

國立交通大學

電信工程學系

碩士論文

考慮節點能量之隨意行動無線網路擇路協定

Energy-Aware Routing Protocols for Mobile Ad
Hoc Network

研究生：程偉祥

指導教授：方凱田

中華民國 95 年 6 月

考慮能量之隨機行動無線網路擇路協定

Energy-Aware Routing Protocols for Mobile Ad Hoc Network

研究生：程偉祥

Student : Wei-Hsiang Cheng

指導教授：方凱田

Advisor : Kai-Ten Feng

國立交通大學

電信工程學系碩士班



A Thesis

Submitted to Department of Communication Engineering

College of Electrical and Computer Engineering

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Master of Science

in Communication Engineering

June 2006

Hsinchu, Taiwan, Republic of China

中華民國 95 年 6 月

考慮節點能量之隨意行動無線網路擇路協定

學生：程偉祥

指導教授：方凱田

國立交通大學電信工程學系碩士班

摘 要

隨意行動無線網路是一種不需要透過共用設備來管理網路的網路架構，可以直接在行動運算節點之間做封包的交換。透過隨意行動無線網路的擇路演算法，網路內的行動運算節點可以摒除傳輸範圍的限制，以其他行動運算節點當作中繼節點幫忙傳遞封包，使來源節點可以傳送資料封包到傳輸範圍之外的目標節點。擇路演算法依照功能可以分成單一擇路演算法與多重擇路演算法，可用來處理單點對單點、單點對多點與多點對多點的資料傳輸。目前的擇路演算法只考量到中繼節點的數量多寡，而忽略到接收功率、電池量限制與移動模式等的影響。在行動運算節點的電池能量有限制的情況下，這篇論文將介紹能量分散擇路演算法 (Distributed Energy Routing - DER)，讓路徑的選擇可以避開已經快要沒電力的中繼節點，提供單點對單點資料傳輸的可靠性。在現實環境之中，常會遇到一些有組織性的移動模式(例如軍隊的移動)，適當的利用樹狀網路架構跟網狀網路架構的優點設計而成的控制混合式擇路演算法(Power-Controlled Hybrid Multicast

Routing - PCHMR), 可以使這種有群組性移動模式的網路擁有更可靠的網路架構
進行多點對多點的資料傳輸。為了使網路中的路徑連結更加可靠而不易斷裂，
PCHMR 演算法還會根據接收平均功率來選擇路徑，藉此提升路徑的可靠性與避
免狀況差的路徑被使用。



Energy-Aware Routing Protocols for Mobile Ad Hoc Network

Student : Wei-Hsiang Cheng

Advisor : Kai-Ten Feng

Department of Communication Engineering

National Chiao Tung University

Abstract

The Mobile Ad hoc NETWORK (MANET) is a kind of network without the existence of infrastructure. The mobile nodes (MNs) in the MANET can exchange their data packets directly with each others.. By adopting the routing protocols within the MANET, the MNs utilize the intermediate nodes to reroute their data packets and finally to the destination node. By using the ad hoc routing, the source node can conduct packet transmission to the destination node which can be outside of its transmission range. In general, the routing algorithms can be categorized into unicast and multicast routing protocols. The unicast routing protocols handle the situations that one source node is transmitting data packets to one destination node. The multicast routing protocols consider one source to multiple destination nodes for packet delivery. In most research work, only the hop counts within the transmission route are considered; while the influence from the receiving power and remaining energy of the MNs are neglected. This may results in insufficient energy remained within MN to successfully transmit the data packets. In this thesis, a Distributed Energy Routing Protocol (DER) is proposed for unicast routing of data packets. The

proposed DER algorithm avoids adopting the low-energy intermediate node in the decision of the transmitting path and also provide the reliability for the transmitting packet. On the other hand, the Power-Controlled Hybrid Multicast Routing (PCHMR) protocol is proposed for multicast routing. It is composed of the advantages of the tree-based and the mesh-based routing protocols in order to provide more reliable network structure to transmit packets between multiple sources and destinations. In order to provide more reliable linkage on the path and to avoid the usage of an unreliable path, the proposed PCHMR protocol chooses the routing path according to the average received power of the MNs. The performance evaluation and comparison of the proposed DER and the PCHMR protocols will be conducted in simulations.



目錄

第一章 導論	3
第二章 隨意行動無線網路的相關資料	7
2.1 媒體擷取控制層	8
2.2 網路通訊層	12
2.2.1 單一擇路演算法	14
2.2.2 多重擇路演算法	19
第三章 能量分散擇路演算法	28
3.1 設計考量	28
3.2 能量分散演算法設計	30
3.2.1 剩餘電池能量的考量方式	30
3.2.2 最佳路徑的選擇	31
3.2.3 反應計時器的設計	34
第四章 功率控制混合式多重擇路演算法	35
4.1 設計考量	35
4.2 功率控制混合式多重擇路演算法設計	38
4.2.1 演算法的表單設計	39
4.2.2 混合式網路架構的建構與加入架構流程	39
4.2.3 混合式網路架構的維持機制	46

第五章 模擬環境	48
5.1 模擬平台 – 網路架構模擬器.....	48
5.2 使用者移動模型.....	49
5.3 無線傳播模型.....	52
5.3.1 自由空間模型.....	52
5.3.2 二路徑射線合成傳播模型.....	52
5.3.3 遮蔽模型.....	53
第六章 效能評估與分析	56
6.1 能量分散擇路演算法的效能表現.....	56
6.2 功率控制混合式多重擇路演算法的效能表現.....	61
第七章 結論	66



第一章 導論

隨意行動無線網路(Mobile Ad hoc NETwork - MANET)是一種能支援移動性且不需要一個固定的共用設備負責管理整個網路的網路架構。目前最常使用的無線區域網路被規範在 IEEE 802.11 標準內，主要是由擷取管理點(Access Point)進行網路架構的運作與維持，網路中所有的行動運算節點只需要與擷取管理點溝通，就可以藉由擷取管理點作資料傳輸的動作(見圖 1-1)。而這類網路架構的運作與維持因為必須使用到類似擷取管理點的固定共用設備，一旦行動運算節點處在沒有任何固定的共用設備之處(例如比較偏僻的鄉下)，所有的行動運算節點就無法透過無線通道做任何的傳輸動作。相較之下，在隨意行動無線網路內，行動運算節點間可以直接的溝通與傳輸資料，而不需要任何固定的共用設備管理網路架構的運作與維持(見圖 1-2)。但也因為隨意行動無線網路缺乏固定的共用設備負責管理網路，因此網路架構的運作與維持就必須由行動運算節點本身來完成。

IEEE 802.11 標準[1]除了有關擷取管理點管理網路的敘述之外，也針對隨意行動網路的媒體擷取控制層制訂了完整的運作方式。在 IEEE 802.11 標準內，隨意行動網路主要是利用 DCF(Distributed Coordination Function)的方式在運作。DCF 是一種可以讓所有的行動運算節點公平的使用到有限的通道媒體的方式。藉由 DCF，行動運算節點就可以在不需要擷取管理點的情況下，公平的競爭通道媒體並完成二個行動運算節點間的傳輸。

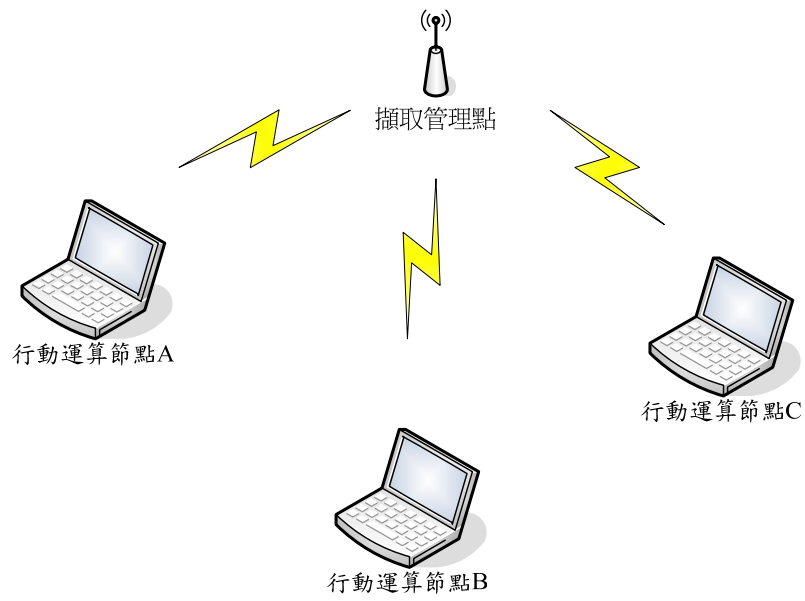


圖 1-1

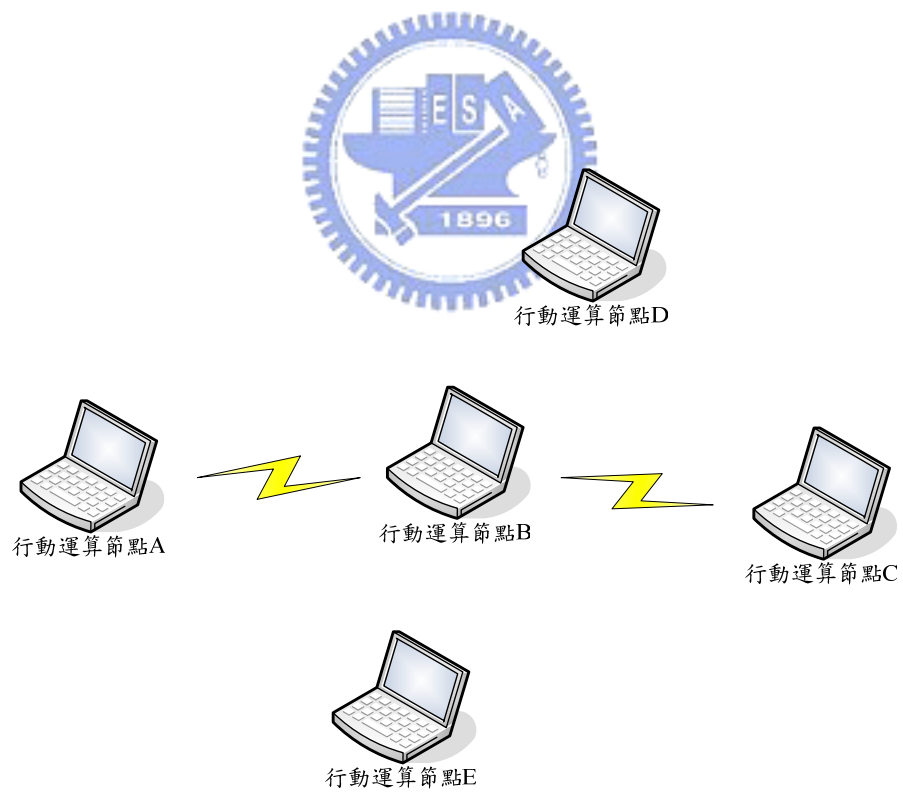


圖 1-2

當二個彼此在傳輸範圍外的行動運算節點想要傳送資料給對方時，隨意行動無線網路可以利用在二個行動運算節點間的行動運算節點當作橋樑的角色，透過這些行動運算節點把資料間接的傳送到目的地。有效的選擇要使用哪些行動運算節點當作橋樑需要使用特殊的演算法運作，目前已經發展出很多演算法完成這個任務，這類型的演算法就統稱為擇路演算法。

擇路演算法依功能可以分成單一擇路演算法與多重擇路演算法。單一擇路演算法使用在一個行動運算節點要傳送封包給另一個行動運算節點的情況。多重擇路演算法則是使用在一個或多個行動運算節點要傳送封包給多個行動運算節點的情況。多重擇路演算法要處理的情況其實使用單一擇路演算法就可以完成，但在多個行動運算節點要傳送封包給多個行動運算節點的情況下，後者可以先把網路內的行動運算節點做組織化，使封包的傳輸比使用單一擇路演算法來的更有效率且節省資源。

擇路演算法在設計時的考量因素很多，一般來說，會把同一個封包的轉傳次數列為考量，稱為跳躍數(Hop Counts)。在這篇論文中，除了考量跳躍數之外，還會把接收功率、節點的電池能量與節點移動模式加入考量，進而設計出更加接近現實環境的擇路演算法。

在一般的單一擇路演算法中，幾乎都假設電池能量是無限制的，也代表該節點是會永遠存在於網路中。在實際情況中，電池能量卻是有限的。如果一般擇路演算法選擇出來的路徑套用在這個限制下，可能會因為某些行動運算節點被過度

使用而電池能量用盡，無法再繼續擔任擇路節點的工作，造成路徑必須再重建，甚至是遺失封包的錯誤，使整體的效能會出現無預期的下降。因此這篇論文提出能量分散擇路演算法(Distributed Energy Routing - DER)，在選擇路徑時，會盡量避免使用到剩餘電池能量較少的中繼節點，使電池能量的耗損可以平均的分佈在整個網路中，使所有節點的生存時間增加。

在一些特殊的情況下，行動運算節點的移動具有群組性(例如軍隊的移動)。這種移動模式具有一些特性：群組內移動相對靜態，而群組間移動相對動態。為了處理這種特性的移動模式，這篇論文將提出功率控制混合式多重擇路演算法(Power-Controlled Hybrid Multicast Routing - PCHMR)，達到群組間的封包傳輸不易失敗且控制封包的數量將會做有效的控制。此外，功率控制混合式多重擇路演算法也考慮了接收平均功率的參數，使混合式網路架構內的連結更加的牢固不易斷裂。

接下來的內容如下：第二章將介紹目前媒體擷取控制層最常使用的 IEEE 802.11 標準，還有擇路演算法的原理與運作方式，另外會介紹一些現存的擇路演算法。第三章與第四章將詳細介紹能量分散擇路演算法與功率控制混合式多重擇路演算法。第五章將介紹模擬所考慮的環境因素。接著第六章會針對能量分散擇路演算法與功率控制混合式多重擇路演算法進行模擬並做效能上的評估與分析。第七章會對這篇論文作最後的結論。最後再附上一些參考文獻。

第二章 隨意行動無線網路的相關資料

根據國際標準組織(International Standard Organization - ISO)所制訂的開放式網路系統架構(Open System Interconnection - OSI)，把網路的傳輸模型大概分成七個層級(見圖 2-1)。在目前隨意行動無線網路相關的研究中，大部分都落在第二層的媒體擷取控制層(Media Access Control)跟第三層的網路通訊層(見圖 2-1)。而這篇論文是針對網路通訊層進行研究與設計，為了可以更快的瞭解這些研究與設計，在這一章會先針對媒體控制層進行相關資料跟協定的介紹，接著針對網路通訊層介紹一些已存在的協定與分析。

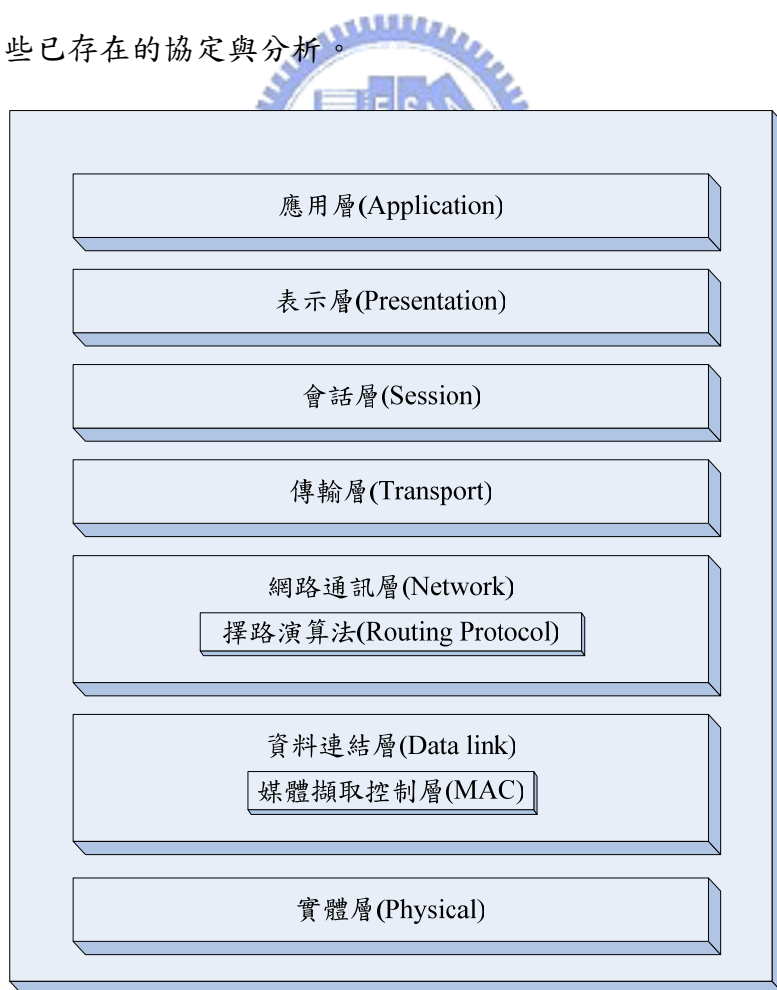


圖 2-1 開放式網路系統架構

IEEE 的 802.11[1]標準大量的被使用在有關於無線區域網路的研究，這個標準制訂了有關無線區域網路在實體層跟媒體擷取控制層的運作方式。目前也已經有大量的無線網路產品利用 IEEE 802.11a/b/g 標準[2][3][4]進行製造，代表 IEEE 802.11 標準已經是一個相當成熟而且可實現的規格，所以這篇論文也以這個標準為基礎，進行更上層的探討。在 2.1 節將會介紹 IEEE 802.11 標準在媒體擷取控制層的規定與運作方式。

在網路通訊層的部分，目前尚未有任何標準出現，但已存在許多有關隨意行動無線網路的擇路演算法研究。針對隨意行動無線網路的擇路演算法，大概可以分成單一擇路演算法跟多重擇路演算法。在 2.2 節將會介紹這二種擇路演算法的差別，跟一些已存在的擇路演算法。



2.1 媒體擷取控制層(Media Access Control Layer - MAC)

