行政院國家科學委員會補助專題研究計畫成果報告

※※※※※※※※※※※※※※※※※※※※※※※※※※
※ ※
※ **寬頻網際網路服務品質保證(III) 總 計劃** ※
※ ※
※※※※※※※※※※※※※※※※※※※※※※※※※※

計畫類別：□個別型計畫 ¦整合型計畫
計畫編號：NSC－89-2219-E-009-027-
執行期間： 89 年 08 月 01 日至 90 年 07 月 31 日

計畫主持人：李 程 輝 國立交通大學電信工程學系 教授

本成果報告包括以下應繳交之附件：
□赴國外出差或研習心得報告一份
□赴大陸地區出差或研習心得報告一份
□出席國際學術會議心得報告及發表之論文各一份
□國際合作研究計畫國外研究報告書一份

執行單位：國立交通大學電信工程研究所

中 華 民 國 90 年 10 月 30 日

Abstract

Continuous rapid growth of the Internet in recent years makes it the most probable future integrated services network. However, current Internet architecture is inadequate in providing real-time applications. It cannot guarantee delay bound requirements of real-time applications. Moreover, non-real-time applications may be terminated if real-time traffic causes congestion. Internet 2 is thus proposed to meet future needs. In order to support the realization of future broadband Internet with guarantee of wide quality of service (QoS) requirements, this integrated project constantly improves the results in the previous year and further investigates the following key technologies:

A. High-capacity (Gigabit) routers: In this project, we design and develop high-capacity routers. We have developed a prototype router, which consists of a data path module, a queueing module, a classifier, and a scheduler in last two years. In this sub-project, we continue developing two key technologies including scheduling and switching. In the prototype, we implemented SCFQ mechanism due to the consideration of implementation complexity. However, the delay bound and fairness of SCFQ is not as good as WFQ. In new design, we implement the FFQ to improve the performance of the scheduler. In the queueing management of the prototype, we adopted the shared memory architecture. It has a disadvantage that its capacity is limited for switch with many ports because of heavy memory access. It is possible to develop a large-scale input queueing system, but there exists a head-of-line blocking problem. In this sub-project, we investigate the CIOQ architecture. We have developed a algorithm to emulate an output-queueing switch and evaluate the performance of CIOQ with finite buffers

B. Measurement-based Admission Control and Congestion Avoidance Schemes for Controlled-Load Service in Broadband Internet: Many real-time applications are capable of adapting their transmission to the network state and can as well tolerate occasional delay

bound violations in the presence of transient network congestion. For this type of applications, an absolutely reliable bound on packet delivery times is not required. Moreover, many non-real-time applications such as on-line transaction processing and distributed simulation would desire a congestion-free packet delivery service from the network. This new type of service is called "Congestion-Free Service" with the guarantee of a maximum packet loss rate. This work considers resource allocation in the support of "Congestion- Free Service" in Broadband Internet. First, we studied traffic characterization under different measurement models via analyzing traffic traces collected from National Taiwan University campus network. The preliminary results show that traffic load can be approximated by the Normal distribution. In the second part of the work, we proposed a dynamic bandwidth and queue management scheme to support "Congestion-Free Service." To better understand the characteristics of the broadband Internet, we have developed a tool called *AppMeasure* which provides three sets of service: *IPFlow*, *AnyTrace*, and *AppFlow*. IPFlow provides on-line tracking and reporting of individual IP flows. In AnyTrace, user can specify the *number of bytes* of a packet to trace. It serves as a measurement base to enable *all-layer* traffic accounting and analysis, and differentiated charging based on Quality of Service. Appflow tracks and reports any *type of traffic* specified by the user. It aims to integrate with policy-based QoS service

C. QoS routing: This sub-project describes the implementation of a class-based QoS routing algorithm, which is designed for low blocking probability and low overload. The class-based QoS routing algorithm is designed in the per-pair granularity. We design a software architecture and divide it into several modules. Then we describe what the modules do and how they work. At least the performance evaluation of the implementation is discussed. The results show that the costs, such as processing time of path computation and memory requirement of routing table, are expensive. The costs are what we have to pay for QoS routing supporting.

D. Integrated Service and DiffServ Technology: Network processors are emerging as a programmable alternative to the traditional ASIC-based solutions in scaling up the data-plane processing of network services. They serve as co-processors to offload data-plane traffic from the original general-purpose microprocessor. In this work, we illustrate the process and investigate performance issues in prototyping a DiffServ edge router with IXP1200, which consissts of one control-plane StrongARM core processor and six data-plane microengines, and stores classification and scheduling per-flow policy rules at SRAM and packets at SDRAM. The external benchmark shows that though the system can achieve aggregated wire-speed of 1.8Gbps in simple IP forwarding, the throughput drops to 200~300Mbps when performing DiffServ due to the double bottlenecks of SRAM and microengines. Through internal benchmarks, we found that performance bottlenecks may shift from one place to another given different network services and algorithms. For simple IP forwarding services, SDRAM is a nature bottleneck. However, it could shift to SRAM or microengines if heavy table access or computation is involved, respectively. We also identify the design pitfall of the hardware called the “MAC buffer overflow”.

