

為 RSVP 而設計之高效率資源保留樣式

An Efficient Resource Reservation Style for RSVP

童曉儒(Sheau-Ru Tong)、林美賢(Mei-Hsien Lin)

國立屏東科技大學 資訊管理系

{[@mail.npust.edu.tw](mailto:srtong,mslin)}

摘要

RSVP(Resource Reservation Protocol) 已漸被公認是一種必要的傳送「服務品質(QoS)」需求的網路協定，其目的是將 QoS 需求動態的通知各網路節點，以便保留適當的資源。RSVP 設計了三種資源保留樣式，分別是：Fixed-Filter(FF)、Shared-Explicit(SE) 和 Wildcard-Filter(WF)，來處理各種溝通模式的需求。然而我們發現此三種保留樣式，不足以將溝通模式的 QoS 需求完整的告知每一網路節點，因而可能導致嚴重的資源浪費，而影響網路的使用效率。有鑑於此，我們提出一套更有效率的保留樣式，此保留樣式能完整的將 QoS 需求傳達至每一網路節點，使每一節點做到最佳的資源保留，因此完全免除了資源浪費的困擾，而所付出的運算與通訊代價則十分有限。我們模擬了在 TANet 2 的部分網路架構下，一套分級式遠距教學系統的運作的情形。其結果驗證了我們所提資源保留樣式的優越性。

關鍵字：服務品質保證(QoS)、資源保留協定(Resource Reservation Protocol, RSVP)、保留樣式(Reservation Style)。

1. 緒論

近年來使用 Internet 的人口大幅的成長，透過 Internet 傳送資料非常普遍，且網路應用多元化，使得今日的資料型態不再侷限於文字資料，更包括資料量大的多媒體資料(如：影片、音樂、動畫)，此類資料需要更高的網路頻寬與更嚴格的傳送速度，雖然網路技術的進步，大幅提升了網路頻寬，但由於目前仍是以 best-effort 為主要的傳送模式，且缺乏有效的資源保留與管理機制，所以無法真正確保資料傳送的品質。有鑑於此，下一代的網路(如 Internet II)，將提供具有保證傳輸「服務品質(QoS)」的功能，服務型態以 Integrated Service 與 Differentiated Service 為兩大主流 [1,2]，兩者各有其特性。Integrated Service 是針對每一個網路串流(flow)給予 QoS 的保證，雖然每個 flow 都享有精確的 QoS 保證，但當 flow 數量過多時，網路的控制的負擔就會變得沈重，所以此服務僅適用於較單純的環境中，不利於系統的擴張；相對的，Differentiated Service 則是將網路上的網路串流依所需要的 QoS 分成為幾大類別，然後針對不同的類別，提供不同程度的 QoS，如此網路的控制只與類別的數量有關，而與 flow

的數量無關，所以較無不良擴充性的問題，但 QoS 的保證無法做到十分精準，可能會造成網路資源保留超過實際的需求量，一般而言，此服務適用於廣域的連結上。目前較為大眾所接受的是一種合併式的架構(如圖 1 所示)：用戶區域(customer premises)採用 Integrated Service；而 ISP 廣域區域則採用 Differentiated Service [5]。

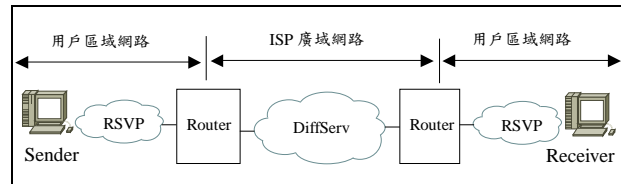


圖 1 IntServ 與 DiffServ 共存圖

不論是用何種服務型態，不可或缺的是一套告知(signal)網路應保留多少資源的機制，此種機制可能是由人為決定，採手動的方式來控制網路節點，此方法是由主導 Differentiated Service 的研究團體所提倡，雖然簡單直接，但缺乏彈性，較不適用於動態變化性明顯的 Internet；或是由需要服務的終端系統主動發出 QoS 需求，網路自動蒐集此訊息，然後主動做出保留決定，此方法是由主導 Integrated Service 研究團體所提倡，富有彈性且符合 Internet 之精神。因此本論文的研究重點將放在後者。「資源保留協定(RSVP/ Resource Reservation Protocol)」 [4,8,9,10] 是屬於 Integrated Service 中的一部分，用來告知(signaling)路徑上的網路節點有關 QoS 的需求，以便在送收兩者之間建立具有 QoS 保證的通道。雖然 RSVP 起源於 Integrated Service，但它適用於前述的合併式的架構中，且漸被公認是用來達成 QoS signaling 的最佳選擇。RSVP 必須運作於既有的路由協定之上，適用於單點(unicast)或群播(multicast)的傳輸模式，採用 Receiver-oriented 的方式運作，也就是由需要服務的用戶端系統主動發出 QoS 需求給伺服器端，同時採用 Soft-State 的方式維護保留資源的狀態 [4,6]，也就是此 QoS 需求必須週期性的送出，以便不斷的更新保留的資源狀態，否則如果用戶端系統離線或者路徑更改，過久未更新的保留資源會被取消。RSVP 是針對 flow 做資源保留，所謂的 flow 可以是一點對點的連線，或者是一 multicast session。

RSVP 為了更有效率的達到網路資源共享，提出

了三種保留樣式(Reservation Style)來處理 QoS 的需求：Fixed-Filter Style (FF)、Shared-Explicit Style (SE) 和 Wildcard-Filter Style (WF)。我們可以依實際應用系統的需要選用適當的保留樣式，當網路上的節點收到數個屬於同一 flow 的 QoS 需求時，會依其中所指定的保留樣式進行需求合併，然後做出適當的資源保留決定，接著產生新的 QoS 需求給下一節點。然而我們發現此三種保留樣式所產生的合併需求，未能反映出確切的 QoS 需求，因此某些網路節點所保留的資源會超出實際上所需要的量，因而導致嚴重的資源浪費，而間接地影響整個網路的使用效率。有鑑於此，在本論文中我們提出一套創新的保留樣式，用此保留樣式所合併後的 QoS 需求即代表著實際的需求，如此一來每一節點可做到最佳的資源保留，完全免除了前述資源浪費的問題，而所付出的運算與通訊代價則十分有限。我們模擬了在 TANet 2 的部分網路架構下，一套分級式遠距教學系統的運作情形，其中依學生付費的不同，分成幾種服務等級。其結果驗證了我們所提資源保留樣式能有效的節省網路資源與降低上線拒絕率。

本文下面的文章結構說明如下，第二章針對 RSVP 的運作原理及所提供之三種保留樣式的運作做詳細介紹，並探討保留資源浪費的問題。第三章接著介紹我們所提出的保留樣式，並討論有關實做此保留樣式所需的額外負擔。第四章描述我們所模擬的網路環境與遠距教學系統的運作，將我們的保留樣式與原有的保留樣式做一比較，針對消耗的網路資源與需求拒絕率做一討論。第五章為本文總結。

