時，有連接到另外的叢集節點成為連接鄰近叢集節點。而節點 6, 9（連接鄰近叢集節點）會計算  $BWV$ ，並且廣播訊息給他的節點 5（叢集標頭）。叢集標頭會選擇  $BWV$  最大的成為閘道節點。叢集標頭選定節點 9 為閘道節點，而節點 9 必須廣播他鄰近叢集的資訊給叢集標頭（如表 3-2）。

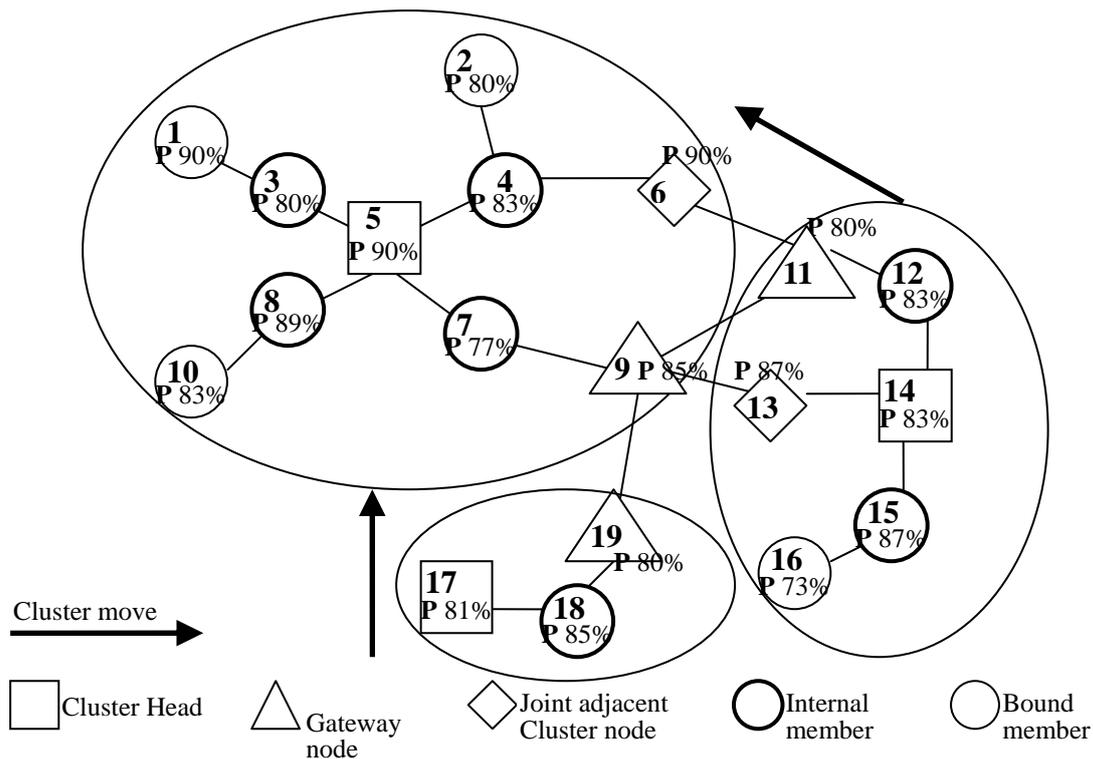


圖 3-4、叢集標頭選擇閘道節點

假設節點 6 的  $BWV$  是最大的，則節點 6 成為閘道節點，並且廣播鄰近叢集資訊給叢集標頭（如表 3-3）。另外節點 9 可連接到鄰近的叢集，此叢集是沒有被節點 6 覆蓋。這樣的情況下，節點 9 只需要對沒被節點 6 覆蓋的區域做傳輸。叢集標頭將接收到節點 9 廣播的鄰近叢集資訊。而叢集標頭會將節點 6 覆蓋的區域刪除（如表 3-4）。刪除的連接不是被實際刪除，而是暫時成為無效的連接。如果有新的叢集接近節點 6、9 或其他邊界成員的傳輸範圍時，則  $BWV$  會重新計算。在選出  $BWV$  最大的成為閘道節點，另外沒被覆蓋的叢集區域，則由其他閘道節點傳輸（如圖 3-4）。

表 3-2、節點 9 成為閘道節點時，廣播給叢集標頭的訊息

Neighbor of Adjacent Cluster	Neighbor of Cluster Head
11,13	14
19	17

表3-3、節點6成為閘道節點時，廣播給叢集標的鄰近叢集資訊

Neighbor of Adjacent Cluster	Neighbor of Cluster Head
11	14

表3-4、沒被覆蓋的區域由節點9傳輸

Neighbor of Adjacent Cluster	Neighbor of Cluster Head
11,13	14
19	17

### 3.2.2 內部傳輸與外部傳輸方法

在傳輸方面有二種，是依每個叢集成員的觀點，分為內部與外部傳輸。其一、為內部傳輸 (Internal traffic)：為叢集內請求的訊息。其二、外部傳輸 (External traffic)：為其他叢集所傳來的請求訊息。在內部傳輸方面是由叢集標頭和閘道節點負責傳輸。叢集標頭接收到他自己叢集成員請求的訊息時，則叢集標頭負責叢集內成員的廣播。而閘道節點接收到叢集標頭的請求訊息時，則將這個請求訊息廣播到鄰近的叢集，繼續往其他叢集請求，直到目的地節點回傳訊息為止。

外部傳輸方面是由叢集標頭、閘道節點及連接鄰近叢集節點負責傳輸。當閘道節點接收到其他叢集傳來的請求訊息時，會將此請求訊息傳送給叢集標頭，由叢集標頭負責廣播給他自己的叢集成員。在外部傳輸裡增加連接鄰近叢集節點可以廣播請求訊息。連接鄰近叢集節點會廣播此請求訊息給叢集標頭，叢集標頭會廣播給他的成員。依序往其他叢集廣播請求訊息，找到目的地為止。

