利用區域演算法將 ad hoc 無線網路下之 Connected Dominating Set 最小化

A Localized Algorithm for Minimizing Connected Dominating Set in Ad Hoc Network

賈坤芳 江茂綸 陳俊榮

中興大學資訊科學系 中興大學資訊科學系 中興大學資訊科學系

kfjea@cs.nchu.edu.tw phd9208@cs.nchu.edu.tw s9356001@cs.nchu.edu.tw

摘要

在 ad hoc 無線網路下,目前最重要的議題就是如何在一群行動裝置中,作最有效率的繞徑。而以 connected dominating set (CDS)為基礎的繞徑方式,被認為是非常好的方法,其主要的優點就是可以把 CDS 當作一個 virtual backbone,便能迅速的適應網路拓樸(network topology)的改變。目前的研究主要是以求得較少的 gateway 數,去建立一個精簡的 virtual backbone,來減少儲存的繞徑資訊。本研究提出一個區域演算法去找出一個 gateway 數較少的 CDS,經由實驗顯示我們的新方法找出的CDS 是小於其他各種方法,特別是在行動裝置無

關鍵詞: Ad hoc 無線網路、繞徑、Connected dominating set。

法提供較大的傳輸半徑時,效果更為明顯。

Abstract

Efficient routing plays a very important role in the mobile ad hoc network (MANET). Routing based on a connected dominating set (CDS) has been recognized as a promising approach to efficient routing. The main advantage of treating CDS as a virtual backbone is its adaptability to the fast change of MANET's topology. Current research proposed various distributed algorithms to find CDS with fewer gateways for reducing the storage of routing information. In this paper, we propose a new localized algorithm to find a CDS and minimize its gateway number. Experimental results show that the CDS generated by our algorithm has gateways fewer than those of other existing algorithms. The result is especially significant when mobile hosts can not provide a larger transmission radius.

Keywords: Ad hoc network, Routing, Connected dominating set.

一、前言

Ad hoc 無線網路是一群備有無線傳送/接收器且能任意自由移動的裝置或是路由器設備。這種網路的優點在於它完全不需要任何有線網路的架構或設備的支援,可隨時改變其網路拓樸型態,且移動時沒有方向性或範圍的限制。這些特性適合應用於緊急救災及軍事的用途上,因為在危險的環境中,一些主要的基礎通訊設施如基地台等,都可能無法使用。而在地理環境的限制與時間緊迫的壓力下,ad hoc 無線網路的無線架構及可以任意移動的特性便可發揮較大的功用。

在 ad hoc 無線網路下的繞徑是一個挑戰,因為這種類型的網路具有低電量 有限的頻寬及可任意移動的特性。相較於傳統有線網路都是使用到link state[10][12]或是 distance vector[7][11]的方式,這些技術不適合直接套用至無線環境上,特別是和無線網路特性有衝突的演算法,例如需要複雜計算或耗費記憶體的演算法,都是不切實際的。

目前 ad hoc 無線網路下的繞徑協定分為兩 種:主動式(proactive)和回應式(reactive)。在主動 式繞徑中,每個行動裝置間,每隔一段時間就會發 送一些路徑相關資訊,各個行動裝置就依據蒐集進 來的資訊去改變自己的繞徑表格(routing table)的 內容,例如 Destination Sequenced Distance Vector (DSDV)[14]就屬於這一類 這類繞徑協定可以讓每 個送出去的封包立刻得知到達目的地之間的路 徑,不會有任何延遲。但是這種協定必須週期性的 去廣播訊息,相當浪費無線網路的頻寬和行動裝置 的電力。然而如果降低廣播所造成大量的消耗,就 要拉長每次廣播的間隔時間,這又會造成繞徑表格 不能即時正確的反應網路拓樸變化。在回應式繞徑 協定中,若行動裝置找不到欲傳送封包的目的地路 徑時,才會開始運作;像 Dynamic Source Routing (DSR)[2], Ad hoc On-demand Distance Vector Routing (AODV)[13]等就屬於這一類。這類協定的 好處就是頻寬使用量較小,只有在某一個行動裝置 未能從繞徑表格中找到欲送封包的路徑時,其平均 延遲時間才會較長。為了有效率解決網路的異變

性,階層式(hierarchical)的繞徑被廣泛的使用。Dominating-set-based routing[17]就是階層式繞徑的一種,此類型的方法利用 GRID[8]的方式來繞路,再加上位址資訊的輔助,可達到延展性的要求,並可以減少尋找路徑時的額外負擔,故本文以dominating-set-based routing 作為研究對象。

Dominating-set-based routing 是利用圖形理論 中的 dominating set 觀點。其定義為在一圖形中, 某一個子集合內的點稱為 dominating set, 假若其 他所有不在此子集合內之點,至少有一個相鄰的點 在此子集合中。在 dominating set 的點我們稱之為 gateway 裝置, 而不在 dominating set 內的點稱之為 non-gateway 裝置。使用 dominating-set 去繞徑的 好處就是只要在 gateway 裝置保留繞徑的資訊即 可。此外,在較小的 connected dominating set (CDS) 下進行廣播(broadcast)時,因為都只透過 gateway 裝置傳送訊息,所以裝置間彼此的訊息也會減少很 多,可以避免額外的頻寬及電力消耗。例如在圖1 中,v和w為相連接的 gateway 裝置,形成了 CDS, 而 u x 和 y 為 non-gateway 裝置, 所有的裝置都必 須透過 v 或 w 來傳送訊息。在圖 1 中的每一個圓 圈代表每一個裝置在無線網路中的傳送範圍。

