

國立政治大學資訊科學系  
Department of Computer Science  
National Chengchi University

碩士論文

Master's Thesis

利用 Per-Hop 封包排程的延遲變化率控制機制  
A New Jitter Control Mechanism by Per-Hop Packet  
Scheduling Approach

研究生：林宗銘

指導教授：連耀南

中華民國九十五年 月

2006

利用 Per-Hop 封包排程的延遲變化率控制機制  
A New Jitter Control Mechanism by Per-Hop Packet  
Scheduling Approach

研究生：林宗銘      Student：Tsung-Ming Lin

指導教授：連耀南      Advisor：Yao-Nan Lien

國立政治大學  
資訊科學系  
碩士論文

A Thesis  
submitted to Department of Computer Science  
National Chengchi University  
in partial fulfillment of the Requirements  
for the degree of  
Master  
In  
Computer Science

中華民國九十五年

2006

# 授權書

# 中文審定書

## 英文審定書

# 利用 Per-Hop 封包排程的延遲變化率控制機制

## 摘要

近年來通訊網路的進步與寬頻網路的發展，使得更多新型的網路服務迅速應運而生，對網路服務的品質(QoS)也更加要求。眾多新興的應用服務中，即時性的應用服務(Real-Time Traffic)，像是 Voice over IP (VoIP)以及 Video on Demand (VoD)，對於封包的傳遞延遲(Delay)以及抖動(Jitter)具有嚴格的要求，但處理即時性服務時，並非將封包快速送出即可，而是要在一定的延遲時間內送出，過快或過慢送出都不好。因此，本研究致力於發展出一套簡單又有效的方法，在封包所經過的每一個 Router 時，以 Multiple FIFO Queue 搭配排程器，再利用我們設計與封包傳遞時間相關的 Profit Function，來對封包進行重新排程，期望減小在接收端之 Jitter 絕對值總和，進而增進網路的效能，最後利用網路模擬工具 NS2 來評估我們的方法，證明我們的方法可以改善 Real-Time Traffic 的 Jitter。

# A New Jitter Control Mechanism by Per-Hop Packet Scheduling Approach

## Abstract

Delivering real-time streaming over the Internet is one of the critical applications nowadays. Novel real-time Internet applications, like Voice over IP (VoIP) and Video on Demand (VoD), have delay and jitter requirements. For jitter, where the goal is to minimize the difference between delay times of different packets, we propose a new jitter control mechanism by per-hop packet scheduling. The approach in this paper is to reschedule all packets in each gateway when packet arrives gateway. Our scheduling algorithm is to reschedule packets according to packets' profit function we design. We evaluate our approach on NS2. Simulation results show that our approach can improve jitter of real-time traffic.

## 目錄

### 第一章 簡介

#### 1.1 研究動機

#### 1.2 論文架構

### 第二章 相關研究

#### 2.1 DeJitter Buffer

#### 2.2 Jitter Control in QoS Networks

#### 2.3 Providing Guaranteed Services Without Per Flow Management

#### 2.4 在 IPv6 的 DiffServ 網路上具有 Credit/Deficit 調整的延遲變化 率減小機制

### 第三章 利用 Profit Function 為基礎的封包排程方法

#### 3.1 Router 架構

##### 3.1.1 Single Preemptive Queue 架構

##### 3.1.2 Multiple Queue 架構

#### 3.2 Profit Function 類型

#### 3.3 排程演算法

##### 3.4.1 Single Preemptive Queue 排程演算法

##### 3.4.2 Multiple Queue 排程演算法



## 第四章 模擬及效能評估

### 4.1 模擬與實驗環境

### 4.2 實驗參數

### 4.3 評估指標

### 4.4 實驗設計

### 4.5 實驗結果

#### 4.5.1 Single Class

##### 4.5.1.1 Single Preemptive Queue

##### 4.5.1.2 Multiple Queue

#### 4.5.2 Multiple Class

### 4.6 總結

## 第五章 結論與未來發展

## 參考文獻

## 圖目錄

## 表目錄

# 第一章

## 簡介

### 1.1 研究動機

近幾年來通訊網路已經成為社會中相當重要的一部份，各式各樣的應用服務油然而生，由於每項應用對於封包的 Delay(傳遞延遲)、Jitter(抖動)以及 Packet Loss(封包遺失)都有不同的要求，使得舊有的網路架構無法符合大多數新型的應用，舉凡像是 VoIP 技術的相關運用對於 Delay 以及 Jitter 有著嚴格的要求。

由於 IP 封包的傳遞時間起伏不定，造成一個資料流在傳送封包時，各個封包到達接收端的時間忽快忽慢，而產生 Jitter 的現象。Jitter 越大，表示網路狀況就越不穩定，難以提供好的服務品質。變動過於劇烈對於交談式以及串流式的應用造成嚴重服務品質下降。對於一個資料流，任意相鄰的兩個封包之 Jitter 的定義如下：

$S_j$  = the send time of packet  $j$

$R_j$  = the arrival time of packet  $j$

$S_i$  = the send time of packet  $i$

$R_i$  = the arrival time of packet  $i$

$Jitter = (R_j - S_j) - (R_i - S_i)$

$j > i$

Jitter 對於服務品質有著關鍵的影響，因此我們必須去對不同的資料流，做不同的處理。針對這點，3GPP 在 1999 年初所開始制訂的 UMTS 依應用服務類型提出 QoS 分類架構，其中包含對 real-time traffic 的服務考量。

在應用服務上，UMTS 將網路上風行的主要應用依時效與品質需求概略分為四大類，如表 1，分別是交談式(Conversational Class)，串流式(Streaming Class)，互動式(Interactive Class)，背景式(Background Class)。

表 1

類別	應用之例
Conversational	VoIP
Streaming	VoD
Interactive	Telnet
Background	E-Mail

交談式主要用來支援人類雙向溝通，根據人類感官之經驗歸納，此種服務對 long Delay time 與 Jitter 相當敏感，使用者在 Delay time 超過 300 ms 時，就難以忍受其通話品質。串流式則要求持續穩定的資料流，因此對 Jitter 相當敏感。互動式與背景式都屬於 data communication 的服務，可容忍較長的資料傳送時間，但要求正確的資料傳送，因此幾乎無法忍受資料的遺失。表 2 顯示各個類別的品質需求。

表 2

	Delay Sensitivity	Jitter Sensitivity	Packet Loss Sensitivity
Conversational	High	High	Low
Streaming	Medium	High	Low
Interactive	Medium	Low	High
Background	Low	No	High

## 1.2 論文架構

本文共分為五個章節，第二章介紹一些近幾年來對 Jitter 相關的研究，並分析其優缺點。第三章介紹我們的排程機制。第四章則介紹利用 NS2 模擬以驗證我們方法改善 Jitter 的效果。第五章為結論與未來發展。

## 第二章

### 相關研究

#### 2.1 DeJitter Buffer

近來有許多關於 Jitter 的研究，最簡單的消除 Jitter 方法是使用 DeJitter Buffer，此方法是在接收端設定一個較大的 Buffer，方法簡單，但是可能會有相當長的 Delay Time。

#### 2.2 Jitter Control in QoS Networks

由 Yishay Mansour 以及 Boaz Patt-Shamir 於” Jitter Control in QoS Networks” 提出的 Jitter Control，此方法是提出一個簡單的 on-line 的 Jitter Control，每個 Router 的 Buffer 大小為  $2B$  個單位封包。當 Buffer 負載達到  $B$  時，則將 Buffer 前端的封包送出，之後每隔  $X_a$  時間送出一個封包，其中  $X_a$  為封包的平均間隔時間。當 Buffer 負載未達到  $B$  時，則以 First Come First Serve 為原則送出封包。所謂 on-line 的演算法是指一個演算法在時間  $T$  的動作是一個根據在時間  $T$  以前或時間  $T$  時封包到達和釋放的函數；而 off-line 演算法的動作則是會根據未來發生的事件。此方法使用 on-line 的 Jitter Control 有效率的使用 Buffer 資源，作者此研究指定了封包送出時間，但並無改變封包送出順序，因此，無法針對封包屬於何種的應用類型作處理，可能會導致 real-time traffic 的 Jitter 變動過於劇烈以及 Delay Time 過長。

#### 2.3 Providing Guaranteed Services Without Per Flow Management

由 Ion Stoica 以及 Hui Zhang 在” Providing Guaranteed Services Without Per Flow

Management”提出的 Jitter Control 為 Core-Jitter-VC(CJVC)，目的在改善 Jitter Virtual Clock(Jitter-VC)的缺點。由於 Jitter Virtual Clock(Jitter-VC) 必須在 Edge Router 以及 Core Router 都記載著 Per Flow State，以讓 Router 根據封包到達時間計算出每個封包的 eligible time 和 deadline，進而對封包排程。因此，作者提出的 CJVC，使用 Dynamic Packet State(DPS)的方法，只在 Edge Router 記載 Per Flow State，當封包進入 Edge Router 時，預先計算 eligible time 和 deadline 的資訊並加入封包表頭。CJVC 可和 Jitter-VC 達到同樣的效果。但封包表頭夾帶的資訊會隨著經過的 Router 個數而增加，會面臨 scalability 的問題。且此研究是在 IntServ 架構下，因此會面臨 scalability 的問題，然而在 All-IP 網路下，使用 DiffServ 架構較為恰當。

## 2.4 在 IPv6 的 DiffServ 網路上具有 Credit/Deficit 調整的延遲變化率減

### 小機制

國立中山大學徐毅銘在”在 IPv6 的 DiffServ 網路上具有 Credit/Deficit 調整的延遲變化率減小機制”提出利用 Credit/Deficit 的方法，將每一個 Hop 的 Queueing Delay 用一個值代表，每個封包都有其預設的 Per-Hop Queueing Delay。封包若是在預設時間內送出，表示封包提早送出(具有 Credit)，反之，則表示封包太晚送出(具有 Deficit)。將 Credit/ Deficit 隨著封包傳輸夾帶在封包表頭中，調整封包傳送順序，將累積的 Credit/ Deficit 消除。對 Jitter 有一定程度的改善，但無針對封包的重要性做不同的處理。



## 第三章

### 利用 Profit Function 為基礎的封包排程方法

Jitter (抖動)在即時性應用服務上，是一個非常重要，必須克服的問題，All-IP 網路受限於 Packet Switching 原有的特性，使得 IP 封包的傳遞時間起伏不定。最簡單的消除 Jitter 方法是使用 DeJitter Buffer 讓到達接收端的封包先在 Buffer 中等待一段時間，等到 Buffer 內有足夠的資料時，再依傳送端原有步調送給使用者，進而降低 Jitter。但有一重大缺點，因為 Delay time 與 Buffer 大小成正比，而 Buffer 大小卻決定了消除 Jitter 之能力。在某些應用，像 VoIP 對於 Delay time 非常敏感，此方法就無法達到良好的效果，在接收端才開始對 Jitter 做控制已經太遲了，為此我們必須提出更有效的方法降低 Jitter。

現今的 Router 對於進來的封包是以 FIFO(first in first out)的方式處理，無法有效處理即時性訊務的 Jitter。當每一個 Router 可獲得所經過之封包預定送出的時間之資訊時，Router 便可利用這些資訊，做出適當的排程，先送出遲到的封包，而將提前到達的封包延後送出，以提高網路效率，包括降低 Jitter，減少封包捨棄，減少 Delay time 等。

我們在此研究中將提出” Per-Hop Jitter Control” 的方法來控制 Per-Hop 封包排程。在封包所經過的每一個 Router 時，以 Multiple Queue 為基礎，搭配排程器，再利用我們設計與封包傳遞時間相關的 Profit Function，來對封包進行重新排程，決定適當的送出順序，期望減小在接收端之 Jitter 絕對值總和。以下以數學式表示，以  $T_{Cb}[a]$  代表 Cb 連結第 a 個封包從傳送端到接收端所花的傳遞時間。假設共有 N 條即時性訊務連結  $C_1 \sim C_n$ ，各連結分別送出  $\#(C_1) \sim \#(C_n)$  個封包，期望使

$$\sum_{b=1}^n \sum_{a=2}^{\#(Cb)} |T_{cb}[a] - T_{cb}[a-1]|$$

減小。然而，Packet Loss 會致使計算出的結果減小，導致此衡量標準不夠客觀，Jitter 變動幅度較大的資料流，可能會因為 Packet Loss，使得計算的結果比 Jitter 變動幅度較小的資料流來的小。因此，我們必須對此計算結果做正規化 (normalization) 的運算，讓計算的結果更具有代表性，我們將接收端之 Jitter 絕對值總和除以接收端接收到封包的個數，以做為最後評估時的基準。

我們假設每個封包從傳送端到接收端中間所經歷的網路元件(Router)都有事先預估的停留時間，因此，每個封包到達每個 Router 都有預定送出的時間，當封包由傳送端進入網路時，因為 IP 封包的傳遞時間本就是起伏不定，因此封包到達每一個 Router 的時間可能會比原先估計的時間早到，也可能比預估的時間晚到。對於即時性的應用服務，我們希望封包能夠以傳送端原有步調到達接收端，封包太早到達或太晚到達接收端都不好，太早到達的話，Router 及接收端需要有較大的 Buffer 將早到的封包暫存，太晚到達的話，接收端必須等待封包到達，對於某些應用，例如交談式服務(例如 VoIP) 以及串流式(例如 VoD)，會造成 QoS 大打折扣。我們於封包所經過的每一個 Router，根據其行程的遲早及重要性調整其傳送的先後次序，以期達到減小在接收端之 Jitter 絕對值總和。

為了最小化 Jitter，每個 Router 對封包排程時必須取得整個網路的資訊，如此作法複雜度太高，難以運用在實際狀況中。因此，每個 Router 各自做排程的決定，在此我們利用與封包傳遞時間相關的 Profit Function 配合排程演算法來決定封包排程順序，亦即傳遞成功一個封包之報酬與其是否準時送達收受者有關。不同的封包排程演算法以及 Profit Function 將會影響控制 Jitter 的效果。因此，我們將研究重點擺在不同的 Profit Function 和排程演算法對 Jitter 的影響。

### 3.1 Router 架構

#### 3.1.1 Single Preemptive Queue 架構

本研究將針對單一 output Queue 對封包進行重新排程，如圖 1。在第一階段將假設 Router 任一個 output Queue 是由 single Queue 所構成，如圖 2，而 Router 可任意將封包插到 output Queue 的任意位置。在此種 Router 的架構下，可將進入 Router 的封包，依其重要性以及行程的遲早(以 Profit Function 表示)進行排程，並插入 output Queue 的適當位置，達到其最高效能指標(亦即最大 Profit)，最終以達到改善接收端之 Jitter 為目的。

