以虛擬路徑爲基礎的 ATM 網路多播繞徑分析

黎碧煌　　施勢帆　　朱玉鳳

國立台灣科技大學　電機工程系

台北市基隆路四段 43 號

lee@ccg.ee.ntust.edu.tw

摘要

ATM 網路中，虛擬路徑觀念的提出大幅簡化了交換機對細胞的處理，若配合較佳的資源管理策略，將可使虛擬通道的建立更快速更有效率。虛擬路徑的建構方式及資源管理配置的策略關係著系統運作的效能。本文提出一個以區域(zone)爲觀念的新架構及對應的虛擬路徑建立策略，並配合相關的頻寬控制法、重新繞徑策略、資源管理方式、多播繞徑策略等問題，分析此模型的各種特性。最後利用模擬程式驗證本模型的各項效能，如頻寬利用率、阻塞機率、重新繞徑比率、調整頻寬成本、回復率等等。

關鍵詞：ATM、虛擬路徑、利用率、阻塞機率、重新繞徑、多播繞徑

1. 簡介

各類型的廣域網路中，繞徑(routing)方式的優劣將影響系統是否能充分的運用資源並發揮應有的效能，例如資源管理的分配、傳遞的效能以及線路的利用率等等。虛擬通道(virtual channel; VC)及虛擬路徑(virtual path; VP)是 ATM 網路中相當重要的技術，它結合了分封交換(packet switching)及電路交換(circuit switching)的優點。虛擬路徑鏈路(VP link)爲實際網路上兩節點間的線路，串聯幾條虛擬路徑鏈路形成虛擬路徑連線(VPC; VP connection)，也就是虛擬路徑；路徑上的中繼節點，即爲網路上實際節點或交換機，稱爲虛擬路徑交換節點(VP switching node)。一條虛擬路徑正好對應一條虛擬通道鏈路(VC link)，虛擬通道連線(VCC; VC connection)由數條虛擬通道鏈路組成，即爲虛擬通道；虛擬路徑上的終端節點就是虛擬通道內各節點，稱爲虛擬通道交換節點(VC switching node)。因此可將幾條虛擬通道內相同的虛擬通道鏈路捆成一束，當做是在同一虛擬路徑上傳輸。虛擬路徑的引入簡化了 ATM 網路的資源管理，並減少其管理的成本，在此我們提出一個路徑建立的策略，使得虛擬路徑的應用能將網路的資源做有效合理的分配。

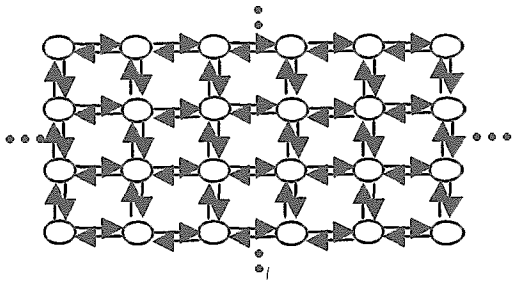
目前的 ATM 網路中，虛擬路徑的建立是採用平均分配或任意指定的方式來完成[1-3]；交換機與交換機之間均沒有一個固定的連結規則，以致於虛擬路徑的建立也無規則可循；所以我們以一網狀(mesh)連結的架構及區域(zone)觀念做爲虛擬路徑建立的原則[4]。這種網路架構不僅容易擴充，同時也簡化了交換機中製作繞徑表(routing table)的麻煩。一方面有規則可循，一方面也藉由區域的觀念劃分資訊蒐集管理的區域。

近年來，有許多應用需要藉助網路的多播連結(multicast connection)來達成資訊的傳送，例如：遠距教學(distance learning)，多媒體(multimedia)，視訊會議(video conferencing)，虛擬實境(virtual reality)等。本文針對以虛擬路徑爲基礎的 ATM 網路，提出三種多播連結繞路演算法建立多播連結，透過模擬求出不同交通流量下各演算法之呼叫阻塞率(call blocking rate)，並比較此三種多播連結繞路演算法之優劣。

