

在電池式無線網狀網路上做最大化總收益並以基因演化為基礎之群播繞徑研究

王培衡、楊文霖

國立台南大學資訊工程學系

xc4869@hotmail.com, wlyang@mail.nutn.edu.tw

摘要

隨著近年來無線網狀網路(WMNs, Wireless Mesh Network)的蓬勃發展,相關的研究及應用也逐年的在增加,由於群播(Multicast)傳輸的方式在 WMNs 架構中可以同時提供影音串流、即時多媒體等服務給大量特定的終端使用者,因此在 WMNs 中無干擾的群播路由應用儼然成為電池式多頻道多網卡 WMNs 最重要的問題之一,由於節點的電量會隨時間消耗而中斷傳輸服務,傳統的研究在探討電池式無線網路時主要著墨在如何拉長整體的網路維持時間,本文探討如何將服務用戶數 N 及整體維持時間 T 做最大化,假設在商業行為中每秒服務一位客戶可收取一單位利潤,可知若 $T * N$ 越大,則所獲得的收益就會越多。因此本文提出以基因演化為基礎之演算法,並自定義函數 Total Profit 用來檢驗演算法在建立群播樹時其總收益的高低,在各節點的電量會隨時間消耗及無線網路頻道數量有限的情況下,本文所提之演算法可建立無干擾的群播路由,同時在維持整體網路時間及最大化服務用戶數兩方之間取得平衡,進而得到最大化總收益之效,最後並藉由實驗結果證實以本文所提出之演算法建立群播路由其效能優於其他現有的方法。

關鍵詞：無線網狀網路、多頻道、頻道分配、最大化收益群播樹。

Abstract

In this paper, we consider the following scenario: an interference-free multicast tree is required to be deployed on a battery-powered multi-channel multi-radio wireless mesh networks, so that a video stream can be delivered to several destinations where a number of clients may subscribe the stream simultaneously. For various multicast routing problems studied on the battery-based ad hoc wireless networks, the traditional design aim is to maximize the life-time of the network. In our problem, however, we are required to maximize both the life-time T and the number serviced clients N at the same time. Given one unit of profit per

second per client, it would be more reasonable to maximize $T * N$ instead of T for the owners of networks to have maximum profit. Hence, to support this business model, a term called profit which is defined to be $T * N$ is used as the optimization goal for constructing the multicast tree. A genetic-based algorithm is devised in this study to solve this problem. According to our experimental results, the GA-based approach can significantly outperform the other previously proposed methods.

Keywords: *Wireless Mesh Network*、*Multi-channel*、*Channel Assignment*、*Profit Maximization of Multicast Tree*。

一、前言

隨著近年來手提式行動裝置的持有者大幅增加,無線網狀網路的重要性也因此日益攀升,舉凡像是影音串流、文書資訊、即時通訊傳輸等服務需求量大增使得無線網路使用量逐年成長,由其是台灣的各個電信業者在今年初時開始積極於取得 4G 執照以提升更多的競爭優勢,在不久的未來透過 4G 此技術標準,用戶可因此享受更多的服務種類以及更高的服務品質。無線網狀網路擁有一般傳統的纜線網路所沒有具備的優點,包括像是傳輸時不受纜線的長度及地形所限制、硬體的建置因無線網路訊號涵蓋範圍大而相對容易、具自我組織能力等優點,有鑑於此,無線網狀網路的在未來的發展相較於有線網路會更趨於主流。

現今主要以 802.11 為基礎的 WMNs 無線技術,其具有大範圍的無線訊號涵蓋面積、建置相對於傳統的有線網路較為容易、自我組織能力強等優點,通訊時不受纜線長度及最後一哩等限制,建置成本比起實體纜線需另外考慮到地域遠近、地形高低起伏因素相對較低,故較易在較偏遠的地區建置和普及[1]。但偏遠地區的 WMNs 路由架構內的節點,也就是無線路由器(MR, Mesh Router)可能會因其地域關係,補充電源的方式不易導致通訊時因節點電量耗盡而中斷。

在無線傳輸中,可分成單一傳播(Unicast)

及群體廣播(Multicast)，由於單播傳輸容易導致網路發生壅塞，在個別傳輸的同時也浪費相當多的頻寬，而群播除了可以在 WMNs 同時提供傳輸影音串流、即時傳輸等服務給大量特定的終端用戶，故在 WMNs 上使用群播路由應用可以節省更多的頻寬進而提供更佳的服务。

除此之外 WMNs 所使用的無線傳輸協定，以 802.11b/g 為例，雖共有 11 個頻道，但正交頻道只有第 1、6、11 共三個，由於資源相當有限，故如何有效的分配無干擾的頻道給使用者一直都是近年來熱門的研究議題，其中頻道分配的方式可分為先建立群播樹後再行分配；或是在建立群播樹的過程中同時進行頻道分配。比起傳統作法通常都是只針對可用的三個正交的頻道進行分配，本文之演算法使用建樹與分配頻道同時進行的方式建立整個群播無線網狀網路架構[5]，同時分配正交與非正交頻道，使得整體頻道分配的靈活度大為提升。

