

Cross layer design of multicasting and channel allocation for MCMR WMNs

Wen-Lin Yang and Wan-Ting Hong*

Department of Computer Science and Information Engineering, National University of Tainan, Taiwan

Abstract — Wireless Mesh Networks (WMNs) are a new broadband Internet access technology with characteristics different from the traditional wired networks, including self-organization, self-management, self-healing and wide coverage. In this paper, we consider a video/audio delivering system deployed on the multi-channel multi-radio (MCMR) wireless mesh networks, where a number of mesh clients serviced by different mesh routers may subscribe the same video/audio program supplied by a multimedia server installed on the gateway node. For this scenario, multicast would be a better mechanism than unicast to deliver video streams to clients in real time, since only one copy of packets is required to send in multicast protocols. However, with the limited number of channels, some links in the multicast tree may not be allocated with interference-free channels. As a result, some client in the multicast tree may not receive video streams. Therefore, channel allocation has become one of the key problems for constructing multicast tree in WMNs.

In this paper, we propose an algorithm to construct an interference-free multicast tree for WMNs. Our goal is to provide high quality video streams and maximize the number of clients who can successfully receive streams they subscribed. Based on cross-layer design paradigm and considers network routing and channel allocation at the same time during the formation of trees. Also, a channel adjusting procedure is provided to improve the quality of the resulting tree. Finally, several sets of experiments are given to evaluate the performances of our algorithm.

Index Terms — Wireless mesh network, multicast, cross layer, channel assignment.



*Corresponding author: roof770904@hotmail.com

結合跨層設計之頻道分配演算法

楊文霖 洪婉婷*

國立臺南大學資訊工程系

摘要

無線網狀網路(Wireless Mesh Networks, WMNs)是近年來新發展出的無線網路架構，WMNs 具有與傳統有線網路截然不同的特性，其中包含隨時可能改變的網路型態、沒有方向或範圍限制的移動、涵蓋範圍極大等等[1]。設想在一個都市區內，許多人想要收看相同的即時視訊，若用單一傳播(Unicast)的方式，則會浪費很多頻寬，最好的做法是在 WMN 上利用群體廣播(Multicast)的方式，提供即時視訊的服務給大量的使用者。在頻道數目有限的情況下，有些路由路徑無法分配無干擾頻道，導致此路徑底下的用戶無法收到影音串流，因此，頻道分配成為在 WMNs 上建立群播樹最重要的問題之一。

本論文提出一個結合跨層設計之調整式頻道分配演算法，目標是最大化成功接受影音串流的用戶數目且能穩定傳輸串流，本演算法同時進行建立群播路由和頻道分配，加上可調整頻道的程序，使得整體效能能夠更進一步的提升。本論文的最後利用完整且嚴謹的實驗和之前的演算法進行比較，並使用模擬軟體 Qualnet 模擬和評估此演算法。

關鍵字：無線網狀網路、群播、跨層設計、頻道分配

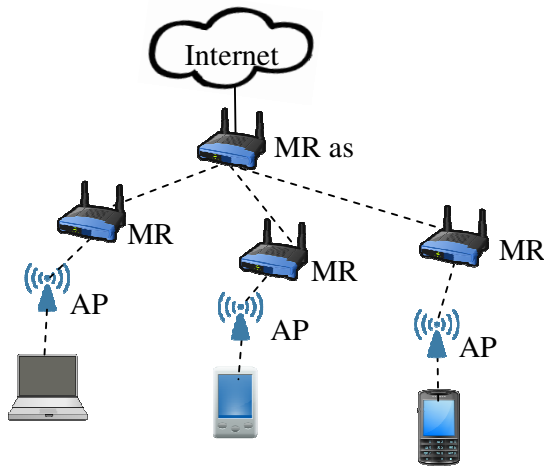
壹、前言

無線網狀網路 (Wireless Mesh Networks, WMNs)是近年來發展出一項新技術，為寬頻網際網路存取服務提供一種新的選擇。WMNs 中的節點兩兩互相連接，來達成一個網狀的網路結構，如果網路中有一節點損壞或因其它因素無法繼續運作時，其它的節點依然可透過多重跳躍通訊(Multi-hop)的方式來傳輸訊息，不會因為有節點損壞而產生通訊中斷或資料無法傳輸的情形發生。除此之外

WMNs 還有架設成本較低、擴充性高、可適用於各種無線網路環境、維護方便與可管理性等等的優點。

WMNs 上的節點可以連到數個不同的節點，而節點又分成兩種，一種是 Mesh routers (MR)，另一種是 Mesh clients(MC)，如圖(一)所示，有些 MR 會扮演 gateway node 的角色，提供其他 MR 以多重跳躍通訊(Multi-hop)的方式連結到網路。在每個 MR 上配置無線基地台 AP(Access point)，藉由 AP 提供網路服務給 Mesh clients，Mesh Clients 可以是筆記型電腦、桌上型電腦、PDA、手機等等。

設想在一個人口密度極高的都市中，例如在臺灣的新北市，在總統大選投票的當天，有大量的人們會透過網路觀看即時開票的過程，若是採用單一傳播(Unicast)的方式，則會造成網路容易擁塞且非常浪費頻寬，若是在 WMNs 上利用群體廣播(Multicast)的方式，將即時視訊傳送給大量的使用者，則會節省許多頻寬。而為了達成此情境，最好的方式是建立群播樹(Multicast tree)來傳送影音串流。在 WMNs 中影響網路效能最大的因素就是頻道之間的干擾問題，若是在每個 MR 上配備多張無線網路介面卡，使其具有多頻道傳輸的特性，就能藉由分配正交頻道(orthogonal channel)的方式來降低干擾並提高網路品質，所以在建立群播樹與分配頻道的議題上，如何改善網路中封包傳送的效能就會顯得特別重要[3], [4]。



圖(一): WMNs example

在建立群播路由(Multicast routing)和頻道分配(Channel assignment)的議題上,在多數論文中,都是使用以下兩種方式:

(1) 先建立群播路由,再做頻道分配[5], [6], [7]

在有線網路的環境中,建立群播路由需考慮的因素有:來源節點至目標節點需經過的節點數目和最長路徑的延遲時間等等,因此常用 shortest path 演算法來建立群播路由,例如論文[8]中所建立的群播路由,就是找出從來源節點到目標節點的最短路徑,使用這些最短路徑來建置群播樹。在無線網路的環境中,建立群播路由需考慮減少中繼節點(rely node)或是資料傳輸 (data transmissions)的數目,例如 Multi-Channel Multicast (MCM) tree [5], [9]與 the Minimum Number of Transmission (MNT) [10]便是以此目的而設計的建立群播樹演算法。以此方式可以先決定較佳的路由路徑再進行頻道分配,但是當建立好的群播路由中的節點無法分配頻道時,則會讓此節點底下的用戶無法接收影音串流。

(2) 利用跨層設計(Cross layer)的方式,同時進行建立群播路由和頻道分配[11] [12] [13] [14]

傳統的網路架構以分層進行為主,而近年來的研究發現此作法無法達到最佳傳輸與資源有效地應用,因此,跨層設計逐漸成為研究的熱門主題之一。方法(1)的方式比較缺乏彈性,當群播路徑若找不到無干擾頻道時,則無法進行調整,若是採用跨層設計的方式,當在網路層決定好路由路徑後,可以得

知 MAC 層的頻道分配以及干擾情形,透過此兩層的互相合作使得整體效能有所提升。

跨層設計被認為會繼續成為 WMNS 中的一個重要課題,然而,還是有許多議題需要被考慮,除了層與層之間的關係被模糊化、現存通訊協定間的相容性、對未來網路設計的影響、維持與管理網路的困難等等,所以當實作跨層設計時,有些規則是需要被遵守的[15]。

