

群組回饋訊息的快速收斂 Fast Convergence of Group Feedback

陳鈺貞
Yue-Jean Chen

元智大學資訊工程研究所
Dept. of Computer Engineering & Science
University of Yuan-Ze
helen@ms.tl.gov.tw

鍾添曜
David Chung

元智大學資訊工程研究所
Dept. of Computer Engineering & Science
University of Yuan-Ze
csdchung@cs.yzit.edu.tw

摘 要

本文提出一個 RTP/RTCP 群組回饋訊息快速收斂的方法，謂之 RTCP 伺服器。RTCP 伺服器是由 RTCP 過濾器、模糊彙總器和重思考排程器等三部份組成。RTCP 伺服器使得大型群組，即使在很多組員同時加入的情況下，仍然能有效保持 RTCP 回饋訊息對服務品質現況的及時反應能力。

This paper proposes a fast convergence approach for RTP/RTCP group feedbacks, called RTCP Server. RTCP Server consists RTCP Filter, Fuzzy Summarizer and Reconsideration Scheduler. The RTCP Server enables a large group to efficiently react on the new QoS reported by the group members, even under the situation that lots of members join the group simultaneously.

關鍵字：RTP/及時傳輸協定，Multimedia/多媒體，feedback control/回饋控制，QoS/服務品質

1、簡介

多媒體應用近年來發展迅速，隨著市場之需求，音、視訊多媒體服務已逐漸多元化。配合著資訊網路技術的進步，愈來愈多的多媒體應用系統在網際網路上出現，如視訊會議、視訊隨選、遠距教學等。但多媒體通訊與一般數據通訊，在網路服務品質的需求上，有其不同之處。在檔案傳送、電子郵件等數據通訊應用中，其資訊的傳輸沒有及時性的要求。當頻寬足夠時，其傳送速度快；當頻寬不足時，延遲一段時間亦可。此類通信服務要求的是絕對的正確性；所使用之錯誤控制方法為dynamic window機制[1]，至於資訊傳送延遲時差的多寡，並不會影響目的端對資料的使用性。而多媒體通訊應用因必須顧慮影、音、視訊媒體之及時性、同步性與多工等問題，所以必須要求(1)足夠的通信頻寬(2)較小的jitters(3)較少的資料流失量(4)有效的多播方式(multicast)等的網路服務品質。

為提供及時性的多媒體通訊服務，在過去已有許多通訊協定被制訂出來，主要的發展有二個領域(1)多播路由協定；包括IGMP、DVMRP、MOSPF、PIM和CBT等協定[10]，(2)及時通訊協定；包括RSVP[11]、RTP[3]等協定。當然，對影、音、視訊等

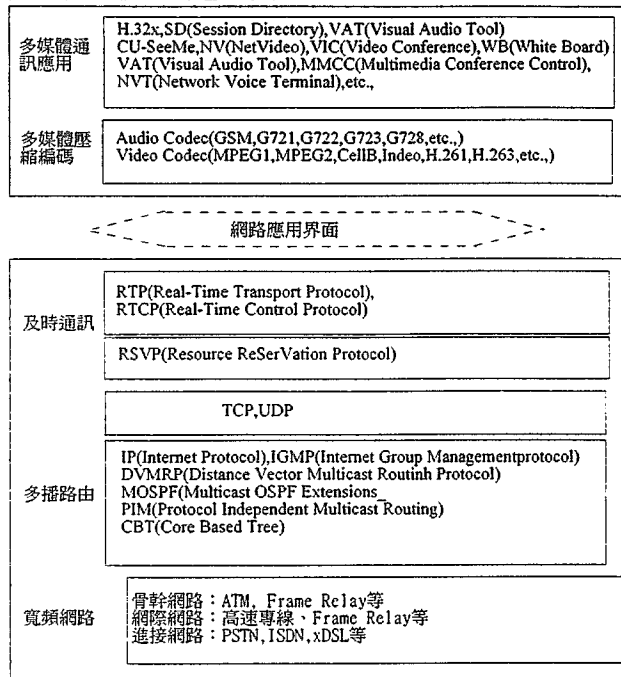
佔用較大資訊容量的媒體，為求資料儲存空間與傳輸時間的精簡，這些訊號的壓縮編碼技術是不可缺少的。透過這些相關技術的結合，使及時性的多媒體應用服務，能以最少的資源，服務更多的參與者。網際網路的資源是有限的，如何在多媒體通訊應用不斷成長的趨勢下，仍然能滿足網路服務品質的要求，是各界努力研究的重點。圖一所示是目前網際網路中，可提供及時性多媒體通訊服務的通訊協定架構[2]。

