第四章 具有優先權的動態負載平衡演算法

DLBP 演算法乃是保留了 DLB 演算法之高網路資源利用率之優點，並能應用在資料流具有優先權的情況下，根據資料流的優先權，給予合理、公平的資源分配。本章將對我們所提出的 DLBP 演算法，利用流程圖詳加描述，並利用一些實際的例子來做模擬，這些實例包括了所有在 DLBP 演算法的各種情況，以期對 DLBP 演算法能更深入的了解。

4.1 DLBP 演算法的描述

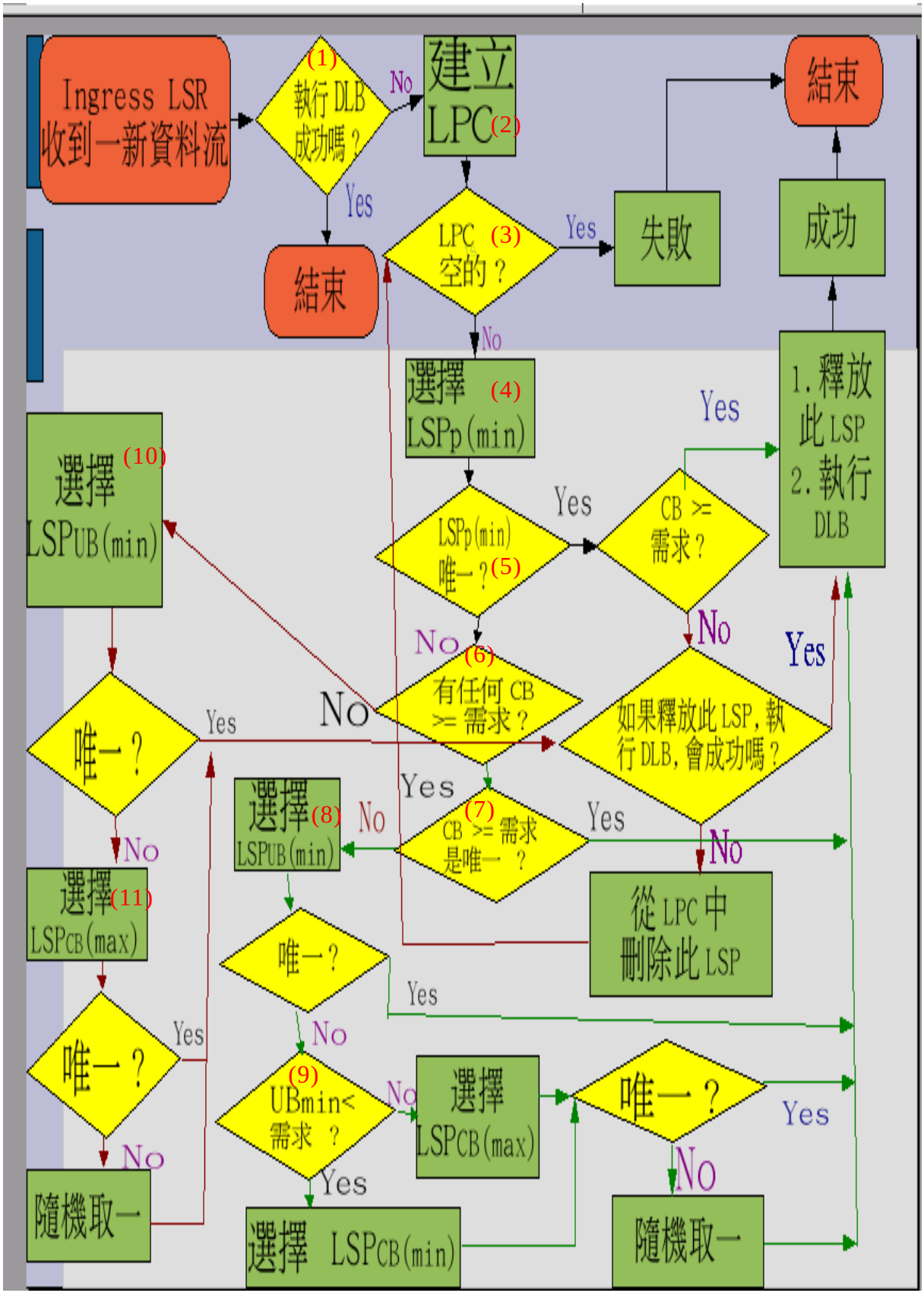

圖 28 DLBP 演算法之流程圖

圖 28 爲 DLBP 演算法之流程圖，我們對照流程圖解說如下：

（1） 當 Ingress LSR 收到一個新的資料流，首先使用 DLB 演算法爲新資料流選擇一個路徑，如果成功，此演算法成功地完成；如果失敗，則建立一個 LPC 集合。

（2） 建立 LPC：在 MPLS 網路裡收集所有 LSPid，其 Ingress LSR 和 Egress LSR 與新資料流相同，且其 holding-priority 小於需求者之 setup-priority 者。我們建構一個 LPC(Low-Priority-Collection)集合，此集合由這些收集到的 LSPid 所組成。

（3） 如果集合 LPC 是空的，表示此 MPLS 網路容量已無法滿足新資料流的頻寬需求，傳送失敗，演算法結束。

（4） 若 LPC 不是空的，則從集合 LPC 的元素 LSPid 中，找尋優先權最低者。

（5） 判斷優先權最低的 LSPid 是否只有一個，如果只有一個則：

- 若此 CR-LSP 之 CB >= 需求的頻寬，則釋放此被選到的 CR-LSP，並利用 DLB 演算法替新資料流建立 CR-LSP，此演算法成功地完成。因爲被釋放的此一 CR-LSP 之 CB >= 需求的頻寬，所以利用 DLB 演算法一定可以替新資料流建立 CR-LSP。（請參考實例（一））
- 如果此 CR-LSP 之 CB 值小於需求的頻寬，則我們必須進一步判斷若將被選到的 CR-LSP 釋放，對新的資料流是否可利用 DLB 演算法找到足夠的頻寬來建 CR-LSP？
  - 如果找得到，則釋放此被選到的 CR-LSP，並利用 DLB 演算法替新資料流建立 CR-LSP，此演算法成功地完成。（請參考實例（二））
  - 如果找不到，則在 LPC 集合中刪除掉此被選到的 LSPid。表示此 CR-LSP 即使被釋放，對新資料流亦無幫助，因此不予考慮。（請參考實例（三））

（6） 這些優先權最低之 LSPid 的 CB 有大於或等於需求的頻寬的嗎？如果 CB 值有大於或等於需求的，則優先考慮。因爲被選中者，可直接釋放掉，便可給新資料流使用，而無重新路由的問題產生。

（7） 如果 CB 值大於或等於需求的頻寬的現存 CR-LSP 只有一個，則釋放此被選到的 CR-LSP，並利用 DLB 演算法替新資料流建立 CR-LSP，此演算法成功地完成。（請參考實例（三））

（8） 從 CB >= 需求的頻寬之 LSPid 中，選擇 LSPid 之 UB 值最低者。若此 UB 值最低者，只有一個，則釋放此被選到的 CR-LSP，並利用 DLB 演算法替新資料流建立 CR-LSP，此演算法成功地完成。之所以要選擇 CR-LSP 中 UB 最低者，因爲可增加整個網路的資源利用率。（請參考實例（四））

（9）

— 如果 LSPid 之 UB 最低者 < 需求的頻寬 ，則從這些 UB 最低者中，選擇 CB 值最小者。CB 值最小者，若不是唯一，則隨機取一。釋放此被選到的 CR-LSP，並利用 DLB 演算法替新資料流建立 CR-LSP，此演算法成功地完成。（請參考實例（六））

— 如果 LSPid 之 UB 最低者 >= 需求的頻寬，則從這些 UB 最低者中，選擇 CB 值最大者。CB 值最大者，若不是唯一，則隨機取一。釋放此被選到的 CR-LSP，並利用 DLB 演算法替新資料流建立 CR-LSP，此演算法成功地完成。（請參考實例（五））

— 目的：根據分析如此可使可用頻寬較集中於某一路徑，對未來要建之 CR-LSP 有幫助。

（10）這些優先權最低之 LSPid 的 CB 值皆小於需求的頻寬，則從這些優先權最低之 LSPid 中尋找 UB 值爲最低者。若 UB 最低者只有一個，則選擇此 LSPid，然後我們判斷若將被選到的 CR-LSP 釋放，對新的資料流是否可利用 DLB 演算法找到足夠的頻寬來建 CR-LSP？

— 如果找得到，則釋放此被選到的 CR-LSP，並利用 DLB 演算法替新資料流建立 CR-LSP，此演算法成功地完成。（請參考實例（七））

— 如果找不到，則在 LPC 集合中刪除掉此被選到的 LSPid。表示此 CR-LSP 即使被釋放，對新資料流亦無幫助，因此不予考慮。（請參考實例（九））

➢ 說明：此處選擇 LSPid 之 UB 最低者，原因亦是可增加資源利用率。

（11）LSPid 之 UB 最低者不只一個時，則從 LSPid 之 UB 最低者中選取 CB 值最大者。若 CB 值最大者不是唯一，則隨機取一。然後我們判斷若將被選到的 CR-LSP 釋放，對新的資料流是否可利用 DLB 演算法找到足夠的頻寬來建 CR-LSP？

— 如果找得到，則釋放此被選到的 CR-LSP，並利用 DLB 演算法替新資料流建立 CR-LSP，此演算法成功地完成。（請參考實例（八））

— 如果找不到，則在 LPC 集合中刪除掉此被選到的 LSPid。表示此 CR-LSP 即使被釋放，對新資料流亦無幫助，因此不予考慮。(請參考實例（十))

- 說明：此處選擇 LSPid 之 CB 值最大者，原因是有可能降低重新路由的次數。

4.2 DLBP 演算法之模擬和分析

爲了更進一步解說 DLBP 演算法，以及測試此具有 DLBP 演算法功能的 MNS 模擬器，是否能遵照我們上節所描述的 DLBP 演算法，我們列舉了十個實例來做模擬。這些實例包括了所有在 DLBP 演算法的各種情況，足以印證具有 DLBP 演算法的 MNS 模擬器的正確性。

我們將這些實例的模擬環境參數設定如下：

資料流傳送資料的型態與方式：Poisson process/UDP。

資料流的傳送順序：依各實例之圖片上所標示。

爲傳送各資料流所建CR-LSP之LSPid：

- N0所送出來的資料流，其所建之CR-LSP之LSPid爲1100
- N1所送出來的資料流，其所建之CR-LSP之LSPid爲1200
- N2送出二個具有不同頻寬需求及優先權的資料流，其所建之CR-LSP之LSPid分別爲1300及1400

模擬過程中所產生的資料流總數：4個。

每個資料流產生的間隔時間：0.5秒。

每個資料流傳送資料的時間：3秒。

總共觀察的時間：3秒。

每個資料流所需求頻寬大小及本身具有的優先權：依各實例之圖片上所標示。

- 優先權標示之說明：如 7_4，7爲setup priority；4爲holding priority。

網路上各鏈接頻寬：依各實例之圖片上所標示。

LSPid=1101,1201,1301,1401分別是爲了重新路由LSPid=1100,1200,1300,1400上的資料流，所新建立的CR-LSP。等資料流重新導向新建的CR-LSP後，便可將原CR-LSP釋放。