2. RSVP 之概述

2.1 RSVP 運作原理

RSVP 採用由收方主動提出 QoS 要求的模式(如圖 2)，主要運作如下[3]：(1)送方負責送出 Path 訊息給收方，Path 訊息主要攜帶著送方資料流量特性描述 Tspec 與送方的辨識資訊 sender template (如 sender IP 與 port number)。Path 訊息會依照繞送協定(如：OSPF, DVMRP(Distance Vector Multicast Routing Protocol))等 [7]送至收方，凡於路徑上接收到該 Path 訊息的網路節點介面，會建立一 Path 狀態，其中記錄了上一個送出此 Path 訊息的網路節點 IP 位址與 sender template。(2)收方負責送出 Resv 訊息給送方，Resv 訊息主要攜帶著資源保留描述 Flowspec 與 Filterspec，Resv 訊息依據先前所記錄於網路節點中的 Path 狀態，循與 Path 訊息相反的路徑抵達送方，凡接收到 Resv 訊息的網路節點介面，會依照 Flowspec 與 Filterspec 建立一 Resv 狀態，保留適當的資源。(3)接著資料會循已建立資源保留的路徑，從送方傳至收方。

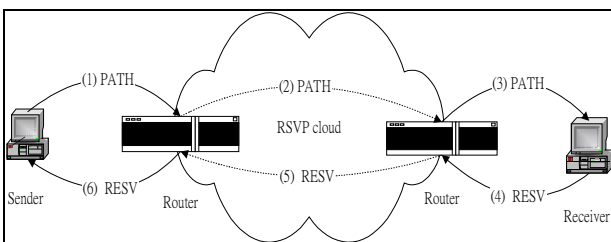


圖 2 RSVP 訊息之流向圖

Flowspec 的值就代表著 QoS 的值(例如：1 Mbps 頻寬或 10 msec end-to-end 延遲)，主要是根據 Path 訊息中的 Tspec 計算出來的，為了方便討論以下我們採用「頻寬」為 QoS 的值¹。而 Filterspec 就描述那些送方以那種保留樣式共享此 QoS。

2.2 RSVP 保留樣式

保留樣式共分以下三種：(1) Fixed-Filter (FF) Style：QoS 是針對某一特定的送方。我們用 FF $\{(S_1, Q_{S_1}), (S_2, Q_{S_2}), \dots\}$ 來表示，其中 S_x 表示所選擇之送方的 ID (即 IP + port ID)， Q_{S_x} 表示所需之頻寬。(2) Shared-Explicit (SE) Style：QoS 是針對某一群特定的送方。我們以 SE $\{(S_1, S_2, \dots) Q\}$ 來表示，其中 (S_1, S_2, \dots) 代表了一串列共享 QoS 的送方的 ID， Q 表示所需之頻寬。(3) Wildcard-Filter (WF) Style：QoS 是針對所有屬於同一 flow (session) 中的所有的送方。我們以 WF $\{*, Q\}$ 來代表，其中 * 表示所有的送方， Q 表示所需之頻寬。我們接著討論在不同的保留樣式下，路由器如何處理所收到來自另一個路由器的 Resv 訊息。

■ FF 保留樣式：收到 FF 訊息的介面，針對不同的來源路由器記錄個別的 (S_x, Q_{S_x}) 。如果來源路由器相同且已有屬於 S_x 的記錄，則採用新收到的 Q_{S_x} 值。在此介面上所需保留的頻寬為所有存於此介面上 Q_{S_x} 值的總和。接著所有介面上屬於同一送方的記錄合併，每一送方對應產生新的 (S_x, Q_{S_x}') ，其中 Q_{S_x}' 所有同屬於 S_x 的記錄中的最大值，然後這些新的記錄包裹成 FF 的格式，透過可抵達送方的介面分送出。以圖 3 為例，說明 FF 保留樣式之運作。

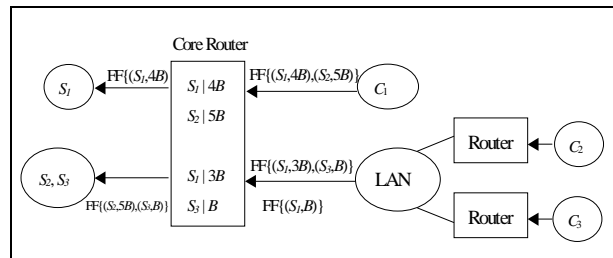


圖 3 Fixed-Filter Reservation

FF 保留樣式之運作說明如下。

- (1) S_1 、 S_2 與 S_3 表示送方， C_1 、 C_2 與 C_3 表示收方。
- (2) $FF\{(S_1, 4B), (S_2, 5B)\}$ ，表示收方 C_1 對送方 S_1 要求 4B 單位頻寬，對送方 S_2 要求 5B 單位頻寬。 $FF\{(S_1, 3B), (S_3, B)\}$ ，表示收方 C_2 對送方 S_1 要求 3B 單位頻寬，對送方 S_3 則要求 B 單位頻寬。而表示式 $FF\{(S_1, B)\}$ ，表示收方 C_3 對 S_1 要求一個 B 單位頻寬。
- (3) Core Router 於右上的介面保留 9B 單位，於右下的介面保留 4B 單位。然後於左上介面送出 $FF\{(S_1, 4B)\}$ ，於左下介面送出 $FF\{(S_2, 5B), (S_3, B)\}$ 。

■ SE 保留樣式：收到 SE 訊息的介面，針對不同的來源路由器記錄個別的 $(S_1, S_2, \dots) Q$ 。如果有舊的記錄，則將之取代。在此介面上所需保留的頻寬為所

¹ 如何考慮一個以上的 QoS 參數是一個值得進一步探討的課題。

有存於此介面上 Q 值的最大值。每個介面產生並送出一個合併的 $SE\{(S_1', S_2', \dots) Q'\}$ ，其中 (S_1', S_2', \dots) 為所有必須透過該介面方可抵達的送方所構成的串列，而 Q' 值為個別成員在介面上所記錄 Q 值的最大值。以圖 4 為例，說明 SE 保留樣式之運作。

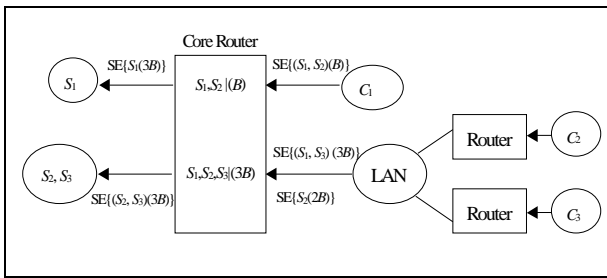


圖 4 Shared-Explicit Reservation

SE 資源保留樣式之運作說明如下。

- (1) $SE\{(S_1, S_2)(B)\}$ ，表示收方 C_1 對送方 S_1 與 S_2 要求 B 單位頻寬。 $SE\{(S_1, S_3)(3B)\}$ ，表示收方 C_2 對送方 S_1 與 S_3 要求 $3B$ 單位頻寬。 $SE\{S_2(2B)\}$ ，表示收方 C_3 對送方 S_2 要求 $2B$ 單位頻寬。
- (2) Core Router 則於右上介面保留 B 單位、於右下介面保留 $3B$ 單位，然後於左上介面送出 $SE\{S_1(3B)\}$ ，於左下介面送出 $SE\{(S_2, S_3)(3B)\}$ 。

