

南 華 大 學

資訊管理研究所

碩士論文

行動隨意網路中高穩定式位置輔助之繞徑協定

High Stability and Location-Aided Routing Method for Mobile

Ad Hoc Network



研 究 生：賴建穎

指 導 教 授：吳光閔 博士

中 華 民 國 九 十 六 年 六 月

南 華 大 學

資訊管理研究所

碩士學位論文

行動隨意網路中高穩定式位置輔助之繞徑協定

研究生：賴建穎

經考試合格特此證明

口試委員：

蔡德誨

阮金聲

吳光燁

指導教授：吳光燁

系主任(所長)：資訊管理學系 系主任 吳光燁

口試日期：中華民國 96 年 6 月 8 日

## 誌 謝

沒想到研究所兩年的時間會過的這麼的快，雖然很短暫但學習上卻很充足；剛進來無線小組時，多虧有士颺學長的帶領，讓我能很快的融入無線的領域，他一派輕鬆的態度以及幽默，也帶給研究室不少歡樂。

至今完成了這篇論文，很感謝吳光閔主任以及蔡德謙老師除了在專業知識的指導，也只教導了我們在待人處事上的很多道理，讓我們成長了很多也很順利完成學業。另外，我也要感謝無線小組的所有夥伴珊珊、岳勳以及可愛的學弟妹小豐、詠生、淳淳，和你們在一起努力研究和聊天的時光，真的讓我覺得很開心，謝謝你們。

最重要的我要感謝我最愛的女友雅君，從大學一路陪伴我到研究所，總是默默的支持我、鼓勵我，我才能有如今的成長。當然還要感謝我的家人爸爸、媽媽、大姊、二姐，他們時常的關心我、支援我，因為有他們我才能無顧慮的去完成我的學位，真的很謝謝你們。

最後，再次深深表達我對所有人的謝意，**非常謝謝你們！**

# 行動隨意網路中高穩定式位置輔助之繞徑協定

學生：賴建穎

指導教授：吳光閔 博士

南 華 大 學 資 訊 管 理 學 系 碩 士 班

## 摘 要

在行動隨意網路 (Mobile Ad hoc Network, MANET) 環境下，每一個行動主機 (Mobile Host, MH) 是透過彼此之間的連線 (Link) 以及多點跳躍 (Multi-Hop) 的方式將資料封包送達目的地。但在 MANET 環境下，過多的控制封包 (Control Packet) 在網路上傳遞，可能會浪費網路資源，甚至造成網路癱瘓；而如何減少路徑上的跳躍次數，也是繞徑協定上重要的議題。

有許多的繞徑協定 (Routing Protocol) 被提出來研究，也有將無線網路上的節點 (Node) 加以定位 (Location)，利用座標 (Position) 的資訊提出不同方式的繞徑協定。本篇研究即是屬於位置輔助繞徑協定 (Location-Aided Routing Protocol)，結合已知的座標位置，可以求得任意兩個節點之間的最短直線 (Baseline)，並且因為座標資訊的關係，能有方向性的找尋路徑，減少封包漫無目的的在網路上流竄。透過本研究，我們提出 High Stable Location Aided Routing Protocol (HSLAR) 希望能控制路徑要求 (Route Request, RREQ) 封包朝向正確的方向傳遞，並且建立出最接近直線距離的路徑。

透過模擬實驗的結果，顯示出本文所提出的 HSLAR 協定可以減少 18.06% 至 28.83% 的 Hop Count，並且有效降低封包 16.13% 至 27.65% 的發送數量，以及減少傳輸路徑約 1.01% 至 12.3% 斷裂的機率。

**關鍵詞：隨意網路、繞徑協定、Location-Aided Routing**

# High Stability and Location-Aided Routing Method for Mobile Ad Hoc Network

Student : Chien-Ying Lai

Advisors : Dr. Guang-Ming Wu

Department of Information Management  
The M.I.M. Program  
Nan-Hua University

## **ABSTRACT**

In Mobile Ad-hoc Network (MANET), the mobile hosts (MH) intercommunicate through single-hop or multi-hop paths in a peer-to-peer mechanism. But more control overhead of routes may waste resources of network. One problem of routing protocol is to cut down hop counts of a route.

Part of routing protocol was proposed with location by using the global positioning system (GPS), for instance, LAR and ILAR scheme. In this paper, we propose a High Stability and Location-Aided Routing (HSLAR) scheme to improve the ILAR. Our method can reduce control overhead, hop counts and to enhance the stability of routes. We use the MHs' location information to calculate a baseline between source node (or forward node) and destination node. And then depends on these baselines to establish a route.

The simulation results show that our scheme is more efficiently on control overhead, hop counts and the stability of routes than ILAR scheme.

**Keywords: MANET, Routing Protocol, Location-Aided Routing**

# 目 錄

書名頁.....	i
博碩士論文授權書.....	ii
著作財產權同意書.....	iii
論文指導教授推薦函.....	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.....	v
圖 目 錄.....	vi
第一章 緒論.....	1
1.1 研究背景.....	1
1.2 相關研究.....	4
1.2.1 DSDV Protocol.....	6
1.2.2 AODV Protocol.....	7
1.2.3 ZRP Protocol.....	9
1.3 Position Based Routing Protocol.....	10
1.3.1 LAR Protocol.....	11
1.3.2 ILAR Protocol.....	15
1.3.3 High Stable Location Aided Routing Protocol (HSLAR).....	20
1.4 章節架構.....	22
第二章 問題描述.....	23
第三章 研究方法.....	26
3.1 基本環境介紹.....	26
3.2 封包格式描述.....	27
3.3 尋找路徑階段.....	31
3.4 路徑維護階段.....	35
第四章 模擬實驗.....	38
4.1 模擬環境.....	38
4.2 模擬結果.....	39
第五章 結論.....	48
參考文獻.....	49

## 表 目 錄

表 1 Table-Driven Routing 和 On-Demand Routing 的比較 .....	5
表 2 Route Request Transmit (RREQ_T) 封包格式 .....	28
表 3 Route Request Revise (RREQ_R) 封包格式 .....	28
表 4 Route Request (RREQ) 封包格式 .....	29
表 5 Route Reply (RREP) 封包格式 .....	30
表 6 Route Error (RERR) 封包格式 .....	31
表 7 模擬環境的參數表 .....	39
表 8 50 個節點下平均 Hop Count 之效能分析 .....	40
表 9 移動速率 40 km/hr 下平均 Hop Count 之效能分析 .....	41
表 10 50 個節點下平均控制封包的發送量之效能分析 .....	43
表 11 移動速率 40 km/hr 下平均控制封包的發送量之效能分析 .....	44
表 12 50 個節點下傳輸路徑平均的斷裂比率之效能分析 .....	46
表 13 移動速率 40 km/hr 下傳輸路徑平均的斷裂比率之效能分析 .....	47

## 圖目錄

圖 1 有基礎架構的無線網路.....	2
圖 2 無基礎架構的無線網路或 MANET.....	3
圖 3 AODV 協定.....	9
圖 4 ZRP 協定.....	10
圖 5 目的地節點 D 在 $t_1$ 的時間可能移動的範圍 (Expected Zone) .....	11
圖 6 當來源節點 S 落在 Expect Zone 之外所定義出的 Request Zone .....	13
圖 7 當來源節點 S 落在 Expect Zone 之內所定義出的 Request Zone .....	13
圖 8 LAR scheme 2.....	14
圖 9 LAR 協定的 Request Zone 所產生的問題.....	15
圖 10 ILAR 協定的 Baseline.....	16
圖 11 DIST 與 VDIST 的關係 .....	17
圖 12 ILAR 協定建立路徑的例子.....	18
圖 13 ILAR 協定的 Partial reconstruction 機制 .....	19
圖 14 Border Node 和 Threshold 的圖例.....	21
圖 15 減少傳輸路徑的 Hop Count .....	25
圖 16 鋸齒狀傳輸路徑的問題.....	25
圖 17 以 Direction 方位取代 Request Zone.....	25
圖 18 不同目的地節點所定義出不同 Direction 的圖示.....	32
圖 19 ILAR 協定建立的傳輸路徑.....	34
圖 20 HSLAR 協定建立的傳輸路徑.....	35
圖 21 Full Repair 的維護機制.....	36
圖 22 Local Repair 的維護機制 .....	36
圖 23 Local Repair 模式的圖例 .....	37
圖 24 50 個節點下平均 Hop Count 之效能分析 .....	40
圖 25 移動速率 40 km/hr 下平均 Hop Count 之效能分析.....	41
圖 26 50 個節點下平均控制封包的發送量之效能分析 .....	42
圖 27 移動速率 40 km/hr 下平均控制封包的發送量之效能分析 .....	44
圖 28 50 個節點下傳輸路徑平均的斷裂比率之效能分析 .....	45
圖 29 移動速率 40 km/hr 下傳輸路徑平均的斷裂比率之效能分析 .....	47



# 第一章 緒論

隨著時代的進步，網路的使用早已融入我們的日常生活之中，網路的使用似乎已成為人們生活的必需品。網路技術也由有線網路慢慢的轉變成以無線網路為主的架構，人們對無線網路的需求也與日俱增。無線網路可以是現有有線網路的擴充，因為克服有線網路的空間限制，使得無線網路架構的使用上非常的有彈性。

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)。[19]

## 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)，另一種是用戶端 (Client)，主要是以基地台為重要的角色，因為它負責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可以透過有線網路來連線，但是在基地台廣播覆蓋範圍下的用戶端是可以透過無線的方式來與伺服器 (Server) 進行連線，如圖 1 的網路架構圖。在這個架構下的廣播排程方式，主要又可分為推式 (Push) [1, 2, 21] 和拉式 (Pull) [2, 14, 21, 22, 28] 兩種。

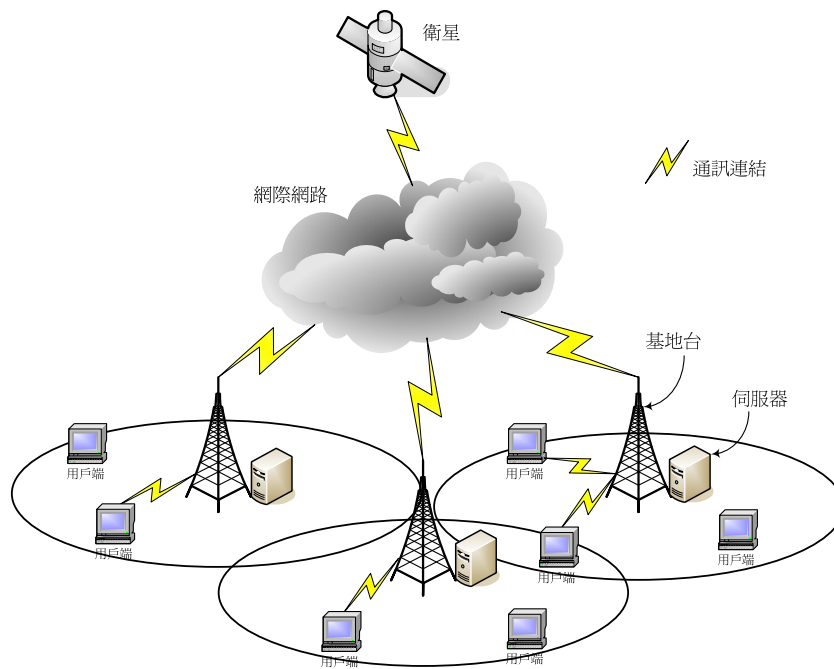


圖 1、有基礎架構的無線網路

以推式廣播為基礎的運作方式，是 Server 端將資料項目推送給用戶端，也就是由 Server 端不斷廣播所有的資料項目，而用戶端則一直監聽從 Server 端廣播送出的資料項目，如果監聽到需要的資料，再將資料截取下來，這樣的方式不需產生任何詳細的用戶端要求 (Request)。而以拉式廣播為基礎的廣播方式，是由用戶端送出所有需要資料項目的要求，經由傳送請求到達 Server 端，然後 Server 端在將此資料項目排入廣播排程來滿足用戶端的請求。

不論推式或是拉式廣播排程方式都有其優缺點，所以 Saxena 等人在[21]上提出一種結合兩種方式優點的混合式排程演算法。Cai 等人也在[5]提出 Hybrid Wireless Broadcast (HWB model) 的模式架構，透過協同合作的方式結合推式和拉式的廣播排程，使得用戶端能更有效率的取得資料項目。

另一方面，在 Non-infrastructure Network 裡，主要的架構我們稱之為 Mobile Ad hoc Network，它與 Infrastructure Network 剛好相反，在 MANET 的環境下，並沒有所謂的基地台存在，如圖 2 所示。它是透過行動主機彼此之間的連線做為通訊的方式，也就是任意一個行動主機必須透過另一個或多個行動主機的跳躍（Hop）來傳遞訊息，或將資料封包傳送至目的地，所以每一個行動主機必須和其他主機互助合作，尋找主機之間的資料傳輸路徑。因此，像是基地台的集中式管理或定期標準化服務的支援是不存在 MANET 的環境底下。在 MANET 的網路系統裡，行動主機透過相鄰的主機彼此將資料封包轉送到目的地，因此，在這樣的情況之下傳輸路徑（sending path）的選擇也變成相當的重要，因為藉由良好的繞徑協定可以更有效率的使用網路資源，同時也提供更快速且穩定的傳輸路徑。