在我們完成了此十個實例的模擬，並對此十個實例加以分析之後，我們不僅印證此具有DLBP演算法功能的MNS模擬器，完全遵照我們上節所描述的DLBP演

算法，並且讓我們對DLBP演算法更深入的了解。我們將此十個實例的模擬結果和分析附於本章最後，以供參考。

實例（一）：

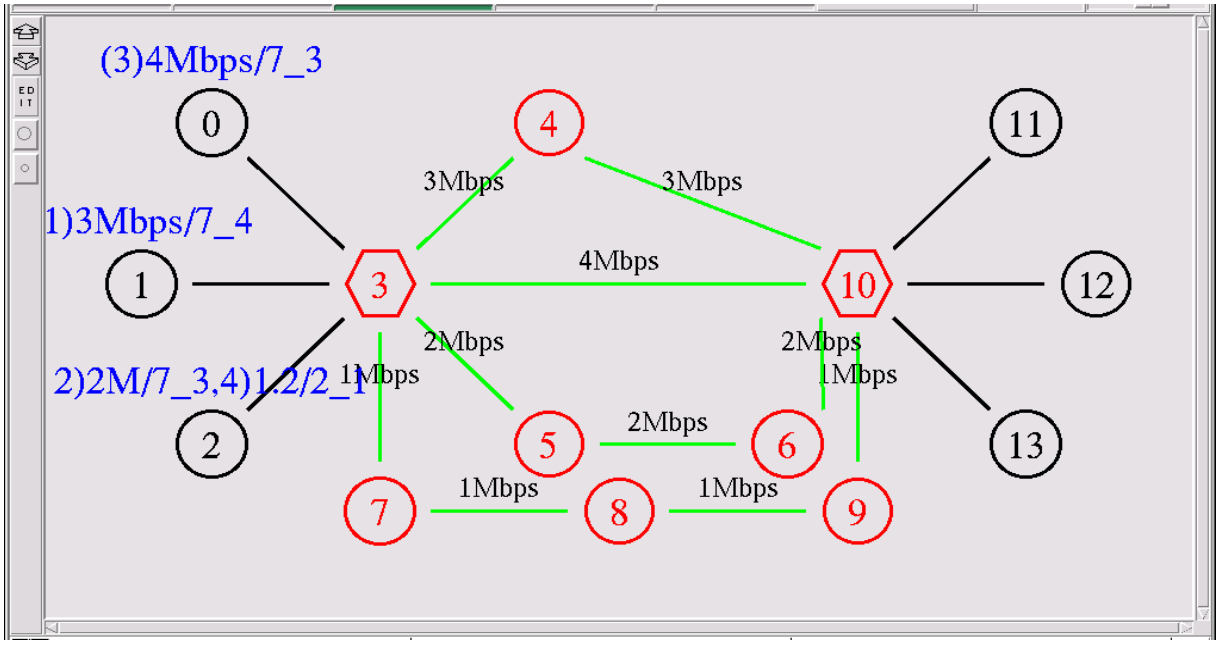

圖 29 實例（一）用來模擬 DLBP 演算法之網路拓撲、各資料流的傳送順序、各鏈接的頻寬及各資料流的需求頻寬

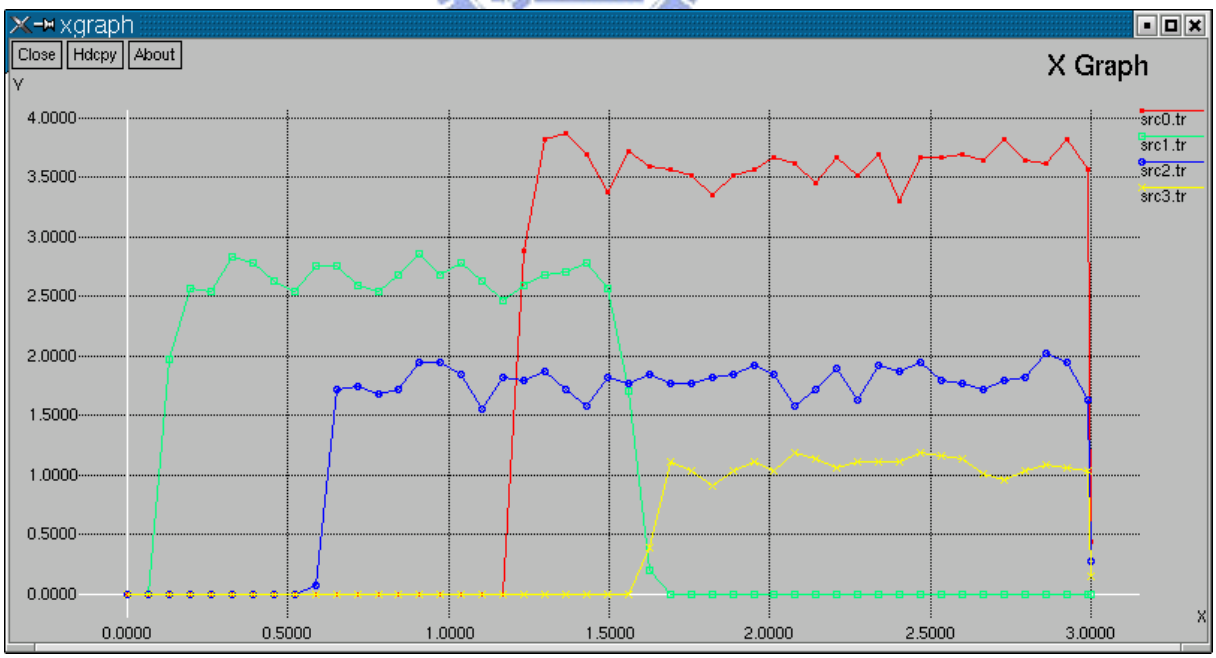

圖 30 實例（一）使用 DLBP 演算法模擬圖 29 得到之每一個資料流的輸出量

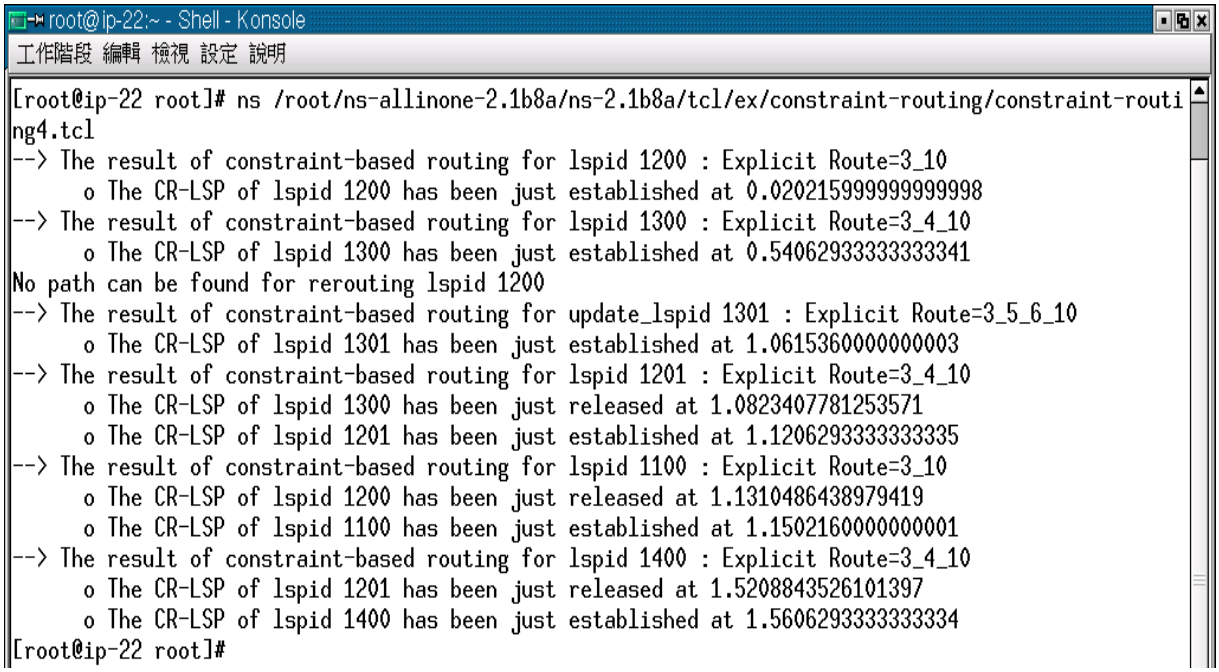

圖 31 實例（一）使用 DLBP 演算法模擬圖 29 得到之每一條 CR-LSP 之路徑、建立時間和被釋放時間

分析：

1. 頻寬需求爲（1）3Mbps 及（2）2Mbps 之資料流，順利根據 TSLB 演算法找到路徑建立 CR-LSP。
2. 當頻寬需求爲（3）4Mbps 之資料流送出時，根據 DLB 演算法，頻寬需求爲 2Mbps 的資料流被重新路由至 3_5_6_10 路徑，而頻寬需求爲 3Mbps 的資料流被重新路由至 3_4_10 的路徑，因而騰出較高容量的 3_10 路徑給頻寬需求爲 4Mbps 的新資料流使用。
3. 當 Ingress LSR LSR3 收到頻寬需求爲（4）1.2Mbps 資料流時，因執行 DLB 演算法無法替此資料流找到足夠頻寬的路徑使用，因此從現已存在的 CR-LSP 中，尋找是否有可侵佔的對象。在這些現已存在的 CR-LSP 中，選擇釋放持有優先權最低 LSPid=1201 之 CR-LSP，其持有優先權等於 4，小於新資料流的設定優先權 2。在釋放 LSPid=1201 之 CR-LSP 後，其 CB 值大於新資料流的頻寬需求，因此新資料流便直接在此路徑 3_4_10 建立 CR-LSP，其 LSPid=1400。