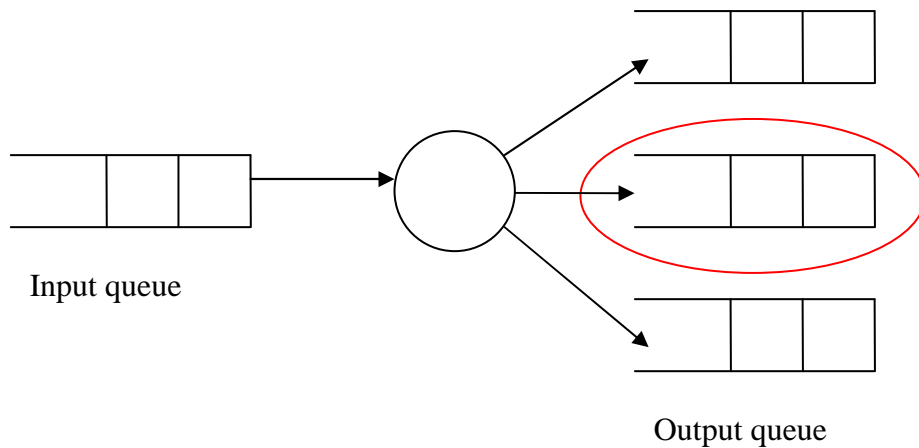


圖 1：簡化 Router 示意圖

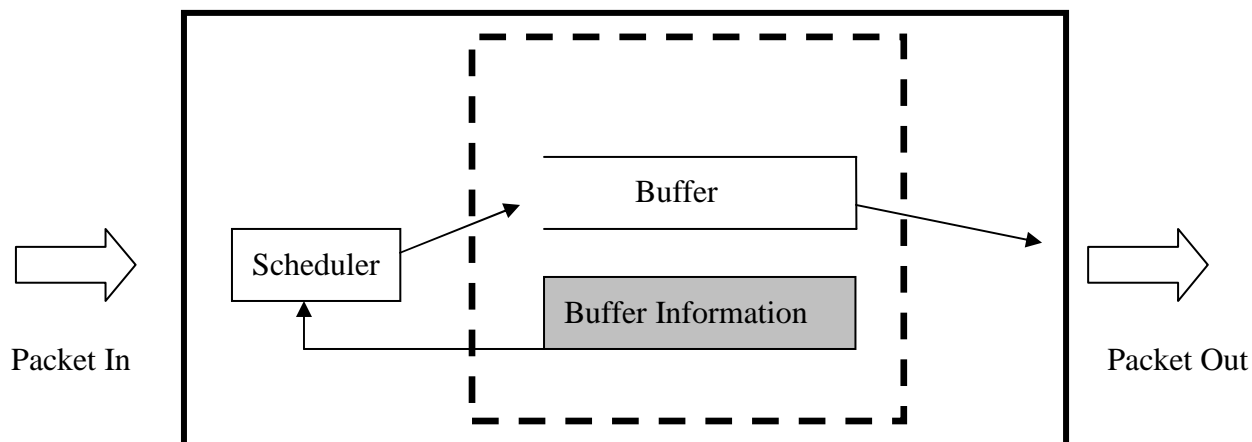


圖 2：Single Preemptive Queue

### 3.1.2 Multiple Queue 架構

第一階段的方法，主要是供作基準比較之用，過於理想不實際，因為實際網路中的 Router 為了加快交換速度不會採取 preemptive Queue 作法（因而排程 Agent 無法任意將近來的封包插到 output Queue 得任意位置）。因此，在第二階段，我們將以較合乎實際的 Router 架構進行我們的封包排程。假設 Router 有數個 FIFO Queue，並利用諸如 WFQ(Weighted Fair Queue)的方法將封包依序傳送出去，以近似 Single Preemptive Queue 的效果。如圖 2 及圖 3。圖 3 之架構加入處理分級的服務方法，可賦予某些 Queue 較高的優先權，在封包轉送過程中，較快的獲得轉送服務。

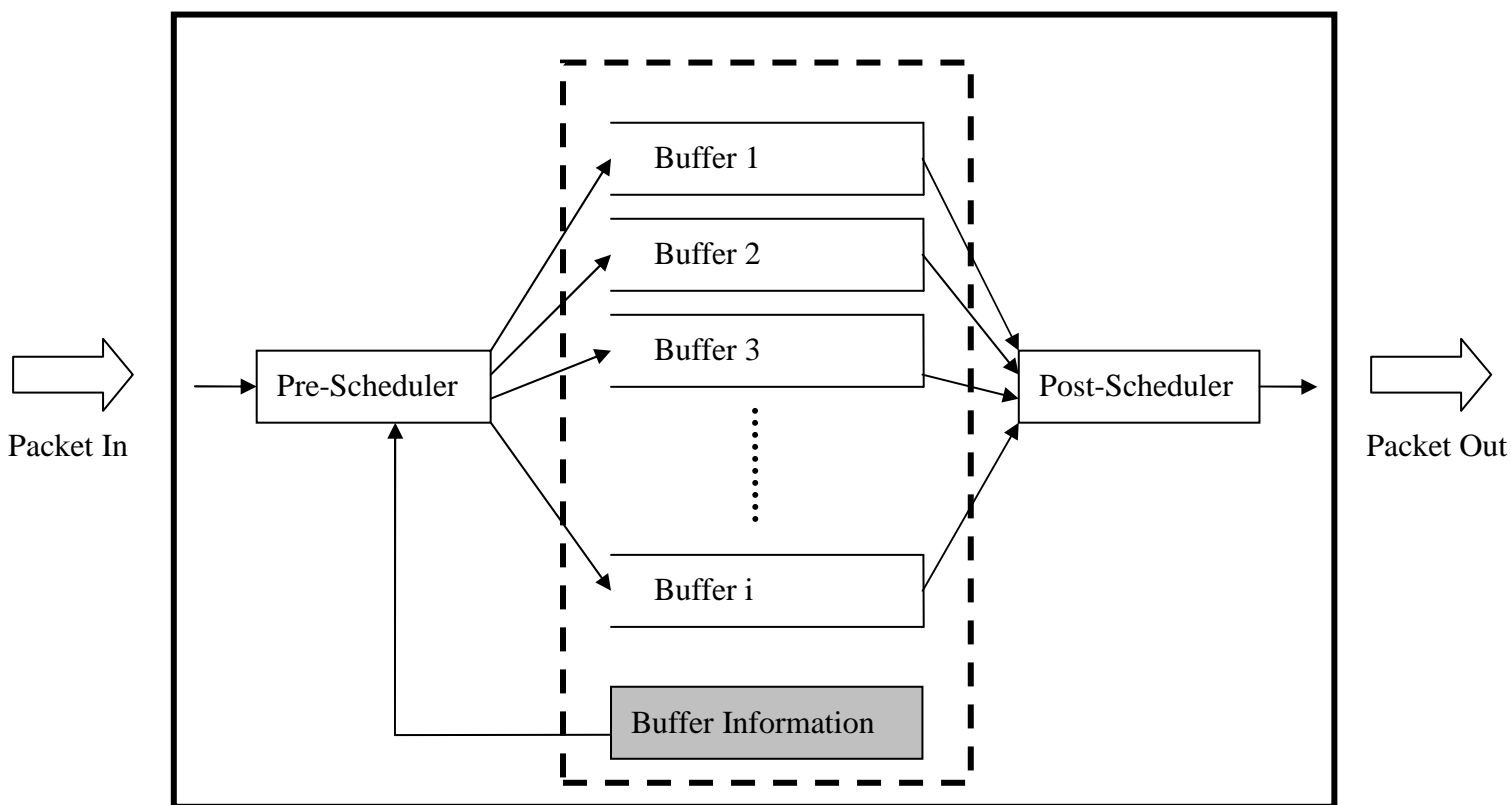


圖 3：Multiple Queue

### 3.2 Profit Function 類型

在本研究中，不同的 Profit Function 不僅影響到 Router 的排程順序，同時也會影響最後接收端控制 Jitter 的效果。然而，並非只用一種類型的 Profit Function 就能適用網路所有的情況，因此，我們設計了多種不同類型的 Profit Function，如圖 4 到圖 9，並觀察不同類型的 Profit Function 對於最後接收端控制 Jitter 的效果。

