

# 隨意網路下支援多對多群播及廣播服務

林俊宏

國立中山大學資訊工程學系

[lin@cse.nsysu.edu.tw](mailto:lin@cse.nsysu.edu.tw)

夏銘君

國立中山大學資訊工程學系

[guu@wamis2.cse.nsysu.edu.tw](mailto:guu@wamis2.cse.nsysu.edu.tw)

## 摘要

在無線網路中，很多時候我們都必須使用到廣播或群播的機制；而一個最直接達成廣播之效果的方式，就是使用氾濫式廣播之方法 (flooding)。但是直接的使用氾濫式廣播之方法來作廣播或群播的動作時，通常都是非常的浪費成本而且會造成所謂的“廣播風暴 (broadcast storm)”之嚴重的問題。所以在這篇論文中，我們將提出一個利用“繞路訊息之交換”來建立轉送群組 (forwarding group) 的一種方法，並進而想要減少轉送群組中的成員，以減少廣播時，所造成之不必要的重覆廣播之影響。而這裡所謂的轉送群組就是一群網路上的節點，負責將收到的資料作轉送的動作。使用轉送群組的主要目的，是為了要利用一群網路上的節點來負責維護網路中所有節點的連通，即維護整個網路的連通性，以減少資料的重覆傳送。

由於我們的方法將儘可能地建立一個最小的轉送群組，因此如能利用這個轉送群組資料傳送與否的考量，我們將可以儘可能地的減少廣播或群播時所造成的額外傳送負載。而建立這個群組所採用的方式是找尋所有網路上的節點到根節點的最短路徑，並將這些路徑上的所有節點除了本身以外都選為選為轉送群組的成員。在這裡所提出的方法，主要是利用 Distance Vector 的繞路方法來作轉送群組的更新及維護；並且由於所利用的繞路協定是被用於動態改變的網路上，所以這個轉送群組的建立及維護方法，亦可以適應在有節點移動的網路上。

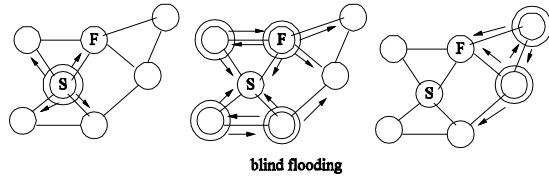
此外，由於所利用的繞路協定，可以防止資料傳送造成迴路，因此利用這個繞路協定，我們可以防止資料在轉送群組幫忙轉送資料時有迴路也發生。這個建立轉送群組的方法並不使用任何明顯及額外的控制封包來達成轉送群組的建立及維護，反而是利用繞路訊息在交換的同時達成轉送群組的建立及維護。所以並不會產生任何明顯負載。

**關鍵詞：**隨意無線網路、廣播、盲目的氾濫、轉送群組、支配集合

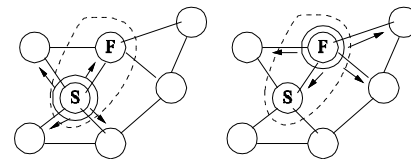
## 一、導論

### 1.1 研究動機

什麼是廣播的任務呢？由於在隨意的無線網路中 (ad hoc wireless networks)，沒有任何的基礎架構可以用來幫忙資料的轉送，因此每一個網路上的節點之運作方式不僅要運作為一般的主機 (host)，而且還必須運作成所謂的路由器 (router)。所以網路上的每一個節點除了能直接跟其所相鄰的節點溝通外，他們只可以幫其它節點轉送資料到那些節點想送達到的目的節點。在這樣的網路中，執行廣播的任務就是將一個訊息送達給網路上的



blind flooding  
圖 1：盲目的氾濫之廣播方式



Forwarding Group

圖 2：轉送群組的好處

所有節點，不論是距送出資料的節點之距離為一個跳躍距離 (one hop) 或是多重之跳躍距離 (multihop) 遠的節點都可以收到。而廣播的任務的執行方法，通常都是使用盲目的氾濫 (blind flooding) 之方式來達成。盲目的氾濫是一種廣播的策略，它是採用以下的特性來作為是否續傳廣播資料的依據：“當網路上的任一節點，在第一次收到廣播的資料時，必須負責將它再廣播 (rebroadcast) 出去”；這樣的作法將使得最後所有網路上的節點都將收到此一訊息，達成所謂的廣播。但是使用盲目的氾濫之方法會產生許多重覆的資料傳送，而重覆的資料傳送可能會產生很嚴重的問題，這個嚴重的問題就是所謂的廣播風暴[12]；廣播風暴會造成更多的網路競爭及碰撞。所以會使得網路的負載變大。

圖 1 為盲目的氾濫之廣播的方法；由圖 1 我們假設此時 S 節點為廣播之來源節點，當它發出廣播訊息時，它周遭的節點將會收到此訊息，且由於它們都是第一次收到這個封包，依氾濫之廣播的

定義，這些點將把這個訊息再廣播出去；接著這些點的週遭之節點亦會有作相同的動作，直到整個封包傳送至整個網路上。但是如我們所見的到，其實在圖 1 中，只有 S 及 F 都送過資料，就可以達到廣播到整個網路的效果了，如圖 2 所示。