由上面討論可知,因為 ad hoc 無線網路的頻寬資源及電池容量有限,在整個網路拓樸圖中找出接近最小化的 CDS,可以減少網路中廣播的訊息量,也節省大量的網路頻寬資源的浪費及電量的消耗。但是找出最小的 CDS,此問題為 NP-hard[6]問題,一般尋找最小化 dominating set 的圖形演算法[9]在 ad hoc 無線網路中無法實際執行,因為每個行動裝置對於整個網路的資訊所知有限,大多只能得知與行動裝置相連的鄰居資訊。若行動裝置要得到整體網路的所有行動位置資訊,需要一套 GPS (Global Positioning System)來提供服務,而使用GPS 是一項昂貴的網路資源,因此本文暫時不考慮使用 GPS 系統,而是使用分散式的區域演算法,透過網路彼此交換訊息,進而獲得一個最小的CDS。

先前學者 Wu 等[4]提出一個有效而簡單的分散式方法,先在 ad hoc 無線網路中建構出 CDS,進而利用行動裝置間的連結關係及節點 id,減少 CDS 的大小。此方法可以從已知的網路相連圖中,迅速的建構一個較小 CDS。如前所述,使用dominating-set-based routing 方法時,只要當 CDS內的 gateway 個數越少時,所需要儲存繞徑表格的節點數相對的減少。但方法[4]卻可能因為節點 id與節點位置設定之對應關係不佳,而造成 CDS 的gateway 數過多。因此我們提出分散式的區域演算法,利用局部的資訊找出接近最小化的 CDS。

本論文結構如下,第二節為相關研究和預備工作,描述在 ad hoc 無線網路中,有關於繞徑的相關研究,並回顧 dominating-set-based-routing 的標記和重新標記方法;第三節描述本論文所提出新的

演算法來改進現有的 CDS 方法及減少 CDS 的大小。第四節為說明模擬實驗和實驗結果分析。第五節為結論和未來工作。

二、相關研究和預備工作

2.1 節說明在 ad hoc 無線網路上,建構 CDS 的相關研究, 2.2 節描述 Wu 等所提出的 dominating-set-based routing[17]及其建構 CDS 的方法。

2.1 相關研究

在 CDS 的建構上,有許多種方式可以選擇。 Das 等之演算法[5]定義一個子網路,來產生一個最 小的 CDS (MCDS)。此演算法第一步驟是從所有節 點中,利用貪婪演算法(greedy algorithm),依序挑 選一個最大有效連接節點數(effective degree)的節 點,形成一個 dominating set C';當 C'可以涵蓋所 有節點時,則此步驟就停止。而 C'可能包含許多 彼此不相連的部分(component)。接下來,使用最少 的節點去連接每個 C'的部分而形成一個最小生成 樹(Minimum Spanning Tree, MST)。在這步驟中, 每個連結(link)都給定一個比重(weight);若一條連 結的兩端不為 C'的節點,且屬於同一個部分,就 把此連結的比重定義為無限大;而其他連結的比重 則由不在 C'中的終端點個數來決定。 再來選擇比 重較輕的連結來相連其他部分,當此最小生成樹形 成後,樹的內部的節點和其連結線構成一個 CDS。

除此之外,Stojmenovic 等[16] 使用分散式的方法來建構 CDS。他們將 CDS 的節點定義成兩種型態,cluster-heads 和 border-nodes。網路上每個節點被劃分成許多聚集(cluster),而每個聚集中有一個節點為 cluster-head 來為其他節點提供服務。在此方法中,每個 cluster-head 間彼此不相連,但每個 cluster-head 在聚集中,必須跟每個節點直接相連接;而 border-node 則負責在不同聚集間的傳輸。此外,在相同聚集中的任兩個節點至少有兩個路程(hop)的距離。在這個結構中,來源端的節點藉由它的 cluster-head 來傳送訊息,cluster-head 再傳遞這個訊息給 border-node。然後,由 border-nodes去傳遞訊息給目的地的 cluster-head;因此任兩相鄰聚集都必須依賴 border-node 當作橋樑來溝通。

Wan 和 Alzoubi[1]提出一個固定的近似值比率(constant approximation ratio)分散式演算法。此演算法有兩個步驟,首先是建立最大獨立集合(Max Independence Set, MIS),也就是所有的節點先建立一個生成樹 T,其根節點 v 由 leader-election[3]演算法來決定。每個節點依照距離根節點的位置和 id 來決定本身的等級。所有節點一開始標記為白色,然後從根節點 v 標記為黑色,並廣播黑色訊息給其他節點;當一個節點收到此訊息且自己本身還

