

# 在無線群播核心網路中提供具服務品質保證之研究

林添財\* 李健銘\*\* 王彥傑\*\*\* 吳中實\*

\*國立中央大學通訊工程研究所

\*\*正歲精密股份有限公司

\*\*\*國立聯合大學電子工程系

E-mail:tslin\_tslin@yahoo.com.tw

## 摘要

隨著即時(Real Time)應用的需求增加，無線網路的普及，多媒體的應用，如視訊會議、遠距教學等，這類的應用是以群播(Multicast)為基礎，加上需要有服務品質(Quality of Service, QoS)的保證，這也使得具服務品質保證的群播路由演算法更顯得重要。

在網際網路通信協定下的移動性支援(Mobility Support in IP)中，訂定了遠端訂閱(Remote Subscription)和雙向轉傳(Bi-directional Tunneling)兩種方法來支援移動式網路環境下的多點傳輸技術。本文提出的改良式基因演算法(Genetic Algorithm, GA)係以 Remote Subscription 為多點傳輸技術，在核心網路(Core Network)中結合最小成本生成樹(Minimum cost Spanning Tree, MST)與一般性 GA 之一種改良式演算法，它以單播(Unicast)的方式建置群播樹(Multicast Tree)，根據群播應用所需的 QoS 參數，如頻寬保證(Bandwidth Guarantee)、端對端延遲(End-to-End Delay) 和 頻寬利用率(Bandwidth Utilization)等，計算出一個滿足各項 QoS 參數之群播樹，也針對以一般性 GA 為基礎的群播路由演算法最大缺點—運算時間做改善，讓具 QoS 保證之群播路由演算法，不因運算時間過長而失去具服務品質保證的本意，實驗結果證明改良式 GA 可有效改善一般性 GA 與 MST 演算法之效能。

**關鍵詞：**無線網路，群播，服務品質，最小成本生成樹，基因演算法。

## 1. 前言

近年來，隨著無線網路技術的成熟，有越來越多的人透過無線設備連接到Internet上，希望能夠隨時隨地的使用網路。支援移動性成為Internet發展的必然要求，為了滿足行動用戶自由遊走於各網路的需求，Internet Engineering Task Force (IETF) 組織提出了移動式網際網路通信協定(Mobile IP)[6]的架構，稱為“在網際網路通信協定下的移動性支援”(Mobility Support in IP)[4][5]。

另一方面，IP Multicast 是一種能同時傳送資料給多個接收者的一樣技術，具有網路利用率高、能節省發送者自身的資源、可擴展性較強的優點，

在大量新型網路應用(例如，視訊會議、遠距教學、即時訊息發布等等)中發揮很大的作用，因此 IP Multicast 一直是國際上互聯網路科學研究領域中的熱門研究課題。但是，由於 IP Multicast 一直沒有在 Internet 上得到大規模的廣泛應用，為此，研究人員提出了應用層(Application Layer)的群播等解決方案，並展開了廣泛的研究，期望能夠實現廣域網路(Wide Area Network)範圍內的群播服務，同時還希望能在無線網路中實現群播服務。

移動和群播的結合給兩者都帶來了新的挑戰。在移動環境中，群播不僅要管理群播組(Multicast Group)成員、建立和維護群播樹，還需要解決成員位置動態變化的問題。但現在 Internet 中使用的群播路由協定(Multicast Routing Protocol)，如距離向量群播路由協定(Distance Vector Multicast Routing Protocol, DVMRP)，核心樹協定(Core Based Tree, CBT)，群播開放式最短路徑優先協定(Multicast Open Shortest Path First Protocol, MOSPF)，獨立群播協定(Protocol Independent Multicast, PIM)等，通常假設其成員是靜態的，而且沒有考慮成員位置動態變化的情況。如果每次節點移動後都當作一個新加入的成員，則會給群播組的管理和群播樹的維護帶來過多的花費，增加網路的負擔。因此本文提出改良式 GA 來處理動態變化之群播的建置時間，達到路由最佳化，並降低切換延遲與降低封包遺失率等，讓 IP 群播服務功能與移動特性能有效結合。