無線區域網路的媒體擷取控制層是依據 IEEE 802.11 標準運作的。針對隨意行動無線網路，IEEE 802.11 標準是利用以 CSMA/CA(Carrier Sense Multiple Access with Collision Avoidance)為基礎的 DCF(Distributed Coordination Function)去運作的。DCF 主要的目標就是要讓所有的行動運算節點都可以公平的使用到有限的通道媒體。

當行動運算節點想要傳送封包時，藉由 CSMA/CA，行動運算節點會先感測通道媒體的狀態，如果通道媒體處在無人使用的狀態，就可以進行封包的傳輸。

但如果通道媒體是處在忙碌的狀態，那行動運算節點會先暫存封包，等待通道恢復無人使用狀態。但為了避免通道媒體恢復無人使用狀態後，數個等待中的行動運算節點同時傳輸封包造成碰撞，行動運算節點會利用二元指數型延遲存取 (Exponential Backoff) 演算法產生一個參數，接著依據這個參數產生一個隨機值 (在零到參數之間隨機取一個值)，讓這個隨機值在通道媒體為等待狀態時，隨時間遞減。當這個隨機值遞減為零的時候，便送出封包。藉由延遲傳送時間的方式可以避免碰撞，而二元指數型延遲存取演算法控制延遲的時間，使每一個想要傳輸封包的行動運算節點可以公平的競爭使用通道媒體。

但單純的利用 CSMA/CA 卻會面臨到隱藏節點 (Hidden Node) 的問題，而這種問題就會產生新的碰撞，這種碰撞會發生在圖 2-2 的情況。在圖 2-2 中，因為節點 A 跟節點 C 互相位在傳輸範圍外，而此二節點對節點 B 的傳輸卻會有可能產生碰撞，所以互相視為隱藏節點。舉個隱藏節點會產生的碰撞問題，當節點 A 正在進行傳送封包給節點 B 的動作，而節點 C 同時也有封包想要傳送給節點 B。因為節點 C 在節點 A 的傳輸範圍之外，所以節點 C 無法藉由通道媒體的感測得知節點 A 正在進行傳輸的動作。因此節點 C 會認為通道媒體處於無人使用狀態，而開始傳送封包給節點 B，最後造成碰撞。由此可知，隱藏節點的問題是無法單純的利用 CSMA/CA 的方式解決的。為了解決隱藏節點的問題，DCF 使用了 RTS (Request To Send) 封包跟 CTS (Clear To Send) 封包的交換來達到清空通道媒體的功用。當節點 A 想要傳輸資料給節點 B 時，會先傳送 RTS 封包給節點 B，接

著 B 也會傳送 CTS 封包給節點 A。但因為節點 C 也在節點 B 的傳輸範圍內，節點 C 也會收到這個 CTS 封包，然後瞭解節點 B 即將跟另外節點 A 做傳輸，雖然此時的通道媒體是無人使用的狀態，但因為節點 B 是忙碌的，此時也不能傳送資料給節點 B。利用了 RTS 封包跟 CTS 封包，節點 A 間接的讓節點 C 知道節點 A 跟節點 B 之間的傳輸，順利的解決隱藏節點的問題。

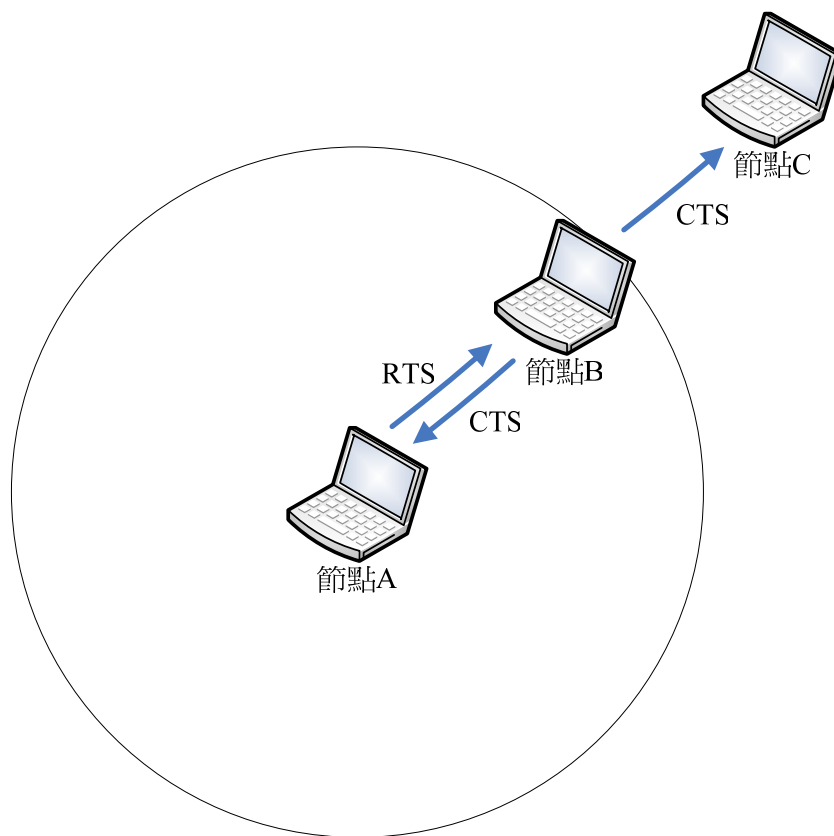


圖 2-2 RTS-CTS 的交換可解決隱藏節點

為了配合 RTS 封包跟 CTS 封包的使用，每個行動運算節點都設有網路配置向量(Network Allocation Vector - NAV)的工具，當行動運算節點接收到 RTS 封包或 CTS 封包時，會藉由這些封包裡面記錄的傳輸資訊，把這次傳輸的時間記錄在 NAV 內。如果現在時間小於 NAV 所記錄的時間，就代表在此節點會影響的範

圍內有封包在進行傳送，則此節點就要避免傳送任何封包以避免去影響正在進行

的封包傳送。圖 2-3 便是在圖 2-2 的情況下，RTS 封包與 CTS 封包交換的時程，

當節點收到節點 B 要送給節點 A 的 CTS 封包，就設定 NAV。

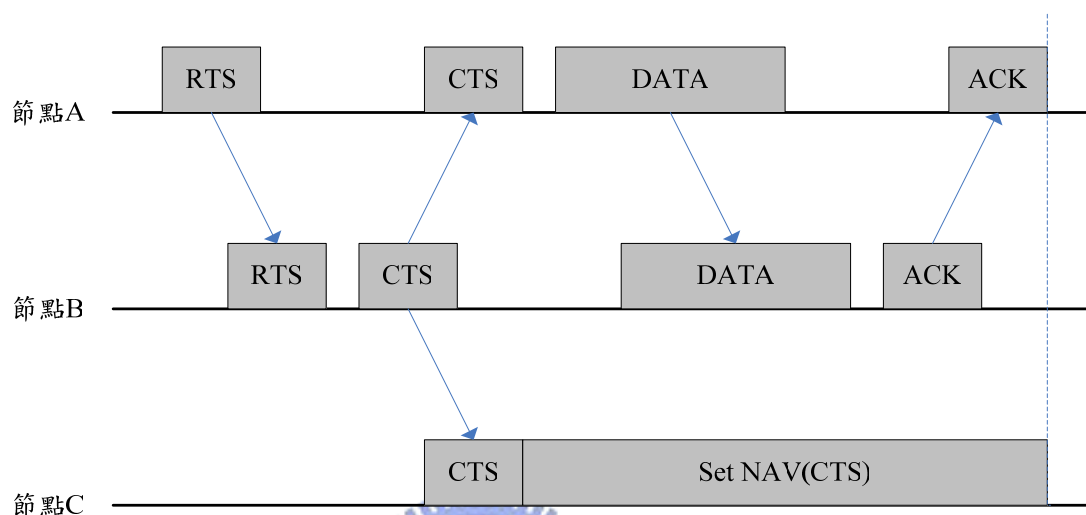


圖 2-3 利用 RTS-CTS 交換後設定 NAV 的示意圖

2.2 網路通訊層(Network Layer)

在網路通訊層之中，針對隨意行動無線網路的擇路演算法，依照傳輸的需求，大概可以分成單一擇路演算法(Unicast Routing Protocol)跟多重擇路演算法(Multicast Routing Protocol)。

當我們要進行單一行動運算節點(來源節點)傳輸封包給在傳輸範圍之外的另一個行動運算節點(目標節點)時，藉由網路內其他的行動運算節點當作中繼節點幫忙傳遞來源節點的封包，間接地使封包可以成功的到達目標節點。從來源節點經過中繼點到目標節點之間所經過的行動運算節點就稱為路徑，而這些中繼節點就稱為擇路節點(Router)。在來源節點跟目標節點之間，通常會有很多條不同的路徑(使用到不同的擇路節點)可以完成傳遞封包的工作，單一擇路演算法就是選擇最佳路徑的方法(所謂的”擇路”就是選擇最佳路徑的意思)，並可以讓路徑上的行動運算節點都知道自己被當成是這一次傳輸的擇路節點。

類似於單一擇路演算法，多重擇路演算法也是選擇最佳路徑的方法。但最大的不同點在於多重擇路演算法是被使用在來源節點要傳輸資料封包給多個目標節點的情況。在這種情況下，如果所有的傳輸通通採用單一擇路的方式，會發生路徑重複率太高的問題(很多擇路節點重複地傳遞同一個封包)，這樣會使封包的傳遞非常的沒效率。有效的解決方法就是在傳遞封包之前，利用少數的控制封包，先把所有的來源節點跟目標節點之間有計畫地建立最省資源的路徑架構。當資料封包被傳遞時，所有的擇路節點都只需要傳遞一次封包就可以使所有的目標

節點都接收到封包，大大的提高傳輸的效率，也使網路資源可以傳輸更多的封包。這種路徑架構的產生方式就稱為多重擇路演算法。

一般來說，一個完整的擇路演算法會分成幾個部分，包括了擇路的搜尋與決定機制、擇路的維持機制與資料封包的傳輸，通常不同擇路演算法之間的差別都會在前二部分，所以之後的介紹也大都著重在這二個部分。

選擇最佳路徑的參數，也會隨著不同擇路演算法而不同，目前最常看到的考量參數就是路徑的跳躍數(從來源節點到目標節點的路徑上，傳遞同一個封包的次數)，而利用這個考量參數所選出的路徑就稱為最短路徑(因為傳遞封包的擇路節點數最少)。由於目前的擇路演算法大都沒有針對特別的移動模式或傳輸環境模式作設計，而是假設所有的行動運算節點都在一個固定的空間與不變的傳輸環境內自由移動，所以設計出來的演算法只考慮了路徑的跳躍數。但實際上，針對不同的移動模式或是傳輸環境模式，應該要有相對應的設計，如此才能更貼切的在實際上使用，所以必須考慮更多的考量因素。在這篇論文中，不僅僅是考量的原本的路徑跳躍數，還另外利用接收功率當作連結是否可靠的依據，也考量每個行動運算節點有能量大小的限制，增加考量的因素，將會使演算法的設計更加的貼切實際情況。

2.2.1 單一擇路演算法

單一擇路演算法依照擇路產生的時間點可以分類成預先維持機制演算法(Proactive Protocol)跟即時搜尋機制演算法(Reactive Protocol)。預先維持機制演算法(例如 DSDV[5]跟 WRP[6])會在傳送資料封包前，週期性的利用控制封包先選擇好到各個目標節點的路徑並維持，而預先選擇好的路徑會存在每個行動運算節點的擇路表單之內。當某個來源節點想要傳送資料封包時，就可以直接利用擇路表單的資訊，直接進行傳輸資料的動作。這種演算法比較強調擇路的維持機制，所以會擁有比較可靠且不易斷裂的路徑，而且因為到目標節點的路徑已經存在擇路表單內，所以不需要任何延遲就可以開始傳輸資料封包。但週期性的利用控制封包維持擇路表單的正確性，當整個網路的變化比較快速的時候，維持的頻率就必須提高，也會造成控制封包過多的問題。即時搜尋機制演算法(例如 AODV[7]、DSR[8]、TORA[9][10]、ABR[11]跟 SSA[12])只會在行動運算節點想要傳送資料時，才開始傳送控制封包進行搜尋路徑然後決定路徑的工作。結束搜尋並決定路徑後，決定的路徑一樣會被存在擇路表單之內，然後才開始傳輸資料封包。這種演算法比較強調擇路的搜尋機制，而比較不強調擇路的維持機制，所以不需要週期性的維持擇路表單的正確性，使控制封包的數量大大減低。但由於維持的機制較弱，所以當整個網路變化比較快速的時候，路徑就會變的相當容易斷裂，而且因為擇路的決定是在要傳送資料封包才開始運作，所以會有時間上的延遲。另外還有一些擇路演算法是合成上述二種擇路演算法(例如 ZRP[13][14]、

CGSR[15][16]跟 Cedar[17])，依照不同的設計觀念，適當的混和這二種擇路演算法，使合成後的演算法可以同時保有各自的優點。目前也有很多研究專門比較上述這些擇路演算法的優劣[18] - [21]。

由於 GPS[22]等定位技術的進步，使每個有使用定位裝置的行動運算節點都可以知道自己的座標。目前也有很多擇路演算法是假設知道自己的座標而進行擇路演算法設計的(例如 DREAM[23]、LAR[24]、GRA[25]、GPSR[26]跟 GLS[27][28])。由於多了座標的資訊，使這些演算法在搜尋或維持路徑時，可以更節省網路資源，相對的就是提高效率，所以利用座標資訊的擇路演算法通常都會有比較好的效能。

AODV(Ad Hoc On Demand Distance Vector Routing)演算法是一種最常見的單一擇路演算法，擇路的流程也是最基本且易懂，所以常常被拿來當作研究的比較對象。因此在這一小節會介紹 AODV 演算法的基本擇路流程。

AODV 演算法屬於即時搜尋機制演算法之一，所以只有在某個來源節點想要傳送資料封包時，才會進行路徑的搜尋。路徑的搜尋是使用最簡單的擇路要求封包(Route Request - RREQ)與擇路回應封包(Route Reply - RREQ)的往返來完成的。AODV 演算法在決定路徑的考量參數就是到目標點的跳躍數(最短路徑)。另外，AODV 演算法也會在每個節點都儲存一份單一擇路表單記錄擇路搜尋之後的結果。最後要注意的就是 AODV 演算法儲存路徑的方式，AODV 演算法不會記錄整條路徑經過哪些擇路節點，而是記錄到目標節點的下一個跳躍點(路徑上

的下一個擇路節點)，因此下面提到儲存路徑都是代表儲存到該節點的下一個跳躍點。

為了確定選擇的路徑是確實存在於網路內，而不是舊有卻以不存在的路徑，AODV 演算法設計了排序數字(Sequence Number - SN)的觀念。對一條路徑來說，其排序數字越大，代表上一次更新的時間越近，越能確定這條路徑是真的存在於網路之內。尤其在網路變化速度較快的網路之內，路徑存在的時期較短，排序數字的設計對於可以使用正確的路徑傳遞資料封包更是重要。

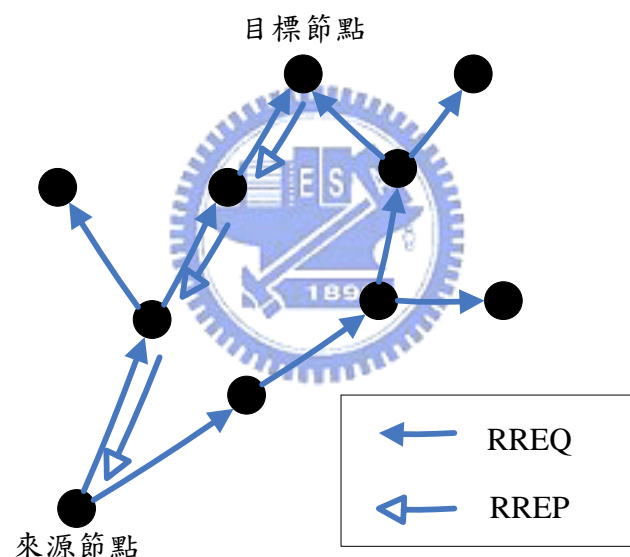


圖 2-4 AODV 演算法示意圖

當來源節點想要傳送資料封包給一個在傳輸範圍外的目標節點，首先搜尋單一擇路表單是否已經存在有到該目標節點的路徑。若是已經存在，就代表已經有路徑可以直接傳送資料封包，所以可以馬上開始資料封包的傳輸。但比較常見的情況就是找不到路徑，便開始搜尋路徑的工作，來源節點會先廣播 RREQ 封包(見圖 2-4)。由於來源節點提出搜尋路徑的要求，也代表網路內的部分路徑也會進行

更新的動作，所以必須在 RREQ 封包內放上最新的排序數字，讓有更新的路徑也藉此更新排序數字(包括通往目標節點的最短路徑)。

當網路內的任何一個中繼節點第一次接收到這個 RREQ 封包，代表這個節點有路徑可以通往來源節點，所以會把通往來源節點的路徑儲存在單一擇路表單內(包括排序數字)，當作之後回傳 RREP 封包的保留路徑。再來檢查自己的單一擇路表單是否有到達目標節點的路徑。如果有，就代表來源節點可以透過這個中繼節點傳送資料到目標節點，所以這個中繼節點就會利用到來源節點的保留路徑回傳 RREP 封包。如果沒有，這個中繼節點就會繼續廣播這個 RREQ 封包(見圖 2-4)。另外要注意，來源節點每一次送出 RREQ 封包，都會有一個固定的編號(廣播 ID)，藉由廣播 ID，收到這個封包的節點都可以知道之前有沒有收過這個 RREQ 封包。當中繼點發現收到的 RREQ 封包是之前曾經接收過的，那就不會對這個封包做任何動作，也不會繼續廣播這個封包。