是白色時,則標記自己為灰色,同時也廣播灰色訊 息給其他節點;當一個白色節點收到灰色訊息,且 本身的等級比發送此訊息的節點還要低時,就會標 記自己為黑色,同時也廣播黑色訊息出去。當所有 的葉節點都被標記完後,此最大獨立集合就被建立 完成。第二步驟,就是建立一個 dominating tree T*。一開始,每個節點都會有個變數z來判斷是否 加入 T*, 而 T*的根節點是從 T 根節點的鄰居中, 挑選有最多被標記為黑色節點的鄰居來當做根節 點。此外在 T*中,每個節點都會儲存自己父親和 兒子的節點。接著,從T*的根節點廣播 INVITE2 的訊息,當黑色節點收到此訊息時,紀錄發送此訊 息的節點為自己的父親節點,並回送加入訊息給發 送節點,然後廣播 INVITE1 訊息。當灰色的節點 收到 INVITE1 的訊息時,也紀錄發送此訊息的節 點為自己的父親節點,並回送加入訊息給此發送節 點,然後廣播 INVITE2 訊息。最後所有的節點都 會加入到 T*中,而在 T*中的內部節點就會形成 CDS.

2.2 預備工作

本研究是以 Wu 等[4][17]之研究為基礎,在本節中,我們將進一步描述 Wu 和 Li 所提出的dominating-set-based routing,以及其 CDS 的建構方式。

假設在 ad hoc 無線網路中, CDS 已經被建構出來,而 dominating-set-based routing 的方法可分成下列三個步驟:

- 1. 當來源端(source)不是 gateway 時,它會 傳送封包給相鄰的 gateway,我們稱之為 來源端 gateway。
- 2. 來源端 gateway 會當作一個新的來源端,經由 CDS 所產生的子網路中找到此封包的路徑。
- 3. 最後,此封包將會傳送到一個目的端 gateway。假若它不是目的端的話,也一 定是與目的端相鄰的 gateway;隨後此目 的端 gateway 將會直接傳送此封包給目 的端。

每個 gateway 都保留 gateway 所支配的成員和 gateway 繞徑表格的資訊。 Gateway 所支配的成員 包含與 gateway 相鄰的 non-gateway 節點; Gateway 繞徑表格包含所有的 gateway 資訊和它所支配的成員。圖 2(a)為一個 ad hoc 無線網路圖,其中節點 4、8 和 9 分別為 gateways,構成一個 CDS;圖 2(b)表示節點 8 所支配的成員,包含節點 3、10 和 11;圖 2(c)表示節點 8 的繞徑表格,包含所有 gateways(4、7 和 9)的資訊和其支配的成員等。

在 ad hoc 無線網路中, Wu 等[17]提出一個簡單且有效率的分散式演算法來建構 CDS。 演算法

有兩個步驟,一是標記方法,另一個是重新標記方法。一開始將整個網路視為一個簡單的圖形G=(V,E);對每個節點 $v(v \in V)$ 做標記 m(v),標記分為 $T(\max E)$ 和 $T(\max E)$ 。標記方法是透過訊息交換,判斷是否存在兩個相鄰的鄰居,若沒有則將自己標記為 T。經過標記工作後,我們假設在 V 中被標記成 T 的集合為 V' , $V'=\{v|v \in V, m(v) = T\}$,而 G' 是從 V' 中,經由 V' 所 導出的子圖,即 G'=G[V']。接下來是重新標記方法,主要可分為下列兩個法則。

法則 **1**: 考慮在 G'中的兩個點 v 和 u , 若 N[v] $\subseteq N[u]$ 且 id(v) < id(u) , 則 v 的標記則改成為 F , 而 G'也被改為 G' - $\{v\}$ 。

法則 2: 假設在 G'中, u 和 w 為 v 兩個相鄰的節點, 若在 G 中 $N(v) \subseteq N(u) \cup N(w)$, 且 $id(v) = min\{id(v), id(u), id(w)\}$, 則 v 的標記被改成 F。

除了上述兩法則外,Wu 等還提出了更廣泛的 法則 k[4]來減少 CDS 的節點數。原先法則 1 是利用 1 個節點使某些節點重新標記為 F,而法則 2 是利用 2 個節點使某些節點重新標記為 F,而法則 k 則可利用 k 個節點使某些節點重新標記為 F,所以法則 1、2 只是法則 k 的特例。

根據 Wu 等的演算法可以找出一個較小的 CDS 來做繞徑工作。並且此演算法可以保證每個 行動裝置都可以經由 dominating set 來跟其他節點 作連繫,以達到繞徑的效果。

但法則 k 卻有個缺點, 若單純只依賴節點的 id 作重新標記的程序,容易因為節點 id 與節點位置對應關係不佳,而使得找出來的 CDS 仍過大。舉例來說,如圖 3 所示,當大的節點 id 在節點分支度高的情況下,所找出的 gateway 數為 2;然而在圖4中,若是節點分支度高的節點 id 較小,則使得找出的 gateway 為 6。所以本研究提出一個新的重新標記方法,改善這個問題,且降低 CDS 的大小。

三、建構 CDS 的演算法

本 節 提 出 RCDS (Reducing Connected Dominating Set)演算法,獲取一個較小的 CDS。3.1 節介紹相關符號和名詞定義,3.2 節描述 RCDS ,包含了我們所提出的 reassignment 及 advance 兩個程序。