### 3.2.3 叢集標頭維護與拓樸改變方法

在 MANET 環境行動主機具有移動性，使得網路拓樸經常改變。會引起網路拓樸改變有二種影響因素：叢集標頭移動或叢集成員移動。先前有論文提出每個種類的恢復機制。此恢復機制把網路拓樸改變描述完成，所以我們延續使用。此恢復機制描述在下面[28]。

**叢集標頭的恢復機制：**在這個案例，因為叢集標頭移動造成鄰居的數量改變。因此，已經建立好的叢集架構便需隨著網路拓樸的改變有所調整，以保持叢集架構的完整。在這個段落會說明叢集內，叢集標頭的解除條件 (Remove termes)。當沒有接收到鄰居廣播的訊號時，叢集標頭將會失去他的領導能力。這些叢集成員有接收到其他鄰居的訊號時，他可能加入此鄰居節點連接到另一個叢集標頭或成員。另外，當叢集標頭的權重值小於  $removeCH\_threshold$  時，則會失去叢集的領導能力。叢集標頭解除的條件：自己範圍內沒有成員或自己的能源電量小於  $removeCH\_threshold$  (公式 2)。

叢集標頭的權重值與  $removeCH\_threshold$  的差異越大。叢集標的能量較多時，表示叢集標頭的存活時間越長，避免經常更換叢集標頭時所需額外花費的成本。相對的叢集標頭可能需服務較多的節點，可以

減緩節點失效的時間。

當叢集標頭的權重值小於  $removeCH\_threshold$  時，立即解除此叢集標頭。叢集內的成員會依據鄰居資訊加入另外叢集，圖 3-5、當叢集標頭廣播即將改變時，依據流程圖的處理程序，如果在鄰居資訊裡有另一叢集，則加入其他叢集。如果在鄰居資訊裡沒有另一叢集，必須重新啟動叢集演算法。透過競爭後，宣佈他自己成為叢集標頭。詳細解除叢集標頭演算法如圖 3-6。

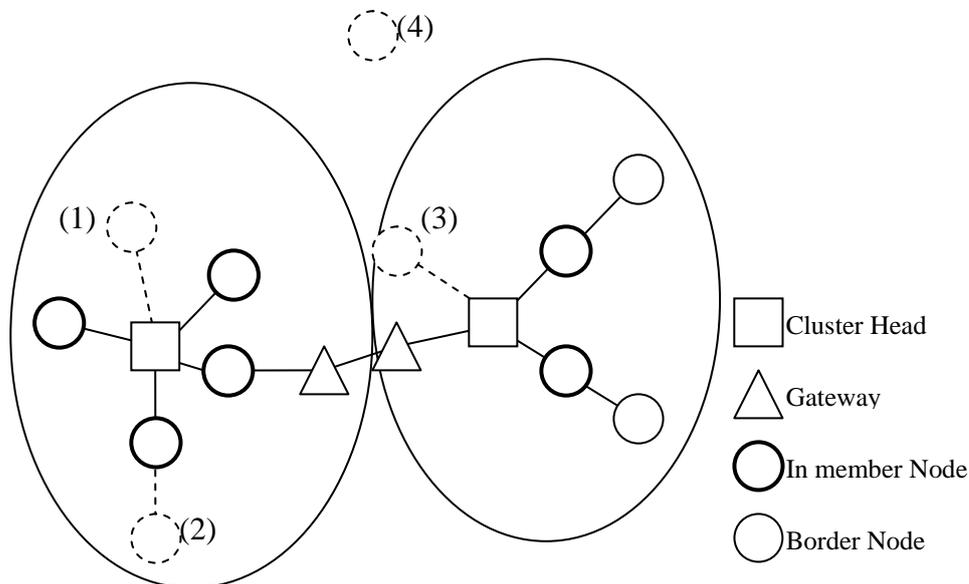


圖 3-5、叢集標頭改變

當內部成員接收到叢集標頭所傳來的訊號，此訊號表明叢集標頭要改變，這個機制是有一點不同。有四種可能（如圖 3-5），(1) 得知叢集標頭要改變時，如果有連接到新的叢集標頭，則加入此叢集成為內部成員，並且更新叢集資訊。(2) 如果有連接到新叢集的鄰居時（1-hop），則加入此叢集成為邊界成員，並且更新叢集資訊。(3) 基於鄰居資訊可以發現另一個叢集時，則加入此叢集，鄰居節點成為內

部成員 (1-hop) 與邊界成員 (2-hop)，並且更新叢集資訊。(4) 如果無法連接到任何叢集時，則宣佈自己成為叢集標頭。

```
Alg.3 Remove Cluster Head Algorithm  
Degree: its neighbor number  
If Cluster Head (Degree = 0 ||  
its Weigh_Value < removeCH_threshold ){  
  Remove Cluster Head  
  If can find another cluster {  
    Cluster member join other Cluster  
  }  
  Else not can find another cluster {  
    Restart Cluster algorithm  
    Go through Contention,  
    and announce itself as a Cluster Head.  
  }  
}
```

圖3-6、解除叢集標頭演算法

**移動叢集成員的恢復機制：**這是恢復演算法主要的部份。恢復機制主要是為了叢集成員，因為在叢集內的內部成員或叢集標頭會時常移動。

假設在一個預先規定的時間內，發現沒有接收到內部成員所廣播的訊號時，則認為叢集成員已經移動。當叢集成員沒有接收到叢集標頭所廣播的訊號時（這個節點是在叢集標頭的鄰近區域裡），或接收到內部成員的訊號，訊號是告知每個成員，叢集標頭已經改變（到叢集標頭的 hop count > 2）。這些都是會引起拓樸改變的因素。

在上面有提到指出，內部成員扮演一個重要角色，內部成員是提供區域網路的資訊給每個節點。就因為內部成員是直接與叢集標頭相鄰，所以接收到的訊息會比邊界成員來得重要。

每個節點丟失他們的內部成員後，如何改變他們的狀態。有三種可能（見圖 3-7），(1) 如果有連接到叢集標頭，則成為內部成員，必須更新他的身份狀態和 hop count 資訊。(2) 當可以連接到另一個叢集的鄰居時，則加入此叢集，則更新叢集資訊。(3) 當無法連接到先前的叢集與其他鄰近叢集時，透過競爭後則宣佈自己成為叢集標頭。