到目前為止，在隨意的無線網路上，廣播給所有網路上的節點已經有很多的應用，例如在 DSR[7] 協定中的詢問訊息以及在移動式的隨意無線網路建立所謂的群播樹或群播網絡時所用到的控制封包，這種封包就像在 AODV[16] 協定中所使用到的 RREP 及 RREQ 兩種控制封包都是屬於在這個範疇內。而近年來有許多議題都在探討動態地可重新設定的隨意的無線網路中，想辦法在網路中找出一群節點來使網路上的所有節點互連，並使用一些特殊的演算法將這群節點中的成員儘量的減少，以減少廣播時可能會造成的負載，如一些支配集合 (Dominating Set, DS) 或連通之支配集合 (Connected Dominating Set, CDS) 以及找出近似的於最小的連通之支配集合 (Approximative Minimum Connected Dominating Set, Approximative MCDS) 的相關之研究。然而，在這些研究中，我們可以看到，那些目前被廣泛使用在靜態網路中的演算法，並無法在動態改變之隨意無線網路中有很好的效能表現。

所以我們想要提出一個方法，目的是想要讓所有網路上的節點，在任何的時間，都可以送出廣播的訊息；就像在有線的區域網路上之 ARP 以及 RARP 的機制一樣。所以我們想建立一個轉送群組，這個轉送群組是一群位在任何一個節點到根節點的最短路徑上之所有節點所構成，主要任務是負責廣播資料的轉送。而這個轉送群組的想法主要來自許多群播之轉送群組的觀念；所以為了要像在群播一樣，利用轉送群組把一份資料送給所有屬於同一群播群組中的人；並且為了要減少廣播所造成的負載，我們只利用這些轉送群組的內部節點來續傳廣播的資料。因此，在本篇文章中，我們將把廣播的機制想像是群播機制中的一個特殊例子；意即每一個在網路上的節點我們把他們想像成都是屬於同一個群播群組中之成員，並且想辦法來為這個群組建立一個轉送群組，以利資料的傳送。然而轉送群組的架構，可分為樹狀或是網狀的，根據很多的關於群播的相關研究[1] [3] [4] [6] [8] [9] [10] [11] [16]，我們知道樹狀的轉送群組是很脆弱且當網路的連通性改變時，還需要一再的被重新調整及維護；因此，我們選擇建立網狀的轉送群組。此外，就像其它建立連通之支配集合[2] [17] 的研究一樣，我們儘可能的減少轉送群組的大小，以期降低廣播時所產生之重覆的續傳之廣播資料所造成的網路負載。

總之，我們使用了來自群播中的轉送群組[3] 之觀念，來建立一個共用的網狀轉送群組，以供廣

播時續傳廣播資料來使用；並且試圖要最小化轉送群組成員的個數，意即將轉送群組的成員縮到最少。而由於我們所提出的方法，是利用現存之繞路機制，並無使用到任何明顯額外的控制訊息來建立轉送群組。所以並不會造成額外的負載。

## 1.2 論文架構

本論文的其它內容描述如下：在第 2 章中，我們將討論一些別人所做過之相關研究，及一些跟本文有關的觀念；第 3 章中，我們將詳細地描述我們所提出的方法之想法及一個簡單的例子；而在第 4 章中，顯示我們所做的一些評估及結果；最後，我們將在第五章為這篇文章作個總結。

## 二、相關研究

### 2.1 轉送群組的建立

一般而言，轉送群組的建立可以分為兩種方式；一種是由發送資料的節點所引發的方式，此種方式，我們稱為“由發送端啟動 (source-initiated)” 的協定[3] [8] [10] [16]；另一種則是由接收端所引發的方式，我們稱之為“由接收端啟動 (receiver-initiated)” 的協定[6]。在由發送端啟動之協定中，只有一個節點負責在一段固定的時間發送控制訊息，用以維護轉送群組。而在由接收端啟動之方法中，所有屬於群組中的節點，只要發現原來接收資料的路徑斷了，就會發出控制訊息，來尋找新的資料傳送路徑。不管是上述的那種方式，都可以建立出所謂的轉送群組；而這些被建立出來的轉送群組，我們又可以區分為二大類：一類網狀架構 (mesh based)[3] [6] [10] 的轉送群組；另一類則是樹狀架構 (tree based)[16] 的轉送群組。在網狀架構的轉送群組中，可能存在多於一條的路徑在任何一對的節點；因此它可以提供比樹狀架構更加穩固的連通性；反之，在樹狀架構中，任何節點間只可能存在唯一一條路徑，由於這樣的架構，使其在效率上較優於網絡架構。

此外，在隨意的無線網路中，由於每個節點都可以隨意的移動，使得網路的拓撲遭受頻繁的路徑斷裂。因此，我們需要一個用來維護轉送群組的方法；而在目前的研究中，這些方法又可以被區分為軟性狀態 (soft state)[3] [10] 及硬性狀態 (hard state)[16]。在軟性狀態的方法中，轉送群組的成員之資格將定期地被更新，而且常常需要利用週期性的控制封包來做更新的動作，所以軟性狀態是屬於一種主動的策略，因為它必須要在一定的時間送出控制封包，以維護轉送群組的連通性。而在硬性狀態的方法上，只有當節點間的鏈結斷裂時，才需要被更新及重新被設定，所以硬性狀態的方法是一種被動的策略，是屬於一種根據鏈結斷裂而作出相關反應的策略。

