

# 網路路由器服務品質之效能分析

陳延禎 李柏毅

明志科技大學 電子工程系、工程技術研究所

E-mail : [yjchen@mail.mit.edu.tw](mailto:yjchen@mail.mit.edu.tw), [boy@java.mit.edu.tw](mailto:boy@java.mit.edu.tw).

## 摘要

路由器的主要功能為為封包找尋路由並遞送之，在遞送的過程中若對封包流執行排程、丟棄、限速、整流...等等服務品質(QoS)的控制功能，將耗用一定程度的能量，最明顯的反應就是路由器的CPU負載增加；因此，啟用路由器的QoS功能，是需要詳盡的效能數據作為依據，否則不但不能提升品質反而導致根本的封包遞送都不正常。然路由器的硬體效能不斷提升，可依循的數據就不斷在變，能夠不變的是探討QoS效能的方法，本研究旨在提出一效能分析模型，以及依據此模型所訂定的效能測試規劃。此模型係以QoS控制功能為分析核心，在不同的考量條件下(如不同型的路由器、不同特徵的輸入流量...等等)所表現出的效能，而評量效能的標準是以路由器的CPU負載、QoS功能的正確性與狀態回穩所需的收斂時間。整個測試規劃分為基本效能測試與QoS效能測試；前者是為了與路由器原廠所提出的效能數據做一比對，以確認自身的實驗環境是否正確；後者則是進行有系統地測試，再以測試的結果驗證理論的推定，以期說明每一QoS功能的啟動所造成的變化。這將使得企業在規劃網路服務品質時，無須更改既有的網路架構，也無須購置昂貴的QoS控制設備，只須採用具有QoS功能的路由器即可。

**關鍵詞：**頻寬、路由器、QoS、ACLs、BAD

## 1. 簡介

我們針對市場85%佔有率的Cisco路由器進行測試與分析其執行服務品質的效能與功能的正確性，我們建置了一效能分析模型並根據此模型規劃實驗測試路由器的執行效能。評量點是在執行QoS功能時，路由器在封包漏失率、系統負載與功能正確性等方面的表現。本研究的貢獻在於我們提供網路路由器執行QoS功能時之效能數據，而網路QoS規劃工程師可依此數據作網路品質服務之規劃與建置。無形的貢獻在於提升了對網路設備路由器的QoS測試相關經驗。

本論文共分為五節：第一節是簡介，先提列本研究之研究動機、目的與研究貢獻與論文架構。第二節是背景知識，介紹路由器的封包交換模式。第三節是研究方法，提出一效能規劃分析模型，依據此模型做相關之效能評估。第四節實驗結果與數據分析，針對所得數據做分析，並定義頻寬分配公

平性的偏差度，以評估頻寬分配的正確性。第五節結論，綜合前述之理論效能分析模型與實際實驗規劃經驗與效能數據分析過程，做完整的結論。

## 2. Cisco 路由器封包交換模式

Cisco 路由器有三種封包交換的模式，分別是Process Switching (PS)、Fast Switching (FS)與Cisco Express Forwarding (CEF) [7][8]。管理者可設定路由器在其中一種模式以執行封包的遞送。在PS模式下，路由器的CPU會將進入的封包自其輸入介面的貯列搬到主記憶體中，然後解讀封包並根據其目的地位址搜尋路由表，在確定輸出介面後，檢查該介面是否有設置出入控制條件[9]，若有則檢視該條件是否允許封包通過，若是則將該封包自主記憶體中搬到輸出介面貯列等待輸出。FS則是將封包的路由資訊記載在一路由快取記憶體(Route Cache)中，當封包進入路由器時，CPU根據封包的目的地位址搜尋cache，若找不到則去讀路由表，並將路由資訊存在cache中，若找得到，在確定輸出介面後，檢查該介面是否有設置出入控制條件，若有則檢視該條件是否允許封包通過，若是則將該封包自輸入介面貯列移到輸出介面貯列等待輸出。很明顯的，FS的封包遞送效能遠比PS高，一則是因為封包不用搬到主記憶體，二則是因為查尋cache是比查路由表要快得多。但Route Cache與路由表有可能不同步，因此有CEF以改善這個問題，由於在路由表不大時FS與CEF的效能幾乎一致，而本研究目前的測試路由極少，因此此處僅以FS做路由器效能測試。