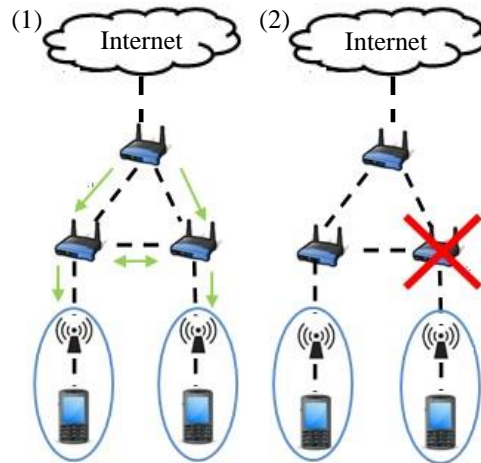
本文主要的目標在於探討如何從無線網路會隨時間消耗的前提下盡可能的減少電量消耗以維持整體網路的服務時間，同時服務更多的用戶數，但在現實情況下每個網路節點的電量不一，且各節點隨著服務需求的用戶數不定，消耗電量的速度亦不相同，再者當整體網路的使用戶越多，消耗的電量也會隨之增加，反之亦然。傳統在探討電池式無線網狀網路的研究中大部分著重在如何拉長整體網路的維持時間，而本文的目標在於如何在增加用戶數量及提升整體網路的維持時間得到一個平衡，也就是整個無線網路可以在無干擾的環境下最大化整體效益，有鑑於此，本文提出自定義函數 Total Profit 來探討演算法在建立群播樹時可得的效益多寡，其可表示為：

$$\text{Total profit} = \text{Clients} * \text{Life time} \dots\dots\dots (1)$$

此函數之效能評比定義為在無干擾的環境下被服務的用戶數(Clients)乘上整體網路的最短維持時間(Life time)，用以檢測一個群播路由架構所能獲得的收益高低，若此數值越高則代表該網路架構所獲得的收益就會越高，反之則越低，舉例來說，今天某一 WMNs 結構提供無線網路服務給用戶來獲取商業利潤，假設每秒服務一位用戶可以獲取一單位的利潤，若供給越多的用戶數越長的時間，則在兩者相乘之下所得整體的總收益就會隨之提升。另外在 Total Profit 數值高的情況下有可能是因為只服務極少的用戶卻服務極長的時間，為了避免以上狀況發生導致 Total Profit 數值雖高卻不符合實際應用之狀況，本文要求被服務之用戶數需大於整體用戶數的一定比例以上來維

持整體服務的品質。

在提供服務的網路最短維持時間 Life time 部分，因假設各個節點電量不一的關係且各節點所服務的用戶數不同，雖然在單一 MR 節點因電量耗盡停止運作後其他節點的用戶依然可以繼續使用網路服務，但為了計算 Total Profit 總收益的便利性，當網路結構中只要有其中一個 MR 節點電量耗盡而無法正常運作時則終止整個網路結構的服務，從開始提供傳輸服務到中斷全體網路的這段時間計為最短維持時間，如圖一。



圖一、(1)正常無線網狀網路運作架構
(2)因其中一個 MR 電量耗盡，故中斷整個網路傳輸以利總收益的計算

另外本文使用了負載值 load 的觀念，將要求服務的用戶數當作 demand 需求數，當 demand 值越大則代表該 MR 節點所連接需要服務的用戶數越多，而負載值利用 demand 來計算，計算方式如圖五，其優點在於該數值與實際用戶數成正比，可藉此了解實際連接用戶數之多寡，來幫助演算法做路由尋徑做最佳化選擇時的一個依據。

同時本文也介紹其他三種不同之建立群播路由之演算法，後面實驗的章節中也會在 MR 節點的電量有限及隨時間消耗的前提下與本文所提出之演算法做效能的比較。

1.1 Profit-Oriented Multicast (POM) Tree

與傳統研究在探討群播路由時設定 MR 節點擁有無限電量的情況不同，WMNs 內的節點會隨傳輸時間而消耗，同時節點所連結的用戶數越多，越加速節點電量的消耗，進而影響到整體網路的維持時間，故整個群播路由的維持時間及服務的用戶數兩者間的平衡就顯得格外的重要。POM 演算法[8]的步驟如下：

- (一) 首先輸入一無向圖 $G(V, E)$ ，由根節點開始，利用廣度優先搜尋方式尋找每個節

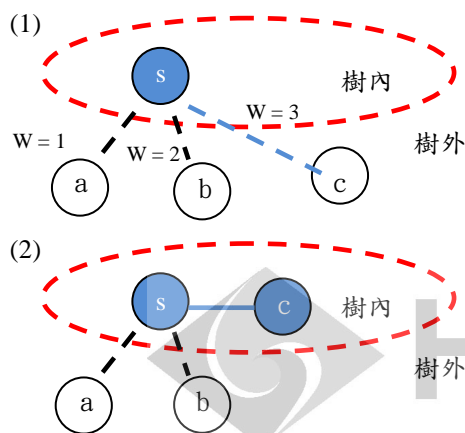
點並賦予 level 值後將 G 轉換成有向圖 $G'(V', E')$ ，接著從最底層 max level 開始依圖五方式計算每個節點的負載值 load 後刪除 G' 中所有同 level 的連結。

- (二) 定義節點 $u, v \in V'$, $level(u) < level(v)$ ，從根節點開始檢查圖 G' 中節點 u 連向節點 v 的每條 $link(u \rightarrow v)$ 連結，依據各個節點 u 電池的容量、 $link(u \rightarrow v)$ 傳輸時所消耗的電量，並使用負載值 load 的概念透過公式(2)得到權重值 weight：

$$Weight = load(u) * \frac{energy\ reserve\ \mathcal{E}(u)}{power\ dissipation\ of\ link\ (u \rightarrow v)} \dots (2)$$

- (三) 建樹時從根節點開始透過尋找群播樹鄰近的連結，假設群播樹內節點為 x ，樹外節點為 y ，尋找所有鄰接群播樹的連結 $link(x \rightarrow y)$ ，並選擇權重值最大的連結來做頻道分配時的檢查。
- (四) 使用動態分配頻道的方式，在建立群播樹的過程中分配無干擾的頻道，如圖二，若發現有可分配的無干擾頻道，則在加入連結到群播樹後進行分配無干擾的頻道動作，若無則刪除掉該連結並檢查其他鄰接群播樹的連結。
- (五) 重複步驟(三)、(四)直到找不到鄰接群播樹的連結或是所有目標節點都已被加入到群播樹中，則進入到步驟(六)。
- (六) 輸出以分配完頻道的群播樹。