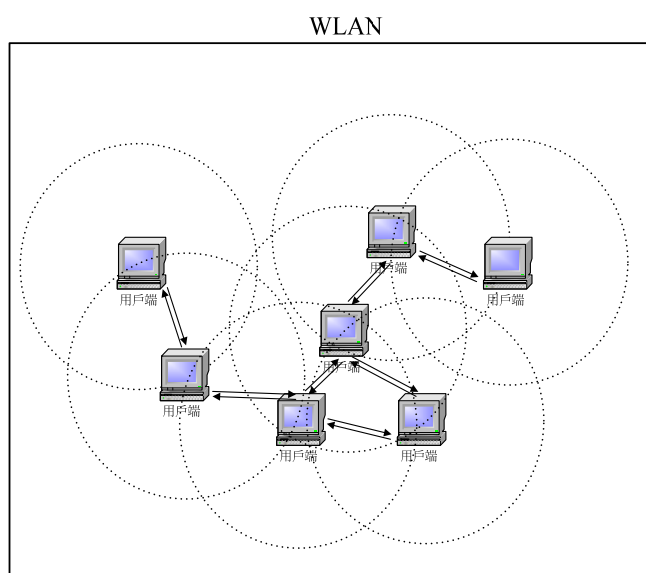


圖 2、無基礎架構的無線網路或 MANET

由於行動隨意無線網路的特性，行動主機的可移動性 (Mobility)、有限的無線廣播涵蓋範圍和有限的電力 (Power)，使得建立好的路徑可能因為行動主機的移動造成網路拓樸 (Network Topology) 的改變，而發生斷裂的情形。因此，我們發現在行動隨意無線網路最關心的議題，在於一個無法預知的網路拓樸，或者沒有像是基地台的集中式管理中心的環境下，如何有效的提高行動主機之間的訊息交換以及資料封包的傳送。除此之外，因為沒有整個網路拓樸的關係，通常廣播都是使用氾濫式 (Flooding) 技術 [8, 16, 20, 27, 29]，所以也容易造成「廣播風暴問題」 (Broadcast Storm Problem) [8, 20, 27]，如何減少重覆性 (Redundancy) 的封包廣播，也是重要的議題之一。

## 1.2 相關研究

在 MANET 的繞徑協定裡，主要可分為兩種類型：Table-Driven Routing 和 On-Demand Routing。所謂的 Table-Driven Routing 方式，是指在 MANET 環境下每一個節點本身都存有 Routing 的資訊，並且週期性的去更新在這個路由表 (Routing Table) 內的資訊，以備在當有節點需要路徑來傳送資料封包時能快速的反應；是屬於主動式 (Proactive Type) 更新的方法。它固定每隔一段時間就必需去維護整個網路的拓樸 (topology)，因此，較適合使用於靜態 (Static) 或移動性較低 (Low Mobility) 的網路環境。Table-Driven Routing 方式的缺點為需要維護 (Maintain) 用的表格，此表格將會佔據不少空間，另一個缺點為在週期性的更新時，所必需發送的廣播資訊，這導致網路頻寬的使用比較沒有效率，也常造成相當嚴重的網路負載量

(Overhead)。著名的 Table-Driven Routing 方式有 DSDV[9, 15]、TORA[7]等等。

MANET 環境下的另一種 Routing 方式為 On-Demand Routing 模式。在 On-Demand Routing 方式中，只有在節點有需要路徑時，才會發送尋找路徑的封包，是屬於被動式 (Reactive Type) 的方式，它亦可以降低在 Table-Driven Routing 方式下的網路，大量封包傳送的網路負載量過大問題，所以它適合使用在動態 (Dynamic) 或移動性較高 (High Mobility) 的網路環境下。而 On-Demand Routing 方式通常可以區分為兩個階段，第一個階段為找尋路徑 (Route Discovery)，第二個階段為資料傳送 (Data Transfer)。由於 On-Demand Routing 方式沒有週期性的更新與廣播，只有在有需要時才去找尋傳輸路徑，因此在網路頻寬的使用上會比 Table-Driven Routing 較有效率，但在找尋路徑程序上卻比 Table-Driven Routing 方式會多花費一些時間來建立路徑。著名的一些 On-Demand Routing 演算法有 AODV[4, 6, 18, 23, 24, 25, 30]、ABR[7]和 DSR[10, 17]等等。表 1 為兩種類型的比較。

表 1、Table-Driven Routing 和 On-Demand Routing 的比較

	Table-Driven Routing	On-Demand Routing
主要優點	能立即提供可用路徑	不需要週期性的廣播來維護路由表，能節省網路資源

主要缺點	必須週期性的廣播來維護路由表，所產生的控制封包，會大量消耗網路資源	路徑的建立與維護都比較耗時
典型代表	DSDV	AODV

由上表得知，雖然兩種類型的繞徑協定各有其有缺點，但是已經有許多著名的研究和實驗證明了 On-Demand Routing 比起 Table-Driven Routing 較能有效的降低廣播風暴問題，進而提升網路資源的使用效率。下一小節裡，我們將分別介紹 Table-Driven Routing 典型的代表 DSDV 協定和 On-Demand Routing 典型的代表 AODV 協定。

### 1.2.1 DSDV Protocol

Destination-Sequence Distance Vector (DSDV) 繞徑協定[9, 15]是屬於 Table-Driven Routing，在這個網路上的每一個節點都必須去建立一個到達網路上其他節點的路由表，並經由週期性的廣播以及路由表的交換，使得在網路拓樸經常性變動下，隨時可有最新的路由可以使用。路由表內包含有：目的地節點的編號 (ID)、Next Hop、到達目的地節點的 Hop Count 數目、循序編號 (Sequence Number) 等相關資訊。Next Hop 欄位記錄的是資料封包該往哪一個鄰近節點傳送；Hop Count 數目表示傳送路徑上會經過幾個行動節點；Sequence Number 用來分辨路由表內路徑的新舊程度，並且避免造成迴路(loop route)。

由於在 DSDV 協定下的網路，當網路拓樸變動時，所有網路上的節點會廣播自己的路由表資訊來告知其他節點，而當接收到一個全新的路由表或更動過的路由表，此新的路由表資訊會再次的被廣播出去，這樣的方式會造成廣播風暴的問題，所以 DSDV 協定採用了兩種路由表更新的方式：一種是「Full dump」方式，廣播封包會帶著所有可用的路由排程資訊以及多重網路協定資料單位（Multiple Network Protocol Data Units, NPDUs），可是所挾帶的資料量過大，只適用在移動性較低的網路環境。另一種方式是「Incremental」，只有當網路拓樸變動到足以影響資料封包的傳送時，才會發送封包來更新路由表的資訊，如此可以降低網路的負載量。

### 1.2.2 AODV Protocol

Ad hoc On-demand Distance Vector（AODV）繞徑協定找尋路徑的方法[4, 6, 18, 23, 24, 25, 30]，是當某個行動主機或稱來源節點（Source Node），想要傳送資料封包給另一個行動主機，也可稱作目的地節點（Destination Node），而來源節點本身有沒有記錄到達目的地節點的相關路徑資訊，則來源節點會啟動 Route Discovery 程序，並且向自己的鄰近節點廣播路徑要求的封包（RREQ），接收到 RREQ 封包的鄰居節點會先確定自己是否有到達此目的地節點的路徑資訊，或是自己就是目的地節點，如果都不是則會記錄發送節點（Sending Node）在路徑表格（Routing Table）中，以作為封包回覆的依據，並且繼續將路徑要求封包廣播出去，直到目的地節點接收到 RREQ 封包為止。若是接收到相同的 RREQ 封包時，則會丟棄（Discard）不以理會。而如果接收到 RREQ 封包的節點，有記錄著到達目的地節點的路徑資訊，或是自己本身就是目的地節點，則會依

照路徑表格中或是封包內的路徑資訊回傳路徑回覆 (Route Reply, RREP) 封包給來源節點，便完成 Route Discovery 階段。當路徑中的行動節點偵測到傳輸路徑斷裂時，此節點會再次廣播 RREQ 封包來找尋到達目的地節點的新路徑。

如圖 3 的例子，當來源節點 S 想要傳送資料封包給目的地節點 D 時，會先檢查自己的路徑表格裡是否有可用的路徑資訊，如果沒有可到達目的地節點的傳輸路徑時，來源節點 S 會廣播出 RREQ 封包，當鄰近的節點 A 收到 RREQ 封包時，首先會判斷自己是否為目的地節點，如果不是則檢查路徑表格內是否有到達目的地節點 D 的路徑資訊，若沒有的話，則記錄來源節點 S 後，再將 RREQ 封包繼續廣播給鄰近節點 B、C、F；而這些節點也會重複相同的程序，直到找到目的地節點 D 為止。而在節點 B 接收來自節點 A 的 RREQ 封包後，也從節點 F 以及節點 C 接收到相同的 RREQ 封包，則會丟棄不理會，避免造成封包在環狀 (loop) 路徑上不斷的傳送。

在目的地節點 D 接收到路徑要求封包後，會反向 RREQ 封包內的路徑資訊，並且以單播 (unicast) 的方式回傳路徑回覆封包 (RREP) 到來源節點 S，當路徑上的每一個節點接收到 RREP 封包時，便會記錄到目的地節點 D 的快取資訊，而來源節點 S 接收到 RREP 封包後便會使用封包內的路徑資訊來傳遞資料封包。



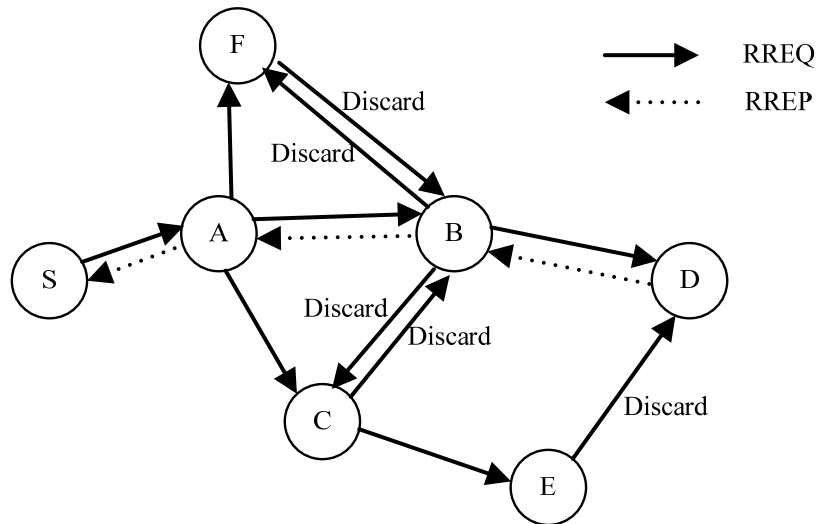


圖 3、AODV 協定

### 1.2.3 ZRP Protocol

除了上述的兩種協定的類型之外，還有一類的繞徑協定混合了主動式與被動式繞徑協定，例如：Zone-Based Routing Protocol (ZRP) [3, 27]就是這種類型的繞徑協定，如圖 4 所示。ZRP 協定定義了一個 Hop Count (HC) 或兩個甚至多個以上的範圍 (Zone) 節點皆採用主動式的繞徑協定，即是維護一個定義好的 HC 內的路徑表格；而超出 Zone 外的目的地節點，會採用被動式的繞徑協定，在需要路徑傳輸資料封包的時候，廣播出 RREQ 封包來建立路徑。即使 ZRP 繞徑協定混合了主動式以及被動式的優點，但如何定義出適合的 Zone 也是此類型協定的難題。如果 Zone 定義的太大容易增加網路的負載量，Zone 定義的太小又無法及時反應節點對傳輸路徑的需求。

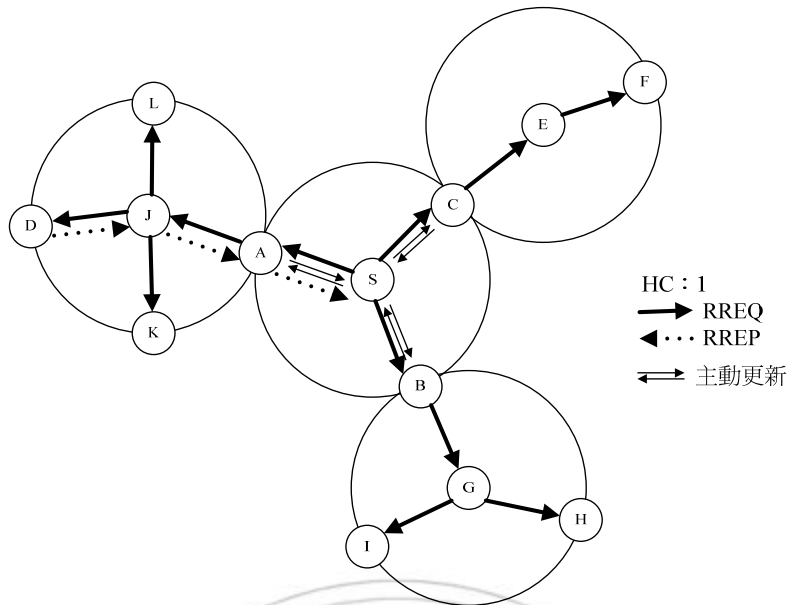


圖 4、ZRP 協定

以上所列舉的都是行動隨意網路常見的繞徑協定，而這些繞徑協定又可歸納為位置不相關 (Position Less)，在 MANET 的網路環境裡還有一類為位置相關 (Position Based) 的應用，這一類型的繞徑協定我們將在下一個小節來探討。

