

第五章 DLBP演算法與其它演算法之效能比較

5.1 網路模擬架構與模擬環境的介紹

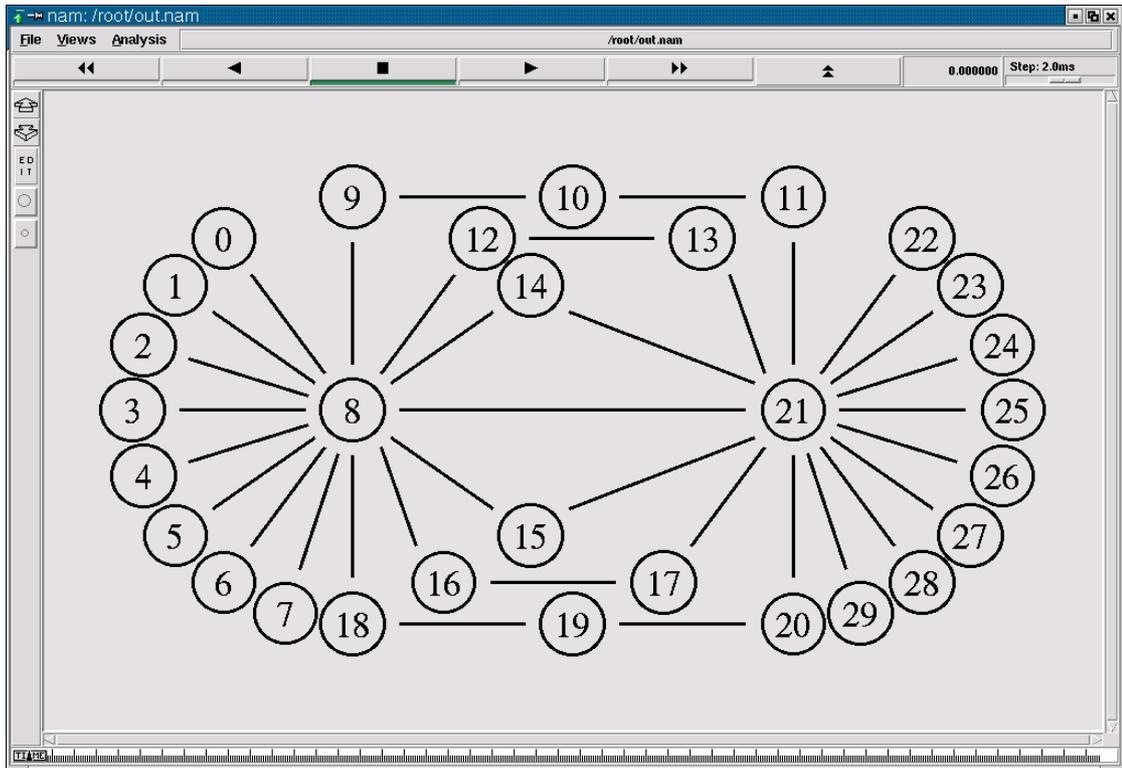


圖 59 用來模擬的網路拓撲

我們用來模擬的網路拓撲如圖 59，此網路拓撲中 N8 為 MPLS 之網域入口節點(Ingress Node)，N21 為 MPLS 之網域出口節點(Egress Node)，而 N0 到 N7 為產生資料流電腦設備，而 N22 到 N29 為接收資料流的電腦設備，而 N8 到 N21 為 MPLS 網域的核心路由器(Core Router)，而 N0 到 N7 與 N22 到 N29 的網路連線頻寬為 1Gbps，核心路由器之間的每一個網路連線頻寬都設定為 3Mbps。

5.2 DLBP演算法與DLB演算法之模擬與模擬結果比較之分析

我們利用圖 59 之網路拓撲來模擬 DLBP 與 DLB 演算法。我們提過 DLBP 演算法，乃是將修正後之 DLB 演算法加以延伸，加入優先權的觀念。

本節我們用相同的模擬情境來模擬此兩演算法，其目的為欲驗證DLBP演算法，不只比DLB演算法多出優先權的功能，並且其資料流的總輸出量和網路資源利用率仍稍優於DLB演算法。