相同的群播樹,若是以不同的頻道配置方式,其傳播效能也會不相同,除了要有較佳的建立群播路由演算法外,頻道分配演算法也是相當重要的一部份。本論文針對無線網狀網路的環境,提出一個 Cross-layer design of multicasting and channel allocation 演算法,此演算法是以能否分配無干擾頻道為依據來建立群播樹,當群播路徑可以找到無干擾頻道後,才將此路徑加入群播樹中,演算法中的頻道分配是採用分配非正交頻道(non-orthogonal channel)的方式,以期望達到一個完全無干擾的傳輸環境,在跨層設計的部份,本論文使用應用層、網路層和 MAC 層共三層之間的合作,來達到整體效能的提升。本論文以分配無干擾頻道為前提的情況下,進而保證用戶端可以接收穩定且高品質的影音串流,目標是最大化同時接收此串流的用戶數目。另外本論文也使用了 Wireless broadcast advantage(WBA)[16],節點可利用廣播的方式發送給傳輸範圍內的子節點。

本論文的章節內容如下:第二章是相關研究工作,裡面包含群播路由建立、頻道分配和跨層設計,第三章是定義本論文主要探討的研究問題,第四章是介紹本論文提出的 Cross-layer design of multicasting and channel allocation 演算法,第五章是實驗的數據和分析,最後則是本論文的結論。

貳、相關研究工作

2.1 Multicast routing

本論文中建立群播路由的部份,我們選擇建立樹狀的群播路由,也就是群播樹(Multicast tree),在

群播樹的建立上需要考慮的因素包含路徑的選擇與建立、節省能量消耗、降低頻寬需求與群播樹的鏈結數總和等等。在論文[17]中比較兩種群播樹，shortest path trees(SPTs) 與 minimum cost trees(MCTs)，經由實驗結果得知，在平均路徑和延遲時間的部份，SPTs都是最短的，而在封包傳送率和吞吐量(throughput)的部份，隨著網路負載值(packets/s)的上升，SPTs的表現也是越來越好的。值得一提的是，若在WMNs中實作MCTs，當有MC移動或是離開WMNs時，必須重新計算樹的cost值，以期望達到一個最佳的MCTs，故實作MCTs還是有一定的困難，所以在許多的應用上例如視訊會議、多方互動遊戲等等，SPTs還是一個比較好的選擇。在本節中將介紹三種群播演算法，分別是GA Multicast routing[18]、Multi-Channel Multicast (MCM) tree [5] 與 Load-based Multi-Channel Multicast (LMCM)[7]。

2.1.1 GA Multicast routing [18]

此演算法以基因演算法(GA)為基礎，目標是找到滿足服務質量(quality of service)條件的群播樹。GA提供較佳的隨機搜尋方式，其主要的作法包含挑選(selection)、交換(crossover)和突變(mutation)，適應函數(fitness function)則是計算此群播樹的頻寬、延遲時間和花費，演算法的步驟如下：

1. 從來源節點開始，隨機連接與其相鄰的節點，重覆此步驟直到所有目標節點都被加到群播樹中。以此演算法產生一個隨機的群播樹。
2. 做基因演算法中的crossover，先將這個群播樹分出好幾個部分後，再重新組合成一個網路，在此網路中重新建立新的群播樹。
3. 做基因演算法中的mutation，挑選群播樹中的某個節點，刪除與之相連的鏈結後，將此節點與別的節點相連。
4. 重覆步驟2和3，每次產生一個新的群播樹，都會計算其適應值，最後保留適應值最高的，就是最佳的群播樹。

2.1.2 Multi-Channel Multicast(MCM) [5]

MCM 演算法的目的在於最小化中繼節點、最小化從來源節點至目標節點需經過的節點數目，其演算法步驟如下：

1. 利用廣度優先搜尋 (Breadth first search, BFS) 從根節點的位置開始，依序給各節點一個 level 值，並根據這個 level 值，把同一層 level 之間的鏈結去除。由於是 BFS，所以每條路徑從來源節點到目標節點經過的節點數目被期望是最小。
2. 從最底層(max level)開始，尋找擁有最少父節點的節點，在這些節點的父節點中，找出擁有最大子節點數的節點，並將其加入群播樹之中。
3. 剩餘的父節點會重新計算子節點數目，接著找出擁有最大子節點數的節點，並依序加入群播樹之中。
4. 當該層的所有節點都加入群播樹後，就可以往上一層並重復步驟 2、3 直到所有節點都加入群播樹。

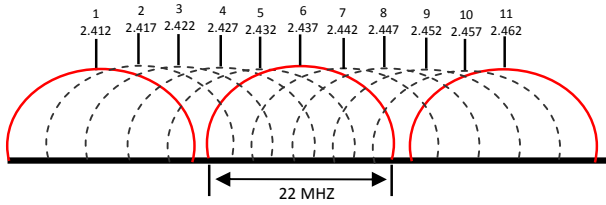
2.1.3 Load-based Multi-Channel Multicast(LMCM)

[7]

LMCM衍生自MCM演算法，LMCM和MCM最大的差別在於挑選節點的方式不同，在LMCM演算法中，額外加入負載 (load)的觀念，其作法是在建立群播樹時，給予每個目標節點(Destination node)從1到5的負載值，負載值越大代表用戶數目越多，而負載值的計算方式為節點本身的負載值加上所有子節點的負載值，依據負載值的大小依序加入群播樹中，完整的演算法請參考論文[7]。建立群播樹時，若能夠最小化中繼節點數，使得鏈結數變少，則利於頻道分配時能夠降低頻道干擾的問題，而LMCM演算法所建立的群播樹，便是具有此點優勢。在本論文第五章的實驗中，便是使用LMCM搭配BFS、DFS和BFB[2]與本論文提出的Cross-layer design of multicasting and channel allocation演算法做比較。

2.2 Channel assignment

頻道分配演算法也是在 WMNs 中被大家廣泛討論的一個議題，依分配的頻道而言，一般主要分為分配正交頻道(orthogonal channel)和分配非正交頻道(non-orthogonal channel)。若兩個節點在傳送串流時產生干擾，藉由分配正交頻道，即可避免產生干擾，但是在 802.11 b/g 的協定中，11 個可用頻道中只有 3 個是屬於正交頻道，由圖(二)可知每個頻道的範圍是 22MHz，因此頻道間會有互相干擾的情形，根據此圖可知正交頻道分別是 1、6、和 11，使用分配正交頻道方式的，主要是分配這 3 個頻道。而在非正交頻道方面，主要是分配 1 至 11 共 11 個頻道，本論文使用的是分配非正交頻道(non-orthogonal channel)的方式，即使頻道數目變多，由於無干擾頻道的數量還是相當有限，造成有些節點依舊無法分配到無干擾頻道，在本節中將會介紹這兩種頻道分配的方式。



圖(二)：802.11 b/g 頻道表

2.2.1 Non-orthogonal channel assignment

由於頻道之間的 SNR 干擾問題[20]，在研究[5]，[9]中，提出以干擾系數(Interference factor)的觀念，根據傳輸範圍(transmission range)與干擾範圍(interference range)的關係，來找出無干擾或是最小干擾的頻道來進行分配，表 1 就是干擾系數與頻道之間的列表，R 是兩節點間的有效傳輸距離，當要分配頻道 c_1 、 c_2 給相鄰兩節點時，定義 channel separation(cs)是 $|c_1 - c_2|$ 。根據表 1 可推導出表 2， δ 是兩條鏈結間的最短距離，舉例來說，若節點 A 至節點 B 的距離是 $0.4R$ ，代表節點 A 與 B 的頻道差必須大於或等於 4，當分配頻道 2 給節點 A 時，則節點 B 只能分配頻道 6、...、11 才能避免產生干擾。