### 1.3 Position Based Routing Protocol

在這個位置相關應用下的繞徑協定，網路上的每一個行動節點可以透過一套定位系統來定位它自己的座標位置、距離等資訊，例如：GPS 定位系統、Sensor Device 等設備；應用在行動隨意網路上便可以輕易獲得相關資訊來進行繞徑協定的運作。結合位置相關的繞徑協定有 GPSR[11]、GOLI[13]、LAR[12]、ILAR[26]和 LAAR( Location-Based Adaptive Ad hoc Routing ) [30]等技術。

### 1.3.1 LAR Protocol

Location-Aided Routing (LAR) Protocol 是由 Ko 和 Vaidya 所提出。LAR 協定的主要目的是在找尋路徑階段，希望能夠有效縮小氾濫式廣播的範圍，減少網路負載的問題。在網路上的所有節點都可以透過 Global Positioning System (GPS) 定位系統來取得自己或是其他目的地節點 D 的座標位置，並使用這些座標資訊來計算預測區 (Expected Zone) 以及廣播區 (Request Zone)。

假設來源節點 S 在  $t_1$  的時間點已知目的地節點 D 在時間點  $t_0$  的座標資訊  $D(X_d, Y_d)$  以及速度  $V$ ，所以來源節點 S 可以在  $t_1$  的時間推算出目的地節點 D 可能移動的區域範圍，我們稱之為 Expected Zone。以圖 5 為例，來源節點 S 已知目的地節點 D 的座標  $D(X_d, Y_d)$  以及平均速度  $V$ ，即可計算出 Expected Zone 的半徑為  $V(t_1 - t_0)$ 。但如果實際速度大於平均速度，那麼目的地節點 D 移動出 Expected Zone，一般而言，目的地節點 D 的移動速度  $V$  會設定在最大移動速度，或是利用數理統計推算出適當的速度來取代平均速度。

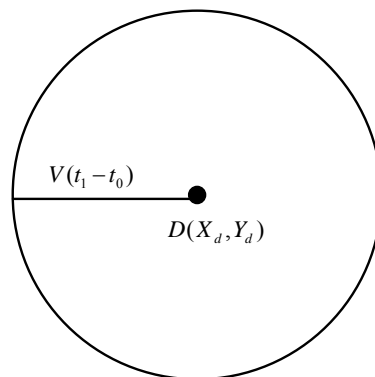


圖 5、目的地節點 D 在  $t_1$  的時間可能移動的範圍 (Expected Zone)

此外 LAR 協定最主要的目的是減少氾濫式廣播所造成的廣播風暴問題以及網路的高負載量，所以來源節點 S 還需要定義出 RREQ 封包的傳遞地區域，我們稱之為 Request Zone。以圖 6 的例子來做說明，由來源節點 S 的座標資訊  $S(X_s, Y_s)$ 、目的地節點 D 的座標資訊  $D(X_d, Y_d)$  和 Expect Zone 的半徑  $R = V(t_1 - t_0)$ ，可定義出 Request Zone 的四個頂點： $S(X_s, Y_s)$ 、 $T(X_s, Y_d + R)$ 、 $U(X_d + R, Y_d + R)$ 、 $W(X_d + R, Y_s)$ 。當來源節點 S 啟動路徑找尋階段發出 RREQ 封包時，只有落在 Request Zone 內的節點有責任繼續傳遞 RREQ 封包，如圖中的節點 B；而節點 A 因為不在 Request Zone 之內，所以當它接收到來源節點 S 廣播的 RREQ 封包時，並不會繼續將封包傳遞出去。

還有另一種情況，當來源節點 S 落在 Expect Zone 之內，則 Request Zone 的四個頂點分別為： $P(X_d - R, Y_d - R)$ 、 $T(X_d - R, Y_d + R)$ 、 $U(X_d + R, Y_d + R)$ 、 $Q(X_d + R, Y_d - R)$ ，如圖 7 所示。因此，LAR 協定能經由 Request Zone 降低氾濫式廣播所導致廣播風暴的問題，並且減輕網路的負載量。

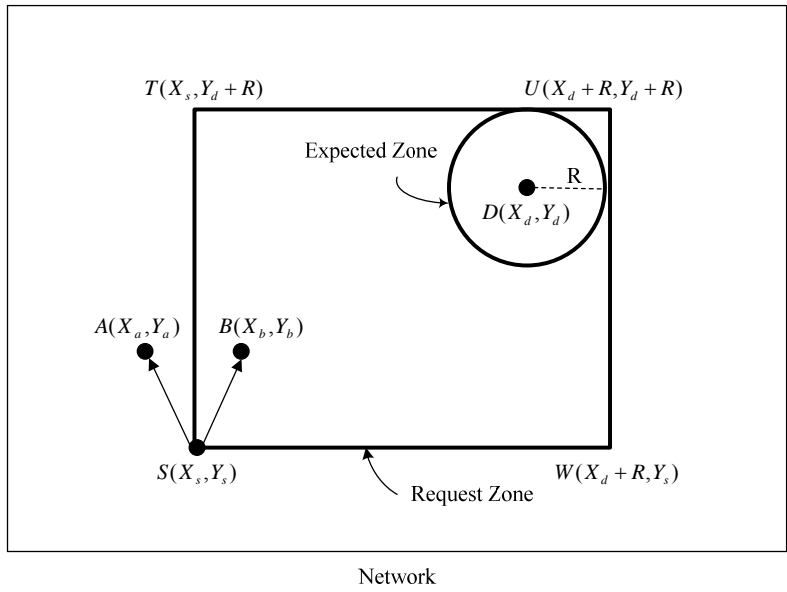


圖 6、當來源節點 S 落在 Expect Zone 之外所定義出的 Request Zone

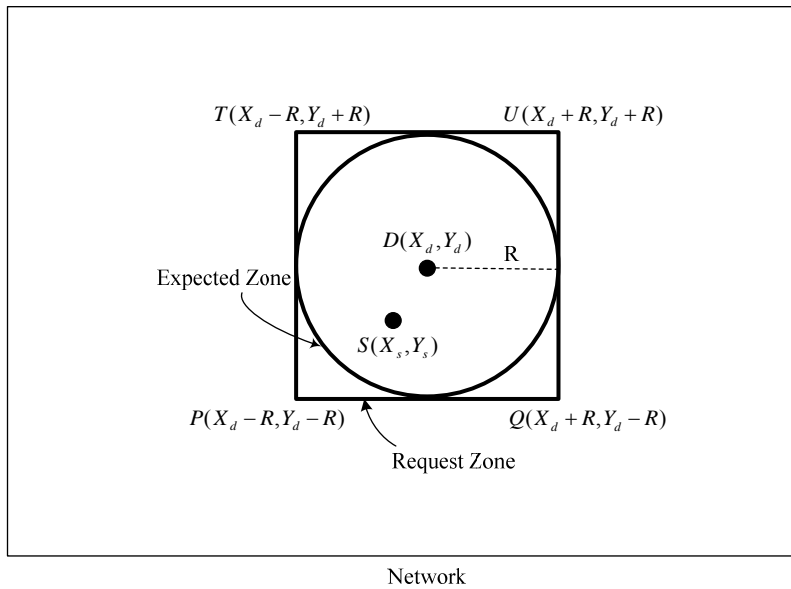


圖 7、當來源節點 S 落在 Expect Zone 之內所定義出的 Request Zone

當目的地節點 D 接收到 RREQ 封包時，首先它會將自己目前的座標資訊 (current location) 以及時間 (current time)，加入 RREP 封包裡，回傳給來源節點 S。來源節點 S 接收到 RREP 的封包，它會將目的地節點 D 的座標位置及相關資訊記錄起來，並在需要的時候使用。

在 LAR 協定下，當來源節點 S 需要建立路徑來傳送資料封包給目的地節點 D 時，它會先定義出 Expect Zone 以及 Request Zone，才開始執行 Route Discovery 階段，此種方式又可稱為 LAR scheme 1。而在[29]還提出了 LAR scheme 2 的方式，如圖 8 所示，來源節點 S 已知目的地節點 D 的座標位置  $D(X_d, Y_d)$ ，計算出與目的地節點 D 的距離長度 ( $DIST_s$ ) 並記錄在 RREQ 封包，當鄰近的節點 i 收到來源節點傳送的 RREQ 封包，也同樣地求得  $DIST_i$ ，如果  $\alpha(DIST_s) + \beta \geq DIST_i$  (通常  $\alpha=1, \beta=0$ )，則節點 i 會將 RREQ 封包內的  $DIST_s$  以  $DIST_i$  取代，再廣播給其他鄰近的節點。依此類推，當網路上其他節點接收到 RREQ 封包時，如果是重複接收則不以理會；反之依照前述的程序來決定是否廣播 RREQ 封包。

LAR scheme 2 下目的地節點 D 仍會將自己目前的座標資訊以及時間加入 RREP 封包回覆給來源節點 S。來源節點 S 也會將目的地節點 D 的座標位置及相關資訊記錄起來，以供需要的時候使用。

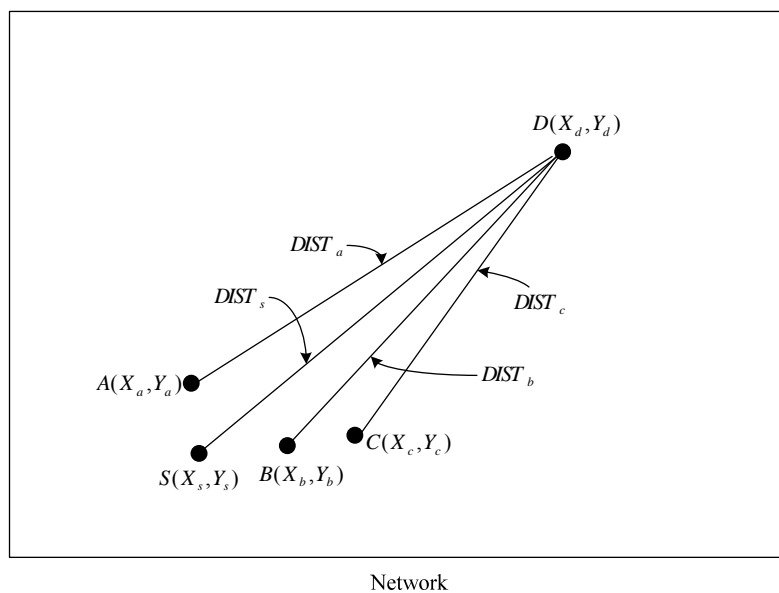


圖 8、LAR scheme 2

### 1.3.2 ILAR Protocol

即使在 LAR 協定裡使用了 Expected Zone 和 Request Zone 的機制來控制封包流向，用以減少在整個網路上傳遞的封包數量，但顯然的仍是不夠的，因為 Request Zone 可能因為來源節點 S 和目的地節點 D 之間的距離越遠而越大，而無法減輕氾濫式廣播所帶來的負擔，請參閱圖 9 所示。因此，Wang 等人便提出了一個改善 LAR protocol 的方法，可以透過更有效率的路徑找尋程序，來進一步的減少封包數量，這個改善的方法稱之為 Improved Location-Aided Routing (ILAR) 協定。

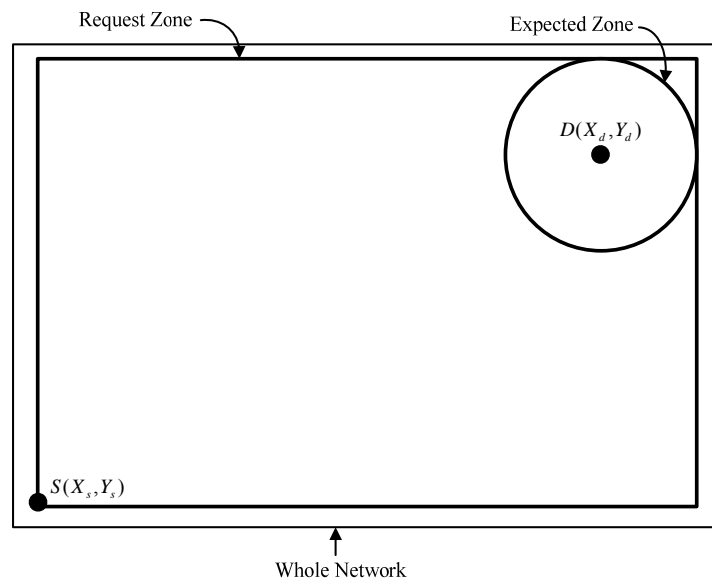


圖 9、LAR 協定的 Request Zone 所產生的問題

ILAR 與 LAR 協定都是透過 Position Based 的方式來建立 Expected Zone 和 Request Zone，請參閱圖 10 中 ILAR 協定的範例。不過，為了能更精確的找尋最佳路徑，ILAR 協定在來源節點 S 和目

的地節點 D 建立了一條基準線 (Baseline)，公式： $(X_d - X_s)(Y - Y_s) - (Y_d - Y_s)(X - X_s) = 0$  (1)，期望節點在 Route Discovery 的階段可以依循著此基準線而進行，更進一步的減少在 Request Zone 裡不必要封包的回應與轉送，請參閱圖 10 所示。

