WDM 網路下具有少量光分割及波長轉換功能節點配置問題之研究

丁德榮　　李嘉裕
Der-Rong Din　　Chia-Yu Li
彰化師範大學資訊工程系
E-mail: deron@cc.ncue.edu.tw　bibbyhouse@yahoo.com.tw

摘要

分波多工(Wavelength Division Multiplexing，WDM)網路使用波長路由的方式傳送資料之技術已確定成為下一代網路的主要架構。為能利用少量的波長(wavelength)及波長通道(wavelength channel)來達到群播(multicast)傳送的目的，因此，在光學交換器上提供具有光分割(light splitting)及波長轉換(wavelength converter)功能，此交換器亦稱為VS(virtual source)節點。VS 節點成本較高，故一般僅有部份節點具有 VS 功能，透過配置具有 VS 功能節點及決定群播的傳送路徑，建構出群播樹(multicast tree)，以使得群播所使用的波長及波長通道數最少是一個值得研究的問題。本論文主要研究在 WDM 網路環境下，如何配置 VS 節點，給定一些群播的需求，在需要配置 k 個 VS 節點時，使得群播之效能為最佳的問題。本論文中提出基因演算法來解決此問題，並透過實驗數據來了解成果。

關鍵字：分波多工群播、配置問題、虛擬原始端

Abstract

Wavelength routing together with wavelength division multiplexing (WDM) technology have been considered as a strong candidate for future high performance networks. In order to make use of a small amount of wavelengths and wavelength channels to achieve the purpose of multicast effectively. It is offered the node which has splitting and wavelength converter switch, this node is also called VS (*virtual source*) node. The topology network via placement the node with VS switch, and determine suitable multicast routing to construct the multicast tree. That makes the multicast tree use the number of wavelengths and wavelength channels can be minimized. In this paper a genetic algorithm (GA) is proposed to solve this problem. Experiments are also given to show the efficiency of GA.
Keywords: WDM, multicast, placement problem, virtual source

1. 前言

隨著網際網路的快速發展，使用網路的人數也迅速地增加，因此提供高頻寬及高品質的網路環境是極重要的問題。近年來分波多工(Wavelength Division Multiplexing，WDM)的傳輸技術，被發展來運用光纖超大頻寬的特性，將一條光纖分割成許多並行傳輸的通道(channel)，每一個通道佔用一個波長(wavelength)，每一個通道的傳輸速度達10Gbps，同時，每一條光纖可允許至少100個以上波長同時傳送，故整體之效能可達1Tbps以上，因此以WDM技術為基礎的光纖網路，勢必成為未來網路技術與應用發展的主流。

為使在 WDM 上的群播服務能有效的降低所使用的波長，保留較多之波長可供傳輸，一個簡單的方法是直接將現存於一般網路上(非 WDM)的群播通訊協定[3][7]應用於 WDM 網路中。但這些方法無法避免光電轉換所產生的延遲，造成無法有效利用光纖大量頻寬的特性。為了希望所有封包傳送都不須轉換成電子訊號，且全都在光學(all-optical)環境中完成，因此許多研究者設計了具有光分割功能(light splitting)節點，稱為 MC(multicast capable)節點[2]，而不具有光分割能力的節點稱為MI(multicast incapable)節點[5]。除此之外更提出一個同時具有光分割及波長轉換(wavelength conversion)的節點，稱為 VS(virtual source)節點[6]，以增進群播之效能。但因 VS 設計複雜、組成元件成本過高，並非所有的交換器(Wavelength Routing Switch，WRS)均具有 VS 之功能，因此只有少數 WRS 升級成為具有 VS 功能的節點。而在 WDM 環境下，如何能有效的配置 VS 節點以提升在全光網路中群播的效能，即是本文研究之方向。

2. 研究問題定義

給定一個 WDM 的網路，假設此網路以圖(graph)G=(V, E)，其中 V 表示網路架構中節點(node)的集合，|V|表示節點個數；E 表示節點與節點間鏈結(link)的集合，|E|表示鏈結數；$W=\{\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{|W|}\}$表示可用波長(wavelength)，|W|代表波長個數。網路中的節點分為二種，一種為具有 VS 功能的節點，其餘則皆為具有 TaC 功能的節點，利用 TaC (Tap-and-Continue)的元件加以改進光波長路由交換器(WRS)，此節點不僅可接收進入的光波訊號且可繼續將光波訊號複製轉送到下一個節點。假設給定 m 個群播群組(multicast sessions)的集合 $M=\{M_i, i=1, 2, ..., m\}$，$M_i=\{S_i, D_i\}$，其中 S_i 為群播群組 M_i

