

有助於提高服務品質的前瞻式封包排程機制

QoS-Aware Packet Scheduling by Looking Ahead Approach

連耀南 溫永全

國立政治大學資訊科學系

{lien,g9441}@cs.nccu.edu.tw

摘要

受到封包網路原本忽略時效性特性之影響，對時效性要求極高的多媒體服務在封包網路上傳遞時，其服務品質低於傳統之電路交換網路。本論文針對單佇列及多佇列的 router 架構，提出一個適合各種不同營運者的品質方案的排程最佳化模型，並提出一套排程方法，根據封包的時效性及重要性賦予合適的 profit function，並參考封包在後續路程上各 router 的負載狀態以調整封包的送出順序，如此能善用有限的資源盡力提高排程的效能。實驗結果顯示我們的方法可以較每個 router 僅根據自身所知的資訊進行排程更可以有效提高網路效能，我們並根據實驗結果提出對於多等級服務的品質管理方針，可讓排程模組提高整體 QoS 滿意度。

關鍵詞：封包排程、服務品質

I. 前言

面對通訊與資訊科技的大幅進步、網際網路的蓬勃發展、以及電信自由化帶來的激烈競爭，通訊網路正在進行一個巨大的變革，要將原有電路交換及封包交換網路整合成一個單一的 All-IP 網路以支援所有網路應用服務。此種整合型 All-IP 網路將以一個單一傳輸平台提供固定網路及行動網路上所有服務，包括語音、多媒體、資料等各類服務 [1]。這些服務對於網路頻寬與服務品質各有不同的要求。現今 IP 網路僅提供簡單的 Best Effort 資料流傳送服務，如此的傳送服務並無法針對不同應用類型的封包做差別處理，以致效能不彰[2]。

封包在網路的傳遞過程中受到各種因素之影響，

於到達目的地時，可能會造成 long delay time, large jitter 或 high packet loss rate, 而在目的地端幾乎已經沒有補救機會，故而如果能在傳遞的過程中，依封包的時效性及重要性做適度的次序調動(rescheduling)而不要依序傳遞(FIFO)，讓過遲的封包提前送出，而將有時間餘裕的封包稍緩送出，如此截長補短，可提高網路效能及整體 QoS 滿意度。

3GPP 制訂的 UMTS 將各類應用服務依時效及品質要求劃分為四類，分別是交談式(Conversational)，串流式(Streaming)，互動式(Interactive)，背景式(Background)。不同服務對於網路頻寬與服務品質各有不同的要求。包括本文在內的很多研究都遵循這個分類，並提出各種提高品質的方案。不同的網路營運者有不同的營運目標，自然也有不同的品質政策。本文將針對單佇列及多佇列的 router 架構，提出一個適合各種不同營運者的品質方案的排程最佳化模型，並提出相應的最佳化方法。

1.1 Budget-Based QoS Management (BBQ)

BBQ[3]針對 All-IP 網路提出一種 Budget-Based 品質管理架構。對於 time-sensitive 和 connection-oriented (TSCO) 的服務，(例如 Conversational class 和 Streaming class)，BBQ 中的 admission control 會以一個快速的 call setup procedure 來處理，針對服務等級給予符合品質需求的路徑並保留所需的資源，對於其他種類的服務，則不保留任何資源，採用 Best-Effort 的政策提供服務。封包在 BBQ 管理下的 All-IP 網路傳遞時，會依循指定的路徑傳遞，而每一個 link 的耗費時間可以事先預估。本論文的方法適用於此類架構。

1.2 Router 架構

圖 1 為一般 router 的內部的簡單架構。當封包進入 router 時，router 裡的 packet forwarding 元件會依照封包所要傳送的目的地，選取適當的 output queue 將封包插入，再將 queue 內的封包依序送出，所謂的封包排程是將對個別的 output queue 中的封包進行送出順序的安排。