■ WF 保留樣式：比較原保留之頻寬與 Q 值，如果 Q 值較大，則在此介面上保留的頻寬更新為 Q 。如果所有介面保留的頻寬的最大值有所改變，則於每個介面上（除了剛接收該訊息的介面）送出 $WF\{*(Q')\}$ ，此處 Q' 為新的最大值。以圖 5 為例，說明 WF 保留樣式之運作。

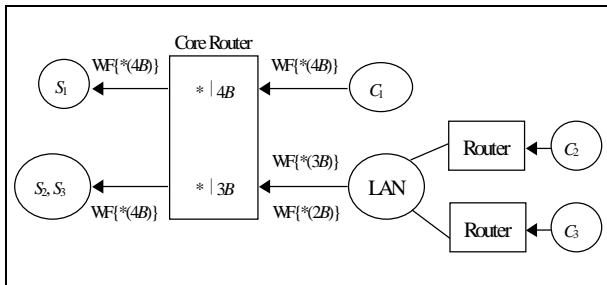


圖 5 Wildcard-Filter Reservation

WF 保留樣式的運作說明如下。

- (1) $WF\{*(4B)\}$ ，表示收方 C_1 對所有的送方(*)要求 $4B$ 單位頻寬。 $WF\{*(3B)\}$ ，表示收方 C_2 對所有的送方(*)要求 $3B$ 單位頻寬。 $WF\{*(2B)\}$ ，表示收方 C_3 對所有的送方(*)要求 $2B$ 單位頻寬。
- (2) Core Router 於右上介面保留 $4B$ 單位，於右下介面保留 $3B$ 單位，然後於左上與左下介面皆送出 $WF\{*(4B)\}$ 。

注意，如果路由器所推導出的新訊息內容與先前送出的訊息內容相同，則放棄立即傳送，而等到更新週期到期時方主動送出。介面上所記錄的狀態資料，如果超過某一時間上限仍未被更新，即會被主動移除。RSVP 僅允許在一個 session 中使用一種保留樣式。為了說明 RSVP 保留樣式所產生資源浪費的問題，我們以下用例子來說明。

2.3 資源浪費問題之探討

一個應用系統可能涉及到多種不同形式的資料流，且資料流之間可能存在某種邏輯關聯，而此邏輯關聯視通訊模式而定，可能很簡單，例如一群人共同分享一個聲音頻道；也可能相當複雜且動態改變，例如一個遠距教學環境，不同的學員可選修不同的課程，不同的上課型式--同步或非同步線上教學，不同的選修方式--正式或旁聽等。為了應付如此多樣而變化的環境，需要一套有效率的方法來描述系統的通訊模式，以方便分析資源的保留。在此我們採用「真值表」的方法來描述某一收方如何選擇接收送方的資料流。假設在整體網路架構中，有 n 個送方可被收方選擇時，我們將他們編號由 1 到 n ，並以 S_i 表示第 i 個送方， S_i 是否被選擇以 0 或 1 表示，0 表示沒被選擇，1 表示被選擇。所以收方對送方選擇的所有組合有 2^n 種。某組合是否合法以 e 表示，若其值為 1 表示合法，0 表示不合法，一種合法的組合又稱之為一個 Truth Assignment。在一個 Truth Assignment 中，所有 S_i 狀態值為 1 的送方所構成的集合稱為一個 Active Set，以 A 表示。

例如：假設網路中有三個送方分別為 S_1 、 S_2 與 S_3 ，則可建立 $2^3=8$ 種組合之真值表，其組合情形如表 1 所示。依需求產生結果 e ，在表 1 中 e 值為 1 有 A_2 、 A_4 與 A_5 ，則 A_2 、 A_4 與 A_5 ，稱為 Truth Assignment 的組合，在每一組合中其 Active Set 的值分別為 $A_2=\{S_3\}$ ， $A_4=\{S_2, S_3\}$ ， $A_5=\{S_1\}$ 。在表 1 中，網底的部分為 Truth Assignment 與 Active Set 的描述。

表 1 真值表範例

	S_1	S_2	S_3	e	w
A_1	0	0	0	0	0
A_2	0	0	1	1	$b_3=7$
A_3	0	1	0	0	0
A_4	0	1	1	1	$b_2+b_3=11$
A_5	1	0	0	1	$b_1=3$
A_6	1	0	1	0	0
A_7	1	1	0	0	0
A_8	1	1	1	0	0

令 S_i 所須消耗的頻寬以 b_i 表示之。在一個 Truth Assignment k 中，所有 Active Set 內的送方皆會被接收，所以需要的頻寬可表示為

$$w_k = \sum_{\forall S_i \in A_k} b_i$$

又由於一個時間僅存在一種組合，所以必須至少保留的總頻寬為所有 Truth Assignment 所需頻寬中最大者，我們稱此頻寬為此真值表 T 的最小頻寬，表示如下。

$$B_T = \max\{w_k\}$$

以表 1 為例，假設三個送方 S_1 、 S_2 與 S_3 的各自頻寬分別為 $b_1=3$ 、 $b_2=4$ 與 $b_3=7$ 。則所需的 w_k 列於最右欄，而 B_T 為 $w_4=b_2+b_3=4+7=11$ 。有了以上的基礎後，我們嘗試推演出 FF 與 SE 保留樣式的頻寬保留需求。(在此我們不特別討論 WF，因為就頻寬保留而言，它可視為 SE 的一種“選擇所有送方”的特例)

■ FF 保留樣式 (FF $\{(S_1, Q_{S1}), (S_2, Q_{S2}), \dots\}$)

主要是針對特定的送方，做個別資源的保留。假設網路架構如圖 6 所示，收方 C_1 的通訊模式為表 1 的真值表。只要某送方曾出現於某一 Active Set 中，就必須對該送方發出需求，所以 C_1 送出的需求便為 FF $\{(S_1,3),(S_2,4),(S_3,7)\}$ 。接著各連線(介面)上頻寬保留情形分述如下。

- (1) $C_1 \rightarrow R_1(S_1, S_2, S_3)$ ：表示要到達送方 S_1, S_2, S_3 ，需經過 $C_1 \rightarrow R_1$ 路徑，此路徑保留頻寬為 $b_1+b_2+b_3=3+4+7=14$ 。
- (2) $R_1 \rightarrow R_2(S_1, S_2)$ ：要到達送方 S_1, S_2 ，需經過 $R_1 \rightarrow R_2$ 路徑，此路徑保留頻寬為 $b_1+b_2=3+4=7$ 。
- (3) $R_2 \rightarrow LAN(S_1 \text{ 與 } S_2)$ ：要到達送方 S_1 與 S_2 ，需經過 $R_2 \rightarrow LAN$ 路徑，此路徑保留頻寬為 $b_1+b_2=3+4=7$ 。
- (4) $R_1 \rightarrow S_3$ ：要到達送方 S_3 ，需經過 $R_1 \rightarrow S_3$ 路徑，此路徑保留頻寬為 $b_3=7$ 。