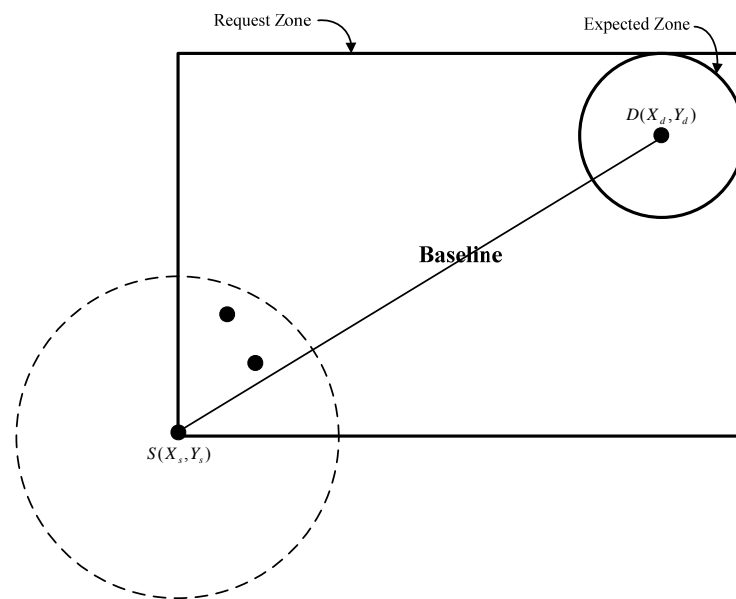


圖 10、ILAR 協定的 Baseline

ILAR 協定一開始將 Expected Zone、Request Zone 和 Baseline 定義好後，並將 Request Zone 的資訊記錄在 RREQ 封包裡，由來源節點 S 發出 RREQ 封包啟動 Route Discovery 程序，一旦鄰近節點接收到 RREQ 封包，他們可以推算自己是否在 Request Zone 裡，而且回應一個 RREQ\_R (Route Request Revise) 的封包給原本的來源節點 S，這個 RREQ\_R 的封包記錄了回覆節點自己的 VDIST 和 DIST。VDIST 是指這個節點到 Baseline 的最短距離，換句話說，也就是與 Baseline 垂直的距離長度；DIST 則是記錄此節點到來源節點 S 的距離長度，



公式如下：

$$DIST = \sqrt{(X_s - X_i)^2 + (Y_s - Y_i)^2} \quad (2)$$

假設 Baseline 的公式為  $ax + by + c = 0$ ，而 VDIST 的公式如下：

$$VDIST = \left| \frac{ax + by + c}{\sqrt{a^2 + b^2}} \right| \quad (3)$$

圖 11 裡明確的顯示節點的 VDIST、DIST、Baseline 和來源節點 S 的關係。圖中節點 A 座標為  $A(X_a, Y_a)$ ，來源節點 S 座標為  $S(X_s, Y_s)$ ，而  $DIST_a$  的值只要運用畢氏定理[32]的觀念，如公式(2)即可輕鬆得到；如圖 11 中的  $DIST_a = \sqrt{(X_s - X_a)^2 + (Y_s - Y_a)^2}$  即可求得。將節點 A 的  $X_a$ 、 $Y_a$  座標值代入公式(3)即可算出  $VDIST_a$  的值。

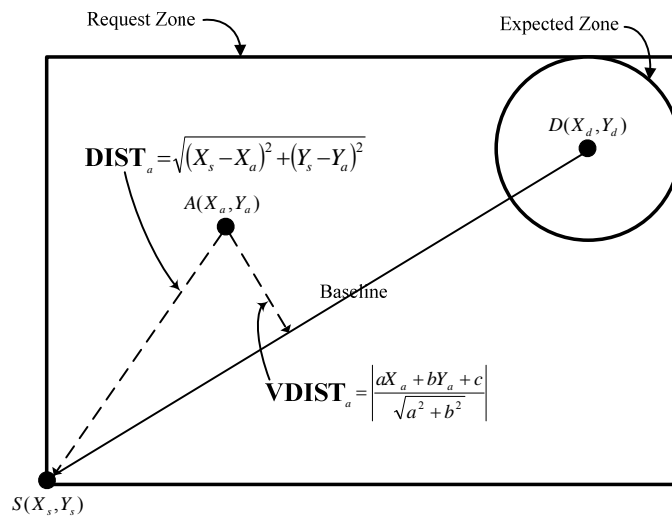


圖 11、DIST 與 VDIST 的關係

然後來源節點 S 接收到 RREQ\_R 的封包之後，將會比對所有回傳 RREQ\_R 封包的節點的 VDIST，選出 VDIST 最小的鄰近節點作為下個廣播 RREQ 封包的轉送節點，被選出的轉送節點繼續負責傳遞 RREQ 封包的責任。此外在轉送節點廣播範圍內的候選節點，它們的 DIST 都必須是越來越大的（比轉送節點的 DIST 還大），這樣才能保證廣播的方向是朝目的地節點 D 方向推送，而不會是逆流向來源節點 S。透過這樣的方式，RREQ 封包可以順利到達目的地節點 D。圖 12 裡顯示 ILAR 方法下的 Routing Discovery 的情況。

目的地節點 D 接收到 RREQ 封包之後，便可以依照附加在 RREQ 封包裡路徑的資訊，透過反轉路徑回傳 RREP 封包給來源節點 S。如此一來便可以成功建立傳輸路徑。如圖 12 的例子說明，路徑是依循著基準線附近的節點所建立起來的。

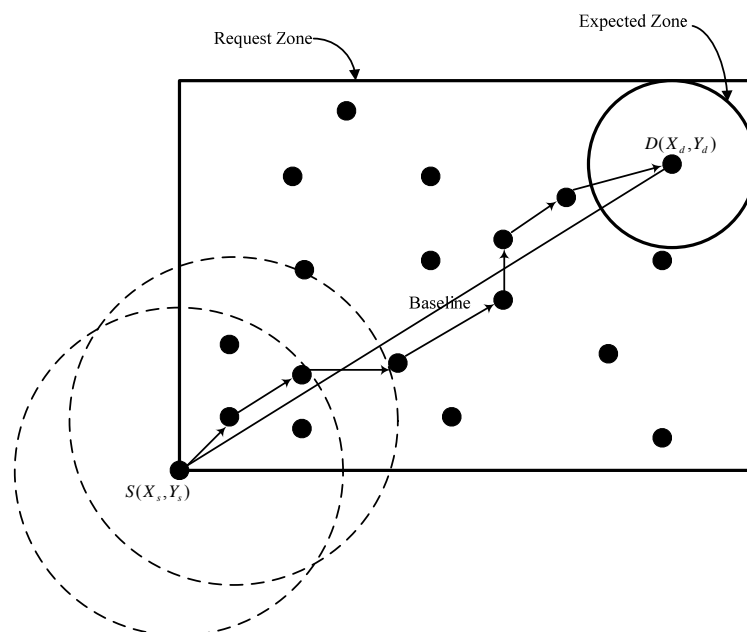


圖 12、ILAR 協定建立路徑的例子

在 MANET 的網路環境下，節點有可能是快速移動的。所以除了考慮 Route Discovery 的方法外，ILAR 協定也明確的描述當路徑斷裂（Route Break）時的路徑維護（Route Maintenance）機制。

在 Route Maintenance 方法上可以分成 Full reconstruction 和 Partial reconstruction 兩種方式。Full reconstruction 是當節點發現路徑失敗時，會回傳 RERR（Route Error）封包給來源節點 S，然後再由來源節點 S 重新執行 Route Discovery 的程序尋找一條到達目的地節點 D 的新傳輸路徑。但在重建路徑上，這樣的方式將會導致有更高的 Overhead。所以 Partial reconstruction 的效能則會比 Full reconstruction 好，因為它在節點發現連接失敗時，會由發現路徑斷裂的節點重新尋找合適的路徑，如果再找不到也會再往回發送 RERR 封包給上個 Node，然後由上一個節點重新尋找合適的傳輸路徑，依此類推。在這樣的情況下，RERR 封包不會一次就回到來源節點 S，而是慢慢的往上一個節點繼續尋找。如果路徑上的節點有另外一條適合的傳輸路徑，那麼就會選擇此路徑做為傳輸路徑。

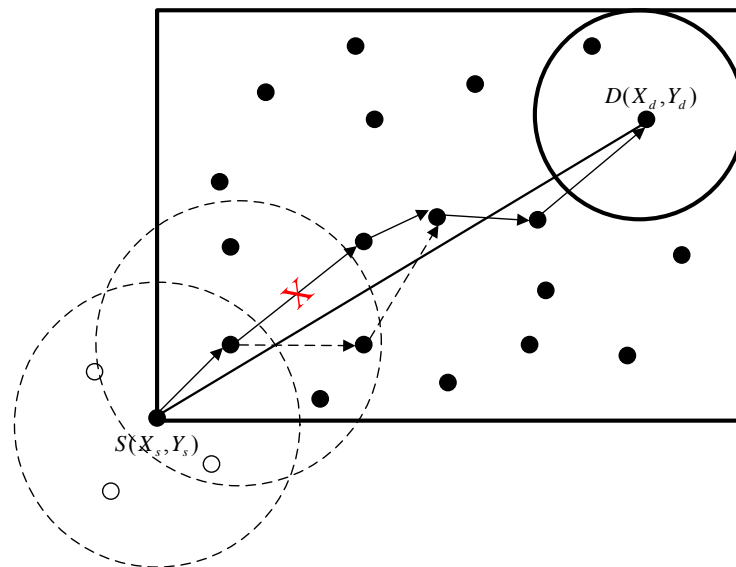


圖 13、ILAR 協定的 Partial reconstruction 機制

因此，Partial reconstruction 會比 Full reconstruction 有更好的維護效果。ILAR 協定在 Route Maintenance 上使用 Partial reconstruction 的方式，如此一來可以更有效的減少不必要的封包消耗網路的資源使用。圖 13 裡顯示 ILAR 協定在路徑維護時的情況。

即使 ILAR 協定提出更有效率的方法來改進 LAR 協定不足的部分，但是我們仍然發現 ILAR 協定有隱藏的問題，在建立路徑上還是會有更好的路徑選擇存在。

### 1.3.3 High Stable Location Aided Routing Protocol (HSLAR)

本篇研究提出 HSLAR 協定，在 Route Discovery 階段，利用已知的座標位置，可以求得來源節點 S 和目的地節點 D 之間的  $Baseline_{(s,d)}$ ，以及目的地節點 D 位於的方位 (Direction)，來源節點 S 在依照目的地節點 D 的方位來發送 RREQ 封包給下一個轉送節點，並非以全面性的氾濫式廣播，來發送 RREQ 封包，因此可以減少 RREQ 封包漫無目的的在網路上流竄；轉送節點的選擇條件必須不是邊緣節點 (Border Node)，如圖 14 中的節點 C 即是屬於 Border Node 的情況；而且權重值必須是最大的候選節點 (Candidate Node) 才能被選為轉送節點。執行相同的程序，直到目的地節點接收到 RREQ 封包並且回傳 RREP 封包給來源節點 S，傳輸路徑才會成功的被建立。經由單一的轉送節點來傳遞 RREQ 封包，可以減少不必要封包的傳送，並且依循著 Baselines 建立路徑，能減少路徑的 Hop 次數。

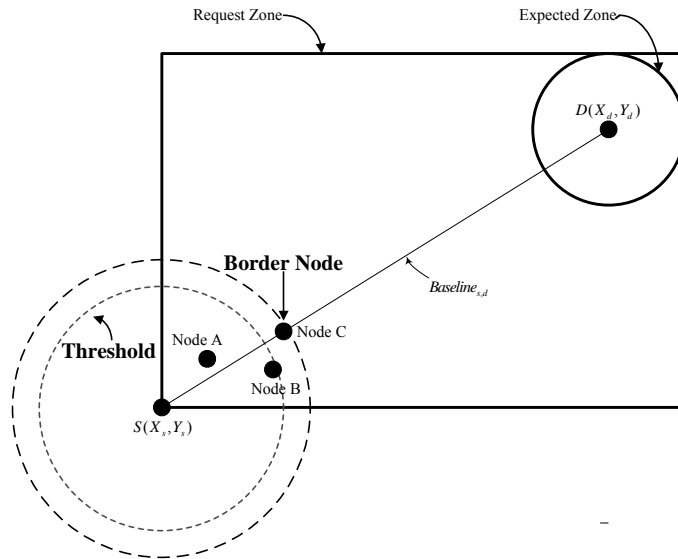


圖 14、Border Node 和 Threshold 的圖例

透過模擬實驗，證明了在網路環境相同的條件下，HSLAR 協定在固定 50 個節點但不同的移動速率的網路環境下，路徑的建立能夠改善 ILAR 協定約 18.06% 的 Hop Count，在封包的平均發送數量來看，最高可降低 16.13% 的封包數量，而傳輸路徑的平均斷裂機率，在高移動速率下約降低了 12.30% 的機率。而在移動速率固定為 40km/hr 但節點數量不同的情況下，HSLAR 協定建立的路徑，能夠改善 ILAR 協定約 28.83% 的 Hop Count，以及降低 27.65% 的封包發送數量，而傳輸路徑的平均斷裂機率，在越多節點數量時降低了約 1.01% 斷裂的機率，很明顯的在越多節點（高密度）的網路環境下，我們所提出的 HSLAR 協定 Route Discovery 階段上的效能會明顯的優於 ILAR 協定。