此演算法著重在電量隨時間消耗的前提下最佳化負載值與網路整體維持時間兩者的平衡，盡可能最大化兩者的數值，進而達到最大化總收益之效，並要求群播路由內所連結的目標節點須占整體節點一定比例以上來符合現實狀況的需求。



圖二、(1)尋找樹外權重值最大的節點
(2)若找到的節點可以分配無干擾頻道，則將其加入到群播樹中

1.2 Multicast Incremental Power (MIP) Tree

在 WMNs 的每個節點裝置因其電池的容量不一，整體網路的運作持續時間取決於個別節點電量耗盡之時間，因節點的總電量在建樹一開始時就已決定無法補充，若能用更有效率的建樹方式節省每個節點電量消耗，則可以變相的增加整體網路的服務時間，故希望藉由演算法減少整體網路所消耗的電量來維持總傳輸時間，本演算法之優點在於會盡量減少加入會消耗大量電量的節點，使整體群播樹的規模縮小進而達到節省能量消耗的目標，但另一方面卻也因此限制了加入節點的數量。

根據[6]，BIP 演算法相似於 Prim 演算法，若要得到 MIP 樹得先行建立 BIP 樹，而 BIP 樹建立需利用 wireless broadcast advantage(WBA)[5]的特性，在節點群中以 source 為原點建樹，檢查樹中及尚未加入樹的節點並算出加入 $link(i, j)$ 時所要增加的消耗 P'_{ij} ，其可表示為：

$$P'_{ij} = P_{ij} - P(i) \dots \dots \dots (3)$$

P_{ij} 表示 $link(i, j)$ 本身所需的能量， $P(i)$ 表示目前處於傳輸狀態中的節點 i 所消耗的能量水平， $P(i) = 0$ 代表節點 i 為樹的葉節點，若 P'_{ij} 小於等於 0 則代表加入節點 j 不用另外消耗電量，並選擇消耗 P'_{ij} 最小的節點加入樹中直至所有節點都已被加入到廣播樹。而 MIP 樹則由 BIP 樹建立廣播樹後根據需求削減樹中的節點或支幹得到群播樹。

1.3 Best-First + Backtracking (BFB) Tree

在多網卡環境建立群播樹時，由於在單一節點可以同時分配兩種或以上不同的頻道，使得建樹後的頻道分配更具彈性，在本文考量到整體網路需在能量有限的情況下運作，採先建樹後分配頻道的方式完成多頻道群播樹 Multi-Channel Multicast (MCM) [9]，而本文使用從 MCM 衍生出的 LMCM[7]，其與 MCM 的差異是 LMCM 在建立 MCM 時是以負載值 load 做為加入群播樹節點的依據。一般來說，因頻道資源有限，傳輸資料可能會因相鄰或相接的連結而產生干擾，若中繼節點的數量越少，則越可以避免干擾情況發生，故建立群播樹時採用最小化中繼節點與最小化來源節點到目標節點所需經過節點數量的方式來減少頻道干擾的情況發生。

在建立 LMCM 時，首先在終端節點設定 demand 數 1~5 不等之數值來當作實際需要服務的用戶數，然後由根節點開始，利用廣度優先 BFS 搜尋方式給予每個節點一個 level 值，把同 level 的連結去除後計算由最底層開始每

個節點的負載值，接著由根節點開始選擇鄰近負載值最大的節點並加入到群播樹中，然後再重覆檢查鄰近的節點直至所有終端節點都已被加入到群播樹中。

之後利用 Best-First[7] + Backtracking 方式分配頻道。首先把樹的 root 點設為父節點後，檢查同父節點的相鄰連結中是否可以分享頻道給自身連結，若無則逐一檢查剩餘可用的頻道是否會和鄰近或相接的連結的頻道發生干擾，然後將各個子節點當作父節點重複上述步驟逐步將各連結分配頻道並記錄整個群播樹的負載值，而後使用 Backtracking 尋找其他可能改善負載值的頻道分配方式，藉由逐一調整每條連結的頻道，來尋找是否有其他連結因此由干擾變為無干擾的狀態並加入到樹中，逐一檢查完每條連結後，去除未被分配頻道的連結、節點，形成一棵已分配完頻道之 MCM 樹。

二、問題定義

本文主要透過多頻道多網卡無線網狀網路(MCMR WMNs)架構，希望透過自定義函數 Total Profit 方式來做為評定建構整體無線網狀網路架構之演算法效能的依據，而在評定 WMNs 效能前須先滿足以下假設：

- (一) 由於在相鄰節點中分配相同頻道容易產生頻道間相互干擾的情況，故每個群播路由中的 MR 節點皆配置兩張或以上之無線網路卡，並使用 802.11b/g 做為無線傳輸通訊標準，其共有 11 個可用頻道，其中第 1、6、11 為正交頻道，以達成多頻道多網卡之環境目標。
- (二) 無線環境下的 MR 節點分佈及用戶需求數 demand 皆為隨機。
- (三) 若要計算群播網路的效益，則此 WMNs 群播樹所服務的用戶需在同一時間內盡可能服務特定的最大用戶數。
- (四) 當群播樹中有一個 MR 節點因電量耗盡時，為了本文計算網路維持時間的便利，會中斷整個網路的傳輸服務，在開始網路服務到其中有任一節點發生電量耗盡情況這段時間為網路最短維持時間。
- (五) 在所有提供用戶服務的過程中全都必須處於無干擾的環境下，以保證所有用戶的網路服務品質。
- (六) 原則上用戶數越多，群播網路會因電量的消耗增加而縮短整體的維持時間，反之則維持時間越長，故群播網路架構在使用服務中的用戶數需到達一定比例以上，避免因服務人數過少但 Total Profit