## 3. 研究方法

本論文提出一系統化的方法以分析路由器的QoS效能[1~6]；首先，建立一QoS效能分析模型；其次，根據此模型訂定QoS測試規劃。QoS效能分析模型乃以Cisco路由器的QoS功能為本，探討影響QoS效能的條件，以及評量QoS效能的標準。圖1是此模型之架構，其細節將詳述於後。QoS測試規劃先以基本效能測試(Baseline Test)對照Cisco官方公佈的效能數據，藉此印證本研究的測試環境是正確的。而後進入QoS效能測試(QoS Test)，以分析QoS功能執行後對路由器CPU Load的衝擊，以及探討QoS功能執行的結果是否正確，還有隨著考量條件的改變(如輸入流量的改變)，QoS功能的反應時間是否夠短。此外，所有測試利用兩種工具

觀察流量的行為，一是 Cisco 發展的 QoS Device Manager(簡稱 QDM)軟體，透過 QDM 可以觀察路由輸入流量的速率及路由器本身的 CPU 使用率；另一是 SnifferPro，透過其擷取封包的功能，可以觀察出封包與封包的間隔時間，藉以觀察分配頻寬達穩定時的收斂時間(Convergence Time)。

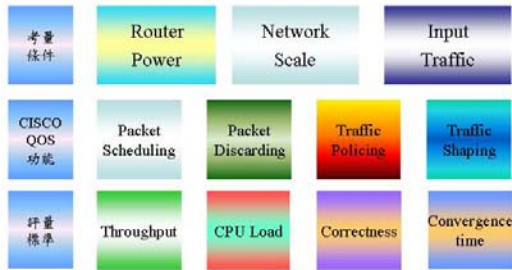


圖 1 效能分析模型圖

### 3.1 QoS 效能分析模型

如圖 1 所示，此模型由三大部分組成：(1) 考量條件、(2) Cisco 之 QoS 功能、(3) 評量標準。

#### (1) 考量條件

考量條件有 Router Power、Network Scale 與 Input Traffic。Router Power 指的是路由器遞送封包的速率，單位是 pps (packet/sec)；Cisco 路由器大致分為大、中、小、微等四型，分別對應於 Cisco 7000、3800/3700、2800/2600、與 1700 等系列，不同的機型所執行的 QoS 效能必定不同，其差異是分析的重點，由於設備取得不易，本研究僅針對 C2600XM 系列做完整之測試，其官方所公佈之封包遞送速率如圖 2 所示，這是未執行 QoS 功能的效能，也就是基本效能，而此論文將揭露執行 QoS 功能的效能。

Platform	MMs	AIMs	WICs	Fixed LAN Ports	Performance (Kpps)	DRAM (Default MB/Max MB)	FLASH (Default MB/Max MB)	Included IOS Feature Set
CISCO2610/31	1	1	2	1 FE/2 FE	15	32/24	8/10	IOS IP
CISCO2610XM/11XM	1	1	2	1 FE/2 FE	20	32/28	16/48	IOS IP
CISCO2612	1	1	2	1 FE/2 FE	15	32/24	8/10	IOS IP
CISCO2620/21	1	1	2	1 FE/2 FE	25	32/24	8/12	IOS IP
CISCO2620XM/21XM	1	1	2	1 FE/2 FE	30	32/28	16/48	IOS IP
CISCO2650/51	1	1	2	1 FE/2 FE	32	32	8/12	IOS IP
CISCO2650XM/51XM	1	1	2	1 FE/2 FE	40	64/28	16/48	IOS IP

圖 2 Cisco 公佈之 2600 series 相關數據

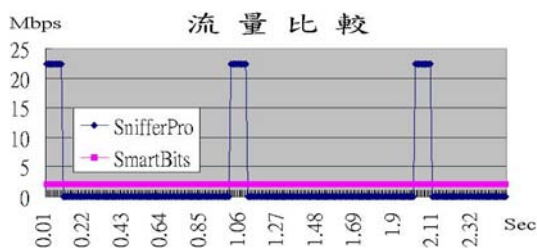


圖 3 SmartBits 與 SnifferPro 產生之流量比較

Network Scale 網路尺度這個條件主要是影響網路在延遲時間方面的服務品質，但因設備不足，無法建立系統化的測試，在此次研究中並未列入，將探討在未來的研究中。