的來源節點(source)，D_i 表群播群組 M_i 的目的節點(destination)的集合，$D_i \subseteq V$ 且 $|D_i| \le |V|$，i=1, 2, …, m。在靜態的傳輸模式下，每個群播群組給定一個來源節點，相同數的目的節點，所配置的 VS 節點數為 k 個，且每一個 VS 節點中波長轉換和光分割的範圍都沒有限制，透過基因演算法的方式[4]，找出最適當配置 VS 節點的個數，以及配置 VS 節點的位置，並以核心樹的路由(routing)方式[1]，架構出群播樹(multicast tree)，以期許使用較少的波長通道數，以達到群播的目的。

圖 1 為一個具有 25 個節點，41 條鏈結所構成的拓樸網路，假設每條鏈結上有 10 個波長，VS 節點的個數 k=5，有三個群播群組的需求分別如表 1 所示：

表 1 Multicast session

MULTICAST SESSION	source	destination set
M_1	S_1	D_1={3, 5, 6, 8, 10, 13, 17, 23}
M_2	S_2	D_2={1, 2, 4, 7, 11, 14, 20, 24}
M_3	S_3	D_3={3, 9, 11, 12, 15, 16, 21, 25}

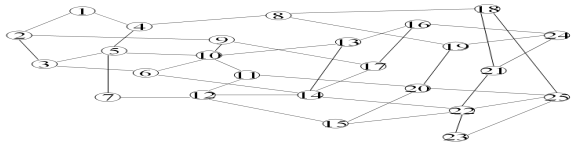

圖 1 拓樸網路架構

如今在如何配置最佳 VS 節點位置的方法中，已有很多探索式(heuristic)的方法紛紛被提出來[6][7]，在文獻[6]中，依網路拓樸中每一個節點分支度(degree)大小來判定是否配置 VS 節點，但此方法中並未考慮到當時群播群組的需求，若分支度大的節點都落於臨界邊上，或落於距離目的節點較遠時，此時若以分支度為考量而配置 VS 節點，可能會因找尋相對應的群播樹而造成連接時不必要鏈結的浪費。因此，在此提出以基因演算法的方式，考慮了群播群組的需求及目的節點距離的問題，以期盼求出最佳配置 VS 節點的位置，並在架構群播樹時使用最少的波長通道數。

3. 基因演算法(genetic algorithm)

基因演算法主要可分成:(1)編碼(encoding)、(2)基因初值設定(initialization)、(3)選擇(selection)、(4)交配(crossover)、(5)基因調整(gene adjustment)、(6)突變(mutation)及(7)適應值函數(fitness function)，透過此七種運算過程操作，以期許找出最佳的 VS 節點配置，其每種操作方式詳述如下：

3.1 編碼(encoding)

將網路中所有的節點以一陣列 $P[]$ 表示，並利用二進位位元串列(binary bit string)來表示各個基因，其中若 $P[i]=1$，則表示節點 i 配置有 VS 功能，否則 $P[i]=0$，表示此節點 i 只具有 TaC 功能，其中 $1 \le i \le |V|$。

3.2 基因初值設定(Initialization)

隨機指派 k 個位置(設 $k = 5$)，配置具有 VS 功能的節點，亦即使陣列 $P[]$ 中共有 k 個 1，$|V|-k$ 個 0。

3.3 選擇(Selection)

以俄羅斯輪盤法(roulette wheel)，將各個基因陣列(gene array)的適應函數值(fitness value)(定義於基因演算法 2.7 節中適應值函數)除以所有基因陣列的適應函數值之總和，每個基因陣列所求出的值則稱之為適應值比率(fitness ratio)。每一代(generation)的演化過程中，依照每個基因陣列適應函數值大小來分割輪盤上的位置，適應函數值越大則在輪盤上所佔有的面積也越大，因此選中去執行交配的機率，亦會隨之增大。

3.4 交配(Crossover)

隨機找出 l 個交配點，$1 \le l < |V|$，將親代基因陣列分割成 $l+1$ 個片段，再以交互選取的方式交換彼此片段的基因陣列，以得出二個新生的子代陣列為 $C_1[]$ 、$C_2[]$。