本文的第二節，將簡介現有的群播路由演算法(Multicast Routing Algorithm)、協定及其對 QoS 保證的解決方案。第三節中將詳細闡述本文提出在無線群播環境下保證服務品質的方法。第四節則說明模擬環境、相關參數及模擬比較結果。第五節為本文結論。

## 2. 相關研究

實現群播通信最普遍的方法是建置群播樹，因為群播樹是將訊務沿著樹枝(Branch)到達不同的接收端，訊務的複製僅在分叉處進行，可有效降低網路的負載。

### 2.1 群播路由演算法及協定

Internet上的群播可以分成兩種類型的路由協定：以源節點為基礎(Source-based)和以核心為基礎(Core-based)兩種。

以源節點為基礎的方法中，以發送端為源節點(Source Node)，並為根節點連結到每個群播組成員，封包由發送端送出經由群播樹的鏈路到所有目的端，這類方法有以下幾種協定使用：DVMRP、MOSPF、密集模式獨立群播協定(Protocol Independent Multicast Dense Mode, PIM-DM)和明確要求單一源節點協定(Explicitly Requested Single Source Multicast, EXPRESS)[9]。

以核心為基礎的方法中，群組中裡會有一個節點(Node)被選為核心點(Core Node)，樹結構即以其為根節點，發送端經由核心點透過樹的鏈路將封包送至各目的端，這類方法有以下幾種協定使用：稀疏模式獨立群播協定(Protocol Independent Multicast Sparse Mode, PIM-SM)、CBT和簡單群播協定(Simple Multicast, SM)[7]。

## 2.2 具服務品質保證的群播路由

要滿足服務品質的需求並成為一個好的群播路由協定需考量以下幾點因素：(1)改善成員加入群組的成功機率 (2)加入路徑所需的花費最小 (3)加入時間最小和適用在大型網路。基於新成員如何連結到群播樹，群播路由協定可以分成兩大類：單一路徑路由(Single Path Routing, SPR)和多重路徑路由(multiple-path routing, MPR)。單一路徑路由提供新成員單一條路徑連結到群播樹，例如 CBT 和 PIM，而多重路徑路由則是提供多條路徑給新成員，例如 Spanning Join、QoS Multicast Internet Protocol(QoSMIC)[11]、QoS Multicast Routing Protocol(QMRP)[10]和 Parallel Probing[3]。

雖然這些方法有考慮某部份的需求，但未考慮到上述一個好的群播路由協定所需的所有要求，例如，Spanning Join 和 QoSMIC 因為他們的泛洪(flooding)方式，在 Internet 上會造成大量的額外 Overhead，因此並不適用在 Internet 上，而 QMRP 在多重路徑搜尋方式也有相同的問題，當他從單一路徑搜尋轉換成多重路徑搜尋時，還會增加更多的加入時間；雖然 Parallel Probing 有考慮到很多指定的目的，但其效能表現仍須根據中間目的點(Intermediate Destination, ID)的選擇。

## 2.3 在移動網路中群播樹的重整

使用者在無線網路環境下移動，會使得原本的群播組的大小有所改變，也因為移動性造成了群播組成員有不明確的動態成員問題產生。針對移動式環境下的群播問題，IETF 組織的 Mobile IP 工作小組[6]提出了兩種建議方式，分別是 Remote Subscription 和 Bi-directional Tunneling[4]，後者因路由沒有最佳化而不適合提供 QoS，以下就以本文