本文所要探討的領域，著重在 RTP(Real-Time Transport Protocol)及時通訊協定的回饋訊息控制機制[3]。RTP 是一個終端對終端的網路傳輸協定，適用於傳送具有及時性要求的資料，如語音、影視訊等。RTP 雖名為及時傳輸通訊協定，卻沒有連線的機制，而是依附在下層通訊協定所提供的連線機制上運作。這些連線機制可以是 TCP(Transmission Control Protocol)或 UDP(User Datagram Protocol)。由於考慮音、視訊的及時性，目前大部份的應用是以 UDP 為主。RTP 也是利用下層通訊協定所提供的多播(multicast)或單播(unicast)服務來傳送封包，其中以多播服務較常被利用。另外，RTP 也不支援資源預留和服務品質保證的功能，但可以透過其它的協定來支持；如 RSVP 等。所以，RTP 大多被視為應用層的一部份。

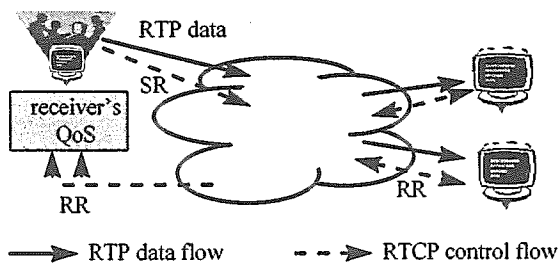
RTP 有一個伴隨的控制用通訊協定，稱為 RTCP(Real Time Control Protocol)[3]。RTCP 有多種控制封包，群組組員利用其中的 Sender Report(SR)和 Receiver Report(RR)二種控制封包，向所有組員報告 RTP 資料封包之傳送或接收的情形等，同時兼具有評估群組規模的用途。RTP 資料的原始端(source node)可以透過這些控制封包，瞭解所有目的端的接收品質，進而做出適當的傳送參數調整。所以，每位組員均定期地產出 RTCP 控制封包，並多播給所有組員。RTP 與 RTCP 的關係如圖二所示。

RTCP 能使用的頻寬，被限制在 RTP 頻寬的 5% 以內，亦即所有組員共同分享這些頻寬。因此，大家必須遵循一定的法則來使用 RTCP 頻寬，使 RTCP 控制封包可以順利到達各組員處。其方法是，在每個組員送出 RTCP 控制封包的同時，就計算下次送出控制封包的合理時間，以避免大家集中在某段時間內送出，導致交通量暴增，而影響 RTP 資料封包的傳送品

質，甚至發生封包流失。



圖一 多媒體通訊服務之通訊協定架構



圖二 RTP/RTCP關係示意圖

由於無法預先知道群組的大小；即組員人數，所以無從計算出精確的報告間距(report interval)。Schulzrinne 等人所提出之合理的報告時間計算法則 [3]，是在時間 t_s 送出控制封包時，立即以當時所認知的群組規模，來計算下次合理的報告時間 t_n 。此法則忽略了在 t_s 和 t_n 二個時間點之間，群組規模的變動性；尤其是當多人同時加入群組的情況，將使得前面計算出來的報告時間 t_n 顯得不合理。如果還是按照原訂的時間 t_n 進行封包報告，將無可避免地發生壅塞現象。有人因此提出了重思考(Reconsideration)的改進方法[5]。此法雖然改善了因多人同時的加入而可能產生的壅塞現象，卻會引發另一個問題，就是應送出的控制封包因不斷地重思考，而延遲了頗長的一段時間才送出。這使得大型群組很難獲致 RTCP 應有的及時功效。

因此，本文從其他角度來思考問題：每個組員的 RTCP 控制封包都有前向轉送(forward)廣域網路的必要嗎？實質地減少 RTCP 控制封包的數量，而不減損它原有之功能的可能性如何？本文為此設計了一個 RTCP 伺服器，使得在群組規模不斷地成長的情況下，甚或瞬間的劇烈成長時，仍能有效保持服務品質

現況的及時反應能力。進而促使原始端依據回饋的品質報告做出最佳的調適，使具有及時特性的多媒體資訊，儘可能迅速確實地送給每位組員，以達到利用 RTCP 監督 QoS 之目的。第 2 節概述相關的研究工作，第 3 節說明我們所提出可使回饋訊息快速收斂的 RTCP 伺服器，第 4 節是我們所做的一些實驗及其結果比較，第 5 節是結論與未來展望。