因此網路各節點所保留頻寬總和為 35。

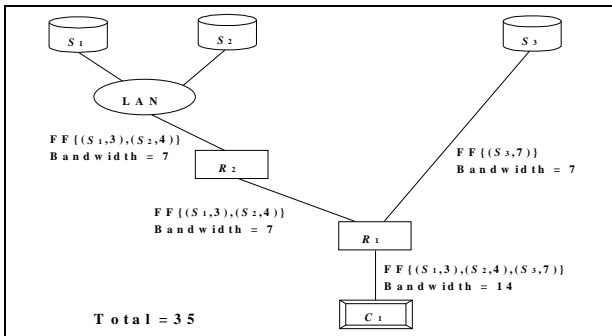


圖 6 C_1 送出 FF 保留樣式後網路各節點頻寬保留示意圖

■ SE 保留樣式 (SE $\{(S_1, S_2, \dots), Q\}$)

從真值表來看，送方串列即所有 Active Set 的聯集，而共享頻寬為 w_k 中最大者，也就是 B_T 。如圖 7 所示， C_1 所送出的需求為 SE $\{(S_1, S_2, S_3), 11\}$ 。在網路路徑上，各介面上頻寬保留情形分述如下。

- (1) $C_1 \rightarrow R_1(S_1, S_2, S_3)$ ：表示要到達送方 S_1, S_2, S_3 ，需經過 $C_1 \rightarrow R_1$ 路徑，因 C_1 送出 SE $\{(S_1, S_2, S_3), 11\}$ ，故 R_1 介面保留頻寬為 11。
- (2) $R_1 \rightarrow R_2(S_1, S_2)$ ：要到達送方 S_1, S_2 ，需經過 $R_1 \rightarrow R_2$ 路徑， R_1 送出 SE $\{(S_1, S_2), 11\}$ ， R_2 介面保留頻寬為 11。
- (3) $R_2 \rightarrow LAN(S_1 \text{ 與 } S_2)$ ：要到達送方 S_1 與 S_2 ，需經過 $R_2 \rightarrow LAN$ 路徑， R_2 送出 SE $\{(S_1, S_2), 11\}$ ，而 LAN 上保留頻寬²為 11， S_1 與 S_2 介面各保留頻寬為 11。
- (4) $R_1 \rightarrow S_3$ ：表示到達送方 S_3 ，需經過 $R_1 \rightarrow S_3$ 路徑， R_1 送出 SE $\{(S_3), 11\}$ ， S_3 介面保留頻寬為 11。

因此網路各節點所保留頻寬總和為 44。

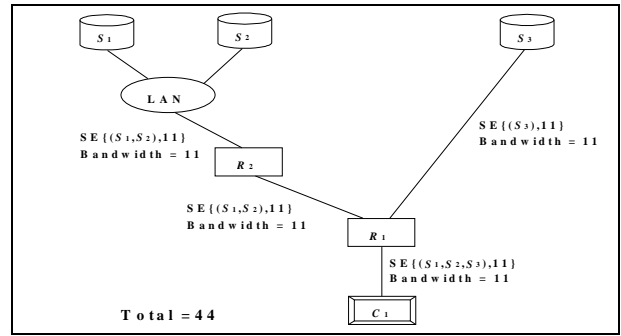


圖 7 C_1 送出 SE 保留樣式後網路各節點頻寬保留示意圖

由以上二種情況觀察到令人意外的結果，使用 SE 保留樣式比 FF 保留樣式需要更多的頻寬，似乎違背了欲用 SE 來提供資源分享的美意。事實上 SE 與 FF 何者為佳，除了取決於應用系統的通訊模式外，亦與網路的拓樸有密切的關係。一般說來，FF 因為忽略了資料流間的分流關係，所以易造成接近收方 link 上超保留；相反的，SE 因為忽略了個別資料流的需求，所以易造成接近送方 link 上的超保留。

此外，SE 保留樣式的使用，並非想像中的簡單，我們用下面的例子來說明，假設一個收方 C_2 欲從 S_4 接收資料，其需求頻寬為 $b_4=14$ (如圖 8)，其真值表如表 2。 C_1 仍沿用先前的真值表。

表 2 C_2 之真值表

	S_4	e	w
A_1	0	0	0
A_2	1	1	$b_4=14$

當使用 SE 保留樣式時，收方 C_1 仍發出需求 SE $\{(S_1, S_2, S_3), 11\}$ ，收方 C_2 則發出需求為 SE $\{(S_4), 14\}$ ，所經路徑其頻寬保留情形分述如下。

- (1) $C_1 \rightarrow R_1(S_1, S_2, S_3)$ ： C_1 發出 SE $\{(S_1, S_2, S_3), 11\}$ ，在 R_1 介面保留頻寬為 11。
- (2) $C_2 \rightarrow R_1(S_4)$ ： C_2 發出 SE $\{(S_4), 14\}$ ，在 R_1 介面保留頻寬為 14。
- (3) $R_1 \rightarrow R_2(S_1, S_2, S_4)$ ：要到達送方 S_1, S_2, S_4 ，需經過 $R_1 \rightarrow R_2$ 路徑，先前的 SE $\{(S_1, S_2, S_3), 11\}$ 與 SE $\{(S_4), 14\}$ 合併成 SE $\{(S_1, S_2, S_3, S_4), 14\}$ ，然後由 R_1 送出，在 R_2 介面上保留頻寬為 14。
- (4) $R_2 \rightarrow LAN(S_1 \text{ 與 } S_2)$ ： R_2 送出 SE $\{(S_1, S_2, S_3), 14\}$ ，LAN 與 S_1 與 S_2 分別保留頻寬為 14。
- (5) $R_2 \rightarrow S_4$ ： R_2 送出 SE $\{(S_4), 14\}$ ， S_4 介面保留頻寬為 14。
- (6) $R_1 \rightarrow S_3$ ： R_1 送出 SE $\{(S_3), 11\}$ ， S_3 介面保留頻寬為 11。

因此網路各節點所保留頻寬總和為 78。

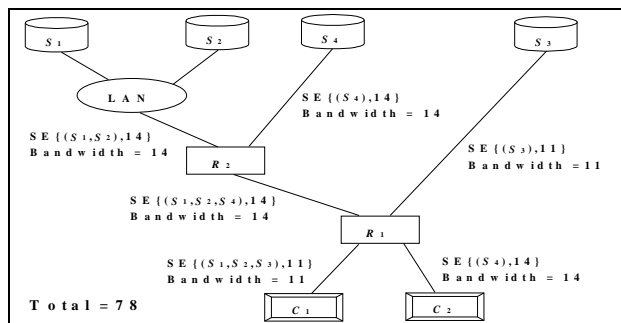


圖 8 C_1 與 C_2 送出 SE 保留樣式後網路節點頻寬保留示意圖

² 有關如何在 LAN 的環境上保留資源，目前正由 IETF issll (integrated Service over Specific Link) working group 定義中。