數值高這種不符實際應用之情況發生。

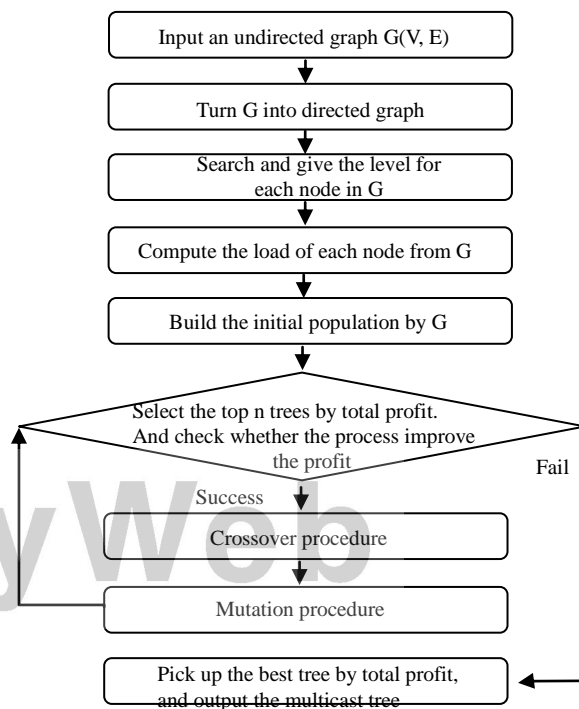
- (七) 所有的 MR 節點電量都不盡相同，其電量皆介於 10~20 mAh(毫安培/小時)之間，但傳輸功率及資料傳輸率皆是相同的，以符合現實環境的需求。

針對以上幾點，本文提出以基因演化為基礎之演算法，其建立群播樹的過程中會分配無干擾的頻道，在 MR 節點電量會隨傳輸時間消耗的同時將最短維持時間及用戶數量做最大化處理，來達成最大化總收益 Total Profit(1) 之目標

三、Profit-Oriented Genetic Algorithm

本文所提之最大化總收益基因演算法，會根據各節點的負載值 load 大小隨機選取連結並動態分配無干擾頻道來建立初始群播樹，負載值高的連結可能被選到的機率較高，再透過基因演化的流程，將 Total profit 高的樹保留以供下個世代繼續演化，當滿足該流程的終止條件，即經過幾個世代其總效益最高的群播樹都沒有進一步的改善時，便會中斷流程並輸出世代中總效益最佳、Total profit 最高的群播樹。

而基因演化的流程，主要為 Crossover、Mutation 程序，在進行這些過程的前皆須先確保所有經過處理前後的群播樹都是負載值大於零且已分配完無干擾頻道，之後會依(三)來選擇效益佳的群播樹進行下一代的世代流程，以下圖三為此演算法之流程圖，圖四為完整演算法。



圖三、演算法流程圖

Input: $G(V, E)$, destination set $M = \{v \mid v \in V, demand(v) > 0\}$, gateway node s ;

1. Make $G(V, E)$ being a directed graph by replacing an undirected edge with two oppositely directed links;
2. Starting from the gateway node s , a breadth-first traversal procedure is called to assign each node a *level* value;
3. A backward link ($u \rightarrow v$) is deleted if $level(u) > level(v)$;
4. Compute load value for each node by calling *Load_computation*(G) procedure;
5. Set N = number of feasible multicast trees in one population;
6. Let $I = \emptyset$ be the initial population set ;
7. for ($i = 1$ to N) {
8. $I = I \cup Random_Tree(G, M, s)$;
9. compute profit η for each tree in I ;
10. Let $S = I, T_{best} = \emptyset, \eta_{best} = 0, counter = 0$;
11. Let p_a denote the mutation probability ;
12. Let *threshold* denote the maximum generations during which no improvement is made;
13. for ($i = 1$ to *max_generations*) {
14. \bar{S} = select the best N trees from S by η ;
15. $\tau = \emptyset$;
16. for($k = 1$ to $N/2$) {
17. randomly select two trees t_1 and t_2 from \bar{S} ;
18. $\{\bar{t}_1, \bar{t}_2\} = CrossOver(t_1, t_2)$;
19. $\tau = \tau \cup \{\bar{t}_1, \bar{t}_2\}$;
20. }
21. $\bar{S} = \bar{S} \cup \tau$;
22. for($k = 1$ to $N * p_a$) {
23. randomly select a tree t from \bar{S} ;
24. $\bar{t} = Mutation(t)$;
25. $\bar{S} = \bar{S} \cup \bar{t}$;
26. }
27. compute profit η for each new tree in \bar{S} ;
28. T_p = the tree with the largest value of η_p in \bar{S} ;
29. if($\eta_{best} < \eta_p$) {
30. $\eta_{best} = \eta_p$; $T_{best} = T_p$;
31. $counter = 0$; }
32. else if($counter > threshold$) break;
33. else $counter++$;
34. }
35. output T_{best} ;

圖四、基因演算法

輸入：一無向圖形 $G(V, E)$ 並定義圖中 demand 大於零的節點為目標節點，以及 gateway 節點 s 。

- (一) 先將 G 轉換成有向圖，將圖中邊置換成兩個方向相反的連結後，由節點 s 開始利用 BFS 搜尋方式搜尋每個節點並給予 level 值，接著將每個連結中 level 值大的點連向 level 值小的點之連結給刪除，藉此來簡化整個網路結構。
- (二) 透過 *Load_computation* 程序(如圖五)將每個節點加入一負載值，負載值越高的