2、相關研究工作

RTCP 通信協定有 4 個目的：(1)執行 QoS 監督和 RTCP 封包流量控管(2)傳遞媒體間同步(intermedia synchronization)的相關資訊(3)提供獨立評估群組大小的能力(4)傳遞組員身份識別的相關資訊。其中以第(1)項為主要目的，也是本文討論的重點所在。以下概述一些學者在回饋訊息控制機制方面的相關研究。

以 Bolot 等人為視訊資料所提出的回饋控制機制而言[6]，其特色是以機率探測(probabilistic polling)的方式進行。原始端每次只能要求群組中的部份組員送出回饋訊息，然後原始端再依據所收到的訊息數量和探測的回合數(rounds)，利用統計理論來評估群組的大小。不過 RTCP 的特性是定期自動地送出回饋訊息，除非關掉 RTCP 本身的報告功能[4]，否則此探測機制不適用於 RTCP。

在 Kanakia 等人所提的回饋控制機制[8]，則是要求交換機將其緩衝區使用情況(buffer occupancies)和服務速率(service rate)傳送給原始端，但是這些訊息的傳送並非以多播方式設計，所以也不適於 RTCP 的多播環境特性。

另外，Aboba 也提出多項改善 RTCP 的可能方案[4]，其中除了 TTL scoping/summary messages 和 RTP translators 方案外，幾乎都是以加強個別終端的自我學習能力或以分流分治方式，來著手改進 RTCP 的控制機能。其中的重可靠方法祇對 RTCP 控制封包的爆量(burst)，具有扁平化的分散效果，但在網路資源的消耗量方面沒有任何改進。尤其是在大型群組的環境下，所得到的時間分散效果，卻使某些 RTCP 控制封包被延遲了頗長的一段時間後，才被送出去。從 Rosenberg 所提之重思考機制的模擬實驗中[5]，可明顯的看出這種報告延遲的現象。因此也難以符合他所提出之 4 個認定 RTCP 改善方案是否有效的評估條件：

- (1)能減少報告間距(reporting interval)趨於穩態所需的時間，也就是能儘早確認群組的規模。
- (2)能減少路由器為 RTCP 封包在 forwarding table 上所花費的處理時間。
- (3)能增進 RR 報告內容的豐富性。
- (4)能機動調整 RTCP 頻率的佔用比例。

顯而易見地，如果只是重思考而不輔以其他方法，重思考機制也不是大型群組理想的回饋控制機制。表一是各種機制在 Aboba 的評估條件下的比較。

表一 各種回饋控制機制之比較

Comparison criterion	Decrease in convergence time	Decrease in forwarding table entries	Improvement in receiver reporting information	Ability to adjust RTCP bandwidth fraction
Scaling alternative				
Probabilistic polling	Yes	Yes	No	No
Separate multicast group for RTP and RTCP	No	Yes	No	No
Separate multicast group for RR and SR	No	Yes	No	No
UnICASTING of RR	No	Yes	No	No
TTL scoping and summary messages	Yes-	Yes	Yes	No
Reconsideration	No	No	No	No
RTCP Server	Yes	Yes	Yes	No

3、快速收斂之回饋訊息控制架構

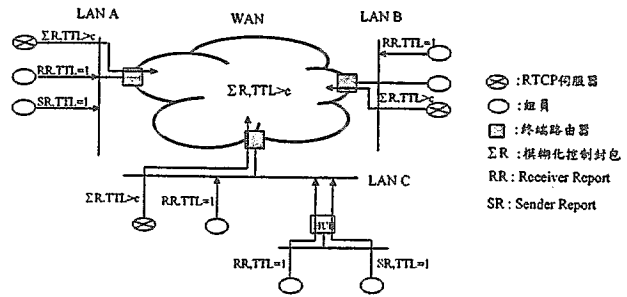
從上節的比較中，我們發現要降低群組規模對回饋控制效能的影響，必須設法使 RTCP 控制封包量的增加速度非正比於群組的成長。我們的想法是，因為次網路具有廣播(broadcast)的特性，所有位於同一個次網路的終端，除了因個別的資料處理能力所造成的差異外，每個終端所得到的網路服務品質應該是一致的。所以將一個次網路作為反應服務品質的最小單位應屬合理，進而有效地抑制 RTCP 回饋訊息的數量的增加。

我們的作法是，在每個次網路上設置一個 RTCP 伺服器，此伺服器可以是次網路上的一台工作站，也可以建置在終端路由器(leaf router)上。但後者的架構較為複雜，且必須和多播通訊協定結合，所以不在本文探討範圍內。RTCP 伺服器的工作，是收集次網路上各組員所送出的 RTCP RR/SR 控制封包，然後交由 RTCP 伺服器內的模糊彙總器(fuzzy summarizer)作模糊集合運算，得到一個模糊化的網路服務品質後，再利用重思考觀念[5]，為這些模糊化 RTCP 控制封包作傳送排程。當然，模糊化的 RTCP 控制封包必須具有宏觀的表達能力，能綜合表達出該次網路上所有組員的網路服務品質。