## 2.2 繞路協定

到目前為止有很多的繞路協定[1] [7] [15]被提出應用於隨意的無線網路；而在這些方法中，我們可以簡單的將它分為兩大類；分別為 table-driven 及 on-demand source-initiated 的方式。所謂的 table-driven 繞路方式，即是每個節點都本身維護著一些繞路的資訊並且在等定的週期更新這些繞路的資訊，是一種主動去更新繞路訊息的機制。而為了週期地更新繞路的資訊，每個節點都定期傳送控制的封包，這個更新的動作會造成網路頻寬的使用比較沒有效率。而在另一種 on-demand source-initiated 的繞路協定中，路徑只有當發送端有需求時，才去獲得；是屬於一種被動式的繞路協定。而這一類的協定通常都具備二個階段，第一個階段我們稱為“找尋路徑 (route-discovery)”，而另一個階段我們則稱為“資料傳送 (data-transfer)”。找尋路徑的階段，會在找到在找到一條或所有可能到達目的地的路徑停止。而找尋路徑所找到的路徑資料會由繞路機制的行程來維護，直到這個路徑斷裂或是資料傳送的階段已經完成，而不再會用到這個路徑的資訊。這些使用 on-demand source-initiated 的繞路協定，由於沒有週期的控制封包傳送，因此在頻寬的使用上，較為有效率，但是由於只有當有需要時才去獲得路徑，所以其連線所浪費的時間較多。

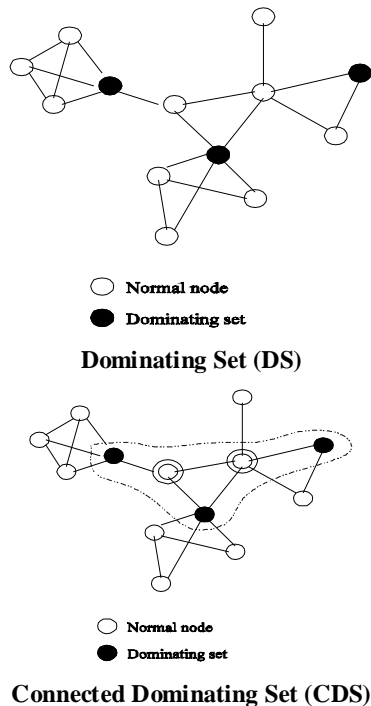


圖 3：DS與CDS

## 2.3 廣播

廣播的任務主要是想要使所有在網路中的節點都收到相同的一份訊息。而廣播的機置通常都是

被利用盲目的洪流 (blind flooding) 之廣播方式來完成。盲目的洪流是一種執行的策略；在這個策略下，所有節點不論何時收到第一個廣播的封包都有義務將這個廣播的封包再續傳給其週遭的人。通常盲目的洪流會造成許多重覆的資料傳送[13] [17]。而重覆的資料傳送可能會造成所謂的廣播風暴[2]這個嚴重的問題。而這個嚴重的問題就是重覆的資料傳送會造成更多的資料傳送通道的競爭及資料傳送的碰撞。

直到目前為止，有很多的方法被提出用來減少或消除這些由廣播所造成的重覆之資料傳送或是建立一群可以用來表示所有節點的最小集合；這些方法的核心想法就是：利用一群人來當所謂的內部節點，並利用它們來負責廣播訊息的續傳。比方說，建立一個最小的洪流樹 (flooding tree)，這個方法跟去找尋網路中最小的連通之支配集合 (Minimum Connected Dominating Set, MCDS)[17]是一樣的。而要知道所謂的最小的連通之支配集合，我們必需先了解支配集合的架構；一個所謂的支配集合，即是一個網路上節點的子集合，而每一個在網路中的節點不是跟這個子集合相鄰，就是屬於這個子集合中的節點，而這個集合內部之節點，並不一定要是連通的，如圖 3 左所示。若是這個集合內部之節點都是連通的，則為一個連通之支配集合，如圖 3 右所示。而所謂的找尋網路中最小的連通之支配集合就是去相辦法找到一個連通之支配集合，其大小是最小的。

## 實作及想法

### 3.1 想法及實作概觀

這篇論文的主要目的是設計一個可以支援在動態的隨意之無線網路下利用一個共用的轉送群組來輔助廣播或群播的執行以減少廣播或群播可能造成的網路負載。所以在我們所想像的網路中，每一個節點不是屬於轉送群組的成員，就是這些轉送群組中之成員的鄰居。就本質而言，這個轉送群組跟相關研究中所提到的連通之支配集合其實是相同的，因此在維護這個轉送群組上所作的努力，並實跟在跟那些研究之性質基本上是相同的。而我們的目的除了建立這個轉送群組外，另一個目的就是將這個轉送群組的成員縮到最小，這其實跟找出近似的於最小的連通之支配集合之研究相同的。

在我們的實作上，我們並沒有使用明顯及額外的控制訊息；取而代之的是，我們利繞路時之訊息的交換來建立或維護這個所謂的轉送群組。由於是利用現有的繞路協定來作為輔助，所以並沒有使用明顯及額外的控制訊息。因此，我們需要一個 table-driven 的繞路協定，在實作上，我們選擇 DSDV 作為我們所需要的繞路協定，但是我們的想法並不只侷限在 DSDV；只要是 table-driven 的繞