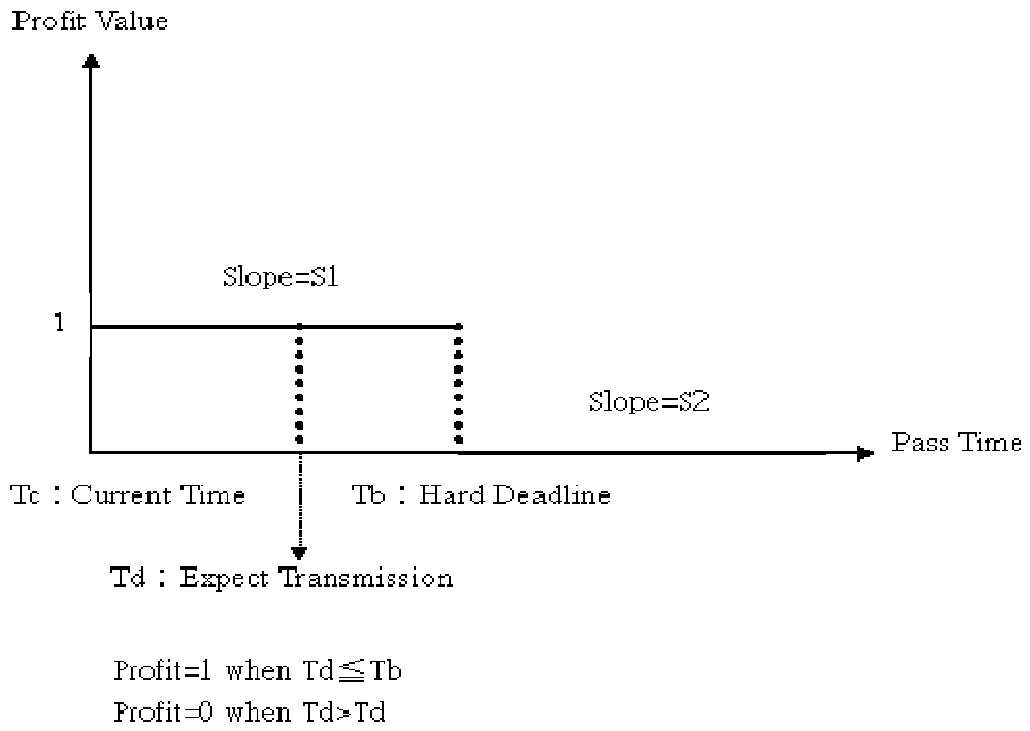


圖 4 : Profit Function 1

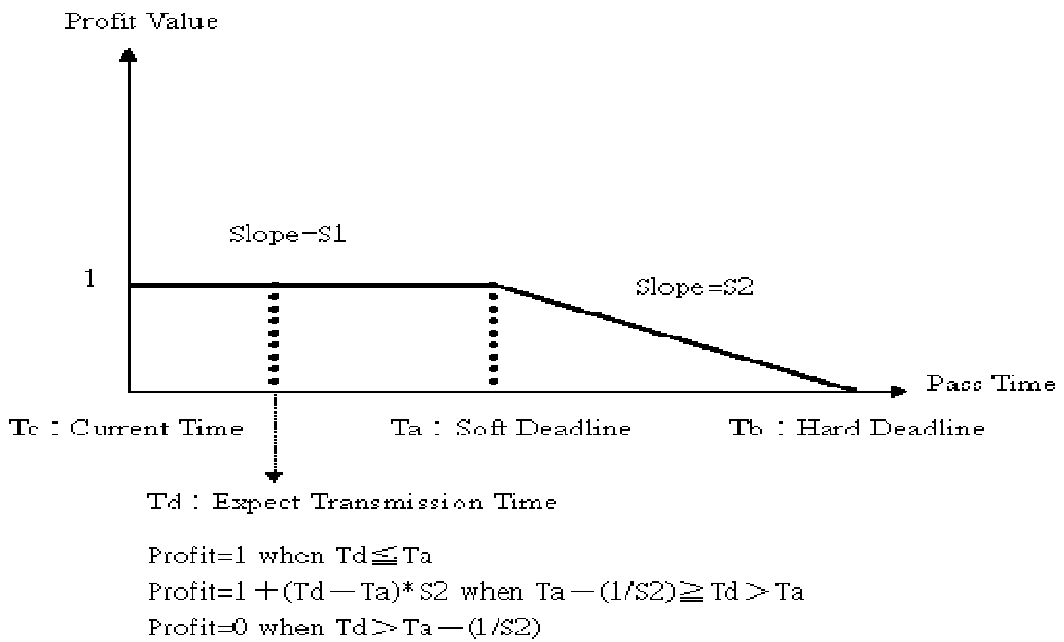


圖 5 : Profit Function 2

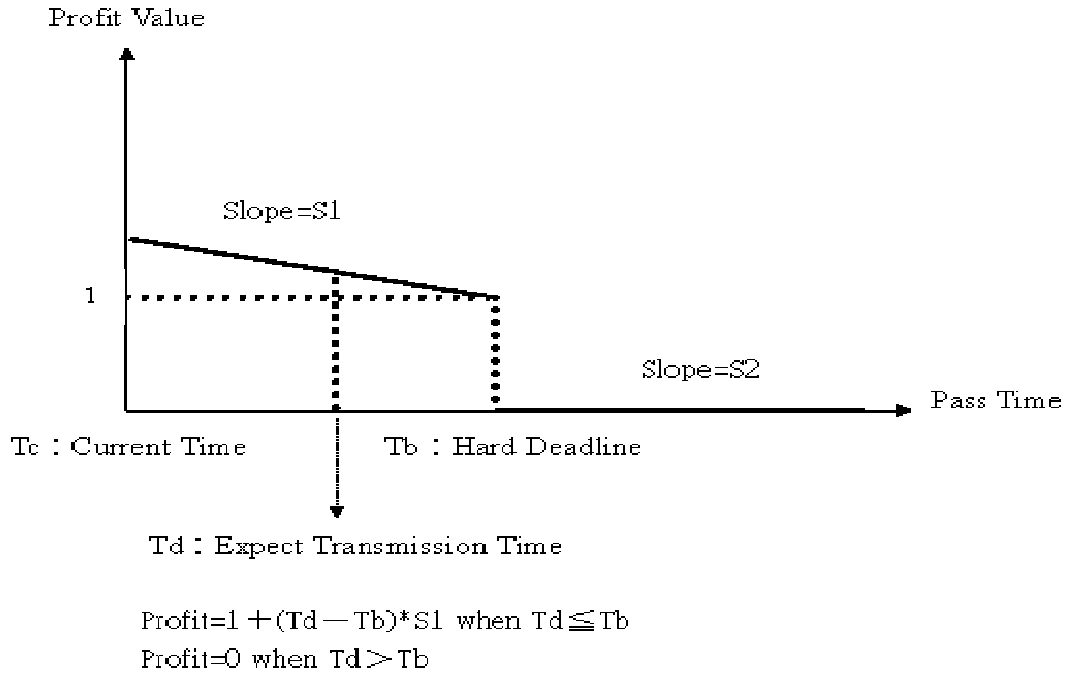


圖 6 : Profit Function 3

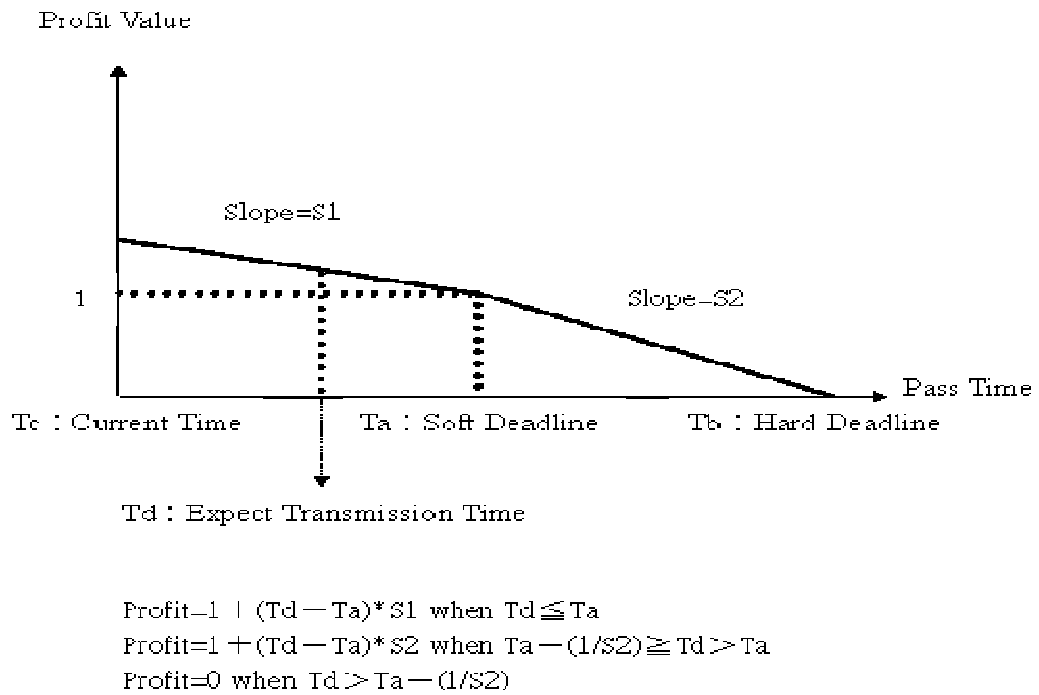
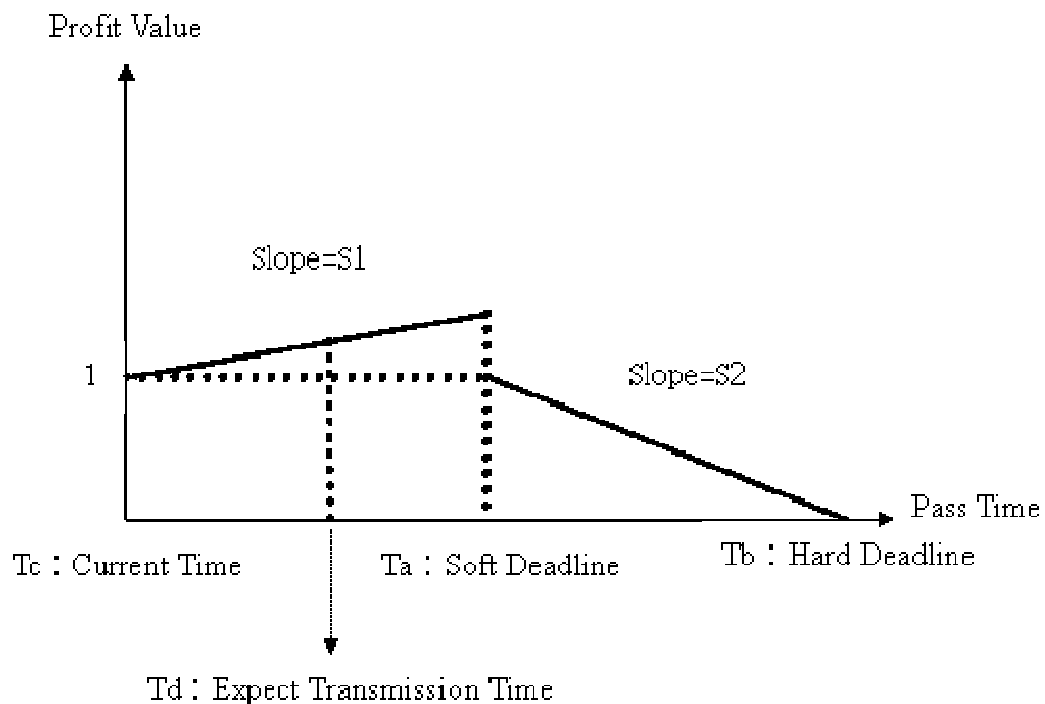


圖 7 : Profit Function 4



$$\text{Profit} = 1 + Td * S1 \text{ when } Td \leq Ta$$

$$\text{Profit} = 1 + (Td - Ta) * S2 \text{ when } Ta - (1/S2) \geq Td > Ta$$

$$\text{Profit} = 0 \text{ when } Td > Ta - (1/S2)$$

圖 8 : Profit Function 5

### 3.3 排程演算法

#### 3.4.1 Single Preemptive Queue 排程演算法

針對一個 Router 的其中一個 output Queue，當 output Queue 為 single Queue 的情況下，假設每個封包的大小都一樣，封包可以插入 Queue 的任意位置，且每一個封包有與傳遞時間相關的 Profit Function，將所有的封包排程，使封包送出的順序所得到的總獲利值能最大，最終以達到改善接收端之 Jitter 為目的。

但是以此方式對封包排程，卻是一個 NP-Hard 的問題，以下我們將加以說明。為此我們必須先介紹何謂 Jensen' s benefit Function，在[18]中提到，Jensen' s benefit Function 是一種考慮到工作的 deadline 以及其重要性對於整體系統影響的利益函數。



我們設計的 Profit Function 便是一種 Jensen's benefit Function。而在[19]中更進一步提到，當我們要設計 non-preemptive 的封包排程演算法時，若是使用 Jensen's benefit Functions，並以得到最大化獲利為目標，則這是一個 NP-Hard 的問題。由於 preemptive 的封包排程演算法比 non-preemptive 的封包排程演算法更為複雜。因此，若是我們使用 Jensen's benefit Function 要設計一個 preemptive 的封包排程演算法時，則為一個 NP-Hard 的問題。

由於我們無法在 polynomial time 內找到解答，因此，我們必須設計 heuristic solution，以在較快速的時間內決定封包送出順序。

Heuristic Solution: output Queue 內原本  $N-1$  個封包順序維持不變，但進入 output Queue 的封包可以任意插入此順序之中，最後選取其中總獲利和最大的方式。此方法需要  $N$  次的計算。如下圖 10 所示。

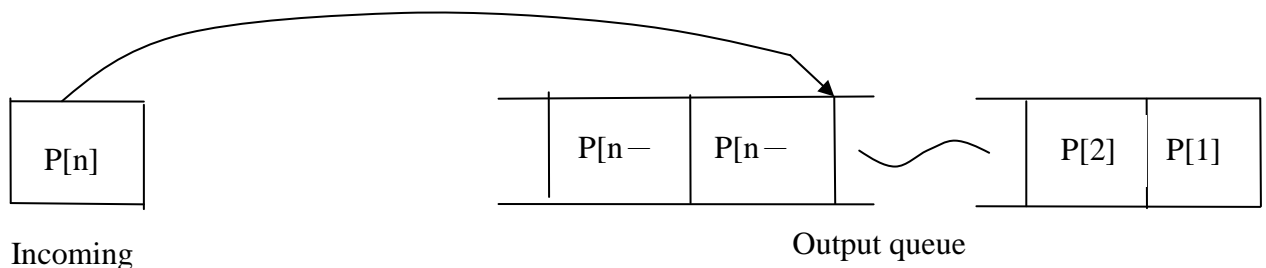


圖 9: Heuristic Solution

### 3.4.2 Multiple Queue 排程演算法

針對一個 Router 的其中一個 output Queue 為 Multiple Queue 的情況下，此時 Multiple Queue 的任一個 Queue 對封包都採用 FIFO (First In First Out) 的原則。我們排程演算法將採取輪詢 (Round Robin) 的方式，對每個 Queue 都服務過一次為一次輪詢，我們將 Queue 裡面的封包依序全部送出稱為對 Queue 服務一次。採用此法的目的在於能

夠近似 Single Preemptive Queue 的封包傳遞順序，以 Multiple Queue 的角度而言，當 Single Preemptive Queue 處理需要插隊的封包時，可以視為將原有在 Single Queue 裡面的封包分為兩段，前半段為沒被插隊的封包，後半段為被插隊的封包，之後將要插隊的封包放置於前半段封包的最後，而前半段則為先被輪詢服務到的 Queue。

當 Multiple Queue 在處理進入的封包時，我們將依照進入封包的 Profit Function 類型，選取獲利值最大的 Queue，將封包置入 Queue 的末端。

## 第四章 模擬及效能評估

### 4.1 模擬與實驗環境

本研究之最後，將以 NS2 模擬器評估我們所提出的方法。實驗的環境如下圖。其中 n0、n1、n2、n3、n4 為 Sender。n9 為 Receiver。n5、n6、n7、n8 為 Router。

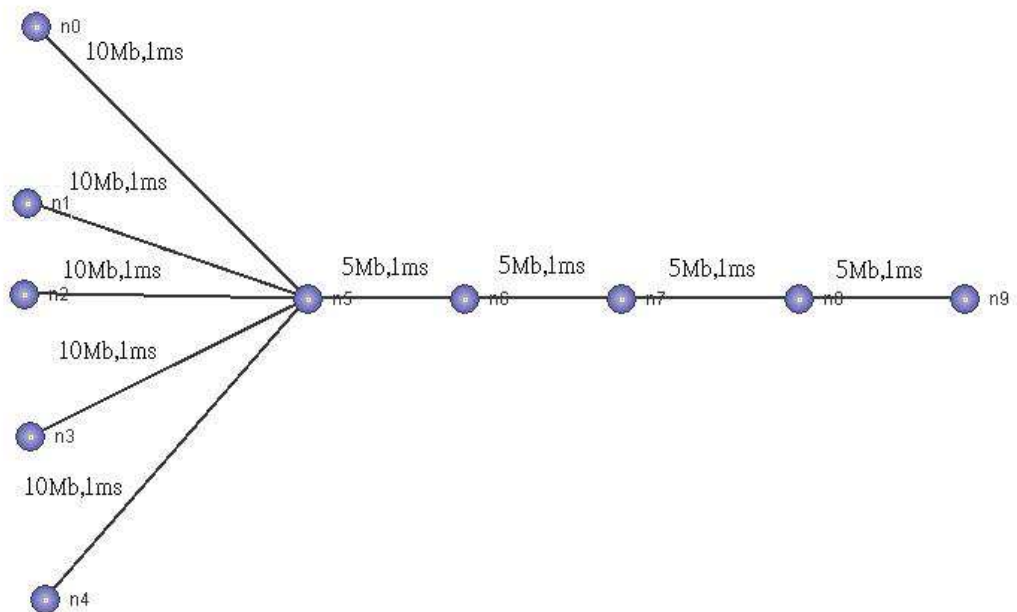


圖 10：實驗拓撲

### 4.2 實驗參數

最後我們將在多個方面來觀察對我們方法的影響，來看是否有效使接收端之 Jitter 絕對值總和能減小。如：

1. 即時性訊務個數。
2. 即時性訊務傳送速率。
3. Router 與接收端的 Buffer 大小。

當網路高負載的時候，若是 Router 與接收端的 Buffer 不夠大時，會造成封包被丟棄，影響整體效能。我們希望封包依原有的步調傳送，以免 Buffer 不足以應付過多太早到的封包。

4. Output Queue 類型。

Output queue 為 Single Preemptive Queue 在實際上不僅硬體實作花費相當龐大並且會減低 router 交換封包的速度，因此，我們必須使用 multiple FIFO queue 來實現我們的方法。

5. 不同的 Profit Function 類型。

對於不同的 Profit Function 得到的封包送出順序可能是不同的，也可能是同樣送出的順序，但是得到的 Profit 是不同的。我們去觀察不同的 Profit Function 對於排程的結果，比較 Jitter 控制的好壞。

### 4.3 評估指標

觀察指標：所有即時性訊務接收端之 Jitter 絕對值總和與封包平均延遲時間 (Average Delay Time)。

由於 Packet Loss 會致使計算出的 Jitter 絕對值總和減小，導致此衡量標準不夠公平，Jitter 變動幅度較大的資料流，可能會因為 Packet Loss，使得計算的結果比 Jitter 變動幅度較小的資料流來的小。因此，我們必須對此計算結果做正規化(normalization)的運算，讓計算的結果比較公平，我們將接收端之 Jitter 絕對值總和除以接收端接收到封包的個數，以做為最後評估時的基準。

### 4.4 實驗設計

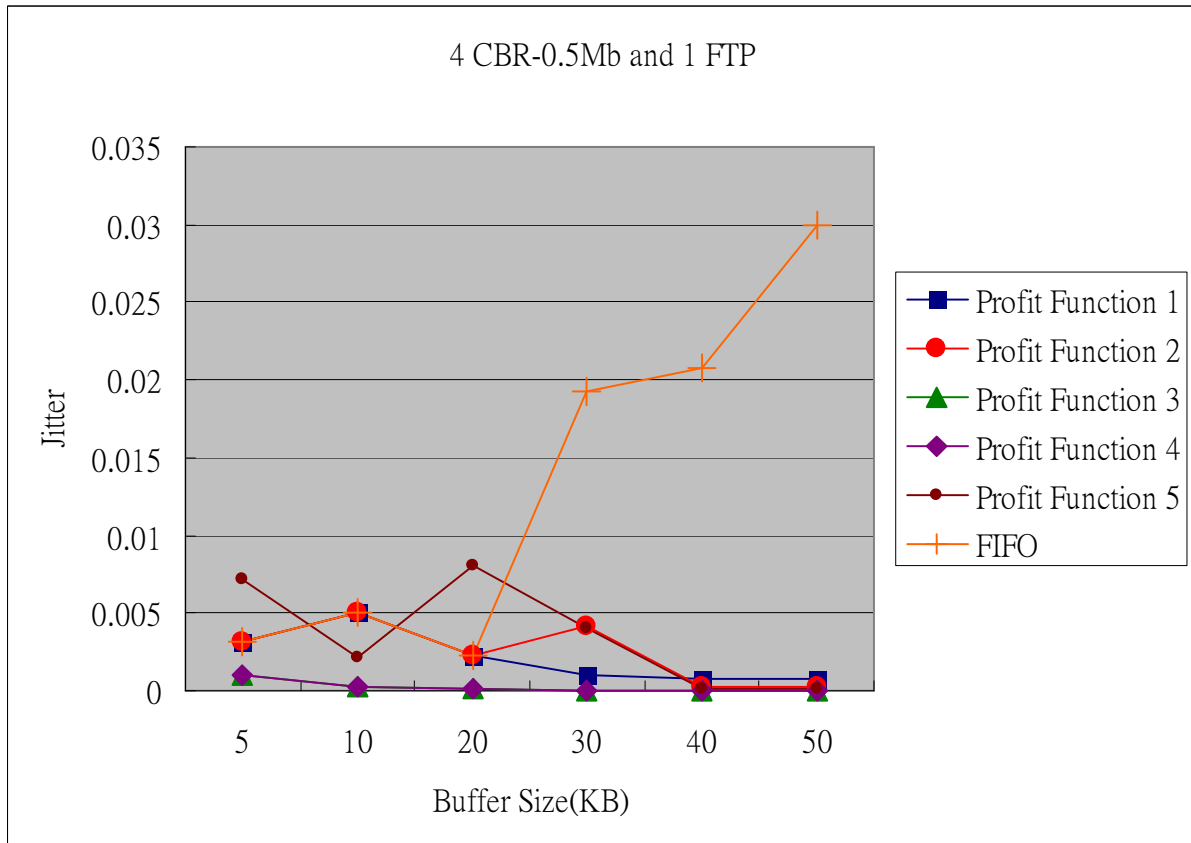
在五個傳送端我們分別有著 real-time traffic 以及 background traffic，並且每個