2. 網路架構及策略描述

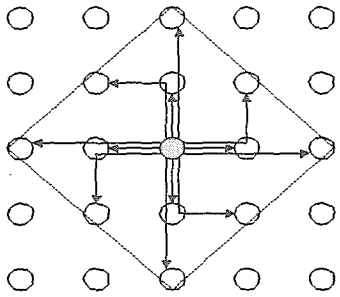
2.1 網路模型

以虛擬路徑爲基礎的 ATM 網路，是在邏輯上將虛擬路徑的端點視爲一個節點，虛擬路徑視爲一條傳輸線路。如圖一所示的實際網路架構，它是一個網狀對稱的結構，圖中每一個圓圈代表一個交換節點(或稱 ATM 交換機)，帶箭號的直線爲實體線路的連線，每一個交換節點都在上下左右四個方向各有一進一出的單工(simplex)輸出入埠，節點與節點間的實體連線依照相關的位置與其它節點相連；亦即一個節點的輸出必定會經過相鄰的四個節點的其中之一。這樣的架構可以向四周無限的擴增，除了顧及未來的擴充性也兼顧到網路的無限性，亦有助於系統的分析。



圖一 網路架構實體連線示

我們採用的 ATM 虛擬路徑的建立[4]有別於其它的建立方式[1-3]。其建立方式是先計算由本身的交換節點到其它節點所需經過的節點數。爲了維持每一個節點有相類似的特性，我們限定每一條虛擬路徑所能經過的最多節點數，這也是與 X-Y 繞徑方式所不同的地方[5,6]，X-Y 繞徑方式雖有指定繞路方向但無限定繞路數目。在此假定之下每一個節點所能連接到最遠的節點所形成的範圍形成一個菱形，我們稱此菱形的範圍爲區域，區域大小是指一條虛擬路徑依規定所能經過的最多節點數，在區域的中心節點與區域內所有節點可做完整連線(full connection)，如圖二爲區域大小爲 2 所造成的區域，其虛擬路徑行走的方向則定爲逆時針的方向。



圖二 虛線所圍成的菱形代表區域的大小

以上的說明只是針對中心節點如何將資料外傳的方式，而其它的節點也都與中心節點一樣具有相同的型式；亦即網路上的任何節點都可是中心節點，也可以用相同的方式造出另一個區域，如圖三所示。如果傳送的目的地超過了一個區域的範圍時則有兩種傳遞方式，一種是使用以虛擬通道爲基礎(VC-based)的方式，另一種則是利用既有的虛擬路徑串連起來。如果使用後者的方式則可以串連的虛擬路徑通常可有許多組選擇，但我們仍延用相同的策略(逆時針的方式)，並以最近的路徑繞徑，如圖四所示。

2.2 交通流量模型 (traffic model)

區域的大小會影響傳輸線路上虛擬路徑數目的多寡。若區域大小爲 N，則可計算出一個由此交換機的一個輸出埠所傳送出來的可能連結的虛擬路徑連線數目會有 N(N+1)/2。如果再加上由別的交換機所建立而經過這個交換機的虛擬路徑連線，則全部的虛擬路徑連線數目可由下式求得：

$$2\sum_{i=1}^{N-1}\frac{i(i+1)}{2}+\sum_{i=1}^{N}i=\frac{N(N+1)(2N+1)}{6}\qquad N>1 \qquad (1)$$

式子(1)中 N 表示區域大小。當 N 愈大，雖然虛擬路徑連線可以跨越的範圍較大，但換來的代價是虛擬路徑連線的個數會成三次方成長，無形中也增加了交換機的負擔。另外，由於交換機內虛擬路徑識別碼(VPI; VP identifier)及虛擬通道識別碼(VCI; VC identifier)位元大小的限制使得虛擬通道識別碼的值不能超過 4096，因此也限制了區域的大小。由於我們必須評估區域的大小對系統的影響，我們考慮 N=2、3、4、5，並使得實體頻寬被切割的份數是四者的公倍數，而且每一個基本單位頻寬有 64 kbps 的傳輸速率(transmission rate)，所以我們取得 2310 單位爲適當之值。如果所有規劃頻寬都被取用，則總頻寬爲 2310*64 kbps = 147840 kbps = 144.375 Mbps。若採用 OC-3 的 155 Mbps，則未使用的部份尚可保留給傳送控制信號之用。