我們將DLBP演算法的模擬環境參數設定如下：

資料流傳送資料的型態與方式：CBR/UDP。

平均資料流封包大小：500bytes。

優先權的設定：N0送出之資料流，其優先權為0；N1送出之資料流，其優先權為1；以此類推，N7送出之資料流，其優先權為7。

資料流的傳送順序：依序為N0.N1.N2.N3.N4.N5.N6.N7，然後

再重覆N0.N1.N2.N3.N4.N5.N6.N7，如此重覆10個循環。

模擬過程中所產生的資料流總數：80個。

每個資料流產生的間隔時間：1秒。

每個資料流傳送資料的時間：65秒。

總共觀察的時間：150秒。

每個資料流所保留頻寬大小：同一循環內各資料流所保留的頻寬相同，而各循環內所保留的頻寬大小在100Kbps ~ 1Mbps 之間隨機產生。

我們之所以模擬環境及參數如此設定，其目的為使每一個優先權等級送出的資料流流量均等。傳送經過具有DLBP機制的MPLS網域後，觀察每一個優先權等級的輸出量。圖60為使用DLBP演算法模擬得到的各不同優先權資料流的輸出量比較、接收端在時間為150秒時收到的總封包數比較及包括所有優先權的接收端在時間為150秒時收到的總封包數。我們發現優先權越高，其輸出量及接收端收到的總封包數也越高；而優先權較低者，其輸出量及接收端收到的總封包數亦較低；在輸出量方面，這種關係可在網路頻寬不足時，明顯的顯現出來。