使用的 Remote Subscription 做說明：

在無線網路的環境中，由於主機具移動性(移動主機，Mobile Host)，這時在移動式網際網路協定裡規定的第一種應變的方法就是 Remote Subscription；當移動主機進入新的網域時，主機會以在新網域取得的新暫時位址，再重新對新網域中所屬的群播路由器發出一個新的加入群組的 IGMP 訊息封包，接著群播路由器就依照原本群播的步驟加入主機所要求的群組，然後主機就又可以接收到群組的封包了。

由於 Remote Subscription 是在每次移動到新的網域之後都重新執行一次加入群組的動作，讓整個邏輯群播樹再重建一次，所以可以得到最佳化的繞送路徑(也就是路由最佳化，Routing Optimum)，但也正因為每次交遞之後都需要重新的加入群組，這時就需要一個快速的演算法，以節省路徑切換的時間，進而降低封包遺失的機率。

## 3. 改良式基因演算法

為了有效建立具服務品質保證之群播樹，必須要有一些服務品質保證的相關參數的設定。文獻[1][2][8][12]提出以 GA 來求最佳化之群播樹，GA 在求最佳化解上的貢獻不小，但所需要的運算時間較為一般的演算法，如 MST 要來的多，因此本文希望能提出一個減少運算所需的時間，以適合移動式網路，並能達到接近最佳化解(Near-optimum)的一項演算法，此演算法是結合 MST 與 GA 的概念而成的，本文稱此演算法為改良式 GA，而以傳統 GA 來求取最佳群播樹的方式則稱為一般性 GA。

### 3.1 服務品質保證之相關參數 (QoS parameters)

為了要發展出一個多項目標(Multi-objectives)的服務品質路由演算法，本文參考[1]將研究重點放在滿足三個主要的目標參數來決定群播路由，這三個參數為：(1)端對端延遲時間；(2)頻寬保證；(3)頻寬利用率。

依據上述要求，本文建立網路模組和制定目標函數使達到一個最佳化。為了要達到這三個參數的最佳化，本文假設網路具有以下幾點特性：

- (1) 每個鏈路皆假設為服務佇列(Service Queues)，也就是封包被傳遞且都能得到服務。
- (2) 服務率(Service Rate)設定為遵循 Poisson 分布(Poisson Distribution)，使得服務時間呈指數分佈(Exponential Distribution)，鏈路延遲(Link Delay)的服務時間  $\mu$  係根據指數分佈，路徑(Path)是包含  $k$  個跳數(Hop)，沿著整條路徑的延遲可以看成  $k$  個獨立的指數隨機變數的總和，Erlang-K 分布(Erlang-K Distribution)基本上是  $k$  個獨立指數隨機變數的迴旋積分(Convolution)，每個都有相同的平均值(Mean)，

因此，上述討論包含  $k$  個鏈路，整個群播路由的延遲根據 Erlang-K 分布，包含長度為  $k$ ，路徑為  $p$ (發送端到一整個群播目的端)，其延遲時間  $d_p$  小於  $t$  的機率為：

$$\Pr(d_p < t) = \frac{\mu^k t^{k-1} e^{-\mu}}{(k-1)!} \quad (3.1)$$

依據古典機率理論，延遲時間為  $d$  的群播樹( $\Gamma$ )其機率可以寫成群播樹上獨立路徑的延遲時間的積：

$$\Pr(\Gamma) = \prod_{p \in \Gamma} \Pr(d_p < t) \quad (3.2)$$

為了找到最佳的路徑，本文試著將(3.2)式最大化。

- (3) 假設服務率和傳送率(Transmission Rate)滿足 Poisson 分布，一個頻寬容量為  $B$  的鏈路  $l$ ，其機率可表示為：

$$\Pr_l(B) = \frac{\mu^B e^{-\mu}}{B!} \quad (3.3)$$

用同樣的方法，滿足整個群播樹的頻寬機率為：

$$\Pr_\Gamma(B) = \prod_{l \in \Gamma} \Pr_l(B) \quad (3.4)$$

試著將(3.4)式最大化。

- (4) 頻寬利用率通常是設定為群播路徑上提供的最大剩餘頻寬(Greatest Residual Bandwidth)，全部的剩餘頻寬表示  $\sum_{l \in E} (c_l - b_l)$ ， $c_l$  是鏈路  $l$  的容量， $b_l$  是頻寬分配，對於整個群播樹  $\Gamma$ ，其頻寬利用率可表示為：

$$R(\Gamma) = \frac{\sum_{l \in E} (c_l - b_l)}{\sum_{l \in E} c_l} \quad (3.5)$$