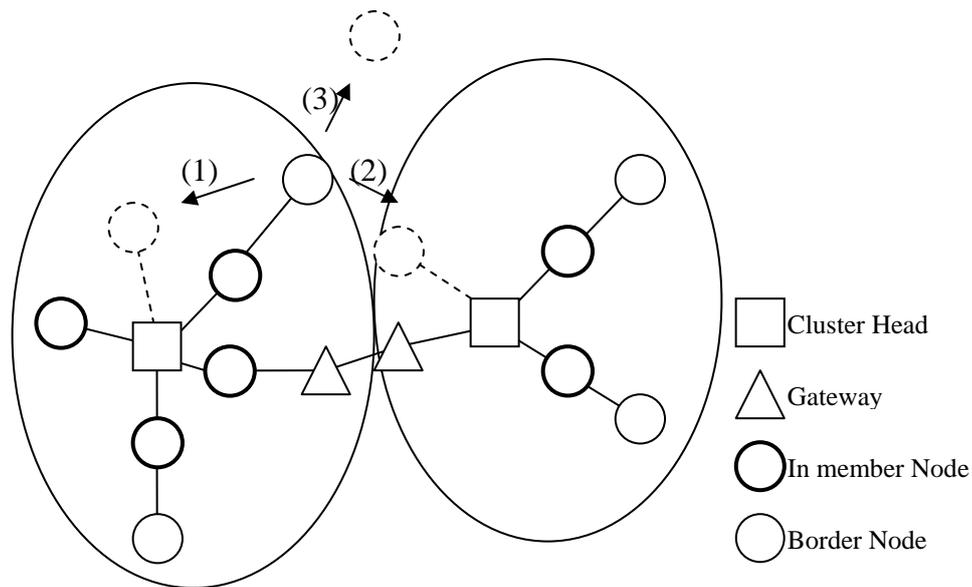


圖 3-7、假設成員移動

### 3.3 叢集路由協定

在無線網路裡，當一個來源節點（Source node）對目的地節點（Destination node）發出 RREQ 服務封包時。他會啟動路由發掘程序（Route Discovery process），發掘程序會被不同的節點執行，直到找到一個路徑（Path），或是所有可能的路徑都被執行過為止。而當路徑被建立起後，路由維護程序（Route Maintain process）便用來維持這個路徑。

### 3.3.1 路由發掘程序(Route Discovery process):

任一來源節點要找尋目的地節點時，先向叢集標頭發出 RREQ 封包。這裡叢集內每個叢集成員會幫忙廣播 RREQ 給叢集標頭。叢集標頭透過有限制的廣播（稱為：finite flooding broadcast）來送出 RREQ。Finite flooding broadcast 指的是 Flooding 時，以叢集標頭為目標，只有叢集標頭才會處理此 RREQ 封包。（可以減少 flooding 時的封包量）。

Finite flooding broadcast 的方式如下：

- 任一叢集一般成員收到 RREQ 封包後皆不回應，除非目的地節點。
- 任一閘道節點只有在第一次收到 RREQ 封包才轉傳或回應。除叢集標頭之外，叢集標頭可以接收 RREQ 封包數次以上，轉傳或回應也是。

目的地節點第一次收到 RREQ 封包後，依記錄在封包內的路徑，回傳 RREP 封包給來源節點，用以建立主要路由（Main Route）。目的地節點在第二次收到此 RREQ 封包後，依記錄在此封包內的路徑，回傳 RREP 封包給來源節點，此第二條路徑稱為候選路由（Candidate Route）。如果來源節點發出 RREQ 封包（如圖 3-8），此叢集標頭接收到來源節點的 RREQ 時，並發出 Finite flooding broadcast，直到目的地節點回應，[1(S) -> 2 -> 5 -> 4 -> 11 -> 22 -> 23 -> 24 -> 26 -> 28 -> 32 -> 33-> 34-> 35(D)、1(S) -> 2 -> 5 -> 7 -> 10 -> 12 -> 13 -> 15 -> 16 -> 17 -> 36 -> 34 -> 35(D)]。這 [1(S) -> 2 -> 5 -> 7 -> 10 -> 12 -> 13 -> 15 -> 16 -> 17 -> 36 -> 34 -> 35(D)] 路徑的 Hop count 為 9 被選為主

要路徑。則 [1(S) -> 2 -> 5 -> 4 -> 11 -> 22 -> 23 -> 24 -> 26 -> 28 -> 32 -> 33-> 34-> 35(D)]路徑的 Hop count 為 10 被選為候選路徑。節點 10、11、12、17、18、22、28、29、32、36 為閘道節點，當接收到 RREQ 時，只可以轉傳或回應一次。而節點 15、24、34 是叢集標頭，可以轉傳或回應數次。

假設以較大的網路環境，每個叢集裡的叢集標頭會負責對內傳輸，這樣的在泛濫的方式可以減少許多的控制封包，不會造成 overhead。當叢集標頭在他自己的叢集裡找不到目的地時，會對外尋找其他鄰近叢集。對外傳輸是由閘道節點所負責，因為閘道節點可以連接到另一個叢集。他會將訊息傳到另一個叢集閘道節點，叢集閘道節點會將訊息廣播給叢集標頭。當叢集標頭接收到此封包時，會在他的叢集裡尋找目的地節點。如果在沒有目的地節點時，則在依序往另一個叢集尋找，直到目的地節點出現為止。

### 3.3.2 路由維護程序(Route Maintain process):

當來源端和目的地端在連線建立之後，也有可能因為點的移動而使得原先建立的路徑發生斷裂，此時就必需做路由修復。假設資料傳送時主要路徑失效(斷裂)，則由上一個節點回傳路由錯誤 (Route error ; RERR) 封包給來源節點。在回傳 RERR 前，此節點會先採用局部修復 (Partial Recovery) 的方式以加速路由發掘 (Route discovery) 運作及減少網路封包量。局部修復是由斷裂處的上一個叢集頭標找尋到目的節點的路徑。若此叢集頭標仍無法找到目的節點的路徑，則再回報給上一個叢集頭標，直到回到來源節點的叢集標頭。透過這樣的方式將可以有效減少當路徑斷裂 (Route broken) 時，不用直接回到來源叢集標頭重新尋找路徑，以減少網路的負載。假設當來源節點的叢集標頭接收到 RERR 封包或局部修復後的路徑比候選路徑來得