表 1 802.11 b/g 干擾係數與頻道關係圖

Channel separation	0	1	2	3	4	5
11 Mbps	2.0R	1.2R	0.7R	0.5R	0.2R	0

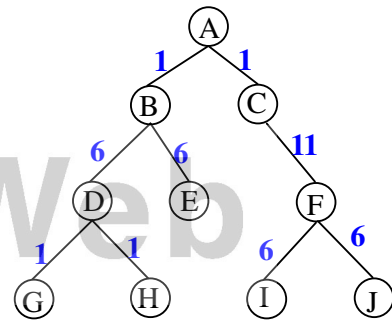
表 2 鏈結間距離與頻道關係圖

Distance between two links	Channel separation
$0 < \delta < 0.2R$	$cs \geq 5$
$0.2R \leq \delta < 0.5R$	$cs \geq 4$
$0.5R \leq \delta < 0.7R$	$cs \geq 3$
$0.7R \leq \delta < 1.2R$	$cs \geq 2$
$1.2R \leq \delta < 2R$	$cs \geq 1$
$2R \leq \delta$	$cs \geq 0$

2.2.2 Orthogonal channel assignment

在論文 [19] 提出一個 Ascending channel allocation 演算法，此演算法採用分配正交頻道(1,6,11)的方式，在群播樹中從 root 節點開始，分配頻道給下一層的節點，等到該層都分配完後，再繼續往下一層。其優點是避免讓鄰近的節點分配到同一個頻道、容易實作等等，其目的是最小化干擾狀況，其演算法步驟如下：

1. 如圖(三)從節點 A 開始，分配與其連接的節點 B 和 C 第一個正交頻道。
2. 節點 B 以下的分配第二個正交頻道，節點 C 以下的分配第三個正交頻道。
3. 節點 D 以下的分配又重新分配第一個正交頻道，節點 F 以下的分配第二個正交頻道。



圖(三)：Channel allocation example

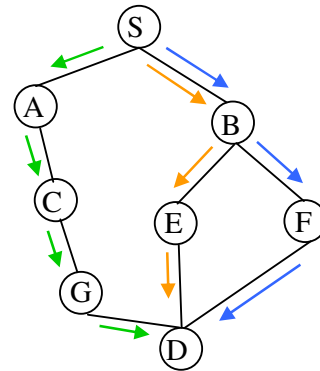
2.3 Cross layer design

在有線網路的 OSI 七層網路架構中，必須是相鄰的協定層才能互相溝通，而無線網路無法保證可靠的連線與百分之百的封包傳送率，使得較高的協定層會受到低層協定的影響，另外像是傳送錯誤封包，導致網路擁塞[21]等等，使得有線網路架構越來越不適用於無線網路。為了能在無線網狀網路上達到更好的效能，越來越多的研究利用跨層設計的方式，試圖去達到最佳的效能。跨層設計定義如下[22]：

- 在不相鄰的協定層之間，可以進行通訊及變數的分享
- 重新定義層與層之間的邊界，稱作建立新的介面

以下將探討論文[23]中提出的token distribution and target programming routing algorithm (TDTPA)，利用跨層設計的概念，在網路層利用Forwarding程序找出路由路徑，然後經由實體層計算其延遲時間與干擾並分配其頻道，當此路徑的延遲時間超過限定的延遲時間，便會回報給網路層，並利用Token Distribution程序，另外找可行的路由路徑。此演算法的目標是找出最佳的路由路徑，並最小化頻道間的干擾，其演算法步驟如下：

1. 如圖(四)所示，從來源節點 S 開始，經由 Forwarding 程序找尋路徑傳送資料給目標節點 D。
2. 選擇節點 B 為下一個節點時，會分配頻道給 B，並計算到節點 B 為止的延遲時間，若超過限定的延遲時間，則會利用 Token Distribution 程序，放棄節點 B，改選擇傳送節點 S 左邊的節點 A。
3. 當找到一個最佳路徑時(例如 S→B→E→D)，利用 The reverse forwarding 程序，通知來源節點 S 後，節點 S 才開始傳送資料。



圖(四)：Algorithm example

在論文[15]中提到三種最常見的跨層設計應用，分別是 MAC/Physical、Routing/MAC 和 Transport/Physical，本論文使用的跨層設計是結合應用層、網路層和 MAC 層，在應用層計算群播樹中每個節點的負載值(load)後，在網路層依據負載值大小決定建立群播路徑的順序，而在建立路徑的同時，從 MAC 層得知此路徑是否可分配無干擾頻道，本論文利用此三層的跨層協同合作，互相傳遞該層的資訊，幫助提昇整體網路效能，期望達到最大效益。

參、問題定義

本論文主要探討的問題是，如何有效的在建立群播樹的過程中，同時分配完全無干擾的頻道，以達到傳送穩定的影音串流與最大化同時接收此串流的用戶數目。在環境設定的部分，在 WMNs 中隨機分布節點，這些節點都配備兩張以上的全向性無線網路卡，而其採用的是 802.11 b/g 標準，雖然此標準只有 3 個正交頻道，但由於此標準被廣泛的應用在很多地方，故本篇還是採取此標準，且在本論文中也會依據 2.2 中的表 1，計算節點間的距離後並分配無干擾的非正交頻道。

本論文中，定義無向圖 $G=(V,E)$ ，其中 V 代表節點的集合， E 是無向邊的集合，另外定義目標節點的點集合 M 使得 $M = \{u | u \in V, demand(u) > 0\}$ ，

其中 $demand(u)$ 是代表此節點中接收串流的用戶

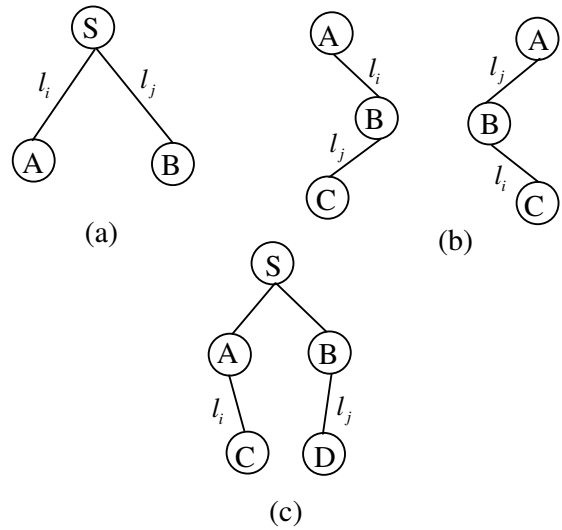
數。定義 $\bar{T} = (\bar{V}, \bar{E})$ 是完成頻道分配後的群播樹，由於頻道數目有限，集合 \bar{V} 中可能沒有包含集合 M 中的所有節點，所以如方程式(1)所示，本論文的目標是讓 gain 值達到最大。設想當一個影音伺服器提供世足盃即時直播的影片，而在此網路中共有 100 個用戶要求接受此影片串流，本論文的目標是最大化同時接收此串流的用戶數[2]。

$$gain = Maximize \sum_{u \in M \ \& \ u \in T} demand(u) \dots \dots (1).$$

肆、演算法

在本章節中將介紹本論文所提出的 Cross-layer design of multicasting and channel allocation 演算法(圖五)，由圖(五)可知，此演算法由三大主要的程序所組成，分別是圖(五)的 Find_aLink_based_on_WBA 程序、圖(六)的 Find_aLink_with_feasible_channel 程序和圖(七)的 Find_aLink_with_Backtracking 程序。此演算法採用建立群播樹的同時並分配無干擾頻道，所以能在確認路徑可以分配無干擾頻道的情况下，選擇較佳的傳輸路徑。