試著將(3.5)式最大化。

- 從這三個參數可求得一適應值函數(Fitness Function)：

$$F = F_1 + F_2 + F_3 = \Pr(\Gamma) + \Pr_\Gamma(B) + R(\Gamma) \quad (3.6)$$

本文將適應值函數求取最大化。

## 3.2 最小成本生成樹(MST)

找出 MST 有三種不同的演算法：Kruskal、Prim 與 Sollin，本文採用的是 Kruskal 的方法，圖 3-1 是 MST 演算法表示式：

```

T = {};
while ( T contains less than n-1 edges && E is not
       empty ) {
    choose the least cost edge ( v , w ) from E ;
    delete ( v , w ) from E ;
    if ( ( v , w ) does not create a cycle in T )
        add ( v , w ) to T ;
    else
        discard ( v , w );
}
if ( T contains fewer than n - 1 edges )
    printf( "No spanning tree \n" );

```

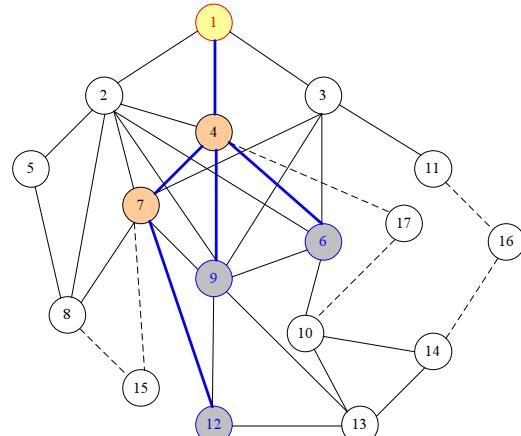
圖 3-1 Kruskal 之 MST 演算法

## 3.3 基因演算法(GA)

GA 法則的基本理論是由 Holland 於 1975 年首先提倡，是基於自然選擇過程的一種最佳化搜尋機構。

應用 GA 法則來解最佳化問題時，需先將所要搜尋的所有參數編碼成稱為染色體(Chromosome)的離散(Discrete)或二元(Binary)字串(String)，然後隨機地重覆產生  $N$  個初始物種(字串)，再依據求解之條件來計算適應值(Fitness Value)，適應值高的物種將被挑選至交配池(Mating Pool)中，此即複製過程，再依交配及突變過程的運算，即完成一代的基因演算法則，如此重覆下去以產生適應性最強的物種。

文獻[1][2][8][12]中所提之物種其染色體編碼方式如圖 3-2 所示，由發送端到其中一個目的端之路徑為一基因，基因與基因之間以「-1」表示間隔。複製與交配的過程與一般的 GA 步驟相同，唯其突變發生的過程，是從族群裡挑選一個染色體與交配後的染色體進行交配的動作。一般性 GA 如圖 3-3 所示。式子中的  $\varepsilon$  為誤差值，當新的適應值比原本的適應值好一定的值( $\varepsilon$ )，則收斂。



1	4	6	-1	1	4	9	-1	1	4	7	12	-1
---	---	---	----	---	---	---	----	---	---	---	----	----