在此以 $l = 3$ 為例，隨機找出三個交配點 $P[i]$、$P[j]$、$P[z]$，以間隔的方式交換四段中的基因片段，使得：

$$C_1[m] = \begin{cases} P_1[m],\ if\ \ 1 \le m \le i\ \ and\ \ j+1 \le m \le z \\ P_2[m],\ if\ \ i+1 \le m \le j\ \ and\ \ z+1 \le m \le |V| \end{cases}$$

$$C_2[n] = \begin{cases} P_2[n],\ if\ 1 \le n \le i\ \ and\ \ j+1 \le n \le z \\ P_1[n],\ if\ \ i+1 \le n \le j\ \ and\ \ z+1 \le n \le |V| \end{cases}$$

圖 2

顯示多點交配的範例，親代基因陣列 P_1、P_2，假設隨機選取出的三個交配點 $i=6$、$j=13$、$z=20$，透過多點交配法以交互選取的方式交換彼此片段的基因陣列，所產生的子代如圖 2(b)中基因陣列 C_1、C_2 所示。

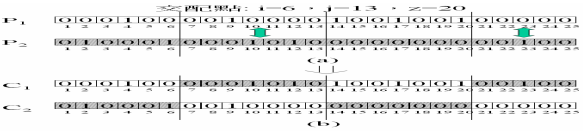

圖 1 多點交配 (a)交配前之基因陣列，(b)交配後之基因陣列

3.5 基因調整(Gene adjustment)

經過交配的運算後，子代基因陣列中位元為

「1」的個數，並不一定恰好等於原先所設定的k值，所產生的陣列並不能成為合理的解(feasible solution)，因此須做適度的調整，以使兩個子代基因陣列中位元為「1」的個數，恰為k個。其中基因調整的演算法敘述如下:

步驟 1、檢視子代C_1、C_2基因陣列中，位元為「1」的個數，是否恰為k個，若以g代表C_1基因陣列中位元為「1」的個數，當$g \neq k$、$count = |k - g| \neq 0$時，則執行步驟 2 或 3 加以調整。

步驟 2、當$g < k$、$count \neq 0$時，隨機選出一點$C_1[i]$，$1 \le i \le |V|$，依陣列順序遞增尋找，直至找到$C_1[i]$後第j個位元，$0 \le j \le |V| - 1$，使得$C_1[x] = 0$且$C_2[x] = 1$；其中x值的計算方法如下:

$$x = \begin{cases} (i + j) \bmod (|V| + 1) & if \quad i + j \le |V| \\ (i + j) \bmod |V| & otherwise \end{cases}$$

相互交換位元，使得$C_1[x] = 1$、$C_2[x] = 0$，並將$count$值遞減，使得$count = count - 1$；反覆步驟 2，直至$count = 0$為止。

步驟 3、當$g > k$、$count \neq 0$時，則找尋$C_1[x]$=1且$C_2[x]$=0的陣列位置，相互交換位元，使得$C_1[x]$=0、$C_2[x]$=1，並將$count$值遞減，使得$count = count - 1$；反覆步驟 3，直至$count = 0$為止。

3.6 突變(mutation)

(1)隨機突變(Random mutation): 以隨機的方式，找出陣列$P[]$中任兩基因$P[i]$、$P[j]$，$1 \le i < j \le |V|$，予以互相交換。如圖 3 所示，在基因陣列P中，隨機選取i = 9、j=18兩基因位置，予以相互交換。

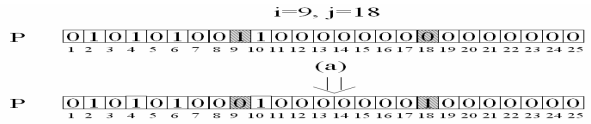

圖 3 隨機突變 (a)突變前之基因陣列，(b)突變後之基因陣列

(2)反轉突變(Inversion mutation): 由陣列$P[]$中隨機選出二點$P[i]$、$P[j]$，$1 \le i < j \le |V|$，將此二點之間的基因依順序反轉放入。如圖 4 所示，在基因陣列中，隨機選取出$i = 5$、$j = 14$兩點，使二點之間的基因反轉並放回基因陣列$P[]$中。

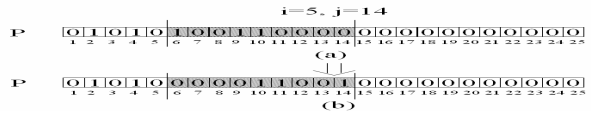

圖 4 反轉突變 (a)突變前之基因陣列，(b)突變後之基因陣列

(3)子字串旋轉突變(Substring rotation mutation): 取一組基因陣列$P[]$，隨機選取陣列$P[]$中第i、j個位置為突變點$1 \le i < j \le |V|$，將$P[i]$位元取出，再將$P[i+1]$至$P[j]$位元依順序左移一個位置，使得$P[x] = P[y]$，$x = i, i+1, i+2, ..., j$，$y = x + 1$，再將$P[i]$位置的位元放置於$P[j]$位置上，其餘位置的內容皆不變。如圖 5 所示，在基因陣列中，隨機選取$i = 6$、$j = 14$兩點，將第$P[7]$至$P[14]$個位元依序往左移一個位置，並將第$P[6]$位置的位元放置於第P[14]位置上。