路協定，並且有作路徑選擇之比較之繞路協定皆可為我們所需要之繞路協定。而會使用 DSDV 的主要想法是，這個協定之核心想法，比較趨近於傳統之 table-driven 的繞路協定。此外，為了要縮減轉送群組的大小，我們修改了原本的 DSDV 之架構。

由於我們的想法是來自群播中的轉送群組之觀念，如圖 4 所示。所以我們還需要一個所謂的根節點之角色。需要這個角色的想法，主要是通常在群播的情況下，在一個群播的群組中，一定會存在著最少一個發送者(source)與一些接收者來組成。因此我們把這個所謂的根節點之角色，當作是群播中發送者的角色，而把其它在網路中的節點都當是接收者的角色，用來建立我們所需要的轉送群組。但是這些人並不是真的是發送者跟接收者，只是為了讓我們建立及維護如群播下的轉送群組所用，真正的資料發送端可以是任何網路上的節點，而接收端則是網路上的所有節點。

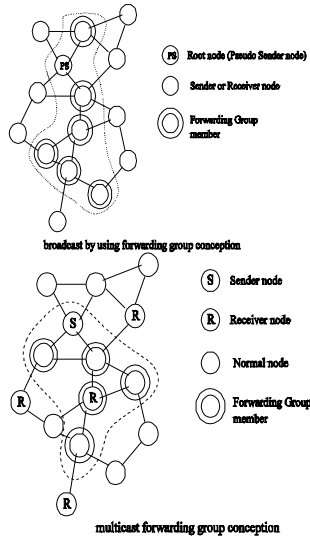


圖 4：群播及廣播下之轉送群組

## 3.2 方法

在轉送群組的建送或維護上，想法下我們可以分為以下兩個部份：第一個部份是轉送群組的建立；而第二個部份是減少轉送群組的成員數。

### 3.2.1 轉送群組的建立

#### 3.2.1.1 加入轉送群組

在實作上，由於我們是架構在 DSDV 繞路協定上，所以每一個在網路上的節點，都必須存在著一個由 DSDV 所維護的繞路表(routing table)；而我們的方式主要是根據 DSDV 繞路協定在運作，在 DSDV 繞路協定中一個節點只有在特定的週期時間；或者當它與週遭有路徑斷裂時才會去作繞路訊息的交換。而為了建造我們所謂的轉送群組，我們採用以下的選擇轉送群組之成員的準則：

- 檢查目的地 (Destination) 欄位是否等於根節點的位置；
- 檢查下一站 (Next Hop) 之欄位是否等於自己。

因此根據這些準則，我們可以知道，當繞路訊息交換時，收到繞路訊息的節點會檢查他收到的繞路訊息，看看是否這個訊息的目的地之欄位為根節點；若是，則它會進一步再去檢查在在這個訊息中的下一站那個欄位是否為它自己；如果是，這個節點會認為他是位於某個節點到根節點的路徑上，因此它必須成為轉送群組的一員，來幫忙資料的轉送。如圖 5 所示，我們可以看到，若以 R 為根節點；則當 D 與 C 在作繞路協定交換時，C 會知道 D 要到根節點的下一站是 C 本身，即 D 要到根節點，必需經過 C；因此，C 會將自己設定為轉送群組的一員。同樣的，B 與 C 在作繞路訊息交換時，B 亦會知道 C 要到根節點的下一站是 B 本身；所以，B 也會將自己設為轉送群組的一員。所以，A 亦會因為 B 而成為轉送群組的一員。

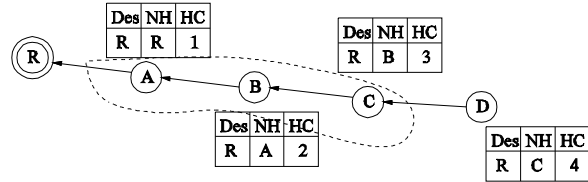


圖 5：建立轉送群組的主要觀念

圖 6 中我們假設 1 號節點是根節點來作為轉送節點建立的依據；一開始作繞路訊息交換時，5 號節點會收到來自 4 號節點的繞路訊息，所以經過比較，它會知道有人要利用它來到達根節點，因此它會自願成為轉送群組中的一員。同樣的，6 號節點也會收到 2 號節點的繞路訊息，而成為轉送群組中的一員；在圖 6(b) 中，我們顯示了這個結果。

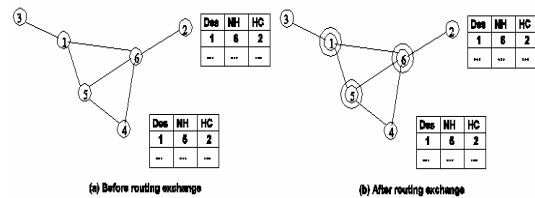


圖 6：轉送群組的建立圖表

除了 5 號節點及 6 號節點因為 4 號及 2 號節點的關係而成為轉送群組的一員外，我們可以看到 1 號節點也是轉送群組的一員，原因是因為 1 號節點是根節點，在我們的實作上，我們將根節點也定為轉送群組的一員。而值得注意的是，利用 DSDV 的繞路協定式所建立出來的轉送群組，雖然可以保證網路上的任一個節點到根節點都是最小路徑的；主要是因為 DSDV 繞路協在繞路表中下一站的選擇上，是採用最少的跳躍數 (hop count) 來作為決策的準則；所以找出來的節點可以就是網路中