圖 3-2 一般性 GA 之編碼方式

<p><i>Input:</i> A given network <math>G = (V, E)</math>, a source and a set of destinations.</p> <p><i>Output:</i> A set of Near-optimal multicast trees.</p> <p>Obtain the initial set of multicast trees (input: source, destinations);</p> <p>Map each of the multicast trees to a string consisting of network nodes;</p> <p>Generate the initial population from these set of strings;</p> <p><i>Repeat</i></p> <p style="padding-left: 2em;">Calculate the initial fitness of three QoS parameters separately;</p>
---

```

Choose two strings at random;
Perform cross-over and mutation;
Obtain the new set of strings to generate new
population;
Until ( $\{fitness\}_{S_{new}} - \{fitness\}_{S_{previous}} < \varepsilon$ )

```

圖 3-3 一般性 GA

### 3.4 結合 MST 與一般性 GA 構想之改良式 GA

改良式 GA 是將一般性 GA 稍做改變，並加入 MST 的選取鏈路概念，兩者結合而成的一種求取最佳群播樹一種方式。其所用的編碼方式與一般性 GA 相同，是以發送端到目的端之路徑為一組串列，不同的目的端有不同的池子(Pool)，相同發送端到目的端之不同路徑的串列放在同一個池子，有 N 個目的端則有 N 個池子，每次從不同的池子各取出一組串列，每個池子都挑選完之後合成一組樣本樹(Sample Tree)，由於不同的路徑結合而得的圖形，可能會有迴路形成，因此需判斷是否有迴路造成，並對造成迴路的鏈路做一個修正的動作，此修正動作是採用 MST 修正迴路的方法，修正過後成為一顆參考的群播樹，針對整顆樹作一個適應值的計算，計算後判斷是否到達收斂條件，未達收斂條件則進入取代(Replace)步驟；隨機挑選一個池子，選出一條路徑與原群播樹上相同目的端的路徑做取代動作，再計算是否達到收斂條件，如此反覆動作直到達到收斂條件。由於考量網路更新時間不能過長，故將收斂條件設定為兩組條件：

- 當新的適應值比原先適應值好一個誤差值( $\varepsilon$ )時，

- 當重複動作達到 X 次時，

第一個條件是參考一般性 GA 所定的，而第二個條件是考量運算所需時間，將測試次數限制在一定範圍內。本文所提之改良式 GA 如圖 3-4 所示。

```

Randomly select one set of path from all pools.
Loop detects and prunes poor fitness links.
Calculate fitness value and put it to  $\{fitness\}_{previous}$ 
Repeat X times
    Randomly choose one path list from one pool.
    Replace the path with correspondent path
    Loop detects and prunes poor fitness links.
    Calculate fitness value and compare with previous fitness value.
    If  $\{fitness\}_{new} > \{fitness\}_{previous}$  then
        If  $\{fitness\}_{new} - \{fitness\}_{previous} < \varepsilon$  then
            Put  $\{fitness\}_{new}$  to  $\{fitness\}_{previous}$ 
            Leave repeat
        Else
            Put  $\{fitness\}_{new}$  to  $\{fitness\}_{previous}$ 
            Continue repeat
        End if
    End if
End repeat

```

圖 3-4 改良式 GA

## 4. 模擬架構與數據分析

在本章節中，將對第三節所提的三種演算法，在不同情況下，其服務品質保證之情形進行比較，根據公式(3.2)、(3.4)計算所得的圖形曲線做對照及分析，並探討每種演算法在不同情形下的網路效能。