遠，則啟用候選路徑。因為主要路徑在過程中，是有可能透過較遠的路徑來尋找到目的地端。若候選路徑再次失效，也採用局部修復的方式，不用直接回到來源節點重新尋找路徑，以減少網路的負載，使得整體無線網路效能提升。

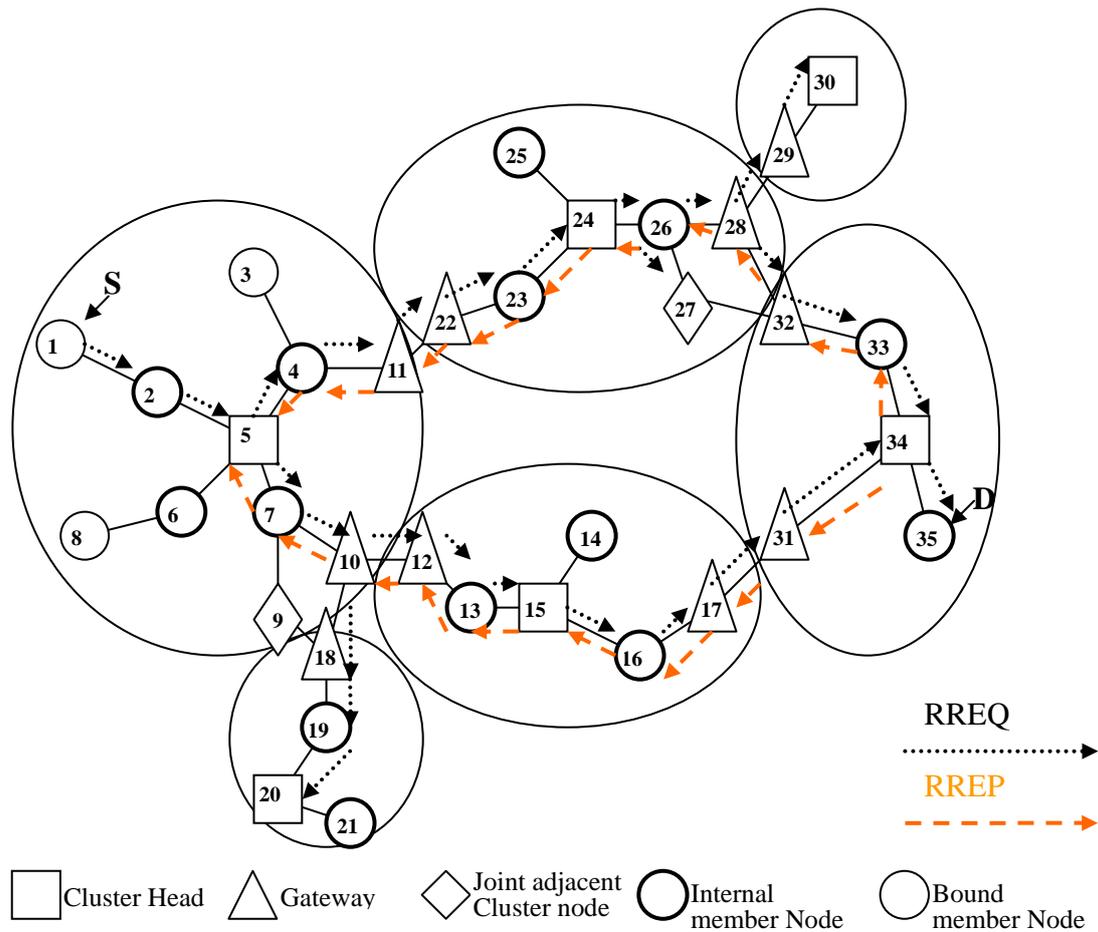


圖 3-8、路徑發掘

如圖 3-9 中主要路由裡節點 17 到 31 的路徑斷裂，則節點 17 必須啟動局部修復機制，回傳給上一個或上上一個節點為叢集標頭者，由叢集頭頭尋找是否有可到目的地端的路徑。假設節點 14、31 互相可以連接，則叢集標頭 15 會透過節點 14 來建立局部修復機制。假設叢集標頭 15 的鄰居或鄰近叢集成員沒有可到目的地端，則回報給上