在 Input Traffic 方面，以兩種設備 SmartBits 與 SnifferPro 的流量產生器來產生平緩的(constant)與突起的(bursty)流量。SmartBits 型號為 SmartBits 2000，上有兩張 ML-7710 網卡，速率為 100Mbps，設定網卡速率的分量，可產生所需的流速，例如：設定流速為網卡速率 2%，則產生 2Mbps 之流量，此研究中以 SmartBits 產生 constant 流量。SnifferPro 控制流速的方法有兩種，一是設定相鄰封包間的延遲時間，例如：以一封包為母持續複製封包，封包與封包間的延遲設為 1ms，則封包產生速率為 512pps，若延遲設為 2ms，則封包產生速率為 341pps，這速率是 SnifferPro 內部設定的，與封包大小無關，延遲的單位是 ms，延遲值只能是整數；以這種方式產生的是 constant 流量；另一是設定網卡傳輸速率的分量，例如：以一封包為母持續複製封包，設定網卡速率的 2%，若網卡速率為 100Mbps 則封包產生速率為 2Mbps，經實測，這種方式產生的是 bursty 流量。圖 3 是 SmartBits 與 SnifferPro 對 100Mbps 的網卡取 2%分量以產生 2Mbps 流速，SmartBits 的流量平緩，但 SnifferPro 卻在前 0.088s 將每秒 2Mbps 的量以 22.5Mbps 的速率送出。這兩類型流量對路由器的 QoS 效能也產生不同的衝擊。

#### (2) Cisco 的 QoS 功能

常用的 Cisco QoS 功能有四類[10][11][15]，如圖 1 所示，這是本研究測試與分析的焦點：Packet Scheduling 對不同的封包流施以頻寬或優先序的控管，Cisco 主要的 packet Scheduling 有 First In First Out (FIFO)、Priority Queueing (PQ)、Custom Queueing (CQ)、Weighted Fair Queueing (WFQ)與 Class-Based WFQ (CBWFQ)[12]，FIFO 是指路由器讓先進入的封包先被送出；PQ 是讓具又高優先權的封包先被送出，PQ 啟動時，Cisco 定義四個 queues，優先權分別是 High, Medium, Normal, Low，較高優先權 queues 內的封包被送完後，較低優先權 queues 內的封包才被送出；CQ 是針對封包流進行粗略的頻寬分配，而 WFQ 則是進行較精準的頻寬分配，因而取代 CQ，而 CBWFQ 更進一步提供管理者定義流量類別(Class)，並在該類別指定保障頻寬，其總合必須小於出口介面之頻寬，有流量的類別會獲得保障額度的頻寬，用不完的類別，將頻寬繳回，形成剩餘頻寬，剩餘頻寬以各類別的保障頻寬為權重，成比例再分配給各類別。例如：有三類穩定且持續的流量 C1、C2、C3，流速分別是 5、2、4 Mbps，他們通過路由器的一出口介面，頻寬是 8Mbps，三者所需的保障頻寬分別是 1、3、2 Mbps，則以 CBWFQ 分配到的頻寬分別是 2、2、4 Mbps，因為三者可獲得 1、2、2 的保障頻寬，C2 因流速小於保障頻寬故得 2Mbps，剩餘頻寬 3Mbps

以 C1、C2 的保障頻寬為比例(1:2)分配給 C1、C2。在此僅探討 CBWFQ，而忽略 CQ 與 WFQ。

Cisco 的 Packet Discarding 有 Tail Drop 與 Weighted Random Early Detection (Weighted RED, 又記為 WRED), Tail Drop 是當 queue 已滿時, 再進入的封包便自然被丟棄; WRED 是 RED 的衍申, RED 在 queue 的長度超越一個門檻值時開始對新進的封包做隨機的丟棄, 當長度超出越大時, 則封包被丟棄的機率越大。WRED 則先丟棄權重值較低的封包, 若所有封包權重值相同, 則退化為 RED。RED、WRED 適用於對壅塞(即路由器內的 queue 長度超出門檻值時)會自動降速的流量, 本次研究未包含此類流量, 故不測試 WRED。

Cisco 的 Traffic Policing [13] 功能稱為 Committed Access Rate (CAR)[14], 以 Leaky Bucket (LL) 機制限制進入或流出路由器的封包流速; Traffic Shaping 在 Cisco 稱為 General Traffic Shaping (GTS), GTS 也利用 LL 機制控制流出路由器的封包流速, 只是封包流進入 LL 前先進入一緩衝區 (buffer), 如此可將 bursty 封包流形塑為較平滑的 (smooth) 封包流。