本論文沿用論文[2]提出 Interference matrix(IM)來定義 link 之間的關係，由附錄一可知， $IM(l_i, l_j) = (\alpha, \beta)$ ，其中 α 代表 l_i 和 l_j 的干擾型態，其干擾型態共有三種，第一種如圖五(a)所示， l_i 和 l_j 擁有相同的父節點(sibling links)，第二種如圖五(b)所示， l_i 的子節點是 l_j 的父節點抑或相反(consecutive links)，第三種如圖五(c)所示， l_i 和 l_j 存在著距離的關係。此三種干擾型態分別有其相對應的 β 值， β 值代表 l_i 和 l_j 之間避免干擾的最小頻道差， β 值可以從 0 至 5。故在分配頻道時，只要先找出欲分配頻道的 link 和已經在群播樹中的 link 的關係，便可知道其最小頻道差值，以利分配無干擾頻道。



圖(五): Relationship between links

本論文所提出的演算法過程如下：

- (1) 由圖(六)可知，先將無向圖 G 轉成有向圖，利用 BFS 的搜尋方式分配 level 值給每一個節點，接著如圖(六)第 3 點所示，去除一些 links，使得網路結構能更加簡單，讓建立群播樹的過程不會產生封閉迴路。
- (2) 本演算法加入負載值的觀念，如圖(六)第 4 點所示，利用 Load_computation 程序(圖七)，計算每個節點的負載值(load)後，才能決定建立群播路徑的順序。
- (3) 在分配頻道的部份，首先會利用圖(九)的 Find_aLink_based_on_WBA 程序，找出可共享相同頻道的 link。
- (4) 接著利用圖(十一)的 Find_aLink_with_feasible_channel 程序，找出可分配無干擾頻道的 link。
- (5) 最後利用圖(十三)的 Find_aLink_with_Backtracking 程序，替剩下的 link 找出無干擾頻道，以下將詳述演算法的內容。

Input: given a undirected graph $G = (V, E)$, a multicast set $M = \{u \mid u \in V, demand(u) > 0\}$ and a gateway node s ;

1. Make G being a directed graph by replacing an undirected edge with two oppositely directed links.
2. Starting from the gateway node, a BFS traversal procedure is used assign each node a level value;
3. Link $(u \rightarrow v)$ is deleted if $level(u) > level(v)$. But, two nodes in the same level may have two oppositely directed links connected to each other. And, some nodes have more than one link from upper-level nodes.
4. Compute load value for each node, call $Load_computation(G(V, E))$;
5. Let multicast tree $T = (S, \bar{E})$, $S = \{s\}$, $\bar{E} = \emptyset$ and $U = V - \{s\}$;
6. While(true) {
7. If $((u, v) = Find_aLink_based_on_WBA(T, U)) \neq null$ { $\bar{E} = \bar{E} \cup (u, v)$; $S = S \cup \{v\}$; $U = U - \{v\}$; }
8. Else if $((u, v) = Find_aLink_with_feasible_channel(T, U)) \neq null$ { $\bar{E} = \bar{E} \cup (u, v)$; $S = S \cup \{v\}$; $U = U - \{v\}$; }
9. Else if $((u, v) = Find_aLink_with_Backtracking(T, U)) \neq null$ { $\bar{E} = \bar{E} \cup (u, v)$; $S = S \cup \{v\}$; $U = U - \{v\}$; }
10. Else break;
11. If $(M \subseteq S)$ break; }

Output: T

圖(六) : Cross-layer design of multicast routing and channel assignment with Backtracking

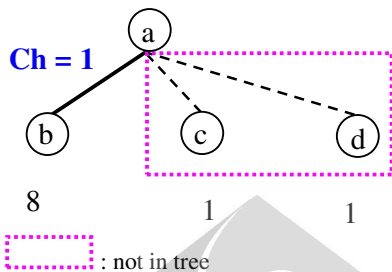
Load_computation (G) {

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 v is adjacent to node u) {
6. If ($level(v) < level(u)$) { $load(v) = load(v) + load(u)$; }
7. Else if ($level(v) == level(u)$) { $load(v) = load(v) + demand(u)$; }
8. }}

}}

圖(七) : Load_computation (G) procedure

4.1 Find_aLink_based_on_WBA procedure



圖(八) : Multicast tree T

- (1) 在群播樹 T 中，假設 $link(a,b)$ 分配到頻道 1，由圖(八)可知， $link(a,c)$ 和 $link(a,d)$ 可以利用 WBA 的特性，和 $link(a,b)$ 使用相同頻道而不會互相干擾，且尚未被加入 T 中，故在圖(九)第 2 點可知， $\mathcal{E} = \{(a,c), (a,d)\}$ 。
- (2) 如圖(九)第 4 點所示，接下來要比較在 \mathcal{E} 中的 link，也就是比較節點 c 和 d 的負載值後，才能決定加入群播樹的順序，故先加入 $link(a,d)$ 再加入 $link(a,c)$ 。
- (3) 由圖(九)第 5 點可知，當負載值相同時，依據 level 值決定加入群播樹的順序。

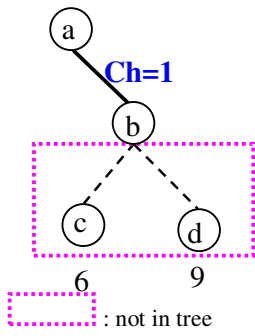
以圖(八)為例，說明圖(九) Find_aLink_based_on_WBA procedure。

```

Find_aLink_based_on_WBA (T,U) {
1. Let link set  $\mathcal{E} = \Omega = \phi$ ;
2. For each link  $(u, w) \in T$  {
    If ( $\exists v \in U$  and link  $(u, v)$  can share the same channel allocated to link  $(u, w)$  without causing any interference )
        {  $\mathcal{E} = \mathcal{E} \cup (u, v)$ ; }
3. If ( $\mathcal{E} == \phi$ ) return null;
4.  $\Omega = \{(u, v) \mid \text{link}(u, v) \text{ with the largest load}(v) \text{ value in } \mathcal{E}\}$ ;
5.  $(\bar{u}, \bar{v}) = \max_{(u,v) \in \Omega} \text{level}(v)$ ;
6. Return link  $(u, v)$  ; }
    
```

圖(九) : Find_aLink_based_on_WBA procedure

4.2 Find_aLink_with_feasible_channel procedure



圖(十) : Multicast Tree T

以圖(十)為例，說明圖(十一)

Find_aLink_with_feasible_channel procedure。

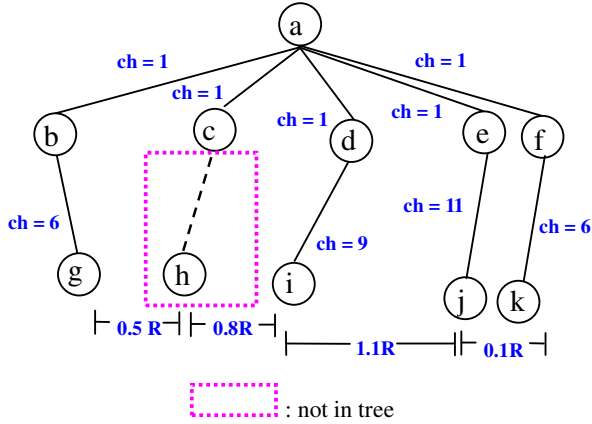
- (1) 在群播樹 T 中，尋找和節點 b 相鄰的節點，由圖(十)可知節點 c 和 d 與節點 b 相鄰，且節點 c 和 d 尚未被加入 T 中。由 Interference matrix[2] 可知，可以分配頻道 6 給 link(b,c)和 link(b,d)，所以如圖(十一)第 2 點所示， $\mathcal{E} = \{(b,c), (b,d)\}$ 。
- (2) 如圖(十一)第 4 點所示，接下來要比較在 \mathcal{E} 中的 link，也就是比較節點 c 和 d 的負載值後，才能決定加入群播樹的順序，故先加入 link(b,d) 再加入 link(b,c)。
- (3) 由圖(十一)的第 5 點可知，當負載值相同時，則依據 level 值決定加入群播樹的順序。

```

Find_aLink_with_feasible_channel (T,U) {
1. Let link set  $\mathcal{E} = \Omega = \phi$ ;
2. For each node  $u \in T$  {
    If(  $\exists v \in U$  and  $v$  is adjacent to  $u$  ) {
        If(an interference-free non-orthogonal channel  $\eta$  that can be allocated to link  $(u, v)$  )
            {  $\mathcal{E} = \mathcal{E} \cup (u, v)$ ; }
    } }
3. If ( $\mathcal{E} == \phi$ ) return null;
4.  $\Omega = \{(u, v) \mid \text{link}(u, v) \text{ with the largest load}(v) \text{ value in } \mathcal{E}\}$ ;
5.  $(\bar{u}, \bar{v}) = \max_{(u,v) \in \Omega} \text{level}(v)$ ;
6. Return link  $(u, v)$  ;
}
    
```