在交通流量模型上，我們採用以下的形式：虛擬通道連線的產生是依據波松(Poisson)分佈，每一個虛擬通道連線的持續時間爲指數(exponential)分佈，至於每一個虛擬通道連線的頻寬需求則採用幾何(geometric)分佈。當一個呼叫(call)產生時，它會到對應的虛擬路徑中取得頻寬，以基本頻寬爲單位，至少取得一個，至多取得 20 個。限制取得超過 20 個基本頻寬的目的是因爲這樣的機率較小，也不希望單一個呼叫佔據過大的頻寬。若真有必要，可以應用程式分成兩個或以上的呼叫來取得需要的頻寬。爲了表現出每個虛擬路徑連線有不同的需求流量密度，並藉此觀察其能否依設定的流量密度來做動態的頻寬分配的效能。

2.3 虛擬路徑的選擇及頻寬配置策略

在以虛擬路徑爲基礎的網路中，爲了快速搜尋路徑來建立連結，我們假定每一條虛擬通道最多不能繞徑超過二條虛擬路徑；也就是網路上任何一個虛擬通道只有兩種狀況，一種是透過一條虛擬路徑直接相連，一種是串連兩條虛擬路徑。我們針對最小負載路徑(least loaded routing; LLR)[7]及動態資源分配(dynamic resource allocation)[8, 9]來選擇虛擬路徑並加以分析討論。

過去的網路中是當虛擬通道建立之初就約定好所能接受的虛擬路徑數[1-3]，此後便無法再行變動，所以如果虛擬通道沒有足夠的頻寬可以滿足需求時，便只好藉由重新繞徑的策略來完成，如果仍然無法找到合適的路徑，則將造成阻塞。此即所謂定額頻寬配置法。在此我們引入動態頻寬的調整，也就是當頻寬不足時，以調整頻寬爲優先考慮，當頻寬調整失敗後才以重新繞徑來解決。此種策略的加入，減少了重新繞徑的比率，同時也明顯的得到較佳的頻寬利用率。此即動態頻寬配置法。

3. 多播繞徑策略

3.1 ATM 多播傳送呼叫的建立順序

多播傳送連結允許單一使用者同時對許多使用者建立通訊，多播傳送的連結建立首先被定義在 ATMF UNI V3.1 and ITU-T Draft Recommendation Q.2971。多播傳送呼叫建立步驟如圖五所示，圖中 A 使用者對 B、C 及 D 使用者建立多播傳送呼叫，假設 B 及 C 使用者是連結在同一個 ATM 交換機上(如 PBX)，D 使用者是連結在另一個 ATM 交換機上，多播傳送呼叫的專業用語中，呼叫的發起者(A)被稱爲根節點(root node) ，其他使用者(B、C、D) 被稱爲葉節點(leaf nodes) [10]。

3.2 群體分佈與建立多播連結的行爲分類 (1)

我們定義多播連結需求爲(s,D)，其中 s 是該連結的來源節點，D 是該連結的目的節點集合，集合{s,D}稱爲群體(group)，群體中的節點稱爲成員(member)，成員可隨時移出與加入群體[11]。本文假設對於多播連結的節點分佈在區域的範圍內；依每節點的能力可將多播連結的行爲分成兩類：

(1) 每一節點都有能力直接建立多播連結，即群體中每一節點都有成員名冊。換句話說，群體中每一節點都是多播路由器(multicast router)。

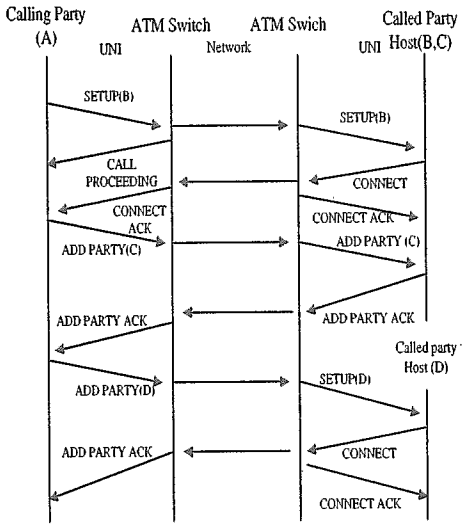
(2) 只有區域的中心節點有能力建立多播連結，即區域內的群體成員名冊只有區域的中心節點有記錄。換句話說，只有區域的中心節點是多播路由器。

若只有區域的中心節點是多播路由器，該群體在建立多播連結時，分成兩步驟。首先由來源節點與區域的中心節點建立連結後，再由區域的中心節點與目的節點建立連結。在建立多播連結的過程中，只要連結其中任一目的節點失敗該多播連結即是失敗。