實例（二）

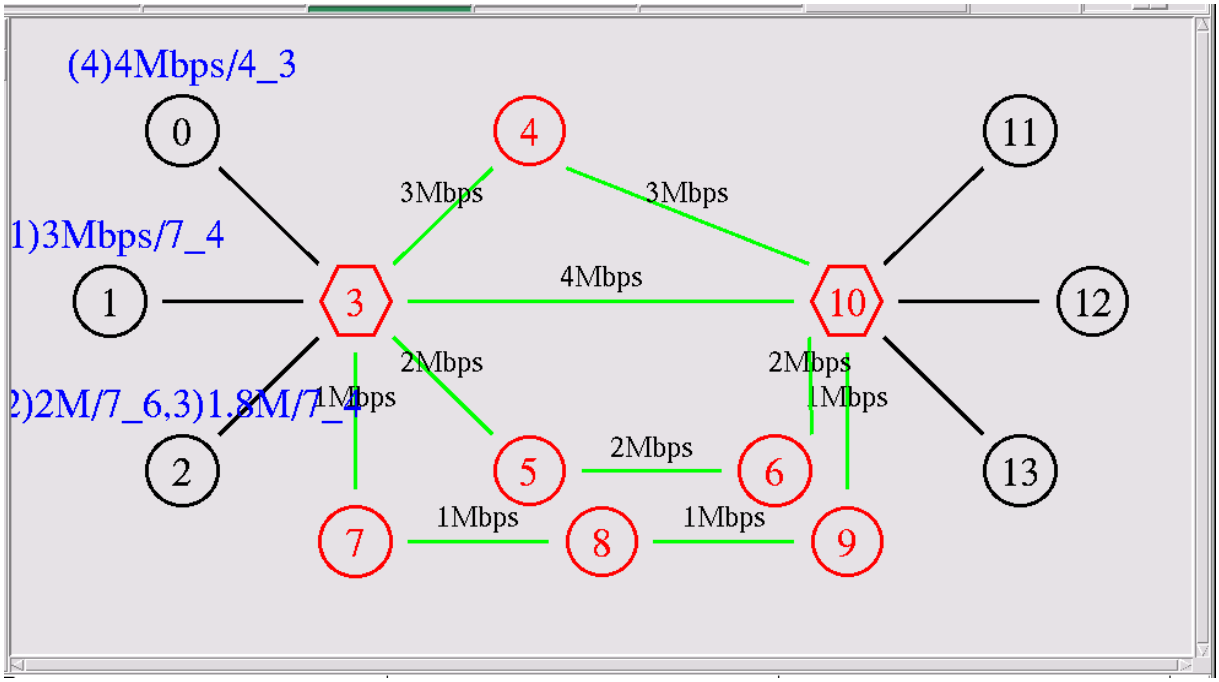

圖 32 實例（二）用來模擬 DLBP 演算法之網路拓撲、各資料流的傳送順序、各鏈接的頻寬及各資料流的需求頻寬

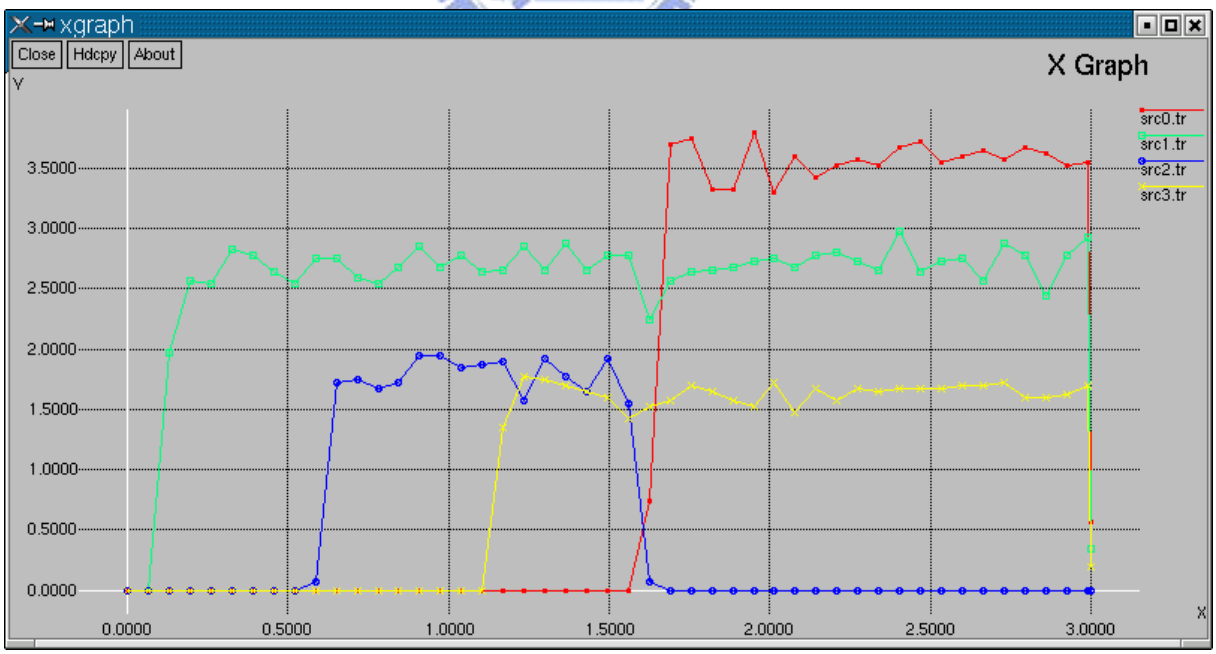

圖 33 實例（二）使用 DLBP 演算法模擬圖 32 得到之每一個資料流的輸出量

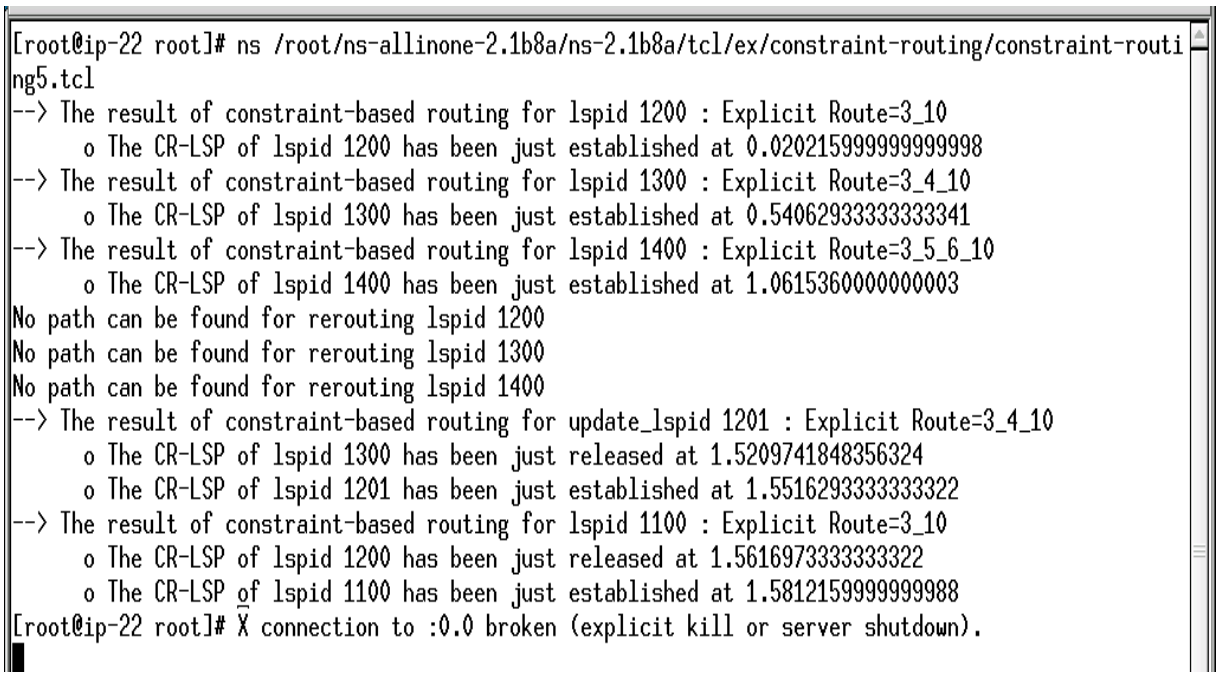

圖 34 實例（二）使用 DLBP 演算法模擬圖 32 得到之每一條 CR-LSP 之路徑、建立時間和被釋放時間

分析：

1. 頻寬需求爲（1）3Mbps 及（2）2Mbps 及（3）1.8Mbps 之資料流，都順利根據 TSLB 演算法找到路徑建立 CR-LSP。
2. 當 Ingress LSR LSR3 收到頻寬需求爲（4）4Mbps 資料流時，因執行 DLB 演算法無法替此資料流找到足夠頻寬的路徑使用，因此從現已存在的 CR-LSP 中，尋找是否有可侵佔的對象。在這些現已存在的 CR-LSP 中，選擇持有優先權最低 LSPid=1300 之 CR-LSP，其持有優先權等於 6，小於新資料流的設定優先權 4。但因 LSPid=1300 之 CB 值小於新資料流的頻寬需求，則我們必須進一步判斷若將被選到的 CR-LSP 釋放，對新的資料流是否可利用 DLB 演算法找到足夠的頻寬來建 CR-LSP？判斷的結果爲可找得到足夠的頻寬，因此釋放此被選到的 LSPid=1300 之 CR-LSP，並利用 DLB 演算法，將頻寬需求爲 3Mbps 的資料流重新路由至 3_4_10 的路徑，因而騰出較高容量的 3_10 路徑給頻寬需求爲 4Mbps 的新資料流使用，其所建的 CR-LSP 之 LSPid=1100。

實例（三）

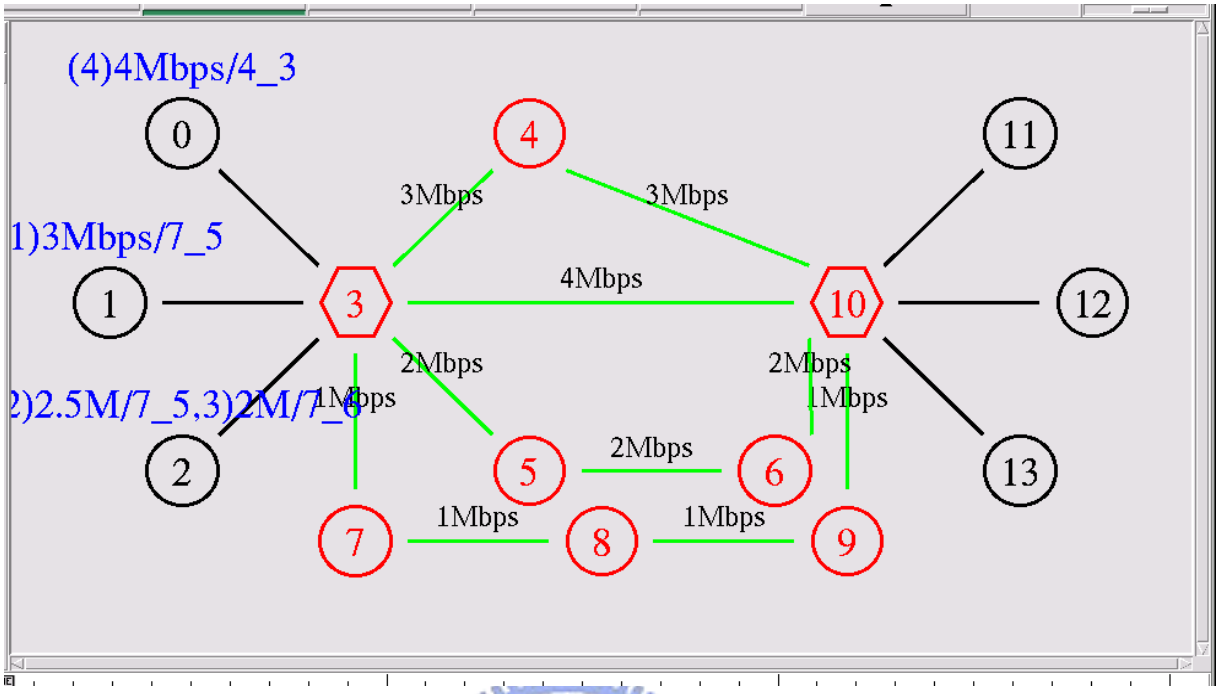

圖 35 實例（三）用來模擬 DLBP 演算法之網路拓撲、各資料流的傳送順序、各鏈接的頻寬及各資料流的需求頻寬

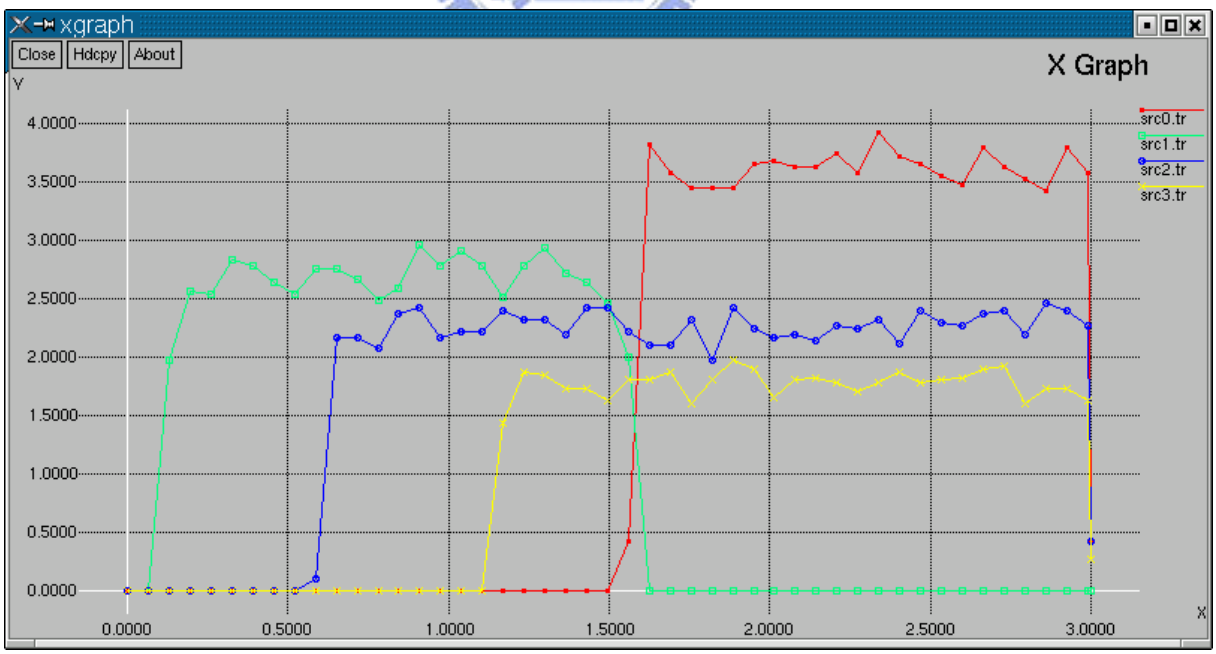

圖 36 實例（三）使用 DLBP 演算法模擬圖 35 得到之每一個資料流的輸出量

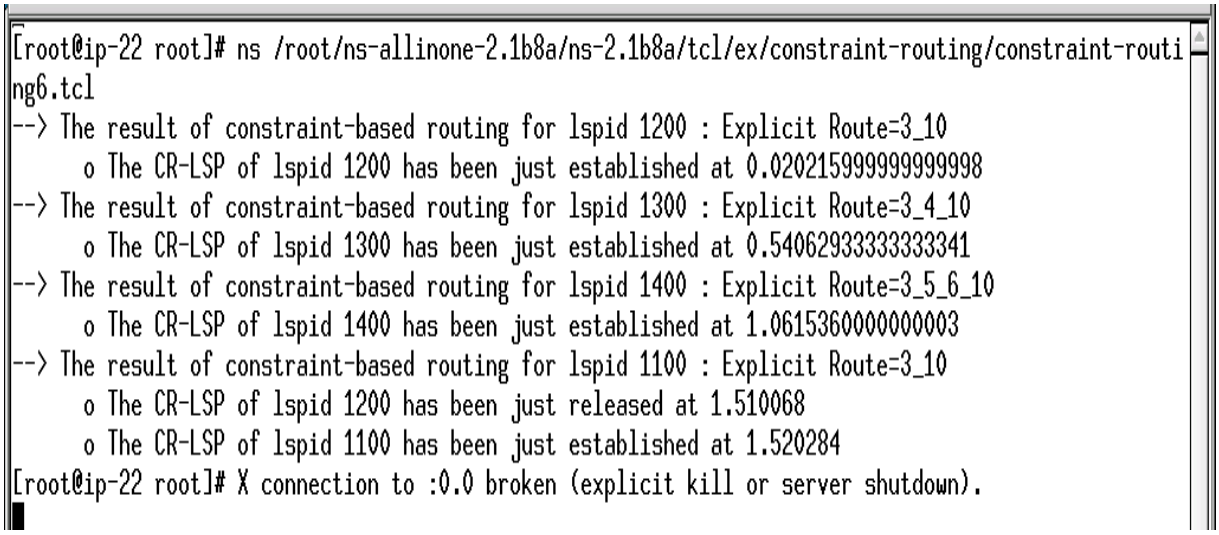

圖 37 實例（三）使用 DLBP 演算法模擬圖 35 得到之每一條 CR-LSP 之路徑、建立時間和被釋放時間

分析：

1. 頻寬需求爲（1）3Mbps 及（2）2.5Mbps 及（3）2Mbps 之資料流，都順利根據 TSLB 演算法找到路徑建立 CR-LSP。
2. 當 Ingress LSR LSR3 收到頻寬需求爲（4）4Mbps 資料流時，因執行 DLB 演算法無法替此資料流找到足夠頻寬的路徑使用，因此從現已存在的 CR-LSP 中，尋找是否有可侵佔的對象。
   - 在這些現已存在的 CR-LSP 中，選擇持有優先權最低 LSPid=1400 之 CR-LSP，其持有優先權等於 6，小於新資料流的設定優先權 4。但因 LSPid=1400 之 CB 值小於新資料流的頻寬需求，則我們必須進一步判斷若將被選到的 CR-LSP 釋放，對新的資料流是否可利用 DLB 演算法找到足夠的頻寬來建 CR-LSP？判斷的結果爲找不到。表示此 CR-LSP 即使被釋放，對新資料流亦無幫助，因此不予考慮 LSPid=1400 之 CR-LSP。
   - 除了 LSPid=1400 之 CR-LSP 之外，持有優先權最低之 CR-LSP 有二個，分別爲 LSPid=1200 及 1300。其持有優先權等於 5，小於新資料流的設定優先權 4，但因爲 LSPid=1200 之 CB 值等於新資料流的頻寬需求，因此選擇侵佔 LSPid=1200 之 CR-LSP。在釋放 LSPid=1200 之 CR-LSP 之後，新資料流便利用此路徑 3_10，建立 CR-LSP，其 LSPid=1100。