### 4.1 模擬拓樸

為了讓模擬產生的圖形能方便查看，本文採用 Borland C++ Builder 撰寫模擬程式，其模擬拓樸示意圖如圖 4-1 所示。

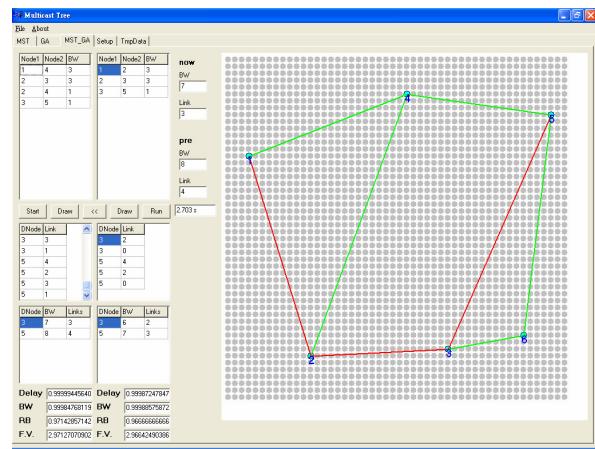


圖 4-1 模擬拓樸示意圖

本文模擬的網路是假設在核心網路中有 100 顆核心路由器(Core Router)，每顆核心路由器均有 3~5 條鏈路與鄰近之核心路由器相連，總共使用的鏈路為 200 條，每條鏈路的頻寬為 2.5Gbps，並假設有 1 個發送端，5 個目的端，傳遞的封包平均長度為 8K Bytes/message，表 4-1 是模擬時使用的參數。

表 4-1 模擬參數

鏈路容量( $c_l$ )	2.5 Gbps
封包大小(MPEG-II)	8 Kbytes/message
服務率( $\mu$ )	2.5 Gbps/8 Kbytes = $3.9 \times 10^4$ message/sec.
最小端點延遲時間( $t$ )	500 μsec.
最低頻寬需求(BR)	3 Mbps
交配率(cross-over rate)	0.8
突變率(mutation rate)	0.1
取代率(replace rate)	0.8
收斂次數(X)	100
收斂誤差值( $\varepsilon$ )	$10^{-5}$
網路拓樸大小(節點數)	100 個節點(node)
網路鏈路數量	200 個鏈路(link)

目的端數量	5 個節點
-------	-------

模擬過程中，設定目的節點從近發送端到遠離發送端，分成三個等級，以跳數來看，近為 5 個跳數，中為 16 個跳數，遠為 30 個跳數，每個等級各又分成三種網路負載狀況，此指鏈路已使用頻寬，輕為 10~20Mbps，中為 750~1000Mbps，重為 2.5~3.0Gbps，也就是有 9 種情況。離發送端遠近，目的要看出跳數，也就是公式(3.1)裡的 k 值與適應值之相關性；網路負載輕重狀況，目的要看出公式(3.3)裡的 B 值與適應值之相關性。從模擬分析的結果探討每種情況的各項數值變化。我們將一般性 GA 的收斂條件與改良式 GA 之收斂條件設定成相同，並以此為比較的基準。圖 4-2 是改良式 GA 模擬過程中的其中一個畫面，左邊顯示相關參數，右邊顯示網路架構與群播樹架構。

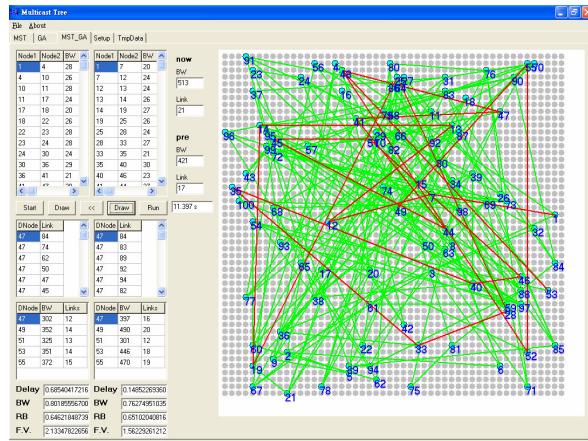


圖 4-2 程式畫面(改良式 GA)

## 4.2 實驗數據說明與比較

對 4.1 節所提的 9 種情況模擬後，每種模擬結果各會產生三個圖表，分別為：「一般性 GA 之適應值」、「改良式 GA 之適應值」與「計算時間」，前面兩個圖均以 MST 之成本函數(Cost Function)為對照組。本節僅以中距離中負載情況下的實驗數據做說明，如圖 4-3a、圖 4-3b 與表 4-2。