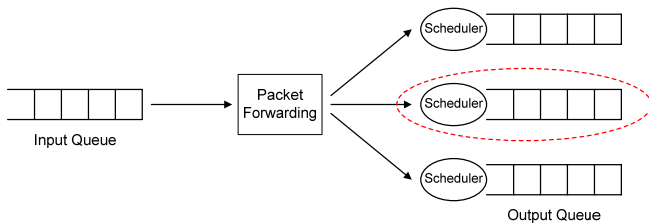


圖 1 Router 內部架構

1.3 排程方法

我們的研究假設兩種排程方算法，一是獨立排程 (Intra-Router Scheduling, IRS)，每個 router 根據自身所知的資訊而不參考其他 router 的狀態逕自進行排程，二是前瞻式排程 (Looking Ahead Scheduling, LAS)，在此排程法下，router 會參考封包在後續路程上各 router 的狀態進行排程。

本論文改進我們先前的研究 [4]，該論文提出獨立排程 (IRS) 方法，我們以該論文為研究基礎，改進 Multi-queue 架構，並利用前瞻式排程 (LAS) 更有效的提高網路效能及整體 QoS 滿意度。

1.4 論文架構

本文共分為五章，第二章介紹本文延續的相關研究 [4]。第三章介紹我們提出的兩個方法。第四章則藉由 NS-2 模擬以驗證我們的方法改善整體 QoS 滿意度的效果。第五章為結論與未來發展。

II. 相關研究

論文 [4] 在 BBQ 的架構下發展一套簡單而有效的方法。假設每個進入 router 的封包都帶有預定送出時

間及服務等級，並設計與封包傳遞時間及服務等級相關的四種 profit function，router 將視情況賦予該封包一個 profit function，scheduler 則根據封包的 profit function，以最大化總獲利和為目標決定其送出的順序，期望能在最終的接收端呈現較高的服務品質。

2.1 獨立排程架構與演算法

論文 [4] 所採取的排程方法分為兩步驟，對於一個待排程的封包，首先根據其重要性 (class)，選取一個合適的 profit function，再根據其時效性 (early or late) populate 所選的 profit function，並賦予該封包，最後以一個簡單的 greedy 演算法求得最恰當的位置插入於 output queue 中。

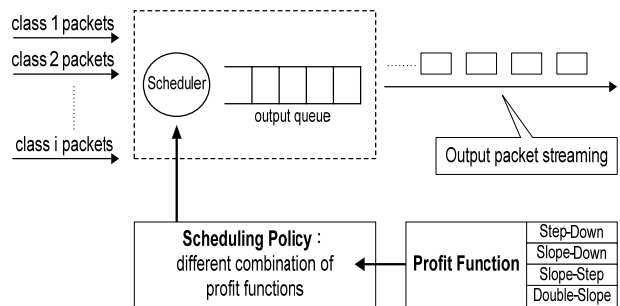


圖 2 Router 排程之整體架構

2.2 Scheduling Queue 架構

該研究提出兩種 Scheduling Queue 架構以支援所需的封包排程。一是 Single Preemptive Queue router (SPQ) 架構 (如圖 3)，另一則是 Multiple FIFO Queue router (MQ) 架構 (如圖 4)

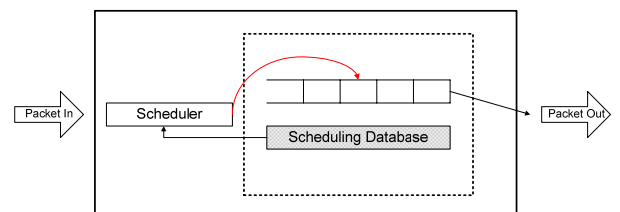


圖 3 Single Preemptive Queue router (SPQ) 架構

在 SPQ 架構下，router 內的個別 output queue 是由單一的 Preemptive Queue 所構成。Scheduler 可將進入

的封包插到 output queue 的任意位置。此種架構在速度要求極高的 router 中並不太實際，不但 scheduler 計算複雜度較高，而且硬體也較複雜。