圖(十一) : Find_aLink_with_feasible_channel procedure

4.3 Find_aLink_with_Backtracking procedure



圖(十二) : Multicast Tree T

由圖(十二)可知節點 h 和節點 c 相鄰且尚未被加入 T 中，故先採用 Find_aLink_based_on_WBA 程序，發現無共用頻道可使用，接著採用 Find_aLink_with_feasible_channel 程序，由 interference matrix[2]可知，link(c,h)和 link(a,c)是 consecutive links，所以其頻道差是 5，故 link(c,h)只能分配頻道 6 至 11，由於 link(c,h)和 link(b,g)的距離是 0.5R，故從表 2 可知其頻道差是 3，同理，link(c,h)和 link(d,i)的頻道差是 2，link(c,h)和 link(e,j)

的頻道差是 1，故在此情況下，link(c,h)無法找到無干擾頻道，此時就會採用 Find_aLink_with_Backtracking procedure，以下將介紹此程序。

- (1) 以圖(十二)為例，由圖(十三)第 1 點可知，找出尚未被加入 T 的 links，故 $\mathcal{E} = \text{link}(c,h)$ ，並將在 \mathcal{E} 中的 links 依照 load 值大小進行排序。
- (2) 由第 2 點可知，找出和 link(c,h)會互相干擾的 link set Θ 。
- (3) 由圖(十二)可知，link(b,g)在 link set Θ 裡面，找出可分配給 link(b,g)的 channel set Λ ， Λ 包含頻道 7、8、10、11。
- (4) 由第 4 點可知，改變 link(b,g)所分配的頻道，當 link(b,g)的頻道是 7 時，link(c,h)可以分配頻道 10、11，但是頻道 10 會和 link(d,i)干擾，頻道 11 會和 link(e,j)干擾，當 link(b,g)的頻道是 8 時，link(c,h)可以分配頻道 11，但是頻道 11 會和 link(e,j)干擾，當 link(b,g)的頻道是 10 時，link(c,h)可以找到無干擾頻道 7，此頻道不會與 link(e,j)和 link(d,i)產生干擾，於是將 link(c,h)加入 T 中。

Find_aLink_with_Backtracking (T,U) {

1. Let link set $\mathcal{E} = \{(u,v) | u \in T, v \in U\}$;
Sort \mathcal{E} into a decreasing order list. The order is determined by the load value on the end point of each link;
2. For each link $(u, v) \in \mathcal{E}$ {
 $\Theta = \{(x, y, ch) | (x, y) \in T, (x, y) \text{ is in interference range of link } (u, v) \text{ and } ch \text{ is the current channel allocated to } (x, y)\}$;
3. For each link $(x, y, ch) \in \Theta$ {
 $\Lambda = \{\text{all the interference-free non-orthogonal channels can be allocated to link } (x, y)\} - \{ch\}$;
4. For each channel $\eta \in \Lambda$ {
 Allocate η to link (x, y) ;
5. if(an interference-free non-orthogonal channel λ is found for link (u, v)) { Return link (u, v) and λ ; }
6. } } }
7. Return null; }

圖(十三) : Find_aLink_with_Backtracking procedure

伍、實驗結果

此章節中設計了許多實驗來評估演算法的效能，本論文使用 LMCM 演算法所建立的群播樹搭配 BFS、DFS、Best first、BFB 四種頻道分配演算法與本論文提出的 Cross-layer design of multicasting and channel allocation 演算法進行分析比較，本論文採用的是隨機產生具有 30、50 和 100 個節點的網路，橫坐標代表目標節點個數是總節點數的 10% ~ 90%，實驗的參數設定如表 3。

表3 實驗參數設定

模擬參數	數值
網路大小	100*100 grid
總節點數	30, 50, 100 node
目標節點數比例 (目標節點數/總節點數)	10%, 30% ~ 90%
目標節點上的客戶數目	1 ~ 5
傳輸距離	10 units
干擾距離	20 units
無線網路通訊標準	802.11b
天線種類	全向性天線
節點的鄰近連接節點數目	8
節點的網卡數目	≥ 2

本次實驗所使用的電腦配備為 AMD Athlon (tm) 64 X2 Dual Core Processor 5000+ 2.61 GHz 1.5 GB RAM，作業系統為 XP SP3。

本次實驗主要比較以下五個項目：

- 用戶端數目
- 吞吐量
- 封包傳送率
- 最長路徑延遲

用戶端數目實驗的分析是透過 JAVA 程式執行 1000 次的平均結果，吞吐量、封包遺失率和最長路徑延遲的分析實驗則是使用 Qualnet 模擬軟體實驗執行 20 次的平均結果。Qualnet 傳輸設定是採用 MCBR(Multicast constant bit rate)，每次傳送 200

個封包(封包/秒)，每個封包大小是 512KB。如表 4 所示，本篇提出的演算法在效能上有一定的提升，且不需付出大量的執行時間。

表 4. CPU Time (目標節點數比例 50%) - (s/run)

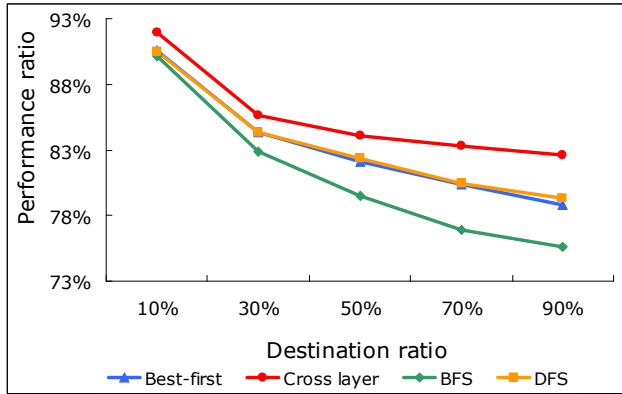
	30 Node	50 Node	100 Node
BFS	0.031	0.101	0.326
DFS	0.032	0.102	0.337
BFB	0.032	0.104	0.349
Cross layer	0.033	0.112	0.36

5.1 用戶端數目

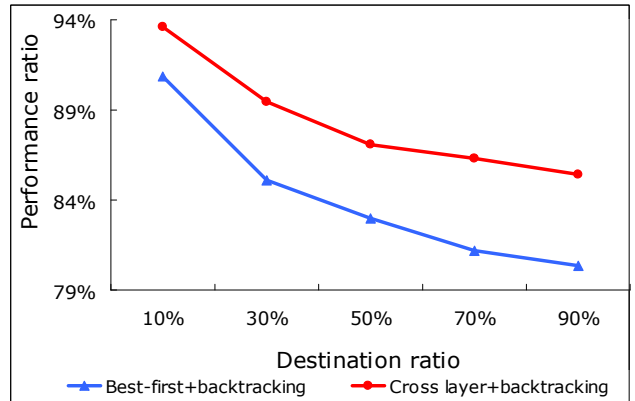
這項效能評比主要是根據該演算法建立的群播樹，其目標節點的數目與原網路的目標節點的比例來做評估，比例越高表示服務到的用戶數目越多，相對遺失的用戶數目也越少。本項目分成兩大類進行比較，第一類的演算法中不包含 Backtracking 程序，第二類的演算法是有包含 Backtracking 程序的。