節點代表其後連結到的目標節點越多。

- (三) 假設 N 為初始族群的個體數量， I 為初始族群的集合，利用 *Random_Tree* 程序產生 N 個已分配完頻道的群播樹，將樹加入到 I 並計算 I 中每棵樹的效益值 η 。
- (四) 設定門檻值及新的族群 \bar{S} ，比較本次世代與前面世代所得效益最高的樹的總收益值，若發現經過一定次數的世代演化後仍然無法獲得改善時，則終止整個程序，門檻值的大小決定幾代演化未改善則跳出程序的中止條件。
- (五) 透過效益值 η 高低選擇上個世代中族群內效益最佳的前 N 棵樹來組成本次世代新的族群，並在新族群內任選兩棵樹做 *CrossOver* 程序產生新的兩棵已分配過頻道的群播樹，共執行 *CrossOver* 程序 $\frac{N}{2}$ 次來產生出另外 N 棵新的群播樹加入到族群 \bar{S} 中。
- (六) 設定突變機率為 p_a 並隨機從族群中挑選一棵樹做 *Mutation* 程序來產生出一新的已分配過頻道的群播樹，共執行 $N * p_a$ 次來產生出另外 $N * p_a$ 棵新群播樹加入到族群 \bar{S} 中。
- (七) 重新計算 \bar{S} 中每棵群播樹的效益值 η ，並檢查是否滿足步驟(四)的終止條件，若有則進入到步驟(八)，若無則回到步驟(五)繼續進行流程。
- (八) 輸出族群中效益值最高的群播樹。

```

Load_computation (G) {
1. Let maxLevel =  $\max_{u \in V} level(u)$ ;
2. Let  $load(u) = demand(u)$ , for each leaf-node  $u \in V$ ;
3. For ( $l = maxLevel; l \geq 0; l = l - 1$ ) {
4.   For (each node  $u$  with  $level(u) = l$ ) {
5.     If (node  $u$  is adjacent to node  $v$ ) {
6.       If ( $level(v) < level(u)$ ) {
7.          $load(v) = load(v) + load(u)$ ; }
8.       Else if ( $level(v) == level(u)$ ) {
9.          $load(v) = load(v) + demand(u)$ ; }
10.    } }

```

圖五、Load_computation 程序

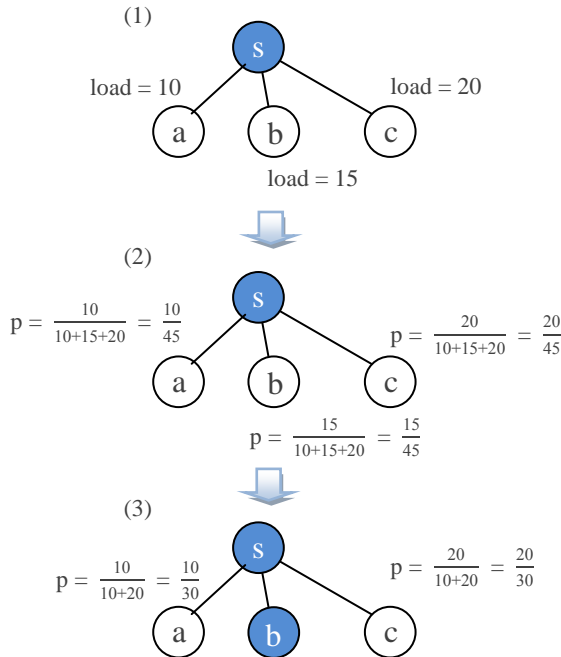
3.1 Random_Tree 程序

本文以圖六為例來說明圖七之 *Random_Tree* 程序。

- (一) 建立群播樹 $T = \{\bar{V}, \bar{E}\}$ $\bar{V} = \{s\}$ $\bar{E} = \emptyset$ 。
- (二) 搜尋所有樹內連向樹外之連結，根據每條連結所連向樹外節點之負載值來計算每條連結被選取的機率 p ， $p =$

該連結所連之樹外節點。
鄰近樹外節點負載值總和

- (三) 透過 p 隨機選取一鄰接的連結，並檢查是否有一無干擾頻道可分配，若有則將該連結加入到群播樹 T 中，若無則在剩下的連結中再次隨機選取一鄰接連結。
- (四) 重複步驟(二)、(三)直至所有目標節點都被加入到樹中或是已搜尋不到鄰近的連結則進入到步驟(五)。
- (五) 輸出已分配完頻道之群播樹。