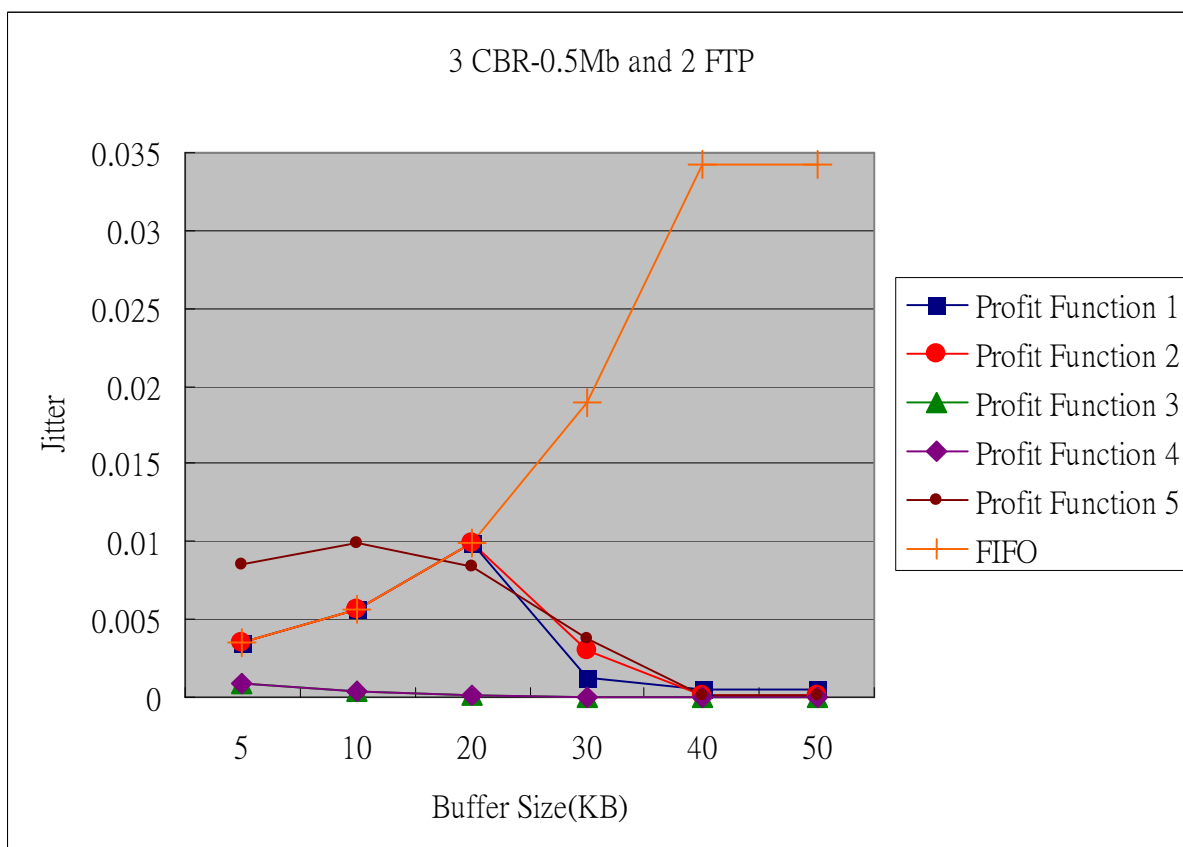
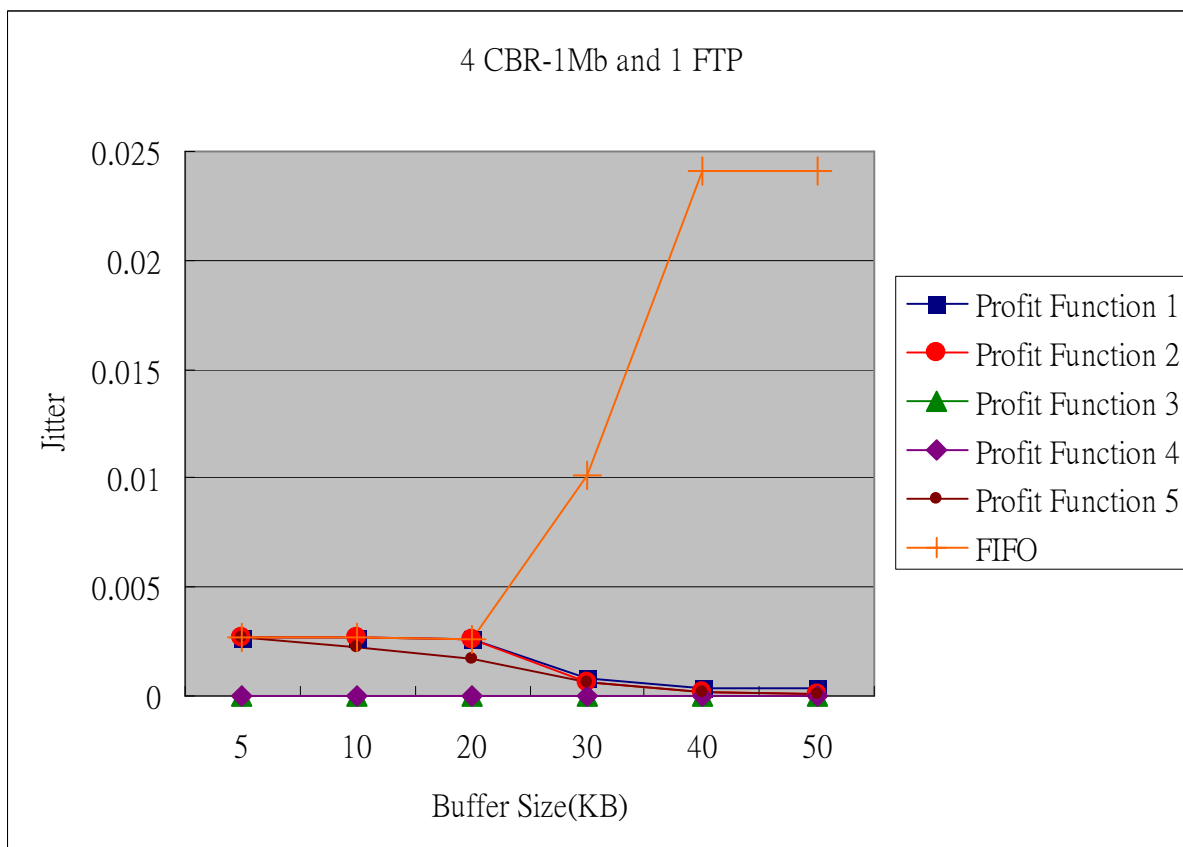
router 都會有 cross traffic 經過。

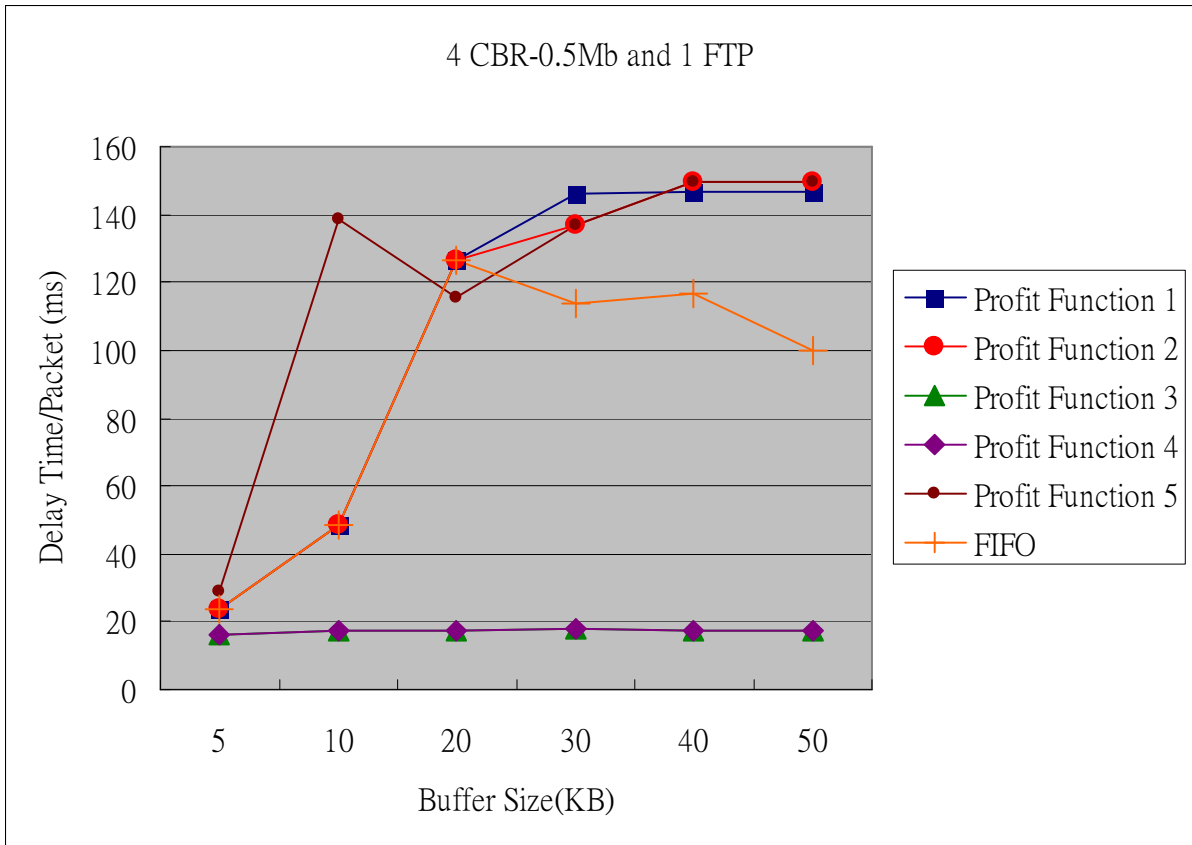
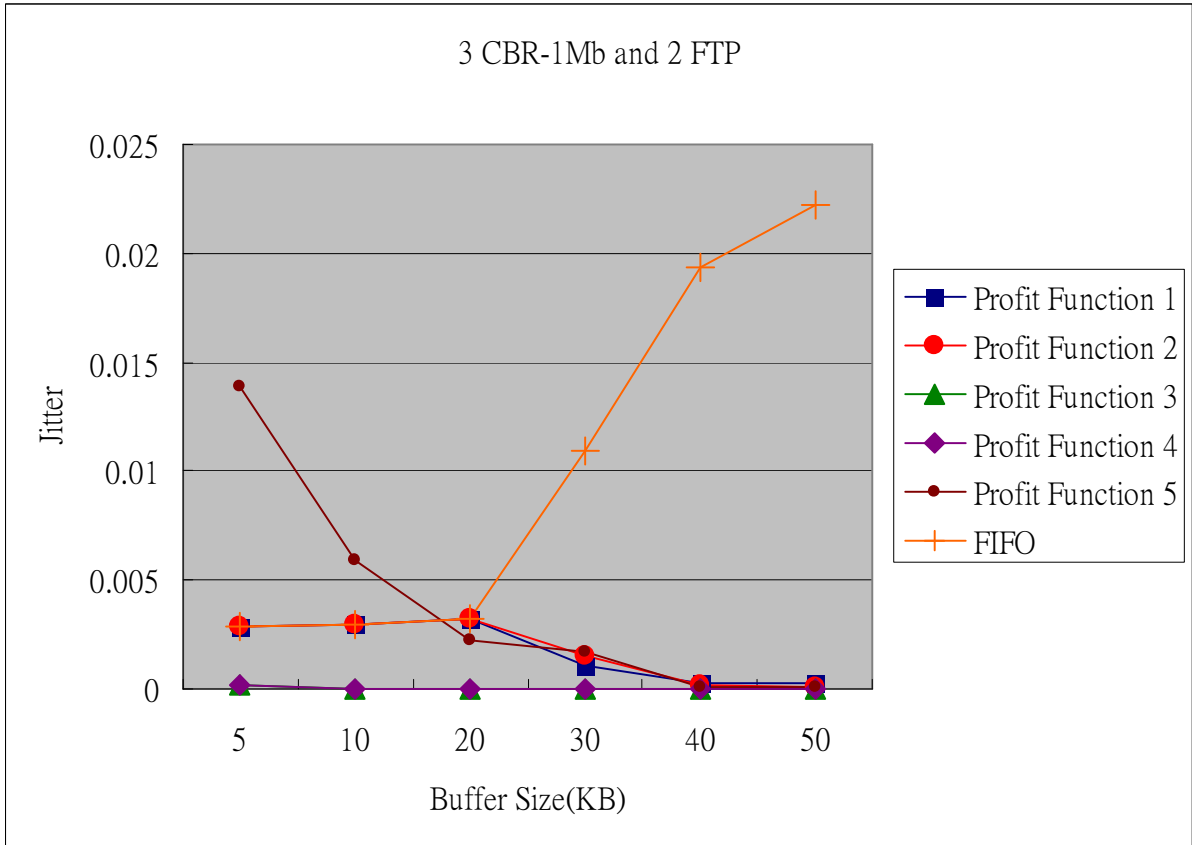
## 4.5 實驗結果

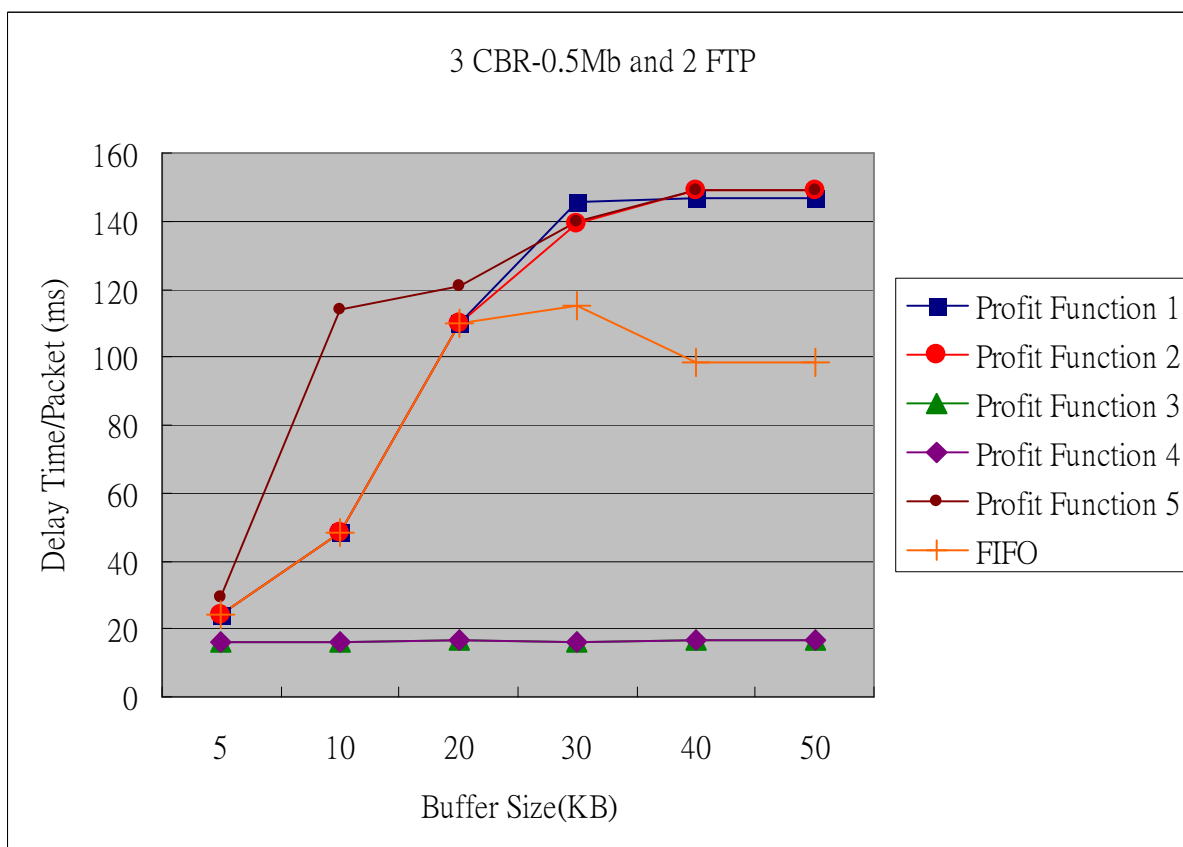
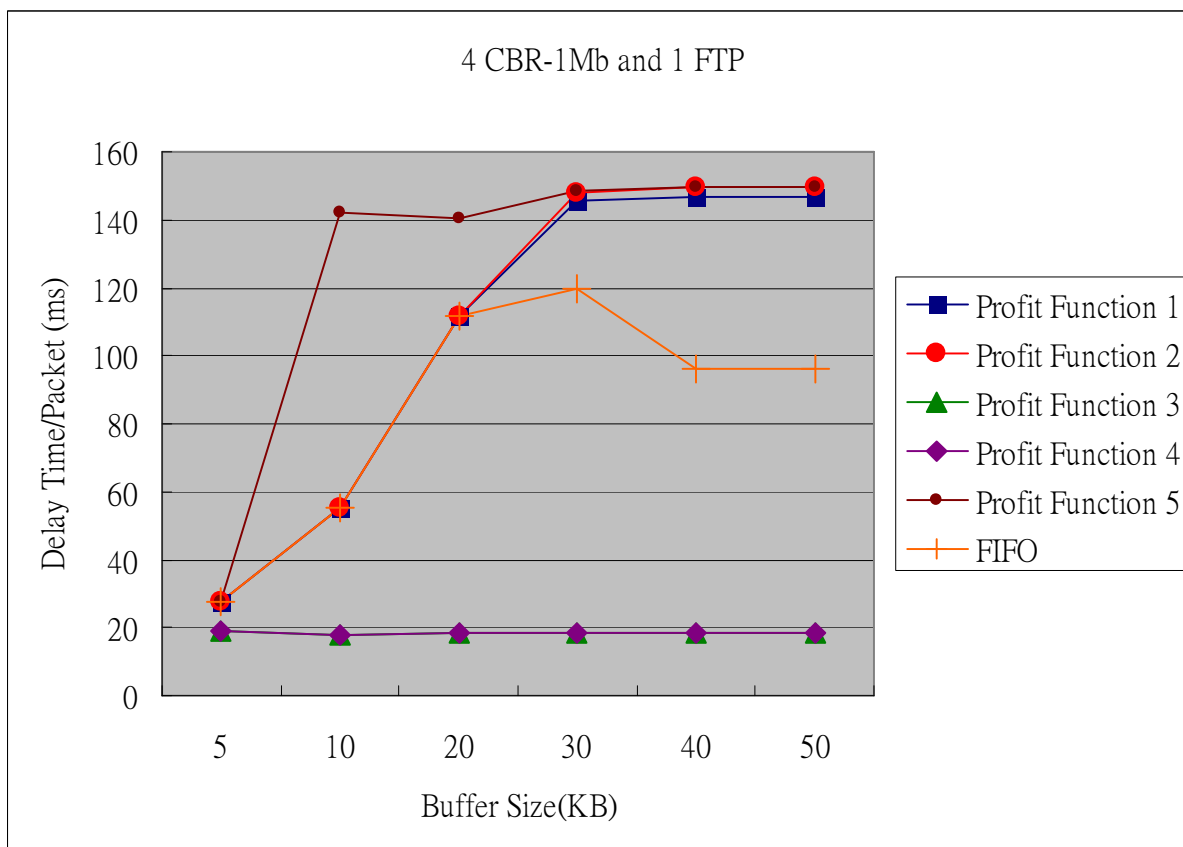
### 4.5.1 Single Class