(一) 圖(十四)至圖(十六)中比較了四個演算法，分別是 BFS、DFS、Best-first[2]、Cross layer，由圖可知當目標節點數的比例增加時，Cross-layer 演算法的效能會逐漸高於其他演算法，主要原因是建立群播樹時，是在確定可分配無干擾頻道的前提下，才選擇此路徑，使得整體結構比較有彈性。

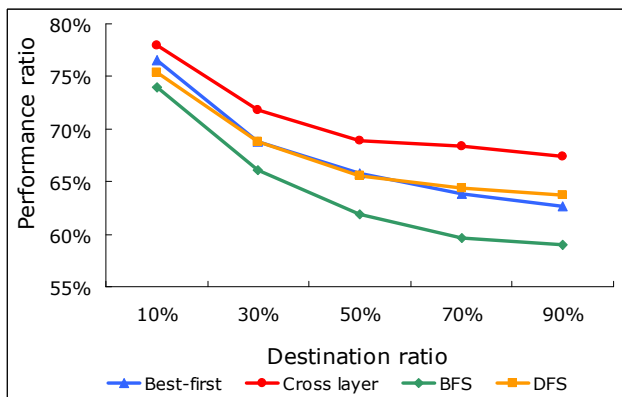
(二) 圖(十七)至圖(十九)中比較有 Backtracking 程序的 cross layer 演算法和 BFB[2]演算法，由於 Backtracking 程序可以調整過去已經分配過的頻道，故使得整體的頻道分配能夠更有彈性，而從實驗結果可知 Backtracking 程序對整體效能而言，有一定的提升作用。



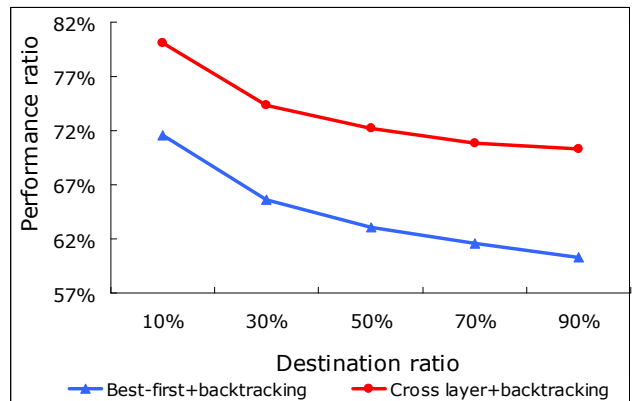
圖(十四)：用戶端數目(30 node)



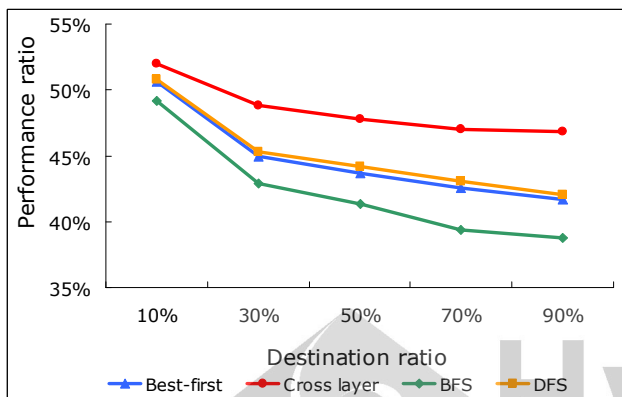
圖(十七)：用戶端數目(30 node)



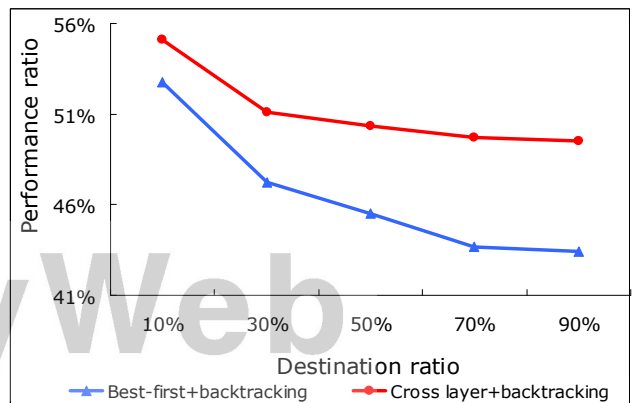
圖(十五)：用戶端數目(50 node)



圖(十八)：用戶端數目(50 node)



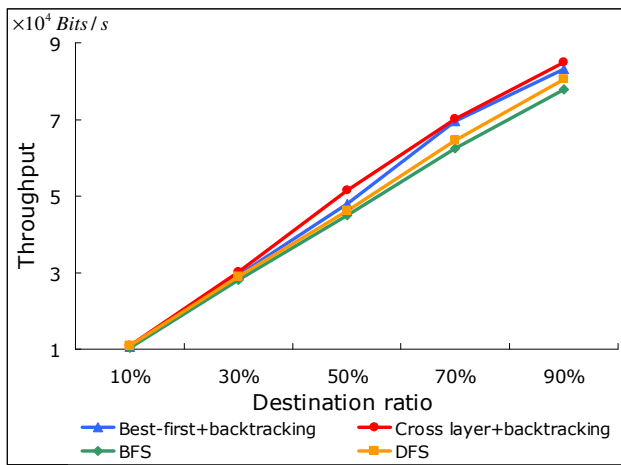
圖(十六)：用戶端數目(100 node)



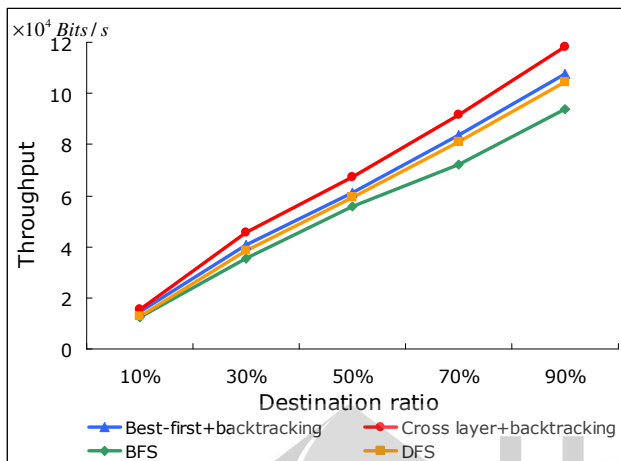
圖(十九)：用戶端數目(100 node)

5.2 吞吐量

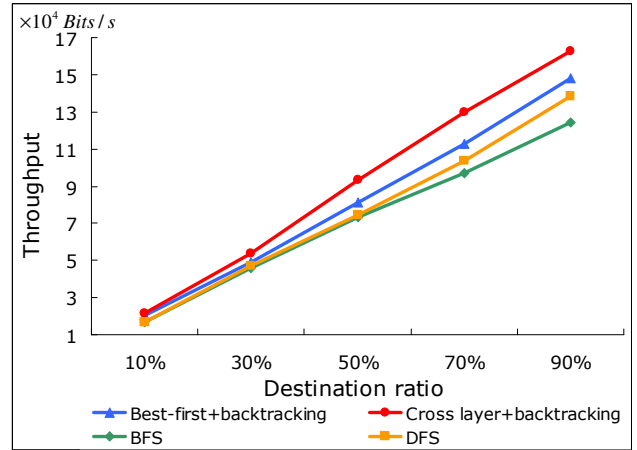
吞吐量主要是比較該演算法建立的群播樹中，目標節點接收到的封包數量，由於傳送的封包數目是固定的，故吞吐量數值越高表示該演算法建立的群播樹連接到的目標節點越多，由圖(二十)至圖(二十二)可知，本篇提出的 cross layer 演算法在目標節點比例增大時，都表現的比其他演算法好。