## 1.4 章節架構

剩下有關本文的部份說明如下：第二章節將說明我們所提出的 HSLAR 所要解決的繞徑問題；第三章節會詳細的介紹我們所提出用來解決所發現四個問題的 HSLAR 協定；第四章節則呈現模擬實驗的結果及分析圖表；最後的章節，是我們對本篇研究的結論。

## 第二章 問題描述

由於 MANET 網路的特性，行動主機具有可移動性，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，使得原本建立好的傳輸路徑發生斷裂的情形。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，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 D 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，所以有專家學者提出 Position Based 的應用來解決這樣的問題。

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，使模擬實驗便於執行；E 表示節點與節點之間的連線數。另外在 Position Based 的繞徑協定下，每一個節點  $i(V_i)$  都有自己的座標位置  $(X_i, Y_i)$ ，例如：來源節點 S 的座標為  $S(X_s, Y_s)$ ，目的地節點 D 的座標為  $D(X_d, Y_d)$ ，然而這些座標資訊一開始都可由 GPS 定位系統或 Sensor Device 來取得。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，而在 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 ILAR 協定已經改善了 LAR 協定所使用氾濫式廣播的問題，但是在路徑上的選擇，仍然有更好的方式存在。透過本章節的說明，我們將描述一般繞徑協定下的問題，並針對這些問題來進行改善。

本研究提出了一個 Position Based 的繞徑協定，透過座標位置的輔助，可以從下列幾方面著手，期望能達到減少路徑上的 Hop Count 以及發送封包的數量：

- 選擇 90% 的無線廣播涵蓋範圍內的 Candidate Node。因為 Border Node 會容易造成傳輸路徑的斷裂，如圖 14。我們的方法期望可以建立出一條穩定的路徑。
- 權重值最大的 Candidate Node 為轉送節點。經由權重值公式的計算，能有效的減少路徑上的 Hop Count。如圖 15 所示，我們的路徑會建立在  $V_b \rightarrow V_e$  上，可以比 ILAR 協定所建立的  $V_a \rightarrow V_c \rightarrow V_e$  減少一個 Hop Count。
- 重新計算 Baseline。為了增加路徑的可靠度，以及避免圖 16 所產生鋸齒狀路徑的情況。
- 以 Direction 來取代 Request Zone。由於 Request Zone 除了一開始能讓來源節點避免選擇一個反方向的  $F_i$  之外，而此後將 Request Zone 紀錄在 RREQ 的封包裡並無太大的用處。Direction 的概念如圖 17 所示。



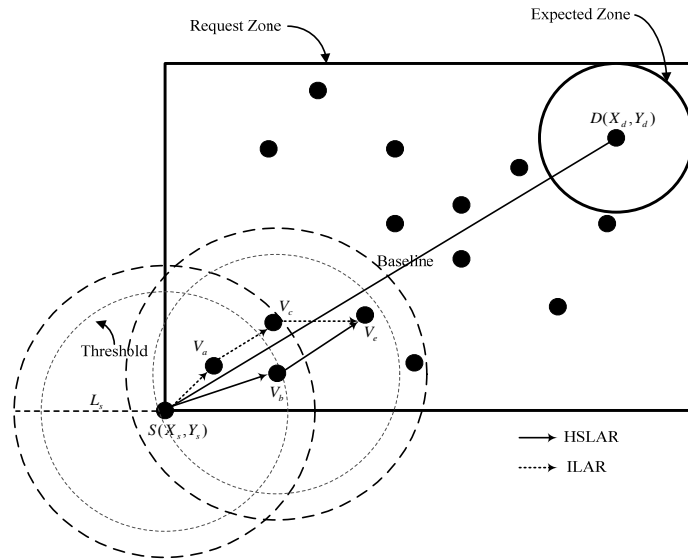


圖 15、減少傳輸路徑的 Hop Count

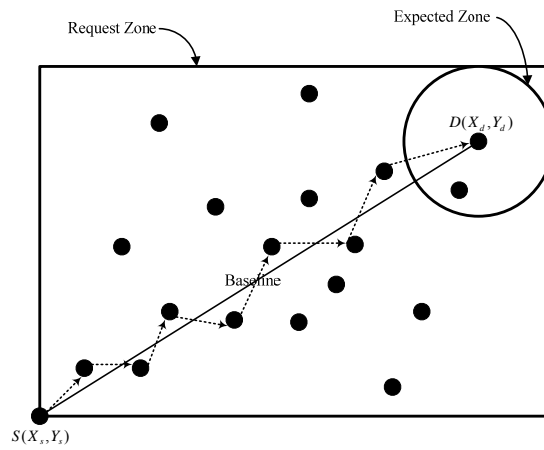


圖 16、鋸齒狀傳輸路徑的問題

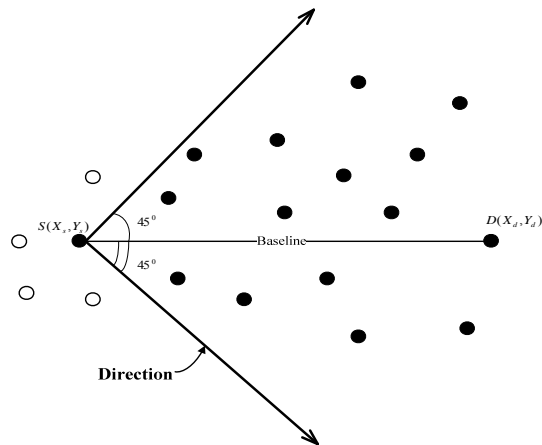


圖 17、以 Direction 方位取代 Request Zone

## 第三章 研究方法

本章節將詳細描述本文所提出的路由協定 HSLAR 的運作過程，並舉出可能產生問題的實例說明。

### 3.1 基本環境介紹

路由協定中，一般可以分成兩大類型，一種是 Table-driven 繞徑協定，另一種是 On-demand 繞徑協定。Improved Location-Aided Routing Protocol 是屬於 On-demand 繞徑協定中，改善過去 Location-Aided Routing Protocol 的方法，但是我們發現 ILAR 協定隱藏著四個問題，就是當網路裡節點的數量增加時，路徑上的 Hop Count 數量可能會隨著增加；第二個問題為節點移動性 (Mobility) 可能會導致路徑的可靠度降低的問題。第三個問題是隨著 RREQ 封包的轉送，最短的直線距離 (Baseline) 也跟著不斷改變，所以 Baseline 也必須跟著重新計算。最後，LAR 協定所使用的 Request Zone 的用途，在本研究中已經大幅度降低其可用性，所以我們將以方位 (Direction) 的概念將之取代。我們所提出的 HSLAR 協定可以有效的降低 Hop Count 的數量，以及提高路徑的可靠度，進而提升整體網路資源運用的效能。為了瞭解我們的 HSLAR 協定，首先將再次簡潔的介紹 LAR 與 ILAR 協定的運作模式。

我們先簡單介紹既有 LAR 協定，以圖 6、7 為例，所謂 LAR 是使用一個矩形的廣播區域，稱之為 Request Zone，此 Request Zone 必須包含來源節點 S 及目的地節點 D 的 Expected Zone；假設來源節點 S 已知目的地節點 D 的座標位置及移動速率，可推測出目的地節點 D 的座標位置為  $D(X_d, Y_d)$ ，以及它的移動半徑  $R = V(t_1 - t_0)$ ，這個

半徑所形成的 Expected Zone。來源節點 S 廣播的 RREQ 封包只有落在 Request Zone 內的節點才會繼續進行傳遞。這樣的方式可以有效減少因氾濫式廣播而可能產生的廣播風暴問題。

而所謂的 ILAR 協定是依據 LAR 協定可能發生的問題加以改善；在這改善的方法中，首先透過一條連接來源節點 S 及目的地節點 D 的 Baseline 來找尋路徑，而任意一個節點距離來源節點 S 的距離稱為 DIST，而節點距離這條基準線的距離稱為 VDIST，比較某個節點的鄰居節點的 VDIST，且選出最小 VDIST 的節點，選做為下一個繼續傳遞 RREQ 封包的轉送節點。這樣的方式更可以減少在 Request Zone 裡不必要的廣播封包。

但是顯然的 ILAR 協定不能滿足我們所發現的問題，所以經由我們所提出的 HSLAR 協定可以有效的解決上述四個問題。在正式進入 HSLAR 協定的 Route Discovery 程序前，我們先描述所使用的封包格式及內容介紹。

## 3.2 封包格式描述

當來源節點 S(或轉送節點)想要建立到目的地節點 D 的路徑時，首先它會發送 Route Request Transmit (RREQ\_T) 的封包給自己的鄰近節點，它的格式請參照表 2。Type 欄位可顯示出封包的類型。Broadcast ID 指的是此廣播的編號，它是一個連續的號碼，而且是唯一的。Sending ID 指的是發送 RREQ\_T 封包的節點，它可能是來源節點 S 或路徑上的任意一個轉送節點。Sending location information 則是記錄發送 RREQ\_T 封包的節點的座標位置資訊。Source ID 欄位記錄著來源節點 S 的編號。Source Location Information 欄位記錄的是來源

節點 S 的座標位置的資訊，以及 Destination Location Information 欄位記錄的是目的地節點 D 的座標位置的資訊，經由這兩個欄位的資訊，可以求得某節點 i 是否與目的地節點 D 落在相同的 Direction 方位。所有接收到 RREQ\_T 封包的鄰近節點，必須與目的地節點 D 落在相同的 Direction 方位，而且在來源節點 S（或轉送節點）之發送功率為 90% 之內才能成為 Candidate Node，並且回覆 Route Request Revise（RREQ\_R）的封包給 Sending Node；RREQ\_R 封包的格式請參照表 3 所示。

表 2、Route Request Transmit（RREQ\_T）封包格式

1	Type
2	Broadcast ID
3	Sending ID
4	Sending Location Information
5	Source Location Information
6	Destination Location Information

表 3、Route Request Revise（RREQ\_R）封包格式

1	Type
2	My ID
3	My Location Information
4	DTS

RREQ\_R 封包裡，My ID 欄位記載節點自己的 ID 編號，提供給來源節點 S（或轉送節點）識別用。而 My Location Information 欄位

將被來源節點 S（或轉送節點）使用在選擇下一個轉送節點（Downstream Node）的用途上，會計算出此 Candidate Node 到 Baseline 的垂直距離的長度值，這個值將被用在權重值公式上。DTS 欄位記錄此節點到來源節點 S 的距離長度值，此值也將被使用在權重值公式上。

Route Request（RREQ）封包是由來源節點 S 向目的地節點 D 發送的控制封包；它的格式請參照表 4。Type 欄位可顯示出封包的類型。Broadcast ID 指的是此廣播的編號，它是一個連續的號碼，而且是唯一的。Source ID 以及 Destination ID 欄位分別記錄著來源節點 S 和目的地節點 D 的編號；Source Location Information 和 Destination Location Information 欄位記錄的是來源節點 S 和目的地節點 D 的座標位置的資訊；Route Record 欄位記錄著傳遞此 RREQ 封包的轉送節點。Hop Count 欄位會記錄路徑上經由跳躍幾個轉送節點來到達目的地節點 D。

表 4、Route Request (RREQ) 封包格式

1	Type
2	Broadcast ID
3	Source ID
4	Source Location Information
5	Destination ID
6	Destination Location Information
7	Route Record
8	Hop Count

如果目的地節點 D 成功的接收到來自來源節點 S 的 RREQ 封包時，目的地節點 D 會回傳 Route Reply (RREP) 封包給來源節點 S，此時目的地節點 D 會藉由傳送過來的 RREQ 封包裡的 Route Record 欄位將路徑反向並回傳 RREP 封包給來源節點 S，並確保路徑沒有斷裂。RREP 封包格式請參照表 5。目的地節點 D 會將自己目前的座標位置記錄在 Destination Location Information 欄位，並且記錄目前的時間在 Current Time 欄位。

表 5、Route Reply (RREP) 封包格式

1	Type
2	Source ID
3	Destination ID
4	Destination Location Information
5	Current Time

當此路徑有任意一個轉送節點找不到適合的節點做為 Downstream Node，以及路徑斷裂時，皆會回傳 Route Error 封包給前一個發送 RREQ 封包的轉送節點，又可以稱此節點為 Upstream Node；並且由 Upstream Node 重新找尋新的替代傳輸路徑。RERR 封包格式如表 6。Error node ID 欄位記錄的是發現路徑斷裂的轉送節點的編號，並且由它來發出 RERR 封包給 Upstream Node。Upstream ID 欄位存放的是前一個跳躍的轉送節點，即 Upstream Node 的編號，而此 RERR 封包就是指名給此欄位內編號的轉送節點。

表 6、Route Error (RERR) 封包格式

1	Type
2	Error node ID
3	Source ID
4	Destination ID
5	Upstream ID

### 3.3 尋找路徑階段

本篇主要研究的方向，是針對我們所發現 ILAR 協定的問題來做改善，經由我們所提出的 HSLAR 協定將可以有效的解決路徑的不穩定和 Hop Count 的問題。