不幸的是，這樣的保留並不正確，因為事實上在 link $R_1 \rightarrow R_2$ 上可能同時 C_1 需要 11， C_2 需要 14，所以正確的保留最少應為 25，而非前例中所提的 14。要做出正確的 SE 需求並不單純，因為必須涉及到參考所有用戶端(收方)的真值表，但收方的加入或離開可能是隨時隨地，此意味著必須存在著一個上層管理系統，動態的集中統合所有真值表，然後做出適當的 SE 需求或調整既存的 SE，而調整 SE 的動作很可能會要求較高的頻寬，因而導致資源不足而需求被拒絕，中斷了正在通訊的連線。有鑑於此，我們將於下一章中提出一套創新的資源保留樣式—Truth Assignment，此保留樣式以分散式的方式，計算出每一連線上最小的頻寬需求，如此一來解決了頻寬超保留的問題，同時降低了因頻寬調整而被中斷的機率。

3. Truth Assignment 保留樣式

3.1 運作原理

所謂 Truth Assignment (TA)保留樣式，就是在 Resv 訊息中包含真值表，與收方個別的頻寬資訊，路由器就可依照這些訊息計算出實際所需的頻寬。同時路由器會將收到的真值表做「合併」與「刪減」的動作產生新的真值表，然後包裹成 Resv 訊息送出。我們以單一路由器為例，說明運作的細節，如圖 9 的網路環境中，有一路由器連結二個收方為 C_1 、 C_2 與三個送方分別為 S_1 、 S_2 與 S_3 ，首先我們定義路由器的訊息之流向關係，因收方訊息指向路由器，所以稱收方為其 incoming interface 或 downstream(下游)，在圖 9 中訊息 a 、 b 分別為 Resv 訊息。因路由器其訊息輸出指向三個送方，則稱送方為其 outgoing interface 或 upstream(上游)，在圖 9 中訊息 x 、 y 、 z 為資源保留要求在合併後所產生的訊息。Resv a 所表達的真值表 T_a 與 Resv b 所表達的真值表 T_b ，分別描述於表 3 與表 4 中，所需保留的頻寬分別為 $B(T_a)$ 對到 C_1 的 link 而言與 $B(T_b)$ 對到 C_2 的 link 而言。

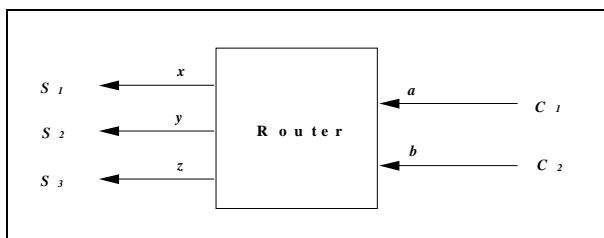


圖 9 路由器需求示意圖

表 3 真值表 T_a

S_1	S_2	e
0	0	0
0	1	1
1	0	1
1	1	0

表 4 真值表 T_b

S_2	S_3	e
0	0	0
0	1	1
1	0	1
1	1	0

為要產生新的 x 、 y 、 z 需求訊息， T_a 與 T_b 必須先行「合併」，合併後的真值表中的 Truth Assignment 就是 T_a 與 T_b 的 Truth Assignment 的所有可能「配對」，所謂的「配對」就是將兩組 Truth Assignment 的 Active Set 成員「聯集」起來，例如： T_a 中的 Active Set $\{S_1\}$ 與 T_b 中的 Active Set $\{S_3\}$ 配對結果產生新的 Active Set $\{S_1, S_3\}$ ，所對應的 Truth Assignment 為 S_1 、 S_2 的狀態為 1， S_3 的狀態為 0。合併後的真值表 $T_{a \times b}$ ，如表 5 所示。我們用“ \times ”來表示真值表合併的動作。

表 5 $T_{a \times b}$ 真值表

S_1	S_2	S_3	e
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

接著為要決定 outgoing interface 所送出的真值表，我們將 $T_{a \times b}$ 簡化成只描述必須經由該 outgoing interface 方能抵達的送方集合的真值表，也就是刪減掉不屬於此方集合的送方資訊。令此集合以 G 表示，其狀態以 $r(G)$ 表示，所謂「刪減」就是將原有真值表中具有相同 $r(G)$ 值的列合併成單獨一列，如果這些合併列中有任一者是 Truth Assignment，則此新的 $r(G)$ 也視為 Truth Assignment。我們將此運算以 $d(T, G)$ 表示，此處 T 為原有的真值表， G 為欲保留的送方集合。例如，訊息 z 內的真值表為 $d(T_{a \times b}, \{S_3\})$ ，其結果如表 6 所示。(註：全為“0”的狀態是否為 Truth Assignment 對 Truth Assignment 頻寬並無影響，為減少合併時的運算負擔，我們刻意將它設為非 Truth Assignment。)

表 6 對 S_3 做刪減後之真值表

S_3	e
1	1
0	0

有了對單一節點的認識後，以下我們以一網路為例(如圖 10 所示)，實際推演訊息的處理與傳送，並瞭解頻寬保留的情形。

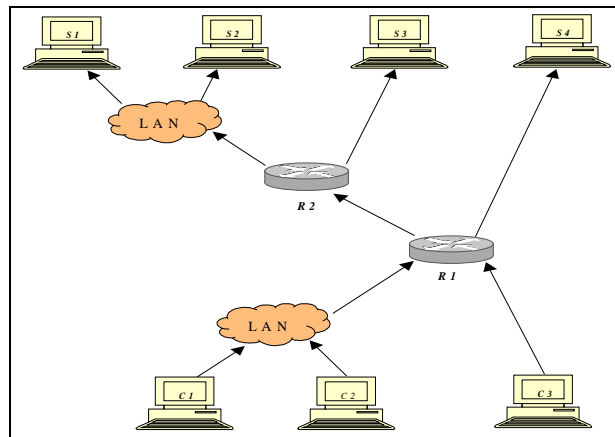


圖 10 網路架構圖

網路上一端有三個收方為 C_1 、 C_2 與 C_3 ，其需求為 C_1 ： $S_1 \oplus S_2$ ， C_2 ： $S_2 \oplus S_4$ 與 C_3 ： S_3 ；(註：“ \oplus ”表示二者擇一)；另一端有四個送方為 S_1 、 S_2 、 S_3 與 S_4 ，其頻寬分別為 $b_1=3$ 、 $b_2=4$ 、 $b_3=14$ 與 $b_4=7$ ；二個路由器分別為 R_1 與 R_2 ，推演過程如下。

1. 收方 C_1 與 C_2 的需求送抵 R_1 同一介面(左邊)，表示資源可分享，所以 R_1 會針對他們的真值表，先做合併動作。表 7 為 C_1 的真值表，表 8 為 C_2 的真值表，在表中網底的部分，表示 Truth Assignment。