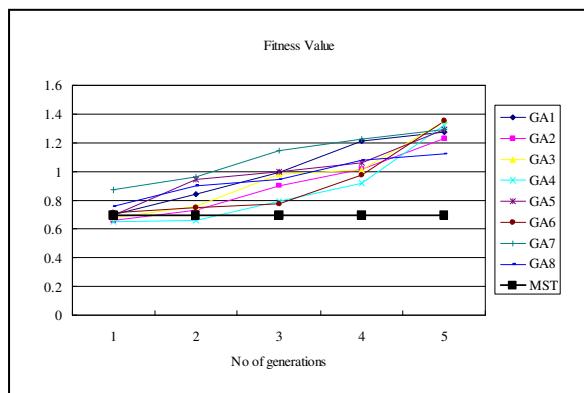


圖 4-3.a 一般性 GA, Fitness Value vs. Hop:16,  
Load:750~1000Mbps(中距離、中負載)

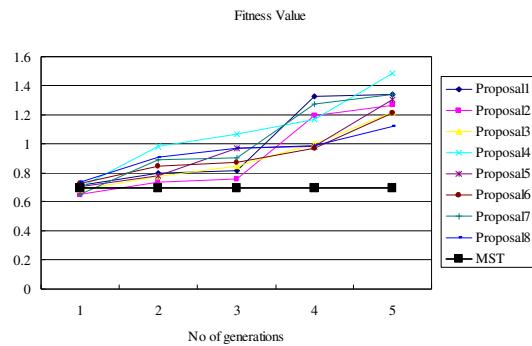


圖 4-3.b 改良式 GA, Fitness Value vs. Hop:16,  
Load:750~1000Mbps(中距離、中負載)

表 4-2 計算時間 vs. Hop:16,  
Load:750~1000Mbps(中距離、中負載)

一般性 GA	改良式 GA
15.79725 s	13.62713 s

一般性 GA 與改良式 GA 在每種情況，皆取八次試驗結果，每次試驗有五代，圖 4-3a 與圖 4-3b 中，不同的曲線代表使用不同的初始值得到的測試結果，每一條曲線代表廣播樹經由基因選擇而得到的適應值軌跡，軌跡上的每一點均由(3.6)式適應函數計算而得。表 4-2 則顯示在時間執行方面，一般性 GA 與改良式 GA 的比較。

## 4.3 數據分析與模擬總結

表 4-3 是將 4.2 節所做之 9 種情況的適應值彙總。在網路輕負載及中負載下，改良式 GA 與一般性 GA 之頻寬保證效能均優於 MST，在網路重負載的情況下，因網路頻寬不足而造成所計算出的適應值幾乎為零，此種情形下表示沒有一種演算法可計算出滿足各項需求的群播樹，所得的數據也不具參考價值。

表 4-3 網路情況與適應值(MST 以成本函數值表示)

負載	適應 距 離 演 算 法	近	中	遠
		MST	一般性 GA	改良式 GA
輕	MST	3.0	2.0	2.0
	一般性 GA	3.0	2.5	2.0
中	MST	1.7	0.69	0.68
	一般性 GA	1.8	1.3	0.66
重	MST	~0	~0	~0
	一般性 GA	~0	~0	~0

改良式 GA	~0	~0	~0
--------	----	----	----

表 4-4 則對 4.2 節所做之 6 種情況的執行所需之時間彙總。

表 4-4 網路情況與程式執行所需之時間

負載 演算法	計算時間 距離	近	中	遠	
		一般性 GA	改良式 GA	一般性 GA	改良式 GA
輕	一般性 GA	9.9s	16.4s	22.7s	
	改良式 GA	8.6s	14.0s	19.7s	
中	一般性 GA	9.6s	15.8s	21.3s	
	改良式 GA	8.8s	13.6s	18.7s	