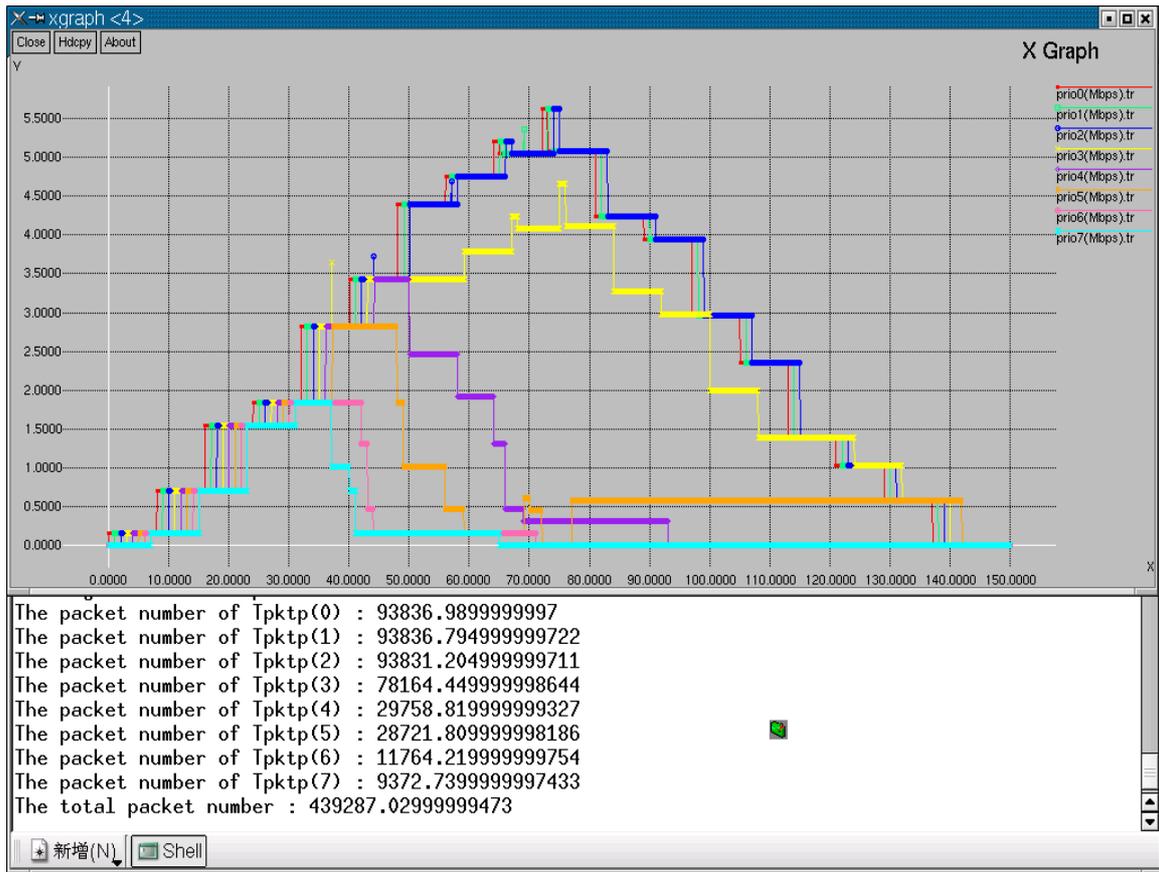


圖 60 使用DLBP演算法模擬得到的各不同優先權資料流的輸出量、接收端收到的總封包數比較及包括所有優先權的接收端收到的總封包數。

再來，我們仍利用圖 59 之網路拓撲來模擬 DLB 演算法。而其模擬環境參數設定乃沿用 DLBP 演算法。每個資料流所要求保留頻寬大小，亦使用模擬 DLBP 演算法時所隨機產生的一組頻寬；唯一不同的是優先權的設定：N0 至 N7 送出之資料流，其優先權為相同。圖 61 為使用 DLB 演算法模擬得到的資料流的總輸出量及接收端收到的總封包數；而圖 62 為使用 DLB 演算法模擬得到的網路資源利用率。圖 63 和圖 64 分別為使用 DLBP 演算法模擬得到的資料流的總輸出量和網路資源利用率。從模擬的結果，我們可以看出 DLBP 演算法的資料流的總輸出量和網路資源利用率稍優於 DLB 演算法；從接收端收到的總封包數來看，使用 DLBP 演算法時，接收端總共收到 439,287 個封包；而使用 DLB 演算法時，接收端總共收到 401,946 個封包，明顯 DLBP 演算法較佳。分析其原因為 DLBP 演算法的侵佔行為是，若優先權最低之 LSP 不只一條時，則選擇 UB 值最低者，當作被侵佔的對象，因此可增加網路資源利用率。而 DLB 演算法並沒有侵佔的觀念。綜合各種可能，長時間下來，平均而言，DLBP 演算法的資料流的總輸出量和網路資源利用率會稍