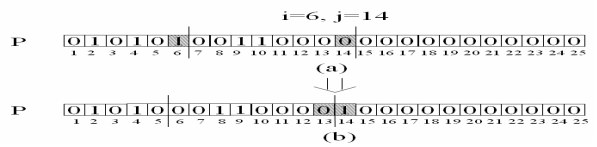

圖 5 子字串旋轉突變 (a)突變前之基因陣列，(b)突變後之基因陣列

3.7 適應值函數(fitness function)

適應值的大小，定義為當 VS 節點的位置決定後，依據所有群播群組(multicast session)的需求，依每一個群組(session)路由的方式，以決定其使用的波長通道(channel)數目，借此定義為此一組基因陣列的適應值(fitness)，以f_i表示第i組之適應值。並以每一代中適應值最大者定為 MAX，以 gen 表示所有產生的基因陣列，以 FP(fitness probability)定義為每個適應值被選取之機率，則$FP_i = (MAX - f_i) / \sum_{j=1}^{gen} (MAX - f_j)$，以將適應值機率較大的基因陣列保留下來，取而代之成為下一代的基因陣列。

4. CBT 方法

為了使每一個具有 VS 功能的節點能更有效率的被使用到，在此透過以建立核心樹(core-based tree, CBT)為基礎共享樹的方法，使所有的群播成員都可以藉由連接此核心樹而來傳送與接收資料，並充分發揮了每一個具有 VS 功能節點的效用。

其中決定核心樹的步驟敘述如下:

步驟 1、依據群播群組(multicast sessions)的所有需求，對每一個群播群組，以最短路徑演算法，找到由來源節點為樹根的最短路徑展開樹(shortest path spanning tree，SPST)，並計算 SPST 經過各節點的次數，並以$|VS_i|$表示第i個 VS 節點被經過的次數。

步驟 2、依據基因演算法節點配置具有 VS 功能的候選集(candidate set，CS={4, 10, 14, 19, 22})中，再根據每個節點所被經過次數的多

寡，選出其中被經過次數最多的 VS 節點當成 CBT 的 root 節點，root $\leftarrow VS_i$，$|VS_i| \geq |VS_j|$，$i \in \{1, 2, ..., k\}$，$j \in \{1, 2, ..., k\}$，並以 root (node 4)作為起始點，執行最短路徑演算法，找出 root 到 CS 中所有節點的 SPST，建構出核心樹 (core-based tree) T_1，如圖 6 所示。

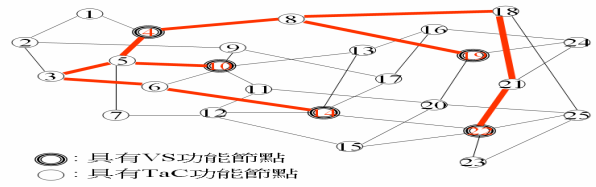

圖 6 核心樹 T_1

4.1 CBT 修正

所建構好的核心樹 T_1 中，可能會經過 TaC 節點，而某些 TaC 節點的分支度(degree)可能大於 2(如圖 4 中之節點 5、8)，因這些節點不具有 VS 功能，無法順利將訊息同時分別傳送兩個以上的節點，因此需修改核心樹 T_1 的架構，使新的核心樹能滿足這些限制。其修改的演算法如下：

步驟 1、查看 T_1 中是否有 TaC 且分支度大於 2 的節點 (| deg *ree* |> 2)，若有，則需執行步驟 2、3 與 4 加以修改。否則，則無需修改此核心樹。

步驟 2、由樹根(root)節點開始，依深度搜尋(DFS)演算法向下尋找，找出第一個 TaC 節點，其分支度大於 2，則由此分支點往葉子節點方向搜尋，查看是否仍有其他 TaC 且分支度大於 2 的節點，依據深度搜尋(DFS)演算法反覆向下尋找，直至無滿足此條件之節點為止，並將所找尋到的節點，依序放入佇列 Q 中。