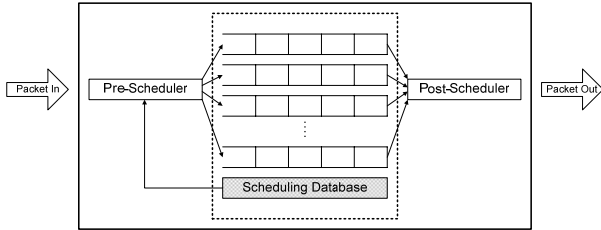


圖 4 Multiple FIFO Queue router (MQ)架構

在 MQ 架構下，router 內的個別 output queue 是由數個 FIFO queue 所組成，Pre-scheduler 只能將進入的封包插到其中一個 queue 的尾端。而後端排程器 (post-scheduler)，則採取輪詢方式依序將每一個 queue 中的封包一一送出。服務一個 queue 時，將該 queue 內的封包依序全部送出才輪到下一個 queue。MQ 架構的排程較簡單且硬體亦較簡單，在實際網路中比較容易被採用。MQ router 架構可視為 SPQ Router 的特例，因為只有固定位置(每個 FIFO queue 的尾端)可以插隊。

III. 前瞻式排程

我們首先提出一個具備一般性適合各種營運目標的排程最佳化模型。

3.1 Objective Function

我們定義了一個統一的最佳化指標如公式(1)所示，以 total charge (類似 utility function)做為整體服務品質的量化指標。假設訊務流分類成數種 class，每種 class 各有數條訊務流。該公式的基本概念便是分別將 quality index (q)、封包單位價格(C)及封包送出(接收)量(λ)相乘，最後加總的結果便是使用該 scheduling policy 的 total charge。而 quality index 最大值為 1，而其值如小於 1，表示品質受損，其獲利則依比例調降。最佳化的目標就是要找出使公式(1)最大的 scheduling policy。

$$c = \sum_{i=1}^3 \sum_{j=1}^{N_i} (q_{i,j} \cdot C_i \cdot \lambda_{i,j}) \quad (1)$$

其中符號及定義如表 1 所示，而公式(1)中針對 class i 中單一訊務流的 quality index 定義如公式(2)所示，其符號及定義如表 2 及表 3 所示。其中除 α 、 β 、 γ 等三種 Satisfaction Factor 之外的其他參數由網路營運者自行依照需求設定。Quality 之各項參數都有一個最低標準，charge threshold，參數值如低於對應的標準則視為不可接受，滿意度定為 0；反之則如公式(2)所示，其中 $B_i^{(1)}$ 、 $B_i^{(2)}$ 及 $B_i^{(3)}$ 就是 charge threshold。

表 1 Total Charge 的參數

Notation	Definition
$q_{i,j}$	quality of the j-th flow of class i ($0 \leq q \leq 1$)
C_i	unit price per byte of class i
$\lambda_{i,j}$	number of bytes admitted in the j-th flow of class i (Conversational or Streaming class)
	number of bytes received in the j-th flow of class i(Background class)
N_i	number of flows of class i

$$q_i = \begin{cases} a_i \cdot \alpha + b_i \cdot \beta + c_i \cdot \gamma; & \text{for average delay time of class } \geq B_i^{(1)} \\ 0 & \text{or average jitter of class } \geq B_i^{(2)} \\ & \text{or loss rate of class } \geq B_i^{(3)} \end{cases} \quad (2)$$

表 2 Quality Index 的參數

Notation	Definition
α (β 、 γ)	delay time (jitter、loss rate) satisfaction factor
a (b、c)	quality coefficient of delay time (jitter、loss rate)
$B_i^{(1)}$ $B_i^{(2)}$ 、 $B_i^{(3)}$	maximum acceptable average delay time (jitter、loss rate)

表 3 Satisfaction Factor 的定義

Notation	Formula
α_i	$1 - (\text{average delay time of class } i) / B_i^{(1)}$
β_i	$1 - (\text{average jitter of class } i) / B_i^{(2)}$
γ_i	$1 - (\text{packet loss rate of class } i)$

3.2 Profit function

我們沿用論文[4]所設計的 profit function 力求簡單有效，避免造成 router 太大負擔，有四個參數：hard deadline、soft deadline、Pre-soft deadline profit rate 及 Post-soft-deadline profit rate。其中 hard deadline 及 soft deadline 是指封包在個別 router 裡的 hard deadline 及 soft deadline，而非封包到達接收端的 deadline。