### (3) 評量標準

本研究用以評量路由器 QoS 效能的標準主要有四, 分別是 Throughput、CPU Load、Convergence Time 與 Correctness。透過 Throughput 用以檢驗路由器的封包遞送率; CPU Load 用以檢驗 QoS 功能的執行對 CPU 耗用程度, 這是主要評量效能的方法; Convergence Time 在流量衝擊後, 頻寬變化重達穩定所需之時間, 時間越長效能越差; Correctness 探討 QoS 功能的正確性, 其中對於 CBWFQ 是以 Bandwidth Allocation Deviation *BAD* 來評量頻寬分配之正確性, *BAD* 之定義如下:

$$BAD = \frac{|R_{real} - R_{ideal}|}{R_{ideal}}; \text{ where } R_{ideal} \leq 1; \dots (1)$$

*BAD* 表示實際量測與理論計算之差異程度經正規化後所得之值, 其中  $R_{ideal}$  是兩測試流量所要求之理論頻寬比, 頻寬大者在分母, 小者在分子, 所以  $R_{ideal} \leq 1$ ,  $R_{real}$  則是相對於  $R_{ideal}$  之兩測試流量實際獲得的頻寬配額比; *BAD* 值越大表示頻寬分配之正確性越差; 反之, 則越佳。

## 3.2 QoS 測試規劃

QoS 測試規劃分為基本效能測試(Baseline Test)與 QoS 效能測試(QoS Test); 在基本效能測試中又細分為二, 以有或無設定「出入控制條件」ACL (Access Control List) 來區分, 若路由器無設定任何 ACL 對進入封包做控制, 僅單純的做封包遞送, 此種測試稱之為壓力測試(Impact Test); 反之, 稱為出入控制測試(ACL Test), 路由器對封包作 ACL 控制, 會耗用更多的 CPU 能量。而 QoS 效能測試, 則針對 Cisco 的 QoS 功能做負載量、正確性、與收

斂時間測試。

### (1) 基本效能測試

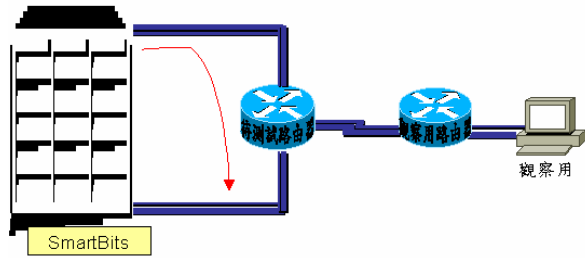


圖 4 基本效能測試之實驗環境

此測試包含 Impact Test 與 ACL Test, 測試環境建立如圖 4 所示, 流量由 SmartBits 穿過待測試路由器回到 SmartBits, 逐步提升流量以增加路由器的壓力, 當 SmartBits 從返回的流量發現有封包漏失時, 便表示待測路由器開始遺失封包, 他已無法承載更高的流量, 此時的流量速率便是該路由器最大的封包遞送率。流量在增加時也同時紀錄該路由器的 CPU Load, 以了解流量的衝擊對路由器 Load 的影響, 為觀察此值, 須從旁接一觀察用路由器, 再由此路由器的 FastEthernet 介面接上一電腦, 在此電腦利用瀏覽器連線待測試路由器以下載 QDM plug-in Java code, QDM 在瀏覽器上啟動後, 便可觀測待測試路由器的 CPU Load。此處的考量條件有二: 一是不同機型的路由器對相同的流量衝擊所產生的 CPU Load 數據; 二是相同的機型在不同的封包交換模式(PS、FS、CEF)對相同的流量衝擊所產生的 CPU Load 數據。

Impact Test 的目的在於對照 Cisco 官方公佈的效能數據, 藉此印證本研究的測試設備與環境是正確的。ACL Test 則是探討路由器對封包做 ACL 檢查所增加的 CPU Load; 由於 Cisco 的 QoS 功能多利用 ACL 將流量做分類以給予不同的服務品質, 因此當執行 QoS 功能時, 應扣除耗用在執行 ACL 檢查的 CPU Load 才是 QoS 功能真正所造成的 CPU Load。此處以 Cisco 的 Standard ACL (SACL) 為測試的條件, 改變 SACL 的數量以觀測 CPU Load 的變化, 以下是兩條 SACLs,

```
access-list 10 permit 192.168.1.0 0.0.0.255
access-list 10 permit 192.168.2.0 0.0.0.255
```