實例（四）

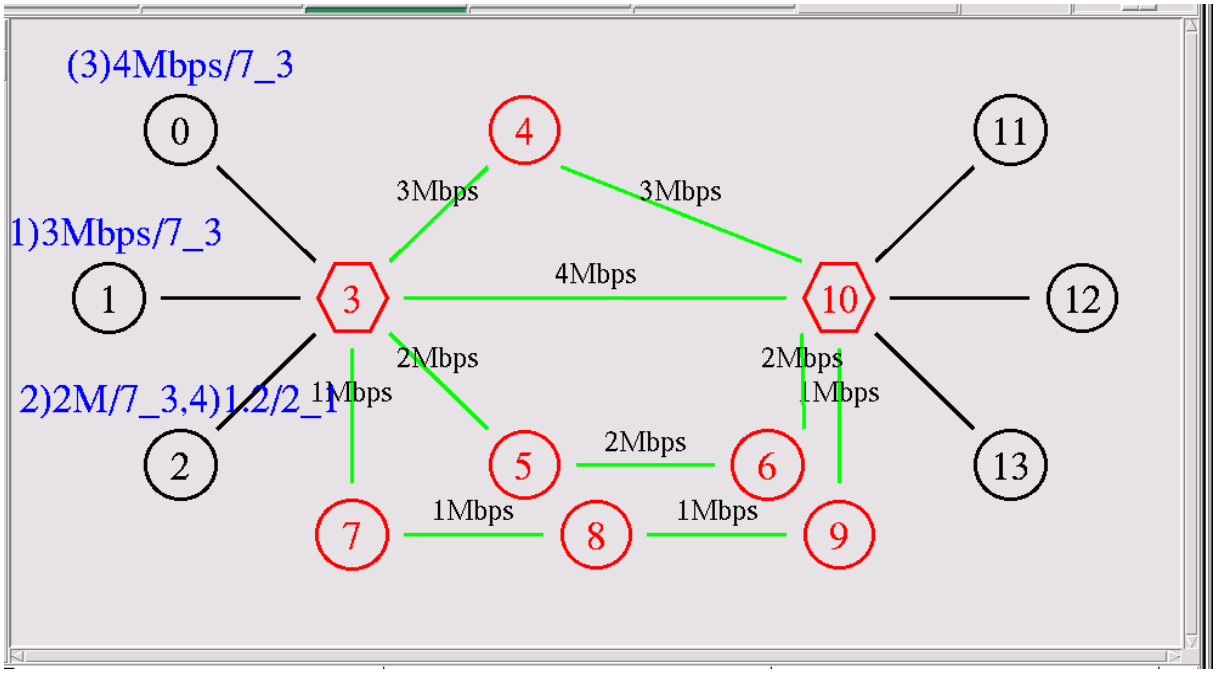

圖 38 實例（四）用來模擬 DLBP 演算法之網路拓撲、各資料流的傳送順序、各鏈接的頻寬及各資料流的需求頻寬

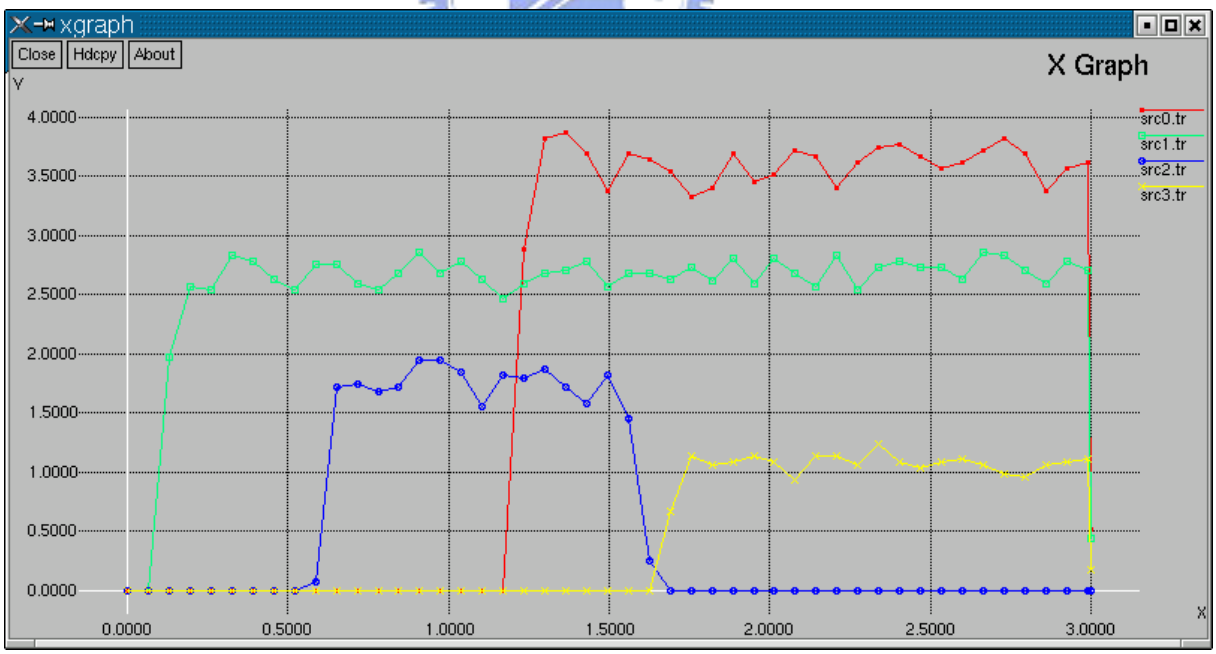

圖 39 實例（四）使用 DLBP 演算法模擬圖 38 得到之每一個資料流的輸出量

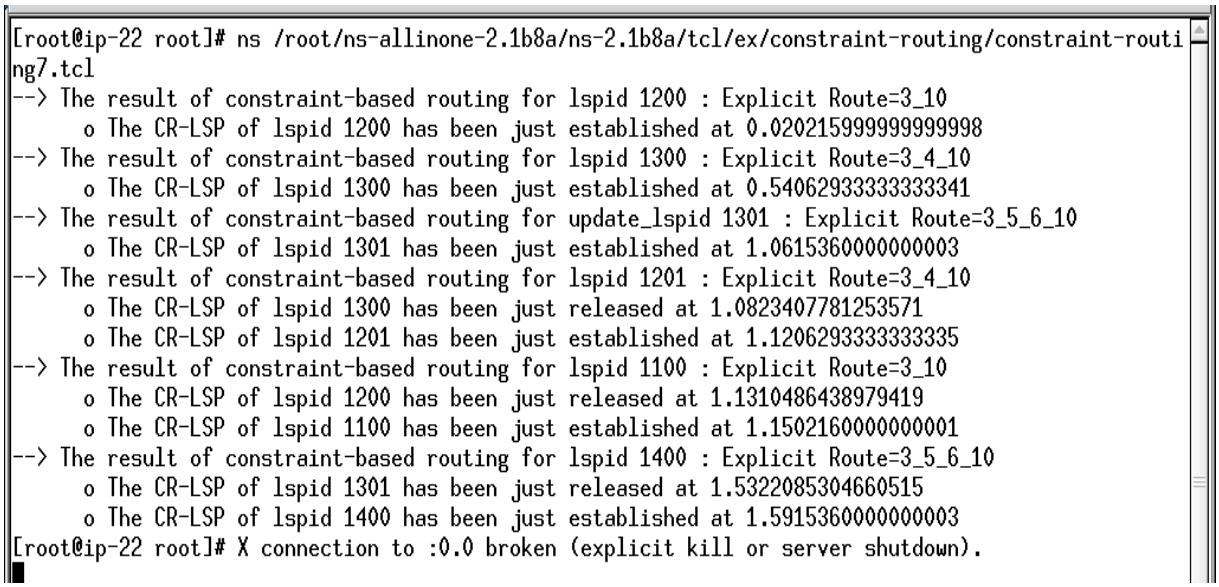

圖 40 實例（四）使用 DLBP 演算法模擬圖 38 得到之每一條 CR-LSP 之路徑、建立時間和被釋放時間

分析：

1. 頻寬需求爲（1）3Mbps 及（2）2Mbps 之資料流，順利根據 TSLB 演算法找到路徑建立 CR-LSP。
2. 當頻寬需求爲（3）4Mbps 之資料流送出時，根據 DLB 演算法，頻寬需求爲 2Mbps 的資料流被重新路由至 3_5_6_10 路徑，而頻寬需求爲 3Mbps 的資料流被重新路由至 3_4_10 的路徑，因而騰出較高容量的 3_10 路徑給頻寬需求爲 4Mbps 的新資料流使用。
3. 當 Ingress LSR LSR3 收到頻寬需求爲（4）1.2Mbps 資料流時，因執行 DLB 演算法無法替此資料流找到足夠頻寬的路徑使用，因此從現已存在的 CR-LSP 中，尋找是否有可侵佔的對象。
   - 在這些現已存在的 CR-LSP 中，持有優先權最低之 CR-LSP 有三個，分別爲 LSPid=1100、1201 及 1301，其持有優先權等於 3，小於新資料流的設定優先權 2。而這些 LSPid 之 CB 值皆大於新資料流的頻寬需求，因此我們選擇侵佔 UB 值最低之 LSPid，其 LSPid=1301。
   - 在釋放 LSPid=1301 之 CR-LSP 之後， 新資料流便直接在此路徑 3_5_6_10 建立 CR-LSP，其 LSPid=1400。
   - 由此實例可看出，選擇侵佔 UB 值最低之 LSPid，可使網路資源利用率最高。