#### 4.5.1.1 Single Preemptive Queue

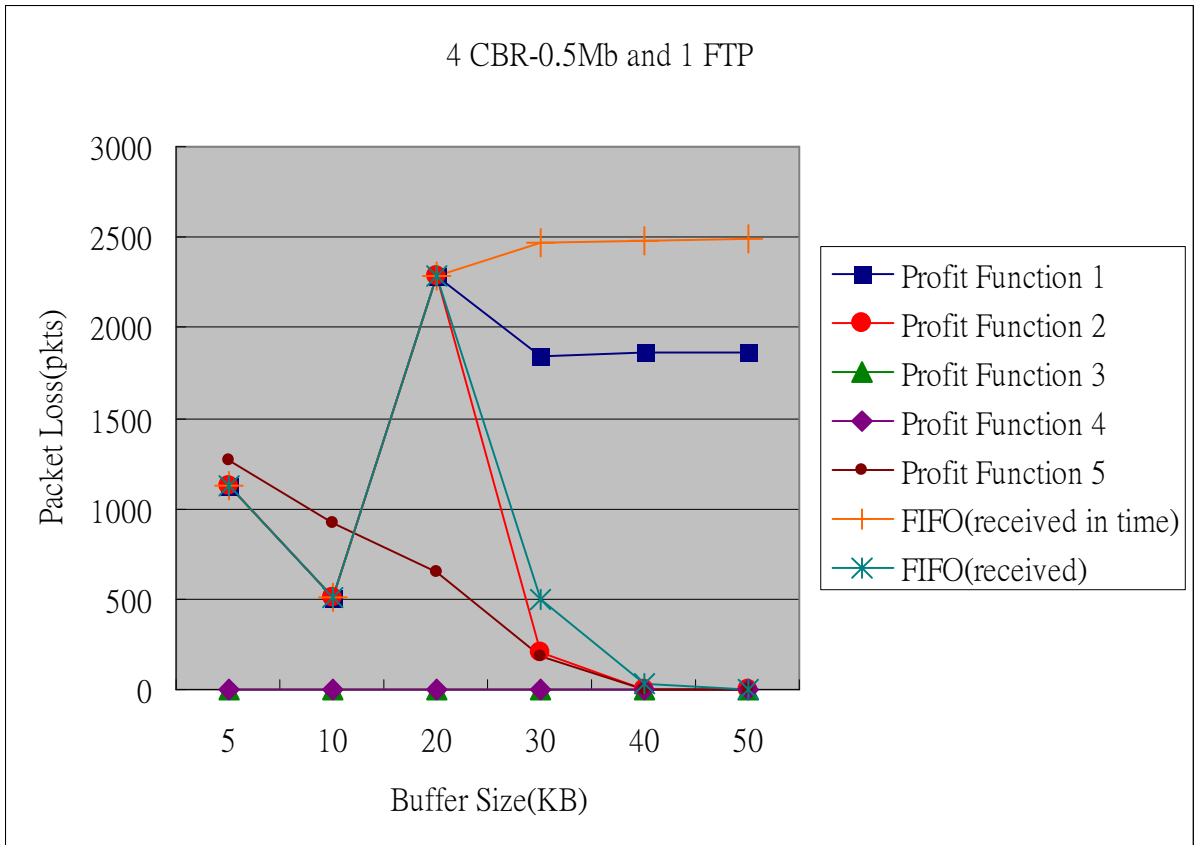
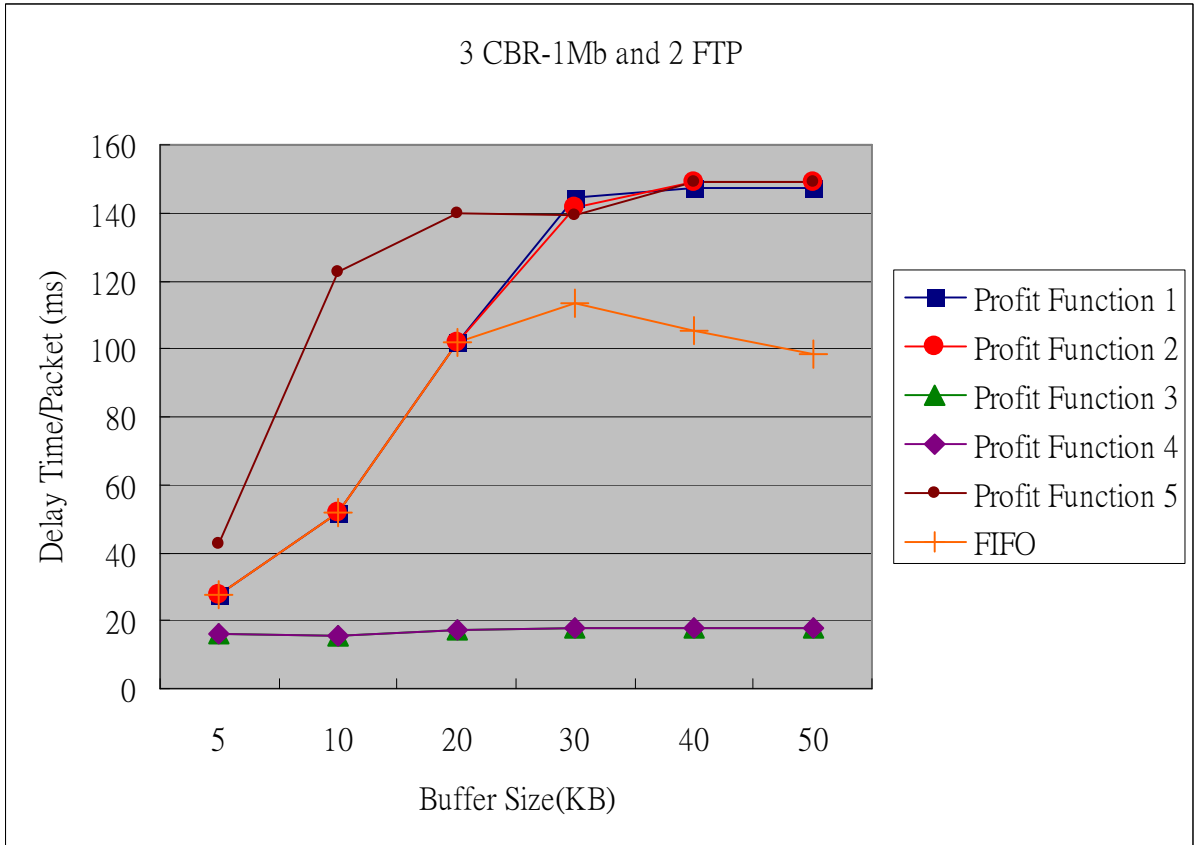


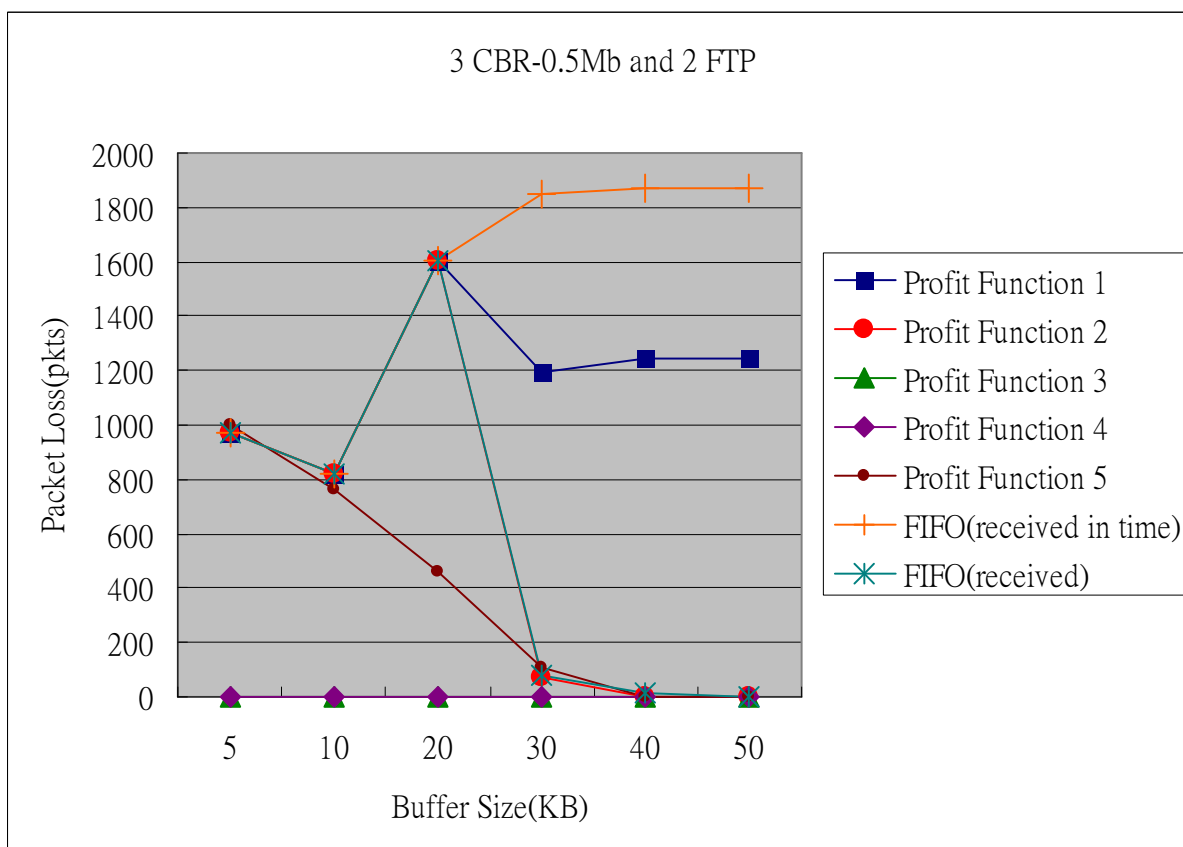
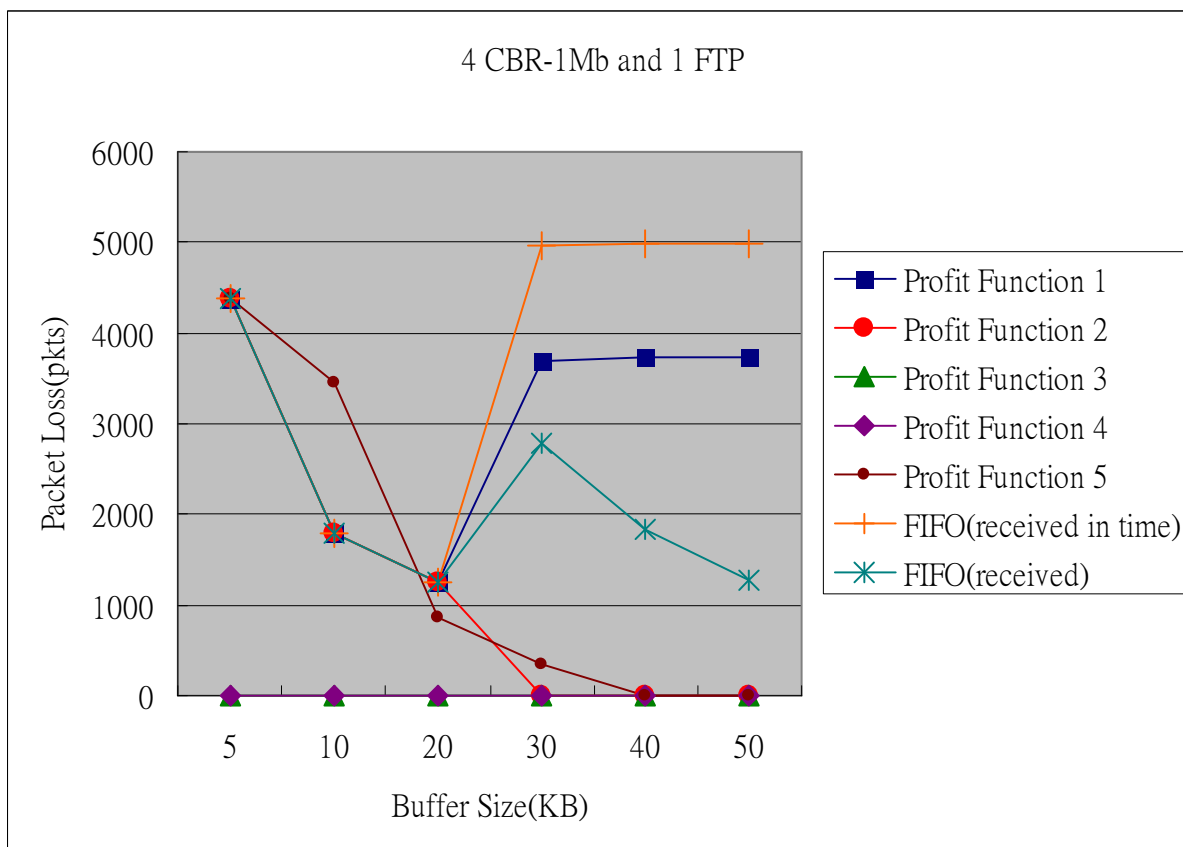