3.1 問題假設、符號及名詞定義

首先我們假設在 ad hoc 無線網路上,每個行動裝置都有個獨一無二的 id,且此 id 為一數值。 而所有的行動裝置都是相同的,即每個行動裝置的 無線傳輸範圍都是一樣的。

接著說明本論文使用到的圖形符號和名詞。 我

們以圖形 G = (V, E)來表示 ad hoc 無線網路,其中 V 表示在無線網路中行動裝置的集合; E 表示邊 (edge)的集合。若兩個行動裝置 u 和 v 都在彼 此的傳送範圍之內,則表示為一個邊(u,v)。 $N(u)=N_1(u)=\{v\mid (u,v)\in E\}$ 代表節點 u 在 1-hop 內的 鄰居集合, $N[u]=N_1[u]=N(u)\cup \{u\}$ 表示節點 u 在 1-hop 內包含本身節點的鄰居集合。依此類推, $N_2(u)$ 表示節點 u 在 2-hop 內的鄰居集合,而 $N_2[u]$ 則表 示節點 u 在 2-hop 內包含本身節點的鄰居集合。此 外,我們假設在V 中標記成 T 的集合為V', $V' = \{v \mid v \in V, m(v) = T\}$, 而 G' = G[V']。 隨後我們 定義若一個節點 u 為 neck node , 則表示該節點可 將其鄰居切成兩個或兩個以上的強連通成份 (strong connected component),且這些強連通成份 沒有共同的鄰居,除了節點 и 外。而這種類型的節 點,我們將用來當使其他節點標記為 F 的條件, 例如圖 5 的節點 3 即為 neck node。

3.2 Reducing Connected Dominating Set (RCDS) 演算法

我們所提出的 RCDS 演算法,可分為下列二個步驟。

- 1. 執行標記方法,產生初始的 CDS。
- 2. 執行重新標記方法。
 - (1) 執行 reassignment 程序, 去減少 CDS 的 gateway 數。
 - (2) 執行 advance 程序,求得更小的 CDS

我們將在 3.2.1 節說明標記方法 , 3.2.2 節詳述 reassignmet 程序 , 3.2.3 節介紹 advance 程序。

3.2.1 標記方法

RCDS 演算法的標記方法是採用 Wu 的演算法 [17]去建構初步的 CDS。在 ad hoc 無線網路下的每個節點,定期廣播一個訊號(beacon)封包來宣告其存在;這個訊號封包包含此節點的 id,因此每個節點都藉由訊息交換來獲得相鄰節點的個數和其他資訊。整個網路視為一個簡單的圖形 G=(V,E),對每個 $u(u\in V)$ 做標記,標記分為 T(marked)和 F(unmarked)兩種,我們假設一開始所有的點都為未標記。緊接著,標記方法執行下列三個步驟。

- 1. 把所有的節點 u 標記為 F。
- 2. 每一個 u 跟它的鄰居交換它的 open neighbor set N(u)。
- 3. 假設 u 存在兩個沒有相鄰的鄰居,則就標記 m(u) 為 T。

經由上述標記方法後,初步的 CDS 已經被建構完成。

由於標記方法所建構出來的 CDS 仍相當大,

所以下一節介紹 reassignment 程序來減少 gateway 的數量。

3.2.2 reassignment 程序

我們的 reassignment 程序利用 2-hop 內的資訊,包含 2-hop 內的最大 id 值 節點分支度(degree) 及節點 id 值,利用這些參數,計算出新的 id'。若 id'值愈大,則愈可能成為 CDS 中的一員。接著我們透過訊息交換去找出 $neck\ node$,且把這類型的節點當成 CDS 中的成員,便可以作為其它節點重新標記為 F 的條件。所以,我們首先執行 find- $neck\ node$ 的程序,找出 $neck\ node$ 後,再利用 reassignment 程序來減少 CDS 的 gateway 數。

find-neck node 程序

- 1. 每一個 $u \in V$,建立子圖 $G[V_{+}]$,其中 $V_{+}^{'} = \{w \mid w \in N(u)\}$ 。
- 2. 從 $G[V_{+}]$ 中,計算出一組強連通成份 $\{V_{c_1}, V_{c_2}, \bot, V_{c_l}\}$,其中 V_c 表示為一個強 連通成份。
- 3. 若 l 2,且滿足任兩個 V_c 交集皆為 $\{u\}$,則節點 u 為 $neck\ node$ 。

除了 find-neck node 程序外, reassignment 程序 還需要利用法則 k[4]來減少 CDS 的 gateway 數。

法則 k 演算法

- 1. 每一個 $u \in V'$,建立子圖 $G[V_{+}]$,其中 $V_{+} = \{w \mid w \in (V' \cap N_{2}(u)) \land (id'(u) < id'(w))\}$
- 2. 從 $G[V_{+}]$ 中 , 計算出一組強連通成份 $\{V_{c_1}, V_{c_2}, \vdash, V_{c_l}\}$, 其中 V_c 表示為一個強連通成份。
- 3. 若存在 V_{ci} , 1 i l, 使得 $N(u) \subseteq N(V_{ci})$, 則將節點 u 標記為 F。

根據 find-neck node 程序,我們可以找出圖形中的 neck node,再加上法則 k 的重新標記方法,進而使用在 reassignment 程序中。隨後,我們將 reassignment 程序分為下列二個步驟:

1. 每一個 $u \in V'$, 收集其 2-hop 節點的資訊,包含節點 id、節點分支度及是否為 $neck \ node$,計算 $N_2[u]$ 內的 id',計算式子如下:

id' = maxid * d + id 式子(1)

maxid 為 N₂[u]中的最大 id 值

d 為節點的分支度

id 為節點原始的 id

若節點為 neck node,則 id'給予無限大 ()的值。

2. 利用計算出的 id', 套用法則 k[4], 減少 標記為 T 的節點數。

在式子(1)的計算方式,是因為我們認為若要找出較小的 CDS,節點的分支度因素比原始 id 還要重要,再加上我們給予 $neck\ node$ 最大的 id'值,所以可以改善原本法則 k 可能會因為節點位置不佳所造成 CDS 較大的情況。

接著我們說明利用式子(1)所計算出的 id'仍是唯一。每一個 $u \in V$,都會去計算其 $N_2[u]$ 的 id',而其中的 maxid 是節點 u 在 2-hop 內的最大 id 值,所以為一個已知固定值,再乘以變數 d(節點分支度),且加上一個唯一的 id 值,使得計算出的每個 id'仍是唯一的。

我們展示一個範例來描述 reassignment 程序,圖 6 為使用標記方法後,產生的 gateway 情形。圖 7 中,節點 4、11、20、23 為 $neck \ node$,而節點 2、9 收集 $N_2[u]$ 內每個標記為 T 的節點資訊,包含節點 2、4、9、11。利用式子(1),計算出新的 id',而 N(9)可被 id'大於 id'(9)的一組強連通成份 $\{2,4\}$ 的鄰居所涵蓋,因此節點 9 可被重新標記成 non-gateway。而節點 2 因為找不到一組可以涵蓋 N(2)的強連通成份,所以仍為 gateway。每個標記為 T 的節點執行 reassignmnet 程序後,最後的結果呈現在圖 8,共有 7 個節點成為 gateway。而使用 Wu 的法則 k [4]與 Wu-ND[18]結果呈現在圖 9、10,分別是 11 和 9 個 gateway。因此我們使用 reassignmnet 方法來改善法則 k 的問題,進而求得較小 CDS。

接著我們展示 RCDS 演算法的第二部分 advance 演算法,透過該演算法可以再降低 CDS 中的 gateway 數,去求得更小的 CDS。

3.2.3 advance 程序

雖然 reassignment 程序能改善法則 k 可能因節點 id 與位置對應不佳而造成 CDS 較大的情況,然而仍有縮小 CDS 大小的空間,所以我們再提出 advance 方法。然而這個方法是類似 Wu 的有限制法則 k 演算法[4],其最主要的不同,就是在建立子圖時,去除只收集節點 id 比自身 id 大的資訊條件,才建立子圖,再利用相互包含關係,求得更小的 CDS。

advance 程序

- 1. 每一個 $u \in V$, 建立子圖 $G[V_{+}]$, 其中 $V_{+} = \{w \mid w \in (V \cap N(u))\}_{\circ}$
- 2. 從 $G[V_+]$ 中,計算出一組強連通成份 $\{V_{c_1}, V_{c_2}, \bot, V_{c_l}\}$ 。其中 V_c 表示一個強連 通成份的節點集合。
- 3. 若存在 $\stackrel{V}{ci}$, 1 i l, 使得 $N(u)\subseteq N(V_{ci})$,則將節點 u 標記為 F。

舉例來說,圖 11 為標記方法的節點個數,節點 2 3 5 6 被選為 gateway。 套用 advance 方法,N(5)可被一組強連通成份 $\{2,3,6\}$ 的鄰居所涵蓋,因此可重新標記為 non-gateway,而最後找出的 gateway 為 $\{2,3,6\}$ 。

但是 advance 程序有一個缺點,就是節點不能同時執行這個演算法,否則可能造成所找出的 CDS 是錯誤的。所以在執行時,必須通知 $N_2(u)$,確認只有一個節點在執行 advance 程序,若有二個以上的節點要執行此程序,則以 id 小的節點優先執行,避免同時重新標記為 F,造成找出來的 gateway,不能涵蓋整個網路。而在 Wu-ND,Wu-EL2[18]和 Sheu[15]所提出的這三種演算法,也皆有這個問題。如圖 11,若 $\{2,3,5,6\}$ 四個節點同時執行這幾種演算法,節點 2、6 則同時被重新標記為 non-gateway,造成整個網路不是一個連通的狀態。

由於我們提出的 RCDS 演算法,只利用局部 的資訊來減少 CDS 的 gateway 數,所以 RCDS 是 一個區域演算法。下一節,將以實作我們提出的演 算法與其他方法,作模擬實驗比較。

四、模擬實驗

4.1 節描述實驗架構和環境,4.2 節描述實驗結 果和分析。

4.1 實驗架構和環境

我們實作了 reassingment 方法、 RCDS (reassignment+advance)這兩個方法,以及 Wu-ID、Wu-ND、Wu-EL2[18]、法則 k [4] 、 Sheu[15]各種演算法,並加以作比較。而其中的 Wu-ND 更是現有分散式演算法中,效果較佳且簡單的方法。本實驗環境皆在 Windows 平台上,使用 C++來撰寫程式。