當目標節點第一次收到 RREQ 封包時，代表目標節點有路徑可以通往來源節點，一樣會把通往來源節點的路徑儲存在單一擇路表單內(包括排序數字)，當作回傳 RREP 封包的保留路徑。由於目標節點已經有回到來源節點的路徑，所以直接利用單一擇路的方式沿著這條路徑回傳 RREP 到來源節點(見圖 2-4)，而這條被選擇的路徑上的中繼節點就稱為擇路節點(Router)。一樣藉由 RREQ 封包的廣播 ID，目標節點可以知道之前有沒有收過這個 RREQ 封包。當目標節點發現收到的 RREQ 封包是之前曾經接收過的，那就不會對這個封包做任何動作。另

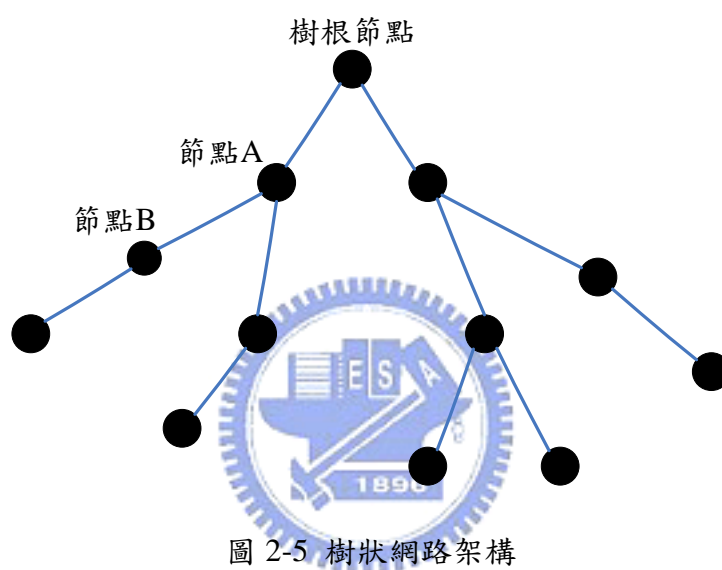
外要注意，雖然目標節點所選擇的路徑就是最小的保留路徑，而保留路徑上所有擇路節點的單一擇路表單也都存有這條保留路徑，但對這條保留路徑上的每個擇路節點來說，這個保留路徑是通往來源節點的路徑，而非通往目標節點的路徑(因為路徑的方向不同，所以上一個跳躍點會不同)。所以我們依然要回傳 RREP，通知這條路徑上的節點，使這些中繼節點知道自己成為擇路節點，並得到通往目標節點的路徑，而保留路徑的功用只是提供回傳 RREP 的路徑。

當被選擇路徑上的擇路節點接收到 RREP 封包時，代表這個節點有路徑通往目標節點，而且已經是被目標節點決定要使用的路徑，所以要把這條通往目標節點的路徑儲存在單一擇路表單，這樣才能在之後傳遞由來源節點所傳出的資料封包。接著再利用單一擇路表單找出通往來源節點的保留路徑(之前在 RREQ 廣播所儲存的)，繼續傳遞 RREP，反覆步驟直到來源節點收到 RREP 為止。當來源節點收到 RREP 之後，代表來源節點已經有路徑通往目標節點，把路徑的儲存在單一擇路表單內之後，就可以開始資料封包傳輸的工作。

AODV 演算法其實也有一些簡單的維持機制。在網路內的所有行動節點會週期性的廣播一個跳躍數的維持訊息(Maintain Message)封包，藉此確定網路內任意二點間的連結是否斷裂。在這種會不斷確定二點間的連結的機制下，當路徑上的某個連結斷裂時，在這個連結上比較靠近來源節點的擇路節點會知道這條路徑已經斷裂，就回傳通知封包告知來源節點路徑斷裂的訊息。但這是屬於比較簡單的維持機制，所以 AODV 演算法依舊被歸類在及時搜尋機制演算法內。

2.2.2 多重擇路演算法

之前提到多重擇路演算法利用少數的控制封包，先把所有的來源節點跟目標節點之間有計畫地建立最省資源的路徑架構，再進行資料封包的傳輸。而多重擇路演算法依照路徑架構的不同大概可以分成樹狀網路架構演算法(Tree-based Protocol)跟網狀網路架構演算法(Mesh-based Protocol)二種不同的架構。



由名稱可知，樹狀網路架構演算法(例如 MAODV[29]、AMRIS[30]、AMRoute[31]跟 GS[32])的路徑架構成樹狀型(見圖 2-5)，會有一個行動運算節點代表樹根的位置，稱為樹根節點。除了樹根節點，其他網路結構內的節點都會有一個上游節點是通往樹狀網路架構的路徑，也就是通往樹根節點的路徑。在圖 2-5 中，節點 A 為節點 B 的上游節點。相對的，這個行動運算節點也會被其他行動運算節點視為上游節點，對這個被視為上游節點的行動運算節點來說，這些視其為上游節點的行動運算節點就稱為下游節點。在圖 2-5 中，節點 B 為節點 A 的下游節點。要注意的是每一個節點的下游節點數量是不一定的。利用這種架構

產生的網路架構，每一點都彷彿是樹枝的交叉處，往上游節點就代表是往樹根方向，往下游節點就代表是往樹葉方向，形狀看起來彷彿就是一棵樹，故稱此網路架構為樹狀網路架構。樹根節點因為沒有任何上游節點，在樹狀圖中，彷彿位在樹根的位置，因此被稱為樹根節點。網路架構內的部分節點因為沒有任何下游節點，在樹狀圖中，彷彿位在樹葉的位置，因此被稱為樹葉節點。觀察這種結構可以發現，樹根節點幾乎位在整個網路架構的中心位置，所以在樹狀網路架構的演算法中，樹根位置往往都要負責整個網路架構的維持。由於每個行動運算節點只有一個上游節點，代表每個行動運算節點只有一條路徑通往整個網路架構，使用到最少的路徑跟擇路節點，對傳送資料封包來說，這是最省網路資源的多重擇路演算法，這是樹狀網路架構最大的優點。但由於所有樹狀網路架構內的成員要通往網路架構的路徑都只有一條，處於較動態的網路時，唯一的路徑容易斷裂，因此樹狀網路架構的路徑可靠性比較差。而且當路徑斷裂時，必須進行重新尋找路徑到網路架構的動作，控制封包的數量會因此大幅上升。

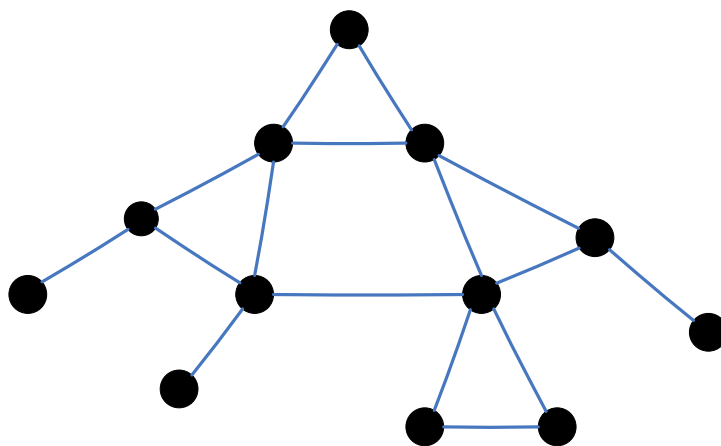


圖 2-6 網狀網路架構

網狀網路架構演算法(例如 ODMRP[33]、CAMP[34]跟 MCEDAR[35][36])是為了避免像樹狀網路架構的路徑可靠性較差的缺點，每一個行動運算節點使用更多條路徑通往整個網路架構。圖 2-6 的節點分佈與圖 2-5 一模一樣，但路徑數卻增加不少，路徑分佈的形狀很像一張網子，故稱為網狀網路架構。當某一條路徑因為網路的變動而斷裂時，仍然會有其他條路徑幫忙傳送資料封包，提高路徑的可靠度。而且路徑不只一條，所以當路徑斷裂時，不需要立即進行搜尋新路徑的動作，所以控制封包的數量可以做有效的控制，不會因為處在叫動態的網路，造成控制封包快速增加。但由於網路多了許多本來不需要的路徑，所以在傳送資料封包時，會消耗較多的網路資源，而這些因為增加路徑而產生的多餘負荷，就稱為多餘資源(Redundancy)。



樹狀網路架構的特色在於使多餘資源降到最低，節省網路資源，但路徑的可靠性卻不佳，並且在動態的網路下，控制封包會大量的提升。網狀網路架構的特色在於犧牲多餘資源的負荷，但路徑的可靠度大量提升，並且在動態的網路下，控制封包仍然維持在一定的數量。在一些研究中[37][38]，有針對這二種多重擇路演算法作分析，在樹狀網路架構演算法的部分選擇 MAODV 演算法，而在網狀網路架構演算法的部分選擇 ODMRP 演算法。分析的結果可以發現，雖然樹狀網路架構演算法少了許多的多餘資源，但在較動態的網路內，由於控制封包的大量增加，造成資料封包的傳送效率也會大幅的降低，甚至比多餘資源較多的網狀網路架構演算法還糟糕。所以目前在多重擇路演算法的設計，逐漸的會比較偏重

從網狀網路架構演算法的方式進行設計。

MAODV(Multicast Operation of the Ad Hoc On Demand Distance Vector Routing) 演算法是一種最常見的樹狀網路架構演算法，擇路的流程是利用最基本的擇路要求與擇路回應循環，可以說是其他樹狀網路架構演算法的基礎，所以也常常被拿來當作研究的比較對象。因此在這一小節會介紹 MAODV 演算法建立整個網路架構的方式。

第一個想要建立這個樹狀網路架構的行動運算節點稱為網路架構領導節點 (Structure Leader)。一般來說，在整個網路架構完全建立後，網路架構領導節點會在整個樹狀架構的樹根節點位置。跟 AODV 演算法使用排序數字(Sequence Number)的想法一樣，MAODV 演算法也有使用結構排序數字(Structure Sequence Number)來決定每條擇路的更新程度，當路徑的架構排序數字越大，就代表該條路徑距離上次更新的時間越近，越符合現在整個網路應該保持的路徑結構。而整個樹狀網路架構就是利用網路架構領導節點所廣播的架構維持訊息(Structure Maintain message)來完成架構排序數字更新的動作，並使樹狀網路架構可以保持在最省網路資源的架構。

每一個樹狀網路架構都有一個獨特的網路架構位址，而架構內的每個行動運算節點都需要建立一份多重擇路表單(Multicast Route Table)。在多重擇路表單之內，除了基本的網路架構位址與架構排列數字之外，還需要儲存該行動運算節點通往樹狀網路架構的所有路徑。跟 AODV 演算法一樣，路徑的儲存方式就是記錄該

條路徑的下一個跳躍點。這裡要注意，這裡要儲存的路徑不像 AODV 演算法的路徑單純地拿來給資料封包傳遞使用的，這些儲存的路徑在 MAODV 演算法內主要是被用來進行網路架構的維持，所以這裡的路徑指的是通往樹狀網路架構的路徑(在之後的敘述中會慢慢瞭解)。在 AODV 演算法內，封包在路徑上的傳遞，是利用擇路表單內存的下一個跳躍點，進行一對一的傳輸，非下一個跳躍點卻收到封包的節點，會在媒體擷取控制層就把該封包給取消，不需要再往上送到網路通訊層。MAODV 演算法的資料封包傳輸與 AODV 演算法不同，封包的檢查與取消都是在網路通訊層進行的，這在之後會有更詳細的說明。另外針對每個下一個跳躍點都要設立一個擇路使用標誌(Route Enabling Flag)，藉此來標示有通過此下一個跳躍點的路徑是否被啟用，就是標示該行動運算節點通往樹狀網路架構的路徑是否有通過此下一個跳躍點。



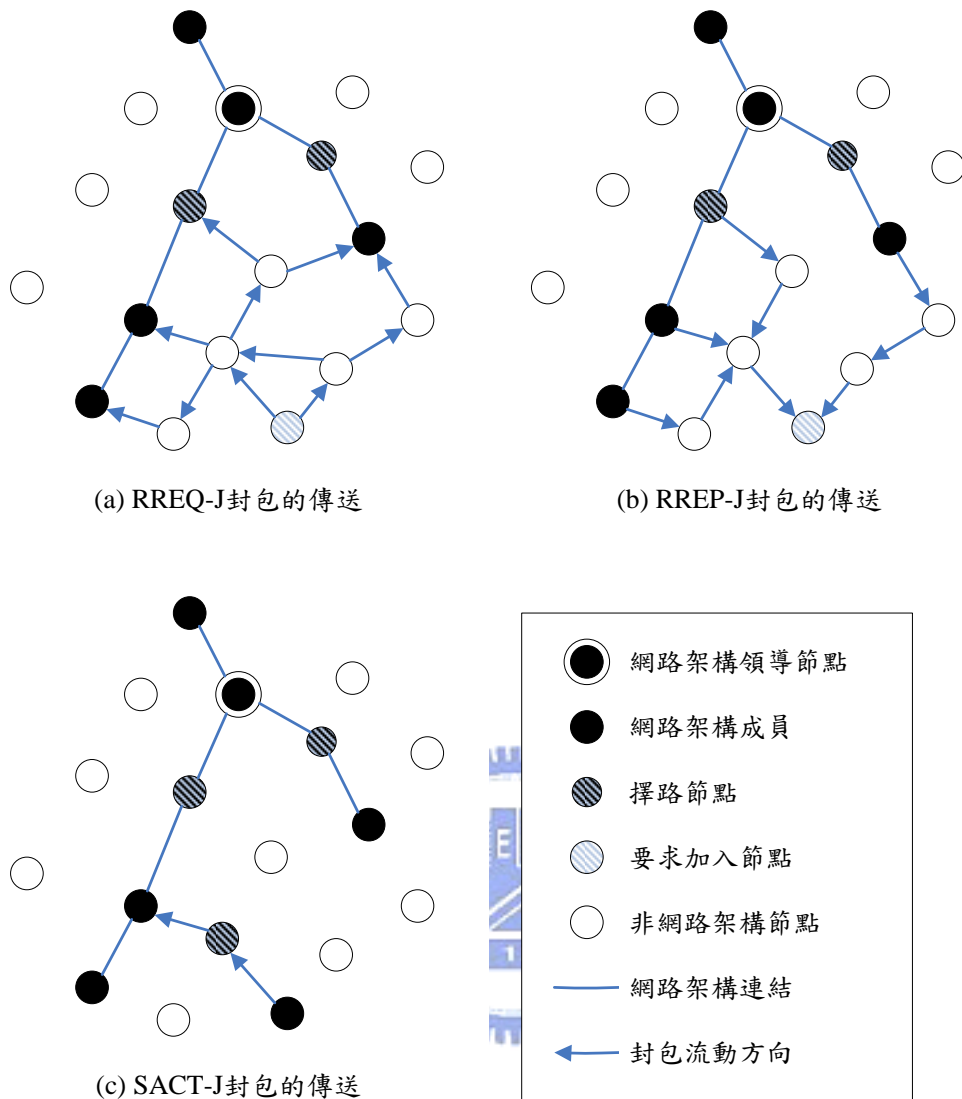


圖 2-7 MAODV 演算法流程示意圖

AODV 演算法的擇路搜尋方式是利用擇路要求封包(Route Request - RREQ)跟擇路回應封包(Route Reply - RREP)的往返來達成的。而 MAODV 演算法就是利用類似 AODV 演算法的方式來達到網路架構的建構與加入架構。當有一個行動運算節點(稱為要求加入節點)試圖加入樹狀網路架構時，要求加入節點會先廣播 RREQ-J 封包(RREQ with a Join flag)搜尋通往樹狀網路架構的路徑(見圖 2-7

(a))。在 RREQ-J 封包在網路內傳遞的過程中，會利用單一擇路演算法的方式來決定並儲存該點回到要求加入節點的最小路徑。當樹狀網路架構內的任何一個網路架構成員接收到這個廣播的 RREQ-J 時，會利用在 RREQ-J 傳遞時產生的最小路徑，利用單一擇路演算法的方式回傳 RREP-J 封包(RREP with a Join flag)給要求加入節點(見圖 2-7 (b))。這裡要注意，由於任何接收到 RREQ-J 封包的樹狀網路架構之成員都可以利用單一擇路的方式回傳 RREP-J 封包，所以要求加入節點將會接收到很多條利用單一擇路演算法的方式而經過不同路徑的 RREP-J 封包，其中每收到一次，就代表在樹狀網路架構中有一個相對應的點回應了 RREQ-J 封包。當要求加入節點同時收到來自不同路徑的 RREP-J 時，先利用 RREP-J 封包上的紀錄，在多重擇路表單內儲存這些路徑相關於樹狀網路架構的資訊，包括這條路徑的上一個跳躍點、樹狀網路架構最新的架構排序數字跟距離網路架構領導節點的跳躍數。

最後要求加入節點會在之前儲存的路徑中選擇一條最短的路徑(依據距離網路架構領導節點的跳躍數)，並把多重擇路表單內相對應於被選擇路徑項目中的 REF 設定起來，代表這條路徑將被選擇使用。最後一樣利用單一擇路演算法的方式，使用這幾條路徑回傳 SACT-J(Structure Activation message with a Join flag)封包，傳送給當初利用這些路徑回應 RREP-J 封包的節點(稱為回應節點)，讓回應節點知道這些路徑已經被要求加入節點決定為通往樹狀網路架構的路徑(見圖 2-7 (c))。在回傳路徑上收到 SACT-J 封包的中繼節點，代表該中繼節點已經被選

擇成為樹狀網路架構內的成員，稱為擇路節點。這些擇路節點會利用 SACT-J 的資訊，把上一個跳躍點儲存在多重擇路表單內，接著利用單一擇路表單的資訊，把通往回應節點的下一個跳躍點一樣儲存在多重擇路表單內，並向此下一個跳躍點送出 SACT-J 封包。這裡要注意，不同於 AODV 演算法，擇路節點必須把路徑上的前後跳躍點都儲存在多重擇路表單內，這是因為多重擇路表單的路徑是代表通往樹狀網路架構的路徑，而不只是通往網路架構領導點的路徑，新加入的節點也算是樹狀網路架構的一部份，所以通往該節點的路徑也要儲存起來。如此反覆動作，直到回應節點收到 SACT-J 封包。