3.3 多播繞徑問題分類

多播繞徑問題被分成靜態(static)和動態(dynamic)兩類[12, 13]。靜態的多播繞徑問題，即在建立連結的過程中不可以有任何葉節點的加入與移出；那麼在建立連結的過程中，可以將所有目的節點同時一起考量，透過繞徑演算法來完成連結建立。本論文採用的靜態繞徑演算法是 Prim's MST (minimum spanning tree) 繞徑演算法。動態的多播繞徑問題，即在建立連結的過程中可以有任

何葉節點的加入與移出；那麼在建立連結的過程中，無法將所有目的節點同時一起考量，必須一個節點接著一個節點透過繞徑演算法來完成連結建立。本論文採用的動態繞徑演算法是 IND (independent)和 SP(shortest path)繞路演算法[12]。



圖五 多播傳送呼叫建立步驟

3.4 IND 繞徑演算法

IND 繞徑演算法屬於動態繞徑問題演算法，將每一目的節點各別獨立與來源節點建立連結，其優點是幾乎不需要改變目前點對點繞路軟體及硬體就能完成多播連結，由於將每一目的節點各別獨立與來源節點建立連結，所以一虛擬路徑可能在來源節點與目的節點間所建立的連結樹中不只出現一次，造成資源浪費， IND 繞徑演算法(1)適用於群體的每一成員都有能力建立多播連結。IND 繞徑演算法(2)適用於只有區域中心節點有能力建立多播連結。

Algorithm 1: IND routing algorithm (1)

```
Remark: given a connection request (s, D)
T : = {{s},Φ}
while (D ≠ Φ)
    remove d from D
        for i = 1 to len do
            let p(s, d) be the path connecting d to s
            with maximum free capacity among all
            paths of length i connecting d to s
            if (the free capacity on path p(s, d) >=
            BW_R+BW_N)
                T: = T + p(s, d)
                break
            endif
        endfor
        if i > len
            block the connection
            free all circuits have been reserved
            abort the call setup procedure
        end if
end while
```

Algorithm 2: IND routing algorithm (2)

```
Remark: given a connection request (s, D)
for i = 1 to len do
    let p(s, z) be the path connecting zone center node
    z to s with maximum free capacity among all
    paths of length i connecting z to s
    if (the free capacity on path p(s, z) >= BW_R+BW_N)
        break
    endif
endfor
if i > len
    block the connection
    abort the call setup procedure
end if
T : = {{z}, Φ}
while (D ≠ Φ)
    remove d from D
    for i = 1 to len do
        let p(z, d) be the path connecting d to z with
        maximum free capacity among all paths of
        length i connecting d to z
        if (the free capacity on path p(z, d) >= BW_R
        +BW_N)
        T : =T + p(z, d)
        break
        endif
    endfor
    if i > len
        block the connection
        free all circuits have been reserved
        abort the call setup procedure
    end if
end while
```

3.5 SP 繞徑演算法

SP 繞徑演算法屬於動態繞徑問題演算法，比 IND 繞徑演算法複雜，該繞徑演算法是依據最短路徑的概念，將目的節點一個接著一個連結到已建立的部份連結樹，由於一虛擬路徑在來源節點與目的節點間所建立的連結樹中只出現一次，所以頻寬利用率較好及遺失率較小。SP 繞徑演算法(1)適用於群體的每一成員都有能力建立多播連結。SP 繞徑演算法(2)適用於只有區域中心節點有能力建立多播連結。

Algorithm 3: SP routing algorithm (1)

```
Remark : given a connection request (s, D)
T : = {{s}, Φ}
while (D ≠ Φ)
    remove d from D
for i = 1 to len do
    let p(T, d) be the path connecting d to T with
    maximum free capacity among all paths of
    length i connecting d to T
    if (the free capacity on path p(T, d) >=
    BW_R+BW_N)
        T : =T + p(T, d)
        break
    endif
endfor
if i > len
    block the connection
    free all circuits have been reserved
    abort the call setup procedure
end if
end while
```