其編號是 10, 允許來自 192.168.1.0 與 192.168.2.0 兩網域的封包, 換句話說 SACL 10 把來自這兩個網域的封包列在同一類。採用 SACL 的原因是一條 SACL 只有一次條件判斷, 如上例, 第一條 SACL 只須判斷一個封包是否來自於 192.168.1.0; 若是採用 Extended ACL (EACL) 則一條 EACL 可能有多個條件判斷, 將增加分析的複雜度, 未來將在此部分著力, 因為 EACL 是最常用的 ACL。

### (2) QoS 效能測試

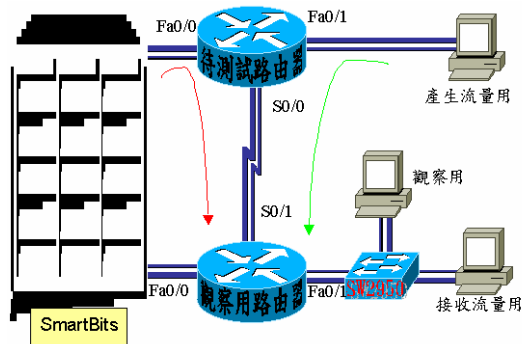


圖 5 QoS 效能測試之實驗環境

QoS 功能主要是要達到標的流量所需的服務品質，在標的流量的眼中，其他的流量便是背景流量。因此，實驗環境建置如圖 5，SmartBits 產生的標的流量穿過待測路由器與觀察用路由器回到 SmartBits，而背景流量則由一台有 SnifferPro 的電腦產生，此流量穿過待測路由器與觀察用路由器流到另一也有 SnifferPro 的電腦；兩路流量在待測路由器的 s0/0 輸出介面競爭頻寬，所有待測的 QoS 功能都設定在此介面，計有 FIFO、PQ、CBWFQ、CAR 與 GTS。將評量的效能標準計有：CPU Load、Correctness 與 Convergence Time。關於 CPU Load，特別探討 QoS 功能不利用 ACL 與利用 ACL 做流量分類所造成 CPU Load 差異；不利用 ACL 分類的方法是以前以介面(Interface 簡記為 Int)名稱來區分流量，在本環境中，標的與背景流量分別來自待測路由器 Fa0/0 與 Fa0/1。Correctness 則研究不同型態(bursty v.s. smooth)之背景流量對標的流量的服務品質的影響，以及同流速但不同封包大小的標的流量可獲得的頻寬配額是否相同。至於 Convergence Time 則是探討在輸入流速改變時頻寬分配重達穩定時所須之時間。

在此考量的條件只有不同的路由器機型，同一機型內不同的封包交換模式(PS、FS、CEF)不列入考量，因為在由基本效能的 ACL Test 已獲知 PS 不夠效能執行 QoS 功能，而 CEF 只有在路由表很大時才會優於 FS，此測試所需路由極少，是看不出 CEF 之效能，因此只考量 FS 所造成的 QoS 效能。

#### 4. 實驗結果與數據分析

實驗之結果與分析分別以基本效能測試與 QoS 效能測試來描述；由於受限於篇幅，在此無法將 Cisco 2600XM 系列的測試數據完全顯示，僅以 Cisco 2651XM 的數據為代表，並分析之。

##### (1) 基本效能測試之結果

如圖 4 的測試環境，SmartBits 產生封包流到待測試路由器，每一封包長 64 Bytes，初始的速率為 0.84Mbps (近似於 1250pps)，傳送 300 秒，每次往上遞增 0.84Mbps，直到 SmartBits 發覺通過路由器