樹狀網路架構的維持利用二種廣播維持封包的機制：

- 1) 網路架構領導節點 (Structure Leader) 會負責週期性的廣播架構維持訊息 (Structure Maintain Message) 封包，藉此來更新一些有關整個合成網路架構的資訊(例如樹狀網路架構的架構排列數字)。
- 2) 為了確定網路內任意二點間的連結是否斷裂樹狀網路架構內的所有成員都會週期性的廣播一個跳躍數的維持訊息 (Maintain Message) 封包，讓連結的另一邊知道這個連結依舊存在。如果樹狀網路架構中的某一點在一段時間內都沒有收到連結另外一邊送來的維持訊息封包，那此行動運算節點就會認定這個連結斷裂，並針對這種情況做出反應。

之前提到 AODV 演算法的資料封包傳遞是利用擇路表單內存的下一個跳躍點，進行一對一的傳輸，非下一個跳躍點卻收到封包的節點，會在媒體擷取控制

層就把該封包給取消，不需要再往上送到網路通訊層。但在 MAODV 演算法做資料封包傳遞時，是不需要在多重擇路表單內找下一個傳輸點的，而是利用一個跳躍點的廣播方式送出封包，封包內會有網路架構位址的資訊，接收到封包的節點只需要藉由網路架構位址判斷自己是不是屬於同一個樹狀網路架構。如果是，就接收封包並再用一樣的方式送出這個封包，如果不是，就把封包取消。因此 MAODV 演算法不同於 AODV 演算法在媒體擷取控制層就決定要不要接收資料封包，而是在網路通訊層才做決定。此外，每個樹狀網路架構內的成員對於同一個資料封包，只會做一次的接收與傳送。



第三章 能量分散擇路演算法(Distributed Energy Routing Protocol - DER)

3.1 設計考量

一般的單一擇路演算法都是考慮電池能量無限制的情況，再針對跳躍數或是其他因素設計單一擇路演算法。目前在電池能量領域上的發展，儘管電池能量的容量已經大大提升，但還是不足以無限制時間地維持整個行動運算節點的電路運作。因此在電池能量有限制的情況下，行動運算節點將會因為電池能量用完而產生一些問題。如果某條路徑上的中繼節點剛好把所有的電池能量使用完，那就會造成路徑斷裂無法繼續做傳遞資料封包的工作。甚至是在這個中繼節點處在整個網路中重要位置的情況下，使得網路中的某些路徑必須透過這個中繼節點才能持續地傳遞資料封包，這時候有限電池能量產生的問題就會更加的嚴重。但也因為這些行動運算節點處在網路中的重要位置，在只考慮跳躍數的單一擇路演算法中，被選擇當作擇路節點的機率也相對的比較高，所以前述的問題也更容易發生。因此要設法讓每條路徑的選擇可以平均地分佈在網路中，避免部分行動運算節點不斷地被選擇當作擇路節點而造成電池能量快速地被使用完，而這樣的概念也代表著把傳遞資料所造成的能量損失平均地分佈在網路中。一般來說，剩餘電池能量較少的節點代表常常被使用到，所以通常就是代表那些處在網路中重要位置的行動運算節點。因此當來源節點提出路徑搜尋的要求時，除非必要，不然會

盡量避免使用到剩餘電池能量較少的節點，這樣就可以很容易地盡量不要使用到重要位置的行動運算節點。DER 演算法便是考量電池能量的因素而設計的單一擇路演算法。

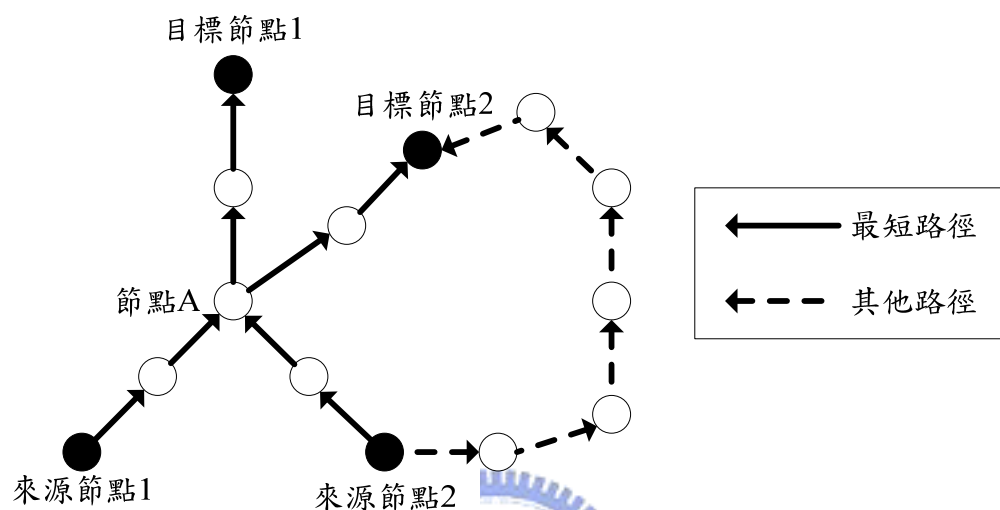


圖 3-1 路徑選擇方式

見圖 3-1，節點 A 因為處在網路中的重要位置而容易擔任擇路節點的工作，因此電池能量通常較少。假設來源節點 1 正在傳送資料封包給目標節點 1。此時來源節點 2 也想傳送資料封包給目標節點 2，依照網路中所有行動運算節點的分佈與一般只考慮跳躍數的單一擇路演算法，來源節點 2 會選擇最短路徑(見圖 3-1 的實現箭頭)，而中繼節點 A 會同時成為這條路徑與其他路徑的共用擇路節點，必須承擔比較多次的封包傳遞動作，會造成節點 A 的電池能量容易被使用完。一旦節點 A 的電池能量使用完而不能幫忙傳遞資料封包，那對來源節點 1 來說，將找不到其他路徑可以進行傳遞資料封包的工作，因此嚴重的影響資料封包傳遞的成功率。若單一擇路演算法有考慮到電池能量的因素，那來源節點 2

就可能會捨棄最短路徑而選擇跳躍數較長的路徑(見圖 3-1 的虛線箭頭)傳送資料封包給目標節點 2，避免節點 A 被重複使用，造成電池能量快速耗損。

3.2 能量分散擇路演算法設計

因為 AODV 演算法假設電池能量無限制而只有考慮到跳躍數，所以在剩餘電池能量的限制下，效能將會大大的下降。DER 演算法是以 AODV 演算法為基礎，加上剩餘電池能量的因素改良演算法，達到 3.1 節敘述的概念。

DER 演算法跟 AODV 演算法類似，路徑的搜尋是使用最簡單的擇路要求封包(Route Request - RREQ)與擇路回應封包(Route Reply - RREQ)的往返來完成的。由於 DER 演算法要使目標節點在決定路徑時，是以剩餘電池能量為考量，所以 DER 演算法針對 AODV 演算法做了三點改良。第一、剩餘電池能量的考量方式。第二、最佳路徑的選擇。第三、反應計時器的設計。

3.2.1 剩餘電池能量的考量方式

DER 演算法在 RREQ 封包內，增加了路徑電池能量最小值的欄位。當中繼節點接收到 RREQ 封包時，會先判斷 RREQ 封包內的路徑電池能量最小值欄位是否大於自己的電池量。如果條件符合，那該中繼節點就會瞭解自己的電池量是目前這條路徑的最小值，所以把自己的電池量存在 RREQ 封包的路徑電池能量最小值欄位並廣播這個 RREQ 封包。反之，該中繼節點就直接廣播這個 RREQ

封包。藉由這種方式，會把每條路徑上所有中繼節點的最小電池量存在 RREQ 封包的路徑電池能量最小值欄位，使目標節點可以透過這個資訊來選擇擁有最大的路徑電池能量最小值的路徑，而這條路徑將比較不容易使用到處於網路重要位置的中繼節點。

3.2.2 最佳路徑的選擇

AODV 演算法決定路徑的方式是選擇最少跳躍數的路徑。因為要選擇最少的跳躍數，所以在廣播 RREQ 封包的過程中，當目標節點或是中繼節點第一次接收到來源節點廣播的 RREQ 封包時，就認定這條 RREQ 封包經過的路徑為最短路徑。因此，當目標節點或是中繼節點收到一次 RREQ 封包之後，就不再處理之後接收到的同一個 RREQ 封包。但 DER 演算法首先考量剩餘電池能量，因此不能跟 AODV 演算法一樣只記錄第一次收到 RREQ 封包的路徑。必須持續對之後收到的 RREQ 封包做處理與記錄路徑的工作。

在 DER 演算法中，當接收到同一個 RREQ 封包時，會先判斷傳送這一條 RREQ 封包的路徑是否有比較好的剩餘電池能量狀況，如果有比之前儲存在單一擇路表單內的路徑有比較充足的剩餘電池能量，那就會把這一條路徑視為通往來源節點的最佳路徑，並儲存在單一擇路表單之內。接著進行 3.2.1 節敘述的動作，最後再次把 RREQ 封包廣播出去，讓之後的中繼節點也知道現在有這一條比較好的路徑可以選擇。這裡要注意的是，RREQ 封包資訊的更改，是在判斷路徑好

壞之後，也就是說在判斷路徑好壞的時候，是還沒有考慮到本身剩餘電池能量的問題，而是只有考慮路徑的其他節點。如果沒有比之前儲存在單一擇路表單內的路徑有比較充足的剩餘電池能量，那就會把這一條路徑視為比較差的路徑，是不可能會被目標節點選擇的，因此直接取消這一個 RREQ 封包。

由圖 3-2 可以觀察到，當中繼節點 B 第一次收到來自中繼節點 C 的 RREQ 封包，由於是第一次收到這個 RREQ 封包，所以把路徑的資訊儲存下來。並更改在 RREQ 封包內有關路徑剩餘電池能量的資訊，最後再廣播這個 RREQ 封包出去。當中繼節點 B 接著收到來自中繼節點 D 的 RREQ 封包，發現這一條路徑的剩餘電池能量狀況比之前儲存的路徑還要良好，就會把之前儲存的路徑刪掉，並儲存這一條新的路徑，再更改在 RREQ 封包內有關路徑剩餘電池能量的資訊，最後在廣播這個 RREQ 封包出去。重複動作，可以發現中繼節點 A 與目標節點都會做路徑的更新，只儲存最佳路徑。因此藉由這種方式，網路中所有節點都可以得到通往來源節點的最佳路徑。

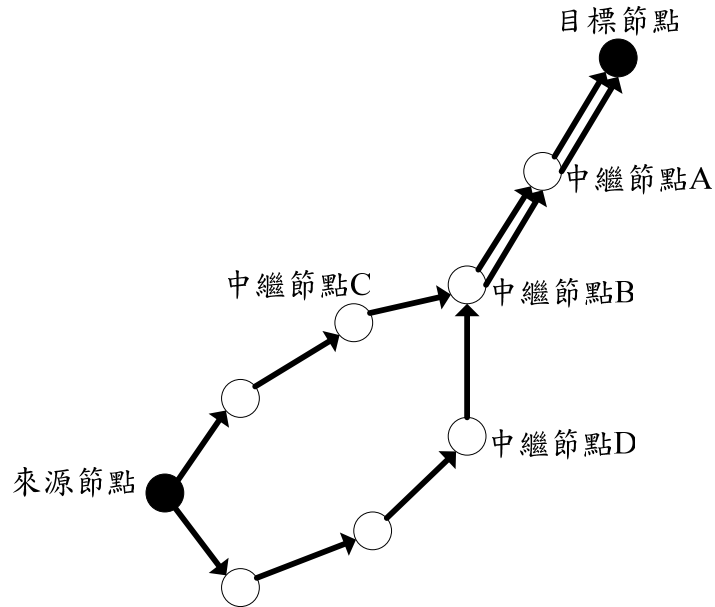


圖 3-2 DER 演算法的 RREQ 封包傳遞方式

另外要注意，用上述敘述的演算法，也可以避免擇路迴圈(Route Loop)的情況。以二點間的擇路迴圈為例，在圖 3-2 中，當中繼節點收到來自中繼節點 C 的 RREQ 封包，把這一條路徑的資訊存在單一擇路表單內，這裡稱為路徑 1。接著中繼節點 B 傳送 RREQ 封包給中繼節點 A，中繼節點 A 再傳送 RREQ 封包給中繼節點 B，因此中繼節點 B 就會認為這一條路徑是另一條可能可以使用的路徑，開始路徑剩餘電池能量的判斷，這裡稱為路徑 2。我們可以發現，如果中繼節點 A 或中繼節點 B 的電池剩餘能量狀況比路徑 2 的其他節點都還要差，由於路徑 1 沒有經過中繼節點 A 與中繼節點 B，所以透過 RREQ 封包內的資訊，中繼節點 B 就會認為路徑 2 的電池剩餘能量狀況比路徑 1 還差，並消除這個 RREQ 封包。如果中繼節點 A 與中繼節點 B 的電池剩餘能量狀況不會比路徑 2 的其他節點都還要差，所以在路徑 1 跟路徑 2 上，電池剩餘能量最差的中繼節點會是在路徑中有重複的地方。因此對中繼節點 B 來說，這二條路徑的狀況會是一樣的，不會

做單一擇路表單的更改。由此可以發現，擇路迴圈的現象會很自然的被解決。利用同樣的分析方式，就可以發現三點以上的擇路迴圈問題也都不會產生。

3.2.3 反應計時器的設計

在 AODV 演算法中，當目標節點第一次接收到來源節點廣播的 RREQ 封包時，就認定這條 RREQ 封包經過的路徑為最短路徑，立刻使用這條路徑回傳 RREP 封包通知來源節點，而且不理會之後接收到的同一個 RREQ 封包。這種方式明顯不適用在 DER 演算法。因為經過不同路徑的 RREQ 封包到達目標節點的時間不同，所以必須讓目標節點回傳 RREP 封包的時間延遲，使目標節點可以得到每一條路徑的資訊並做更新成最佳路徑的動作。因此當目標節點第一次接收到來源節點廣播的 RREQ 封包時，目標節點不會馬上使用這條路徑回傳 RREP 封包，而是設定一個負責倒數的回應計時器(Reply Timer)。在回應計時器倒數到零的這段時間內，持續的接收 RREQ 封包得到新的路徑資訊並更新成最佳路徑。當計時器倒數為零時，目標節點就假設已經檢查過所有通往來源節點的路徑，且單一擇路表單內的路徑就是最佳路徑。最後目標節點就利用這一條最佳路徑回傳 RREP 封包。

第四章 功率控制混合式多重擇路演算法

(Power-Controlled Hybrid Multicast Routing - PCHMR)

4.1 設計考量

在某些實際的情況下，行動運算節點的移動是有群組特性的，這種移動模式就稱為群組移動模式。整個網路內的行動運算節點在移動的過程中，根據其移動的行為，大概可以分成好幾個不同的小群組。而在同一群組內的行動運算節點，因為有相類似的移動行為，節點間相對距離與相對移動速度較小，所以可以視為一個節點數較少且相對移動較靜態的網路結構，因此在每一個群組內的連結相對地可靠不易斷裂。相較之下，群組之間的相對移動比較動態且多變化，彼此之間的連結也更容易斷裂而不可靠。PCHMR 演算法就是針對此種群組移動模式而設計的演算法。

PCHMR 演算法是依據樹狀網路架構與網狀網路架構二種不同的概念，經過適當的設計組合成一種新的網路架構，在此稱為混合式網路架構(Hybrid Structure)。PCHMR 演算法主要的觀念就是根據二個行動運算節點是否在同一個群組內的條件，決定使用樹狀網路架構或是網狀網路架構的概念進行路徑連結的工作。由於樹狀網路架構的連結數且需要的控制封包較少，但只能適用在相對移動較靜態而連結不容易斷裂的網路結構。所以當二個行動運算節點處在同一個群

組之內時，使用樹狀網路架構可以降低路徑連結的數量與控制封包，並足以達到此二點間路徑連結的可靠性，因此群組內行動運算節點間的路徑連結只需採取樹狀網路架構的模式。當二點行動運算節點處在不同群組之內時，由於群組之間相對的移動比較動態，路徑連結自然也相對地容易斷裂，所以採取連結較可靠而不易斷裂的網狀網路架構。雖然網狀網路架構需要更多的多餘連結數，但由於整個網路內的群組數量遠小於行動運算節點的數量，所以 PCHMR 演算法需要使用到網狀網路架構的路徑連結數量遠小於直接使用網狀網路架構演算法的全部路徑連結數量。因此，使用網狀網路架構而增加的路徑連結數被大大的降低而產生連結數量最小化的情況。



PCHMR 演算法是利用樹狀架構網路為主，網狀網路架構為輔。先利用樹狀網路的架構方式為基礎，讓網路內所有的行動運算節點產生路徑連結。在一般的樹狀網路架構內(例如 MAODV 演算法)，每一個行動運算節點只有一個上游節點 (Upstream Node)可以當作往樹狀網路架構的領導節點的回溯路徑。如此先確定在群組之內的行動運算節點已經形成樹狀網路架構。在此基礎之下，另外考慮群組之間比較靠近的行動運算節點，使這些節點除了含有在樹狀網路架構下的路徑連結，在群組之間還增加了網狀網路架構的路徑連結，使群組之前的路徑連結更加的可靠不易斷裂。