一個叢集標頭 5。回傳 RERR 給來源叢集標頭節點 5，馬上啟動候選路徑做為傳送封包的路徑。

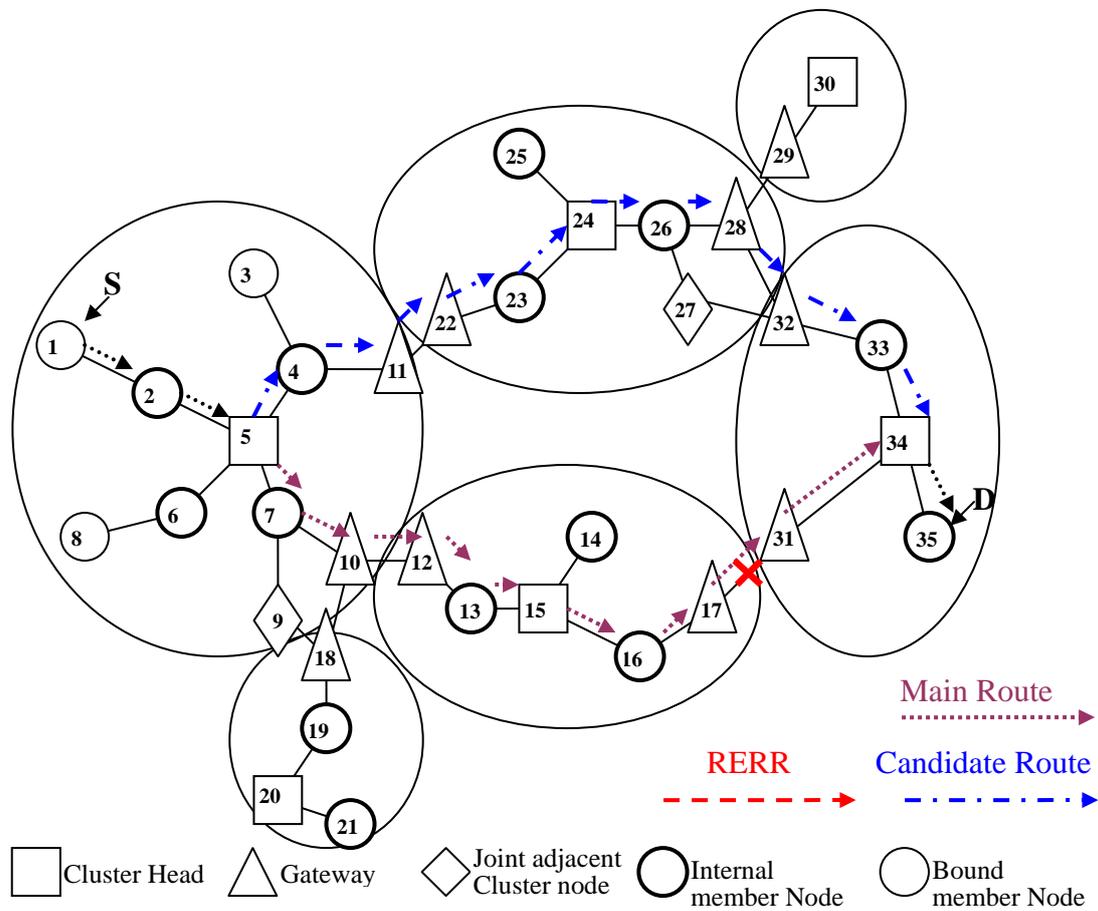


圖 3-9、路徑維護

## 第四章 效能評估

在這個章節，我們將模擬類似 HCC 以及本文所提出的 CPEE 的模擬做比較以及分析討論其結果。此模擬實驗所使用的作業系統為 Microsoft Windows XP，程式語言則使用 JAVA 來撰寫模擬實驗的程式，使用的 JDK 版本為 1.5.0\_10，開發工具是 GeI RC39。

### 4.1 模擬環境

在模擬實驗的環境，網路的大小為 600m×600m，在網路上節點的個數介於 20 至 90 之間變化；移動速率在 20 km/hr 至 80 km/hr 之間來進行移動； $\alpha$  值在 0.05 至 0.95 之間變化；而每一個節點的廣播範圍大小為 100m；在網路形成的一開始，每一個節點的座標位置採取隨機定位 (Random Location) 模式，之後每一個節點會在 20 km/hr 至 80 km/hr 之間隨機選擇速率以及方向 (總共為 360°) 來進行移動，所有參數如表 4-1 所示。

表 4-1、模擬環境的參數表

Parameter	Value
Network size	600 m × 600 m
Number of nodes	20、30、40、50、60、70、80、90
Mobility speed	20、40、60、80 (km/hr)
$\alpha$	0.95、0.65、0.25、0.05
Pause time	1 sec
Communication range	100 m
Power value	90~100%
Number of data packets	10

## 4.2 模擬結果

在這裡我們將模擬類似 HCC 與 CPEE 叢集演算法的效能分析，透過模擬結果可以很明確的知道兩者叢集演算法間差異。並且依各種不同的節點數量與移動速率的變化，來觀察兩種協定的 Network Lifetime 的差異。進行 Network Lifetime 模擬分析時，我們是依網路內行動主機的能源 $\leq 0$ 時，則將模擬分析停止後，並計算其平均值。

圖 4-1 裡，假設在 70 個節點以不同的移動速率中， $\alpha$  以 0.95、0.65、0.05，顯示出兩個演算法的網路的存活時間，在本文所題出的 CPEE 叢集演算法，透過公式 1 所選出的叢集標頭來管理叢集內的成員，使得整體網路可以有效的延長壽命。當  $\alpha$  為 0.95 時，是著重節點鄰居數量因素，在建立叢集時會跟 HCC 類似，但在叢集維護的方式有所差異，使得 Network Lifetime 較 HCC 來的長（如表 4-2）。當  $\alpha$  為 0.65 時，較偏重於鄰居數量因素，但有增加能源因素的比重，使得 Network Lifetime 比  $\alpha$  為 0.95 時來的長（如表 4-3）。當  $\alpha$  為 0.05 時，是著重節點能源因素，建立叢集時會依節點的能源值決定叢集標頭，使得 Network Lifetime 較其他實驗長（如表 4-4）。而類似 HCC 叢集演算法在選擇叢集標頭時，是依每個節點的鄰居數量為最高成為叢集標頭。但在能源消耗的因素卻沒有考量，這樣會使得叢集標頭的能源消耗迅速，而讓整體網路壽命縮短。模擬結果顯示 CPEE 演算法的整體網路壽命比另一演算法來長，平均延長 284.34 秒，這樣延長整體網路時間，使得網路效能最佳化。