我們首先對各種建構 CDS 的方法作比較。 Wu-ID 使用節點 *id* 當作重新標記的判斷依歸;而 Wu-ND 利用 *id* 和節點相鄰個數當作重新標記的衡量標準:Wu-EL2 也額外加入電力條件到重新標記

中。Sheu 也根據 Wu 的方法提出一個新的電力衡量方式來建構 CDS。當節點之電力高於某個門檻值,就設為 bottleneck 節點,依相鄰 bottleneck 節點的個數當作建構 CDS 的標準。而法則 k 則是與前述方法有些不同,是可利用多點 id 的連結關係來作為重新標記的判斷標準。

我們的模擬環境是假設在一個 100 X 100 的二維空間下,隨機任意分配 20~100 個節點,每個節點的傳輸半徑皆相同,若兩節點之間的距離小於傳輸半徑範圍(R),就表示這兩個節點互相相連。為了實作包含電力條件的演算法,我們參照 Sheu[15]的論文,將每個節點的電力上限設為 10000 個單位,隨機分配給各個節點電力為 1000 至 10000 個單位,將 Sheu 所提出的 $E_{threshold}$ 定為 4000 個單位。因為節點位置是隨機分佈的,為了確保圖形的相連,我們使用 depth-first search (DFS)演算法,來判斷圖形是否相連通;若圖形不相連通,則捨棄此圖形,再重新產生一個圖形,直到此圖形為連通圖為此。本論文暫不詳細討論若訊息在相同頻道裡,如何去避免爭奪和衝突等問題,我們假設這些問題已在 MAC 層被處理完畢。

模擬的步驟遵循下列程序:

- 1. 產生一個靜態的無向網路圖。
- 2. 利用標記方法和重新標記方法來計算 gateway 節點,記錄每個方法所形成 CDS 的 gateway 個數。

我們依照上述程序,執行模擬實驗。

4.2 實驗結果和分析

這一節,將我們提出的演算法和其他方法所產生的 gateway 數作比較。我們將其他方法分為兩類個別比較,一類是沒有使用電力條件的方法,另一類為有使用電力條件的方法。我們觀察在不同的傳輸半徑 R(25、40、60)下,各種方法所產生 CDS的大小。假若在相同節點數下,使用愈小的傳輸半徑,表示網路的連結愈稀疏(sparse);相對地,使用愈大的傳輸半徑,則表示網路的連結愈密集(dense)。實驗結果以圖表示,圖中的 x 軸為網路節點個數, y 軸為 CDS 大小。

圖 12、13 和 14 是與沒有使用電力條件的方法相互比較,傳輸半徑 R 分別為 25、40、60。在圖 12、13 和 14 中 CDS 的 gateway 數由大到小排列分別為 Wu-ID、法則 k、reassignment、Wu-ND、RCDS。其中 reassignment 在 R=25 的情況下,改善了法則 k 可能因為節點 id 與位置對應不佳而造成 CDS 較大的情形。而 reassignment 效果較 Wu-ND 差是因為 Wu-ND 有個利用包含關係而不用考慮節點 id、節點分支度的條件,使得 CDS 的 gateway 數較少。而 advance 方法也是利用只考慮包含關係,來縮小 CDS 的大小,所以在經過 advance 方

法的程序後,找出的 CDS 大小是小於其他各個方 法。圖 13 的傳輸半徑 R=40, 顯示在節點數 80 以 上時,reassignment 效果是稍差於法則 k,這是因 為 reassignment 雖然是挑選節點分支度較大的節點 當 gateway,但是節點多且傳輸半徑大的情況下, 使得兩個距離較近且分支度較大的節點是重覆連 接某些節點,而且因這些節點的節點分支度較大, 不能被重新標記為 non-gateway , 所以會比法則 k找出的 CDS 稍為大一點 圖 14 的傳輸半徑 R=60, 呈現 reassignment 效果優於法則 k , 這是因為節點 分支度大的節點連接大部份的節點,所以只要些許 個節點分支度大的節點即可涵蓋整個網路。圖 15、16 和 17 是與有使用電力條件的方法作比較, 所顯示的 CDS 大小由大排到小為 reassignment、 Wu-EL2、Sheu、RCDS。reassignment 的效果在大 部份情況下是差於 Sheu、Wu-EL2, 理由與上述的 Wu-ND 相同。RCDS 在上述六個圖中,僅在傳輸 半徑 R=60 時,與 Wu-ND、Sheu 所得到的結果相 去不遠,這主要是因為網路密集,節點互相涵蓋的 原因。RCDS 在實驗中幾乎皆得到最佳的結果,這 是因為 advance 方法只利用包含關係來重新標記 gateway,因此可以得到較好的效果。

從實驗結果得知,我們所提出的 reassignmnet 方法在大部分情況下,改善了法則 k 易因節點 id 與位置對應不佳造成 CDS 較大的缺點,且加入 advance 方法的 RCDS,移除了多餘的 gateway,使得 CDS 的 gateway 數明顯減少,且優於其他方法,特別是在行動裝置無法提供較大的傳輸半徑時,效果更為明顯。