一開始當來源節點 S 想要傳送資料封包給目的地節點 D 時，先會查看自己的路徑表格裡是否到達目的地節點 D 的可用路徑，如果傳輸路徑太舊或者是沒有傳輸路徑可用時，來源節點 S 會啟動 Route Discovery 程序，發出 RREQ 封包來建立傳輸路徑。假設來源節點 S 已知目的地節點 D 的座標資訊，可計算出目的地節點 D 所在的 Direction；在此我們將以 Baseline 為基準，左右各展開 45 度角的方向來尋找路徑，如圖 17 所示，只有落在定義出的 Direction 內的實心節點才會幫忙轉送封包，落在 Direction 外的空心節點則不會回應。而不同的目的地節點 D 會形成不同的 Direction，以圖 18 為例，目的地節點  $D_1$  所定義出的 Direction1 (實線) 與目的地節點  $D_2$  所定義出的 Direction2 (虛線) 是不相同的；我們以 Direction 來取代 Request Zone 不僅可以省去紀錄在 RREQ 封包內的欄位，也可讓在 Request Zone 以外可能存在的傳輸路徑能夠被建立。並且再建立出一條  $Baseline_{s,d}$

來做為找尋路徑的用途。

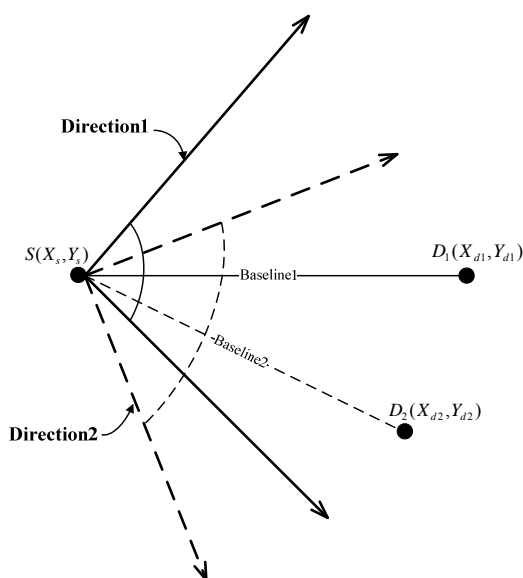


圖 18、不同目的地節點所定義出不同 Direction 的圖示

之後，來源節點 S 會向它自己的鄰近節點發送 RREQ\_T 封包，所有與目的地節點 D 位在相同 Direction，並且在來源節點 S 發送功率為 90% 內的鄰近節點；這些節點又稱為 Candidate Node。當 Candidate Node 接收到來自來源節點 S 的 RREQ\_T 封包時，會將自己的 ID、座標資訊和 DTS 記錄在 RREQ\_R 封包內，並回傳給來源節點 S。若是選擇在來源節點 S 發送功率為 90% 內的節點為 Candidate Node，可以期望降低因為節點移動而造成路徑不穩定的情形發生。

接著，來源節點 S 會從這些回傳 RREQ\_R 封包的 Candidate Node 中，經由權重值公式： $W = (DTS_i - DTB_i) \cdot (4)$ ；DTS (Distance To Source Node) 代表 Candidate Node i 到來源節點 S 的距離長度值；DTB (Distance To Baseline) 表示 Candidate Node i 到 Baseline 的垂直距



離。選出權重值 ( $W$ ) 最大的節點作為轉送節點；因為我們期望路徑的 Hop Count 數最少並且接近最短的直線距離 (Baseline)，因此 DTS 值需要越大，而且不超過發送端之發送功率為 90% 之外越好，而 DTB 值需要越小，代表節點越靠近  $Baseline_{s,d}$  則越接近最短的直線距離；來源節點將傳送 RREQ 封包給  $W$  值最大的轉送節點。同樣地轉送節點也執行相似的程序，向自己的鄰近節點發送 RREQ\_T 的封包，所有比轉送節點 (在此又可稱為 Upstream Node) 距離來源節點 S 更遠且在發送端之發送功率為 90% 內的鄰近節點，即所謂 DTS 值比 Upstream Node 的 DTS 值大的鄰近節點皆為 Candidate Node，只有 Candidate Node 才會回傳 RREQ\_R 的封包給 Upstream Node；在本文提出的 HSLAR 協定中，轉送節點會重新計算一條  $Baseline_{i,d}$ ，建立於自己與目的地節點 D 之間的直線；同樣的，經由權重值公式選出  $W$  值最大的 Candidate Node 作為在下一個傳遞 RREQ 封包的轉送節點 (在此又可稱為 Downstream Node)，並將 RREQ 封包傳送給 Downstream Node。

依此類推，直到目的地節點 D 成功收到來自來源節點 S 的 RREQ 封包時，目的地節點 D 則會回傳 RREP 的封包給來源節點 S，藉此建置出一條穩定且快速的傳輸路徑。接下來我們以實際例子來說明 ILAR 與 HSLAR 協定之間的差別。

圖 19 和圖 20 說明 ILAR 與 HSLAR 協定在相同的網路環境下所不同的運作模式。圖 19 中，虛線的箭頭線表示 ILAR 協定所選出來的傳輸路徑，而圖 20 中，實線的箭頭線則表示我們的 HSLAR 協定所找尋出來的傳輸路徑。ILAR 協定由於只考慮節點的 VDIST 值，而

節點 A 和節點 B 都在來源節點的廣播範圍內，即使可能選擇節點 B 會比選擇節點 A 來的更好，但是 ILAR 協定仍會選擇節點 A 做為轉送節點。然後再由節點 A 執行相同的 Route Discovery 程序。因此，在圖 19 裡經由 ILAR 協定的運作，來源節點 S 到達目的地節點 D 的傳輸路徑為  $V_s \rightarrow V_a \rightarrow V_c \rightarrow V_e \rightarrow V_f \rightarrow V_g \rightarrow V_h \rightarrow V_d$ 。圖 20 中，不同於 ILAR 的 HSLAR 協定，考慮權重值比較大的節點，所以來源節點 S 會選擇較好的節點 B 做為下一個廣播 RREQ 的轉送節點。同樣的，再由節點 B 執行前述的 Route Discovery 程序，直到抵達目的地節點 D。透過 HSLAR 協定的方式，來源節點 S 到達目的地節點 D 的傳輸路徑為  $V_s \rightarrow V_b \rightarrow V_e \rightarrow V_f \rightarrow V_h \rightarrow V_d$ 。對照 ILAR 與 HSLAR 協定所建立的傳輸路徑，我們可以從例子中很明確的指出 HSLAR 比 ILAR 協定減少了 2 個 Hop Count，而我們的方法也比 ILAR 協定的傳輸路徑較為平滑，降低路徑在 Baseline 附近鋸齒狀的移動，可使得傳輸路徑有更佳的可可用性。

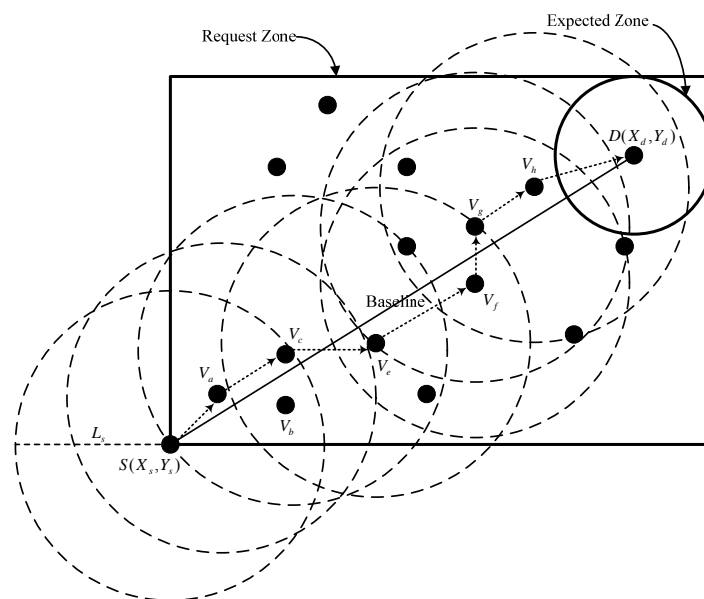


圖 19、ILAR 協定建立的傳輸路徑

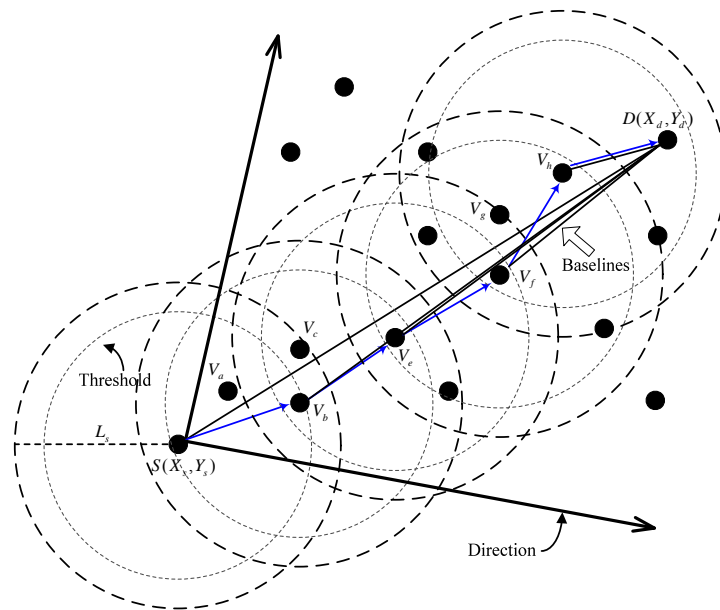


圖 20、HSLAR 協定建立的傳輸路徑

### 3.4 路徑維護階段

由於在 MANET 的網路環境下，每一個節點都具有可移動性，當節點的移動速率越高時，所建置出來的傳輸路徑就越可能斷裂；因此，我們還需要一個維護的機制（Route Maintain）來確保資料的傳輸率。在傳輸路徑的維護方式，大致上可分為兩種，一種是直接傳送 RERR 封包回到來源節點 S，RERR 封包經過的節點則會依序將有關這條路徑的資訊清除；當 RERR 封包回到來源節點之後，才由來源節點 S 重新建立一條到達目的地節點 D 的新傳輸路徑。如圖 21，當節點 C 發現與節點 D 的路徑斷裂，節點 C 會發送 RERR 封包回到來源節點 A，並由節點 A 重新發送 RREQ 封包來找尋新的傳輸路徑。另一種稱為 Local Repair 模式的作法是，由斷裂的節點發出 RREQ 封包來重新建立新的傳輸路徑，此種方式可以更快速的來接續資料封包的傳送。本研究則是使用第二種 Local Repair 的維護機制，如圖 22 所示。

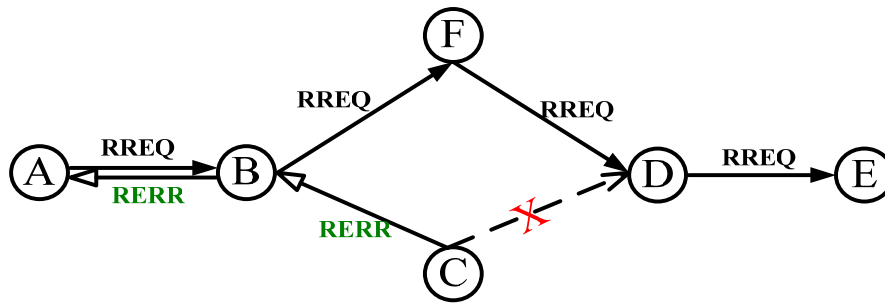


圖 21、Full Repair 的維護機制

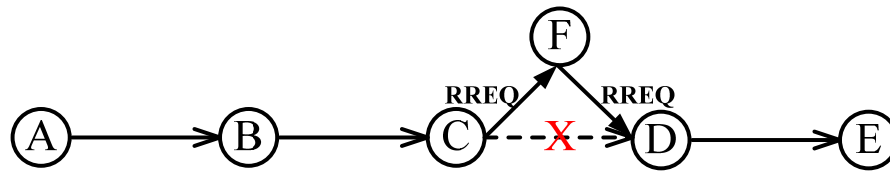


圖 22、Local Repair 的維護機制

傳輸路徑維護的運作步驟如下：

- 我們假設當路徑上的一個轉送節點  $F_i$  發現與 Downstream Node 的連結中斷時，此時  $F_i$  會啟動傳輸路徑維護機制，重新執行 Route Discovery 程序來找尋其他新的替代傳輸路徑。
- 若是  $F_i$  無法找到適合的 Downstream Node，則會以 unicast 的方式發送 RERR 封包給 Upstream Node，由它來重新尋找新的替代傳輸路徑。依此類推，若再找不到則繼續返回 Upstream Node 來尋找替代傳輸路徑。

如此一來可以在傳輸路徑斷裂時，不用重回到來源節點 S 執行 Route Discovery 程序，而且能快速的重建傳輸路徑。

圖 23 裡顯示傳輸路徑斷裂的例子。在這案例裡，Route Discovery 程序應該是照著藍色的線往 Destination Node 傳送封包，而虛線則表示原本應該要傳送的路徑。但是在節點 F 要傳送封包給節點 H 時，發現與節點 H 的 Link 斷裂，此時表示此傳輸路徑已經發生錯誤，所以必須啟動維護機制。我們的 HSLAR 協定採用的傳輸路徑維護方法為 Local Repair 模式。所以當節點 F 發現路徑斷裂時，自己會先執行 Route Discovery 程序，找尋其它適合的傳輸路徑。如果沒有找到可用的傳輸路徑，再廣播出 RERR 封包給 Upstream Node，並由它重新執行 Route Discovery 程序。但是此程序將不會考量已斷掉的這條路徑。如果此節點仍然找不到可用的替代路徑，則依相同的程序繼續通知再上一個 Upstream Node，再由它負責 Route Discovery 程序的執行。如此一來有很大的機會不必回到原來的來源節點 S 就可以找到另一條可用的傳輸路徑，有效降低在傳輸路徑維護時所耗費的網路資源。