優於DLB演算法。

由這一個模擬結果，我們驗證了DLBP演算法，不只比DLB演算法多出優先權的功能，並且其資料流的總輸出量和網路資源利用率稍優於DLB演算法。

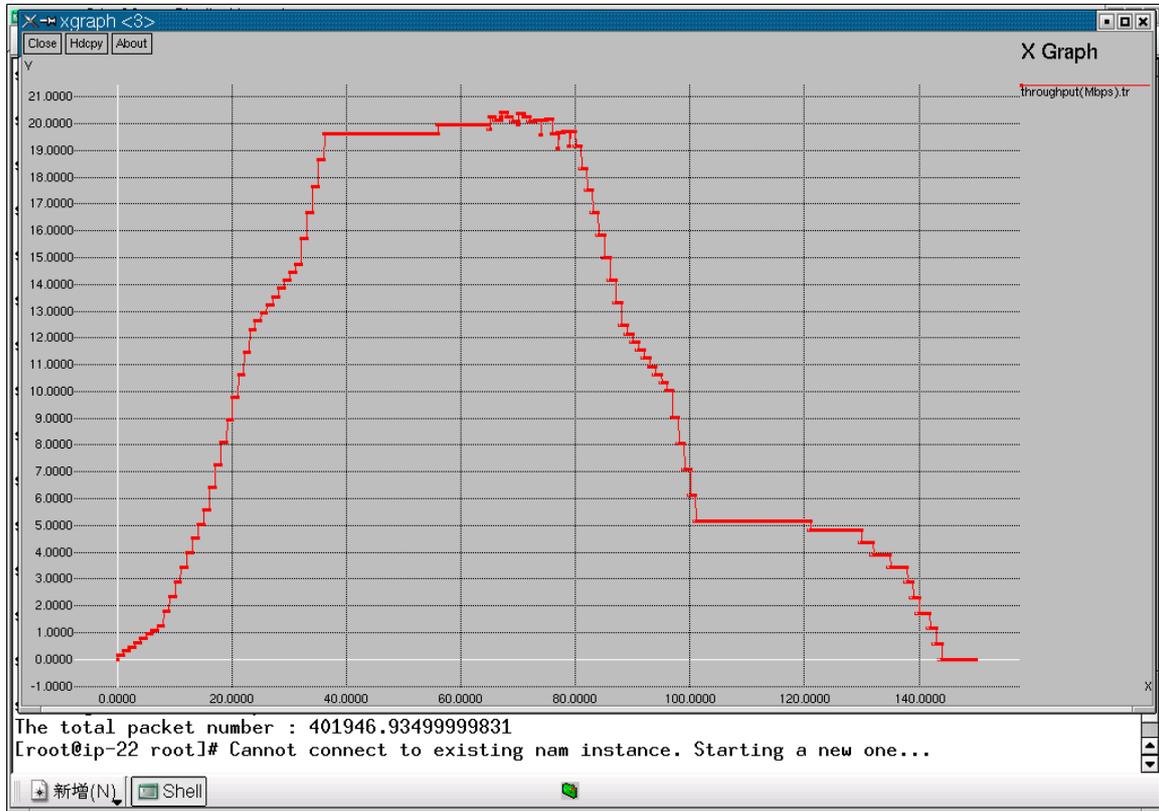


圖 61 使用 DLB 演算法模擬得到的資料流的總輸出量及接收端收到的總封包數

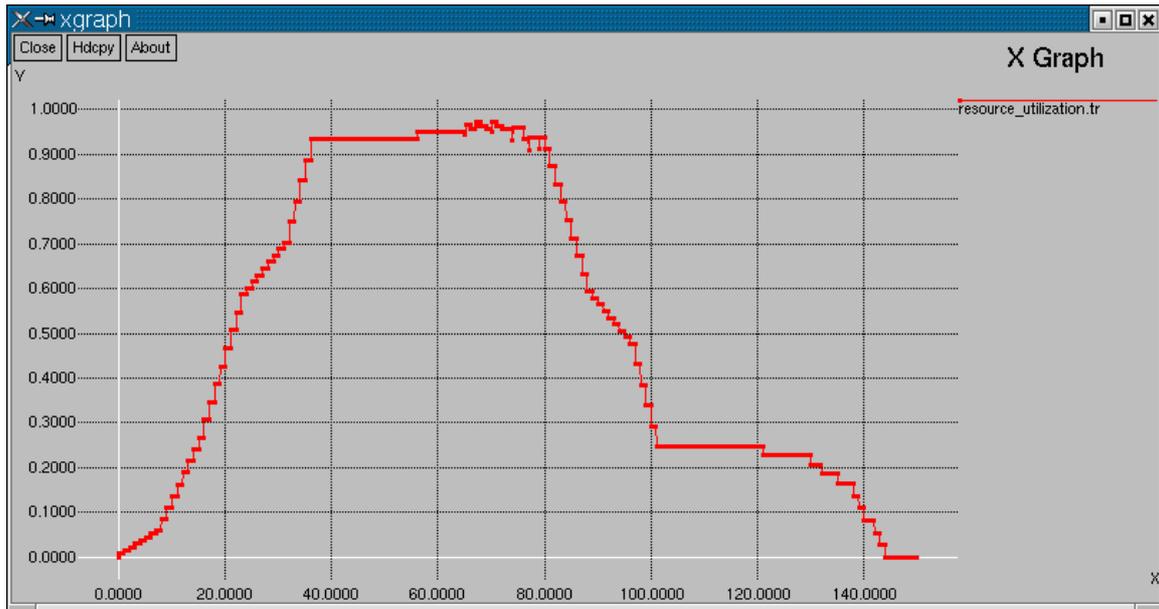


圖 62 使用 DLB 演算法模擬得到的網路資源利用率



圖 63 使用 DLBP 演算法模擬得到的資料流的總輸出量

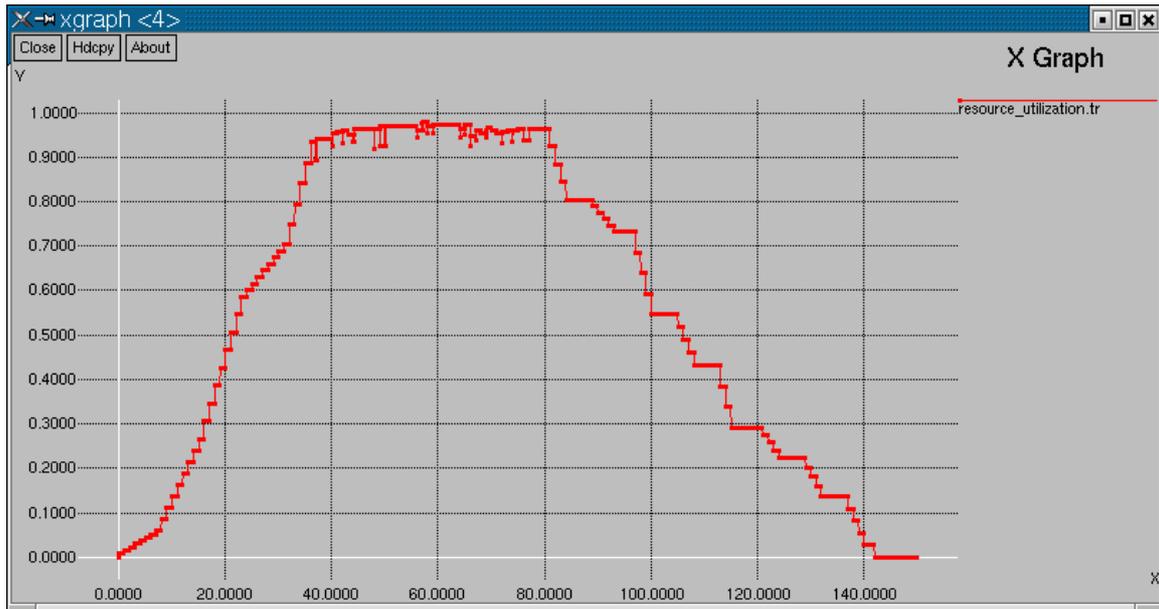


圖 64 使用 DLBP 演算法模擬得到的網路資源利用率

5.3 強迫式頻寬侵佔演算法

我們提過在 MPLS 使用 CR-LSP 建立符合資料流需求的 CR-LSP 時，具有以下功能：

1. 可指定路徑。
2. 具有頻寬保留的功能。
3. 當高優先權在建 CR-LSP 時，若發生頻寬不足時，可以去侵佔正在建或已存在之較低優先權之 CR-LSP 保留的頻寬。

但根據第三點，當高優先權在建 CR-LSP 時，若發生頻寬不足時，可以去侵佔正在建或已存在之較低優先權之 CR-LSP 保留的頻寬，則低優先權的資料流的資料傳輸連線在頻寬被侵佔之後，就不再有傳輸品質的保證，這種只依據優先權高低，侵佔頻寬的機制，對於低優先權資料連線的傳輸品質，造成不可預期的影響。