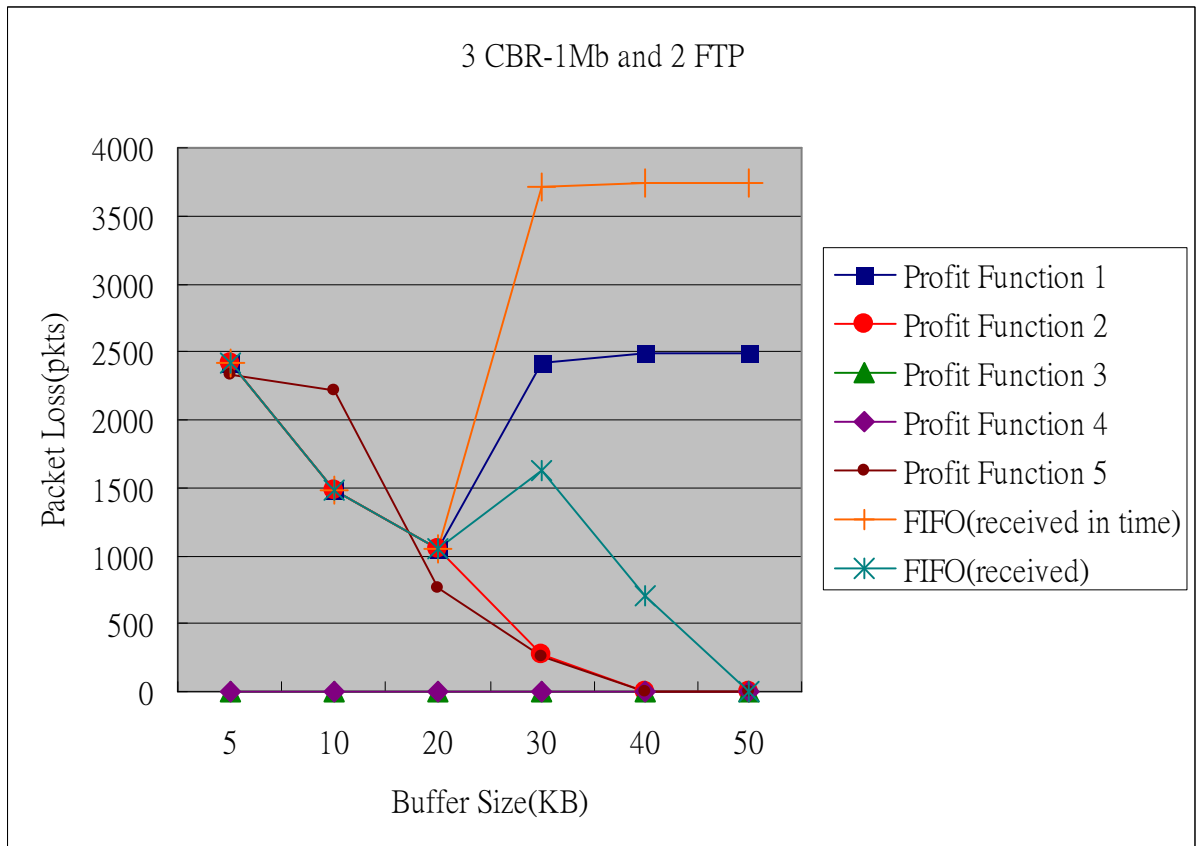




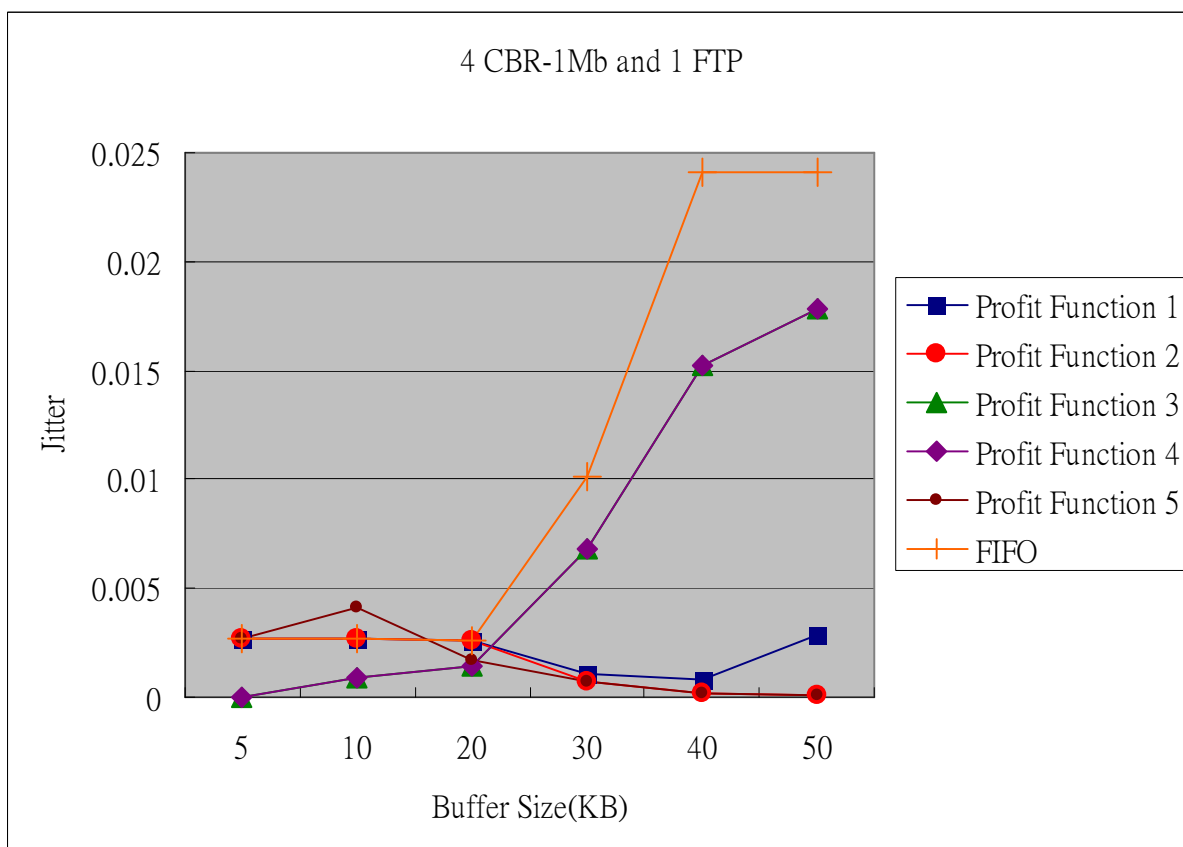
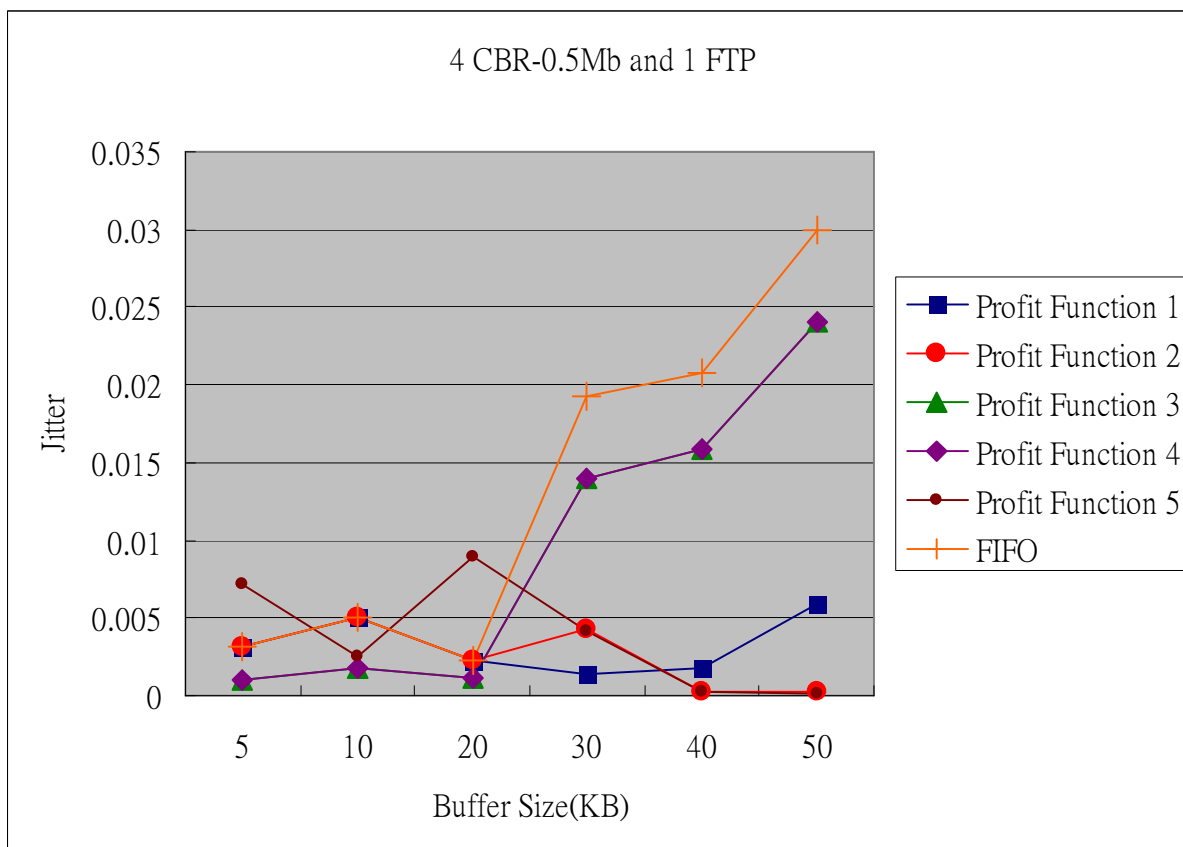


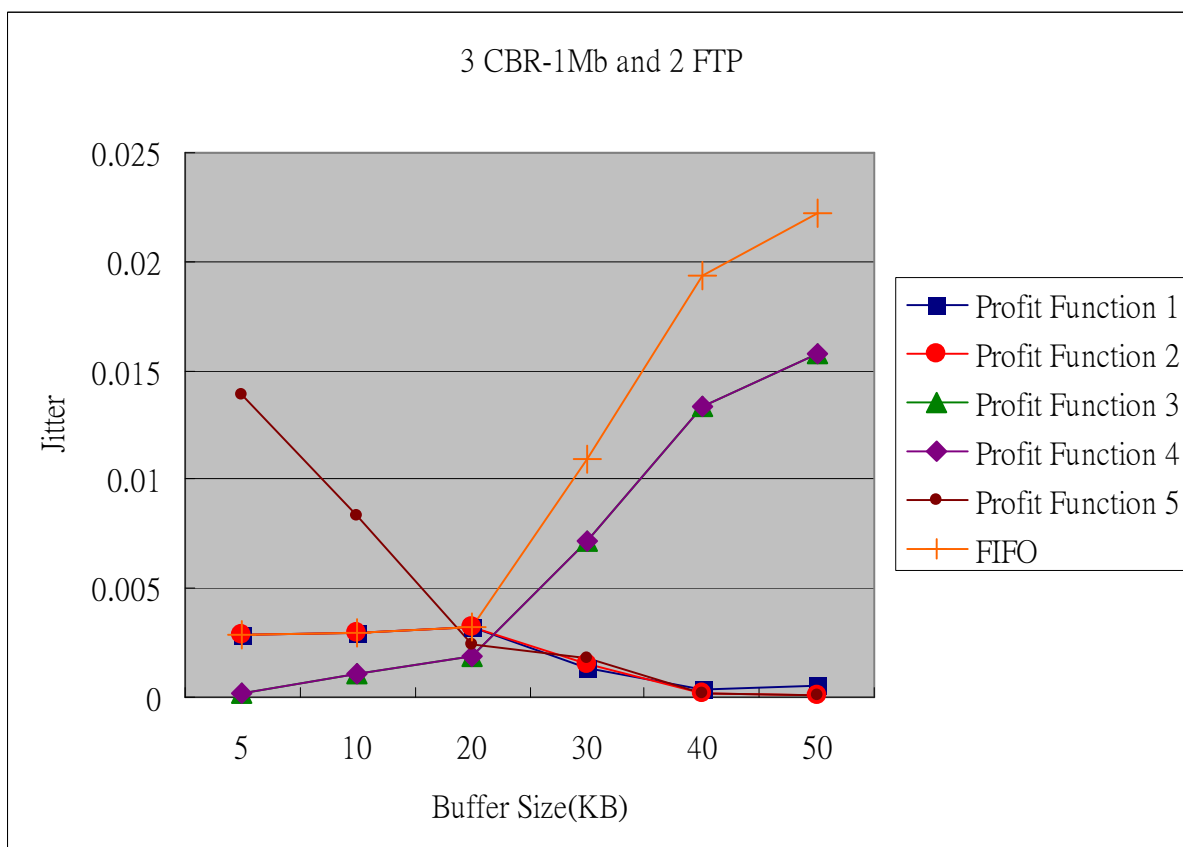
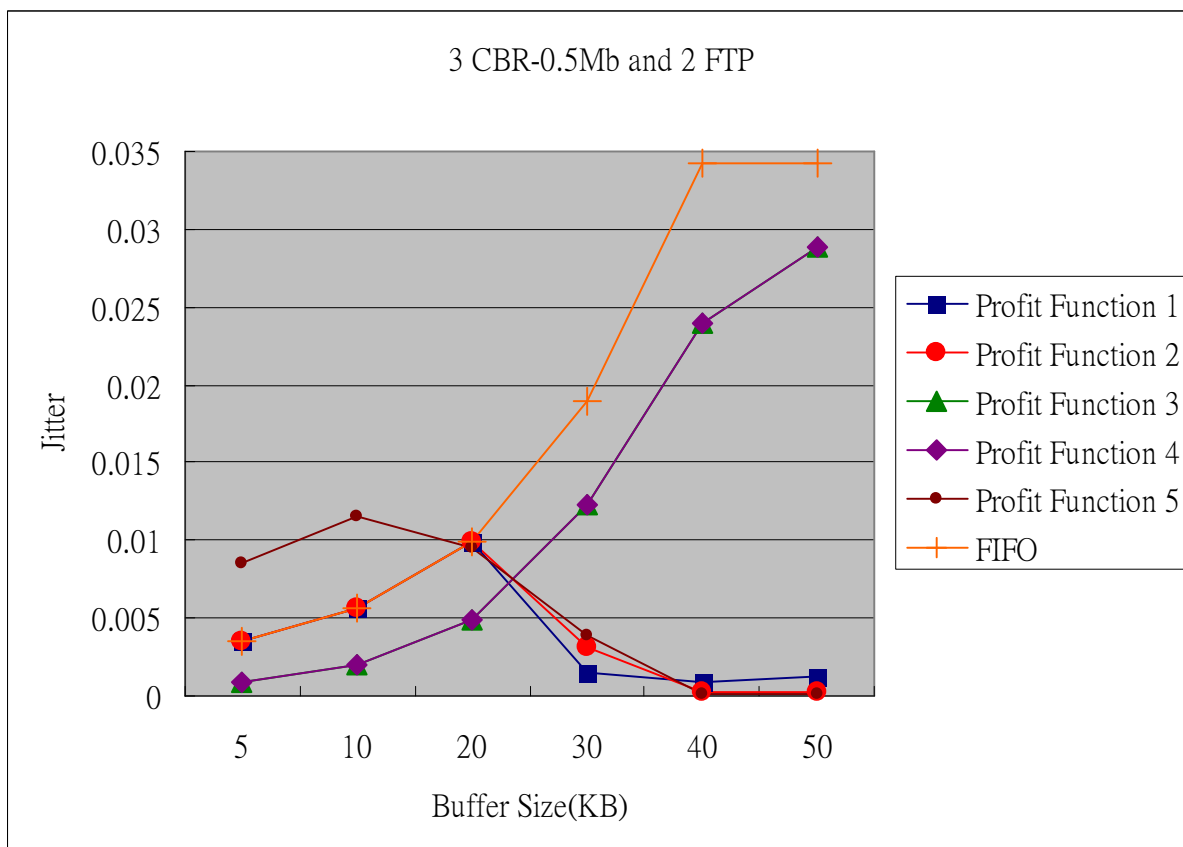