五、結論

在 ad hoc 無線網路環境下,通訊和計算都有 一定的限制,因此如何利用有限的資訊去找一個精 簡的 virtual backbone 是一件不容易的事。先前學 者提出了許多分散式的方法去建構 CDS 以形成 virtual backbone。本研究提出 RCDS 演算法,延伸 了法則 k[4]的想法,並套入節點分支度及 neck node 的概念,利用多點互相包含的關係,提出了 reassignment 方法, 改善法則 k 可能會因節點 id 與 位置對應不佳造成 CDS 仍過大的情形。此外,本 文再提出 advance 方法, 進一步找出一個最小化的 CDS。實驗顯示我們所提出的 reassignment 方法因 為改善了法則 k 的缺點,得到較法則 k 小的 CDS。 隨後再執行 advance 方法,則可以找到一個 gateway 數較其他演算法少的 CDS , 特別是在行動裝置無 法提供較大的傳輸半徑時,效果更為明顯。所以, 我們所提出的 RCDS 演算法,的確可以獲得一個 較小的 CDS,以便建立一個精簡的 virtual backbone 來減少儲存的繞徑資訊。

然而,在 ad hoc 無線網路的行動裝置,都具

有低電量的限制,如何在有限的電力下,去延長網路的生命週期,也是個重要的議題。未來我們將思考加入電力條件的影響,以及如何在 ad hoc 無線網路中去延長網路的生命週期。

六、參考文獻

- [1] K. M. Alzoubi, P. J. Wan, and O. Frieder, "New Distributed Algorithm for Connected Dominating Set in Wireless Ad Hoc Networks," in Proc. 35th Hawaii International Conference on System Sciences, pp. 1-7, 2002.
- [2] J. Broch, D.Johnson, and D.Maltz, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks," IETF, Internet Draft, http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-00.txt, 1988.
- [3] I. Cidon and O.Mokryn, "Propagation and Leader Election in Multihop Broadcast Environment," in Proc. 12th International Symposium on Distributed Computing (DISC98), pp.104-119, 1998.
- [4] F. Dai and J. Wu, "Distributed Dominant Pruning in Ad Hoc Wireless Networks," in Proc. IEEE International Conference on Communications (ICC), pp.353-357, 2003.
- [5] B. Das and V. Bhargavan, "Routing in Ad-Hoc Networks Using Minimum Connected Dominating Sets," in Proc. IEEE International Conference on Communications (ICC '97), pp.376-380, 1997.
- [6] M.Garey and D.Johnson, "Computers and Intractability: A Guide to the Theory of NP-Completeness," Freeman, 1997.
- [7] C. Hedrick, "Routing Information Protocol," Internet Request for Comments (RFC) 1058, http://www.faqs.org/rfcs/rfc1058.html, 1988.
- [8] W. K. Liao, Y.C. Tseng, and J.P. Sheu, "GRID: A Fully Location-Aware Routing Protocol for Mobile Ad Hoc Networks," Telecommunication Systems, A Special Issue on Wireless Networks, vol. 18, pp.37-60, 2001.
- [9] T. Lin, S. F. Midiff, and J. S. Park, "Minimal Connected Dominating Set Algorithms and Application for a MANET Routing Protocol," in Proc. IEEE International Performance, Computing, and Communications Conference, pp.157-164, 2003.
- [10] J. M. McQuillan, I. Richer, and E. C. Rosen, "The New Routing Algorithm for ARPANET," IEEE Trans. Commun., vol. 28. no 5, pp. 711-719, 1980.
- [11] J.M. McQuillan and D.C. Walden, "The ARPA Network Design Decisions," Computer Networks, vol. 1, no. 5, pp. 243-289, 1977.
- J. Moy, "OSPF Version 2," Internet Request For Comments RFC 1247, http://www.faqs. org/rfcs/rfc1247.html, 1991.

- [13] C. Perkins and E. M. Royer. "Ad Hoc on Demand Distance Vector Routing," in Proc. 2nd IEEE Workshop on Mobile Computing Systems and Applications, pp. 90-100, 1999.
- [14] C.E. Perkins and E.M. Royer, "Highly Dynamic Destination Sequenced Distance Vector Routing (DSDV) for Mobile Computers," in Proc. ACM Specical Interest Group on comm.. (SIGCOMM '94), pp. 234-244, 1994.
- [15] P. R. Sheu and Y. P. Lee, "On Calculating Stable Connected Dominating Sets Base on Battery Power for Mobile Ad Hoc Networks," in Proc. International symposium on Communications, 2003.
- [16] I. Stojmenovic, M. Seddigh, and J. Xunic, "Dominating Sets and Neighbor Elimination Based Broadcasting Algorithms in Wireless Networks, " IEEE Trans. Parallel and Distributed Systems, vol. 13, no. 1, pp.14-25, 2002.
- [17] J. Wu and H. Li, "On Calculating Connected Dominating Set for Efficient Routing in Ad Hoc Wireless Networks," in Proc. 3rd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, pp. 7-14, 1999.
- [18] J. Wu, M. Gao, and I. Stojmenovic, "On Calculating Power-Aware Connected Dominating Sets for Efficient Routing in Ad Hoc Wireless Networks," Journal of Communications and Networks, vol. 5, no. 2, pp.169-178, 2002.