由圖 4-1 可以瞭解針對群組移動模式而設計的 PCHMR 演算法是如何運作的。整個網路的行動運算節點根據移動的行為被分成三個小群組，分別是 A 群

組、B 群組跟 C 群組，實線代表依據樹狀網路架構而產生的連結。由於三個群組

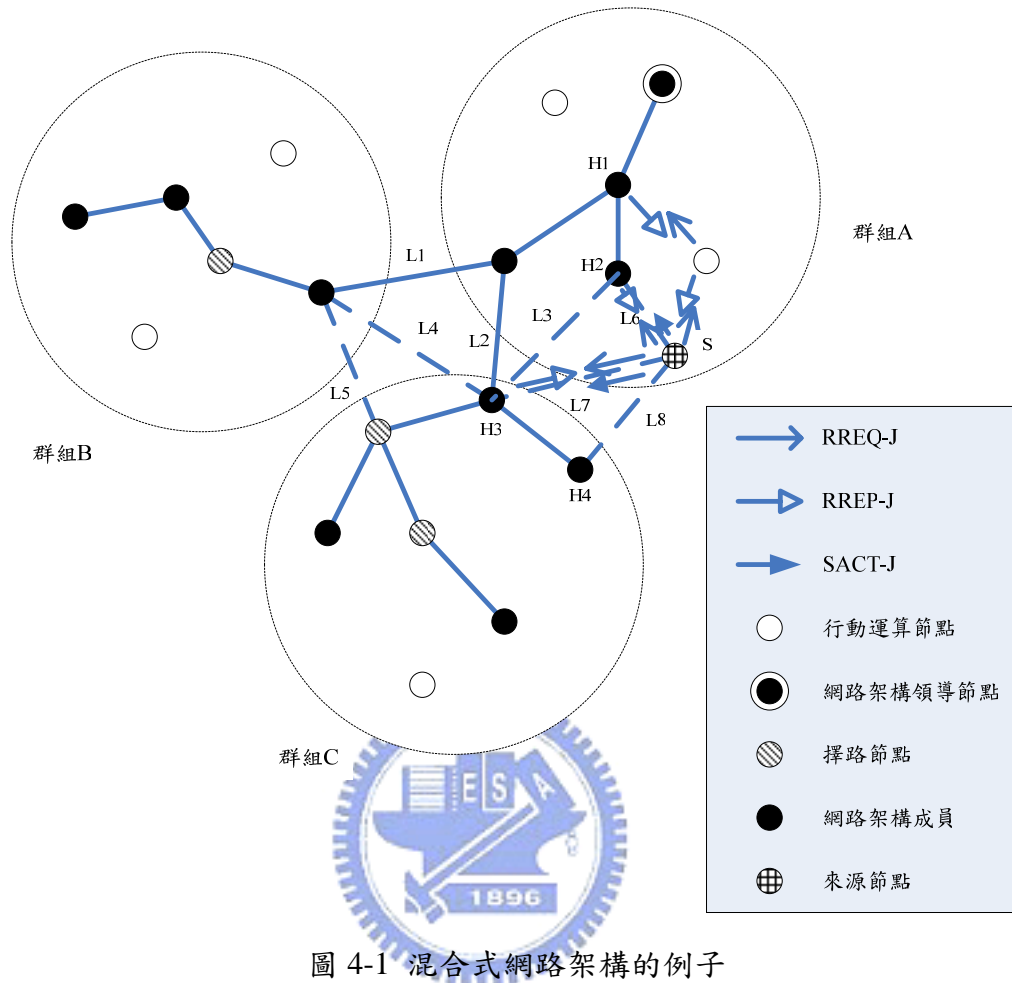


圖 4-1 混合式網路架構的例子

之間移動行為的不同，造成群組之間的連結容易斷裂(i.e. 群組A與群組B的 L_1 跟群組A與群組C的 L_2)。在PCHMR演算法內，群組之間另外增加了網狀網路架構的路徑連結，確保群組之間的路徑連結更加的可靠不易斷裂。在圖4-1內，PCHMR演算法增加了 L_3 、 L_4 跟 L_5 這三個路徑連結來提高三個群組間的連結可靠性。因此當資料封包在傳遞時，即使當群組間部分的路徑連結因為群組間不同的移動行為而造成斷裂時，仍然會因為另一部份增加的路徑連結，所以群組之間的資料封包傳遞可以持續地進行，大大地增加資料封包在群組間傳遞的成功率。所以說PCHMR演算法是在樹狀網路架構的基礎下，利用網狀網路架構的方式提高群組

間的路徑連結可靠性而產生的混合式網路結構。

4.2 功率控制混合式多重擇路演算法設計

PCHMR 演算法以樹狀網路架構演算法的 MAODV 演算法為基礎，再把利用 MAODV 演算法產生的樹狀網路架構加入部分網狀網路架構的觀念，產生新的混合式網路架構。每一個混合式網路架構有一個獨特的網路架構位址，並且由網路架構成員(Structure Member，圖 4-1 內的實點)與擇路節點(Router，圖 4-1 內的陰影點)所組成。網路架構成員代表真正屬於這個網路架構的行動運算節點，想要接收屬於這個網路架構內的封包。擇路節點代表為了組成混合式網路架構而必須加入網路架構內的行動運算節點，當網路架構內的封包在傳遞時，要適當的擔任中繼節點的工作。要注意擇路節點不等於網路架構成員，儘管都屬於混合式網路架構內的成員，但實質上的意義是不同的。而混合式網路架構內的成員就統稱為混合式網路架構成員(Hybrid Structure Member)。

第一個建立這個混合式網路架構的行動運算節點稱為網路架構領導節點。此網路架構領導節點(Structure Leader)要負責維持混合式網路架構的網路結構。類似於MAODV使用排序數字(Sequence Number)的想法，PCHMR演算法使用了架構排序數字(Structure Sequence Number - SSN)來決定每條路徑的更新程度，當路徑的SSN越大，就代表距離該條路徑上次更新的時間越近(i.e. $SSN(t_i) \geq SSN(t_j)$ for $t_i \geq t_j$)，越符合現在整個網路應該保持的路徑連結結構。整個混合式網路架構更

新SSN的工作會由網路架構領導節點來完成。

4.2.1 演算法的表單設計

每個行動運算節點內都需要建立一份多重擇路表單(Multicast Route Table - MRT)。在多重擇路表單之內，除了基本的網路架構位址與 SSN，還需要儲存該行動運算節點所處群組的 ID 與通往混合式網路架構的所有路徑中的下一個跳躍節點群(通常不只一個)。另外針對每個下一個跳躍點都要設立一個擇路使用標誌(Route Enabling Flag - REF)，藉此來標示此下一個跳躍點的路徑是否被使用，也就是標示該行動運算節點通往混合式網路架構的路徑是否有通過此下一個跳躍點。



4.2.2 混合式網路架構的建構與加入架構流程

AODV 演算法的擇路搜尋方式是利用擇路要求封包(Route Request - RREQ)跟擇路回應封包(Route Reply - RREP)的往來溝通來達成的。混合式網路架構就是利用類似 AODV 演算法的方式來達到網路架構的建構與加入架構。

當有一個行動運算節點試圖加入某個混合式網路架構時，此節點就稱為要求加入節點。要求加入節點會先廣播 RREQ-J 封包(RREQ with a Join flag)搜尋通往混合式網路架構的路徑。在 RREQ-J 封包在網路內傳遞的過程中，接收到 RREQ-J 封包的中繼節點會利用單一擇路演算法的方法來決定該中繼節點回到要求加入

節點的最小路徑。當混合式網路架構內的混合式網路架構成員(包括網路架構成員跟擇路節點)接收到 RREQ-J 封包時，會利用在 RREQ-J 封包傳遞時產生的最小路徑，以單一擇路演算法的方法回傳 RREP-J 封包(RREP with a Join flag)給要求加入節點。這裡要注意，由於任何接收到 RREQ-J 封包的混合式網路架構成員都可以利用單一擇路演算法回傳 RREP-J 封包，要求加入節點將會接收到很多個利用單一擇路演算法而經過不同路徑的 RREP-J 封包。每收到一次 RREP-J 封包，就代表在混合式網路架構中有一個相對應的混合式網路架構成員回應了 RREP-J 封包。當行動運算節點同時收到來自不同路徑的 RREP-J 封包時，先把 RREP-J 封包上的關於通往混合式網路架構的路徑資訊儲存在多重擇路表單內，當中包括這條路徑的下一個跳躍點、混合式網路架構最新的 SSN、回傳 RREP-J 封包的混合式網路架構成員所屬的群組 ID 跟到該點的跳躍數。最後要求加入節點會適當的選擇幾條路徑(選擇路徑的方式會在後面說明)，並把多重擇路表單內相對應於被選擇路徑項目的 REF 設定起來，代表這條路徑已被選擇使用。最後一樣利用單一擇路演算法，使用這幾條路徑回傳 SACT-J(Structure Activation message with a Join flag)封包，傳送給當初利用這些路徑回應 RREP-J 封包的混合式網路架構成員，讓這些節點知道這些路徑已經被要求加入節點決定為通往混合式網路架構的路徑。

以圖 4-1 為例，當要求加入節點 S 想要加入混合式網路架構的群組 A 時，首先要求加入節點 S 先廣播 RREQ-J 封包(見圖 4-1 的箭頭)。當混合式網路架構

成員(圖 4-1 內的 H_1 、 H_2 跟 H_3)收到 RREQ-J 封包時，便使用單一擇路演算法回傳 RREP-J 封包(見圖 4-1 的箭頭)。當要求加入節點 S 收到這三條路徑傳來的 RREP-J 封包時，先利用 RREP-J 封包上的紀錄，儲存這三條路徑相關於混合式網路架構的資訊，包括這條路徑的下一個跳躍點(N_{H_j})、混合式網路架構最新的 SSN(SSN_{H_j})、回傳 RREP-J 封包的混合式網路架構成員所屬的群組 ID(GID_{H_j})跟到該混合式網路架構成員的跳躍數(HC_{H_j})。在要求加入節點 S 適當的選擇之後(在後面會利用圖 4-2 說明)，決定建立通往 H_2 跟 H_3 的二條路徑(L_6 跟 L_7)，於是要求加入節點 S 先把多重擇路表單內相對應於通往 H_2 跟 H_3 路徑項目中的 REF(REF_{N_i})設定成驅動狀態，代表這二條路徑已被選擇使用。最後使用單一擇路演算法，從這二條路徑回傳 SACT-J 封包，啟用這二條路徑(見圖 4-1 的箭頭)。

平均接收功率(\bar{P}_{H_j})是在 PCHMR 演算法決定使用哪幾條路徑回傳 SACT-J 封包的過程中一個很重要的參數，藉由這個參數，將可以確保被選擇的路徑連結具有比較高的可靠性，也可以摒除狀況較差的路徑，減少不必要的封包傳輸。在 RREP-J 封包使用單一擇路演算法傳遞封包的過程中，路徑上的每一個中繼節點會進行計算平均接收功率的工作並記錄在 PPEP-J 封包內。在每一條被選擇回傳 RREP-J 封包的路徑上，每一次傳輸的接收端都要利用這一次傳輸的接收功率跟記錄在接收到的 RREP-J 封包內的平均接收功率再做一次平均接收功率的計算並更新(見算式 4-1)。最後再把 $\bar{P}_{H_j}(HC_{H_j})$ 記錄到即將送出的 RREP-J 封包內。藉由這種不斷計算平均接收功率並記錄在 RREP-J 封包內送到下一個跳躍點的方式，

當要求加入節點 S 接收到 RREP-J 封包時，就可以得到這一條路徑的平均接收功率，並拿來當作決定使用哪幾條路徑回傳 SACT-J 封包的選擇流程內的重要參數。

$$\bar{P}_{H_j}(HC_{H_j}) = \frac{(HC_{H_j} - 1)\bar{P}_{H_j}(HC_{H_j} - 1) + \Pr_{N_j}}{HC_{H_j}} \quad 4-1$$

為了達成 4.1 節 PCHMR 演算法的設計考量（利用群組內的樹狀網路架構混和群組間的網狀網路架構產生混合式網路架構），在選擇利用哪些路徑回應 SACT-J 封包時，必須經過特殊的決定流程。首先 PCHMR 演算法考慮了路徑中所有連結的平均接收功率，若平均接收功率小於一個與預設的標準值，這條路徑是完全不受考量的。再來要確定有回應 RREQ-J 封包的混合式網路架構成員中，哪些跟要求加入節點屬於同一群組（檢查群組 ID）。若是屬於同一群組，由於必須建構樹狀網路架構，所以只能在符合之前條件的路徑中，選擇其中一條通往該混合式網路架構成員的路徑（當然這些路徑中所有連結的平均接收功率必須大於標準值），而選擇的方式就是考慮在所有從有回應 RREQ-J 封包的混合式網路架構成員到要求加入節點的路徑中，選擇跳躍數最小的路徑。也就是在符合同一群組、路徑連結平均功率大於標準值的路徑中，選擇跳躍數最小的路徑，當成是群組內樹狀網路架構所需要的路徑，並利用這條路徑傳遞 SACT-J 封包。此外，若那些回應的混合式網路架構成員跟要求加入節點屬於不同群組，就只要符合路徑連結平均功率大於標準值的條件，就會被選擇通往該混合式網路架構節點的路徑為網狀網路架構所需要的路徑，並使用這些路徑傳遞 SACT-J 封包。

利用圖 4-2 可以更完整地瞭解在從多條路徑接收到 RREP-J 封包之後，有哪

些條件在選擇路徑時被考慮。只有在這些條件都符合的路徑，才會被選擇回傳

SACT-J 封包：

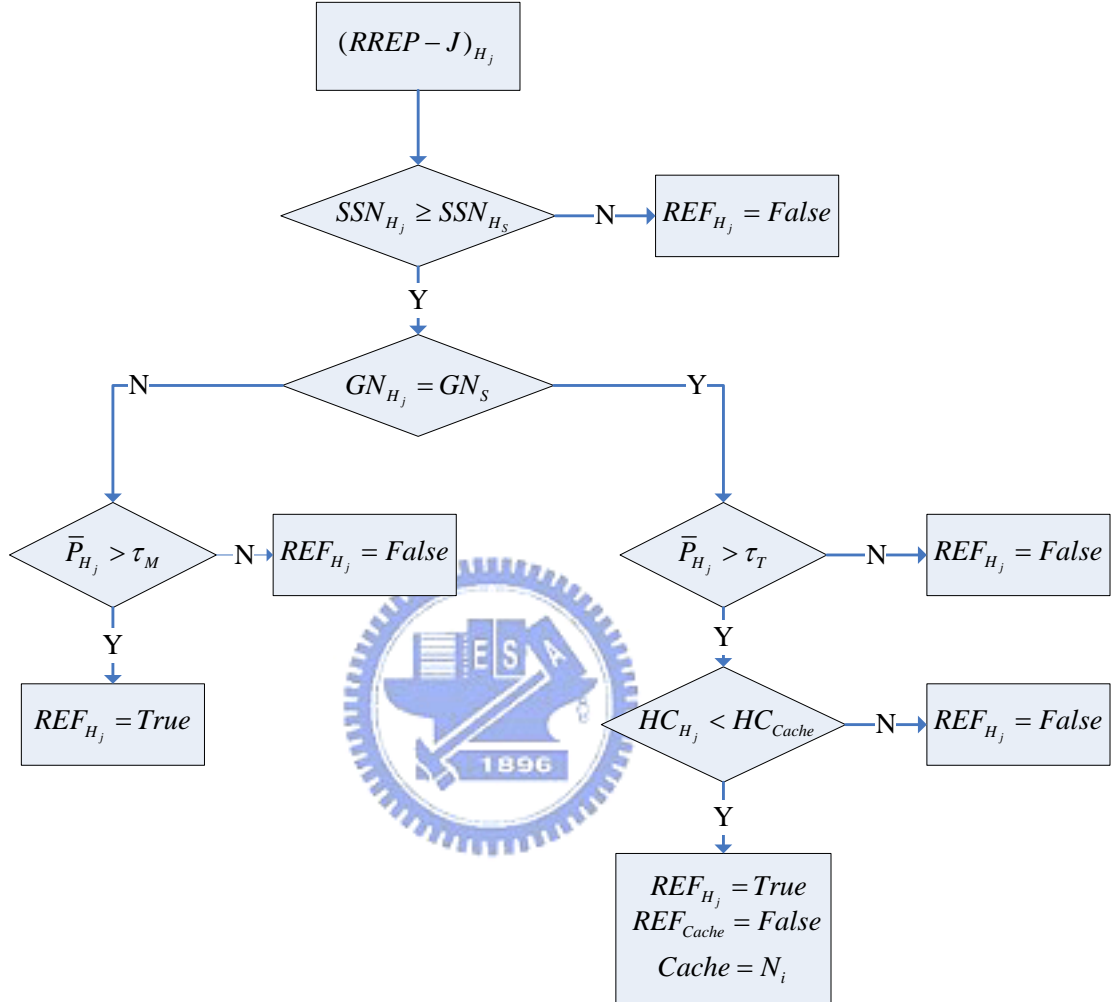


圖 4-2 選擇回傳 SACT-J 封包路徑的流程圖

- 1) 當要求加入節點接收到 $RREP-J_{H_j}$ 封包時，先藉由 $RREP-J_{H_j}$ 封包內的資訊判斷 H_j 的 SSN 是否大於或是等於行動運算節點的 SSN (i.e. $SSN_{H_j} \geq SSN_s$)。如果條件符合，代表 H_j 到混合式網路架構的路徑是比要求加入節點到混合式網路架構的舊路徑還要新(如果要求加入節點是第一次要加入，就會連舊路徑都沒有， SSN 為 0，也符合條件)，允許進入下一個步

驟。藉由 SSN 的檢查可以避免一些曾經屬於混合式網路架構，但現在卻已不是屬於混合式網路架構的行動運算節點也回應要求加入節點送出 RREQ-J 封包，造成要求加入節點使用這條路徑時會產生錯誤。如果條件不符合，為了避免錯誤，這條路徑將不會被使用，而在多重擇路表單內相對應於 H_j 的 REF_{H_j} 將會被設定成不驅動狀態。

- 2) 當要求加入節點確定 $SSN_{H_j} \geq SSN_S$ 時，接著判斷回應 RREP-J _{H_j} 封包的 H_j 跟要求加入節點是否屬於同一個群組(i.e. $GID_{H_j} = GID_S$)。如果要求加入節點跟 H_j 屬於同一個群組，將可能採用樹狀網路架構的方式產生路徑連結，所以接著會進入利用樹狀網路架構的條件判斷是否要使用這條路徑的步驟(圖 4-2 中右下角的部分)。如果條件不符合，就代表要求加入節點跟 H_j 屬於不同群組，將可能採用網狀網路架構的方式產生路徑連結，所以接著會進入利用網狀網路架構的條件判斷是否要使用這條路徑的步驟(圖 4-2 中左下角的部分)。
- 3) 如果要求加入節點跟 H_j 屬於同一個群組，接著判斷 RREP-J _{H_j} 封包的平均接收功率是否有大於樹狀平均接收功率標準值(i.e. $\bar{P}_{H_j} > \tau_r$)，如果符合條件，代表這條路徑上的所有連結可靠不易斷裂，可以被考慮為樹狀網路架構部分的路徑連結，進入下一個判斷跳躍數的步驟。如果條件不符合，代表這條路徑上的連結很不穩定且很可能斷裂，而在多重擇路表單內相對應於這條路徑的 REF_{H_j} 將會被設定成不驅動狀態。