Algorithm 4: SP routing algorithm (2)

```
Remark : given a connection request (s, D)
for i = 1 to len do
    let p(s, z) be the path connecting zone center node
    z to s with maximum free capacity among all
    paths of length i connecting z to s
    if (the free capacity on path p(s, z) >=
    BW_R+BW_N)
        break
    endif
endfor
if i > len
    block the connection
    abort the call setup procedure
end if
T := {{z}, Φ}
while (D ≠ Φ)
    remove d from D
    for i = 1 to len do
        let p(T, d) be the path connecting d to T with
        maximum free capacity among all paths of
        length i connecting d to T
        if (the free capacity on path p(T, d) >= BW_R
        +BW_N)
            T := T + p(T, d)
            break
        endif
    endfor
    if i > len
        block the connection
        free all circuits have been reserved
        abort the call setup procedure
    end if
end while
```

3.6 MST 繞徑演算法

MST 繞徑演算法屬於靜態繞徑問題的演算法，依據最小擴展樹(minimum spanning tree)的演算法 "Prim's minimum spanning tree algorithm，來建立連結樹； MST 繞徑演算法(1)適用於群體的每一成員都有能力建立多播連結。MST 繞徑演算法(2)適用於只有區域中心節點有能力建立多播連結。

Algorithm 5: MST routing algorithm (1)

```
Remark : given a connection request (s, D)
T := {{s}, Φ}
for i = 1 to len
    do while (D ≠ Φ)
        for each d belong to D, let p(T, d) be the
        path connecting d to T with maximum free
        capacity among all paths of length i
        connecting d to T
        if (the free capacity on path p(T, d) >= BW_R
        +BW_N)
            remove d from D
            T := T + p(T, d)
        else
            break
        end if
    end while
endfor
if (D ≠ Φ)
    block the connection
    free all circuits have been reserved
    abort the call setup procedure
end if
```

Algorithm 6: MST routing algorithm (2)

```
Remark : given a connection request (s, D)
for i = 1 to len do
    let p(s, z) be the path connecting zone center node
    z to s with maximum free capacity among all
    paths of length i connecting z to s
    if (the free capacity on path p(s, z) >=
    BW_R+BW_N)
        break
    endif
endfor
if i > len
    block the connection
    abort the call setup procedure
end if
T := {{z}, Φ}
for i = 1 to len
    do while (D ≠ Φ)
        for each d belong to D, let p(T, d) be the
        path connecting d to T with maximum free
        capacity among all paths of length i
        connecting d to T
        if (the free capacity on path p(T, d) >= BW_R
        +BW_N)
            remove d from D
            T := T + p(T, d)
        else
            break
        end if
    end while
endfor
if (D ≠ Φ)
    block the connection
    free all circuits have been reserved
    abort the call setup procedure
end if
```

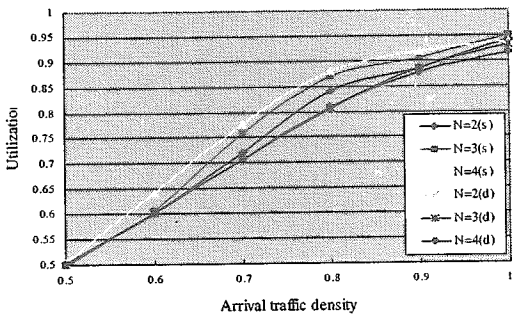
4. 模擬結果分析及討論

本節針對我們提出的硬體架構模型，配合前述之繞徑法則及各種資源頻寬配置策略進行程式模擬。模擬結果分爲兩大部分：網路區域大小之效能分析(4.1 節)及多播繞徑演算法之效能評估。4.2 節探討群體中每一節點都是多播路由器之情況，4.3 節則探討只有區域的中心節點是多播路由器之效能。我們主要的輸入參數是整體平均的流量密度，它是由虛擬路徑所發出的虛擬通道連線，稱爲需求流量密度，此一參數將爲大部份圖的橫座標，因輕載情況下其值甚小，所以圖中只顯示大於 0.5 的部份。