表 7 收方 C_1 ： $S_1 \oplus S_2$ 需求真值表

S_1	S_2	e
0	0	0
0	1	1
1	0	1
1	1	0

表 8 收方 C_2 ： $S_2 \oplus S_4$ 需求真值表

S_2	S_4	e
0	0	0
0	1	1
1	0	1
1	1	0

利用合併後的真值表，來計算收方 C_1 與 $C_2 \rightarrow R_1$ 路徑的保留頻寬。表 7 有 2 個 Truth Assignment，表 8 有 2 個 Truth Assignment，因此會產生 $2 * 2 = 4$ 種組合。合併後，產生新的需求真值表，如表 9 所示，表中每一種組合皆為一 Truth Assignment，利用 Truth Assignment 中之 Active Set，算出每一種組合之需求頻寬 w ，從所有 w 中選出最大者，為該段路徑的保留頻寬，在本例中保留頻寬為 11(在表 9 中以下網底表示)。

表 9 收方 C_1 與 C_2 需求合併後頻寬真值表 (僅顯示 Truth Assignment)

S_1	S_2	S_4	w
0	1	1	$b_2+b_4=11$
0	1	0	$b_2=4$
1	0	1	$b_1+b_4=10$
1	1	0	$b_1+b_2=7$

2. 計算收方 $C_3 \rightarrow R_1$ 路徑的保留頻寬。收方 C_3 的需求為 S_3 ，其真值表內容說明如表 10，在表中網底的部分，表示其結果 e 為真之 Truth Assignment，該路徑上保留的頻寬為 14。

表 10 收方 C_3 ： S_3 需求真值表

S_3	e	w
0	0	0
1	1	$b_3=14$

3. 當資料流傳送到路由器 R_1 時，因其下游分別來自 C_1 、 C_2 與 C_3 ，於此將下游之需求合併成一新的真值表，其合併原則同上所述，利用表 9 與表 10 之 Truth Assignment 組合，產生一個 $4*2=8$ 種組合之真值表，其結果如表 11 所示。

表 11 送方 C_1 、 C_2 與 C_3 合併後之真值表 (僅顯示 Truth Assignment)

S_1	S_2	S_3	S_4
0	1	0	1
0	1	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

4. 以表 11 的內容，來計算路徑 $R_1 \rightarrow R_2$ 的保留頻寬。而在本段路徑上選擇了送方 S_1 、 S_2 與 S_3 為對象，經刪減處理，產生新的真值表及需求保留頻寬，接著從 w 從中選擇最大者 21 為其保留頻寬，其結果如表 12 所示。

表 12 R_1 到 R_2 路徑保留頻寬真值表 (僅顯示 Truth Assignment)

S_1	S_2	S_3	w
0	1	0	$b_2=4$
0	1	1	$b_2+b_3=18$
1	0	0	$b_1=3$
1	0	1	$b_1+b_3=17$
1	1	0	$b_1+b_2=7$
1	1	1	$b_1+b_2+b_3=21$

5. 以表 11 內容，來計算路由器 $R_2 \rightarrow \text{LAN}(S_1 \text{ 與 } S_2)$ 路徑的保留頻寬。對表 11 做刪減之處理，主要針對經過該路徑之送方為其選擇處理的對象，而在本段路徑上選擇了送方 S_1 與 S_2 為對象，經刪減處理，產生新的真值表及需求保留頻寬，再從 w 從中選擇最大者 7 為其保留頻寬，其結果如表 13 所示。

表 13 R_2 到 LAN 路徑保留頻寬真值表 (僅顯示 Truth Assignment)

S_1	S_2	w
0	1	$b_2=4$
1	0	$b_1=3$
1	1	$b_1+b_2=7$

6. 以表 11 內容，來計算路由器 $R_2 \rightarrow S_3$ 路徑的保留頻寬。對表 11 做刪減之處理，主要針對經過該路徑之送方為其選擇處理的對象，而在本段路徑上選擇了送方 S_3 ，經刪減處理，產生新的真值表及需求保留頻寬，再從 w 從中選擇最大者 14 為其保留頻寬，其結果如表 14 所示。

表 14 R_2 到 S_3 路徑保留頻寬真值表 (僅顯示 Truth Assignment)

S_3	w
0	0
1	$b_3=14$

7. 以表 11 內容，來計算路由器 $R_1 \rightarrow S_4$ 路徑的保留頻寬。對表 11 做刪減之處理，主要針對經過該路徑之送方為其選擇處理的對象，而在本段路徑上選擇了送方 S_4 ，經刪減處理，產生新的真值表及需求保留頻寬，再從 w 從中選擇最大者 7 為其保留頻寬，其結果如表 15 所示。

表 15 R_1 到 S_4 路徑保留頻寬真值表
(僅顯示 Truth Assignment)

S_4	w
0	0
1	$b_4=7$

8. 總結上述之狀況，是以 Truth Assignment 保留樣式作法，求得整體網路所需保留之頻寬總和為：
收方 C_1 與 C_2 到 R_1 路徑+收方 C_3 到 R_1 路徑+ R_1 到 R_2 路徑+ R_2 到送方 S_1 與 S_2 路徑+ R_2 到送方 S_3 路徑+ R_1 送方 S_4 路徑 = 11 + 14 + 21 + 7 + 14 + 7 = 74

如果使用 FF 的保留樣式，則總共需要之頻寬為 81(我們省略此部分的推演)。這是因為 TA 保留樣式在每一 link 上頻寬的保留，皆根據真值表來做，其中完整記錄了所有會使用到該 link 的送方關係，所以保留出來的結果亦是最佳值，而避免掉了 SE 或 FF 中過多保留或保留不夠的問題。

3.2 訊息格式與負荷分析

事實上 Truth Assignment 即代表真值表的內涵，因此我們採用傳送 Truth Assignment 的方式，來節省訊息傳送的資料量，以下我們利用 BNF 語法描述 TA 保留樣式的 Resv 訊息規格(如表 16 所示)。基本上 TA 保留樣式與 FF 保留樣式類似，只是外加了一串 Truth Assignment 的描述，我們只針對與 TA 保留樣式有關的部分(即網底部分)說明，其它部分的說明請參考[4]。

表 16 Truth Assignment 保留樣式保留訊息格式表

格式文法說明	
1	<Resv Message> ::= <Common Header> [<Integrity>] <Session> <RSVP_Hop> <Time_Value> [<Resv_Confirm>] [<Scope>] [<Policy_Data> ...] <Style> <Flow Descriptor List> <TA List>
2	<Style> ::= TA
3	<Flow Descriptor List> ::= <Flowspec> <Filter_Spec> <Flow Descriptor List> <Flow Descriptor>
4	<Flow Descriptor> ::= [<Flowspec>] <Filter_Spec>
5	<Truth Assignment List> ::= <Empty> <Truth Assignment list> <Truth Assignment>

- Style：指定保留樣式，為 Truth Assignment (TA)。
- Flow Descriptor List：是由一串 Flow Descriptor 所構成，每一個 Flow Descriptor 說明了一個送方的 ID (<Filter_Spec>)，與它所需的頻寬(<Flowspec>)。
- Truth Assignment List：是由一串 Truth Assignment 所構成，每一筆 Truth Assignment 是一個 k -bits 的 binary 碼，此處 k 代表著 Flow Descriptor List 中送方的個數，第 i 個 bit 的值代表著 Flow Descriptor List 中排行第 i 的 Flow Descriptor 所指的送方狀態。