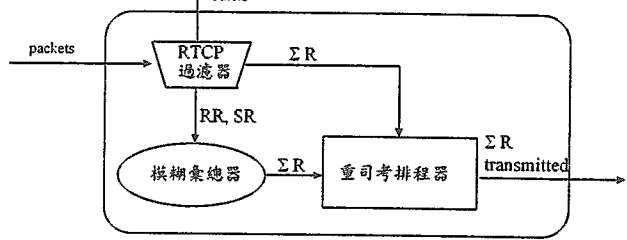
在 RTCP 伺服器機制下，組員產生的 RTCP RR/SR 控制封包，將被同一次網路上的 RTCP 伺服器消化，而無須再前向轉送，因此控制封包的 TTL 必須設為 1。而 RTCP 伺服器產生之模糊化控制封包(ΣR)，因必須前向轉送給其他所有組員，所以這些模糊化控制封包的 TTL 必須大於一定範圍。如此可避免不必要的控制封包在廣域網路上流竄。RTCP 伺服器的控制機制如圖三所示。

3.1 RTCP 伺服器

RTCP 伺服器包括 RTCP 過濾器(RTCP Filter)、模糊彙總器(Fuzzy Summarizer)及重思考排程器(Reconsideration Scheduler)三部分，其內部架構如圖四所示。



圖三 RTCP伺服器控制機制



圖四 RTCP伺服器組成

(1) RTCP 過濾器：RTCP 伺服器對不同的 RTCP 封包有不同的處理方法。所有屬於同一次網路之組員送出來的 RTCP RR/SR 控制封包，必須經過模糊彙總器的運算，產生模糊化控制封包後，再前向轉送到廣域網路。至於外來的 ΣR 控制封包可用來概估群組規模，作為重思考排程器計算報告間距的重要參考數據。

(2) 模糊彙總器：QoS 監督是 RTCP 的目的之一。所以原始端在它送出的 SR 上，載有已送出之 RTP 資料封包數和位元組數，作為目的端計算資料流失率和 jitter 等 QoS 的依據。目的端再將此 QoS 載於 RR 中送給其他組員。由於原始端無法顧及每個終端的異質環境來做調適，以免陷入顧此失彼的窘境；例如以最差者為調適的標準。所以原始端對目的端最近的 QoS 報告所作的調適反應，必定是概略性和多數決的。因此如果以減速(slow-down)、適速(loaded)或加速(speed-up)來表達目的端次網路的 QoS，應可幫助原始端做出適當的決定。尤其在大群組的環境下，單純化的狀態報告更有助於原始端在調適幅度上的決策。因此，如何客觀的表達出次網路的服務品質，是模糊彙總器的主要工作，我們將於下一章節中再詳述。

(3) 重思考排程器：由於 RR/SR 只送到 RTCP 伺服器，不會前向轉送到廣域網路。所以，重思考排程器在計算報告間距時所參考的群組大小，是接收的 ΣR 控制封包的數量。

當 RTCP 伺服器在時間 t 送出模糊化控制封包後，應立即依據下列之間距計算法則(A)[5]，來計算下次發送模糊化控制封包的合理時間 t_n ：

$$t_n = t_{n-1} + R(0.5, 1.5) * \max(T_{min}, CR * L(t)) \quad (A)$$

t_{n-1} ：前一個模糊化控制封包的送出時間，對初始封包的 t_0 而言， $t_{n-1} = 0$ 。

$R()$ ：為避免因同步發送可能導致的震盪效應(oscillation)，利用此隨機均勻分配(uniform

distribution)函數產生亂數，作為分散的比重，亂數的分配區間在 0.5 ~ 1.5 之間[3]。

T_{min} ：最小的報告間距，初始封包的最小間距為 2.5 秒，以後的封包則是 5 秒。

CR：控制封包的間隔間距(interarrival time)， $CR = RTCP$ 封包大小 / (RTP 頻寬 * 5%)。

$L(t)$ ：在時間 t 所認知到的群組大小。

重思考機制的主要精神是，當時間 t_n 到達時，再以目前所認知的群組大小，重新計算距離上一個控制封包的合理報告時間 t_{new} ，並比較 t_{new} 和 t_n 。若新時間 t_{new} 在時間 t_n 之前，則模糊化控制封包應立即送出。若新時間 t_{new} 在時間 t_n 之後，則模糊化控制封包不被送出而是繼續等待。一直到新時間 t_{new} 抵達時再作重思考。如此反覆直到模糊化控制封包送出為止，才算完成一個控制封包的傳送。