步驟3、由佇列 Q 中取出一點，設此分支點為A，$T(A)$ 表示以節點 A 為樹根的子樹，以 $BLP_{T(A)}(A, B)$ 表示在樹 $T(A)$ 上，節點 A 的分支路徑中最長的一條路徑，其中 B 表示此路徑上距離 A 最近的一個 VS 節點 ($B \in CS$)。並將 T_1 中節點 A 與節點 B 之間路徑上所有經過的鏈結及節點以 $l_{T_1}(A, B)$ 表示(不包含節點 A、B)。$T(B)$ 表示在 T_1 中移除 $l_{T_1}(A, B)$ 後，而以節點 B 為 root 所形成的子樹 (sub-tree)，並定義子樹 $T'(A) = T_1 \setminus (l_{T_1}(A, B) \cup T(B))$。以 $CS(T'(A))$ 表示 $T'(A)$ 中所有具有 VS 功能節點的集合，並以 $TAC(T'(A))$ 表示在 $T'(A)$ 中所有具有 TaC 功能節點及與其相連接的鏈結集合，定義圖 $G' = G \setminus (TAC(T'(A)) \cup T(B))$。

步驟 4、找出以節點 B 為樹根的子樹 ($T(B)$)，其中所有具有 VS 功能的節點以 $CS(T(B))$ 表示，$CS(T(B)) \subset CS$，由 $CS(T(B))$ 在 G' 中以最短路徑尋找與 $CS(T'(A))$ 候選節點加以連接，取其中距離最近的兩節點 $SP_G(x, y)$，$x \in CS(T(A)); y \in CS(T(B))$，並將核心樹 $T'(A)$ 加以連接 $l_{G'}(x, y)$ 之間所有的鏈結，更新為核心樹 T_1，使得 $T_1 = T'(A) \cup l_{G'}(x, y) \cup T(B)$，並重新執行步驟 3，直至佇列 Q 中已不含有任何節點為止，其建構核心樹的演算法，其虛擬碼(pseudocode)如下圖 7 所示：

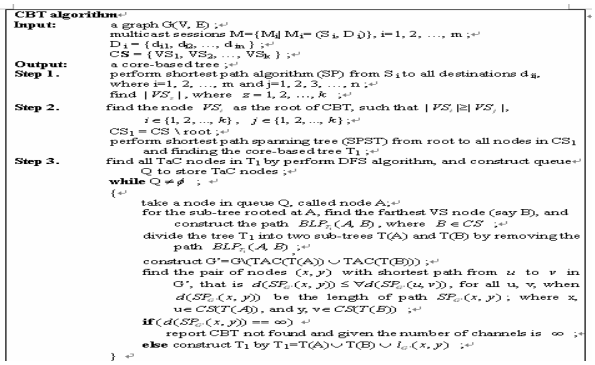

圖 7 CBT algorithm

4.2 決定 multicast tree

每一個群播群組(multicast session)加入前，先檢查此核心樹 T_1 中來源節點是否為 TaC 節點，因為來源節點不具有 VS 的功能，若來源節點為 TaC 節點，卻沒有調整核心樹的架構，則會因來源節點無法同時傳送出資料給予兩端，而造成傳送的延遲及波長通道(channel)數之增加，主要可分成以下三種情形討論：

(1)群播群組的來源節點落於 CBT 上，且來源節點不具有 VS 功能，此時 CBT 必需做一些調整，其調整方式如下：

假設此來源節點為 node A，找出在 CBT T_1 上距離 A 最遠的 VS 節點 B，亦即找出 $BLP_{T_1}(A, B)$，$B \in CS$。移除了節點 A 與節點 B 之間路徑的所有鏈結，使得核心樹分割成為以來源節點 A 為樹根的子樹 $T(A) = T(A) \setminus l_{T_1}(A, B)$，以及以節點 B 為樹根的子樹 $T(B)$，並令 $G' = G \setminus (TAC(T(A)) \cup TAC(T(B)))$，找出 $T(A)$ 中 VS 節點到 $T(B)$ 中 VS 節點間，在 G' 上的最短路徑。假設 $SP_{G'}(x, y)$ 表示在圖 G' 中所找到最短路徑的兩節點 x 與 y，$x \in CS(T(A))$，$y \in CS(T(B))$，將子樹 $T(A)$ 與 $T(B)$ 加以連接 $l_{G'}(x, y)$，更新為核心樹 T_1，使得 $T_1 = T(A) \cup l_{G'}(x, y) \cup T(B)$。