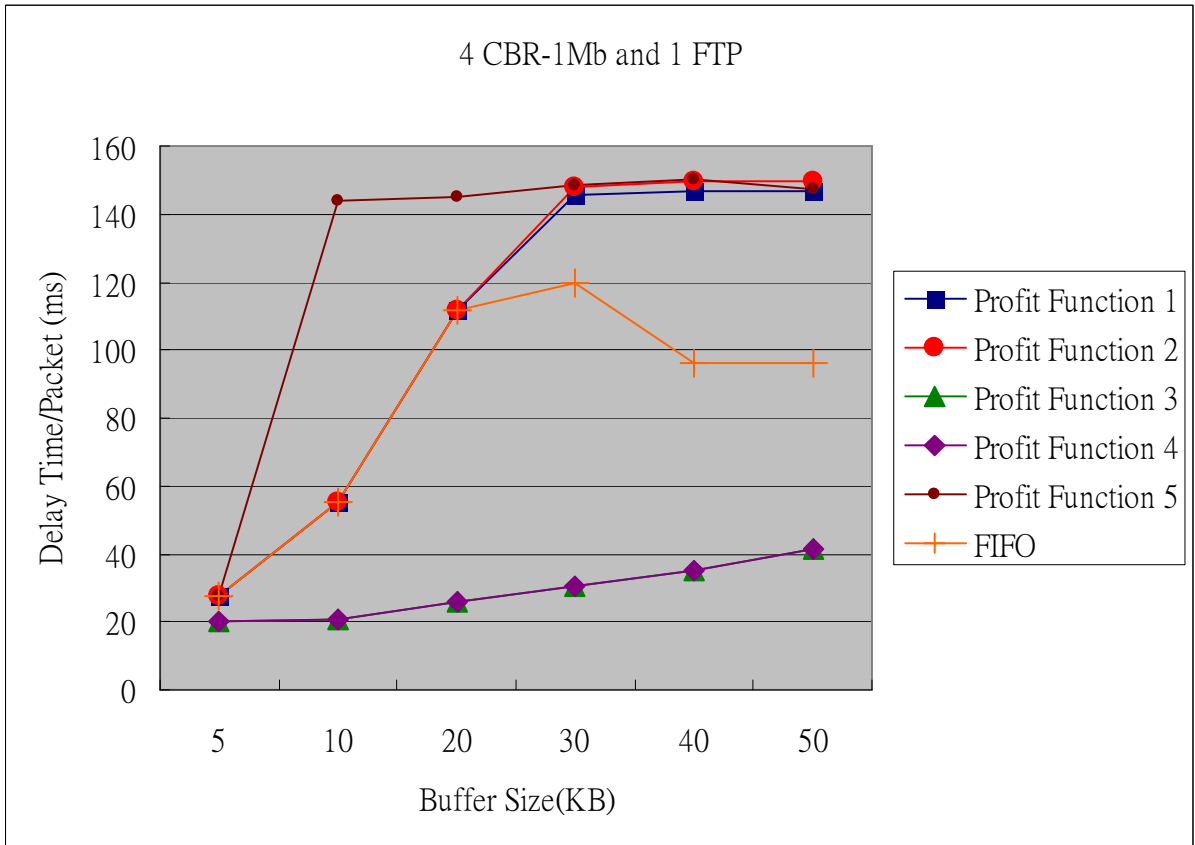
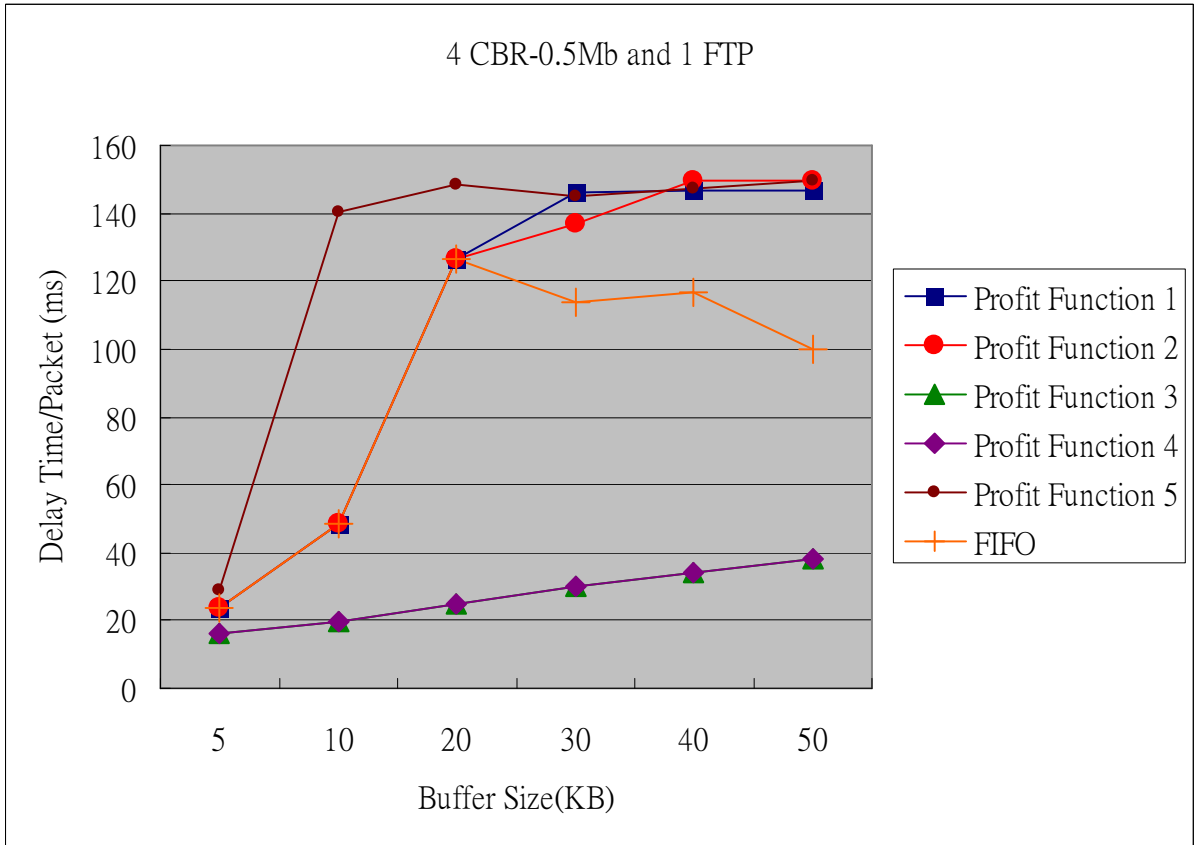


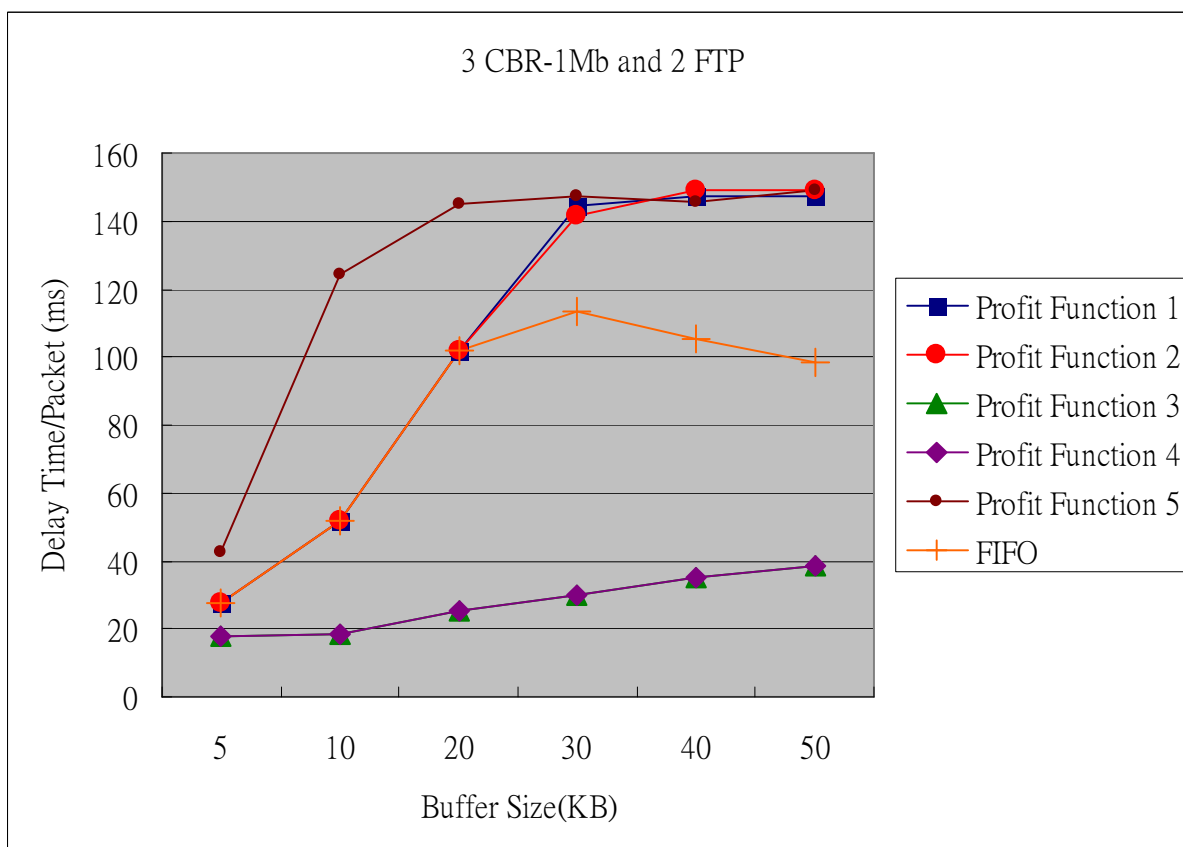
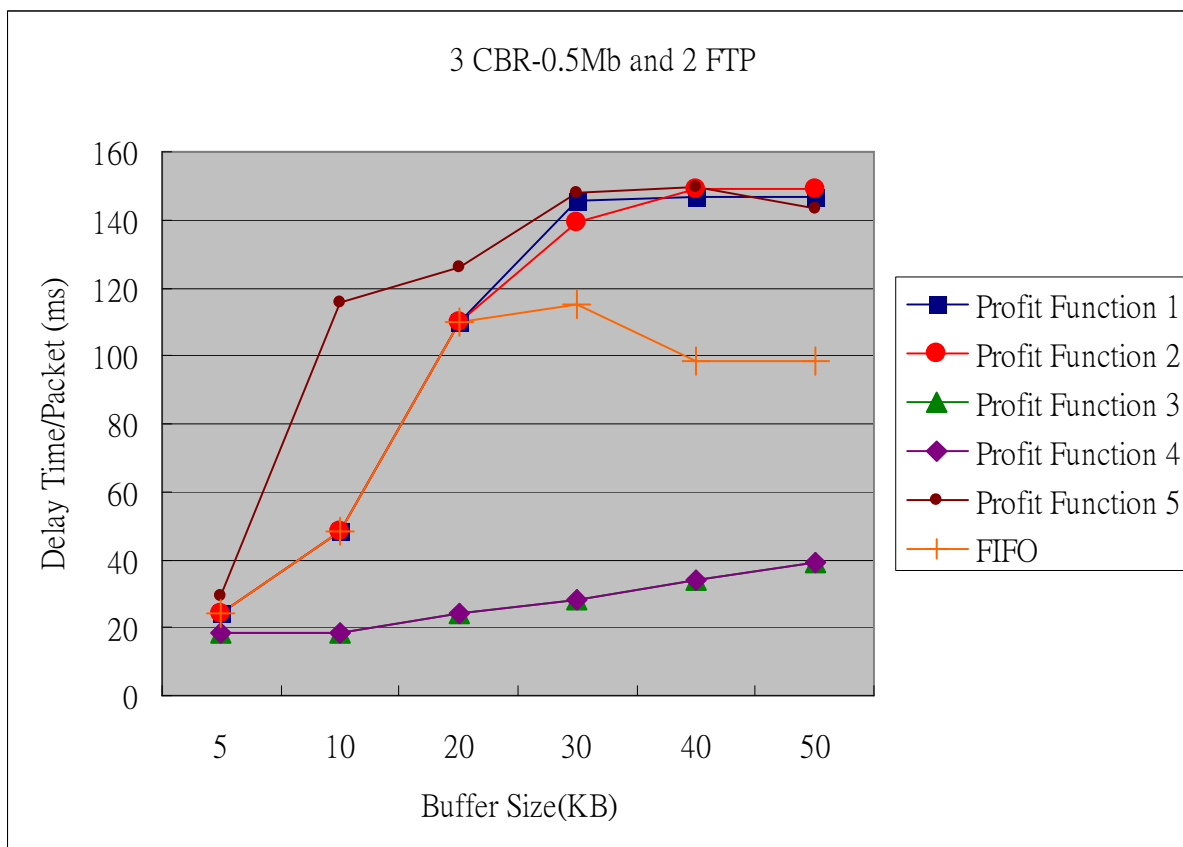