- 4) 如果要求加入節點跟 H_j 不屬於同一個群組，接著判斷 RREP-J _{H_j} 封包的平均接收功率是否有大於樹狀平均接收功率標準值(i.e. $\bar{P}_{H_j} > \tau_M$)，如果符合條件，代表這條路徑上的連結可靠不易斷裂，可以直接被選擇為網狀網路架構部分可以使用的路徑，在多重擇路表單內相對應於這條路徑的 REF_{H_j} 將會被設定成驅動狀態。如果條件不符合，代表這條路徑上的連結很不穩定且很可能斷裂，而在多重擇路表單內相對應於這條路徑的 REF_{H_j} 將會被設定成不驅動狀態。
- 5) 如果要求加入節點跟 H_j 屬於同一個群組，而且這條路徑上連結可靠不易斷裂，接著判斷 RREP-J _{H_j} 封包的跳躍數是否小於暫存區內暫存路徑的跳躍數 ($HC_{H_j} < HC_{Cache}$)。如果條件符合，代表這一條路徑跟上一次被決定為樹狀網路架構路徑(針對同一次 RREQ-J 封包的要求，上一次符合前述幾點條件的路徑)一樣被視為可靠不易斷裂，但所需要的跳躍數比較少，在其他條件都符合的情況下，這一條路徑被視為更佳的樹狀網路架構路徑，可以被選擇為樹狀網路架構部分可以使用的路徑，在多重擇路表單內相對應於這條路徑的 REF_{H_j} 將會被設定成驅動狀態，並把這條路徑存在暫存區內的暫存路徑。而之前被選擇的路徑，在多重擇路表單內相對應於這條路徑的 REF_{H_j} 將會被設定成不驅動狀態。如果條件不符合，代表這條路徑的條件不會比暫存區內的暫存路徑還好，所以在多重擇路表單內相對應於這條路徑的 REF_{H_j} 將會被設定成不驅動狀態。

最後要求加入節點會檢查多重擇路表單內，那些路徑的 REF_{H_j} 被設定成驅動狀態，並利用單一擇路的方式，利用這些路徑回傳 SACT-J 封包。

以圖 4-1 為例，要求加入節點 S 收到三個經由不同路徑傳來的 RREP-J 封包 (圖 4-1 內，混合式網路架構成員 H_1 、 H_2 跟 H_3 回傳 RREP-J 封包的路徑)，發現 RREP-J 封包內的 SSN 都大於等於要求加入節點 S。其中 H_1 跟 H_2 都與要求加入節點 S 屬於同一個群組，且 \bar{P}_{H_1} 跟 \bar{P}_{H_2} 都大於 τ_T ，但由於 RREP-J _{H_1} 封包的跳躍數大於 RREP-J _{H_2} 封包的跳躍數 ($HC_{H_1} > HC_{H_2}$)，所以要求加入節點 S 選擇通往 H_2 的路徑 L_6 回傳 SACT-J 封包，以樹狀網路架構的概念跟混合式網路架構產生連結。至於 H_3 跟要求加入節點 S 不屬於同一個群組，而且 \bar{P}_{H_3} 大於 τ_M ，所以要求加入節點 S 使用通往 H_3 的路徑 L_7 回傳 SACT-J 封包，以網狀網路架構的概念跟混合式網路架構產生連結。



4.2.3 混合式網路架構的維持機制

混合式網路架構的維持利用二種廣播維持封包的機制：

- 3) 網路架構領導節點 (Structure Leader) 會負責週期性的廣播架構維持訊息 (Structure-Maintain Message) 封包，藉此來更新一些有關整個混合式網路架構的資訊 (例如混合式網路架構的 SSN)。
- 4) 為了確定網路內任意二點間的連結是否斷裂，混合式網路架構內的所有成員都會週期性的廣播一個跳躍數的維持訊息 (Maintain Message) 封

包，讓連結的另一邊知道這個連結依舊存在。如果混合式網路架構中的

某一點在一段時間內都沒有收到連結另外一邊送來的維持訊息封包，那

此行動運算節點就會認定這個連結斷裂，並針對這種情況做出反應。

當群組之間不斷移動時，週期性的廣播維持訊息封包也可以被利用來增加群組之間的連結性。當二個群組接近時，其中一個群組內的某一個混合式網路架構成員藉由維持訊息封包發現自己已經接近另外一個群組，且封包的接收功率也大於 τ_M ，此時這個混合式網路架構成員就會把廣播維持訊息封包的節點視為另一條通往混合式網路架構的路徑，在多重擇路表單儲存這一條路徑的資訊，並把這條路徑的 *REF* 被設定成驅動狀態。

在圖 4-1 中，假設網路架構成員 *S* 接收到來自 H_4 的維持訊息封包，發現屬於不同群組，而且封包的接收功率大於 τ_M 。此時網路架構成員 *S* 就瞭解自己已經跟另一個群組接近，可以增加網狀網路架構概念的路徑來增強群組間的連結性。所以在自己的多重擇路表單內增加這一條路徑 L_8 ，並把這條路徑的 *REF* 被設定成驅動狀態。

第五章 模擬環境

為了證明上一章所敘述的演算法對於封包的傳輸有實質上的效能提升，在這篇論文的最後將進行演算法的模擬並與其他類似的演算法做比較。而在這一章將會先介紹模擬平台、使用者移動模型與無線傳輸模型等模擬環境參數的考量與設計。

5.1 模擬平台 – 網路架構模擬器

網路架構模擬器(Network Simulator – NS2)[39]是一種離散式事件的模擬器，提供了從通道到實體層再到媒體擷取控制層、網路通訊層與傳輸層(Transport Layer)的網路架構模擬。將自行發展出來的新擇路演算法實現，再加上網路架構模擬器原本就有提供的其他網路架構，就可以進行新擇路演算法的模擬。

在整個網路中，每次封包的傳送就代表網路架構模擬器的一次事件。當所有網路架構、節點移動模型與通道環境等要素都建構完成後，網路架構模擬器就會依據這些指定過的要素，逐一產生封包事件，並詳細記錄每個封包事件在傳輸中的過程(包括傳送、接收等)。接著利用其他的工具分析整個封包事件傳輸的紀錄，就可以瞭解新擇路演算法在實質上的效能提升程度。

儘管網路架構模擬器在實體層的模擬相當的簡略，但這次的模擬是著重在網路通訊層的擇路演算法，加上其他階層的模擬也都已經很接近真實的演算法，所

以藉由網路架構模擬器，已經可以相當逼真的模擬整個網路架構的系統，是一個相當方便使用的模擬平台。

5.2 使用者移動模型

為了更貼切真實環境的情況，針對不同的網路環境，在設計擇路演算法時，會有不同的考量因素。這種依據特殊的考量因素而設計的擇路演算法，在模擬時，需要不同的移動模型來配合。針對擇路演算法的特性設計特殊的移動模型來配合模擬與分析，將會讓擇路演算法的特性更加的突顯出來，更能讓觀察者瞭解這個擇路演算法是否真的有解決當初設計時想解決的問題。所以徹底的了解擇路演算法的特性並設計適合的移動模型是很重要的。有些研究就是專門研究移動模型對模擬的重要性。



移動模型可以分成二種，一種是藉由長期的觀察，記錄使用者在實際生活上的移動方式，並進行統計與分析，找出適當的移動模型。但這種方式相當耗時，並且需要不同領域的分析方法，所以暫時先不考慮利用這種方式產生移動模型。另一種是利用數學上的理論來決定移動的方式，再利用程式實現這個數學理論，進而實現設計的移動模型。當然利用這種方式產生的移動模型可能跟實際上的情況有所誤差，但已經足以突顯或分析出擇路演算法之間的差異，所以採用這種方式產生模擬需要的移動模型。

經由數學理論產生的移動模型[40]可以分成個體隨機移動模型跟群組移動

模型。個體移動模型是比較基本的移動模型，而群組移動模型則是把個體移動模型再加上群組移動的觀念。

個體移動模型最常使用的模型便是隨機移動模型(Random Walk Model)[40]。當模擬其中一個行動運算節點時，一開始在整個模擬範圍之內，隨機的決定一個初位置。接著隨機決定一個方向跟速度，在一個隨機決定的結束時間內移動。當隨機決定的時間結束時，行動運算節點就會再決定下一個方向跟速度，接著再隨機決定下一個結束的時間，重複步驟直到模擬時間結束。決定方向的方式有二種，一種是在模擬範圍之內決定一個目標位置，依照隨機決定的速度，往目標位置的方向移動一段時間。另一種是決定一個移動的角度，依照隨機決定的速度，沿著該角度的方向移動一段時間。另外，為了使移動模型更加的貼近實際情況，有幾種方式可以運用。一、加上停頓時間(Pause Time)：在節點到達定點時，不會馬上決定一個方向跟角度，而是先停頓一段隨機決定的時間。二、控制角度的大小：在決定移動方向時，讓比較大的角度出現的機會比較小，可以模擬實際上移動時，較少有突然折回的現象。三、移動到模擬範圍邊緣的處理方式：因為模擬的範圍有限，所以當行動運算節點跑到模擬環境的邊緣時，要特別的處理。

在實際生活上，常會發生個體之間不是隨機移動的情況，而是有組織性的移動，例如學生排隊與參加升旗，或是軍隊整體移動。所以在個體移動模型的基礎之下，可以發展出群組移動模型，配合實際生活上會遇到的情況。群組移動模型

最常看到的就是參考點群組移動模型 (Reference Point Group Mobility Model)[40]。每一個小群組都設定一個群組參考點，可以決定整個群組移動的方向。群組內的行動運算節點會以群組參考點為中心，用個體移動模型的方式移動。利用這樣的方式就可以表現出比較接近實際的群組移動模型(見圖 5-1)。

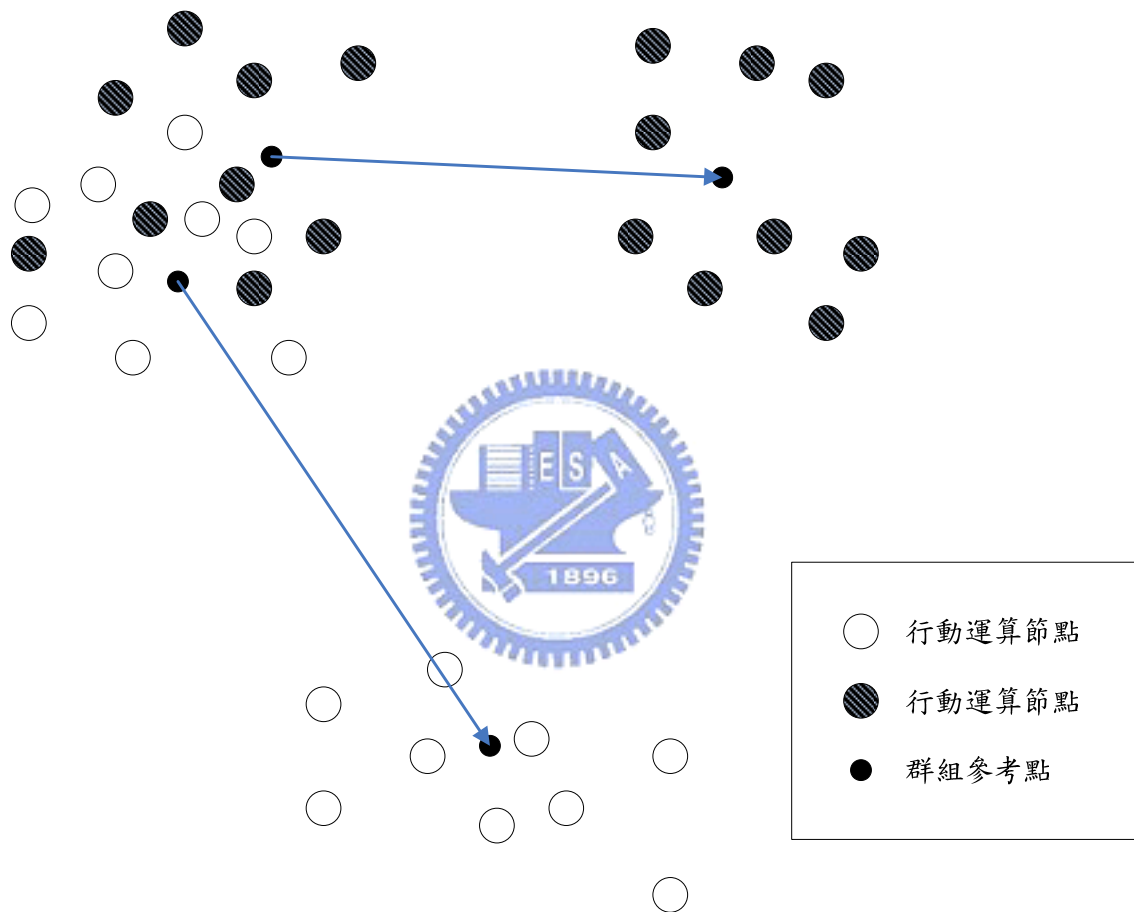


圖 5-1 群組移動模型示意圖

由於 PCHMR 演算法是針對群組移動模型而設計的，因此在之後的模擬將採用參考點群組移動模型，讓 PCHMR 演算法在群組移動的環境下，凸顯出 PCHMR 演算法的設計特性。

5.3 無線傳播模型

在模擬通道環境的部分，直接採用網路架構模擬器提供的遮蔽傳播模型[39]。為了更加瞭解遮蔽傳播模型的概念，在介紹遮蔽傳播模型之前，會先介紹比較基本的自由空間傳播模型跟二路合成射線傳播模型。

5.3.1 自由空間模型

假設在理想的傳播環境內，介於傳送端和接收端之間只有一條清楚的直線可達(Line-of-Sight - LOS)的傳播路徑。假設 P_t 為傳送端的傳送訊號功率， G_t 和 G_r 分別為傳送端和接收端的天線增益， $L(L \geq 1)$ 為系統損耗而 λ 為波長，在距離傳送端長度 d 的位置可以接收到的接收訊號功率為 $P_r(d)$ ， $P_r(d)$ 的大小可以利用算式 5-1 算出。通常在模擬中，會設定 $G_t = G_r = 1$ 和 $L = 1$ 。


$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L} \quad 5-1$$

5.3.2 二路徑射線合成傳播模型

但在一般情況下，傳送端和接收端會貼近在地面上移動，所以不太可能會處在一個類似自由空間模型的環境中。因此在傳送端和接收端之間，不太可能只有一條直接可達的傳播路徑。二路徑射線合成傳播模型除了考慮直接可達的傳播路徑之外，還有一條地面反射路徑，這種模型在模擬長距離的傳輸時，會比自由空間模型更準確。假設 P_t 為傳送端的傳送訊號功率， G_t 和 G_r 分別為傳送端和接收

端的天線增益， h_t 和 h_r 分別是傳送天線和接收天線的高度， $L(L \geq 1)$ 為系統損耗

而 λ 為波長，在距離傳送端長度 d 的位置可以接收到的接收訊號功率為 $P_r(d)$ ，

$P_r(d)$ 的大小可以利用算式 5-2 算出。

$$P_r(d) = \frac{P_t G_t G_r h_t^2 h_r^2}{d^4 L} \quad 5-2$$

5.3.3 遮蔽模型

真實的傳輸環境中，在多重路徑傳播等因素的影響下，接收端的接收功率不再只是與傳送端距離的函數(假設 P_t 、 G_t 、 G_r 、 L 、 λ 等參數皆為定值)，還需要再加入許多其他的參數，甚至含有隨機的部分。因此在距離傳送端固定距離的接收功率不再只是一個定值，而是一個可分析的隨機變數。在一些相關的研究中，推導出一種比較簡單卻又滿接近真實情況的遮蔽傳播模型。遮蔽傳播模型考慮了二個參數來模擬在真實的傳輸環境中，功率大小隨著距離的變化方式。

$$\left[\frac{P_r(d)}{P_r(d_0)} \right]_{dB} = -10\beta \log\left(\frac{d}{d_0}\right) \quad 5-3$$

遮蔽傳播模型考量的第一個參數是針對路徑耗損(Path Loss)。假設 $P_r(d_0)$ 是由自由空間模型得到(d_0 代表很接近傳送端的距離長度，用來當作距離 d 的參考距離)， β 則為路徑耗損指數(需藉由場效的測量才能得知)。在距離傳送端長度 d 的位置可以接收到的接收訊號功率為 $P_r(d)$ ， $P_r(d)$ 的大小可以利用算式 5-3 算出。當遮蔽傳播模型的 $\beta=2$ 時， $P_r(d)$ 就會跟利用自由空間模型算出的大小相

等。在一般情況， β 會大於 2，通道的環境越差， β 就會越大，也代表路徑耗損越大。表 5-1 列出一些路徑耗損指數 β 的參考值。

環境		β
室外環境	自由空間	2
	有建築物遮蔽的城市	2.7~5
室內環境	射線直接到達處	1.6~1.8
	射線不能直接到達處	4~6

表 5-1 路徑耗損指數 β 的參考值

遮蔽傳播模型考量的第二個參數是針對在相同距離且相同路徑耗損指數的情況下，接收功率一樣會有隨機的變動，這個隨機的變動可以用一個高斯隨機變數(Gaussian)來代表。在考量第二個參數之後，遮蔽傳播模型可以由算式 5-4 計算。

$$\left[\frac{P_r(d)}{P_r(d_0)} \right]_{dB} = -10\beta \log\left(\frac{d}{d_0}\right) + X_{dB} \quad 5-4$$