實例（五）

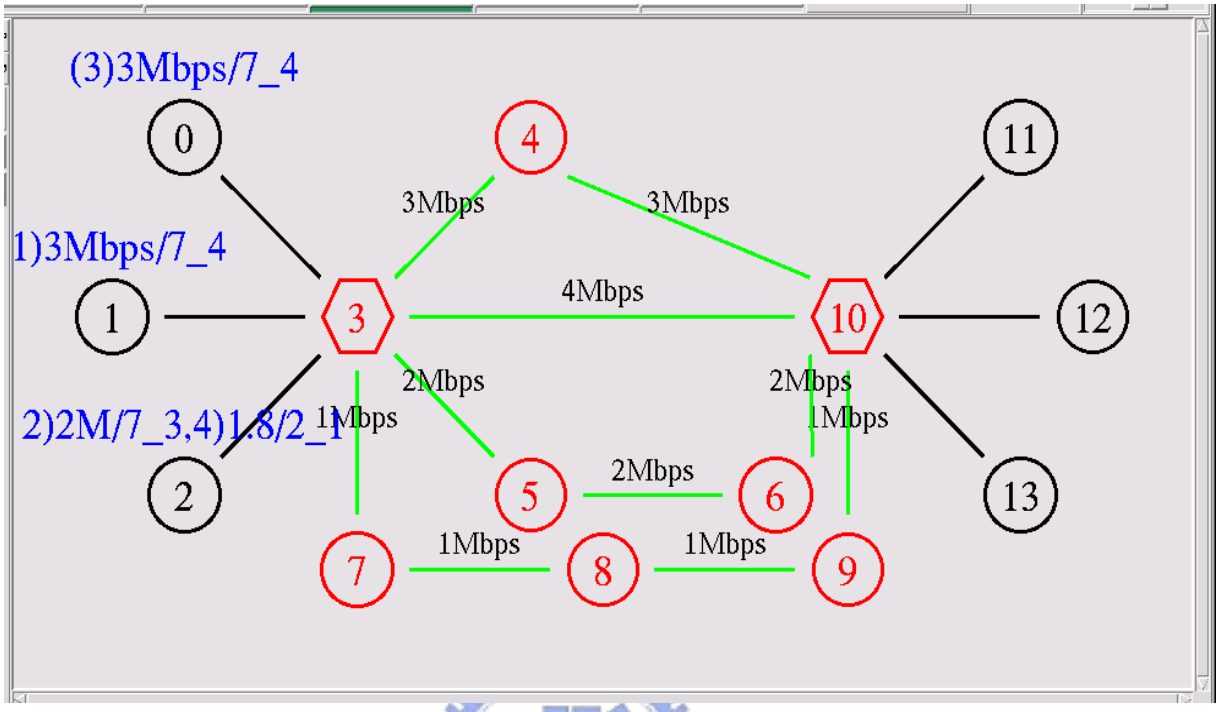

圖 41 實例（五）用來模擬 DLBP 演算法之網路拓撲、各資料流的傳送順序、各鏈接的頻寬及各資料流的需求頻寬

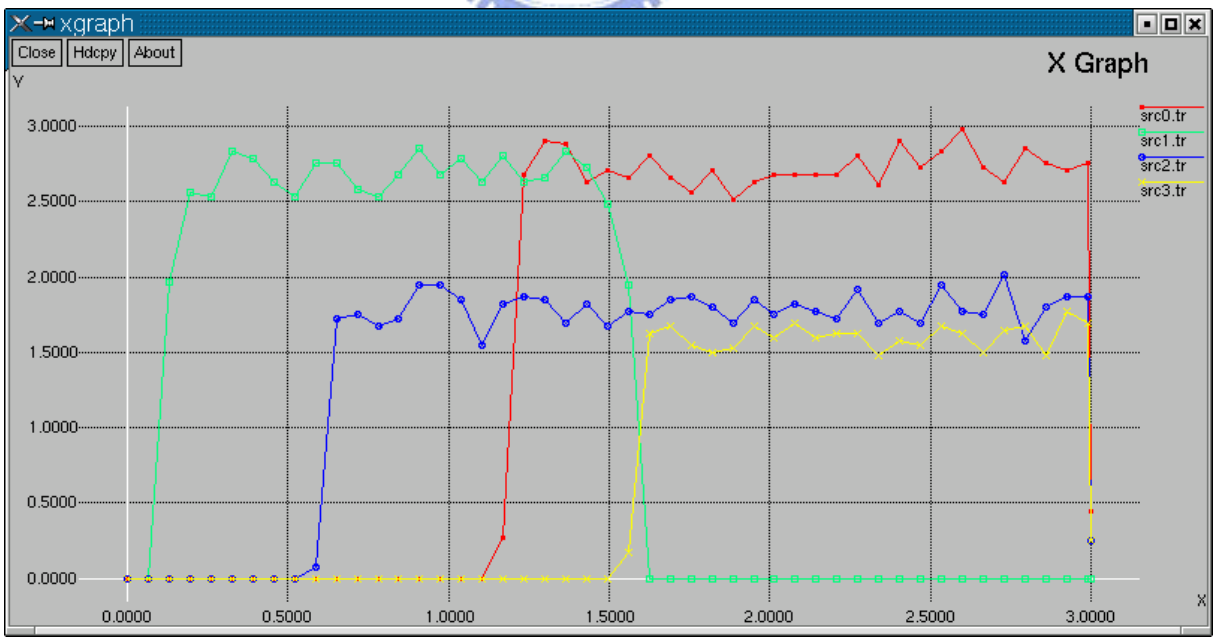

圖 42 實例（五）使用 DLBP 演算法模擬圖 41 得到之每一個資料流的輸出量

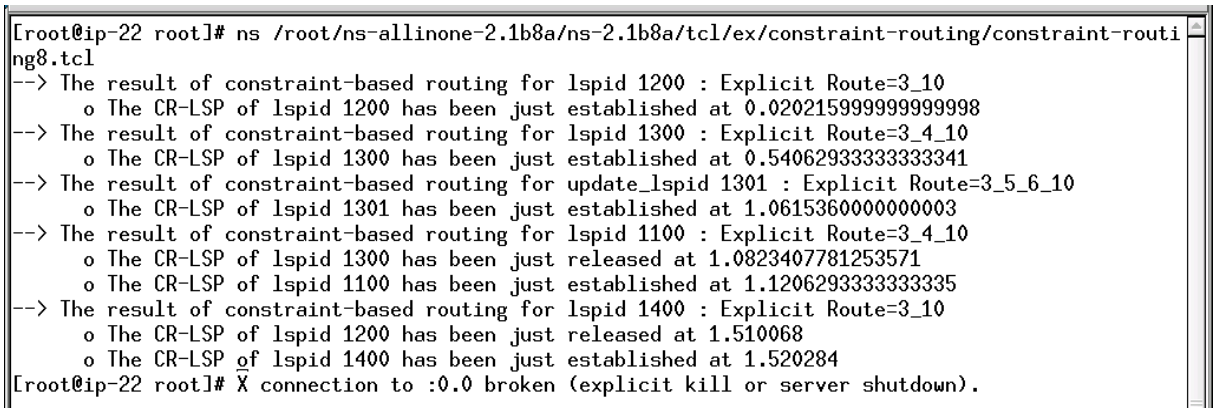

圖 43 實例（五）使用 DLBP 演算法模擬圖 41 得到之每一條 CR-LSP 之路徑、建立時間和被釋放時間

分析：

1. 頻寬需求爲（1）3Mbps 及（2）2Mbps 之資料流，順利根據 TSLB 演算法找到路徑建立 CR-LSP，其 LSPid 分別爲 1200 及 1300。
2. 當頻寬需求爲（3）3Mbps 之資料流送出時，根據 DLB 演算法，頻寬需求爲 2Mbps 的資料流被重新路由至 3_5_6_10 路徑，其 LSPid 爲 1301，因而騰出較高容量的 3_4_10 路徑給頻寬需求爲 3Mbps 的新資料流使用，其所建之 CR-LSP 之 LSPid 爲 1100。
3. 當 Ingress LSR LSR3 收到頻寬需求爲（4）1.8Mbps 資料流時，因執行 DLB 演算法無法替此資料流找到足夠頻寬的路徑使用，因此從現已存在的 CR-LSP 中，尋找是否有可侵佔的對象。
   - 在這些現已存在的 CR-LSP 中，持有優先權最低之 CR-LSP 有二個，分別爲 LSPid=1100 及 1200，其持有優先權等於 4，小於新資料流的設定優先權 2。而這些 LSPid 之 CB 值皆大於新資料流的頻寬需求，且其 UB 值都是 3Mbps。
   - 根據演算法，在這種情況下，因爲 UBmin>需求，因此我們選擇侵佔 CB 值最高之 LSPid，其 LSPid=1200，CB 值爲 4Mbps。
   - 在釋放 LSPid=1200 之 CR-LSP 之後， 新資料流便直接在此路徑 3_10 建立 CR-LSP，其 LSPid=1400。
   - 我們在演算法的分析中有提到，如果需求的頻寬 < LSPid 之 UB 最低者，則從這些 UB 最低者中，選擇 CB 值最大者。其目的爲可使可用頻寬較集中於某一路徑，對未來要建之 CR-LSP 有幫助。由此實例可看出，可用頻寬集中於 3_10 路徑，其剩餘可用頻寬爲 2.2Mbps。

實例（六）

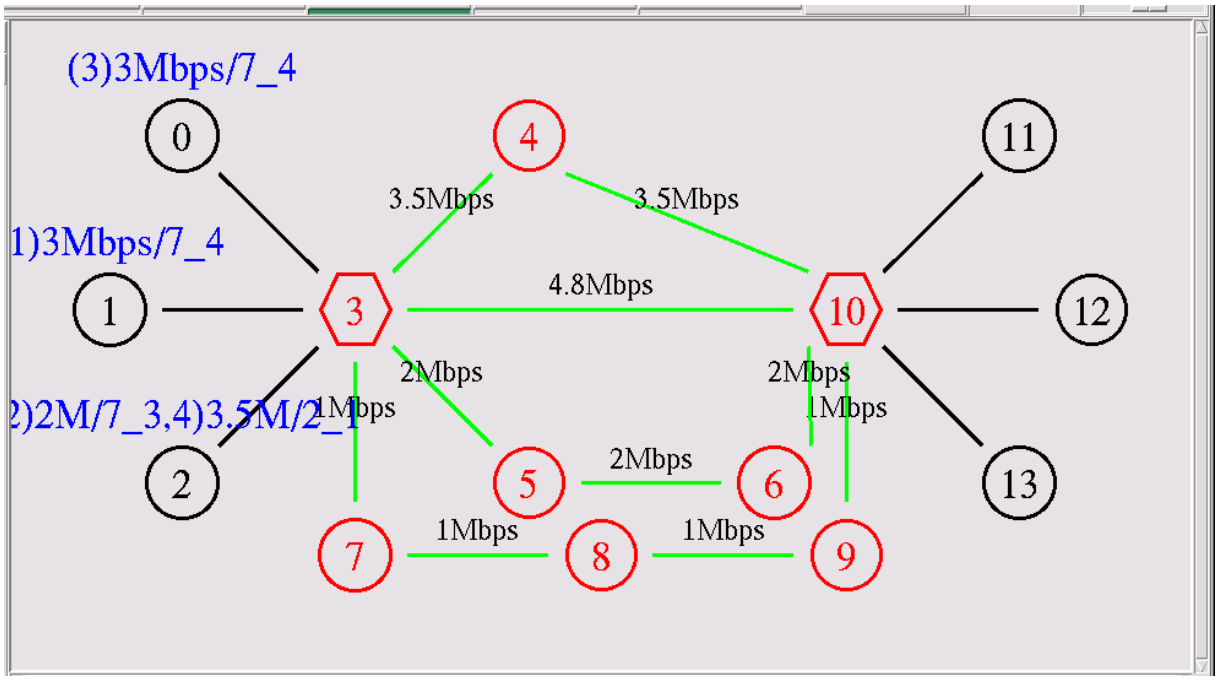

圖 44 實例（六）用來模擬 DLBP 演算法之網路拓撲、各資料流的傳送順序、各鏈接的頻寬及各資料流的需求頻寬

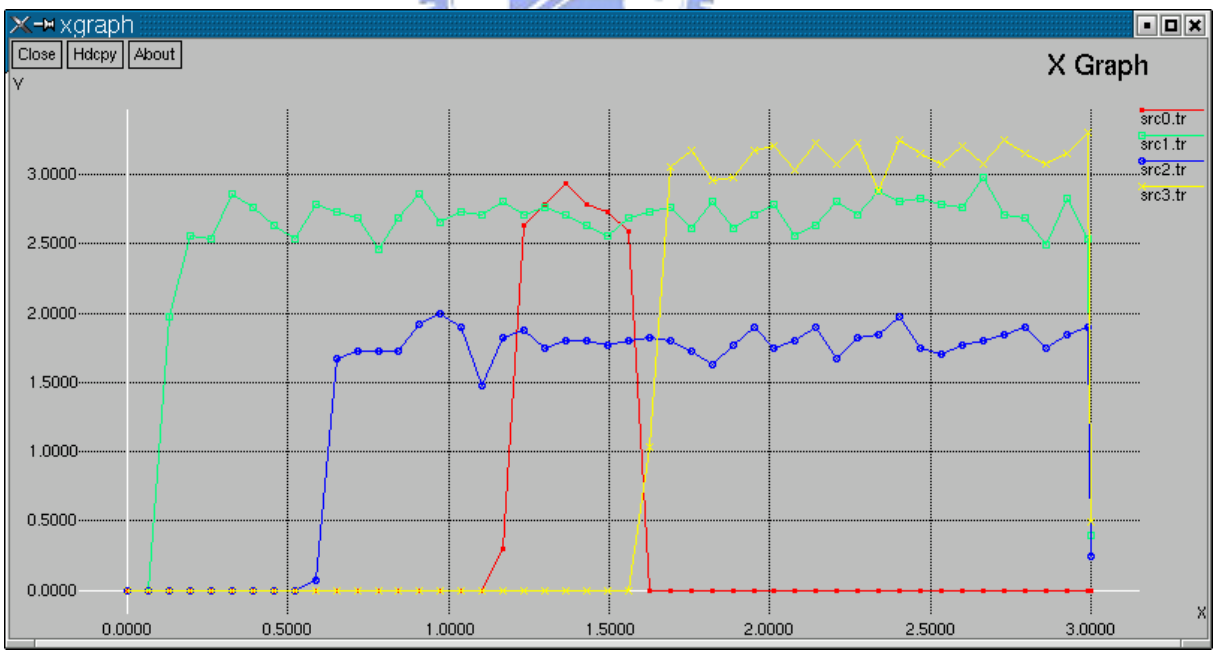

圖 45 實例（六）使用 DLBP 演算法模擬圖 44 得到之每一個資料流的輸出量

```
[root@ip-22 root]# ns /root/ns-allinone-2.1b8a/ns-2.1b8a/tcl/ex/constraint-routing/constraint-routi
ng9.tcl
--> The result of constraint-based routing for lspid 1200 : Explicit Route=3_10
      o The CR-LSP of lspid 1200 has been just established at 0.02018
--> The result of constraint-based routing for lspid 1300 : Explicit Route=3_4_10
      o The CR-LSP of lspid 1300 has been just established at 0.54053942857142856
--> The result of constraint-based routing for update_lspid 1301 : Explicit Route=3_5_6_10
      o The CR-LSP of lspid 1301 has been just established at 1.0615360000000003
--> The result of constraint-based routing for lspid 1100 : Explicit Route=3_4_10
      o The CR-LSP of lspid 1300 has been just released at 1.0820711940124026
      o The CR-LSP of lspid 1100 has been just established at 1.1205394285714287
--> The result of constraint-based routing for lspid 1400 : Explicit Route=3_4_10
      o The CR-LSP of lspid 1100 has been just released at 1.5201554285714285
      o The CR-LSP of lspid 1400 has been just established at 1.5605394285714287
[root@ip-22 root]# X connection to :0.0 broken (explicit kill or server shutdown).
```