從以上所得的數據，我們可以看出，在一般較實際的網路狀態(距離由近到中，負載由輕到中)下，改良式 GA 所求的之適應值與一般性 GA 的適應值相當，而兩者均優於 MST，在執行所需之時間而言，改良式 GA 約只須一般性 GA 的 87%，有效改善約 13%的計算時間，也證明了改良式 GA 可有效改善一般性 GA 應用在群播樹的效能。

## 5. 結論

具服務品質保證之群播路由演算法是目前最熱門的一項主題，為了要同時滿足多項目標的服務品質要求，許多文獻常利用一般性 GA 來求得最佳解，但其演算所需的時間費時，為此，本文提出一個能夠達到一般性 GA 的效果又能減少運算所需時間的方法。

一般性 GA 中，由單播的方式建立的群播樹，會有迴圈形成的可能，但在一般性 GA 所提的方法中並未提及對迴圈如何處理，本文針對此缺點提出以一般性 GA 為基礎，結合 MST 對迴圈處理的方式，而成為改良式 GA，使收斂時間能夠縮短，所計算出的群播樹的適應值也與一般性 GA 效能表現相當。

從第四節的模擬結果可以看出，在運算時間上，根據跳數的多寡有所不同，改良式 GA 在各種情況下，運算時間皆比一般性 GA 時間短，證明改良式 GA 演算法具實務的可行性。

未來，可針對在網際網路中對群播樹做訊務管控探討，使珍貴的網路資源能達到最佳的利用率，讓網路業者能使用最小的投資，達到最大的效益，也讓使用者能得到自己所預期的服務品質。

## 參考文獻

- [1] A. Roy and S. K. Das, “QM<sup>2</sup>RP: A QoS-based Mobile Multi- cast Routing Protocol Using Multi-Objective Genetic Algorithm”, Wireless Networks, vol. 10, pp 271-286, May 2004.
- [2] A. Roy, N. Banerjee and S. K. Das, “An Efficient Multi- Objective QoS Routing Algorithm for Real-Time Wireless Multicasting”, In Proc. Of IEEE 55<sup>th</sup> Vehicular Technology Conference 2002.
- [3] A. Striegel and G. Manimaran, “A Survey of QoS Multicasting Issues”, IEEE Communications Magazine, vol. 40, pp 82-87, June 2002.
- [4] C. Perkins, “IP mobility support”, RFC 2002.
- [5] D. Johnson, C. Perkins, J. Arkko, “Mobility support in IPv6”, RFC 3775, IETF, 2003.
- [6] IETF Mobile IP Working Group.  
<http://www.ietf.org/html.charters/mobileip-charter.html>
- [7] J. Hou and B. Wang, “Multicast routing and its QoS extension: Problems algorithms and Protocols”, IEEE Networks, Jan./Feb. 2000.
- [8] N. Banerjee and S. K. Das, “Fast Determination of QoS- based Multicast Routes in Wireless Networks using Genetic Algorithm”, ICC 2001.
- [9] Q. Wu, J. P. Wu, K. Xu, Y. Liu, “A Survey of the Research on IP Multicast in Mobile Internet”, Journal of Software, March 2003.
- [10] S. Chen, K. Nahrstedt, and Y. Shavitt, “A QoS-Aware Multi- cast Routing Protocol”, Proc. IEEE INFOCOM, 2000.
- [11] S. Yan, M. Faloutsos and A. Banerjea, “QoS-Aware Multicast Routing For the Internet: The Design and Evaluation of QoSMIC”, IEEE/ACM TRANS. on Networking, vol. 10, Feb. 2002.
- [12] X. Cui, C. Lin, Y. Wei, “A Multiobjective Model for QoS Multicast Routing Based on Genetic Algorithm”, Proc. of International Conference on Computer Networks and Mobile Computing, 2003.