的封包有遺失，全程 CPU Load 之變化如圖 6 所示。

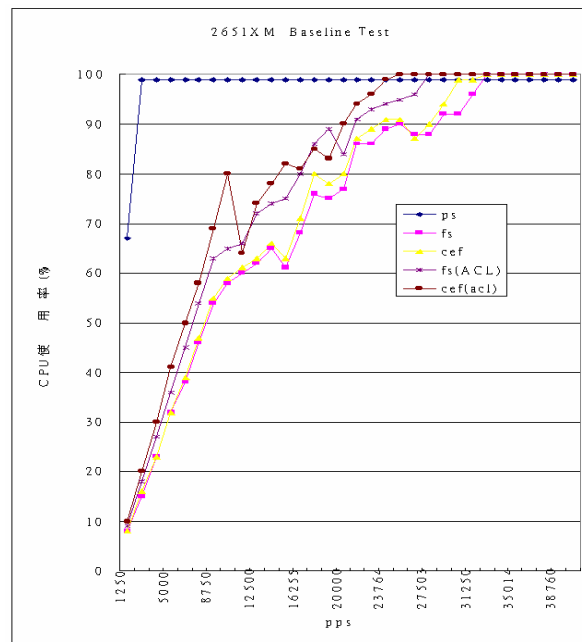


圖 6 Cisco 2651XM 基本效能測試之 CPU Load

圖 6 顯示 5 種測試，包含 3 種 Impact Tests(對應 PS、FS、CEF 等交換模式)與 2 種 ACL Tests(對應 FS 與 CEF)，由於 PS 的效能太差幾乎無法執行 ACL Test 故圖中無 ps(ACL)之測試。ACL Test 是在路由器中分別加入 1、5、10 條 ACLs 作為測試條件，由於 FS 或 CEF 都會啟動 Cisco Netflow 功能，而 Netflow 能加速 ACL 處理，它使得封包流(flow)的第一個封包才須檢查所有的 ACLs，往後的封包都直接到 Netflow 所建立的 Flow Cache 中取得出入控制的資訊，而不須去檢視所有的 ACLs，以致 ACL 的數量不對 Netflow 的效能產生影響，也因此不影響 FS 與 CEF 的效能。圖 6 之 fs(ACL)測試的 CPU Load 約比 fs 測試高了 10% 左右，這應該是 fs(ACL)需額外去搜尋 Flow Cache 的原因。

測試結果顯示 fs 的 CPU Load 與 CEF 相仿卻遠低於 ps；ACL Tests 較 Impact Tests 的 CPU Load 約增加 10%。表 1 是兩種測試的最大封包遞送率，ACL Tests 的封包遞送率在各型路由器均下降最少 5K pps。

表 1 封包遞送率比較表

	C2611XM	C2621XM	C2651XM
官方公佈數據	20 Kpps	30 Kpps	40 Kpps
Impact Test	25 Kpps	30 Kpps	37.5 Kpps
ACLs Test	20Kpps	23.7 Kpps	31.25 Kpps

##### (2) QoS 效能測試之結果

如圖 5 的測試環境，SmartBits 產生一標的流量穿過待測試路由器再回到自身，而背景流量則由一電腦的 SnifferPro 產生，穿過待測試路由器後抵達

另一電腦。圖 7 顯示所有 QoS 功能的 CPU Load 測試，所有測試的背景流量均相同，由 SnifferPro 產生，是一 bursty 流量，流速為 7142pps(約 4.8Mbps)，標的流量之初始速率 744pps(約 0.5Mbps)，傳送 300 秒，每次往上遞增 0.5Mbps，兩流量通過測試路由器的 s0/0 輸出介面，其介面頻寬為 4Mbps，QoS 功能加諸在此介面用以控制標的流量的服務品質。

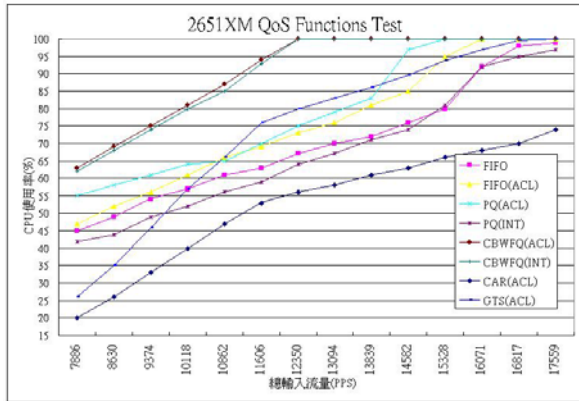


圖 7 Cisco 2651XM QoS 效能測試之 CPU Load

首先比較 FIFO、PQ 與 CBWFQ 之測試，FIFO(ACL)是以 ACL 區分標的與背景流量但 s0/0 設為 FIFO，PQ(ACL)與 CBWFQ(ACL)則是 s0/0 分別設為 PQ 與 CBWFQ；PQ(INT)與 CBWFQ(INT) 是以輸入介面(Int)來區分標的與背景流量而 s0/0 分別設為 PQ 與 CBWFQ，對於 PQ，標的與背景流量的 priority 分別設為 High 與 Low，對於 CBWFQ，標的與背景流量的保障頻寬均設為 1Mbps。FIFO(ACL)較 FIFO 的 CPU Load 高，在 Load 低時約差 5%，在 Load 高時約差 10%；PQ(ACL)較 PQ(INT)的 CPU Load 高出 10~15%，而 PQ(INT)反而低於 FIFO 的 CPU Load，這主要是因為背景流量在 PQ 的作用下頻寬都讓與標的流量，當背景流量的 burst 出現時，較多的 burst 流量被丟棄，故而留在路由器的封包量就少了，因此 CPU Load 就較輕；PQ(ACL)、PQ(INT)與 FIFO 的現象說明 PQ 本身並不會增加太多 CPU Load，增加 Load 的是 ACL。比較 CBWFQ(ACL)、CBWFQ(INT)與 FIFO 則得知 CBWFQ 較 FIFO 增加約 20%的 Load，這主要來自於 CBWFQ 的運算，而非來自於 ACL。