圖 46 實例（六）使用 DLBP 演算法模擬圖 44 得到之每一條 CR-LSP 之路徑、建立時間和被釋放時間

分析：

1. 頻寬需求爲（1）3Mbps 及（2）2Mbps 之資料流，順利根據 TSLB 演算法找到路徑建立 CR-LSP，其 LSPid 分別爲 1200 及 1300。
2. 當頻寬需求爲（3）3Mbps 之資料流送出時，根據 DLB 演算法，頻寬需求爲 2Mbps 的資料流被重新路由至 3_5_6_10 路徑，其 LSPid 爲 1301，因而騰出較高容量的 3_4_10 路徑給頻寬需求爲 3Mbps 的新資料流使用，其所建之 CR-LSP 之 LSPid 爲 1100。
3. 當 Ingress LSR LSR3 收到頻寬需求爲（4）3.5Mbps 資料流時，因執行 DLB 演算法無法替此資料流找到足夠頻寬的路徑使用，因此從現已存在的 CR-LSP 中，尋找是否有可侵佔的對象。
   - 在這些現已存在的 CR-LSP 中，持有優先權最低之 CR-LSP 有二個，分別爲 LSPid=1100 及 1200，其持有優先權等於 4，小於新資料流的設定優先權 2。而這些 LSPid 之 CB 值皆大於或等於新資料流的頻寬需求，且其 UB 值都是 3Mbps。
   - 根據演算法，在這種情況下，因爲 UBmin<需求，因此我們選擇侵佔 CB 值最低之 LSPid，其 LSPid=1100，CB 值爲 3.5Mbps。
   - 在釋放 LSPid=1100 之 CR-LSP 之後，新資料流便直接在此路徑 3_4_10 建立 CR-LSP，其 LSPid=1400。
   - 我們在演算法的分析中有提到，如果需求的頻寬 > LSPid 之 UB 最低者，則從這些 UB 最低者中，選擇 CB 值最小者。其目的爲可使可用頻寬較集中於某一路徑，對未來要建之 CR-LSP 有幫助。由此實例可看出，可用頻寬集中於 3_10 路徑，其剩餘可用頻寬爲 1.8Mbps。

實例（七）

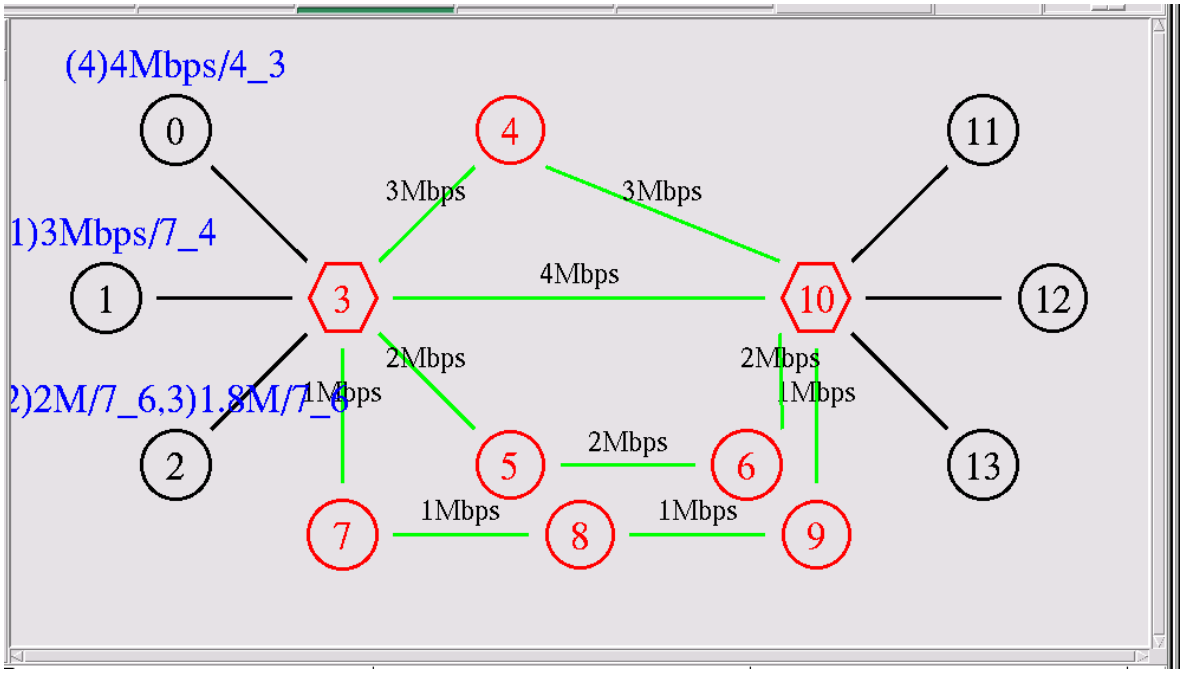

圖 47 實例（七）用來模擬 DLBP 演算法之網路拓撲、各資料流的傳送順序、各鏈接的頻寬及各資料流的需求頻寬

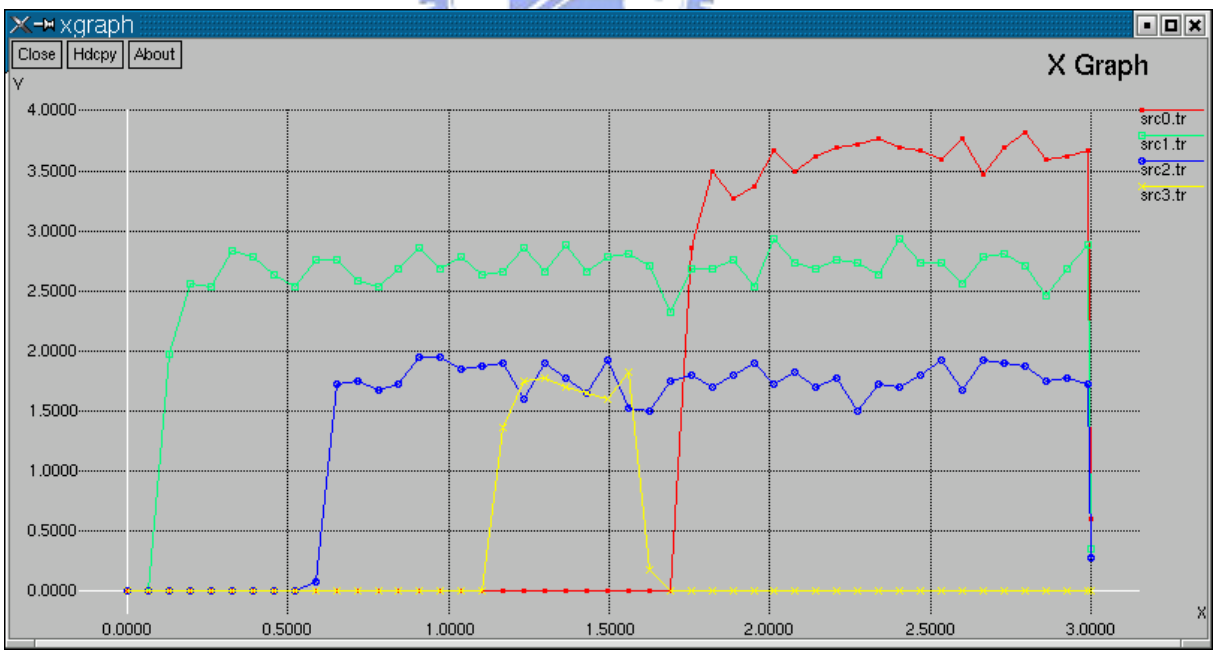

圖 48 實例（七）使用 DLBP 演算法模擬圖 47 得到之每一個資料流的輸出量

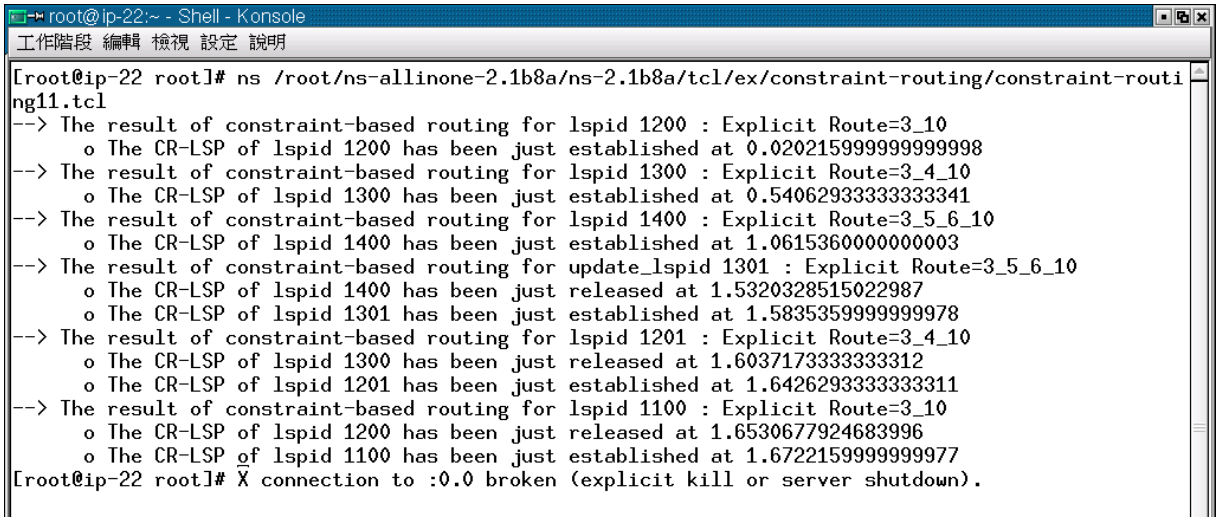

圖 49 實例（七）使用 DLBP 演算法模擬圖 47 得到之每一條 CR-LSP 之路徑、建立時間和被釋放時間

分析：

1. 頻寬需求爲（1）3Mbps、（2）2Mbps 及（3）1.8Mbps 之資料流，順利根據 TSLB 演算法找到路徑建立 CR-LSP，其 LSPid 分別爲 1200、1300 及 1400。
2. 當 Ingress LSR LSR3 收到頻寬需求爲（4）4Mbps 資料流時，因執行 DLB 演算法無法替此資料流找到足夠頻寬的路徑使用，因此從現已存在的 CR-LSP 中，尋找是否有可侵佔的對象。
   - 在這些現已存在的 CR-LSP 中，持有優先權最低之 CR-LSP 有二個，分別爲 LSPid=1300 及 1400，其持有優先權等於 6，小於新資料流的設定優先權 4。而這些 LSPid 之 CB 值皆小於新資料流的頻寬需求。
   - 根據演算法，在這種情況下，我們選擇侵佔 UB 值最低之 LSPid，其 LSPid=1400，UB 值爲 1.8Mbps。
   - 但因 LSPid=1400 之 CB 值小於新資料流的頻寬需求，則我們必須進一步判斷若將被選到的 CR-LSP 釋放，對新的資料流是否可利用 DLB 演算法找到足夠的頻寬來建 CR-LSP？判斷的結果爲可找得到足夠的頻寬，因此釋放此被選到的 LSPid=1400 之 CR-LSP，並利用 DLB 演算法，將頻寬需求爲 2Mbps 的資料流重新路由至 3_5_6_10 的路徑，而將頻寬需求爲 3Mbps 的資料流重新路由至 3_4_10 的路徑，因而騰出較高容量的 3_10 路徑給頻寬需求爲 4Mbps 的新資料流使用，其所建的 CR-LSP 之 LSPid=1100。
   - 由此實例可看出，選擇侵佔 UB 值最低之 LSPid，可使網路資源利用率最高。