(2)群播群組的來源節點落於 CBT 上，且來源節點

具有 VS 功能

因來源節點具有 VS 功能，故不需再做任何的修改，只要將群播群組中成員，逐一找尋距離最近且具有 VS 功能的節點或葉子節點加以連接(若遇距離相同時，則以節點編號(index)最小為選擇)，但不可再經過已加入群播樹(multicast tree)之 TaC 節點,最後將在 CBT 中未用於傳送資料的節點之間的路徑移除。

(3)CBT 不含 source 節點

因核心樹中不含有來源節點，因此需先將來源節點與核心樹中距離最近的 VS 節點相連，再將群播群組中成員，逐一找尋距離最近且具有 VS 功能的節點或葉子節點加以連接(若遇距離相同時，則以節點編號(index)最小為選擇)，但不可再經過已加入群播樹(multicast tree)之 TaC 節點，最後將在 CBT 中未用於傳送資料的節點之間的路徑移除。建構群播樹的演算法,其虛擬碼(pseudocode)下圖 8 所示。

Multicast tree routing algorithm

Input : a graph G(V, E) ;
multicast sessions M={M_i| M_i= S_i, D_i}, i=1, 2, …, m ;
D_i = {d_{i1}, d_{i2}, …, d_{im}} ;
CS = {VS_1, VS_2, …, VS_k} ;

Output : a multicast tree ;

Step 1. perform the CBT or LDCBT algorithm to find the core-based tree T_1 ;
if the core-based tree not found, let $T = \phi$ and return channels= ∞;
otherwise find the shortest path $SP_G(source, y)$ on the graph G,
where $y \in CS(T_1)$;
for each multicast session M_i perform Steps 2, 3, and 4 ;

Step 2. if the source node S_i in the core-based tree T_1, and S_i has no VS capability perform Step 3 ;
otherwise goto Step 4 ;

Step 3. A = S_i ;
for the sub-tree rooted at A, find the farthest VS node (say B), and
construct the path $BLP_{T_1}(A, B)$, where $B \in CS$;
divide the tree T_1 into two sub-trees T(A) and T(B) by removing the path $BLP_{T_1}(A, B)$;
construct G'=G\(TAC(T(A)) ∪ TAC(T(B))) ;
find the pair of nodes (x, y) with minimal distance from u to v in G', that is $d(SP_G(x, y)) \le \forall d(SP_G(u, v))$, for all u, v, when $d(SP_G(x, y))$ be the length of path $SP_G(x, y)$; where x, $u \in CS(T(A))$, and $y, v \in CS(T(B))$;
if ($d(SP_G(x, y)) == \infty$) ;
report T_1 not found and given the number of channels is ∞ ;
then construct T_1 by T_1=T(A) ∪ T(B) ∪ $l_G(x, y)$;

Step 4. **while** $D_i \neq \phi$;
{
construct graph G'=G\TAC(T_1) ;
find the shortest path $SP_G(x, y)$ in the graph G', where $x \in D_i$ and $y \in CS(T_1)$;
if ($d(SP_G(x, y)) == \infty$)
report T_1 not found and given the number of channels is ∞ ;
else construct T_1 by T_1=T(A) ∪ T(B) ∪ $l_G(x, y)$;
$CS = CS \cup leaf(T_1)$;
$D_i = D_i - \{x\}$;
}

圖 8 Multicast tree routing algorithm

4.3 波長指派(Wavelength assignment)

當每個群組(session)需求的群播樹(multicast tree)被決定後，可以計算出所使用的波長通道數目，但未能得知該群組群播樹之中所使用的波長，故需要做波長指派的動作，經過波長指派後，不僅能計算出每個群組的群播樹所需的波長量，更可透過每一個群播樹所使用的波長，將經過同一鏈結的路徑，指派不同的波長以避免波長碰撞的問題發生。

波長指派的演算法步驟如下:

步驟 1、依每個群組(session)中群播樹(multicast tree) T_i，$i = 1, 2, ..., m$ 的架構，並定義 $leaf(T_i)$ 表示在群播樹 T_i 上之葉子節點(leaf node)所成的集合，依深度搜尋演算法在此樹上由來源節點 S_i 視為樹根(root)節點向下搜尋，找出節點 S_i 與第一個 vs（$vs \in CS'$, $CS' = CS \cup leaf(T_i)$）節點連接的路徑，視為一個區段，並將此區段在群播樹 T_i 中移除，使得 $T_i = T_i \setminus l_{T_i}(S_i, vs)$，再以此 vs 節點為樹根向下搜尋，同上述方法，找到與第一個 CS' 集合中節點所連接的路徑，視為第二個區段，再將此節點至 CS' 集合中移除，依此分段方式反覆搜尋及移除，則可以將整個群播樹分割成數個區段，並以 $Seg_{ij}(vs_x, vs_y)$ 表示在此群播樹 T_i 中第 j 區段 vs_x 與 vs_y 二節點之間行經的所有鏈結，$vs_x \in CS'$, $vs_y \in CS'$。