任何一點到根節點的最小路徑之點的集合。但是這個集合卻不是近似最小的連通之支配集合，且含有許多的重疊的成員存在；因此在中，我們將提出一些方法來消除這些個重疊的成員，以達到近似最小的連通之支配集合。

### 3.2.1.2 退出轉送群組

在轉送群組建立後，我們必須要有一個機制存在，使得在原來為轉送群組之成員，不再需要為成員時，可以退出轉送群組，以減少不必要的成員存在。在原來本的 DSDV 中，我們並沒有辦法可以作到讓轉送群組的成員可以退出轉送群組。因此，我們使用了一個上游節點串列 (Upstream Host, UH) 來因應我們的需求。這個所謂的上游節點的資料結構所存放的資料主要是那些會經過這個節點的節點。如所圖 7 所示。

在圖 7 中，我們可以看到，由於中間黑色的節點是節點 A、B、C 及 D 要到達根節點所必經，所以在經過繞路訊息交換之後，中間黑色的節點就會知道在節點 A、B、C 及 D 的繞路表中，要到根節點的欄位之下一站皆是它本身，所以黑色節點它會將自個啟動為轉送群組的成員；並且，他會將上游節點的串列記錄為 {A、B、C、D}，用以表示 A、B、C、D 要經過它才能到根節點，所以它有存在為轉送節點的價值。

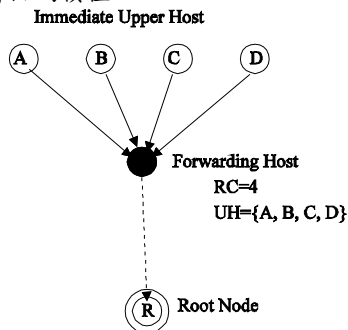


圖 7：上游節點

而在退出的方法上，我們是採用一個所謂的參考數 (Reference Count, RC)，這個參考數會依據節點的上游節點串列的長度來決定；以圖 7 為例，我們知道中間黑色的節點的上游節點串列為  $UH = \{A, B, C, D\}$ ，而其長度為 4，所以這個黑色節點的參考數即為 4 (即為  $UH$  之長度  $|UH|$ )。故當參考數為零時，就表示沒有人要再經過這個節點到達根節點，即這個節點沒有再當轉送群組成員的價值。所以就可以退出轉送群組。

### 3.2.2 減少轉送群組的成員數

由於依原本之 DSDV 的繞路協定，我們可以建立出一個簡單的轉送群組，但是這個群組之成員數卻不是最少的。如圖 6 所示，我們可以看到在 4

號節點的繞路表中，存在著往根節點 (1 號節點) 的下一站是 5 號節點，而在 2 號節點的繞路表中，往根節點的下一站則是 6 號節點。所以當繞路訊息作交換時，5 號及 6 號節點都會檢查上面所提到過的條件；而且會確定它們都是位在某些人到根節點的路徑上，因此它們就會自願地當起轉送群組的成員。但是如果 4 號節點一開始就把 6 號節點當作是往根節點的下一站，而不是利用 5 號節點，那麼我們就可以得到最小的轉送群組。因此為了建立一個最小的轉送群，我們在每個節點中加入了兩個資料，詳述如下。

- Ø 連通性 (Connectivity, C)：每個節點可以偵測到的鄰居數；在此代表此節點最多可以用來代表鄰居的數量。使用這個資料的主要原因是，假如我們可以儘量把一個擁有較多鄰居之節點當作是轉送群組的成員，那麼就可能可以將轉送群組集中，進而縮小轉送群組的大小。
- Ø 參考數 (Reference Count, RC)：每個節點會把它紀錄有多少節點會以它，到根節點的下一站，這裡的參考數其實就是上一個章節所講的參考數，也就是所謂的上游節點串列的長度。其作用如下一章所述，我們在這裡就不在累述。

增加了這兩個欄位，主要的目的是為了得到最小的轉送群組。因為我們是利用轉送群組中的成員來幫忙資料的轉送，也就是說轉送群組限制了那些節點可以幫忙廣播資料的續傳。因此如果我們可以讓轉送群組儘可能的小，則需要被續傳的廣播資料也就跟著變少，所以可以節省下來的重覆廣播就越多。

有了這兩個欄位之後，我們更修改了原來的繞路資訊。這些改變表示如下：

Destination	Next Hop	Hop Count	Sequence Number
-------------	----------	-----------	-----------------

圖 8：修改前的 DSDV 繞路訊息

Destination	Next Hop	Hop Count	Sequence Number	NH Connectivity	NH Reference Count
-------------	----------	-----------	-----------------	-----------------	--------------------

圖 9：修改後的 DSDV 繞路訊息

如我們所看到的，我們增加了兩個欄位在原來的 DSDV 之繞路訊息及繞路表中，這兩個欄位分別為下一站的連通性 (NH Connectivity, NHC) 及參考數 (NH Reference Count, NHRC)；而這兩個欄位所代表的意義分別是下一站的連通性與下一站的參考數。因為一個節點的下一站，一般都是這個節點的鄰居，所以當繞路訊息交換時，每一個人都必需將自身的連通性及參考數給送出去，而收到鄰居就可以得到這些欄位，並存起來。