圖六、(1)由樹內節點開始，找尋鄰近連結。
(2)根據鄰近節點負載值總和，算出機率 p 。
(3)根據機率 p 隨機選擇連結，加入群播樹並重新計算鄰近節點之機率 p 。

```

Input:  $G(V, E), M = \{v | v \in V; demand(v) > 0\}, s;$ 
1. Make gateway  $s$  be the root of tree  $T = \{\bar{V}, \bar{E}\};$ 
2.  $\bar{V} = \{s\}, \bar{E} = \emptyset;$ 
3. while (true) {
4.    $L = \{(u, w) | u \in \bar{V}, w \in V - \bar{V}\};$ 
5.   while( $L \neq \emptyset$ ) {
6.     For each link  $(u, w) \in L$  {
7.        $p = load(w) / \sum_{(x, y) \in L} load(y);$ 
8.     }
9.     select a link  $(u, w) \in L$  based on  $p;$ 
10.    allocate an interference-free channel  $\alpha$  to link $(u, w);$ 
11.    if( $\alpha = \emptyset$ )  $\{L = L - \{(u, w)\};\}$ 
12.    else {
13.       $\bar{V} = \bar{V} \cup \{w\};$ 
14.       $\bar{E} = \bar{E} \cup \{(u, w)\};$ 
15.      break; }
16.   }
17.   if( $L = \emptyset$ ) return  $T;$ 
18.   if( $M \subseteq \bar{V}$ ) return  $T;$ 

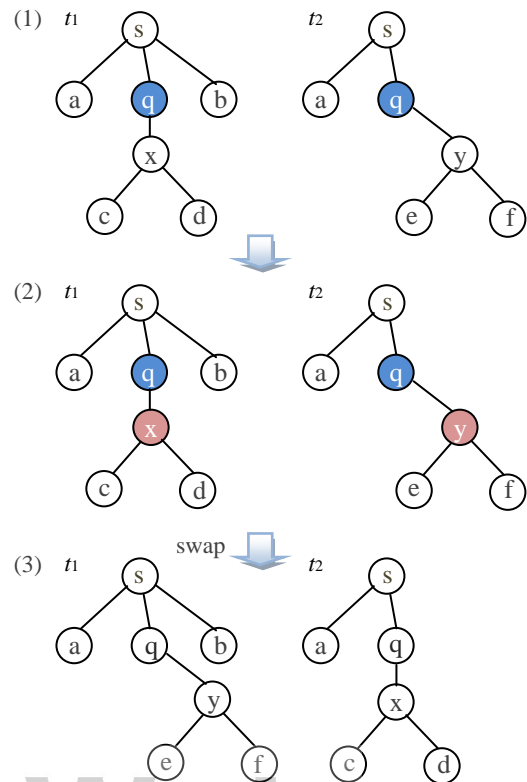
```

圖七、Random_Tree 程序

3.2 CrossOver 程序

本文以圖八為例來說明圖九之 CrossOver 程序。

- (一) 從隨機挑選的兩棵已被分配頻道的群播樹 t_1 及 t_2 中找尋相同的內部節點 q ，並從這些節點中隨機挑選一個出來。
- (二) 各自從 t_1 及 t_2 中根據剛挑選出來的內部節點 q 隨機找尋各自的子節點 x, y 。
- (三) 根據步驟(二)選出來的子節點 x, y 以 q 為中心進行子樹的交換。
- (四) 交換子樹後，重新對新產生出來的群播樹做頻道分配，由節點 q 開始往下檢查新交換後子樹的各個連結，若可以找到一無干擾的頻道分給該連結，則將其保留，若不能則刪除掉該連結。
- (五) 檢查新子樹的每個連結其頻道的分配情況直至所有連結都已檢查完畢。
- (六) 輸出已分配完頻道的新群播樹 t_1 及 t_2 。



圖八、(1)找出兩棵樹中相同的內部節點並在其中隨機選擇一個節點 q 。
(2)根據(1)由節點 q 隨機選擇各自的子節點 x, y 。
(3)透過 x, y 來交換子樹，並檢查新子樹的頻道分配狀況後輸出新子樹 t_1 及 t_2 。

Input: channel-allocated tree t_1 and t_2 ;

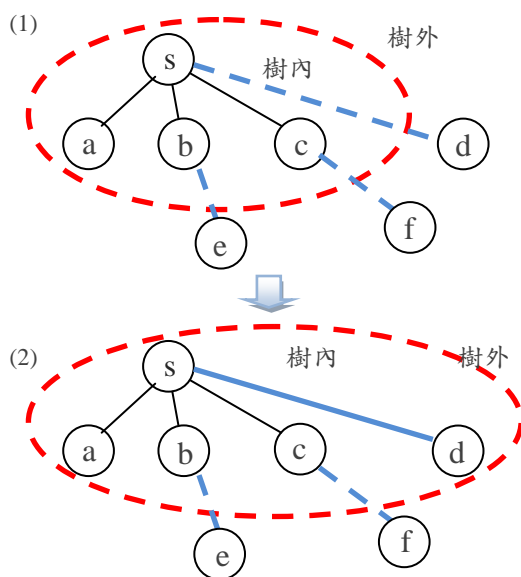
1. Randomly select a common internal node q from t_1 and t_2 ;
2. Randomly select link (q, x) in t_1 , and link (q, y) in t_2 ;
3. swap the sub-tree root at x and the sub-tree rooted at y ;
4. Reallocate interference-free channels to these two sub-trees in a top-down fashion, and delete those links with no channel allocated.
5. return t_1 and t_2 ;

圖九、CrossOver 程序

3.3 Mutation 程序

本文以圖十為例來說明圖十一之 Mutation 程序。

- (一) 搜尋群播樹 t 所有樹內連向樹外的連結。
- (二) 依序檢查每條連結是否有一無干擾頻道可分配，若有則進入到步驟(三)，若無則刪除掉該連結並檢查其他的連結。當所有的連結都被檢查完時進入步驟(四)。
- (三) 檢查加入連結後效益值 $profit$ 是否比加入前還要高，若有則加入該連結到樹中，若無則回到步驟(二)。
- (四) 輸出已分配完頻道的群播樹 t 。



圖十、(1)搜尋每條樹內連向樹外的節點。

(2)檢查每條連結是否有無干擾的頻道可分配及是否加入連結後效益更高，若兩者皆是則將該連結加入到群播樹中。

Input: A channel-allocated tree t ;

1. Let $L = \{(u, w) | u \in t, w \notin t, load(w) > 0\}$;
2. for (each link $l \in L$) {
3. for (each channel η allocated to l) {
4. $\bar{t} = t + \{l\} - \{h | h \in t, h \text{ interferes with } l\}$;
5. if($profit(\bar{t}) > profit(t)$) $t = \bar{t}$;
6. }
7. return t ;

圖十一、The Mutation 程序

四、實驗結果

本文實驗可分為 Java 及 Qualnet 兩個平台來模擬現實情況並藉此評比各演算法之效能，本文利用 Total profit 此自定義函式來比較各演算法之優劣，並使用負載值 Load 來代表被服務的用戶數，而除了本文所提出之演算法外，另使用其他演算法來做效能方面的比較，模擬環境之硬體設備明細如表一。