3.2 模糊化的回饋訊息

RTP 原始端送出的 SR，載有 RTP 資料封包的傳送開始時間標籤(timestamps)和已送之 RTP 資料封包數及位元組數的訊息，使目的端可藉以評估自己的資料流失率和 jitter 值[3]，然後在目的端送出的 RR 上報告出來。RTCP 伺服器會為所連結之次網路上的每位組員建立一個記錄，記載每位組員的前一個 RR 所報告的流失率和 jitter 值，作為模糊化程序的運算基準。

在作模糊化運算[9]之前，必須先決定模糊變數，也就是可能影響結果的參數，再為這些變數訂出對應的變數值及變數範圍，這樣就能為這些變數值賦予適當的表示，也就是以模糊歸屬度(Fuzzy Degree)來表示，並形成各變數值的模糊集合。我們先定義下列的模糊變數：

L：控制封包之資料流失率與次網路總體資料流失率之差異值。

J：控制封包之 jitter 與次網路總體 jitter 之差異值。

S：次網路的 QoS 報告。

然後定義模糊變數值：

L_i ：表示資料流失率的差異是“加劇”的模糊集合。

L_k ：表示資料流失率的差異是“不變”的模糊集合。

L_d ：表示資料流失率的差異是“改善”的模糊集合。

J_i ：表示 jitter 是“延長”的模糊集合。

J_k ：表示 jitter 是“持穩”的模糊集合。

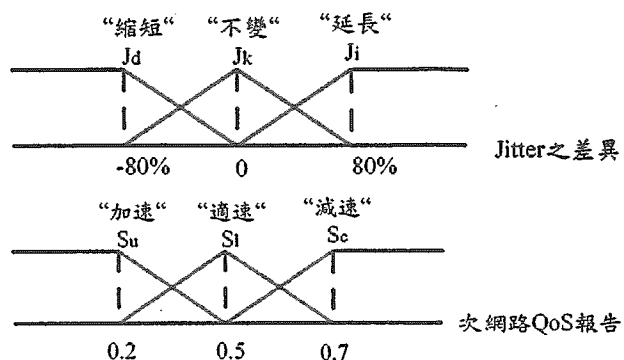
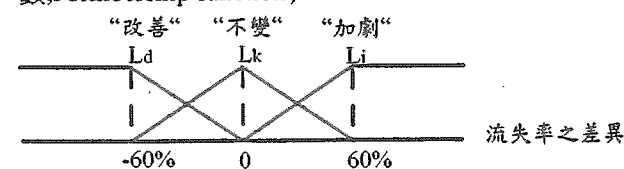
J_d ：表示 jitter 是“縮短”的模糊集合。

S_c ：表示次網路的 QoS 報告是“減速”的模糊集合。

S_l ：表示次網路的 QoS 報告是“適速”的模糊集合。

S_u ：表示次網路的 QoS 報告是“加速”的模糊集合。

再訂定各變數值的模糊集合(即歸屬函數;Membership function)：



當資料流失率和 jitter 都有改善時，可以視為網路是輕負荷的前兆。若僅一項有改善而另一項沒有變動時，表示目前狀況無法判讀，仍有待觀察，應保持不變。如果流失率稍有增加，應視為壅塞的前兆，必須立即反應出來。換句話說，對“減速”狀態的判定持較嚴謹的態度。根據上述可得到下列的模糊推論規則：

規則 1：若 L 是 L_i 且 J 是 J_i 則 S 是 S_c 。

規則 2：若 L 是 L_i 且 J 是 J_k 則 S 是 S_c 。

規則 3：若 L 是 L_i 且 J 是 J_d 則 S 是 S_c 。

規則 4：若 L 是 L_k 且 J 是 J_i 則 S 是 S_c 。

規則 5：若 L 是 L_k 且 J 是 J_k 則 S 是 S_l 。

規則 6：若 L 是 L_k 且 J 是 J_d 則 S 是保持不變。

規則 7：若 L 是 L_d 且 J 是 J_i 則 S 是 S_c 。

規則 8：若 L 是 L_d 且 J 是 J_k 則 S 是 S_l 。

規則 9：若 L 是 L_d 且 J 是 J_d 則 S 是 S_u 。

模糊彙總器利用這些推論規則，來推論彙總器目前的狀態和新近所收到之資料流失率和 jitter 值，以得到模糊化的次網路 QoS 報告，重思考排程器會在適當時間，將最新的模糊化次網路 QoS 報告傳送出去。