而如何利用這兩個資料來得到最小的轉送群組呢？在原本的 DSDV 繞路協定中，因為當繞路訊息在作交換時，決定哪一個節點為下一站是利用

跳躍數那個欄位，所以有時我們可能在所比較之兩個訊息的跳躍數這個欄位是相同的，這表示說有超出我們所預料的節點可能會成為轉送群組的一員，而且他們的存在有可能是重覆的。所以，在這個時候，我們使用了額外的兩個欄位來作為要使用那一個節點來作為下一站之考量；我們把參改數作為第二個比較的決策準則而連通性當作是第三個比較的決策準則。因此收到的訊息與繞路表中的訊息之標準的決策準則(這裡指跳躍數)相同時，我們就比較它們之中的參考數。若再相同時，我們再去比較連通性這個決策準則。而藉由使用這些個比較的決策準則，當繞路協定收斂時，我們可以得到最小的轉送群組。就像圖 10 一樣。

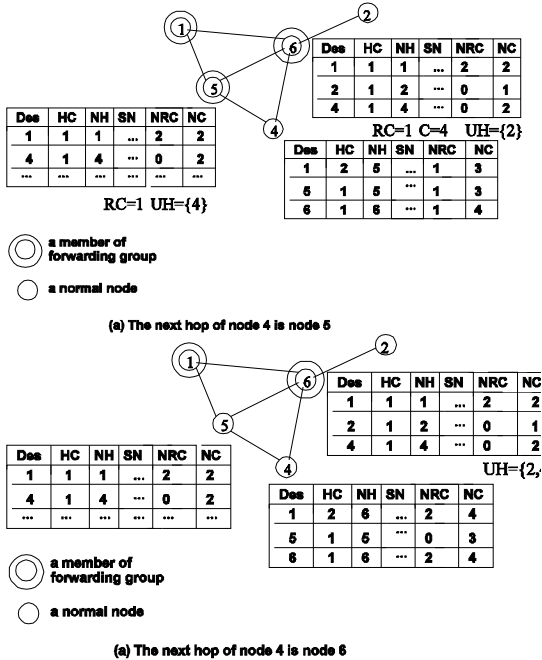


圖 10：最小的轉送群組

因此，現在我們以圖 6 及圖 10 為例，作個說明。當繞路訊息交換時，4 號節點會將他的繞路表之資訊送給跟它的鄰居，5 號及 6 號節點；而此時，5 號節點會認為某一個節點會經過它到根節點(1 號節點)。所以他將上游節點記錄下來 (UH = {4})，更新參考數 (RC = |UH| = 1)，並且變成轉送群組的一員(因為參考數大於零，這代表某一個人要往根節點必須經過這個節點)。同時，6 號節點收到來自 4 號節點的繞路訊息，並且跟 4 號節點做訊息的交換。由於 6 號節點的繞路表中及 4 號節點的繞路訊息中之跳躍數及參考數是相等的，但是 6 號節點的連通性大於 4 號節點的訊息之連通性。所以 6 號節點把自己當作是 4 號節點要往根節點的下一站，因此將此一結果當作繞路訊息送出，此時 4 號及 5 號節點因跟 6 號節點相鄰，因此收到了這個修改過後的訊息。而收到這個訊息後，4 號節點會將其前往根節點的下一站改為 6 號節點，並將這

個改變再次通知它的鄰居，此時 5 號節點會知道 4 號節點，不再利用他來當下一站，所以會將上游節點作更新 (UH = {})，並更新參考數 (RC = |UH| = 0)，所以此時 5 號節點的參考數會等於零，這表示 5 號節點不再需要是轉送群組的一員，所以他會自動放棄掉這個角色。

由此可知，藉由這兩個新增的資料，我們可以儘可能的把轉送群組的成員集中在一群較少但有較好的條件的節點身上，可以有效的減少轉送群組的成員數，並進而趨近於最小的連通之支配群組。而在實驗中我們將把這些比較的順序作一下調換。以比較結果。

## 效能評估

### 4.1 模擬環境

我們所使用的網路拓撲是亂數產生的，且有 100 個節點由亂數產生，散佈在 500 公尺 x 500 公尺到 3000 公尺 x 3000 公尺的區域內。而每個節點的傳輸距離為 500 公尺。使用這些參數的原因是因為我們想將我們的結果跟 [12] 及 [17] 作比較。而我們採用的移動模組如下所述：一個節點會決定它的下一個目的地，並且會以 0m/s 到 10m/s 不等的速度朝著目的地前進。直到到達目的地之後，它會再以亂數決定下一個目的地，並以同樣的方式向目的地前進，直到模擬結束為止。

而這裡所用到的比較準則如下所示。其中第一個及第二個準則來自 [12]。

- Ø Reachability (RE):  $t/N$ ,  $r$  是收到廣播訊息的所有節點數；而  $N$  是指根節點所可以到達的所有節點數，不論是直接或是間接到達。(因為節點移動的關係，可能造成拓撲的不連通)；
- Ø Saved ReBroadcast (SRB):  $(r - t)/r$ ,  $r$  是收到廣播訊息的所有節點數；而  $t$  是有作續傳動作的節點數；
- Ø Number of Forwarding Group Member (FGS): 轉送節點的成員數。