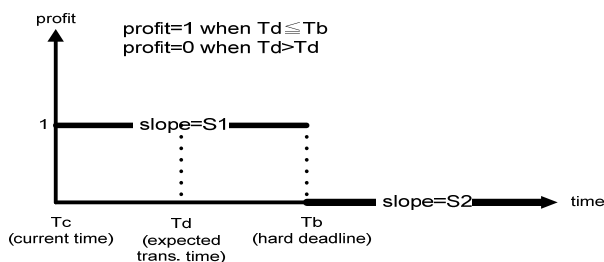


圖 5 Profit function of Step-Down Type

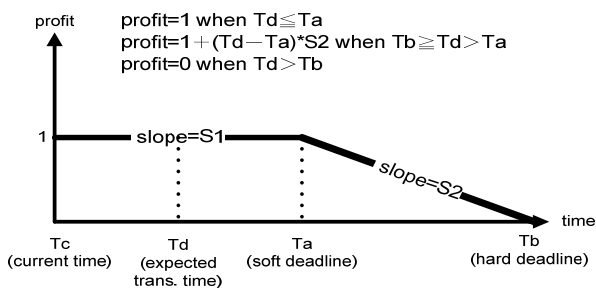


圖 6 Profit function of Slope-Down Type

當 scheduler 處理 Step-Down type 封包只須在 hard deadline 之前送出封包即可，提早送出並無額外利益，若無法在 hard deadline 之前送出則無 profit，便將之丟棄。

棄。

而處理 Slope-Down type 封包在 soft deadline 之前送出並無額外利益，但過了 soft deadline 斜率下降必須儘早送出封包否則利益受損，如過了 hard deadline 則無 profit，便將之丟棄。

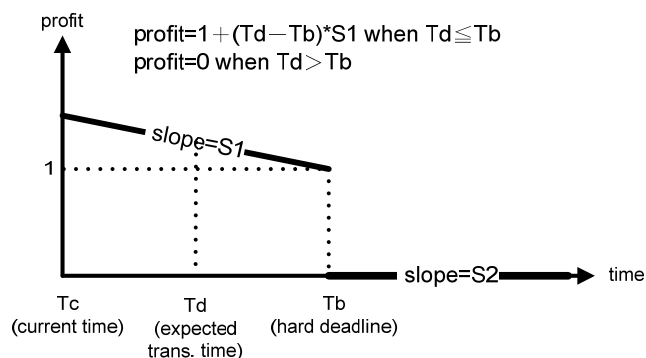


圖 7 Profit function of Slope-Step Type

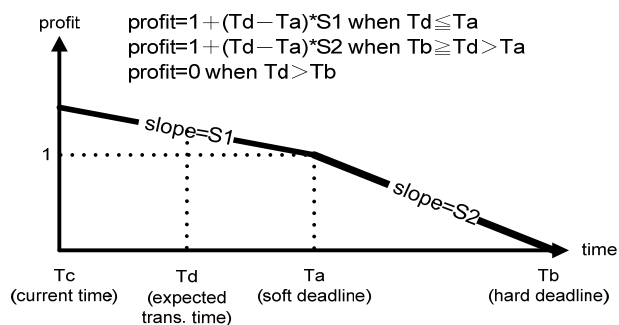


圖 8 Profit function of Double-Slope Type

當 scheduler 處理 Step-Down type 封包在 hard deadline 之前送出且越早送出 profit 越大。若無法在 hard deadline 之前送出則無 profit，便將之丟棄。

當 router 處理 Double-Slope type 封包時，除了在 soft deadline 之前越早送出 profit 越高之外，且 profit function 在 soft deadline 前後斜率不同代表送出時間所獲之不同利益，類似 Slope-Down 所描述。

我們將四種 profit function 分別賦予不同的服務等級，不同的搭配有不同的效果，每種組合我們稱之為 scheduling policy。

3.3 Priority Multiple Queue router (PMQ)

若將[4]中 MQ 與 SPQ router 架構比較，可以發現可以插隊的位置大幅減少，而且缺少最前面的位置以供插隊，導致非常緊急的封包也要排隊影響排程效果。我們提出 Priority Multiple Queue router (PMQ)以彌補這個缺點，讓緊急的封包可以優先被送出。