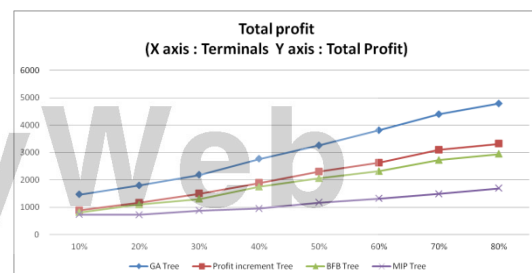
表一：模擬環境之硬體設備

CPU	Intel(R) Xeon(R) 3.2GHz
核心數	8 核心
作業系統	Windows 7
系統位元數	64 位元
記憶體	4.0 GB

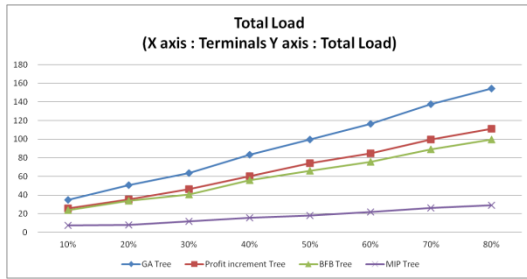
4.1 Java 實驗

在一個 $100*100$ 的平面圖中隨機散布各個節點，每個節點設定 1~5 不等的用戶數，在此以負載值 $load$ 做為代表，並在這些節點中隨機決定 terminal 節點，在整體用戶數為 100 個 nodes 的情況下 terminal 節點數依總用戶數的 10%~80% 做調整來進行實驗，圖中各個節點的電量不一且依據節點的現有電量除以所消耗的最大電量即為該節點所能維持的服務時間，而整體網路的維持時間為建立完無干擾的群播樹後在這些樹內節點中選擇服務時間最短的為主。

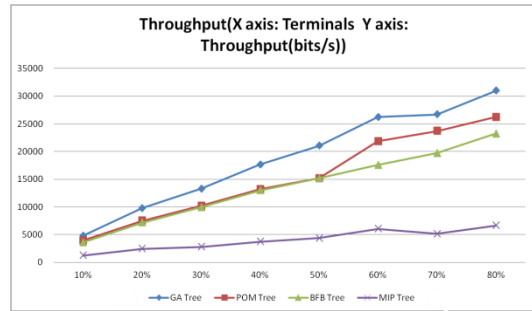
根據以上條件進行實驗，各個演算法之 Total profit 以及負載值 $load$ 之結果分別如圖十二及圖十三，可知本文所提之演算法效能優於其他演算法。



圖十二、各演算法之 Total Profit 結果，藍色線條部分為本文所提之演算法



圖十三、各演算法之負載值 load 結果，藍色線條部分為本文所提之演算法



圖十四、Qualnet 實驗所得之 Throughput 圖，藍色線代表本文所提出之演算法的 Throughput

4.2 Qualnet 實驗

Qualnet 平台可以模擬多網卡多頻道之無線環境，以及無線行動裝置和無線網狀路由節點的狀態，不僅可以設定個別裝置之數量、發送封包速率及封包大小，也可自由設定每個 MR 節點的電量，可以模擬出實際硬體設備需求大的群播路由傳輸時的現實狀況，故利用此平台來模擬各演算法建立群播樹時的狀況，並藉此取得本文所需要的數據。

實驗初始條件的資訊如表二，在 Java 程式模擬完之輸出檔案(App、Member、Node、Display、Config)並利用 Qualnet 進行模擬，結束後會產生 Static 檔，其會紀錄該次實驗之詳細數據，以下所有實驗數據皆為各演算法產生之二十顆群播樹結果來做平均而得。

由於此次實驗希望模擬傳輸資料時電力耗盡而中斷傳輸之情況，故設置較長的傳輸時間以確保傳輸任務進行到一半時部份節點就會因電量消耗完畢而中斷，本次實驗傳輸總封包數及傳輸時間分別設為 2000 個及 2000 秒，並用每秒鐘一個封包之速度進行傳輸，實驗初始條件如表二。

表二：實驗初始條件

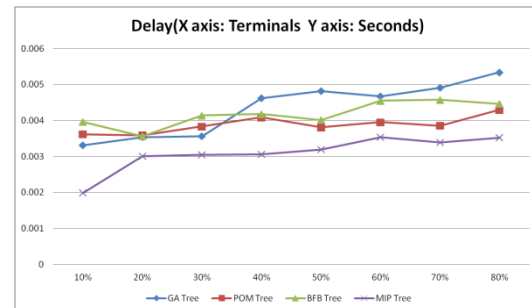
參數	設定
總結點數目	100 nodes
目標節點比例	10%~80%
各節點初始電量	10~20mAh 隨機給予
傳輸時間	2000 秒
總傳送封包數	2000 個
傳送封包速度	1 秒/個
封包大小	512 bytes

4.2.1 Throughput 吞吐量

其代表整個群播樹之目標節點所接收到的資料量大小，由於各演算法模擬之傳輸速度及傳輸的總封包量是固定不變的，所以在無干擾的環境下其數值與連接到的目標節點數量成正比關係，本文之演算法與其他演算法之 Throughput 結果對照如圖十四。