為了解決這個問題，因此我們才提出 DLBP 演算法。此演算法可以達到，當頻寬不夠時，不因優先權低就放棄建 CR-LSP，也不因優先權高就隨意侵佔優先權低的資料流頻寬，若不得已需要侵佔優先權低的資料流頻寬，也要使網路資源使用及傳輸品質最佳

化。

爲了驗證 DLBP 演算法，確實可以達到以上目標，因此我們另外提出一個與 DLBP 演算法做比較的演算法稱爲強迫式頻寬侵佔演算法 (Force-Mode Bandwidth Preemption Algorithm, FMBP)。FMBP 演算法流程如下：

當 Ingress LSR 收到一個新的資料流時，

1. 我們使用 TSLB 演算法替此新資料流找尋一個最短路徑；如果可找到，則沿著此一路徑建立一條 CR-LSP，然後將此資料流導向此新建的 CR-LSP。
2. 如果找不到可用路徑，則在 Ingress LSR 和 Egress LSR 之間隨機選擇一條路徑，做爲侵佔的對象。若侵佔失敗，則傳送失敗，演算法結束。

此演算法的缺點爲，當新資料流找不到路徑可建 CR-LSP 時，便直接在 Ingress LSR 和 Egress LSR 之間隨機選擇一條路徑，做爲侵佔的對象。而和我們提出的此一周詳設計的 DLBP 演算法，當然不能相比的。我們要同時來模擬 DLBP 演算法和 FMBP 演算法，以印證 DLBP 演算法確實有很好的行爲表現，而可以解決像 FMBP 等產生的問題。