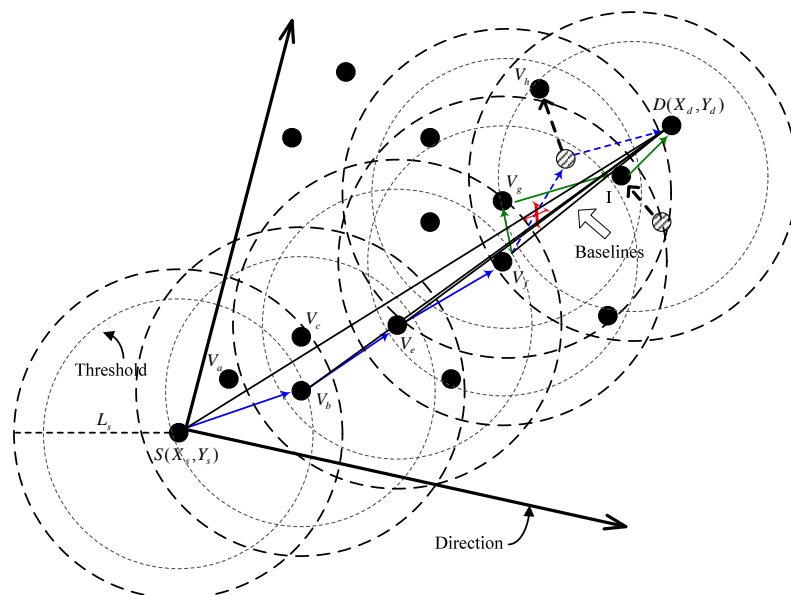


圖 23、Local Repair 模式的圖例

## 第四章 模擬實驗

在這個章節，我們將針對 ILAR 協定以及本文所提出的 HSLAR 協定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 GeIrc39。

### 4.1 模擬環境

在模擬實驗的部份我們將與 ILAR 協定做比較，所有的環境參數設定皆與 ILAR 協定相同；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；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，每一個節點在移動一段距離後，會依據移動的方向以及距離來重新定位自己的座標值，並且假設每一個網路上的節點皆可透過 GPS 定位系統、Sensor Device 等方式，來取得在此網路上其他節點的座標資訊，透過座標位置的輔助來完成繞徑協定的運作，所有參數如表 7 所示。所有進行分析的每項參數皆進行 600 秒，共執行 10 次後計算其平均值，以下將對每張模擬結果的折線圖表來做分析說明。

表 7、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)
Pause time	1 sec
Communication range	100 m
Threshold of range	90 %
Number of data packets	10
Simulation time	600 sec

## 4.2 模擬結果

在這個小節裡，我們將透過實驗的結果來評估 ILAR 與 HSLAR 協定效能的差異。我們假設網路上每一個的座標資訊都可以正確的取得，並且分別以不同的節點數量和移動速率的變化，來觀察兩種協定在路徑上的平均 Hop 數量、封包的平均發送量以及路徑的失敗率上的差異。

圖 24 和表 8 裡，顯示了在 50 個節點但不同的移動速率中，每一條成功建立的路徑上的平均 Hop Count，在本文中所提出的 HSLAR 協定中，透過公式 (3) 來挑選轉送節點，能有效的減少路徑上的 Hop Count。而 ILAR 協定在選擇路徑上，則是依循著 Baseline 來建立，所以可能造成多點的跳躍才能到達目的地節點 D，無法快速的傳遞封包。即使移動速度增加，HSLAR 協定所建立的路徑仍然可以維持在穩定的 Hop Count，並且比 ILAR 協定平均減少約 18.06% 的 Hop Count。

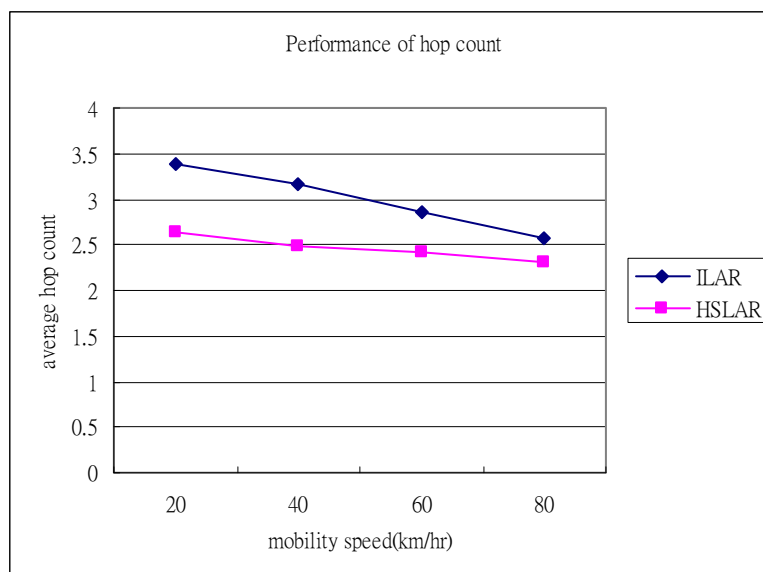


圖 24、50 個節點下平均 Hop Count 之效能分析

表 8、50 個節點下平均 Hop Count 之效能分析

Mobility speed (km/hr)	Average Hop Count		Improve
	ILAR	HSLAR	
20	3.392	2.633	22.38%
40	3.173	2.484	21.71%
60	2.861	2.418	15.48%
80	2.576	2.299	10.75%
<b>Average</b>	3.001	2.459	18.06%

圖 25 和表 9 顯示了在移動速率 40km/hr 下不同節點的數量，ILAR 協定會因為節點數量的增長而增加路徑的，因為 ILAR 協定在選擇路徑時，只考慮了 VDIST 的因素，所以無法縮短路徑上的 Hop Count。而本文所提出的 HSLAR 協定，不僅是考慮 DTB 還加入了 DTS 的因



素，所以在 HSLAR 協定下，所建立的路徑可以 ILAR 協定約減少 28.83% 的 Hop Count，當節點數越多時差距越大，表示 HSLAR 協定在網路密度越高的狀態下效能越好。

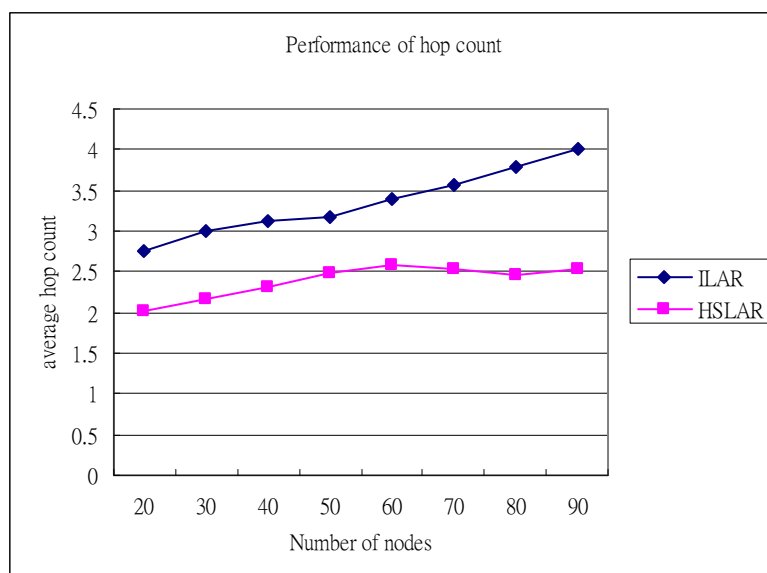


圖 25、移動速率 40 km/hr 下平均 Hop Count 之效能分析

表 9、移動速率 40 km/hr 下平均 Hop Count 之效能分析

Number of nodes	Average Hop Count		Improve
	ILAR	HSLAR	
20	2.755	2.016	26.82%
30	3.012	2.166	28.09%
40	3.131	2.311	26.19%
50	3.173	2.484	21.71%
60	3.397	2.592	23.70%
70	3.573	2.538	28.97%

<b>80</b>	3.781	2.462	34.88%
<b>90</b>	4.013	2.525	37.08%
<b>Average</b>	3.354	2.387	28.83%

圖 26 和表 10 裡的模擬結果，顯示了在 50 個節點但不同的移動速率，在 Route Discovery 階段以及資料封包的傳輸過程，所發送的平均封包數量（或稱網路的負載量），此模擬實驗結果與平均 Hop Count 有關，原因是當路徑的轉送節點越多時所需要發送的封包數量就會越多；所以我們所提出的 HSLAR 協定能比 ILAR 協定有效的減少 Hop Count，相對的也能降低封包 16.13% 的使用量。低負載量下的網路，能讓網路資源做其他更有重要的運用。

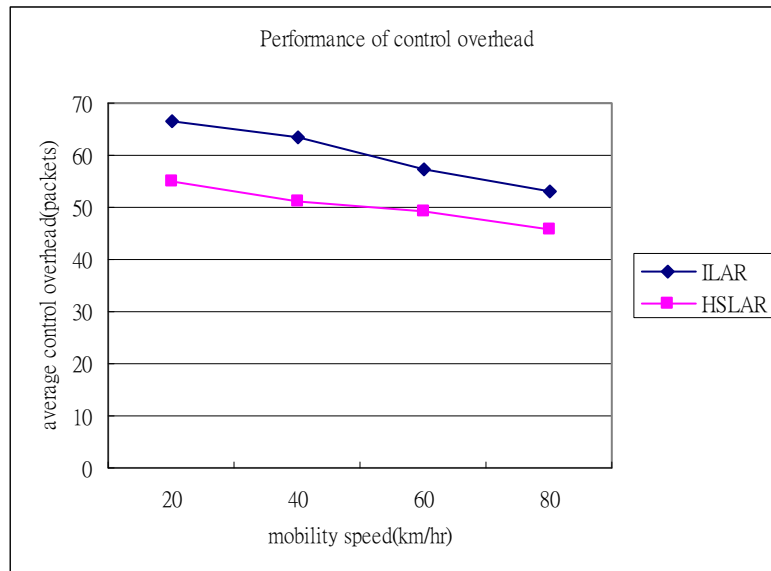


圖 26、50 個節點下平均控制封包的發送量之效能分析

表 10、50 個節點下平均控制封包的發送量之效能分析

Mobility speed (km/hr)	Average control overhead		Improve
	ILAR	HSLAR	
20	66.388	54.819	17.43%
40	63.281	51.213	19.07%
60	57.348	49.411	13.84%
80	52.918	45.785	13.48%
Average	59.984	50.307	16.13%

在移動速率 40km/hr 下不同節點的數量，由圖 27 和表 11 來看平均封包的發送數量，如前述所言，ILAR 協定隨著節點數量的增加，會導致路徑的 Hop Count 增多，相對地也會產生高負載量的網路。然而，我們的 HSLAR 協定可以有效的控制路徑上的 Hop Count，因此，即使在節點增加（也可以說是高密度網路）的情況，可以比 ILAR 協定減少約 27.65%的封包數量，使網路的負載量能維持在比較穩定的狀態之下。

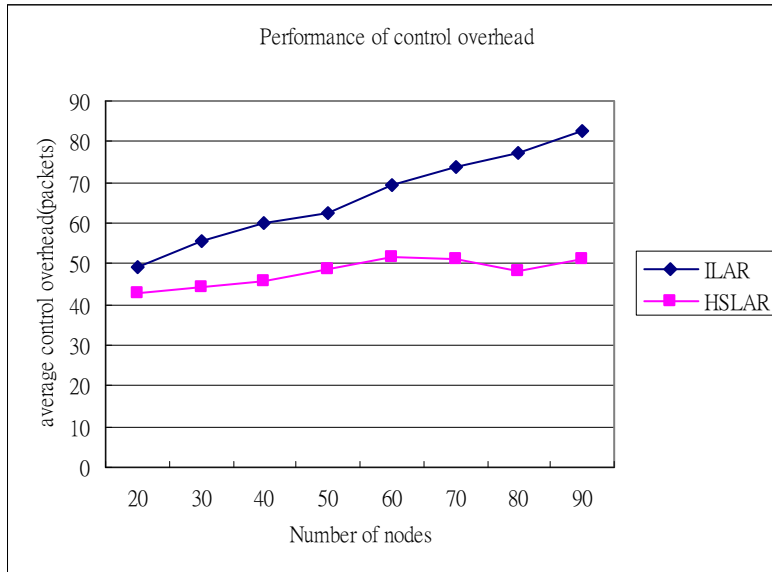


圖 27、移動速率 40 km/hr 下平均控制封包的發送量之效能分析

表 11、移動速率 40 km/hr 下平均控制封包的發送量之效能分析

Number of nodes	Average control overhead		Improve
	ILAR	HSLAR	
20	49.136	42.979	12.53%
30	55.682	44.457	20.16%
40	59.956	45.957	23.35%
50	62.392	48.545	22.19%
60	69.547	51.431	26.05%
70	73.646	51.046	30.69%
80	77.337	48.218	37.65%
90	82.826	51.193	38.19%
<b>Average</b>	66.315	47.978	27.65%