其次，比較 CAR、GTS 與 FIFO，對於標的與背景流量，CAR 與 GTS 的速率限制均在 2Mbps；CAR 與 GTS 無法用 Int 來區分流量，故沒有 CAR(INT)與 GTS(INT)的測試；比較 CAR(ACL)、GTS(ACL)與 FIFO 發現在 CPU Load 輕時，CAR 與 GTS 的 Load 均低於 FIFO，在 Load 高時 CAR 仍低於 FIFO，這是因為 CAR 將輸出總流量限制在 4Mbps，如此低的流量當然 Load 較低，但 GTS 由於有 buffer 可儲存 bursty 流量，存在路由器的流量便較大，此外多了 buffer 須管理，Load 就較 FIFO 為高了。這結果顯示 GTS 較 CAR 耗費 CPU Load。



圖 8 Cisco 2611XM 之 PQ 正確性測試(一)

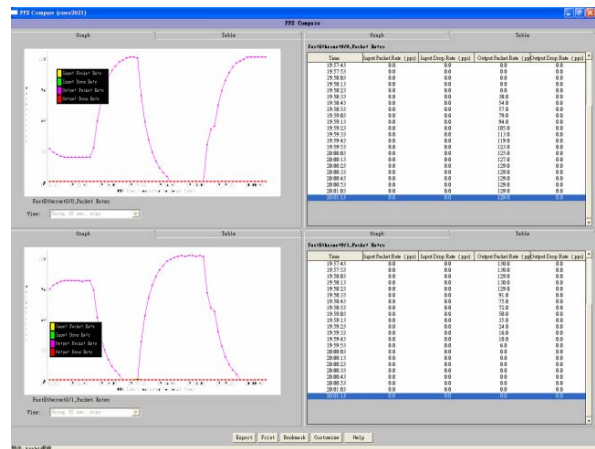


圖 9 Cisco 2611XM 之 PQ 正確性測試(二)

圖 8 與 9 顯示 PQ 的 Correctness 測試結果。圖 8 的測試條件是：高優先權的 SnifferPro 流量 100 Kbps(圖上方曲線)與低優先權的 SmartBits 流量 4 Mbps(圖下方曲線)，匯入瓶頸頻寬為 56 Kbps 的介面。SnifferPro 流量出現時，SmartBits 流量並未被壓制到 0，實乃由於 SnifferPro 產生的流量是屬於 Bursty 型態，所以抵達的封包數超出路由器在該單位時間內的能力範圍而未能處理導致遺失，也因此雖然有高優先權的保證，低優先權的流量卻仍能獲得頻寬。圖 9 的測試條件是：高優先權的 SnifferPro 流量 90 Kbps(圖上方曲線)與低優先權的 SmartBits 流量 3.6 Mbps(圖下方曲線)，我們透過調整 100M 網路卡的傳輸速率為 10M，讓 SnifferPro 產生的流量 Bursty 效應減緩再做測試，SnifferPro 流量出現時，SmartBits 流量則被壓制到 0。圖 8 與 9 顯示 PQ 是正確的，但輸入流量的型態往往導致迷思。

表二顯示封包流的封包大小對 CBWFQ 的 Correctness 測試結果。兩封包流分別由 SnifferPro 與 SmartBits 產生；SnifferPro 每隔 1ms 送出一個 64 byte 封包，顯示送出 512 pps 即 344.064 Kbps (= 512\*(64+20)\*8)；SmartBits 的 FastEthernet 介面則送出 1.72%的分量速率，等於 1720 Kbps；兩流量穿過待測路由器介面 s0/0(頻寬為 768 Kbps)，兩流量的保障頻寬分別為 96 Kbps 與 480 Kbps，其中 96 Kbps

相當於 200 pps ( $= 96K/((64 - 4)*8)$ )，Test0 與 Test1 分別設定 SmartBits 產生 64 與 1514 byte 的封包流，他們的保障頻寬相當於 1000 與 39.735 pps。在 Test0 兩流量封包大小相同，其 BAD 值趨近於零，表示 CBWFQ 運作正確，但在 Test1 **小封包流量與大封包流量無法獲得公平的頻寬分配**，在此例約有 21% 的偏差。