在算式 5-4 中， X 是一個平均值為 0 且標準差為 σ_{dB} 的高斯隨機變數。這裡的 σ_{dB} 稱為遮蔽標準差(Shadowing Deviation)，需藉由場效的測量才能得知。表 5-2 列出一些遮蔽標準差 σ_{dB} 的參考值。

環境	σ_{dB}
戶外環境	2
辦公室環境	2.7~5

表 5-2 遮蔽標準差 σ_{dB} 的參考值

遮蔽傳播模型考慮了路徑耗損指數 β 與遮蔽標準差 σ_{dB} ，使藉由這種方式模擬的通道環境會更加的符合現實情況，因此在這篇論文使用遮蔽傳播模型模擬無線的通道環境。



第六章 效能評估與分析

6.1 能量分散擇路演算法的效能表現

由於 DER 演算法是以 AODV 演算法為基礎進行改良的，所以在這一節將會進行 DER 演算法與 AODV 演算法的效能比較。模擬會在行動運算節點以不同速度做移動的情況下，觀察資料封包送達率(Delivery Ratio of Data Packet)、資料封包平均延遲(Average Delay of Data Packet)、平均節點生存時間(Average Node Life Time)與平均 RREQ 封包傳送次數(Average Number of Sending RREQ)的效能表現。



模擬環境參數的設定

如上一章所述，模擬將使用網路架構模擬器(2.26 版)，通道的無線傳播環境使用遮蔽傳播模型，而移動模式如圖 6-1 所示。為了凸顯 DER 演算法，在圖 6-1 內，假設部分行動運算節點的電池能量較少，且來源節點 S1、S2、S3 與 S4 分別傳送資料封包給目標節點 D1、D2、D3 與 D4。每個行動運算節點都會在限制的小框框內(300 公尺乘以 300 公尺)，以隨機移動模型(Random Walk Model)的方式移動，速度移動分別是每秒 2 公尺、4 公尺、6 公尺、8 公尺跟 10 公尺。每個傳送節點是使用固定位元傳送率(Constant Bit Rate - CBR)協定傳送資料封包，而封包位元的傳送率設為 256Kbps，每 0.5 秒傳送一次封包。遮蔽傳播模型的參數

分別是： $\beta = 2.0$ 、 $P_t = 10dBm$ 。

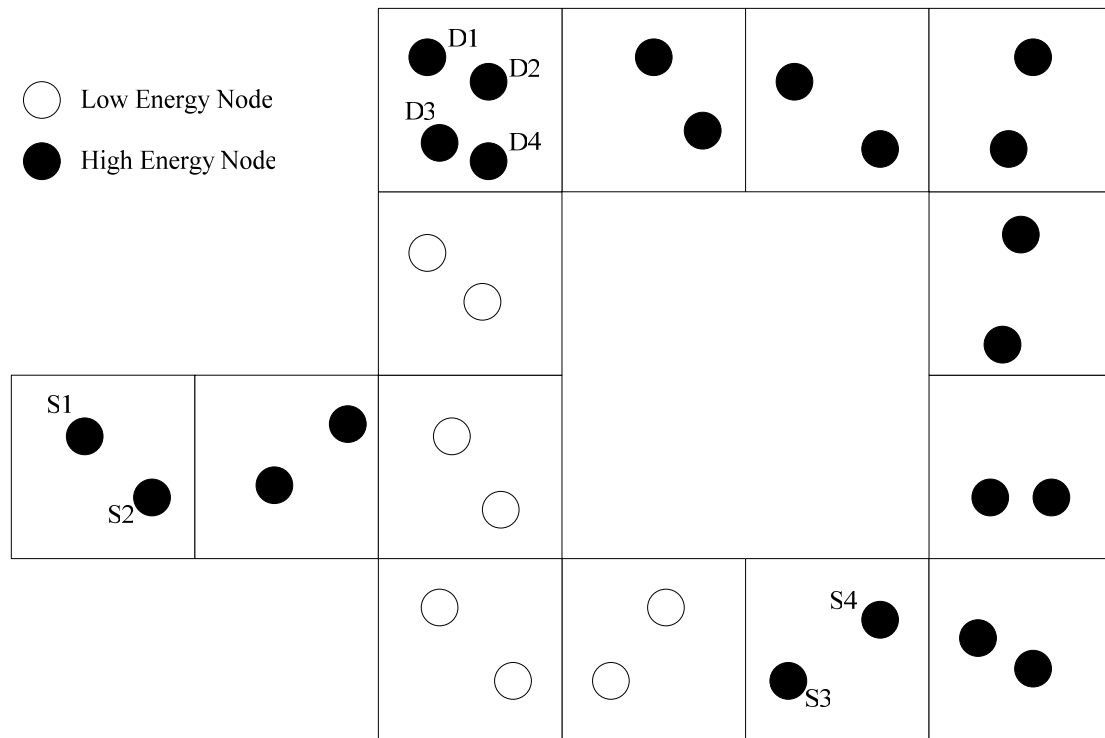


圖 6-1 模擬 DER 演算法之移動模型

由圖 6-1 的節點分佈可以大概的推測，假如現在使用 AODV 演算法，那因為電池能量較少的節點剛好處在網路的重要位置，因此這些節點會很容易被指定為擇路節點。假如現在使用的是 DER 演算法，除了 S1 跟 S2 必須選擇電池能量較少的節點當作中繼節點以外，S3 與 S4 會避開這些電池能量較少的節點，而選擇跳躍數較多、路徑剩餘電池能量最小值較大的其他中繼節點當作是擇路節點。所以在 AODV 演算法的運作下，電池能量較低的節點將會很容易的耗盡能量，造成 S1 與 S2 的資料封包不能順利傳遞。DER 演算法就比較不會有這種問題發生，資料封包可以順利的傳遞。

模擬結果與討論

由圖 6-2 的觀察可以發現跟預期的一樣，由於 DER 演算法把電池能量的耗損平均的分佈到二個方向的路徑，所以電池能量較低的擇路節點有比較少的負擔。雖然犧牲了資料封包的延遲時間(見圖 6-3)，但卻可以讓所有的擇路節點都維持比較久的生存時間(見圖 6-4)，所以可以達到比較高的資料封包送達率。隨著速度的增加，因為路徑更容易斷裂，所以資料封包送達率的下降是可以預期的。但 DER 演算法的效能依舊可以高於 AODV 演算法。

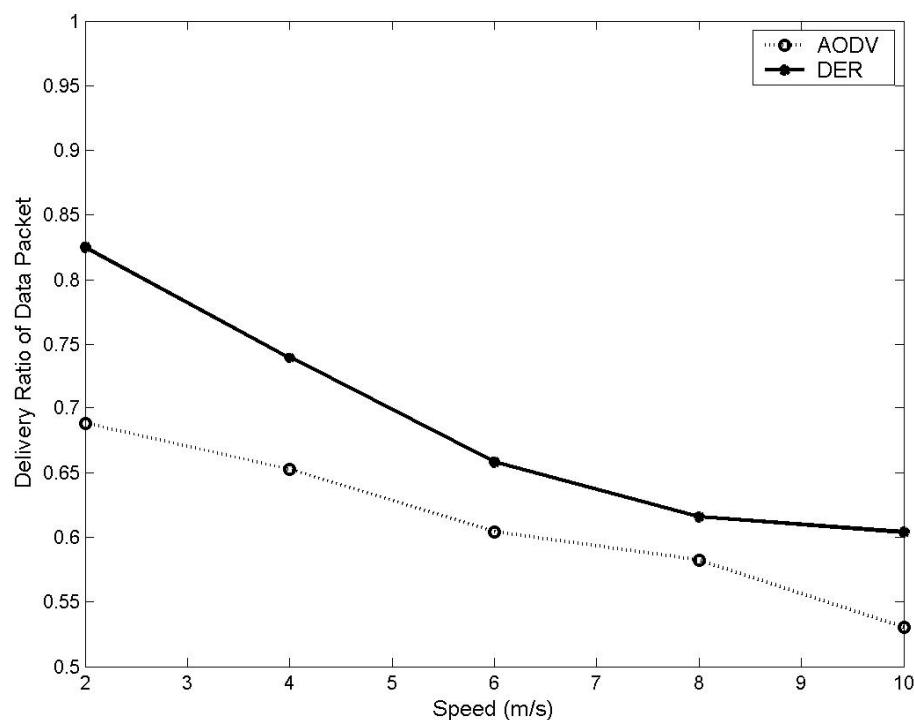


圖 6-2 資料封包送達率

觀察圖 6-3，DER 演算法為了跳過剩餘電池能量較低的中繼節點，選擇跳躍數比較多的路徑，因此資料封包的延遲時間也相對的增加。隨著速度的增加，AODV 演算法跟 DER 演算法都因為路徑越來越容易斷裂而必須花時間重建，所

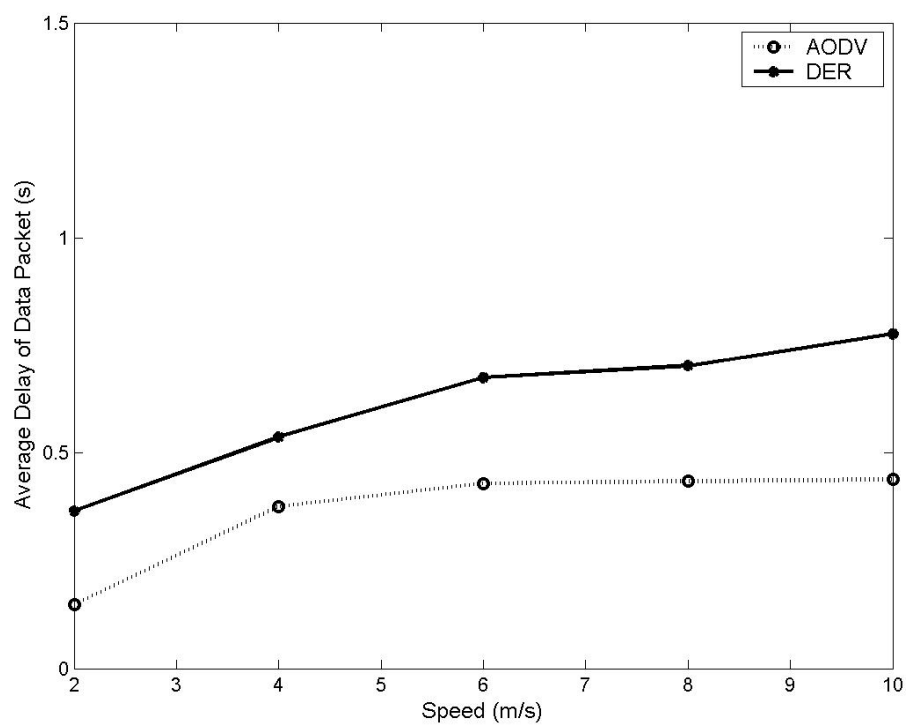


圖 6-3 資料封包平均延遲

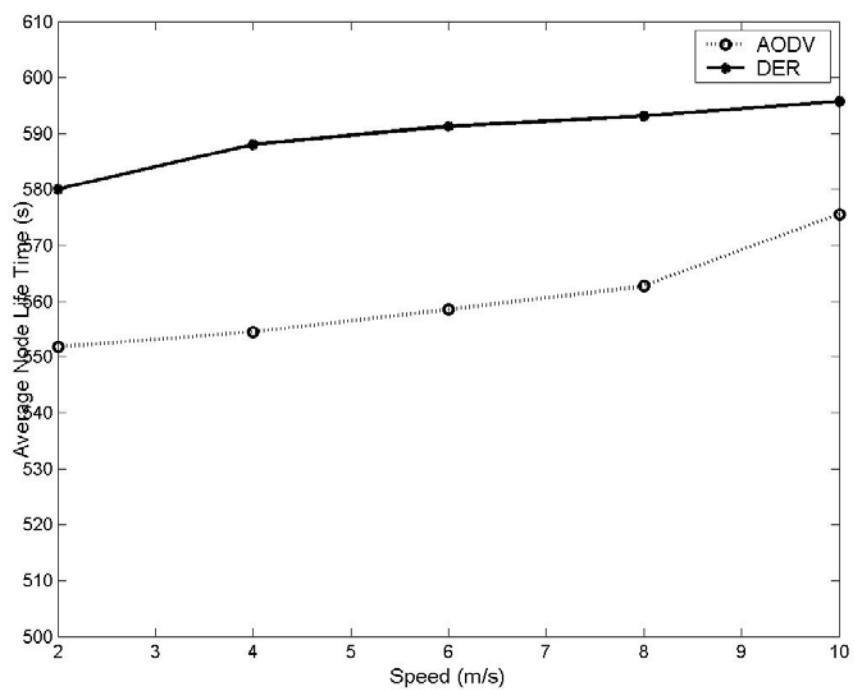


圖 6-4 平均節點生存時間

以整體的延遲時間都會增加。

觀察圖 6-4，由於 DER 演算法把電池能量的耗損平均地分佈在不同的路徑上，因此電池能量較低的節點可以有比較長的生存時間，也讓整體的生存時間增加。隨著速度的增加，不論是 DER 演算法或是 AODV 演算法，因為路徑容易斷裂造成可以傳送的資料封包較少，電池能量的耗損也會減少，使平均生存時間稍微增加。

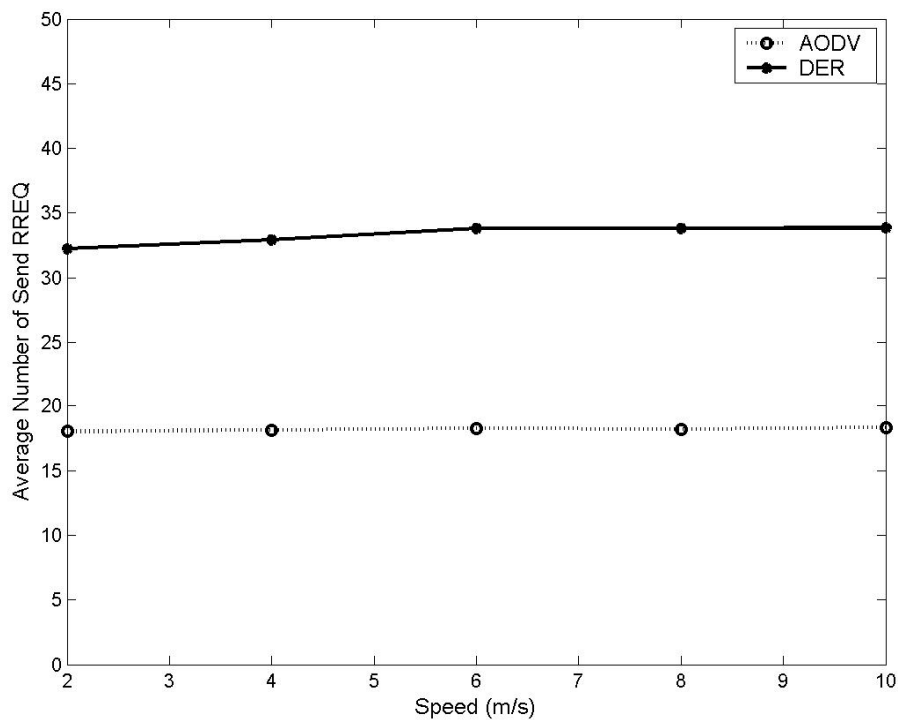


圖 6-5 平均 RREQ 封包傳送次數

當來源節點廣播一次 RREQ 封包時，觀察整個網路內一共會轉傳這個 RREQ 封包的次數。由於 DER 演算法會儲存所有通往來源節點的路徑，而在圖 6-1 可以發現大概有二條路徑可以選擇。又因為 DER 演算法增加了一些降低控制封包的機制，因此可以推論 DER 演算法造成的 RREQ 封包數量應該是 AODV 演算法

的二倍再少一點，相當符合圖 6-5 的結果。

6.2 功率控制混合式多重擇路演算法的效能表現

由於 PCHMR 演算法是以 MAODV 演算法為基礎而設計出來的，因此模擬出來的結果將會跟 MAODV 演算法做比較，來判斷是否有效能上的進步。為了觀察 PCHMR 演算法考慮平均接收功率所造成的影響，會模擬 PCHMR 演算法在沒有使用平均接收功率下的效能表現。因為只考慮跳躍數，故稱為 PCHMR-H 演算法，而有考慮平均接收功率的 PCHMR 演算法就稱為 PCHMR-HP 演算法。模擬會在行動運算節點以不同速度做移動的情況下，觀察資料封包送達率(Delivery Ratio of Data Packet)、資料封包平均延遲(Average Delay of Data Packet)與控制封包與資料封包比(Ratio of Control Packet to Data Packet)的效能表現。

模擬環境參數的設定

這一小節的模擬一樣使用網路架構模擬器(2.26 版)，通道的無線傳播環境會使用遮蔽傳播模型，而移動模型就使用參考點群組移動模型。

在模擬中，假設合成網路架構內一共有 48 個行動運算節點在 1000 公尺乘上 1000 公尺的範圍內，分別以每秒 0 公尺、10 公尺跟 20 公尺的速度移動。所有行動運算節點分成 3 個群組，每個群組內有 16 個行動運算節點。在每個群組內，有 2 個行動運算節點(傳送節點)會提出傳送資料封包的要求，另外有 8 個行動節

點(接收節點)會想要接收合成網路架構內的資料封包。要注意資料封包的傳輸是跨群組的，因此一共是 6 個傳送節點提出傳送資料封包的要求，每一個送出的資料封包會有 24 個接收節點想要接收。每個傳送節點是使用固定位元傳送率 (Constant Bit Rate - CBR) 協定傳送資料封包，而封包位元的傳送率設為 256Kbps。遮蔽傳播模型的參數分別是： $\beta = 2.0$ 、 $P_t = 25dBm$ 、 $G_t = 1.2$ 、 $G_r = 1.2$ 、 $L = 1.0$ 、 $\lambda = 0.125$ 跟 $\sigma_{dB} = 4.0$ 。