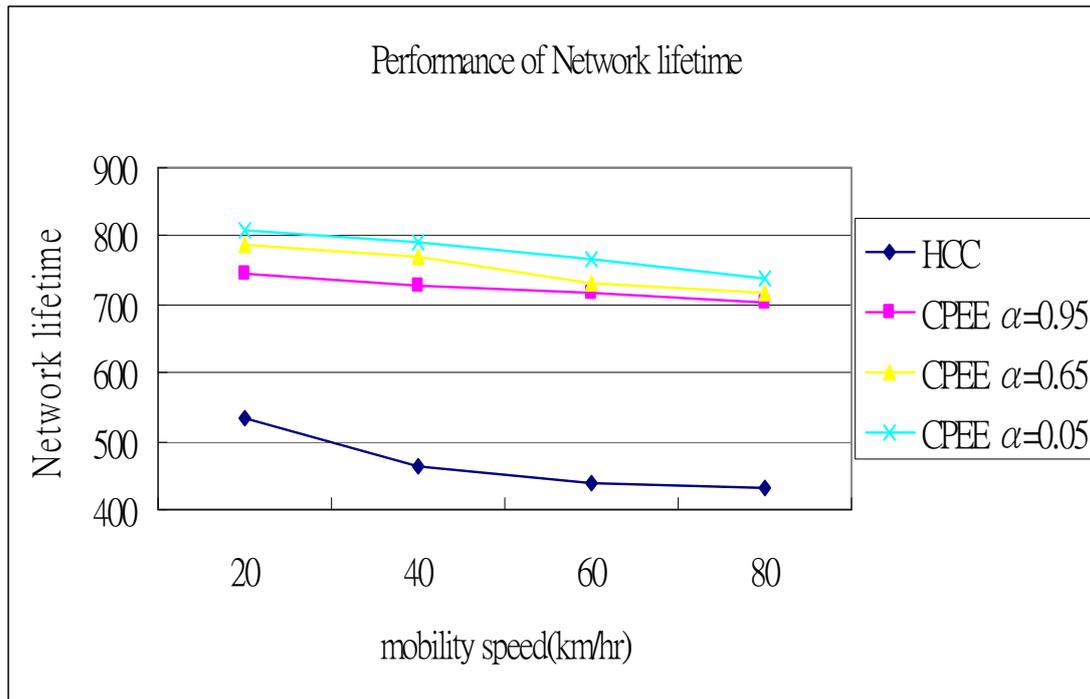


圖 4-1、70 個節點下平均 Network Lifetime 之效能分析

表 4-2、70 個節點下平均 Network Lifetime 之效能分析 ( $\alpha=0.95$ )

Mobility speed (km/hr)	Average Network Lifetime		Improvement
	HCC	CPEE $\alpha=0.95$	
20	532.40	744.40	40%
40	462.00	729.00	58%
60	437.83	716.00	64%
80	431.67	703.80	63%
<b>Average</b>	465.77	723.30	55%

表 4-3、70 個節點下平均 Network Lifetime 之效能分析 ( $\alpha=0.65$ )

Mobility speed (km/hr)	Average Network Lifetime		Improvement
	HCC	CPEE $\alpha=0.65$	
20	532.40	789.00	48%
40	462.00	768.33	66%
60	437.83	731.00	67%
80	431.67	715.50	66%
Average	465.77	750.96	61%

表 4-4、70 個節點下平均 Network Lifetime 之效能分析 ( $\alpha=0.05$ )

Mobility speed (km/hr)	Average Network Lifetime		Improvement
	HCC	CPEE $\alpha=0.05$	
20	532.40	807.17	52%
40	462.00	792.17	71%
60	437.83	767.67	75%
80	431.67	737.33	71%
Average	465.77	766.09	64%

圖 4-2 裡，假設在移動速率 40km/hr 以不同的節點的數量中， $\alpha$  以 0.95、0.65、0.05，顯示出兩個演算法的網路的存活時間，在本文所題出的 CPEE 叢集演算法，透過公式 1 所選出的叢集標頭，來管理叢集內的成員，使得整體網路可以有效的延長壽命。當  $\alpha$  為 0.95 時，跟 HCC 的趨勢相同，節點密度高時，使得 Network Lifetime 遞減（如表 4-5）。當  $\alpha$  為 0.65 時，遞減趨勢就沒有  $\alpha$  為 0.95 時來的嚴重，因為這

有考慮能源因素較多（如表 4-5）。當  $\alpha$  為 0.05 時，著重於節點能源，這可能會造成能源高，但鄰居數量少，使得 MANET 效能下降。圖 4-2 在  $\alpha$  為 0.05 與節點 20 時，節點密度底而使得效能下降(如表 4-6)。模擬結果顯示 CPEE 演算法的整體網路壽命比另一演算法來長，平均延長 226.34 秒，這樣在高密度的環境下延長整體網路時間，使得網路效能最佳化。

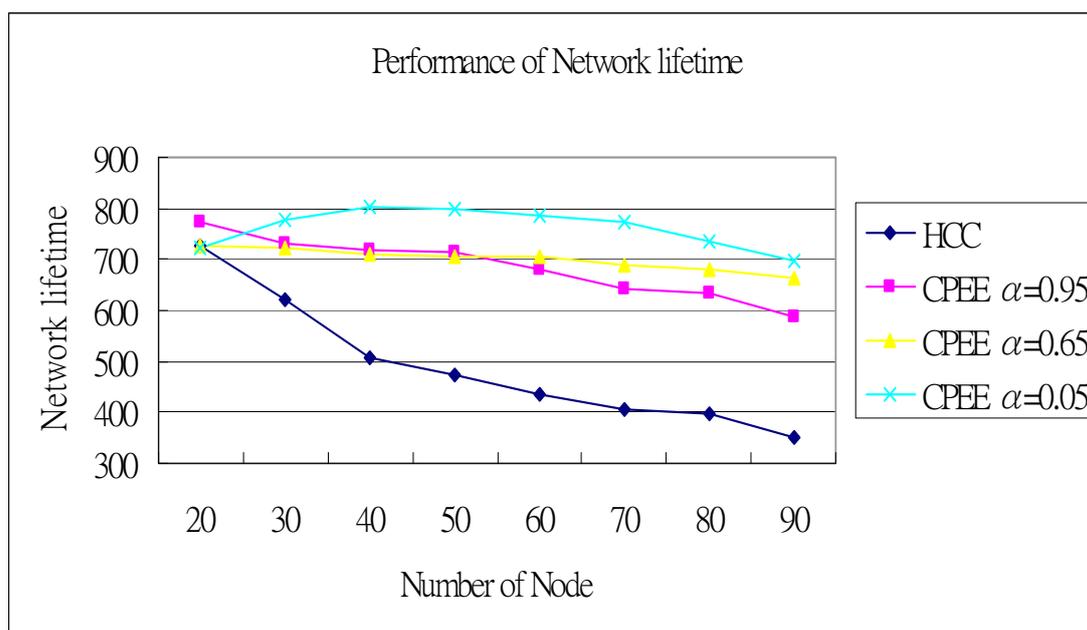


圖 4-2、移動速率 40 km/hr 下平均 Network Lifetime 之效能分析

表 4-5、移動速率 40 km/hr 下平均 Network Lifetime 之效能分析表  
( $\alpha=0.95$ )