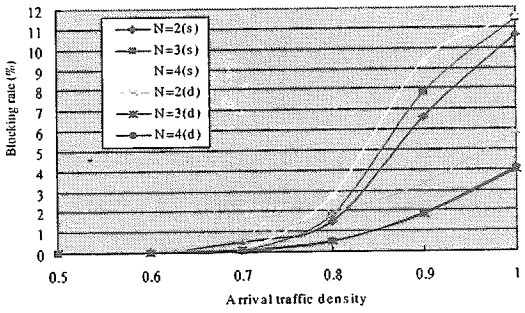
4.1 網路區域大小之效能分析

圖六至圖八所示爲在不同的流量下，三種區域大小(N=2,3,4)所表現出的頻寬利用率、阻塞機率及重繞徑比例。圖中(s)代表使用統計式的資源配置策略，(d)代表使用定論式的資源配置策略。圖六中，當輸入流量密度大於 0.5 之後，由於重新繞徑的影響，使得線路上的利用率比應有的利用率略高。其中動態頻寬配置法因減少了重新繞徑的比率，明顯地得到較佳的頻寬利用率。雖然如此，當輸入的流量大於 0.8 之後，這樣的趨勢將會減緩，因爲此時負載加重，重新繞徑亦不能解決而造成阻塞，由圖七亦可看出此種現象。很明顯地在阻塞機率方面，區域大小爲 2 時阻塞機率比區域大小爲 3 或 4 來得高，因爲在區域大小爲 2 的時候每個交換機上建立的虛

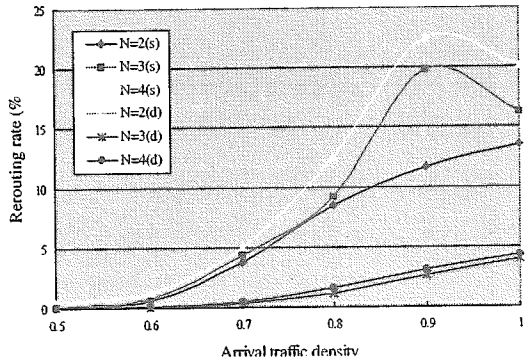
擬路徑數較少，能被選擇的也就不多，所以它阻塞的機率要比其它兩者高。其中動態頻寬配置法所得到的結果仍較佳。就利用率來說我們可以很清楚的看出動態配置頻寬的利用率幾乎與需求流量密度一樣，除了能較有效的使用頻寬外，也大大降低了阻塞機率。就重新繞徑的比例來說，動態配置頻寬的比例小於定額頻寬分配下的結果，如此將可以減少重新繞徑下所造成的不良後果，如八所示。區域大小愈大重新繞徑的比例相對較高，但是於定額頻寬分配下，負載過大時因爲阻塞機率增加使得重新繞徑的百分率反而下降。總而言之，當區域大小愈大時，頻寬控制的成本愈高、重新繞徑的頻率提高、頻寬利用率較大、阻塞的機率也愈大。但是，動態頻寬配置法優於定額頻寬配置法，尤其是利用率和阻塞機率。



圖六 不同流量下的頻寬利用率



圖七 不同流量下的阻塞機率

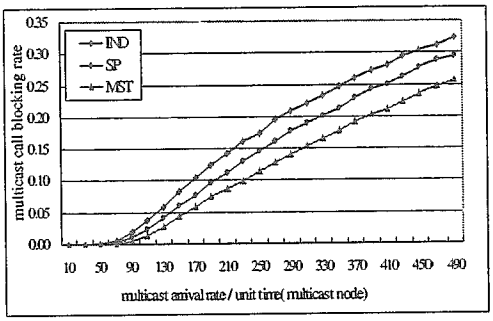


圖八 不同流量下的重新繞徑的比例

4.2 每一節點都可建立多播連結之效能分析

我們假定區域的大小取 n = 3，群體的成員個數爲 6，頻寬保留程度 0%，unicast 交通流量密度 50%，各演算法所得出多播呼叫阻塞率，如圖九所示。由圖中可明確看出在多播到達率大於 50 以後的多播呼叫阻塞率方面，MST 的表現最佳，SP 次之，IND 最差。

若區域的大小取 n = 3，群體的成員個數爲 6，unicast 交通流量密度 50%，頻寬保留程度 0%, 2%, 4%, 6%, 8% 和 10%，各演算法所得出多播呼叫阻塞率仍是以 MST 的表現最佳，SP 次之，IND 最差。



圖九 多播呼叫阻塞率(zone size=3, 群體成員=6,頻寬保留程度=%, 交通流量密度=50%)