模擬結果與討論

由圖 6-6 到圖 6-8 的觀察可以發現整體的效能表現都是 PCHMR-HP 演算法優於 PCHMR-H 演算法，再優於 MAODV 演算法。

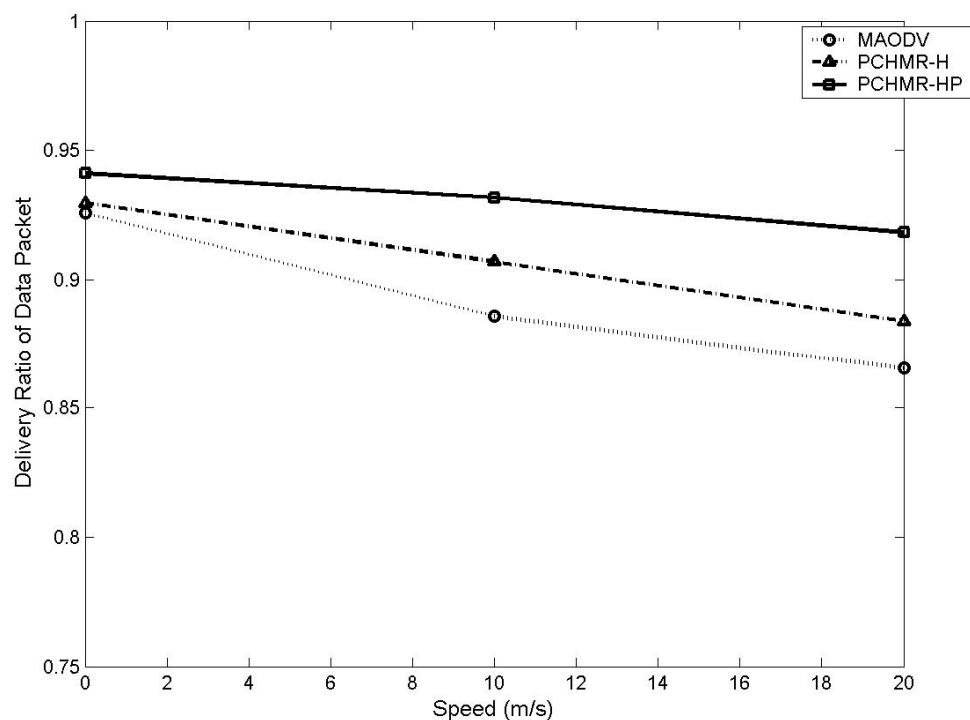


圖 6-6 資料封包送達率

在圖 6-1 內，因為 PCHMR-H 演算法與 PCHMR-HP 演算法在群組間使用了網狀網路架構的方式，當群組間移動較動態的情況下，群組間的連結不容易完全斷裂，因此可以確保資料封包有比較高的機會可以繼續傳送。至於網路架構比較脆弱的 MAODV 演算法就容易因為群組間的動態移動，造成群組間的資料封包傳送比較容易失敗。PCHMR-HP 演算法因為考量到了平均接收功率，因此選擇到的路徑比只考慮跳躍點次數的 PCHMR-H 演算法來的牢固不易斷裂，因此資料封包的傳送也更確定可以完成。

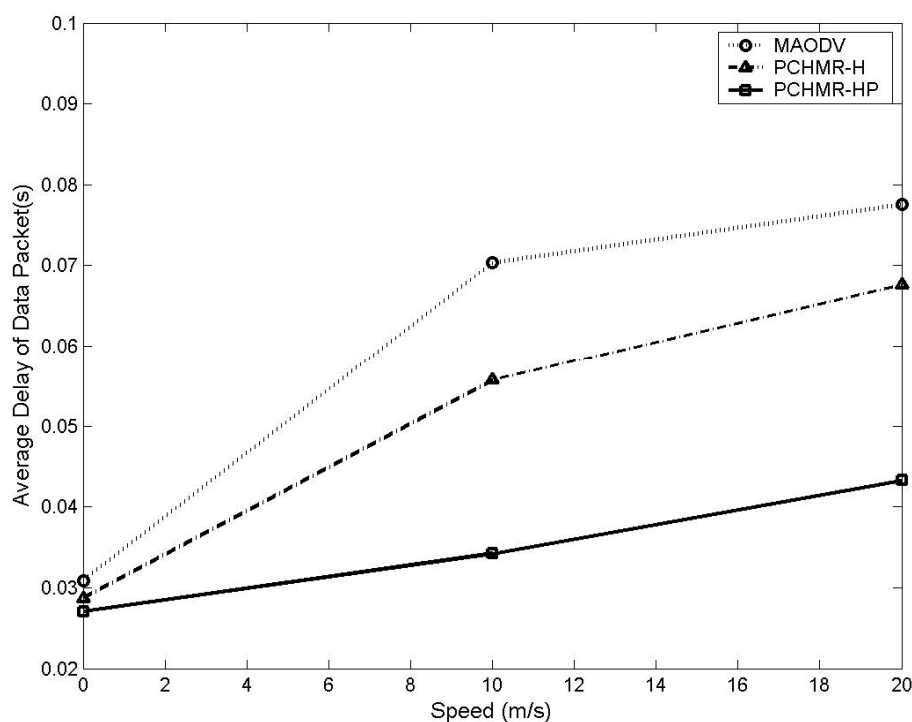


圖 6-7 資料封包平均延遲

在圖 6-7 可以發現到 MAODV 演算法的資料封包傳送會需要比較長的延遲時間，那是因為 MAODV 演算法群組間的路徑比較容易斷裂，這時候就需要浪費相當多的時間做路徑重建，此時資料封包就必須被暫存起來，造成延遲時間的

增加。而 PCHMR-HP 演算法與 PCHMR-H 演算法因為群組間可以使用的路徑較多，因此當部分路徑斷裂時，其他的路徑依舊可以傳遞資料封包，不會造成過大的延遲。因為 PCHMR-HP 演算法的路徑多考量了平均接收功率的因素所以狀況比較差的路徑會不被採用，因此使用到的擇路節點會比 PCHMR-H 演算法少。當資料封包在傳輸時，網路比較不會這麼的壅塞，造成延遲的狀況就不會這麼嚴重。另外在群組內的樹狀網路架構，PCHMR-HP 演算法也是有考慮到平均接收功率的因素，所選擇的路徑會比 PCHMR-HP 演算法或 MAODV 演算法來的牢固不易斷裂，因此較短的延遲時間是合理的。

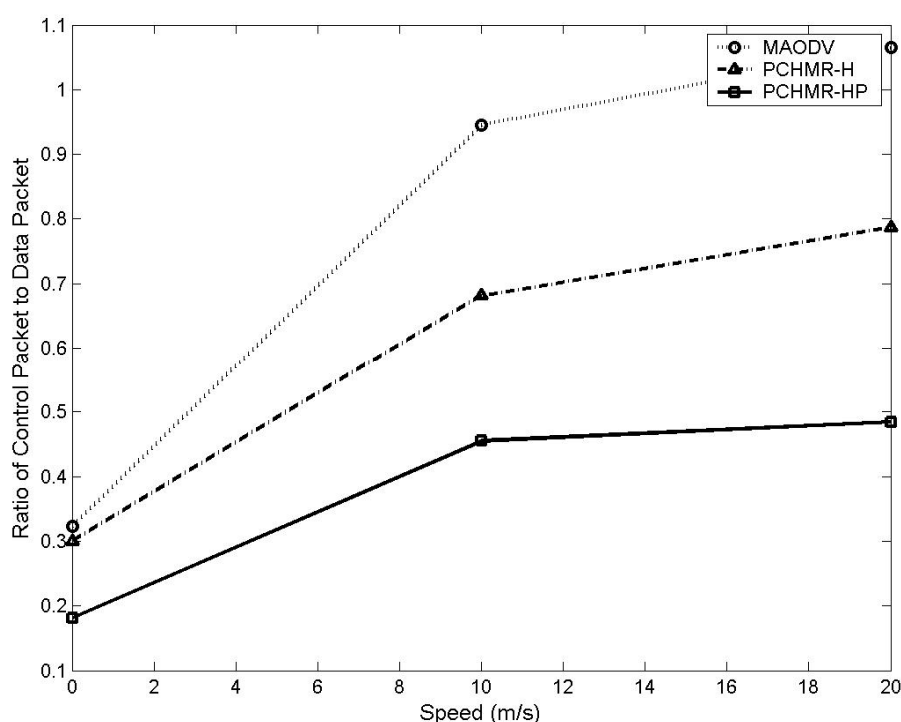


圖 6-8 控制封包與資料封包比

觀察圖 6-8 之後，可以意外的發現，路徑較少的 MAODV 演算法居然有比較高的控制封包與資料封包比。其實原因很簡單。首先，因為 PCHMR-HP 演算

法與 PCHMR-H 演算法的路徑較多，資料封包的到達率也較高，因此可以降低控制封包與資料封包比。再來是因為 MAODV 演算法的路徑比較容易斷裂，因此需要更多的控制封包做路徑重建的工作，且在路徑重建的同時，如果暫存資料封包的緩衝區不夠用，這時候也會造成資料封包的損失，更多的控制封包與更少的資料封包，造成控制封包與資料封包比的上升。因為 PCHMR-HP 演算法的路徑較牢固，而且不包含狀況比較差的路徑，因此 PCHMR-H 演算法的效能表現只能介於 MOADV 演算法與 PCHMR-HP 演算法之間。



第七章 結論

在這篇論文之中，考慮剩餘電池能量的因素，設計了 DER 演算法，讓電池能量的消耗可以平均地分佈在網路中。藉由這種方式，讓網路中的行動運算節點可以維持比較長的壽命，讓網路中的擇路節點都可以正常的運作。在觀察模擬結果之後，儘管 DER 演算法必須犧牲一點資料封包的延遲時間，但可以提供較高的封包傳遞成功率，但卻有 RREQ 封包增加許多的問題。另外還考量群組移動模式與平均接收功率的因素，發展出 PCHMR 演算法，增加一定數量的路徑幫忙傳輸，提高資料封包在傳送時的可靠性。由模擬結果可以發現，PCHMR 演算法可以提高資料封包的傳送成功率，卻反而不會產生更多的控制封包，使整個網路的效能都提升。



參考文獻

[1] Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications

[2] Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz Band High-speed

[3] Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: High-speed Physical Layer in the 5 GHz Band

[4] Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 4: Further Higher Data Rate Extension in the 2.4 GHz Band



[5] C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination Sequence Distance Vector (DSDV) Routing for Mobile Computers," *Proceedings of the ACM SIGCOMM '94 Conference*, Aug. 1994, pp. 234-244.

[6] S. Murthy and J. J. Garcia-Luna-Aceves, "An Efficient Routing Protocol for Wireless Networks," *ACM Mobile Networks Appl. J., Special Issue on Routing in Mobile Communication Networks*, Oct. 1996, pp. 183-197.

[7] C. E. Perkins and E. M. Royer, "Ad-hoc On-demand Distance Vector Routing," *Proceedings of the Second IEEE Workshop on Mobile Computing Systems and Applications*, Feb. 1999, pp. 90-100.

[8] D. B. Johnson, D. A. Maltz, and J. Broch, "DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks," *Ad Hoc Networking*, edited by C. E. Perkins, Addison-Wesley, 2001.

[9] V. D. Park and M. S. Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks," *Proceedings of IEEE Infocom '97*, Apr. 1997, pp. 1405-1413.

[10] V. D. Park and M. S. Corson, "Temporally-Ordered Routing Algorithm (TORA) version 1: Functional specification" Internet-Draft, draft-ietf-rnanet-tora-spec-00.txt, Nov. 1997.

[11] C.-K. Toh, "A novel Distributed Routing Protocol to Support Ad-hoc Mobile Computing", *Proceedings of 15th IEEE Annual International Phoenix Conference on Computers and Communications*, Mar. 1996, pp. 480-486.

[12] R. Dube, C. D. Rais, K. Y. Wang, and S. K. Tripathi, "Signal Stability Based Adaptive Routing (SSA) for Ad-hoc Mobile Networks," *IEEE Personal Communications*, Feb. 1997, pp. 36-45.

[13] Z. J. Haas and M. R. Pearlman, "The Performance of Query Control Schemes for the Zone Routing Protocol," *IEEE/ACM Trans. Networking*, vol.9, no.4, Aug. 2001, pp. 427-438.

[14] P. Samar, M. R. Pearlman, and Z. J. Haas, "Hybrid Routing: The Pursuit of an Adaptable and Scalable Routing Framework for Ad Hoc Networks," in *The Handbook of Ad Hoc Wireless Networks*, Boca Raton, FL: CRC Press, 2003.

[15] C. C. Chiang, H. K. Wu, W. Liu, and M. Gerla, "Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel," *Proceedings of IEEE SICON '97*, Apr. 1997, pp. 197-211.

- [16] A. Iwata, C. C. Chiang, G. Pei, M. Gerla, and T. W. Chen, "Scalable Routing Strategies for Ad Hoc Wireless Networks," *IEEE Journal on Selected Areas in Communication*, vol.17, no.8, Aug. 1999, pp. 1369-1379.
- [17] R. Sivahumar, P. Sinha, and V. Bharghavan, "CEDAR: a Core-Extraction Distributed Ad hoc Routing Algorithm," *IEEE Journal on Selected Areas in Communication*, vol.17, no.8, Aug. 1999, pp. 1454-1465.
- [18] J. Borch, D. Maltz, D. Johnson, Y. Hu, and J. Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols," *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, Oct. 1998, pp. 25-30.
- [19] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark, "Routing Protocols for Mobile Ad-Hoc Networks - A Comparative Performance Analysis," *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, Aug. 1999, pp. 195-206.
- [20] E. R. Royer and C. K. Toh, "A Review of Current Routing Protocols for Ah-hoc Mobile Wireless Networks," *IEEE Personal Communication Magazine*, Apr. 1999.
- [21] S. R. Das, R. Castaneda, J. Yan, and R. Sengupta, "Comparative Performance Evaluation of Routing Protocols for Mobile, Ad hoc Networks," *Proceedings of the International Conference on Computer Communications and Networks (ICCCN)*, 1998, pp. 153-161.
- [22] B. Parkinson and S. Gilbert, "NAVSTAR: Global Positioning System - Ten Years Later," *Proceedings of IEEE*, Oct. 1983, pp. 1177-1186.
- [23] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward, "A Distance Routing Effect Algorithm for Mobility (DREAM)," *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, October 1998, pp.76-84.

[24] Y. B. Ko and N. H. Vaidya, "Location-Aided Routing (LAR) in Mobile Ad Hoc Networks," *ACM Wireless Networks Journal*, vol.6, no.4, 2000, pp. 307-321.

[25] R. Jain, A. Puri, and R. Sengupta, "Geographical Routing using Partial Information for Wireless Ad Hoc Networks" *IEEE Personal Communications*, vol.8, no.1, 2001, pp. 48-57.

[26] B. Karp and H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, Aug. 2000, pp. 243-254

[27] J. Li, J. Jannotti, D. S. J. De Couto, D. R. Karger, and R. Morris, "A Scalable Location Service for Geographic Ad Hoc Routing", *Proceedings of the ACM/IEEE MobiCom*, Aug. 2000, pp. 120-130.

[28] R. Morris, J. Jannotti, F. Kaashoek, J. Li, and D. S. J. De Couto, "Carnet: A Scalable Ad Hoc Wireless Network System," *Proceedings of the 9th ACM SIGOPS European Workshop: Beyond the PC: New Challenges for the Operating System*, Sep. 2000.

[29] E. M. Royer and C. E. Perkins, "Multicast Operation of the Ad-Hoc On-Demand Distance Vector Routing Protocol," *Proceedings of the Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, Aug. 1999, pp. 207-218.

[30] C. W. Wu and Y. C. Tay, "AMRIS: A Multicast Protocol for Ad Hoc Wireless Networks," *Proceedings of the IEEE Military Communications Conference*, vol.1, Oct. -Nov. 1999, pp. 25-29.

[31] J. Xie, R. R. Talpade, A. Mcauley, and M. Liu, "AMRoute: Ad Hoc Multicast Routing Protocol," *Mobile Networks and Applications*, vol.7, iss.6, Dec. 2002, pp. 429-439.

[32] S. K. S. Gupta and P. K. Srimani, "An Adaptive Protocol for Reliable Multicast in Mobile Multi-Hop Radio Networks," *Proceedings of the Second IEEE Workshop on Mobile Computing Systems and Applications*, Feb. 1999, pp. 111-122.

[33] S. J. Lee, M. Gerla, and C. C. Chiang, "On-Demand Multicast Routing Protocol," *the IEEE Wireless Communications and Networking Conference*, vol.3, Sep. 1999, pp. 1298-1302.

[34] J. J. Garcia-Luna-Aceves and E. L. Madruga, "The Core-Assisted Mesh Protocol," *the IEEE Journal on Selected Areas in Communications*, vol.17, iss.8, Aug. 1999, pp. 1380-1394.

[35] P. Sinha, R. Sivakumar, and V. Bharghavan, "MCEDAR: Multicast Core-Extraction Distributed Ad Hoc Routing," *the IEEE Wireless Communications and Networking Conference*, vol.3, Sep. 1999, pp. 1313-1317.

[36] P. Sinha, R. Sivakumar, and V. Bharghavan, "CEDAR: a Core-Extraction Distributed Ad Hoc Routing Algorithm," *the IEEE Journal on Selected Areas in Communications*, vol.17, iss.8, Aug. 1999, pp. 1454-1465.

[37] K. Viswanath, K. Obraczka, and G. Tsudik, "Exploring mesh and tree-based multicast. Routing protocols for MANETs," *the IEEE Mobile Computing Journal*, vol.5, iss.1, Jan. 2006, pp. 28-42.

[38] T. Kunz and E. Cheng, "On-demand multicasting in ad-hoc networks:comparing AODV and ODMRP," *Distributed Computing Systems, Proceedings of the 22nd International Conference*, Jul. 2002, pp. 453-454.

[39] <http://www.isi.edu/nsnam/ns/ns-documentation.html>

[40] T. Camp, J. Boleng, and V. Davies, "A Survey of Mobility Models for Ad Hoc Network Research," *Wireless Communications and Mobile Computing: Special Issue on Mobile Ad Hoc Networking*, vol.2, no.5, Dec. 2002, pp. 483-502.