5.4 DLBP 演算法與 FMBP 演算法之模擬與模擬結果比較之分析

我們將 FMBP 演算法的模擬環境參數設定成與 DLBP 演算法的模擬環境參數設定完全一樣，以便於將二者的模擬結果做比較。我們將此模擬環境參數設定重新列出如下：

資料流傳送資料的型態與方式：CBR/UDP。

平均資料流封包大小：500bytes。

優先權的設定：N0送出之資料流，其優先權爲0；N1送出之資料流，其優先權爲1；以此類推，N7送出之資料流，其優先權爲7。

資料流的傳送順序：依序爲N0.N1.N2.N3.N4.N5.N6.N7，然後再重覆N0.N1.N2.N3.N4.N5.N6.N7，如此重覆10個循環。

模擬過程中所產生的資料流總數：80個。

每個資料流產生的間隔時間：1秒。

每個資料流傳送資料的時間：65秒。

總共觀察的時間：150秒。

每個資料流所保留頻寬大小：同一循環內各資料流所保留的頻寬相同，而各循環內所保留的頻寬大小在100Kbps~1Mbps之間隨機產生。(DLB、DLBP、FMBP三者使用相同的隨機值，以利於三者性能表現的比較)

圖65為使用FMBP演算法模擬得到的各不同優先權資料流的輸出量、接收端收到的總封包數比較及包括所有優先權的接收端收到的總封包數。我們將此結果與圖60作比較，我們會發現FMBP演算法會有一個不合理的現象：那就是有的優先權較低的，反而得到較高的輸出量；而有的優先權較高的，卻得到較低的輸出量。之所以會如此，乃是因為當新資料流找不到路徑可建CR-LSP時，便直接在Ingress LSR 和 Egress LSR之間隨機選擇一條路徑，做為侵佔的對象。而此隨機選擇的路徑，未必是優先權最低者的路徑，換句話說，也就是被侵佔者，其優先權未必是最低的，所以會造成此不合理現象。而我們會發現DLBP演算法，並不會有此一個不合理的現象，追究其原因為，因為DLBP演算法在選擇侵佔的對象時，其選擇方式是在MPLS網路裡尋找（1）其Ingress LSR和Egress LSR與新資料流相同。（2）其優先權是所有現存CR-LSP中最低的，除非此CR-LSP被移除是對新資料流建CR-LSP是沒有助益的。因此像FMBP此不合理現象，在DLBP演算法是不易發生的。

再來，我們從圖66使用FMBP演算法模擬得到的網路資源利用率和從圖64使用DLBP演算法模擬得到的網路資源利用率相比，明顯可以看出DLBP演算法的網路資源利用率優於FMBP演算法；從圖60和圖65接收端收到的總封包數來看，使用DLBP演算法時，接收端總共收到439,287個封包；而使用FMBP演算法時，接收端總共收到382,443個封包，明顯DLBP演算法較佳。分析其原因為DLBP演算法是當利用TSLB演算法無法替新資料流找到路徑時，則利用DLB演算法試圖去找到另一條路徑，若不得已需要侵佔優先權低的資料流頻寬，也要使網路資源使用及傳輸品質最佳化；而不像FMBP演算法，直接就去侵佔優先權較低的傳輸連線所保留的頻寬，如果