4.2.2 最長延遲 Delay

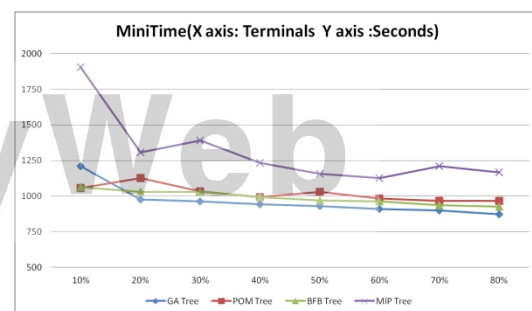
接著觀察每個演算法之最長延遲，其數值為每棵群播樹中比較樹內的各個連結並挑出延遲最長時間再取平均而得，結果如圖十五，可發現雖然延遲時間較長，但彼此間的差異不算太大，且由 4.2.1 之結果可知本文所提出之演算法可得較大之 Throughput 值。



圖十五、Qualnet 實驗所得之 Delay 圖，藍色線代表本文所提出之演算法的 Delay

4.2.3 最短維持時間 MiniTime

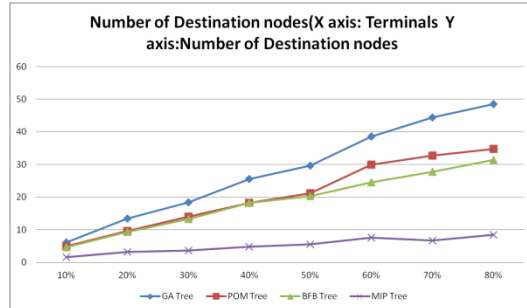
由於每個節點所設電量不一，且每棵群播樹結構不盡相同，故整體網路的維持時間也不一致，為了方便記錄相關數據，只要群播路由中有任一 MR 節點耗盡電量而停止運作，則中斷整個網路的傳輸，從建立完群播樹開始到中止傳輸服務這段時間即為最短維持時間，本次實驗所取得的時間數據為二十組各個演算法所得之群播樹最短維持時間的平均，其結果如圖十六所示。



圖十六、Qualnet 實驗所得之 MiniTime 圖，藍色線代表本文所提出之演算法的 MiniTime。

4.2.4 目標節點數量

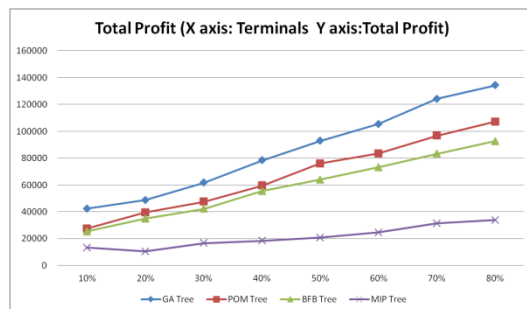
若節點數量越多，代表該群播樹之網路結構可服務的用戶數就越多，也就代表沒被服務到或遺失的用戶數相對較少，另一方面來說，可代表一個演算法所建立之群播樹它的規模大小，其模擬之結果如圖十七。



圖十七、Qualnet 實驗所得之目標節點數量圖，藍色線代表本文之演算法的目標節點數。

4.2.5 Total Profit 總收益

此數值代表一個演算法在建立有限電量網路架構且運行後所得到的總收益，數值越高代表該演算法之效能越佳，在前文中定義該數值為網路最短維持時間乘上接受服務的用戶數，故在此將實驗中分別自 4.3 及 4.4 所得 MiniTime 及目標節點數量相乘以得到 Total Profit 數值，其模擬結果對照如圖十八。



圖十八、Qualnet 實驗所得之 Total Profit 圖，藍色線代表本文之演算法的 Total Profit 數值。

五、結論

本文以基因演算法為基礎，結合跨層式頻道分配法進行以有限電量消耗為前提之無線網狀網路之研究，並使用 Java 及 Qualnet 兩套模擬軟體來進行多種演算法之實驗，透過本文自定義 Total Profit 效能評比函數及實際實驗數據證明本文所提之基因演算法效能優於其他演算法，顯示本文所提之演算法相較於其他方式，在建置可同時服務多用戶及在有限電量環境下維持更長時間之無干擾無線網狀網路架構可獲得更高之收益。

參考文獻

- [1] I.F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: a survey," Computer Networks, vol.47, pp. 445-487, 2005.
- [2] He Shiming, Zhang Dafang, Xie Kun, Qiao Hong, Zhang Ji, "A Simple Channel Assignment for Opportunistic Routing in Multi-radio Multi-channel Wireless Mesh Networks," IEEE CONFERENCE PUBLICATIONS, pp.201-208, 2011.
- [3] X. Li, N. Qiu, J.Y. Zhang and Y.Q. Liu, "An improved GA for QoS multicast routing algorithm," Control and Decision Conference, Issue 2, pp.393-396, July, 2008.
- [4] M. Peng, Y. Wang, W. Wang, "Cross-layer design for tree-type routing and level-based centralised scheduling in IEEE 802.16 based wireless mesh networks," IET Communications, pp. 999-1006, 2007.
- [5] Wen-Lin Yang, Wan-Ting Hong, "A Cross-Layer Optimization for Total-Revenue based Multicast in Multi-Channel Multi-Radio Wireless Mesh Network," Int. J. Commun. Syst., pp. 1-19, 2013.
- [6] Jefferet E. Wieselthier, Gam D. Nguyen, "Energy-Efficient Broadcast and Multicast Trees in Wireless Network," Mobile Network and Applications 7, pp. 481-492, 2002.
- [7] W.L. Yang and W.T. Hong, "a Study on Channel Allocation Strategies for Interference-Free Multicast in Multi-Channel Multi-Radio Wireless Mesh Networks," IEEE ICMT, July, 2011.
- [8] W.L. Yang and P.H. Wang, "在電池式無線網狀網路上以最大化總收益為目標的群播繞徑研究," 全國電信研討會.
- [9] G. Zeng, B. Wang, Y. Ding, L. Xiao and M. Mutka, "Multicast Algorithms for Multi-Channel Wireless Mesh Networks," IEEE ICNP, pp. 1-10, 2007.