如圖 9 所示，PMQ 與 MQ 差不多，唯讓其中一個 queue 的優先權提成最高以供較為緊急的封包插入，稱作 priority queue。其他的 queue 稱作 round-robin queue。只需改變 post-scheduler 的服務順序即可。只要 priority queue 中有封包進入，post-scheduler 就會先送出，而 round-robin queue 的服務則維持輪詢的方式。其中 scheduling database 包含某些排程資訊，例如 profit function、前瞻式排程相關資訊等。

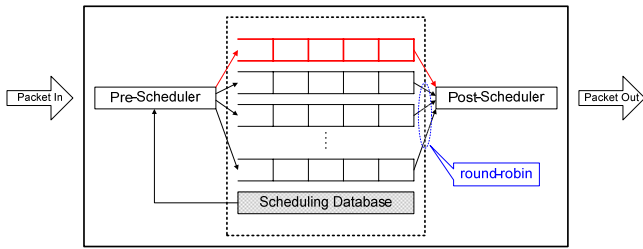


圖 9 Priority Multiple Queue router (PMQ)架構

3.4 前瞻式排程 (Looking Ahead Scheduling)

本研究所提出的前瞻式排程相對於[4]所提出的獨立排程，最主要的差別是前瞻式排程中 router 會參考封包在後續路程上各 router 的狀態讓 scheduler 能更準確的預估封包及時到達目的地的機會。

A. 估計後續 router 負載

為了不增加網路的負載或演算法的複雜度，我們利用 routing table 中的資訊來估計後續 router 之負載，以進行前瞻式排程。假設 router 使用 Link-State routing protocol，在這類路由協定中，每個 router 都擁有相同且完整網路拓撲及 link cost 的資訊，Link-State routing protocol 提供了幾項 metrics 來計算 link cost，本研究中假設 routing metric 為 delay time。

B. 動態調整 profit function 參數

本研究假設每個封包在傳遞間所經歷的網路元件都有事先預定的停留時間，因此可以預定封包到達每一個 router 的時間及其送出時間。而由於 IP 封包的傳遞時間起伏不定，會使得封包已傳遞時間比預定的時間早到或遲到。

當封包到達每個 router 時，我們可以透過 routing table 中的資訊預估該封包的剩餘傳遞時間比預定剩餘時間長或短。我們可以利用這些資訊來動態調整 profit function 的參數，使得封包的緊急程度不僅決定於目前所耗用時間之狀況，更根據未來可能的時間消耗狀況決定其緊急程度，如圖 10 所示。

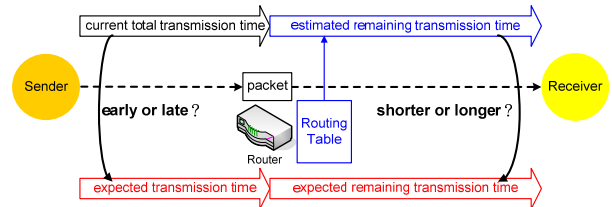


圖 10 動態調整 profit function 參數的依據

在前瞻式排程中，我們將實際已傳遞時間減去預定已傳遞時間作為該封包在傳遞中的已拖延時間 (d_{pre})，而將估計剩餘傳遞時間減去預定傳遞剩餘時間作為後續預計延遲的時間 (d_{post})，兩者相加結果即為封包在整段傳遞過程中可能的預計總拖延時間。若預估該封包落後預定時程，就將 local deadline 適當地往前調(亦即提高其優先度)，使 scheduler 將封包提早送出。我們另外設定一個調整係數(c)調整各 router 的 local deadline 以逐步調整封包的傳遞時間。我們將會以實驗找出最佳的調整係數。調整公式如表 4 所示。

表 4 Local deadline 調整公式

Profit function	Adjusting formula
Step-Down Slope-Down	hard local deadline = $c \cdot (d_{pre} + d_{post})$
Slope-Down Double-Slope	soft local deadline = $c \cdot (d_{pre} + d_{post})$