圖 28 和表 12 顯示了在 50 個節點但不同的移動速率的網路環境下，傳輸路徑平均斷裂的機率。在 Route Discovery 階段中，ILAR 協定與 HSLAR 協定同樣會在發送功率為 Threshold 內選擇 Downstream Node，避免路徑是透過 Border Node 來進行傳送。即使 ILAR 協定所建立的傳輸路徑 Hop Count 會高於 HSLAR 協定，但是不見得會使傳輸路徑較為穩定不容易斷裂，例如，鋸齒狀路徑上的轉送節點，分別往反向的兩邊移動時，傳輸路徑斷裂的機率就會提高。然而，HSLAR 協定經由每次選擇 Downstream Node 時，會重新建立 Baseline 作為傳輸路徑建立的參考，因此建立出來的傳輸路徑比較平滑，相對的也會比較穩定。所以在高移動速率下，HSLAR 協定所建立的路徑會比較不容易斷裂。由下列圖表顯示，雖然在低移動速率 HSLAR 協定傳輸路徑平均斷裂的機率略高於 ILAR 協定，但是在高移動速率(80km/hr)下，HSLAR 協定傳輸路徑平均斷裂的機率反而比 ILAR 協定約降低 12.3%，所以在高速移動下 HSLAR 協定所建立的傳輸路徑能有比較好的效能。

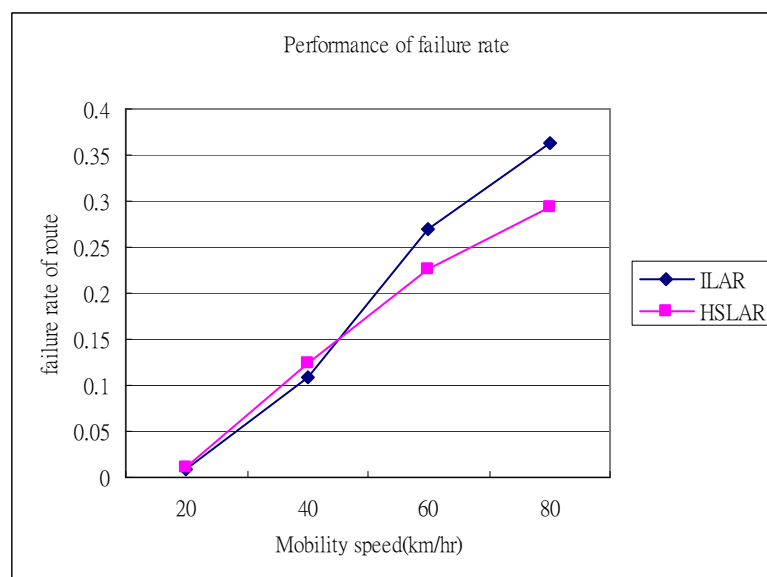


圖 28、50 個節點下傳輸路徑平均的斷裂比率之效能分析

表 12、50 個節點下傳輸路徑平均的斷裂比率之效能分析

Mobility speed (km/hr)	Failure rate of route		Improve
	ILAR	HSLAR	
20	0.009	0.011	-22.22%
40	0.109	0.124	-13.76%
60	0.269	0.226	15.99%
80	0.362	0.293	19.06%
Average	0.187	0.164	12.30%

圖 29 和表 13 顯示了在移動速率 40km/hr 下不同節點數量的網路環境下，傳輸路徑平均斷裂的機率。因為 ILAR 協定所建立的傳輸路徑容易產生鋸齒狀的情況發生，或者是傳輸路徑上每一個之間的傳輸距離不一，當選擇了傳輸距離較遠的 Downstream Node 時，發生傳輸路徑斷裂的機率就會提高；然而，HSLAR 協定透過每條 Baselines 以及權重值的公式，能建立起平滑且穩定的傳輸路徑。經由下列圖表的數據顯示，HSLAR 協定在越多節點數量（80-90 個節點）的網路環境下，能夠比 ILAR 協定降低傳輸路徑約 27.68%至 50%斷裂的機率。

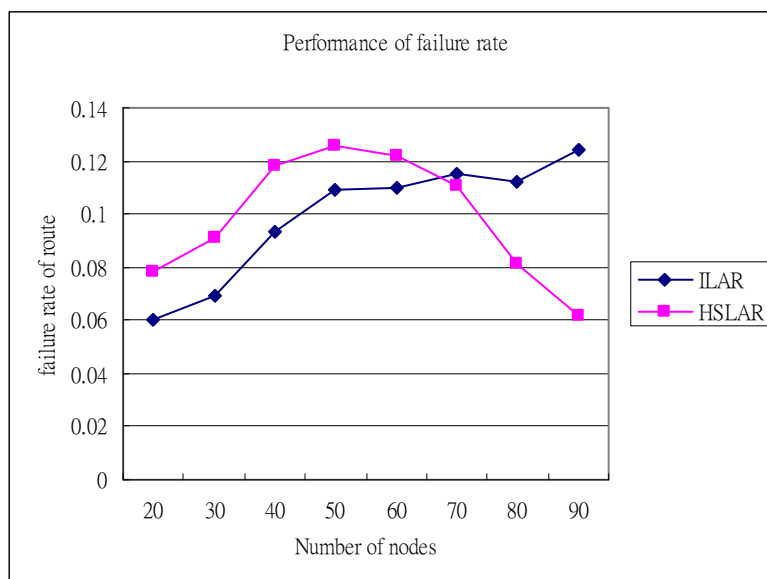


圖 29、移動速率 40 km/hr 下傳輸路徑平均的斷裂比率之效能分析

表 13、移動速率 40 km/hr 下傳輸路徑平均的斷裂比率之效能分析

Number of nodes	Failure rate of route		Improve
	ILAR	HSLAR	
20	0.060	0.078	-30.00%
30	0.069	0.091	-31.88%
40	0.093	0.118	-26.88%
50	0.109	0.126	-15.60%
60	0.110	0.122	-10.91%
70	0.115	0.111	3.48%
80	0.112	0.081	27.68%
90	0.124	0.062	50.00%
<b>Average</b>	0.099	0.098	1.01%

## 第五章 結論

本文我們提出了一個以位置輔助的 HSLAR 協定，它是以 LAR 協定為基礎，改善 ILAR 協定在 Route Discovery 階段上的不足。HSLAR 協定能配合已知的座標位置來建立 Baselines，也因為座標資訊的關係，可以控制 RREQ 封包著正確的方向傳遞來找尋路徑，減少在 Route Discovery 階段時所產生的網路負載量。

透過模擬實驗的結果，也證明了我們提出的 HSLAR 協定在路徑的建置上，在移動速度改變以及節點數量改變，分別可以找到 Hop Count 約少 18.06%和 28.83%的傳輸路徑，並且有效的降低封包 16.13%和 27.65%的發送數量，以及減少傳輸路徑約 12.30%和 1.01%斷裂的機率，所以，本篇研究所提出的 HSLAR 協定確實可以比 ILAR 協定有更好的效能。



## 參考文獻

- [1] B.N. Amotz, N. Joseph and S. Baruch, “Pushing dependent data in clients-providers-servers systems”, IEEE/ACM Wireless Networks (2003), pp. 421-430.
- [2] D. Aksoy and M.S.F. Leung, “Pull vs Push: A Quantitative Comparison for Data Broadcast”, IEEE/GLOBECOM’04 (2004), pp.1464-1468.
- [3] L. Barolli, Y. Honma, A. Koyama, A. Durresi and J. Arai, “A Selective Border-casting Zone Routing Protocol for Ad-hoc Networks”, IEEE/DEXA’04 (2004), pp. 326-330.
- [4] H.L. Chen and C.H. Lee, “Two Hops Backup Routing Protocol in Mobile Ad Hoc Networks”, IEEE/ICPADS’05 (2005), pp. 600-604.
- [5] J. Cai, T. Terada, T. Hara, and S. NISHIO, “On a Cooperation of Broadcast Scheduling and Base Station Caching in the Hybrid Wireless Broadcast Environment”, IEEE/ACM MDM’06 (2006), pp. 72.
- [6] L. Cheng, A.G. Bourgeois and B.H. Yu, “Power Management in Wireless Ad Hoc Networks using AODV”, SNPD-SAWN’05 (2005), pp. 436-443.
- [7] M.S. Corson and V.D. Park, “A Performance Comparison of the Temporally-Ordered Routing Algorithm and Ideal Link-State Routing”, IEEE/ISCC (1997), pp. 592.
- [8] T.C. Chiang, P.Y. Wu and Y.M. Huang, “A limited flooding scheme for mobile ad hoc networks”, IEEE/WiMob2005 (2005), pp. 473-478.
- [9] Y.C. Hu, D.B. Johnson and A. Perrig, “SEAD: Secure Efficient Distance Vector Routing for MobileWireless Ad Hoc Networks”, Elsevier B.V. /Ad hoc Networks (2003), pp. 175-192.

- [10] D.B. Johnson, and D.A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks", IEEE/Transactions on Mobile Computing (1996), pp.153-181.
- [11] B. Karp and H.T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks", IEEE/ACM MobiCom (2000), pp. 63-72.
- [12] Y.B. Ko and N.H. Vaidya, "Location-Aided Routing(LAR) in mobile ad hoc networks", Wireless Networks 6 (2000), pp. 307-321.
- [13] H. Nakagawa, K. Ishida, T. Ohta, and Y. Kakuda, "GOLI: Greedy On-Demand Routing Scheme Using Location Information", IEEE/ICDCSW'06 (2006), pp. 1-1.
- [14] W. Ni, Q. Fang and S.V. Vrbsky, "A Lazy Data Request Approach for On-demand Data Broadcasting", IEEE/ICDCSW'03 (2003), pp. 790-796.
- [15] C.E. Perkins and P. Bhagwat, "Highly Dynamic Destination Sequenced Distance Vector Routing (DSDV) for Mobile Computers", IEEE/ACM Applications, Technologies, Architectures, and Protocols for Computer Communication (1994), pp. 234-244.
- [16] R. Purtoosi, H. Taheri, A. Mohammadi and F. Foroozan, "A High Performance Cluster-Based Flooding Algorithm for Wireless Ad Hoc Networks", IEEE/CIT'05 (2005), pp. 417-423.
- [17] S. Ramakrishnan<sup>1</sup>, T. Thyagarajan<sup>2</sup>, P.K. Ram<sup>1</sup> and R. Vinodhl, "Design and Analysis of PADSAR Protocol for Routing in MANETs", IEEE/INDICON'05 (2005), pp. 193-197.
- [18] X. Ren and H. Yu, "Multipath Disjoint Routing algorithm for Ad Hoc Wireless Sensor Networks", IEEE/ISORC'05 (2005), pp. 253-256.
- [19] C.R. Shiung, C.W. Yeh, and W.Y. Fu, "Hybrid Wireless Network Protocols", IEEE/TVT (2003), pp. 1099-1109.
- [20] M. Sheng, J. Li and Y. Shi, "Relative Degree Adaptive Flooding Broadcast Algorithm for Ad Hoc Networks", IEEE/Transactions on

- Broadcasting (2005), pp. 216-222.
- [21]N. Saxena, K. Basu and S.K. Das, “Design and Performance Analysis of a Dynamic Hybrid Scheduling Algorithm for Heterogeneous Asymmetric Environments”, IEEE/IPDPS’04 (2004), pp. 223.
- [22]R. Sakellariou and H. Zhao, “A Hybrid Heuristic for DAG Scheduling on Heterogeneous Systems”, IEEE/IPDPS’04 (2004), pp. 111.
- [23]Y. Sakurai and J. Katto, “AODV Multipath Extension using Source Route Lists with Optimized Route Establishment”, IEEE/IWWAN’04 (2004), pp. 63-67.
- [24]M. Tauchi and T. Ideguchi and T. Okuda, “Ad-hoc Routing Protocol Avoiding Route Breaks Based on AODV”, IEEE/HICSS (2005), pp. 322a-322a.
- [25]S. Tang and B. Zhang, “A Robust AODV Protocol with Local Update”, IEEE/10th dsia-pacific Conference on Communications and 5th International Symposium an Multi-Dimensional Mobile Communications (2004), pp. 418-422.
- [26]N.C. Wang and S.M. Wang, “An Efficient Location-Aided Routing Protocol for Mobile Ad Hoc Networks”, IEEE/ICPADS’05 (2005), pp. 335-341.
- [27]S.H. Wang, M.C. Chan, and T.C. Hou, “Zone-based Controlled Flooding in Mobile Ad Hoc Networks”, IEEE/WIRLES’05 (2005), pp. 421-426.
- [28]X. Wu and V.C.S. Lee, “Preemptive Maximum Stretch Optimization Scheduling for Wireless On-Demand Data Broadcast”, IEEE/IDEAS’04 (2004), pp. 413-418.
- [29]P. Yi, Z. Dai, Y. Zhong and S. Zhang, “Resisting Flooding Attacks in Ad Hoc Networks”, IEEE/ITCC’05 (2005), pp. 657-662.
- [30]Z. Yang, Q. Zhang, X. Du and L. Yuan. “Location-based adaptive ad

hoc routing (LAAR)”, IEEE/ISCIT’05 (2005), pp. 1013-1017.

[31]K. Zheng, N. Wang and A.F. Liu, “A new AODV based clustering routing protocol”, IEEE/WCNM’05 (2005), pp. 728-731.

[32]<http://140.114.32.181/summer00/12/17/2-2.htm> 畢氏定理