圖(二十)：平均吞吐量(30 node)



圖(二十一)：平均吞吐量(50 node)



圖(二十二)：平均吞吐量(100 node)

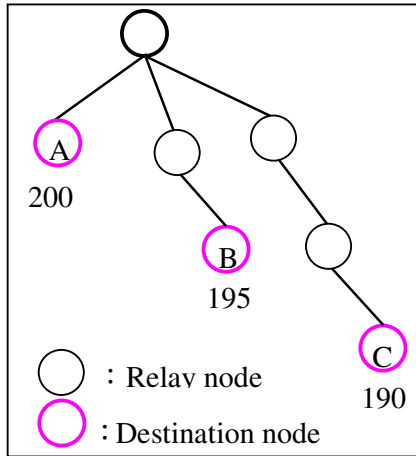
5.3 封包傳送率

封包傳送率計算方式如下：

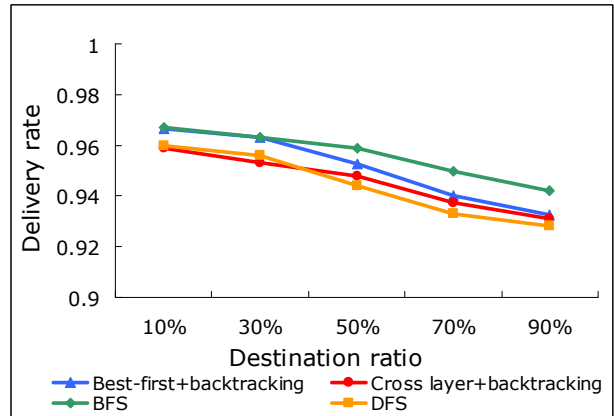
- (1) 收到封包數 = 目標節點收到封包數。
- (2) 平均收到封包數 = 此次模擬結果中目標節點收到的封包數平均。
- (3) 封包傳送率 = 加總所有模擬結果中的平均收到封包除以模擬次數。

在圖(二十三)中，平均收到封包數等於 195，加總模擬 20 次的平均收到封包數後除以 20，即可得到封包傳送率。由圖(二十四)至圖(二十六)可知，BFS 演算法封包傳送率較高，主要原因是群播樹的結構，從 root 節點至目標節點可能會經過較少的節點數，使得封包傳送率較高。雖然 BFS 具有較高的封包傳送率，但其吞吐量卻較低，代表其連接到的目標節點較少，但是傳送效率卻較高。

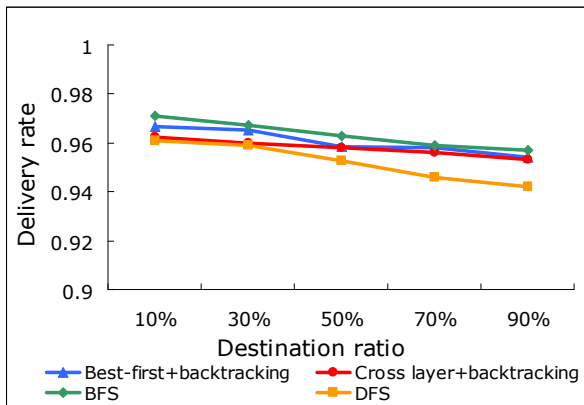




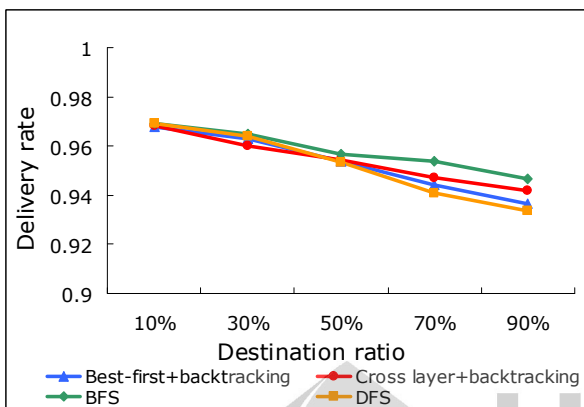
圖(二十三)：Simulation example



圖(二十六)：平均封包傳送率(100 node)



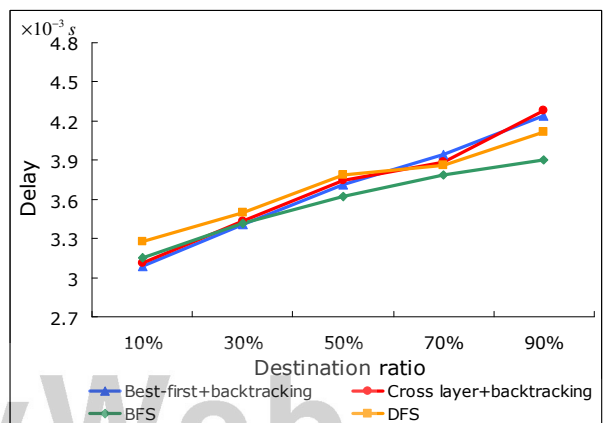
圖(二十四)：平均封包傳送率(30 node)



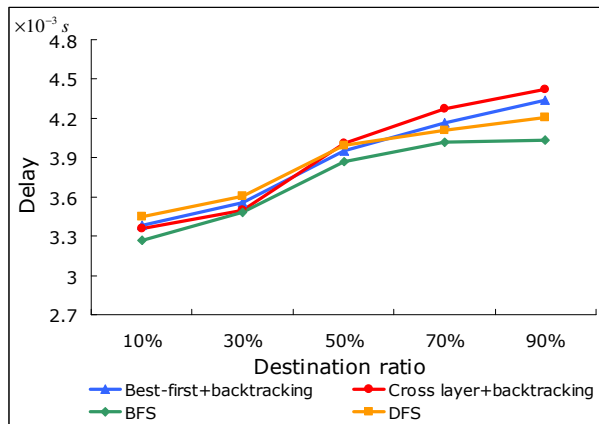
圖(二十五)：平均封包傳送率(50node)

5.5 最長路徑延遲

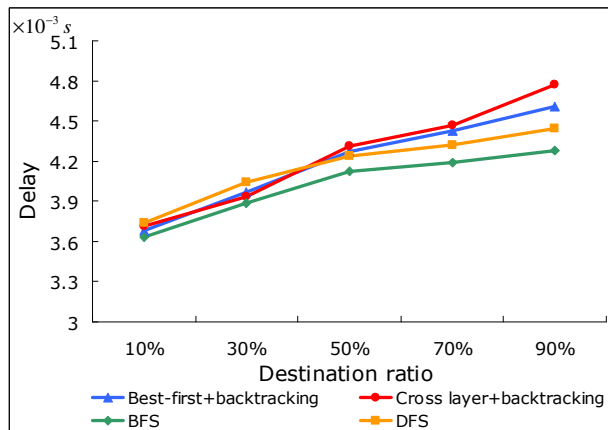
最長路徑延遲時間主要是計算每次產生的群播樹中，最長路徑延遲時間的平均值。由圖(二十七)至圖(二十九)可知，BFS 演算法的延遲時間最短，主要原因是 BFS 演算法建立的群播樹結構，從 root 節點至目標節點所經過的節點數是最少的，而另外三種演算法，其建立的群播樹結構，從 root 節點至目標節點所經過的節點數量都是比較多的。本篇提出的 Cross layer 演算法，雖然延遲時間較長，但是卻可以連接到較多的目標節點並服務更多的用戶。