IV. 效能評估

我們使用 NS-2[5]網路的模擬平台評估我們提出的排程方法。

4.1 實驗環境及拓模

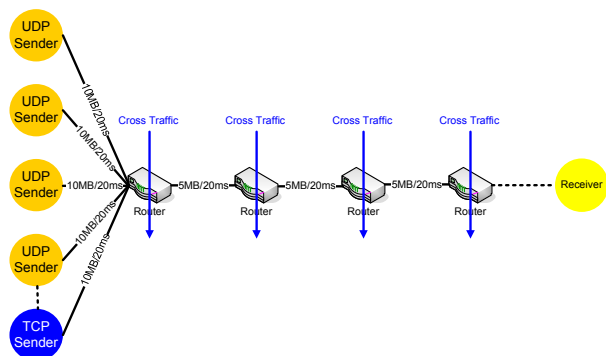


圖 11 實驗拓模一

圖 11 是我們大部分實驗的拓模，其中虛線部分代表實驗控制變數。在傳送端我們設定 real-time 以及 Background class traffic，每個 router 都會有 cross traffic 經過，cross traffic 採用 Background class traffic，其中 real-time class traffic 以 CBR 的方式傳送，而 Background class traffic 以 FTP 的方式傳送。

4.2 實驗設計

我們先以單一 TSCO 等級作實驗，觀察使用不同 router 架構及排程演算法的效能表現。

再以多個 TSCO 等級服務進行實驗。觀察多 TSCO 等級服務的封包調配不同 profit function 間的資源搶奪行為與不同等級服務間的差別化待遇。這部份是我們研究的重點，我們將藉由實驗找出適合的 policy。

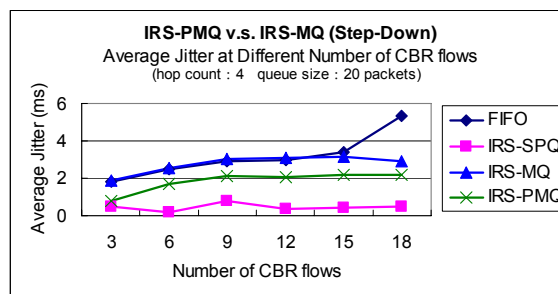
4.3 評估指標

我們觀察 jitter 平均值、封包遺失率及封包平均延遲時間的變化作為單一服務等級實驗評估指標，並作為設計多服務等級 scheduler 之參考。而以公式(1)所定義的 total charge 來做為多服務等級實驗評估指標。

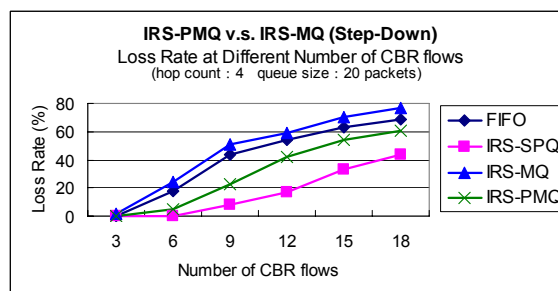
4.4 實驗一：在單一服務等級環境中比較

IRS-PMQ 與 IRS-MQ

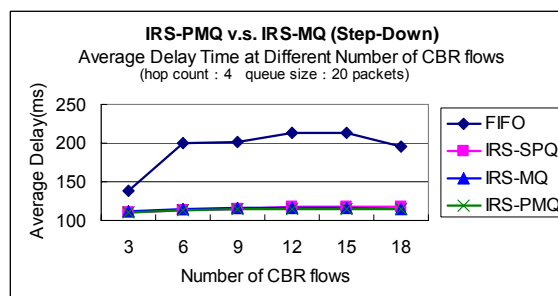
本實驗調整網路負載以觀察我們提出的 PMQ router 架構的效能變化。在實驗一我們放入傳統 FIFO 與[4]中的 SPQ router 架構的實驗結果以供比較。



(a)



(b)



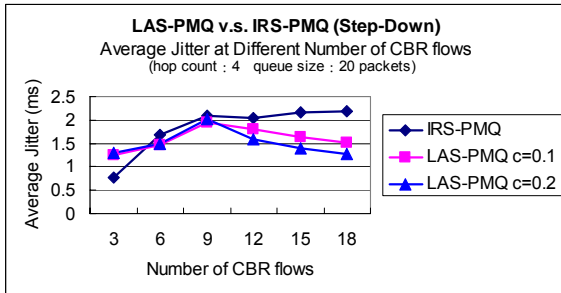
(c)