實例（八）

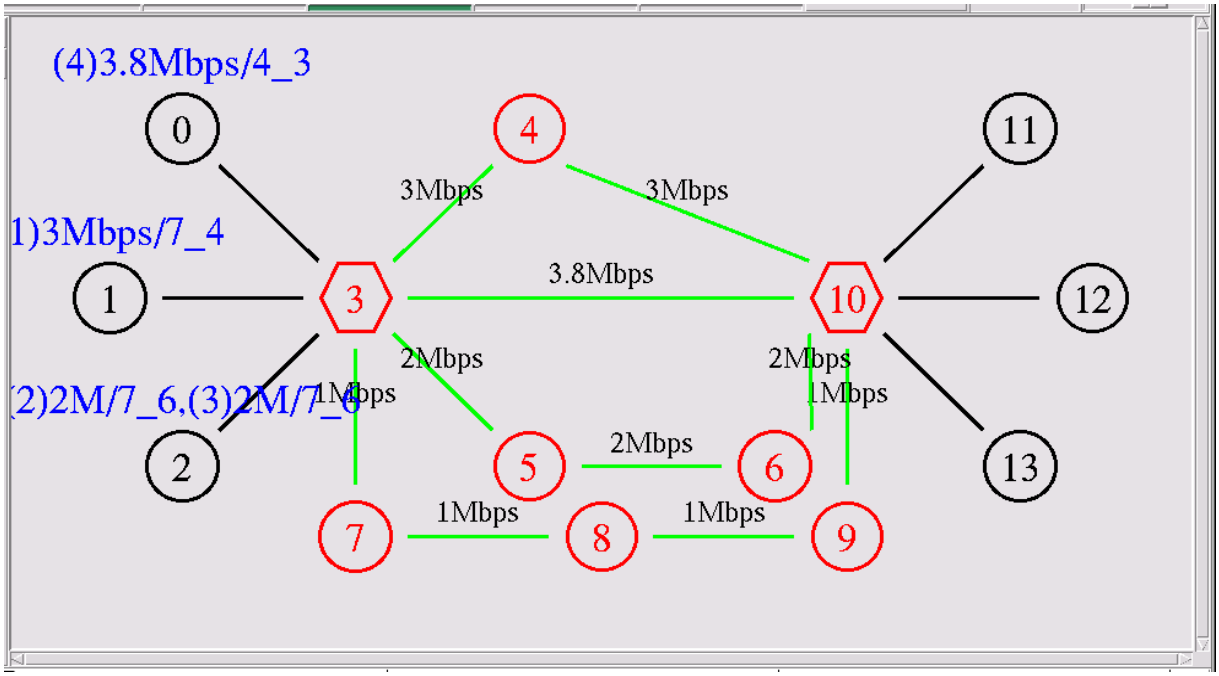

圖 50 實例（八）用來模擬 DLBP 演算法之網路拓撲、各資料流的傳送順序、各鏈接的頻寬及各資料流的需求頻寬

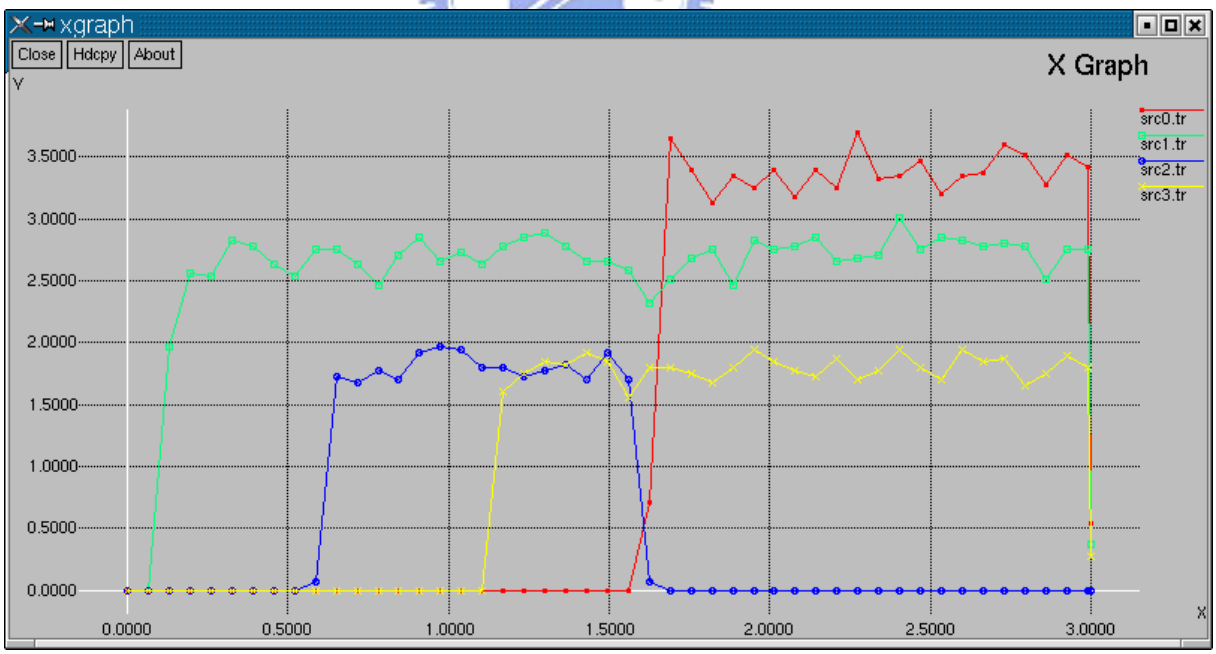

圖 51 實例（八）使用 DLBP 演算法模擬圖 50 得到之每一個資料流的輸出量

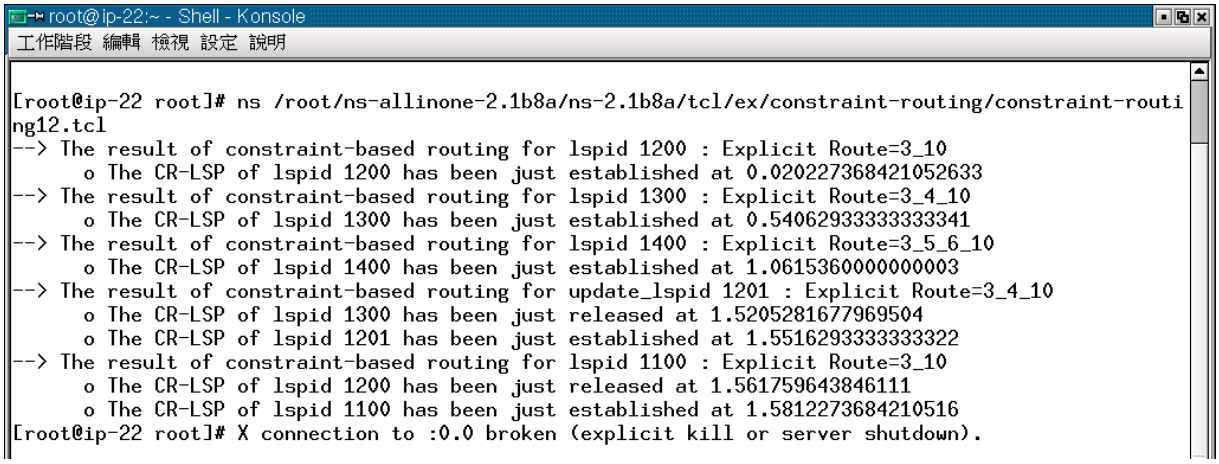

圖 52 實例（八）使用 DLBP 演算法模擬圖 50 得到之每一條 CR-LSP 之路徑、建立時間和被釋放時間

分析：

1. 頻寬需求爲（1）3Mbps、（2）2Mbps 及（3）2Mbps 之資料流，順利根據 TSLB 演算法找到路徑建立 CR-LSP，其 LSPid 分別爲 1200、1300 及 1400。
2. 當 Ingress LSR LSR3 收到頻寬需求爲（4）3.8Mbps 資料流時，因執行 DLB 演算法無法替此資料流找到足夠頻寬的路徑使用，因此從現已存在的 CR-LSP 中，尋找是否有可侵佔的對象。
   - 在這些現已存在的 CR-LSP 中，持有優先權最低之 CR-LSP 有二個，分別爲 LSPid=1300 及 1400，其持有優先權等於 6，小於新資料流的設定優先權 4。而這些 LSPid 之 CB 值皆小於新資料流的頻寬需求，且 UB 值都是 2Mbps。
   - 根據演算法，在這種情況下，我們選擇侵佔 CB 值最高之 LSPid，其 LSPid=1300，CB 值爲 3Mbps。
   - 但因 LSPid=1300 之 CB 值小於新資料流的頻寬需求，則我們必須進一步判斷若將被選到的 CR-LSP 釋放，對新的資料流是否可利用 DLB 演算法找到足夠的頻寬來建 CR-LSP？判斷的結果爲可找得到足夠的頻寬，因此釋放此被選到的 LSPid=1300 之 CR-LSP，並利用 DLB 演算法，將頻寬需求爲 3Mbps 的資料流重新路由至 3_4_10 的路徑，因而騰出較高容量的 3_10 路徑給頻寬需求爲 3.8Mbps 的新資料流使用，其所建的 CR-LSP 之 LSPid=1100。
   - 由此實例可看出，選擇侵佔 CB 值較高之 LSPid，其 CB=3Mbps，只需要重新路由一次。但若選擇侵佔 CB 值較低之 LSPid，其 CB=2Mbps，則需要重新路由二次，因此我們選擇侵佔 CB 值較高之 LSPid。