4、模擬實驗

為了驗證 RTCP 伺服器控制機制，我們執行了多個模擬實驗，為我們提出的 RTCP 伺服器機制和重思考機制[5]作對比模擬實驗。實驗的項目包括封包傳送速率和群組規模的認知學習能力等二項重要的效能測度對比，與控制封包的頻寬使用量比較。其中封包傳送速率是以全體組員每秒送進廣域網路的累積封包數量來表示。認知學習能力則是以在時間 t_i 所認知到的總人數來表示。

我們所假設的模擬實驗環境是，RTP 的頻寬為 746Kbps，網路的延遲時間為 0 ~ 600ms，我們同樣利用隨機均勻分配法則，為每個封包產生此項自然延遲時間。RTCP 封包的平均長度定為 1KB。模擬實驗是以群組規模為 10,000 人的樣本進行，並假設這些人在同一時間加入群組，以觀察兩種機制對大型群組的效能反應。至於 RTCP 伺服器所需的次網路組員分佈情形，也是以隨機均勻分配的方式來分派人數。

圖五和圖六是當 10,000 人同時於時間 $t=0$ 加入群組時，二種機制對所有初始封包所表現之處理效能比較。其中 RTCP 伺服器的效能表現，係以 10 人為次網路上組員人數的上限所得到的觀察。經過隨機均

勻分配的結果，10,000 個組員是分佈在 1759 個次網路內。亦即在 RTCP 伺服器機制下，雖然 10,000 個初始控制封包會在 1759 個次網路內被送出，但隨之被 1759 個 RTCP 伺服器吸收並彙總，最後只有 1759 個模糊化的初始控制封包會前向轉送到廣域網路。為表現出全程的變化，這些圖示都是以 log-log 的度量表示。

圖五是以累積的控制封包數量來比較兩種機制的傳送速度。RTCP 伺服器在第 560 秒就完成整個群組模糊化初始控制封包的傳送，而重思考機制則是等到第 3211 秒才完成初始控制封包的傳送。而且在單位時間內，RTCP 伺服器機制送出去的封包數也比重思考機制小，降低了網路壅塞的機率。如果嗣後的加入者，都位於初始的 1759 個次網路內，那麼這些新加入者絲毫不會降低 RTCP 伺服器機制的效能表現。這是因為模糊化控制封包的數量，不會因組員的增加而增多。但是對重思考機制而言，表示此後將有一段期間，每個組員會因這些新加入者，而忙著調整合理的送出時間，因為控制封包的數量與人數是成正比關係。

圖六是以群組規模認知速度的曲線。圖中顯示在 1.5 秒時，RTCP 伺服器機制就認知了 164 個加入者（這 164 個加入者是分佈在 31 個次網路上），而重思考機制只認知了 79 個加入者。

另外，我們也作了每個次網路最多 20 人、10 人、5 人和 2 人等數種不同人數分佈密度的模擬，以探討不同的人數分佈密度對 RTCP 伺服器的效能影響，模擬的觀察結果如表二所示。從表中可以明顯看出 RTCP 伺服器機制，在高人數分佈密度下的優勢。即使是低人數分佈密度，如 2 人，RTCP 伺服器機制所需的完成時間，仍然比重思考機制（分佈密度 1 人）來得短，而且封包數量也比重思考機制減少了近半數。證明 RTCP 伺服器機制至少符合了 Aboba 所提的 4 個評估條件中的前二個[4]。圖七則是以累積的控制封包數量來觀察不同密度分佈的傳送速度。

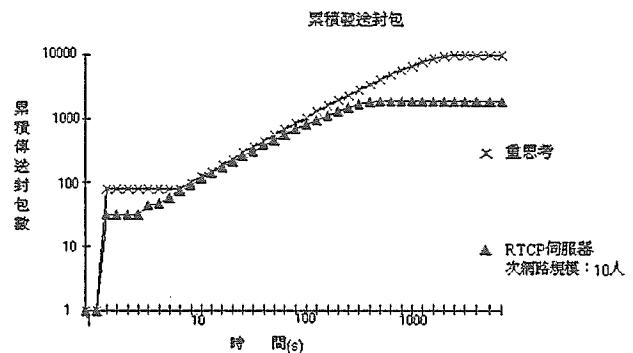
表二 不同人數分佈密度之效能比較

分佈密度	初始封包總數	完成封包傳送所需時間
20 人	963 個	303 秒
10 人	1759 個	560 秒
5 人	3281 個	1047 秒
2 人	6663 個	2508 秒
1 人	10000 個	3211 秒