圖 12 實驗一 變動負載下 IRS-PMQ 與 IRS-MQ 之效能比較 (a)Jitter 平均值 (b)封包遺失率 (c)平均延遲時間

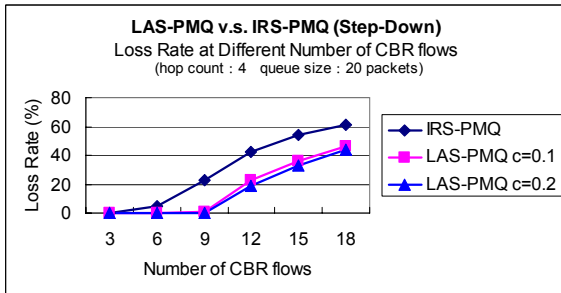
由實驗結果可以看出，在網路負載越來越大的情況下，我們所提出的 PMQ router 架構的確比 MQ router 架構更可以改善整體 QoS，尤其以 jitter 與封包遺失率更為明顯，更逼近 SPQ 排程演算法。

4.5 實驗二：在單一服務等級環境中比較 LAS-PMQ 與 IRS-PMQ

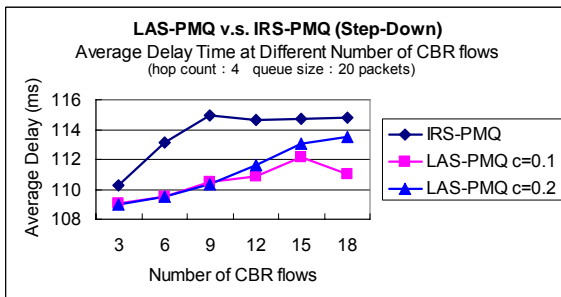
本實驗調整網路負載以觀察我們提出的前瞻式排程的整體效能。在實驗二中，我們與 PMQ router 架構下的獨立排程演算法做比較，並且藉以找出最佳的調整係數。



(a)



(b)



(c)

圖 13 實驗二 變動負載下 LAS-PMQ 與 IRS-PMQ 之效能比較 (a)Jitter 平均值 (b)封包遺失率 (c)平均延遲時間

由實驗結果可以看出，在網路負載越來越大的情況下，前瞻式排程演算法的 jitter 與平均延遲時間都小

有改善，尤其封包遺失率的改善更為顯著。對於前瞻式排程的調整係數，我們選擇 0.1 當作我們前瞻式排程的最佳調整係數，並進行後續實驗。

4.6 實驗三：探討雙 real-time 服務等級共存的效能表現

圖 14 是實驗三的拓模。在這個實驗中我們逐漸增加 VoIP 的訊務流數量以觀察不同的 VoIP 比例在網路中的整體效能表現，並藉由前述定義的多服務等級評估指標找出使整體網路效能及滿意度最佳的 scheduling policy。本實驗將我們的方法與 Priority Queue[6]的方法做效能之比較。

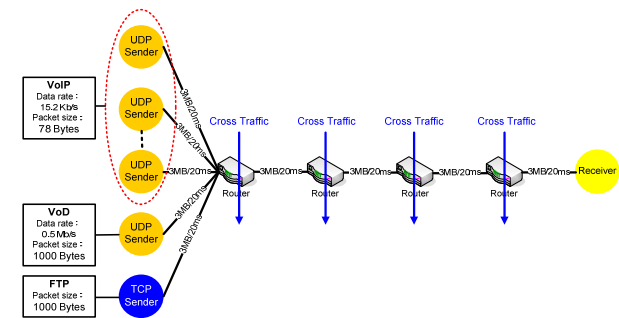


圖 14 多服務等級實驗拓模

在雙 real-time 服務等級的情況下，我們將四種 profit function 兩兩組合，之後賦予 Conversational class 及 Streaming class，共有十二種不同的 scheduling policy。