4.3 透過 zone 中心節點建立多播連結之效能分析

我們假定區域的大小取 n = 3，群體的成員個數爲 6，頻寬保留程度 0%，unicast 交通流量密度 50%，各演算法所得出多播呼叫阻塞率，如圖十五所示。由圖中可明確看出在多播到達率大於 50 以後的多播呼叫阻塞率方面，MST 的表現最佳，SP 次之，IND 最差。若區域的大小取 n = 3，群體的成員個數爲 6， unicast 交通流量密度 50%，頻寬保留程度 0%, 2%, 4%, 6%, 8% 和 10%，各演算法所得出多播呼叫阻塞率仍是以 MST 的表現最佳，SP 次之，IND 最差。

5. 結論

由前面的探討，很明顯的可以看出使用動態頻寬調整的策略將優於固定頻寬調整，基本頻寬劃分得愈小也使得重新繞徑的比例減少。但這些優點都建立在交換機必須耗廢相當多時間做頻寬調整，過於頻繁的變動可能造成緩衝區沒有足夠的空間容納在調整時的暫存資料。本文主要討論在我們提出的架構下配合各種策略的應用，藉以得到一些相關數據及網路特性。

總而言之，當區域大小愈大時，頻寬控制的成本愈高、重新繞徑的頻率提高、頻寬利用率較大、阻塞的機率也愈大。但是，動態頻寬配置法優於定額頻寬配置法，尤其是利用率和阻塞機率特別明顯。其實在硬體速度夠快的前提之下，區域大小的增加將有以下的優點: 減少交換機的緩衝區、增加交換機所能控制的範圍、有效的運用虛擬路徑。

在多播繞徑方面，MST 演算法之多播呼叫阻塞率表現最佳，SP 演算法次之，IND 演算法最差。但硬體結構方面，IND 演算法的率優點是不需要改變現有的點對點繞路硬體，MST 演算法與 SP 演算法則必須做很大的調整。在建立多播連結花費的時間上，MST 演算法花費最長，SP 演算法次之，IND 演算法最短。由於硬體技術快速進步，硬體速度大大提升，加上市場需求而大量生產

降低硬體價格，而將這三種演算法在硬體的價格上與建立多播連結所花的時間上的差距縮小。因此，綜合評估起來，MST 演算法是較佳的選擇。

參考文獻

[1] Murakami K. and H. S. Kim, "Virtual Path Routing for Survivable ATM Networks," *IEEE/ACM transaction on Networking*, Vol.4, No.1, pp.22-39, February 1996.

[2] Hwang R. H., "Adaptive Routing in VP-based ATM Networks," *Journal of Information Science and Engineering* 11, pp.595-624, 1995.

[3] Gupta S., K. W. Ross and M. E. Zarki, "Routing in Virtual Path based ATM Networks," *GLOBECOM*, Vol. 1, pp. 571-575, 1992.

[4] B. H. Lee, S. S. Shih, C. C. Wu, and I. H. Lin, "A VP based ATM 網路繞徑及故障回復之分析," to appear on Journal of Technology.

[5] Boura Y. M. and C. R. Das, "Efficient fully adaptive wormhole routing in n-dimensional Meshes," *IEEE INFOCOM*, pp.589-595, 1994.

[6] Dally W. J. and C. L. Seitz, "Deadlock-free message routing in multiprocessor interconnection networks," *IEEE Trans. Computer*, C-36 (5), pp.547-553, May 1987.

[7] Hwang R. H., "LLR routing in homogeneous VP-based ATM networks," *IEEE INFOCOM*, pp. 587-593, 1995.

[8] Yamashita A., R. Kawamura and H. HADAMA, "Dynamic VP rearrangement in an ATM network," *IEICE Trans. Communications*, Vol. E80-B, No. 2, pp.289-294, Feb. 1997.

[9] Arvidsson Ake, Department of Communication Systems, Lund Institute of Technology, Lund, Sweden, "Real Time Management of Virtual Paths," *IEEE INFOCOM*, pp.1399-1403, 1994.

[10] Gary C. Kessler, and Peter V. Southwick, ISDN : Concepts, Facilities, and Services, 3nd edition, McGraw-Hill, 1996.

[11] Network Working Group, FRC 1112, "Host Extensions for IP Multicasting", Stanford University, August 1989.

[12] Waxman B. M., " Routing of Multipoint Connections", IEEE Journal on Selected Area in Communication, Vol.6, pp.1617-1622, December, 1988

[13] Hwang R. H., "Adaptive Muiticast Routing in Single Rate Loss Networks", INFOCOM'95, pp 571-578, April 1995.