實例（九）

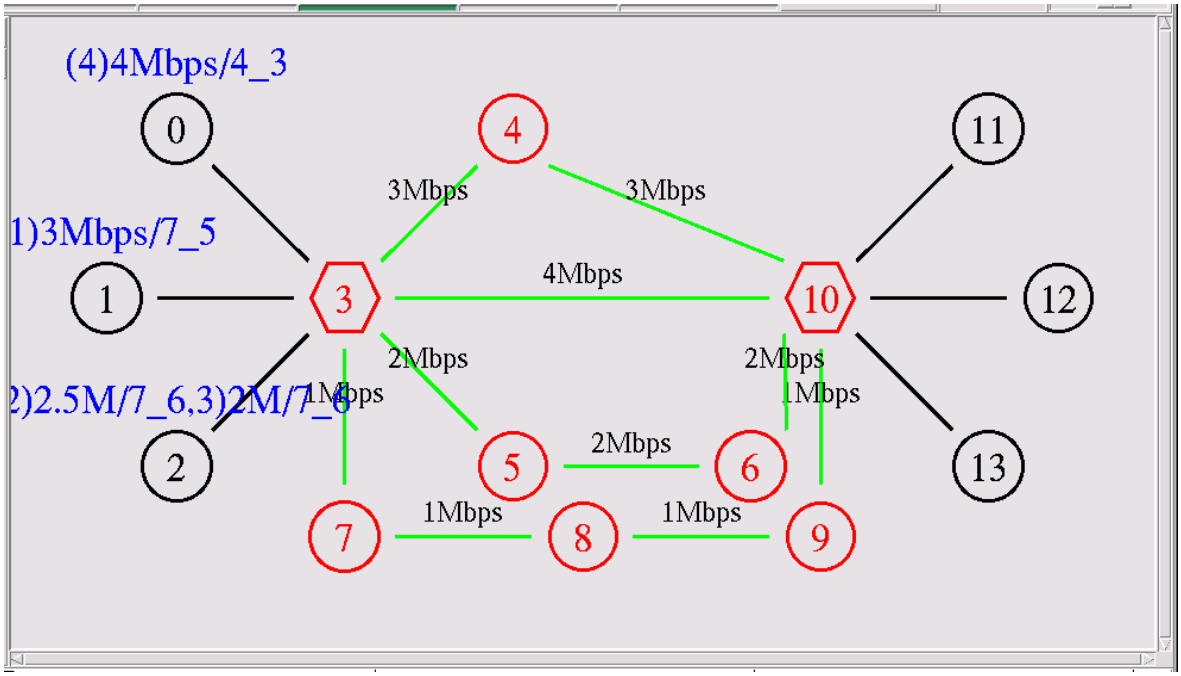

圖 53 實例（九）用來模擬 DLBP 演算法之網路拓撲、各資料流的傳送順序、各鏈接的頻寬及各資料流的需求頻寬

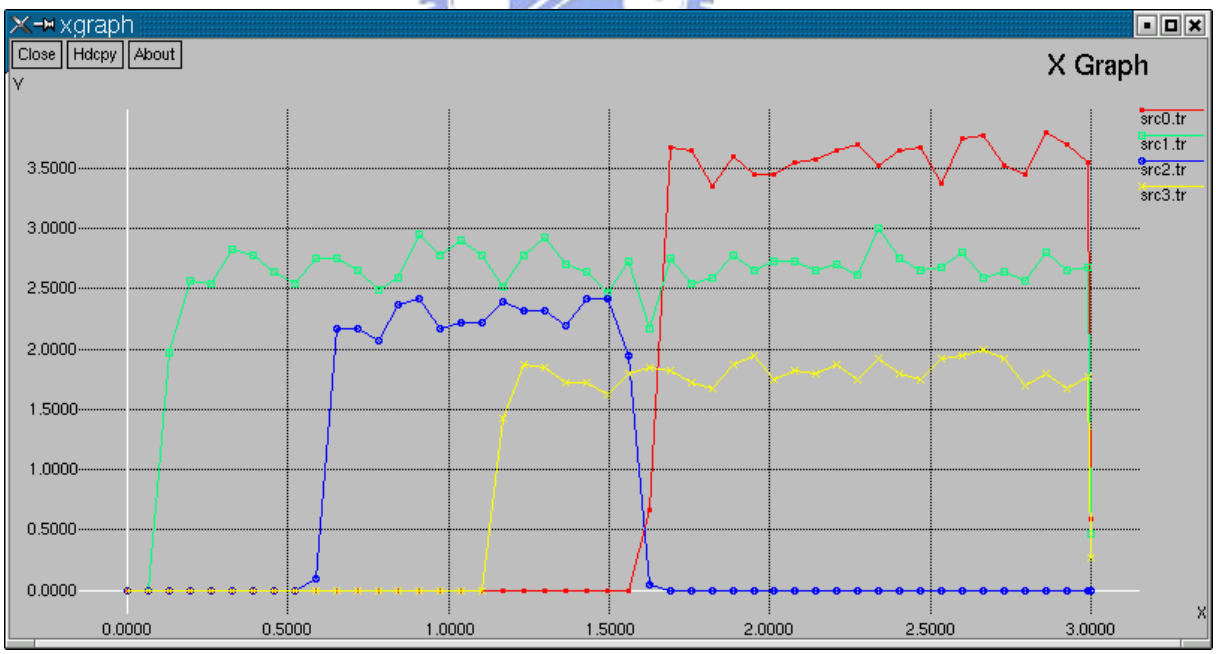

圖 54 實例（九）使用 DLBP 演算法模擬圖 53 得到之每一個資料流的輸出量

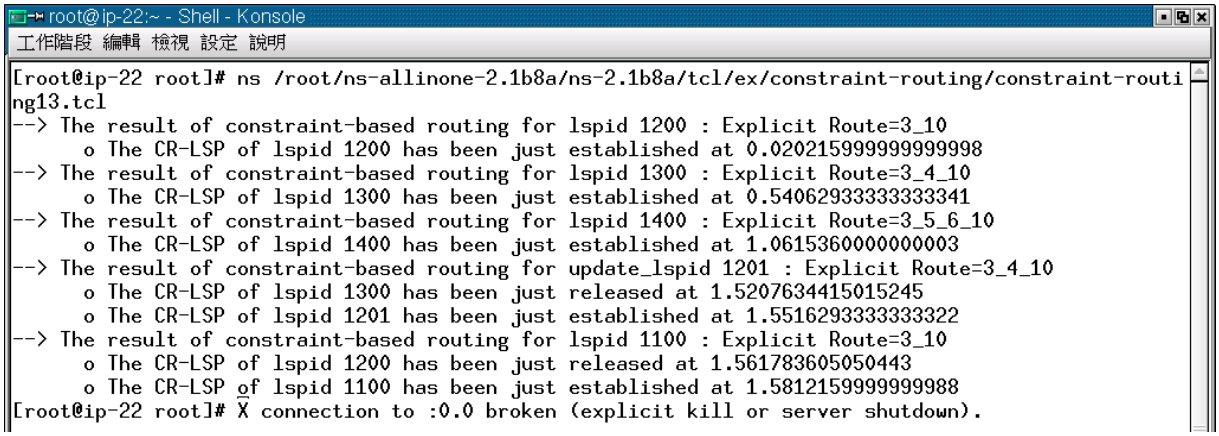

圖 55 實例（九）使用 DLBP 演算法模擬圖 53 得到之每一條 CR-LSP 之路徑、建立時間和被釋放時間

分析：

1. 頻寬需求爲（1）3Mbps、（2）2.5Mbps 及（3）2Mbps 之資料流，順利根據 TSLB 演算法找到路徑建立 CR-LSP，其 LSPid 分別爲 1200、1300 及 1400。
2. 當 Ingress LSR LSR3 收到頻寬需求爲（4）4Mbps 資料流時，因執行 DLB 演算法無法替此資料流找到足夠頻寬的路徑使用，因此從現已存在的 CR-LSP 中，尋找是否有可侵佔的對象。
   - 在這些現已存在的 CR-LSP 中，持有優先權最低之 CR-LSP 有二個，分別爲 LSPid=1300 及 1400，其持有優先權等於 6，小於新資料流的設定優先權 4。而這些 LSPid 之 CB 值皆小於新資料流的頻寬需求。
   - 根據演算法，在這種情況下，我們選擇侵佔 UB 值最低之 LSPid，其 LSPid=1400，UB 值爲 2Mbps。
   - 但因 LSPid=1400 之 CB 值小於新資料流的頻寬需求，則我們必須進一步判斷若將被選到的 CR-LSP 釋放，對新的資料流是否可利用 DLB 演算法找到足夠的頻寬來建 CR-LSP？判斷的結果爲找不到。表示此 CR-LSP 即使被釋放，對新資料流亦無幫助，因此不予考慮 LSPid=1400 之 CR-LSP。
   - 除了 LSPid=1400 之 CR-LSP 之外，持有優先權最低之 CR-LSP 爲 LSPid=1300。因此選擇侵佔 LSPid=1300 之 CR-LSP。在釋放 LSPid=1300 之 CR-LSP 之後，利用 DLB 演算法，將頻寬需求爲 3Mbps 的資料流重新路由至 3_4_10 的路徑，因而騰出較高容量的 3_10 路徑給頻寬需求爲 4Mbps 的新資料流使用，其所建的 CR-LSP 之 LSPid=1100。

實例（十）

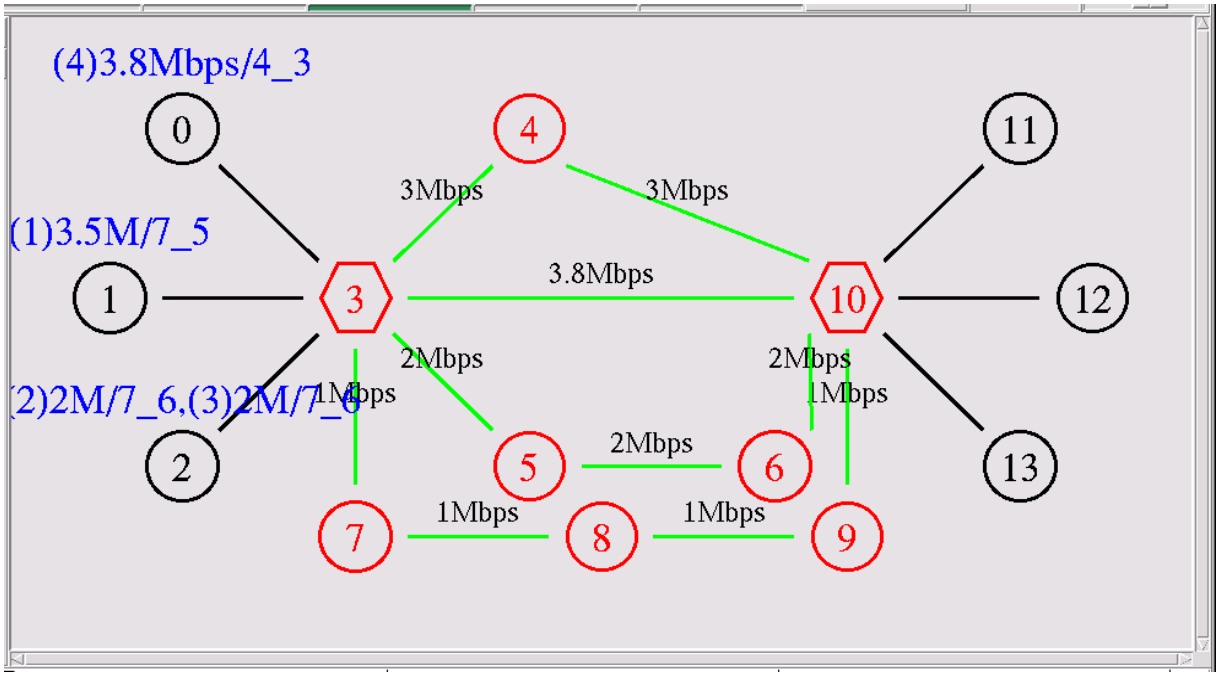

圖 56 實例（十）用來模擬 DLBP 演算法之網路拓撲、各資料流的傳送順序、各鏈接的頻寬及各資料流的需求頻寬

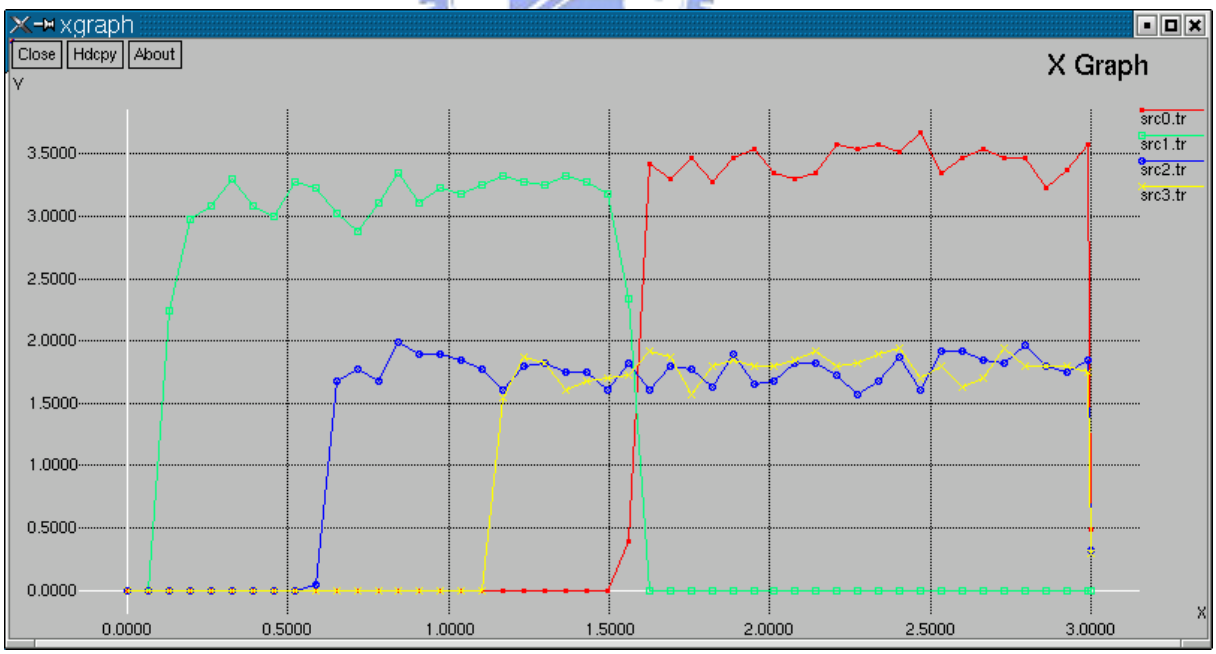

圖 57 實例（十）使用 DLBP 演算法模擬圖 56 得到之每一個資料流的輸出量

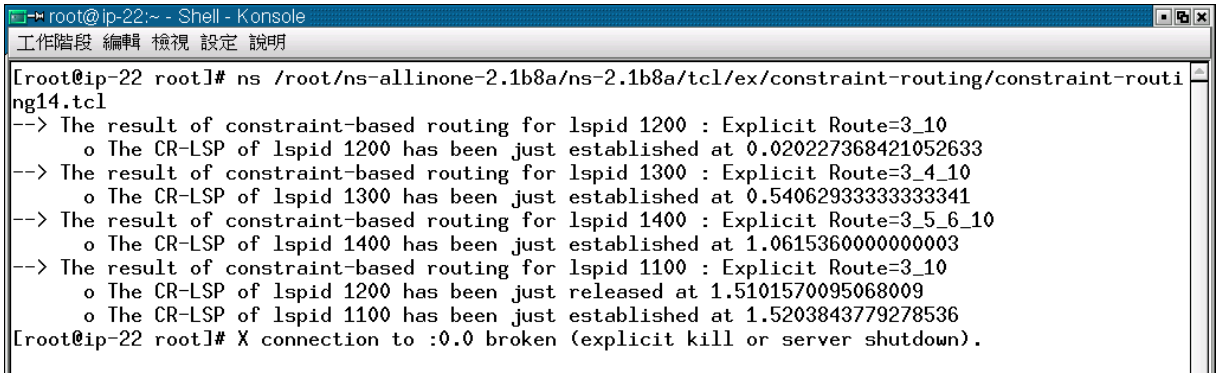

圖 58 實例（十）使用 DLBP 演算法模擬圖 56 得到之每一條 CR-LSP 之路徑、建立時間和被釋放時間

分析：

1. 頻寬需求爲（1）3.5Mbps、（2）2Mbps 及（3）2Mbps 之資料流，順利根據 TSLB 演算法找到路徑建立 CR-LSP，其 LSPid 分別爲 1200、1300 及 1400。
2. 當 Ingress LSR LSR3 收到頻寬需求爲（4）3.8Mbps 資料流時，因執行 DLB 演算法無法替此資料流找到足夠頻寬的路徑使用，因此從現已存在的 CR-LSP 中，尋找是否有可侵佔的對象。
   - 在這些現已存在的 CR-LSP 中，持有優先權最低之 CR-LSP 有二個，分別爲 LSPid=1300 及 1400，其持有優先權等於 6，小於新資料流的設定優先權 4。而這些 LSPid 之 CB 值皆小於新資料流的頻寬需求，且 UB 值都是 2Mbps。
   - 根據演算法，在這種情況下，我們選擇侵佔 CB 值最高之 LSPid，其 LSPid=1300，CB 值爲 3Mbps。
   - 但因 LSPid=1300 之 CB 值小於新資料流的頻寬需求，則我們必須進一步判斷若將被選到的 CR-LSP 釋放，對新的資料流是否可利用 DLB 演算法找到足夠的頻寬來建 CR-LSP？判斷的結果爲找不到。表示此 CR-LSP 即使被釋放，對新資料流亦無幫助，因此不予考慮 LSPid=1300 之 CR-LSP。
   - 在考慮選擇侵佔 LSPid=1400 時，亦因釋放此 CR-LSP 對新資料流亦無幫助，因此亦不予考慮。
   - 最後，考慮侵佔優先權較高的 CR-LSP，其 LSPid=1200，持有優先權等於 5，小於新資料流的設定優先權 4。但因爲 LSPid=1200 之 CB 值等於新資料流的頻寬需求，因此釋放 LSPid=1200 之 CR-LSP 之後，新資料流便利用此路徑 3_10，建立 CR-LSP，其 LSPid=1100。