在這個實驗中我們使用前述定義的多服務等級評估指標來衡量不同 scheduling policy 所獲得的 total charge。其中針對 Conversational class 的 a、b、c 分別為 0.6、0.6、0.2， $B^{(1)}$ 、 $B^{(2)}$ 、 $B^{(3)}$ 分別為 150(ms)、15(ms)、0.05，c 為 0.001；針對 Streaming class 的 a、c 分別為 0.8、0.2， $B^{(1)}$ 為 1(min)，c 為 0.0003；而針對 Background class 直接將 q 設為 1，因為我們只考慮此類封包是否有到達目的地，而 c 為 0.0001。

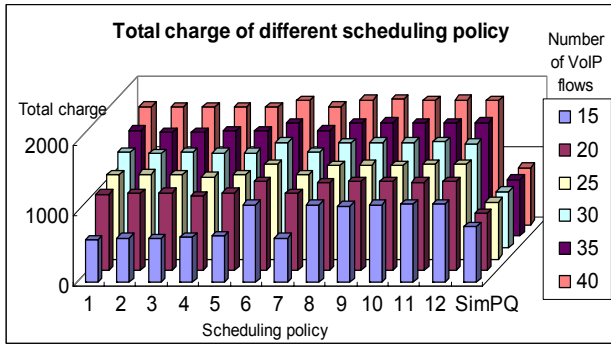


圖 15 實驗三 雙 real-time 服務等級共存時的效能表現

圖 15 是實驗三的結果，其中包含所有 scheduling policy 的實驗結果，而 SimPQ 是指我們根據[6]所實作的 Simulated Priority Queue 排程演算法。由實驗結果可以看出在我們定義的評估指標下，只要是將 Conversational class 賦予 Slope-Down 或 Double-Slope 的 scheduling policy 整體效能都有不錯的表現。而觀察整體效能的結果，我們推薦 scheduling policy 6，即 Conversational class 的封包賦予 Slope-Down，而 Streaming class 的封包賦予 Double-Slope 的 profit function，這個組合有最佳的整體網路效能。

觀察 SimPQ 的實驗結果可以發現在 VoIP 訊務流數量較少(15 條)的情狀下會優於我們方法中較差的 scheduling policy；但在 VoIP 訊務流數量越來越多的情況之下，當服務 VoIP 的 queue 不敷使用以後，則會開始捨棄 VoIP 封包，相當不利於整體滿意度，所以所獲得的 total charge 相當低。在我們的實驗環境、所設計的 charge function 及參數設定下，網路高負載(VoIP 訊務流數量多於 20 條)的情況，我們的方法比 SimPQ 排程演算法能提升 34% 以上的 total charge，如圖 15 所示。

V. 結論

本研究延續[4]提出 per-hop 封包排程的方法，利用與封包傳遞時間相關的 profit function 搭配排程演算法，首先改善該研究中的 MQ router 架構，並且在每個 router 對封包排程的同時，參考封包在後續路程 router 的負載，改變封包送出的順序，並透過實驗模擬驗證我們方法的效果表現，最後實驗結果顯示我們的

方法能有效改善整體效能。

參考文獻

- [1] 3GPP, "Technical Specification Group Services and Systems Aspects: Architecture for an All IP network," 3GPP TR 23.922 version 1.0.0., Oct. 1999.
- [2] P. Lorenz, "Quality of service and new architectures for future telecommunications networks," *Proc. of MILCOM*, vol. 2, pp. 695-698, Oct. 2000.
- [3] Yao-Nan Lien, Hung-Ching Jang, Tsu-Chieh Tsai, and Hsing Luh, "Budget Based QoS Management Infrastructure for All-IP Networks," *Proc. of ICACT*, vol. 1, pp. 185-190, Feb. 2005.
- [4] Tsung-Ming Lin, "A New Jitter Control Mechanism by Per-Hop Packet Scheduling Approach," Master thesis, National Chengchi University, Dec. 2006.
- [5] "The Network Simulator - ns-2," <http://www.isi.edu/nsnam/ns/>.
- [6] S.-W. Moon, K.G. Shin, and J. Rexford, "Scalable Hardware Priority Queue Architectures for High-Speed Packet Switches," *Proc. of RTAS*, pp. 203-212, Jun. 1997.