步驟 2、重覆步驟 1 之方法，依序計算出每個群播群組中群播樹的區段。

步驟 3、波長依 $\lambda_1, \lambda_2, \lambda_3, ..., \lambda_w$ 優先權由高至低，依序指派給每個區段，並以 $W(e)$ 表示鏈結上可用波長的集合，當要指派波長給 $Seg_{ij}(vs_x, vs_y)$ 區段時，則找出此區段每一個所經過鏈結，並找出其中可用波長的集合，以 W_A 表示所有選出集合中同時都可使用波長的集合，使得 $W_{Seg_{ij}} = \bigcap_{\forall e \in Seg_{ij}(vs_x, vs_y)} W(e)$，並依 W_A 集合中下標最小的波長(假設為 λ_z)，指派給此一區段。指派完後同時需要更新每一個鏈結上可用波長的集合，使得 $W(e) = W(e) - \{\lambda_z\}$，$\forall e \in Seg_{ij}(vs_x, vs_y)$，以避免下一區段經過同一鏈結時，因使用了同一波長而發生波長碰撞的問題。

步驟 4、重覆步驟 3，直到將波長指派給了所有群播群組(multicast sessions)中的每個區段為止，其波長指派演算法的每個步驟之虛擬碼如下圖 9 所示。

Wavelength assignment algorithm

Input : a graph G(V, E) ;
multicast sessions M={M_i| M_i= S_i, D_i}, i=1, 2, …, m ;
D_i = {d_{i1}, d_{i2}, …, d_{in}} ;
CS = {VS_1, VS_2, …, VS_k} ;
$j \leftarrow 1$, $SEG = \phi$;

Output : assign the wavelength to multicast tree ;

Step 1. for each multicast session M_i, perform Steps 2, and 3 ;

Step 2. **while** multicast tree $T_i \neq \phi$;
{
perform the DFS algorithm to find the shortest path $SP_{T_i}(x, y)$ in the multicast tree T_i, where $x = root$ and $y \in (CS \cup leaf(T_i))$;
find segment $Seg_{ij}(x, y)$ and let $SEG = SEG + Seg_{ij}(x, y)$;
j++, root← y ;
$T_i = T_i \setminus l_{T_i}(x, y)$;
}

Step 3. **for** all segments in SEG ;
{
pick a segment say $Seg_{ij}(x, y)$;
find out the wavelengths not used by the segment $Seg_{ij}(x, y)$, and let $W_{Seg_{ij}} = \bigcap_{\forall e \in Seg_{ij}(x, y)} W(e)$;
assign the minimal wavelength $\lambda_a \in W_{Seg_{ij}}$ to $Seg_{ij}(x, y)$, and $W(e) = W(e) - \{\lambda_a\}$, for all $e \in Seg_{ij}(x, y)$;
}

圖 9 Wavelength assignment algorithm

為計算出所有群播群組(multicast sessions)所需的波長數目，以評估出 routing 方法的效能，因此以 CBT 的方式，依照表 1 中所給定的 multicast session 的需求，建構出三個群播樹(multicast trees)。並執行上述波長指派的演算法，以計算出所使用的波長通道及波長數。

5. 實驗結果

為了評估所提出基因演算法之效能，所有演算法均以 C 語言撰寫程式，在 Pentium Ⅳ 2.8 GHz CPU、512MB RAM、Windows XP 平台下，給予隨機的網路架構$|V|$=50，每個節點分支度小於 10，對於不同的 population (pop)、mutation rate (p_m)、配置 VS 節點的個數(k)、目的節點個數(d)、multicast session 數(M)與所提出之三種不同 routing 的方法加以評估其效能，以期許在求解最佳化的過程中，能夠快速找到最佳的解以符合我們的需求。

(一) 當 V、k、d、M、pop 為固定值，對於不同 pm 值，以評估時間之長短與 channels 數使用多寡之比較，下圖 10 中為|V|=50, k=5, d=7, |M|=50 與 pop=100，在 CBT 下，不同 p_m 值對於波長通道數之比較，以執行五次基因演算法下取平均之數據。當 p_m=0.2 時可以得到較好的收斂效果。