它指定的路徑上的傳輸連線的優先權都比它高，則立刻放棄建立傳輸連線，因此DLBP演算法可比FMBP演算法得到較高的網路資源利用率是必然的結果。

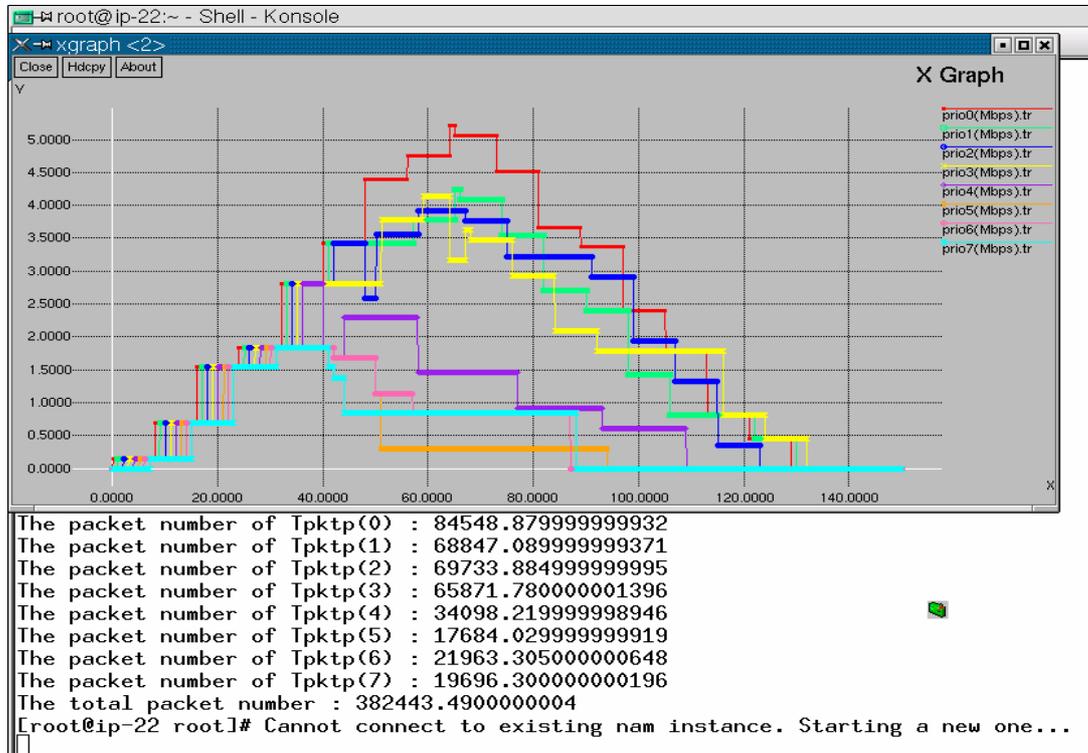


圖 65 使用FMBP演算法模擬得到的各不同優先權資料流的輸出量、接收端收到的總封包數比較及包括所有優先權的接收端收到的總封包數。

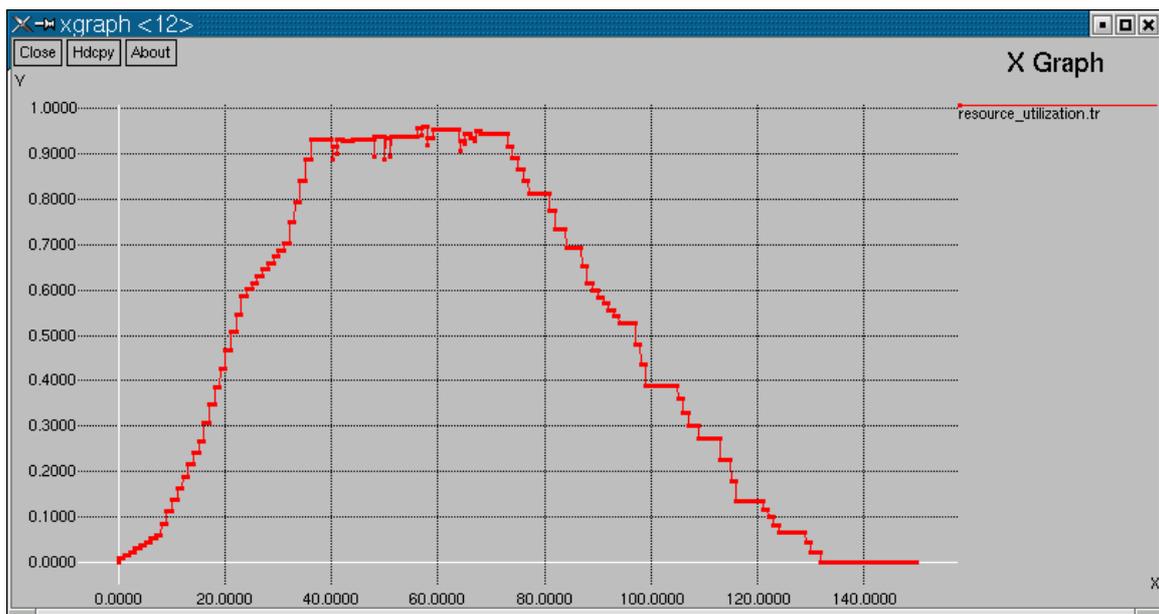


圖 66 使用 FMBP 演算法模擬得到的網路資源利用率