圖(二十七)：平均最長延遲時間(30 node)



圖(二十八)：平均最長延遲時間(50 node)



圖(二十九)：平均最長延遲時間(100 node)

陸、結論

在本論文中，提出一個新的演算法，利用跨層設計的方式建立路由與分配頻道，本論文使用分配無干擾頻道，建立起一個無干擾的傳輸環境，已達到最大化用戶數目與提供穩定傳輸串流的目標。使用 Qualnet 這套模擬軟體分的實驗分析所得到的數據顯示，在用戶端數目和吞吐量的方面，本演算法有一些進步，而在封包傳送及遺失率方面，則是有穩定表現。在未來研究工作的部份，希望繼續使用跨層設計的理念，讓整體的效能能夠更加的提升。

參考文獻

- [1] 無線寬頻網路建設, 2011/6/18
<http://www.gss.com.tw/index.php/eis/476>
- [2] 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.
- [3] I.F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: a survey," Computer Networks, vol. 47, 2005, pp.445-487.
- [4] Ashish Raniwala and Tzi-cker Chiueh, "Architecture and Algorithms for an IEEE 802.11-based Multi-Channel Wireless Mesh Network," IEEE INFOCOM, pp. 2223- 2234, 2005.
- [5] G. Zeng, B. Wang, Y. Ding, L. Xiao and M. Mutka, "Multicast Algorithms for Multi-Channel Wireless Mesh Networks," IEEE ICNP, 2007, pp. 1-10.
- [6] Chia-Sheng Chou; Chien Chen; Ying-Yu Chen; Rong-Hong Jan and Cheng-Chung Hsieh, " Multicast routing and channel assignment in wireless mesh networks," Mobile Adhoc and Sensor Systems, 2009. MASS '09. IEEE 6th International Conference, pp. 676 - 681, Oct. 2009
- [7] W.L. Yang and W.T. Huang, " The study of interference-free multicast using non-orthogonal channels for Multi-Radio and Multi-channel Wireless Mesh Networks," International Computer Symposium (ICS), pp. 547-552, Dec. 2010.
- [8] U.T Nguyen and J. Xu, "Multicast Routing in Wireless Mesh Networks: Minimum Cost Trees or Shortest Path Trees," IEEE Communications Magazine, 2007, pp.72-75.
- [9] H.L. Nguyen and U.T. Nguyen, "Minimum Interference Channel Assignment for Multicast in Multi-Radio Wireless Mesh Networks," IWCMC'08, 2008, pp. 626-631.
- [10] P.M.Ruiz and A.F. Gmez-Skarmeta, "Approximating Optimal Multicast Trees in Wireless Multihop Networks," ISSC 2005, pp. 686-691.
- [11] M. Jahanshahi, M. Dehghan, M. R. Meybodi, "A Mathematical Formulation for Joint Channel Assignment," Journal of Network and Computer Applications (Nov, 2011), 34 (6), pp. 1869-1882.
- [12] Hui Cheng, Shengxiang Yang, "Joint Multicast Routing and Channel Assignment in Multiradio Multichannel Wireless Mesh Networks Using Tabu Search," Natural Computation, 2009. ICNC, Issue 14 Aug. 2009, pp. 325 - 330.
- [13] Bononi, L.; Di Felice, M.; Molinaro, A.; Pizzi, S, "Joint Channel Assignment and Multi-path Routing for Multi-radio Wireless Mesh Networks," Distributed Computing Systems Workshops, 2009. ICDCS, issue 22, June 2009, pp. 476 - 481.
- [14] Ngoc Thai Pham; Won-Joo Hwang; "Joint Disjoint Path Routing and Channel Assignment in Multi-Radio Multi-Channel Wireless Mesh Networks," Vehicular Technology Conference, issue 21, Sept. 2008, pp. 1-5.
- [15] I. F. Akyildiz and X. Wang, "cross layer design in wireless mesh network," Vehicular Technology, vol 57, issue 2, March 2008, pp. 1061 - 1076.
- [16] N. Lan and N. Trang, "Channel assignment for multicast in multichannel multi-radio wireless mesh networks," Wiley Journal of Wireless Communications & Mobile Computing, vol. 9, pp. 557-571, April 2009.
- [17] Uyen Trang Nguyen, Jin Xu, "Multicast Routing in Wireless Mesh Networks: Minimum Cost Trees or Shortest Path Trees?," Communications Magazine, IEEE, vol.45, issue11, pp.72-77, Nov, 2007.
- [18] Xia Li, Qiu Ning, Zhang Jun-Ya, Liu Yang-Qian, "An improved GA for QoS multicast routing algorithm," Control and Decision Conference, Issue 2, page 393-396 July, 2008.
- [19] Guokai Zeng, Bo Wang, Yong Ding, Li Xiao, Matt W. Mutka, "Efficient Multicast Algorithms for MultiChannel Wireless Mesh Networks," IEEE Transactions on Parallel and Distributed Systems, vol 21, issue 1, pp.86-99, Dec, 2009.
- [20] A. Mishra, E. Rozner, S. Banerjee, and W. Arbaugh, "Exploiting partially overlapping channels in wireless networks: Turning a peril into an advantage," in ACM/USENIX Internet Measurement Conference, 2005.

- [21] V. Kawadia and P. R. Kumar, "A cautionary perspective on cross-layer design," *Wireless Communication.*, vol. 12, no. 1, pp. 3–11, Feb. 2005
- [22] V. Srivastava and M. Motani, "Cross-Layer Design : A Survey and the Road Ahead," *IEEE Communications Magazine*, vol. 43, no. 12, pp. 112-119.
- [23] Xiang-lin Zhu, "Cross-Layer Routing Algorithm Design for QoS Provisionings over 802.11s Wireless Mesh Networks." *Vehicular Technology Conference*, Issue 20, pp1-5, Sept, 2009.

BIOGRAPHIES



Wen-Lin Yang received the PhD Degree in Computer Science from the Pennsylvania State University, University Park, USA, 1993. He is currently a Professor at the Department of Computer Science and Information Engineering in the National University of Tainan, Tainan, Taiwan. His primary research interests include routing protocols, quality of service, ad hoc networks and distributed computing.



Wan-Ting Hong was born in Taiwan, R.O.C., in 1988. She received her MS degree from the Department of Computer Science and Information Engineering, National University of Tainan, Taiwan, in 2012. Her research interests include wireless networks and multicast routing

附錄一 Interference matrix [2]

For two links l_i and l_j , $\delta_{i,j}$ = distance between l_i and l_j
 $IM(l_i, l_j) = (\alpha, \beta)$ is defined as follows:

- (1) $\delta_{i,j} = 0$, l_i and l_j are sibling links, $IM(l_i, l_j) = (1, 0)$;
- (2) $\delta_{i,j} = 0$, l_i and l_j are consecutive links, $IM(l_i, l_j) = (2, 5)$;
- (3) $0 < \delta_{i,j} < 0.2R$, $IM(l_i, l_j) = (3, 5)$;
- (4) $0.2R \leq \delta_{i,j} < 0.5R$, $IM(l_i, l_j) = (3, 4)$;
- (5) $0.5R \leq \delta_{i,j} < 0.7R$, $IM(l_i, l_j) = (3, 3)$;
- (6) $0.7R \leq \delta_{i,j} < 1.2R$, $IM(l_i, l_j) = (3, 2)$;
- (7) $1.2R \leq \delta_{i,j} < 2.0R$, $IM(l_i, l_j) = (3, 1)$;
- (8) $2.0R \leq \delta_{i,j}$, $IM(l_i, l_j) = (3, 0)$;