● Non-gateway裝置 ● Gateway裝置

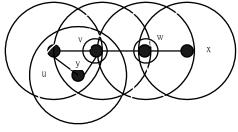
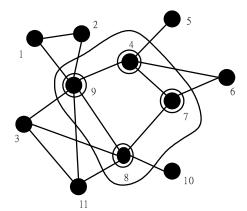


圖 1 ad hoc 無線網路的簡單圖形



Ad hoc 無線網路的例子(a)



Gateway 8 支配成員(b)

目的端	支配的成員	到目的端的	距離
		下一個端點	(路程)
9	{1,2,3,11}	9	1
4	{5,6}	7	2
7	{6}	7	1

Gateway 8 繞徑表格(c)

圖 2 繞徑範例

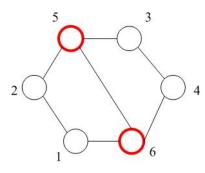


圖 3 法則 k 在節點 id 位置好的情形

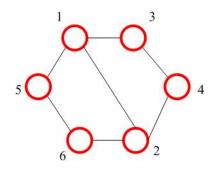


圖 4 法則 k 在節點 id 位置差的情形

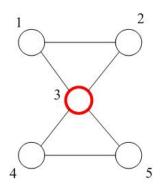


圖 5 neck node 的例子

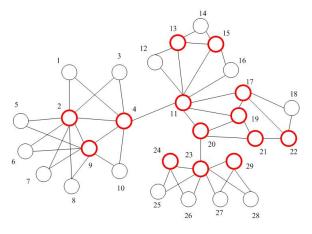


圖 6 使用標記方法後的 CDS

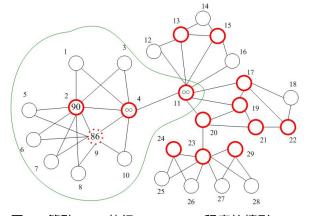


圖 7 節點 2、9 執行 reassignmet 程序的情形

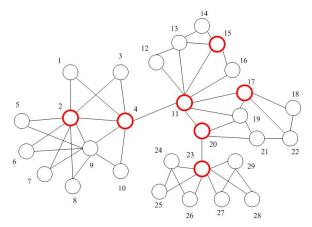


圖 8 由 reassignment 程序所產生的 CDS

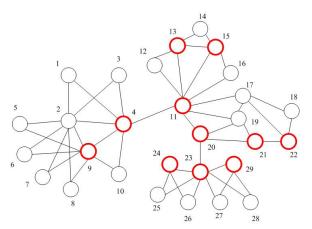


圖 9 由法則 k 所產生的 CDS

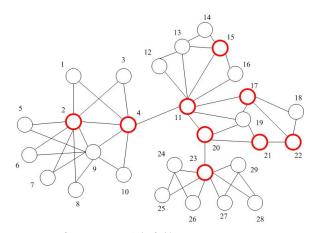


圖 10 由 Wu-ND 所產生的 CDS

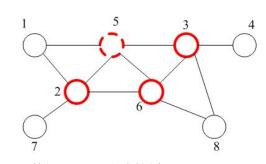


圖 11 執行 advance 程序的例子

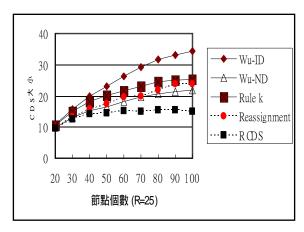


圖 12 各方法所產生 CDS 的大小 (傳輸半徑 R=25)

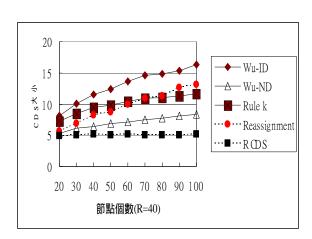


圖 13 各方法所產生 CDS 的大小 (傳輸半徑 R=40)

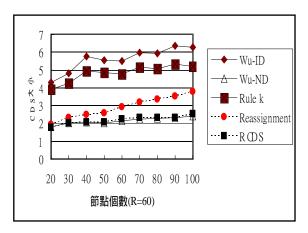


圖 14 各方法所產生 CDS 的大小 (傳輸半徑 R=60)

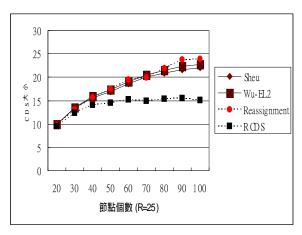


圖 15 各方法所產生 CDS 的大小 (傳輸半徑 R=25)

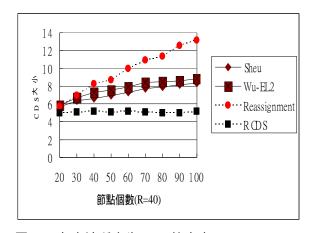


圖 16 各方法所產生 CDS 的大小 (傳輸半徑 R=40)

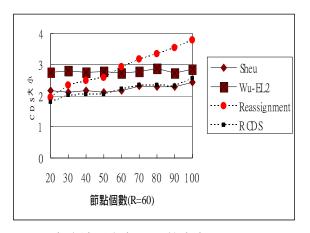


圖 17 各方法所產生 CDS 的大小 (傳輸半徑 R=60)