我們採用 TA $\{(S, Q)^k, \{TA^p\}\}$ 的表示法來表示一個 TA 樣式的 Resv 訊息(k 與 p 分別為送方個數與 TA 個數)。以 3.1 節中圖 10 網路為例，在網路上一端有三個收方為 C_1 、 C_2 與 C_3 ，其需求為 $C_1: S_1 \oplus S_2$ 、 $C_2: S_2 \oplus S_4$ 與 $C_3: S_3$ ，以 Truth Assignment 保留樣式發出上述的需求，則個別所發出之 Resv 訊息為(我們只顯示與頻寬保留有關資訊)： $C_1: TA(\{(S_1, 3), (S_2, 4)\}, \{01, 10\})$ 、 $C_2: TA(\{(S_2, 4), (S_4, 7)\}, \{01, 10\})$ 、 $C_3: TA(\{(S_3, 14)\}, \{1\})$ 。

而負荷主要來自兩方面：一者為 Resv 訊息為攜帶 Truth Assignment 所需要增加的資料量；另一者為在節點介面上，為做 Truth Assignment 合併所需的運算負荷。就第一點而言，我們除了需要攜帶 FF 保留樣式的所有資訊外，必須附加上有關的 Truth Assignment 的資訊，假設訊息中共指出了 n 個送方，則每一筆 Truth Assignment 的值可以用一組 n -bits 的 binary code 來代表，在最差的情況下共有 $2^n - 1$ 個 n -bits 的 Truth Assignment (binary code) 要送出(註：不會有“00..00”的組合)，如果我們另增一個額外的 check bit 來表示 Truth Assignment List 所代表的是正面含意或反面含意，則我們可以選擇需要資料量較少的表示方式，來傳送 Truth Assignment List，如此 Truth Assignment 的個數會小於 2^{n-1} 。一般而言，收方為一般的用戶端，處理資料的能力有限，因此不會向太多的伺服器(送方)同時要求資料，所以 n 值不會太大，再者 Truth Assignment 的個數會隨接近送方端而減少(因 n 變小)，所以額外花費在傳送 Truth Assignment 的資料量應不至於過大。

就第二點而言，兩個真值表的合併，牽涉到個別 Truth Assignment 的兩兩合併，而每兩項 Truth Assignment 的合併，又牽涉到 binary code 的合併，假設一個 truth table T_a 描述了 n_a 個送方，含有 k_a 個 Truth Assignment；另一個 truth table T_b 描述了 n_b 個送方，含有 k_b 個 Truth Assignment，則兩者合併必須做 $k_a \times k_b$ 次的 Truth Assignment 合併，且每次 Truth Assignment 合併必須檢視 $n_a + n_b$ 個 bits，然後做必要的“OR”動作，所以整個 computational complexity 為 $k_a \times k_b \times (n_a + n_b)$ 。再一次的說明，如果在送方數目與 Truth Assignment 的個數不至於太多的話的情況下，此運算負荷是可以接受的。

4. 環境定義與效能分析

4.1 模擬系統架構

為了驗證 TA 保留樣式的優越性，我們模擬 TANet2 的網路環境，執行了一套分級式的遠距教學系統，觀察比較 TA 與 FF 保留樣式在「頻寬的使用率」與「上線拒絕率」上的表現。就 TANet2 所模擬的範疇而言，其中包括國家高速電算中心(NCHC)、屏東科技大學(NPUST)、中山大學(NSYSU)、成功大學(NCKU)與中正大學(CCU) (如圖 11 所示)。其中以 NCHC 的 POP 為中心點，分別連接 NSYSU、NCKU 與 CCU 的 POP 然後各校的 POP 再向下連接至各校的校園網路。在遠距教學系統中，中山大學(NSYSU)與屏東科技大學(NPUST)各建構了一間遠距教學教室，分別以 Classroom1 及 Classroom2 表示，Classroom1 提供上課老師的視訊伺服器以 S_1 表示，其產生每秒 1.5Mb/Sec 頻寬的資料量，上課老師聲音伺服器以 S_2 表示，其產生每秒 128Kb/Sec 頻寬的資料量(MPEG Layer3 Audio format)，與上課資料視訊伺服器以 S_3 表示，其亦產生每秒 1.5Mb/Sec 頻寬的資料量。Classroom2 提供的伺服器同上所述分別為 S_4 、 S_5 及 S_6 。因此伺服器系統均在 NSYSUPOP 之下。

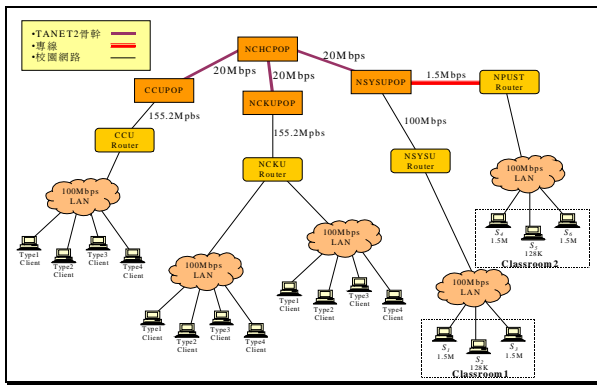


圖 11 模擬系統網路架構

在 NCKU 的 Router 之下連接著兩個 100 Mbps LAN，在 CCU 的 Router 之下連接著一個 100 Mbps LAN，學生的機器直接連上這些 LAN，收看由 NSYSU 或 NPUST 所播送出來的上課節目。將學生分成幾種不同等級(如依照付費程度)，分別為 Type 1、Type 2、Type 3 及 Type 4，各個 Type 所享有的服務不同，分述如下。
 Type 1：學生可同時收看 NSYSU (Classroom1) 與 NPUST (Classroom2) 老師上課的視訊，只聽其中一間教室的聲音，且同時可收看 Classroom1 與 Classroom2 上課資料伺服器的資料，所對應的 Truth Assignment，如表 17 所示。

表 17 Type 1 的 Truth Assignment

S_1	S_2	S_3	S_4	S_5	S_6
1	1	1	1	0	1
1	0	1	1	1	1

Type 2：學生同一時間可選擇接收 Classroom1 或 Classroom2 老師上課的視訊、教室的聲音、上課資料伺服器的資料，同時允許學生暫時切換到另一 Classroom 監聽上課內容，所對應的 Truth Assignment，如表 18 所示。

表 18 Type 2 的 Truth Assignment

S_1	S_2	S_3	S_4	S_5	S_6
1	1	1	0	0	0
1	0	1	0	1	0
0	0	0	1	1	1
0	1	0	1	0	1

Type 3：學生同一時間可選擇接收 Classroom1 或 Classroom2 老師上課的視訊、教室的聲音，同時允許學生暫時切換到另一 Classroom 監聽上課內容。但無法收看上課資料伺服器的資料，所對應的 Truth Assignment，如表 19 所示。