除此之外，在實驗中，我們將以 “Dominating Sets and Neighbor Elimination-Based Broadcasting Algorithm in Wireless Networks” (DNEBA) 為對照組，並跟其結果作比較。會選擇這篇當對照組的主要原因是，這一篇是目前在這方面研究中，可以得到較佳解的作法。

### 4.2 結果

一開始，我們先來看到，我們所需要的根節點之角色對我們的影響，而我們的作法是考慮一個節點若其擁有整個網路之最大連通數的，我們把他考慮為趨近位於整個網路的中央，而若一個節點其所擁有的連通數是整個網路中最小的，我們則將它想

像是位在整個網路的角落。

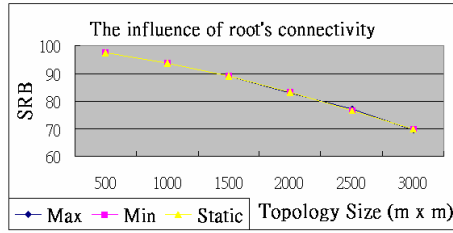


圖 11：根節點位置的影響

其中 Max 代表根節點趨於網路中心，而 Min 則表示趨於網路的邊際角落最後的 static 表示任何位置皆有可能。而由我們的結果，我們知道，不論根節點位在何處，我們皆可以得到相似的結果，所以可知根節點之位置並不影響我們的結果。

而在我們的研究中，我們發現在 DSDV 中，其決定路徑的主要的準則是跳躍數(Hop count)，但是若是在我們的實驗中採用我們的參考數當主要的準則，則我們會得到比較好的結果。如下圖所示。

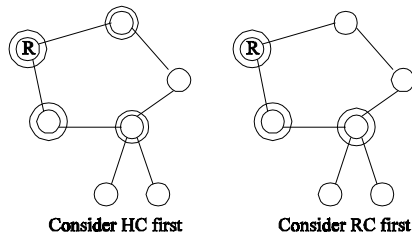


圖 12：主要準則的影響

在上圖左中，我們是採用跳躍數當主要的準則，而我們所得到的結果是需要四個節點來當轉送群組；但是若是我们改用參考數來當主要準則時，我們卻可以得到更加小的轉送群組，只需三個節點即可。而圖 13 展示了主要準則分別為跳躍數及參考數的結果。

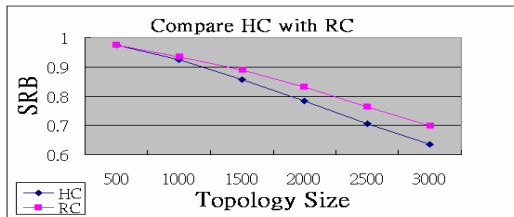


圖 13：主要準則之比較

接下來，我們將比較我們的結果和之前所述的 DNEBA。如圖 14，圖 15 所示：

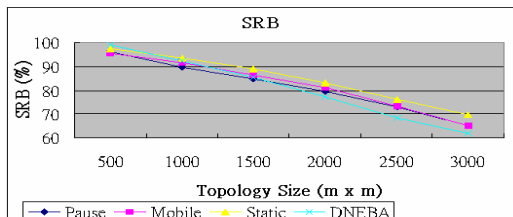


圖 14：SRB

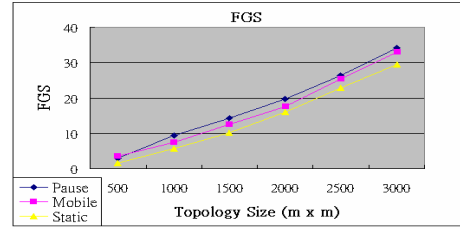


圖 15：FGS

中 Pause 表示在動態的網路中，但是節點卻擁有所謂的 pause time，而 Mobile 表示在動態網路下，但卻沒有 Pause time 的機制。Static 則表示在靜態固定的網路中。最後的 DNEBA 則是我們所要比較的對照組。

如結果所展示的，我們的結果比 DNEBA 還要好；主要的原因是在 DNEBA 的演算法下，我們常常沒有辦法得到最佳的結果，而在我們的演算法下，我們常常可以得到最佳的結果，這就是我們會優於 DNEBA 的原因。在此，我們以圖 16 為例，作一下說明。

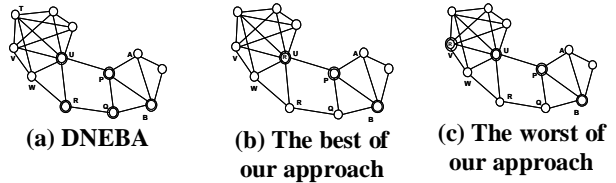


圖 16：我們優於 DNEBA 的原因

在圖 16 中，我們可以看到，若以 DNEBA 之演算法，我們會得到如圖 16 (a) 的結果，他需要有五個節點來當作轉送群組的成員；而若是以我們的演算法，在這樣的網路中，在最好的情況下，我們只需三個節點來當轉送群組的成員，而在較差的情況下，我們也只需要四個節點來當我們轉送群組的成員。所以在我們的模擬中，我們的結果會優於 DNEBA 的原因在於，我們的結果常會趨近於最佳的解，但是在某些情況下，DNEBA 需要更多的節點來當轉送群組的成員。