Number of nodes	Average Network Lifetime		Improvement
	HCC	CPEE $\alpha=0.95$	
20	727.33	774.83	7%
30	620.17	730.80	18%
40	508.50	719.17	41%

<b>50</b>	471.25	716.17	52%
<b>60</b>	434.33	678.83	56%
<b>70</b>	406.33	641.17	58%
<b>80</b>	395.33	633.50	60%
<b>90</b>	352.00	588.60	67%
<b>Average</b>	489.41	685.38	40%

表 4-6、移動速率 40 km/hr 下平均 Network Lifetime 之效能分析表  
( $\alpha=0.65$ )

<b>Number of nodes</b>	<b>Average Network Lifetime</b>		<b>Improvement</b>
	<b>HCC</b>	<b>CPEE <math>\alpha=0.65</math></b>	
<b>20</b>	727.33	727.00	-0.001%
<b>30</b>	620.17	722.67	17%
<b>40</b>	508.50	710.00	40%
<b>50</b>	471.25	707.33	50%
<b>60</b>	434.33	703.67	62%
<b>70</b>	406.33	689.00	70%
<b>80</b>	395.33	679.25	72%
<b>90</b>	352.00	661.75	88%
<b>Average</b>	489.41	700.08	88%

表 4-7、移動速率 40 km/hr 下平均 Network Lifetime 之效能分析表  
( $\alpha=0.05$ )

Number of nodes	Average Network Lifetime		Improvement
	HCC	CPEE $\alpha=0.05$	
20	727.33	721.40	-0.08%
30	620.17	778.00	25%
40	508.50	804.67	58%
50	471.25	796.83	69%
60	434.33	786.00	81%
70	406.33	773.00	90%
80	395.33	730.00	86%
90	352.00	698.50	98%
Average	489.41	761.80	56%

## 第五章 結論

這個演算法使用 2-hop 在 MANET 網路環境下，相對的在網路內的叢集數量較少，在路徑發現較迅速及減少過多的控制封包，使得 MANET 網路的效能提高。同時也考慮每個節點的能源，以提升 MANET 網路整體存活時間及每個節點的平均存活時間延長，排除 MANET 網路被分割。並且來源節點找到目的節點的路徑，可以迅速的找到主要路徑方式，來做第一時間上的回應，並且有候選路徑來做備份路徑。

模擬實驗結果，本文中所題出的新叢集協定基於有效能源演算法，在 MANET 網路可以延長 40% 至 88% 的平均網路壽命。MANET 網路整體的效能提升，使得 MANET 網路達到較佳的叢集效能及減少能源快速消耗。

## 參考文獻

- [1] A.D. Amis, R. Prakash, T.H.P. Vuong and D.T. Huynh, “Max-Min D-Cluster Formation in Wireless Ad Hoc Networks,” Proceedings of IEEE INFOCOM'2000, ( 2000), pp. 32-41.
- [2] B. An and S. Papavassiliou, “A mobility-based clustering approach to support mobility management and multicast routing in mobile Ad hoc wireless networks,” International Journal of Network Management, (2001), pp.387-395.
- [3] D. J. Baker and A. Ephremides, “The Architectural Organization of a Mobile Radio Network via a Distributed Algorithm,” IEEE Transactions on Communications, (1981), pp. 1694– 1701.
- [4] S. Basagni, “Distributed and mobility-adaptive clustering for multimedia support in multi-hop wireless networks,” in Proc. IEEE Vehicular Tech. Conf. (VTC), (Amsterdam, Netherlands), (1999), Vol. 2, pp. 889-893.
- [5] C.-W. Charng, W.-C. Lo, L.-F. Sung, and S.-I. Hwang, “EDAB: Energy-Efficient Diffusion Algorithm for Broadcast in Wireless Ad hoc Networks,” Wireless Pervasive Computing, 2006 1st International Symposium, (2006), p16-18.
- [6] M. Chatterjee, S.K. Das, and D. Turgut, “An On-demand weighted clustering algorithm (WCA) for Ad hoc networks,” IEEE Global Telecommunications Conference, (2000), vol. 3, pp. 1697-1701.
- [7] C.-C. Chiang and Mario Gerla, “Routing and multicast in multihop, mobile wireless networks,” IEEE 6th International Conference on Universal Personal Communications Record, (1997), vol. 2, pp. 546-551.
- [8] C.-C. Chiang, H.-K. Wu, W. Lin, and M. Gerla, “Routing in Clustered Multihop Mobile Wireless Network with Fading Channel,” Proceedings of IEEE Singapore International Conference on Networks, (1997), pp. 197-211.
- [9] C.-C. Chiang, H.-K.g Wu, Winston Liu, and Mario Gerla, “Routing in Clustered Multihop, Mobile Wireless Networks With Fading Channel,” IEEE Singapore International Conference on Networks, (1997), pp. 197-211.
- [10] A. Ephremides, J. E. Wieselthier, and D. J. Baker, “A Design Concept for Reliable Mobile Radio Networks with Frequency Hopping