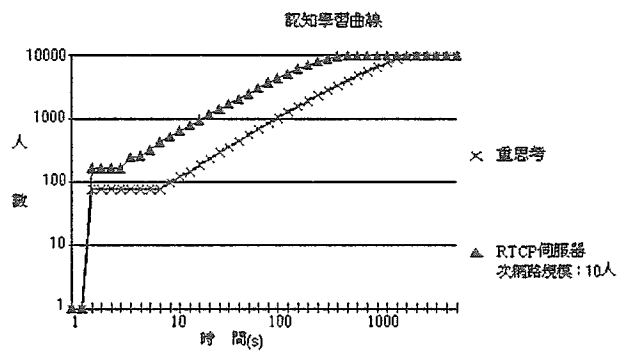
我們也模擬觀察不同機制對頻寬的使用量，此模擬增加了只作彙總卻不重思考之機制的觀察，而且僅統計前 3 個報告的使用量，其結果如圖八、九、十所示。這三個機制的頻寬使用都超過理想的使用量；每秒 5 個封包。從圖形觀之，圖九的效果似乎最好，

其實不然。因為非重思考彙總器機制的前 4 秒有每秒 735 個封包流出的情形，這個量幾乎是整個 RTP 的頻寬，將使得多數封包遺失，認知群組大小的時間勢必拉長，因此完成 3 個控制封包傳送的時間就非圖九所示的 1019 秒而已。重思考機制的前 4 秒則有每秒 100 個封包流出，RTCP 伺服器機制的前 4 秒則只有每秒 40 個封包流出。至於完成 3 個控制封包傳送的時間，重思考機制需時 9097 秒，RTCP 伺服器機制則要 1537 秒。

綜合上述，不論是在流量控制或縮短回饋訊息收斂的時間上，RTCP 伺服器都有其可取的優勢。



圖五 初始控制封包的傳送速度(群組規模10,000人)



圖六 群組規模之認知學習能力(群組規模:10,000人)

5、結論與未來展望

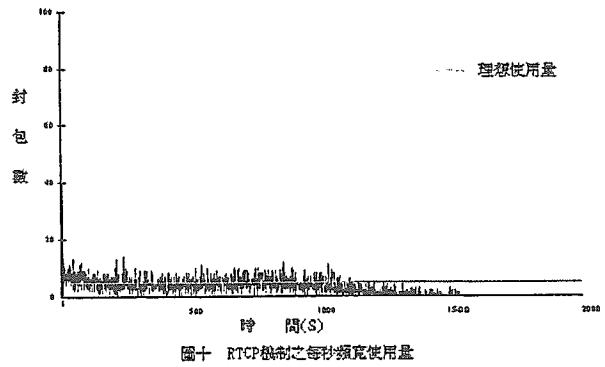
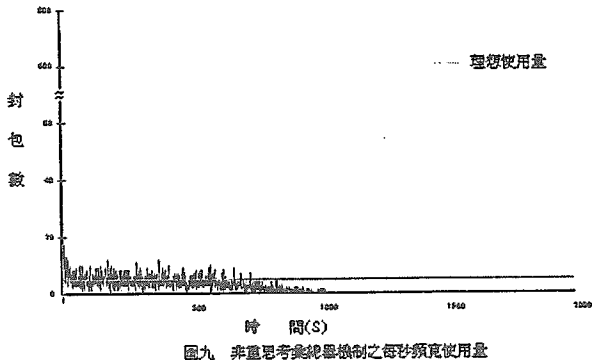
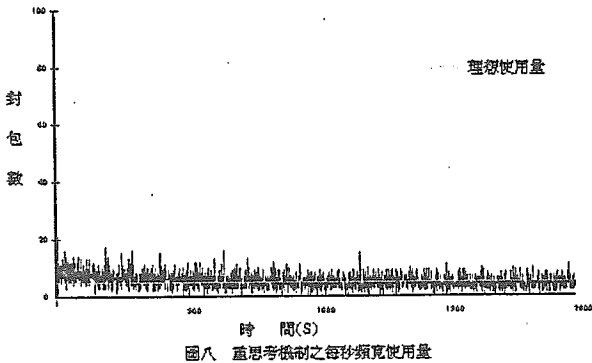
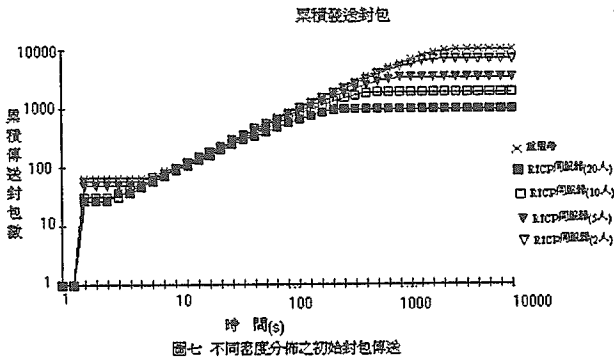
RTP 是可載送媒體資訊的及時通訊協定，並利用它的控制用通訊協定 RTCP，來反應終端的接收品質，作為原始端調適傳送參數的參考。在適當的群組規模下，一人一個反應報告，可以讓原始端很清楚的瞭解整個群組的狀況，並做出最佳的調適。一旦群組規模擴展到某種程度時，如果仍然是一人一個反應報告的作法，最後的結果不是網路因大量的報告而導致壅塞，就是因延遲報告的送出時間，使報告無法及時到達原始端，而影響調適的效果。