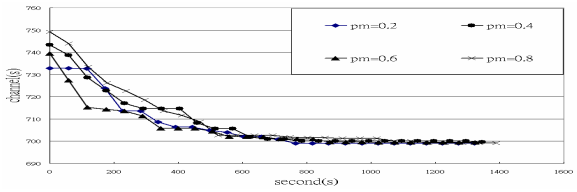

圖 10 不同 pm 值對波長通道數與時間之比較

(二) 當 V、k、d、M、pm 為固定值，對於不同 pm 值，以評估時間之長短與 channels 數使用多寡之比較，下圖 11 中為|V|=50,k=5, d=7, |M|=50 與 pm=0.2，在 CBT 下，不同 pm 值對於波長通道數之比較，以執行五次基因演算法下取平均之數據。當 pop=400 時可以得到較好的收斂效果。

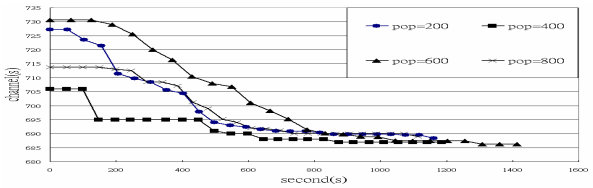

圖 11 不同 pop 值對波長通道數與時間之比較

(三) 當 V、d、M 固定時，對於 GA、Random 及 Max-degree(選取分支度最大的前 k 個節點放置)三種不同的 VS 節點配置方法，以評估 VS 節點使用數目與 channels 數使用多寡之比較，下圖 12 中為|V|=50, d=7 與|M|=50，在 CBT 下，配置不同 VS 節點數目對於波長通道數之比較，其中以 GA 方式使用了最少的波長通道數。

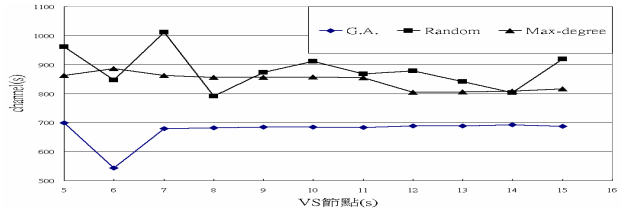

圖 12 不同 VS 值對波長通道數之比較

6. 結論

在本論文中，提出了以基因演算法計算出不同網路架構，在靜態需求下，求出最佳的 VS 節點配置位置，並以建立核心樹為基礎共享樹的方式，使這些少量擁有 VS 功能的節點都能藉由此與群播成員連接來傳送與接收資料，發揮了 VS 節點中具有光分割與波長轉換功能的最大效益。由實驗中可看出，以核心樹為基礎共享樹的方法，當分別以基因演算法、隨機以及分支度最大的方式配置 VS 節點位置時，使用基因演算法能最有效的節省了建立群播樹時，波長通道數目的使用，並以充分節省了 VS 節點所花費的成本。

致謝:

本論文由國科會計劃 **NSC-95-2221-E-018-012** 部份補助。

參考文獻

[1] A. Ballardie, “Core Based Trees (CBT version 2) Multicast Routing,” RFC, No. 2189, September, pp.1-23, 1997.

[2] Aijun Ding, Gee-Swee Poo and Sun-Teck Tan, “An expanded graph model for MCRWA problem in WDM networks,” IEEE Conference on Local Computer Networks, No. 6, November, pp.557-564, 2002.

[3] D. Estrin, D. Farinacci, et al., “Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification,” RFC, No. 2362, June, pp.1-66, 1998.

[4] Mitsuo Gen, Runwei Cheng, “Genetic Algorithms and Engineering Design,” John wiley & Sons, Inc, December, 1996.

[5] Kuo-Chun Lee, Victor O. K. li, , “A Wavelength-Convertible Optical Network,” IEEE Journal of lightwave technology, Vol. 11, No. 5, May-June, pp.962-970, 1993.

[6] N. Sreenath, C. Siva Ram Murthy, “Virtual source based multicast traffic routing in IP-over-WDM networks,” Journal of High Speed Networks, Vol.13, No. 4, pp.265-281, 2004.

[7] N. Sreenath, N.Krishna Mohan Reddy, G. Mohan and C. Siva RamMurthy, “Virtual Source Based Multicast Routing in WDM Networks with Sparse Light Splitting,” Workshop on High Performance Switch and Routing, May, pp.141-145, 2001.

[8] D. Waitzman, C. Partridge, and S. Deering(editors), “Distance Vector Multicast Routing Protocol,” RFC, No. 1075, November, pp.1-23, 1998.