- Signaling,” Proceedings of IEEE, (1987), pp. 56–73.
- [11]F. Foroozan, S.Datta, “A Low-Maintenance Energy-Aware Clustering Algorithm for Wireless Ad-Hoc Network,” in Proceedings of IEEE, (2006), pp. 457- 462.
- [12]M. Gerla and J.T.C. Tsai, “Multicluster, mobile, multimedia radio network,” ACM Baltzer Journal of Wireless Networks, (1995), pp. 255–265.
- [13]S. Giordano, I. Stojmenovic, and L.Blazevic, ”Position Based Routing Algorithms for Ad Hoc Network:A Taxonomy,” in Proceedings of IEEE, ( 2001), vol.40, no.7, pp.128–134.
- [14]Z.J. Haas and M.R. Pearlman, “Determining the Optimal Configuration for the Zone Routing Protocol,” IEEE Journal on Selected Areas in Communications: Wireless Ad hoc Networks, (1999), vol. 17, pp. 1395-1414.
- [15]Z.J. Haas and M.R. Pearlman, “The performance of a new routing protocol for the reconfigurable wireless networks,” IEEE International Conference Record on Communications, (1998), vol. 1, pp. 156-160.
- [16]C. C. Huang, M. H. Guo and R. S. Chang, “Weight-Based Clustering Multicast Routing Protocol for Mobile Ad hoc Networks,” IEEE Wireless Communications and Networking Conference, (2003), pp. 1112-1117.
- [17]D.B. Johnson and D.A. Maltz, “Dynamic Source Routing in Ad hoc Wireless Network,” Mobile Computing, (1996), pp. 153-181.
- [18]D. Kim, S. Ha, and Y. Choi, “K-hop cluster-based dynamic source routing in wireless Ad hoc packet radio network,” IEEE Vehicular Technology Conference, (1998), vol. 1, pp. 224-228.
- [19]T.J. Kwon and M. Gerla, “Clustering with power control,” Proceedings of the IEEE Military Communications Conference, (1999), vol. 2, pp. 1424-1428.
- [20]H.C. Lin and Y.H. Chu, “A clustering technique for large multihop mobile wireless networks,” The 2000 IEEE 51st Vehicular Technology Conference VTC 2000-Spring Tokyo, (2000), vol. 2, pp. 1545-1549.
- [21]W. Lou, J. Wu, “On reducing broadcast redundancy in ad hoc wireless networks,” in Proceedings of IEEE Transactions on mobile computing, (2002), pp. 111-123.
- [22]D. Lundberg, “Ad hoc Protocol Evaluation and Experiences of Real

- World Ad hoc Networking,” Uppsala Universitet, (2002).
- [23]S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen and J.-P. Sheu, “The Broadcast Storm Problem in a Mobile Ad Hoc Network,” in Proceedings of IEEE/ACM MOBICOM, (1999), pp. 151-162.
- [24]F.G. Nocetti and J.S. Gonzalez, “Connectivity Based k-Hop Clustering in Wireless Networks,” Telecommunication Systems, (2003), vol. 22, pp. 205-220.
- [25]E. Pagani and G. P. Rossi, “An On-demand Shared Tree with Hybrid State for Multicast Routing in Ad hoc Mobile Wireless Networks,” International Workshops on Parallel Processing, (1999), pp. 4-9.
- [26]E. Pagani and G. P. Rossi, “Providing reliable and fault tolerant broadcast delivery in mobile Ad hoc networks,” Mobile Networks and Applications, (1999), vol. 4, pp175-192.
- [27]C.E. Perkins and E.M. Royer, “Ad hoc On-demand Distance Vector Routing,” Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, (1999), pp. 90-100.
- [28]R. Purtoosi, H. Taheri, A. Mohammadi, F. Foroozan, “A Light-Weight Contention-Based Clustering Algorithm for Wireless Ad Hoc Networks,” Proceedings of the Fourth International Conference on Computer and Information Technology, (2004), pp. 627-632.
- [29]A. Qayyum, L. Viennot, A. Laouiti, “Multipoint relaying: An efficient technique for flooding in mobile wireless networks,” (2000), INRIA report RR-3898.
- [30]E.M. Royer, “Multi-Level Hierarchies for Scalable Ad hoc Routing,” Wireless Networks, (2003), vol. 9, pp. 461-478.
- [31]E.M. Royer and Santa Barbara Chai-Hai-Keong Toh, “A Review of Current Routing Protocols for Ad hoc Mobile Wireless Networks,” IEEE Personal Communications, (1999), p46-55.
- [32]P. Sinha, S.V. Krishnamurthy, and S. Dao, “Scalable Unidirectional Routing with Zone Routing Protocol(ZRP) Extensions for Mobile Ad hoc Networks,” In Proceedings of Wireless Communications and Networking Conference, (2000), pp. 1329-1339.
- [33]S.-S, M.-W, and C. S.-R , “Power-aware routing in mobile ad hoc networks,” Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking, (1998), pp. 181-190.
- [34]C.-K. Toh, H. Cobb, and D. A. Scott, “Performance evaluation of

- attery-lifeaware routing schemes for wireless ad hoc networks,” IEEE International Conference on Communications, (2001), Vol. 9, pp. 2824-2829.
- [35]B. Williams, and T. Camp, “Comparison of Broadcasting Techniques for Mobile Ad Hoc Networks,” in Proceedings of ACM MOBICOM, (2002), pp. 194-205.
- [36]Y. Yi, M. Gerla and T. J. Kwon “Efficient Flooding in Ad hoc Networks: a Comparative Performance Study,” in Proceedings of ICC, (2003), pp. 1059-1063.
- [37]N. Beijar, “Zone Routing Protocol (ZRP)” <http://www.tct.hut.fi/opetus/s38030/k02/Papers/08-Nicklas.pdf>, (2002).
- [38]J. Broch, D.B. Johson, D.A. Maltz, and Y.-C. Hu, “The dynamicsource routing protocol for mobile Ad hoc networks” <http://www-2.cs.cmu.edu/~dmaltz/internet-drafts/draft-ietf-manet-dsr-09.txt>, (2003).
- [39]Z.J. Haas, M.R. Pearlman, and P. Samar, “The Zone Routing Protocol (ZRP) for Ad hoc Networks” <http://www.ietf.org/internet-drafts/draft-ietf-manet-zone-zrp-04.txt>, (2002).
- [40]J. Schaumann, “Analysis of the Zone Routing Protocol,” <http://www.netmeister.org/misc/zrp/zrp.pdf>, (2002).
- [41]藍明宗，一個行動隨意網路環境下的叢集群播繞徑演算法，世新大學資訊管理學系碩士論文，民國93年。
- [42]陳彥峰，在無線隨意式網路中以功率為基礎的叢集演算法，國立中山大學電機工程學系碩士論文，民國93年。