改善這些現象，唯一的手段就是為報告數量作有效的減產，而減產的最佳方案是以次網路為對象。因為次網路具有廣播環境的特性，只要一家收到資料，其他人絕對也有一份。所以本文結合了有效減產（報告彙總）和延遲報告時間（重思考）兩項觀念的優點，提出回饋訊息快速收斂的控制機制—RTCP 伺服器。使大型群組也可以快速地得到網路服務品質的回饋報告，並做出適當的調適。這種優勢在第 4 節的模擬

實驗中可明顯看出，而且越高密度的次網路規模，所獲得的效果越好。

我們未來的發展有兩個重點，其一是對模糊彙總器作進一步的學理探討，以驗證我們訂定之歸屬函數的合理性和推論規則的通用性。其二是將模糊彙總器所產生的模糊化服務品質報告，應用到多媒體系統的流量控制上，驗證 RTCP 伺服器的實用性。

另外還有一個遠程的展望是，希望也能應用此機制所產生的模糊化 QoS 回饋報告，結合資源預約通訊協定 RSVP，以改進組員真正獲得調適的等待時間。由於 RTCP 的 QoS 反應必須到達原始端後，原始端才會有相對的改善措施出現，然後再反向沿路反射到各個組員身上，這樣一個過程似乎稍嫌冗長。也許我們可以利用 RSVP 的資源重協調(re-negotiation)功能，在回饋報告送給原始端的沿路上，就通知所經過的 RSVP 先進行資源預約的調整，使組員可在最短時間內獲得某種程度的改善。



參考文獻

- [1] V. Jacobson, "Congestion avoidance and control," *ACM Computer Communication Review*, vol. 18, pp. 314-329, Aug. 1988. Proceedings of the Sigcomm '88 Symposium in Stanford, CA, August, 1988.
- [2] 朱榮華, "網際網路之多媒體通信技術研究," 中華電信研究所。
- [3] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time application." Request for Comments(Proposed Standard) RFC 1889, Internet Engineering Task Force, Jan. 1996.
- [4] B. Aboba, "Alternatives for Enhancing RTCP scalability," Internet Draft, Internet Engineering Task Force, Jan. 1997. Work in progress.
- [5] J. Rosenberg, H. Schulzrinne, "Scaling Feedback to Very Large Groups," tech. rep., Withheld for Anonymity, Feb. 1997.
- [6] J.-C. Bolot, T. Turetti, and I. Wakeman, "Scalable Feedback Control for Multicast Video Distribution in the Internet," in *SIGCOMM Symposium on Communications Architectures and Protocols*, (London, England), pp.58-67, ACM, Aug. 1994.
- [7] I. Busse, B. Deffner, H. Schulzrinne, "Dynamic QoS Control of Multimedia Applications based on RTP," May 1995
- [8] H. Kanakia, P. Mishra, and A. Reibman, "An Adaptive Congestion Control Scheme for Real-Time Packet Video Transport," in *SIGCOMM Symposium on Communications Architectures and Protocols*, (San Francisco, California), pp.20-31, ACM/IEEE, Sep. 1993.
- [9] H.-J. Zimmermann, "Fuzzy Set Theory and Its Applications," Kluwer Academic Publisher, 1991.
- [10] T. Maufer, C. Semeria, "Internet Draft: Introduction to IP Multicast Routing," Mar. 1997.
- [11] L. Zhang, "RSVP: A New Resource ReSerVation Protocol," *IEEE Network Magazine*, vol. 7, pp. 8-18, Sep./Oct. 1993.