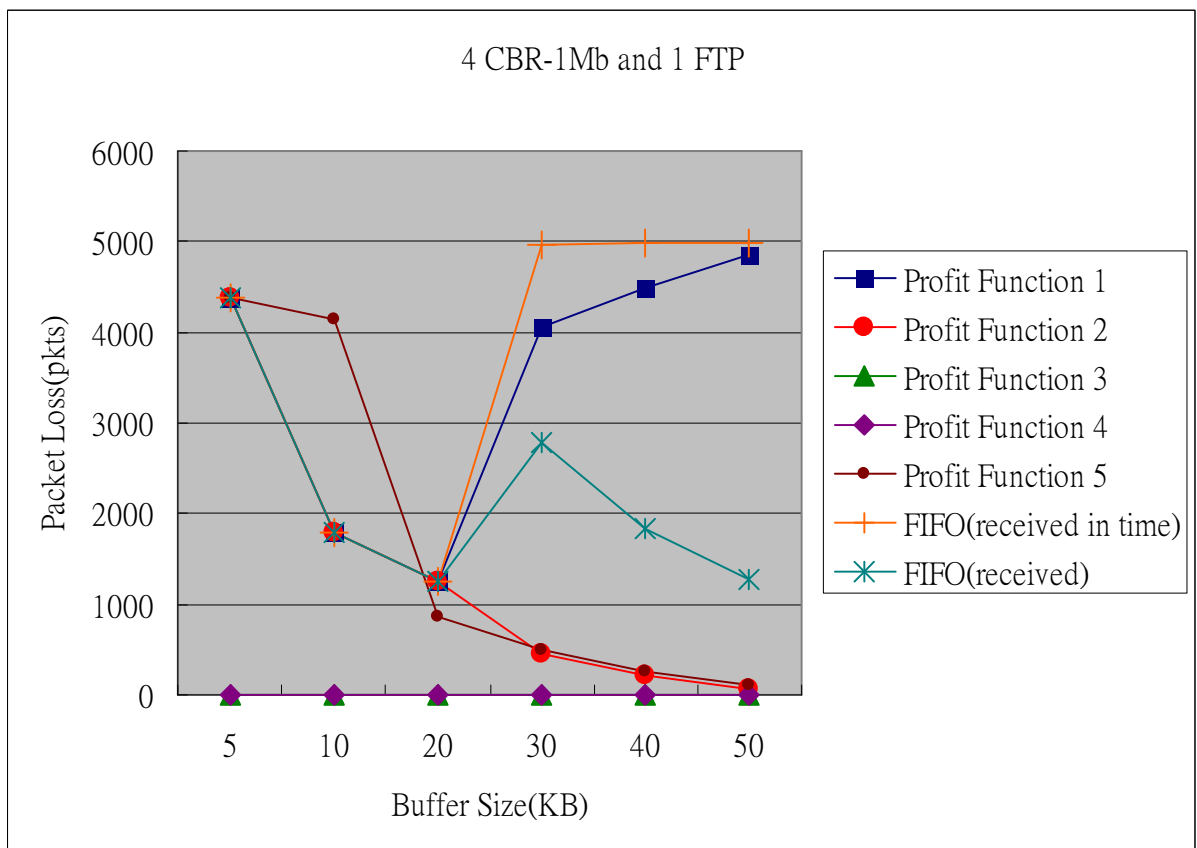
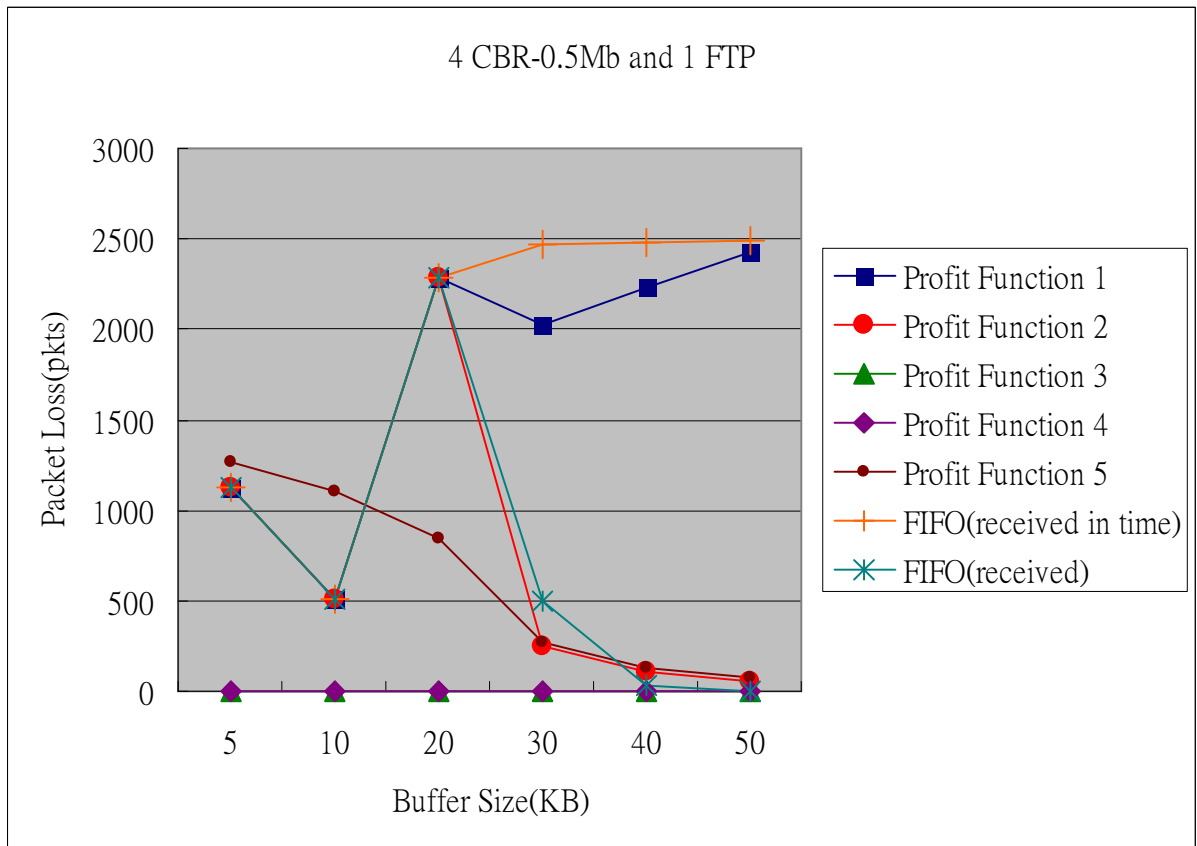
#### 4.5.1.2 Multiple FIFO Queue



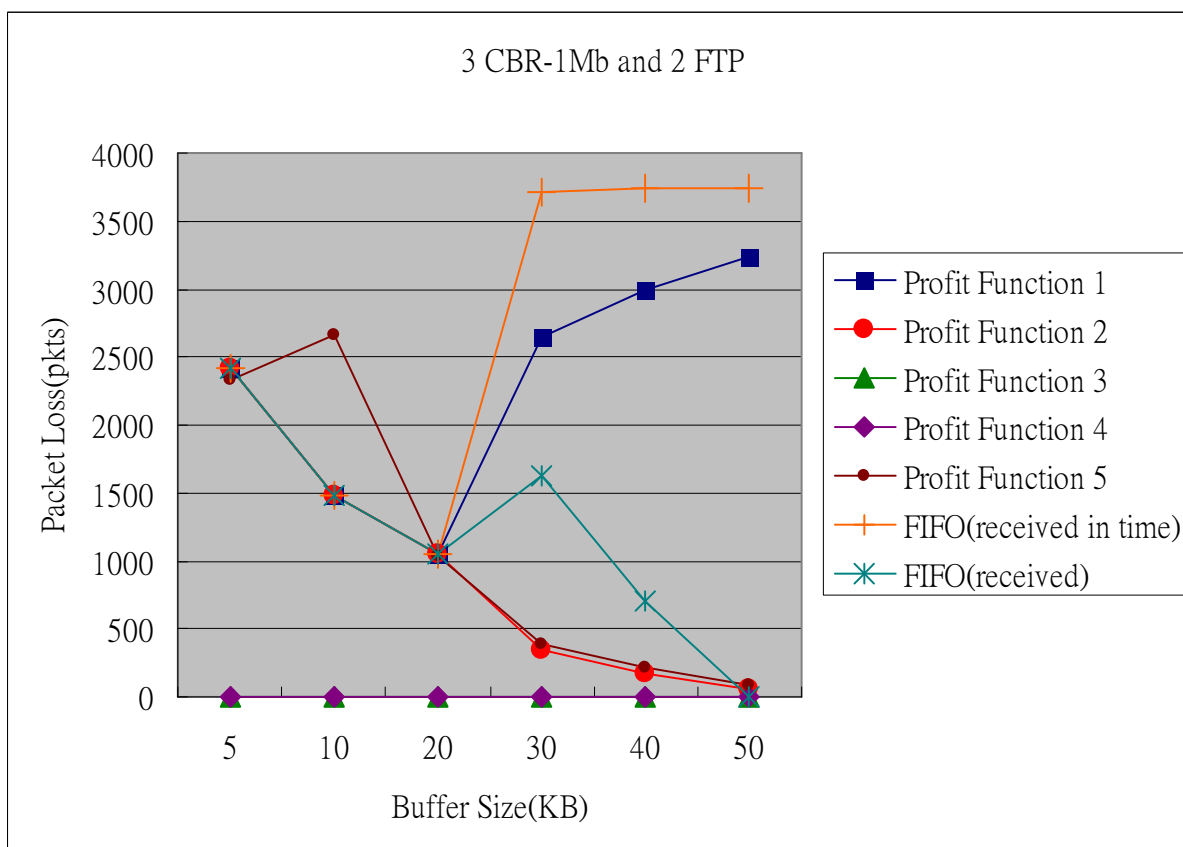
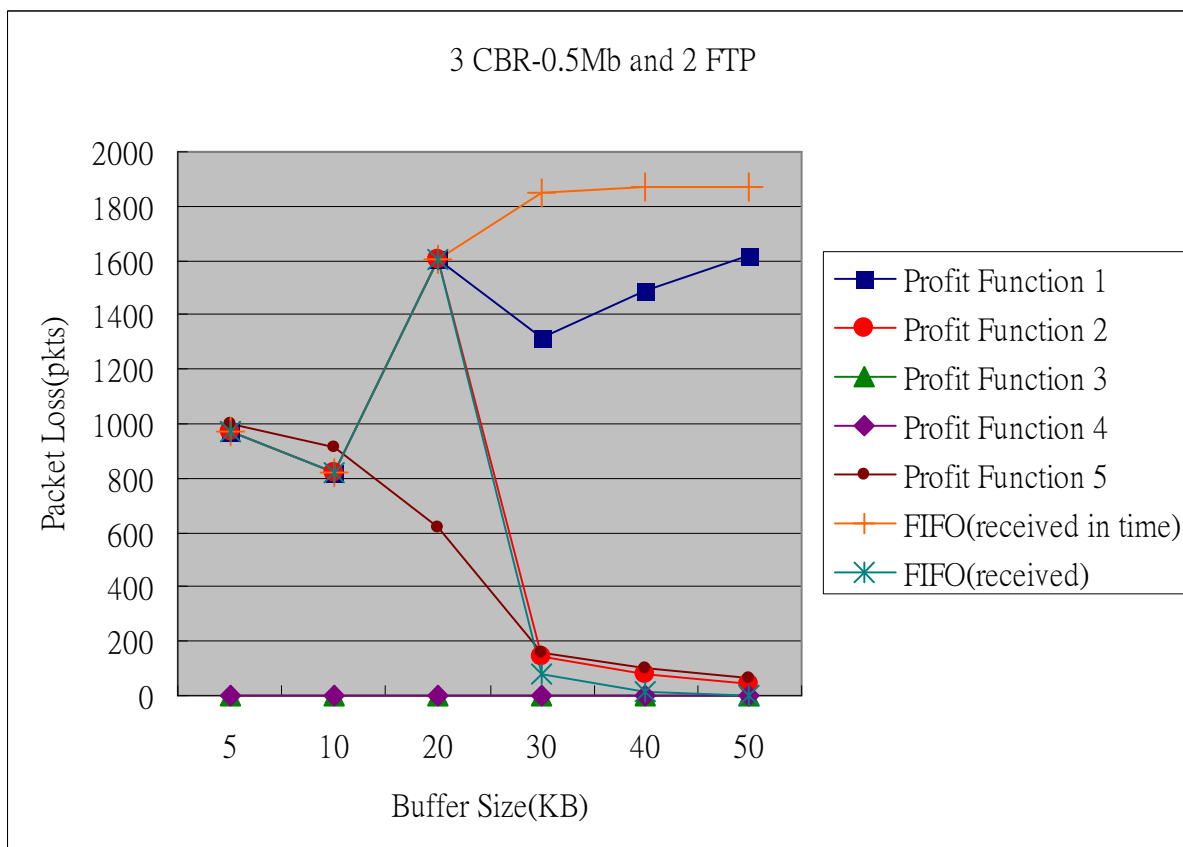












### 4.5.2 Multiple Class

## 4.6 總結

## 第五章

### 結論與未來發展

本研究提出利用與封包傳遞時間相關的 profit function 搭配排程演算法，在每個 router 對封包進行重新排程，改變封包送出順序，透過實驗模擬的結果顯示能有效改善 real-time traffic 的 jitter。

本研究提出的方法仍有不足之處，未來將加以考慮封包剩餘的 hop-count 的因素，以及 Router 排程時，是否有更好的排程演算法可以運用。

## 參考文獻

- [1] 徐毅銘, "在 IPv6 的 DiffServ 網路上具有 Credit/Deficit 調整的延遲變化率減小機制," 國立中山大學.
- [2] Wenjiang Zhou, Chuang Lin, and Fengyuan Ren, "A Model for the Integration of Buffer Management and Packet Scheduling," *Proc. of PDCAT' 03*, pp. 27-29, Aug. 2003.
- [3] Hui Zhang, and Domenico Ferrari, "Rate-Controlled Service Disciplines," *Journal of High Speed Networks*, pp. 389-412, 1994.
- [4] Jon C. R., and Hui Zhang, "Hierarchical Packet Fair Queueing Algorithms," *IEEE/ACM Transactions on Networking*, pp. 675-689, Oct. 1997.
- [5] Qiong Li, and David L. Mills, "Jitter-Based Delay-Boundary Prediction of Wide-Area Networks," *IEEE/ACM Transactions on Networking*, pp. 578-590, Oct. 2001.
- [6] Sally Floyd, and Van Jacobson, "Link-sharing and Resource

- Management Models for Packet Networks,” *IEEE/ACM Transactions on Networking*, Aug. 1995.
- [7] Dimitrios Stiliadis, and Anujan Varma, “Latency-Rate Servers: A General Model for Analysis of Traffic Scheduling Algorithms,” *IEEE/ACM Transactions on Networking*, pp. 611-623, Oct. 1998.
- [8] Aman Kansal, and Abhay Karandikar, “Adaptive Delay Estimation for Low Jitter Audio over Internet,” *Proc of IEEE INFOCOMM*, pp. 680 - 686, June. 1994.
- [9] Ian R. Philp, Klara Nahrstedt, and Jane W.S. Liu, ” Scheduling and Buffer Management for Soft-Real-Time VBR Traffic in Packet-Switched Networks,” *Proc of the 21<sup>st</sup> Conference on Local Computer Networks*, pp. 143-152, 1996.
- [10] Yishay Mansour, and Boaz Patt-Shamir, “Jitter Control in QoS Networks,” *IEEE/ACM Transactions on Networking*, Aug. 2001.
- [11] Hui Zhang, “Service Disciplines For Guaranteed Performance Service in Packet-Switching Networks,” *Proc. of the IEEE*, pp. 1374-1396, Oct. 1995.
- [12] Ion Stoica, and Hui Zhang, “Providing Guaranteed Services

- Without Per Flow Management,” *Proc. of SIGCOMM '99*, pp. 81-94, 1999.
- [13] Jon C.R. Bennett, and Hui Zhang, “WF<sup>2</sup>Q : Worst-case Fair Weighted Fair Queueing,” *IEEE INFOCOM' 96*, March 1996.
- [14] Jon C. R. Bennett, Kent Benson, Anna Charny, and Jean-Yves Le Boudec, “Delay Jitter Bounds and Packet Scale Rate Guarantee for Expedited Forwarding,” *IEEE/ACM Transactions on Networking*, pp. 529-540, Aug. 2002.
- [15] Zheng Wang ,and Jon Crowcroft, “Analysis of Burstiness and Jitter in Real-Time Communications,” *Proc. of the ACM SIGCOMM*, pp. 13-19, Sept. 1993.
- [16] Mark Claypool, and Jonathan Tanner, “The Effects of Jitter on the Perceptual Quality of Video,” *ACM Multimedia '99*, Nov. 1999.
- [17] Yan Xu, Yilin Chang, and Zengji Liu, “Calculation and Analysis of Compensation Buffer Size in Multimedia Systems,” *IEEE Communications Letters*, pp. 355-357, Aug. 2001.
- [18] Tamir A. Hegazy, " Using Application Benefit for Proactive

Resource Allocation in Asynchronous Real-Time Distributed Systems," 2001.

- [19] Jinggang Wang and Binoy Ravindran, " BPA: A Fast Packet Scheduling Algorithm for Real-Time Switched Ethernet Networks," *Proc. of ICPP' 02*.
- [20] Fu-Ming Tsou, Hong-Bin Chiou, and Zsehong Tsai, " Design and Simulation of an Efficient Real-Time Traffic Scheduler with Jitter and Delay Guarantees," *IEEE Transaction on Multimedia*, pp. 255-266, Dec. 2000.