最後，我們看看所得到的 Reachability 結果，如圖 17 所示。

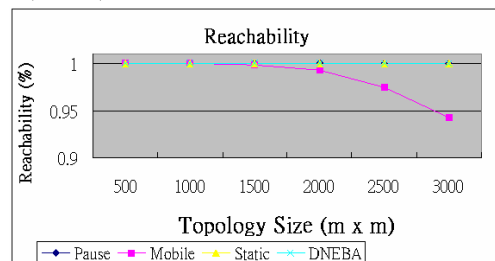


圖 17：Reachability

如果網路拓撲是連通的，則我們可以得到



Reachability 為 1；但是如果網路拓撲是不連通的，則我們所得之 Reachability 將不會為 1；所以在動態網路下，我們所得之結果常會因為網路的不連通，而造成 Reachability 的情況，而且在越大的網路下越容易。

## 結論

在這篇文章中，我們提出了一個利用繞路訊息來建立並維護最小的轉送群組之協定。這裡所謂的轉送群組其實就是相關研究所提及到的連通之支配群組。而我們的方法也跟找尋最小的支配群組是相同的。而利用這個轉送群組，任何節點可以送資料給網路上的任何一個節點。在討論上我們只考慮廣播的情況，但是在實際上，我們的方法也適用在群播，因為我們是把廣播當作是群播的一個特殊的例子。

而由於我們所建立出之轉送群組是最少成員數的，所以我們可以省下很多不必要的續傳廣播。且此方法並不像群播的情況一樣，需要大量的資料傳送，也不需要有一個固定的發送者，更不需要去維護一個特殊的架構，如樹狀的維護。

由我們的結果，我們知道我們的想法是非常的趨近於 MCDS 的建構；且我們採用的方法是在原本的繞路訊息作一些修改，所以並沒有其它明顯的負載。

最後，由於我們的想法是架構在原有的繞路之機制上，所以只要在網路中有支援繞路表之機制，則我們的方法即可應用於上。故目前有一些這方面的繞路協定在動態改變之隨意的無線網路中被研究，因此我們相信，在這些協定上應用我們的想法，一定還是可以在動態改變之隨意的無線網路中，運行的不錯。

## 五、參考文獻

- [1] S. H. Bae, S. J. Lee, W. Su and M. Gerla, "The Design, Implementation, and Performance Evaluation of the On-Demand Multicast Routing Protocol in Multihop Wireless Networks," IEEE Network 2000.
- [2] Y. P. Chen and A. L. Liestman, "Approximating Minimum size Weakly-Connected Dominating Sets for Clustering Mobile Ad Hoc Networks," MOBIHOC '02.
- [3] C. Chiang and M. Gerla, "On-Demand Multicast in Mobile Wireless Networks," Proc ICNP 98, Texas, October 14-16 1998.
- [4] S. K. Das, B. S. Manoj and C. S. R. Murthy, "A Dynamic Core Based Multicast Routing Protocol for Ad hoc Wireless Network," MOBIHOC '02, June 9-11, 2002
- [5] M Gerla, T. J. Kwon and G. Pei, "On Demand

Routing in Large Ad Hoc Wireless Networks with Passive Clustering," Proceedings of IEEE WCNC 2000, Chicago, IL, Sep. 2000

- [6] J. J. Garcia-Luna-Aceves, "The Core-Assisted Mesh Protocol," IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL. 17, NO. 8, Aug. 1999.
- [7] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," Mobile Computing, eds. T. Imielinski and H. Korth (Kluwer Academic, 1996) pp. 153-181.
- [8] T. Kunz and E. Cheng, "On-Demand Multicasting in Ad-Hoc Networks: Comparing AODV and ODMRP," ICDCS'02.
- [9] S. J. Lee, M. Gerla and C. C. Chiang, "On-Demand Multicast Routing Protocol," Proceedings of the IEEE Wireless Communications and Networking Conference, WCNC 99, pages 1298-1304, September 1999.
- [10] S. Lee and C. Kim, "Neighbor Supporting Ad hoc Multicast Routing Protocol," MobiHOC 2000, Aug. 2000, Boston.
- [11] M. Lee and Y. K. Kim, "PatchODMRP: An Ad-hoc Multicast Routing Protocol," ICOIN 2001: 537-543.
- [12] S. Y. Ni, Y. C. Tseng, Y. S. Chen and J. P. Sheu, "The Broadcast Storm Problem in Mobile Ad Hoc Network," Mobicom '99.
- [13] W. Lou and J. Wu, "On Reducing Broadcast Redundancy in Ad Hoc Wireless Networks," IEEE TRANSACTIONS ON MOBILE COMPUTING, VOL. 1, NO. 2, April-June 2002.
- [14] C. E. Perkins and P. Bhagwat, "DSDV: Routing over a Multihop Wireless Network of Mobile Computers", Mobile Computing, pp.183-206
- [15] E. M. Royer, "A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks," IEEE Personal Communications, April 1999.
- [16] E. M. Royer and C. E. Perkins, "Multicast Operation of the Ad-hoc On-Demand Distance Vector Routing Protocol," Mobicom '99.
- [17] Stojmenovic, M. Seddigh and J. Zunic, "Dominating Sets and Neighbor Elimination-Based Broadcasting Algorithms in Wireless Networks," IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEM, VOL. 13, NO. 1, January 2002. sequences," in Proc. Fourth Asian Conference on Computer Vision, pp. 961-964, 2000.