表 19 Type 3 的 Truth Assignment

S_1	S_2	S_3	S_4	S_5	S_6
1	1	0	0	0	0
1	0	0	0	1	0
0	0	0	1	1	0
0	1	0	1	0	0

Type 4：學生只能選擇收聽一間教室的聲音。無法收看到任何視訊，所對應的 Truth Assignment，如表 20 所示。

表 20 Type 4 的 Truth Assignment

S_1	S_2	S_3	S_4	S_5	S_6
0	1	0	0	0	0
0	0	0	0	1	0

我們假設整個系統中，學生發出收看聽節目的需求為一 Poisson process，且收看聽的時間長度為 Exponential distribution。當一需求發生時，我們假設它屬於 Type 1、Type 2、Type 3 與 Type 4 的機率分別為 5%、10%、30% 與 55%，也就是點播的機率隨類型編號增高而增高。

4.2 系統效能評估與分析

我們使用 LabVIEW (Laboratory Virtual Instrument Engineering Workbench) 軟體，來建構我們的模擬環境。我們發現頻寬的消耗量與 request arrival rate 的關係不顯著，這是因為 multicast 的特性—所有收方共享從同一送方送出的資料，所以當每個區域有少許的收方加入時，已保留路徑上所需要的資源，而趨於穩定狀態，縱然當稍後有更多的收方加入接收時，並不會造成資源需求明顯的增加。因此在下面的討論中我們將省略討論改變 request arrival rate 對系統效能的影響，而將它固定在 10 requests/hr。我們將針對 Bandwidth 及 Rejection Probability 等兩項指標進行分析與討論。

4.2.1 保留頻寬

我們模擬使用 TA 與 FF Style 來產生 Resv 訊息，並記錄每一時間點的總保留頻寬(也就是所有 link 上所保留頻寬的總和)，如圖 12 與圖 13 所示，橫軸的單位為小時，縱軸的單位為 Kb/Sec。我們先假設 link 的頻寬無限大(也就是說不會有因頻寬不夠而保留被拒絕的現象)。如我們所預期的，TA Style 所保留的頻寬總額小於 FF Style 所保留的頻寬，且 FF Style 所保留的頻寬均集中在 60Mb/Sec 左右，超保留的現象明顯；反觀 TA Style 所保留的頻寬則會依實際的需求做調整，而不會發生有超保留的現象。

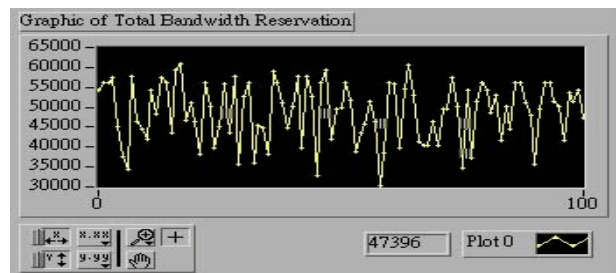


圖 12 使用 TA Style 的保留頻寬

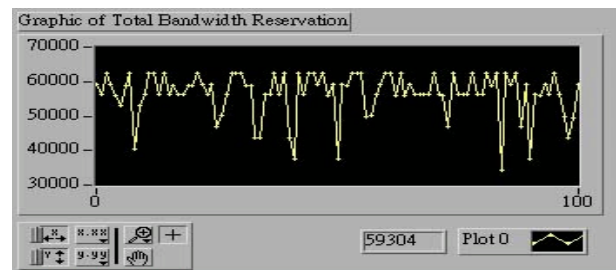


圖 13 使用 FF Style 的保留頻寬

我們收集 100 個小時內的頻寬使用狀況，求得 FF Style 所需的平均頻寬大小為 57093 Kb/Sec，TA Style 所需的平均頻寬大小為 49363 Kb/Sec，使用 TA Style 比 FF Style 節省了 13.5% 的頻寬。

4.2.2 拒絕機率

當使用者的需求超過實體網路所能供應的頻寬時，就會發生拒絕(rejection)的現象，一個理想的頻寬保留機制應當盡可能的降低此機率值。我們想瞭解的是，相較於 FF Style，TA Style 能帶來多顯著的改善。我們假設 TAnet 2 backbone 的 POP 與 POP 間只保留 5Mb/Sec 的頻寬供應用系統使用，且 request arrival rate 為 10 requests/hr，所觀察到的 Rejection Probability 在 TA Style 的環境中為 0.47，而在 FF Style 的環境中為 0.73。大約改善了 35.6%。由此再一次說明 TA Style 能提供較 FF Style 更精準的資源保留，而有效提升系統的使用率。

5. 結論與未來的研究

針對 RSVP 的三種保留樣式 FF、SE 和 WF，經實例推演後，似乎沒有一定的規則可推得三種保留樣式中何者最佳，且皆出現超保留網路資源的現象，這主要是由於此三種保留樣式未能描述出完整的資源共享關係，而導致網路無法做精確的資源保留。針對此缺點我們提出一套嚴謹而可行的保留樣式— Truth Assignment 保留樣式，來達到保留最小頻寬的目的。

基本上我們傳送描述應用系統的通訊模式 (Communication Paradigm) 的真值表 (即 Truth Assignment)，每個網路節點對收到的真值表做「合併」處理，然後計算出必須保留的最小頻寬，接著做「刪減」處理，產生簡化的真值表傳給其它上游節點。我們使用模擬的方式來驗證我們所提演算法的效能，以 TAnet2 部分群播網路環境為我們的實驗網路，模擬學生分級選修遠距教學課程之情境，經比較模擬結果後，證明了 TA 保留樣式之優越性。經過分析，使用此種保留樣式在網路資源使用效能上的獲益，與所必須付出的通訊與計算代價上相比，是值得的。

參考文獻

- [1] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and Weiss, W., "An Architecture for Differentiated Services", RFC 2475, December 1998.
- [2] Braden, R., Clark, D., and Shenker, S., "Integrated Services in the Internet Architecture: an Overview", RFC 1633, June 1994.
- [3] Braden, R. and Zhang, Ed., "Resource ReSerVation Protocol (RSVP) — Version 1 Message Processing Rules", RFC 2209, September 1997.
- [4] Braden, R., Zhang, L., Berson, S., Herzog, S., and Jamin, S., "Resource ReSerVation Protocol (RSVP) — Version 1 Functional Specification", RFC 2205, September 1997.
- [5] Chris, M., "RSVP: General-Purpose Signaling for IP", IEEE Internet Computing, May/June 1999.
- [6] Pan, P., Schulzrinne, H., and Guerin, R., "Staged Refresh Timers for RSVP", Internet Draft,

- November 1997.
- [7] Waitzman, D., Partridge, C., and Deering, S., "Distance Vector Multicast Routing Protocol", Internet RFC 1075, November 1988.
- [8] White, P., "RSVP and Integrated Services in the Internet - A Tutorial", IEEE Communications Magazine, May 1997, Vol 35, Iss 5, pp 100-106.
- [9] Wroclawski, J., "The Use of RSVP with IETF Integrated Services", RFC 2210, September 1997.
- [10] Zhang, L., Deering, S., Estrin, D., Shenker, S., and Zappala, D., "RSVP: a new Resource ReSerVation Protocol", IEEE Network, September 1993.