表 2 封包大小之於 CBWFQ 之正確性

s0/0: 768Kbps	SnifferPro	SmartBits	
流速(Kbps)	344.064	1720	
保障頻寬(Kbps)	96	480	
		Test0	Test1
封包大小(bytes)	64	64	1514
保障頻寬(pps)	200	1000	39.735
Test0 頻寬分配(pps)	297	1479	X
Test1 頻寬分配(pps)	317	X	52
<b>BAD</b>		0.0041	0.2112

最後分析 Convergence Time，我們以 SnifferPro 抓取標的流量的封包，發現當 CBWFQ 頻寬重新分配時，封包與封包間的 inter-arrival time 是立刻反應這樣的變化，所以 Convergence Time 趨近於 0。

## 5. 結論

本研究的貢獻在於我們提供給一網路路由器執行 QoS 功能時之效能數據，而網路 QoS 規劃工程師可依此數據作網路品質服務之規劃與建置。無形的貢獻在於提升了對網路設備路由器的 QoS 測試相關經驗。這些經驗分述如下：

- 1) ACL Tests 較 Impact Tests 的 CPU Load 約增加 10%。關於兩種測試的最大封包遞送率，ACL Tests 的封包遞送率在各型路由器均下降最少 5K pps。
- 2) 在 QOS 之 CPU Loading 測試方面：PQ 本身並不會增加太多 CPU Load，增加 Load 的是 ACL。CBWFQ 較 FIFO 增加約 20% 的 Load，這主要來自與 CBWFQ 的運算，而非來自於 ACL。在相同的測試條件下，GTS 比 CAR 耗費較多的 CPU Load。
- 3) 至於 QOS 之 Correctness 測試：Bursty 封包流會使 PQ 的測量者誤以為 PQ 運作不正確，實則是測量工具的不精確。然而，封包大小差距大的兩封包流在 CBWFQ 的頻寬分配下是無法得到公平的分配，這使得 CBWFQ 的正確性在某些情況下是有待商榷的。
- 4) 最後，CBWFQ 之 Convergence Time 趨近於 0

### 參考文獻

- [1] ANDREW S. TANENBAUM, 蔡明志譯, 第三版, “電腦網路”
- [2] G. Hasegawa, M. Murata, “Survey on Fairness

Issues in TCP Congestion Control Mechanisms,” IEICE TRANS. COMMUN. vol. E84-B, NO.6, Page(s):1461-1471, June 2001.

- [3] Joerg WidMer, Robert Denda, and Martin Mauve, "A Survey on TCP-Friendly Congestion Control", IEEE Network, May/June 2001.
- [4] W. Richard Stevens, “TCP/IP Illustrated, Volume 1”
- [5] 林盈達, “寬頻網際網路品質的解決方案”, 網路通訊, 101 期, 1999 年 9 月, 頁 122-128
- [6] 蔡建新, 第一版, “網路工程概論”
- [7] Cisco Systems, “Cisco - How to Choose the Best Router Switching Path for Your Network,” <http://www.cisco.com>.
- [8] Cisco Systems, “Cisco - Cisco 2600 Series Router Architecture,” <http://www.cisco.com>.
- [9] Cisco Systems, “Cisco Networking Academy Program CCNA 1 and 2 Companion Guide,” 3<sup>rd</sup> Ed., Cisco Press, 2005.
- [10] [http://www.cisco.com/warp/public/63/2600\\_architecture\\_23852.pdf](http://www.cisco.com/warp/public/63/2600_architecture_23852.pdf)
- [11] [http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/QoS\\_c/qcdintro.pdf](http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/QoS_c/qcdintro.pdf)
- [12] <http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120t/120t5/cbwfq.pdf>
- [13] [http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/QoS\\_c/qcprt4/qcdpolsh.pdf](http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/QoS_c/qcprt4/qcdpolsh.pdf)
- [14] [http://www.cisco.com/warp/public/cc/pd/iosw/tech/carat\\_wp.pdf](http://www.cisco.com/warp/public/cc/pd/iosw/tech/carat_wp.pdf)
- [15] [http://www.cisco.com/univercd/cc/td/doc/cisint/wk/ito\\_doc/QoS.pdf](http://www.cisco.com/univercd/cc/td/doc/cisint/wk/ito_doc/QoS.pdf)