Keywords: Broadband Internet, Quality of service, QoS routing, Gigabit router, Signaling, Traffic measurement, Network Processor, DiffServ

中文摘要

隨著網際網路的盛行，許多網路應用程式相繼出現，這些應用需要得到一定程度的傳輸頻寬與服務品質保證，才能讓使用者感受到其實用性，但目前的網際網路只提供盡力服務，並不能給予這些應用程式有效的支援與滿足使用者的需求。為了支援未來寬頻網際網路的實現及有效保證服務品質，本整合計畫除了延續前兩年的研究成果，並深入發展下列關鍵技術：

A. 使用於寬頻網際網路之 Gigabit 路由器與訊務管制：在前兩年的計劃中，我們已經完成了第三層路由器的雛形設計，其中包括了資料路徑模組、佇列管理模組、訊務區分器模組等等。本年度計劃將持續研究高速路由器的關鍵技術，包含了排程技術的開發與交換技術的研究。在排程技術方面，在實作上複雜度的考量之下，第一版的排程器採用了 SCFQ (Self-Clocked Fair Queueing)機制，但其延遲的保證上並不如 WFQ (Weighted fair queueing)來的理想，因此我們研究了 FFQ (frame-based fair queueing)的設計考量與實作，以其改善訊務的延遲與公平性。在交換技術的研究方面，在雛型機上使用的佇列管理模組是採用共享式記憶體架構，其交換容量會受限於記憶體的速度，而輸入佇列(input queueing)架構對記憶體的需求較低，但會有佇列前端擁塞(head-of-line blocking)的問題，因此我們研究混合輸入輸出佇列的架構，證明了它可以完全仿效一個輸出佇列的交換機，並探討 CIOQ (combined input and output-queued)交換機在輸出埠 buffer 有限的情況下的行為。

B. 支援寬頻網際網路「負載控制服務」以量測為基礎的允諾控制與擁塞避免機制：目前網際網路所提供的「盡力服務」，由於無法保證服務品質，所以並不能給予即時應用程式有效的支援。為了解決這問題，目前已經發展出一些即時媒體應用能夠視封包遭遇到的延遲而動態地調整資料的傳送。這些應用並不需要絕對可靠的封包延遲上限，但必須盡可能避免封包在傳輸過程中被丟棄。本研究即針對這類應用程式提供『無壅塞服務』，我們將其定義為『保證聚合資料流的最大封包遺失率為 ε』，希望將來可以應用於 IETF 整合服務架構下的負載控制服務。本研究首先對台大校園骨幹網路做實際流量的量測及趨勢分析，以瞭解網路資料流的特性。

初步結果顯示網路負載具有近似常態分配的特性，可進一步用於網路流量的預測以及網路資源的規劃與控制。此外，我們針對『無壅塞服務』的提供，提出一個動態頻寬與緩衝區分配的方法。實驗結果也顯示避免頻寬使用率太高及動態分配頻寬可以有效抑制壅塞的發生。為了進行更精確的量測以及效能評估，我們發展了一套量測工具。藉著這套工具的使用，我們可以獲得網路中流動封包對應到每一層通訊協定的相關資料；量測的基本單位可以細分至每個資料流。不但讓預測更精準，也便於 off-line 效能分析。

C. 寬頻網際網路中服務品質路由之研究：由於在現今的服務品質保證路由的演算法中，低阻絕率和低計算負擔是難以兩全的。有一個新的演算法以等級作為服務品質保證的依據，可在追求低阻絕率的同時，維持一定程度的計算負擔。這個演算法稱為分類服務品質保證的路由協定，本文主要在說明如何實作具有此種演算法的路由器，並解釋它的軟體架構及各模組的功能。之後的效能測量顯示要支持服務品質保證，則路由器必須承受較高的額外計算負擔，如路徑計算所需的計算時間、路由表的大小、路由器對網路狀態的更新和路由資訊的交換等等。

D. 整合服務與差別服務網路之相容運作技術：網路處理器已漸漸成為傳統以 ASIC 為主用來處理使用者平面封包的另一可程式化的選擇。它利用其共同處理器(co-processors)協助處理原本一般用途處理器(general-purpose processor)所負責的使用者平面的封包。在本論文中，我們將描述將差別式服務邊緣路由器(DiffServ edge router)實作於 IXP1200 網路處理器的流程，並探討其效能。IXP1200 網路處理器具有一個處理控制平面的 StrongARM 核心處理器(core processor)和六個共同處理器，並將分類(classification)和排程(scheduling)的規則寫在 SRAM，封包則儲存於 SDRAM。根據外部測試顯示，就一條輸入埠(input port)而處理能力(throughput)為 50Mbps 時，本系統可以支援符合個別行為(Per-Hop Behavior)的 500 個資料流(flow)，且可隨著 SRAM 的增加而繼續擴充。經由內部測試我們發現效能瓶頸(bottleneck)會隨著不同的服務和實作而轉移到不同的地方。就簡單的遞送服務(forwarding service)而言，SDRAM 為一當然瓶頸。然而當涉及眾多的規則表查詢和計算時，SRAM 和 microengine 則分別成為其效能瓶頸。另外，我們也指出了 IXP1200 硬體設計的可能缺失，稱之為’媒體存取控制緩衝儲存器

的溢流問題’ (MAC buffer overflow)**。**

關鍵詞：寬頻網際網路、服務品質、服務品質路由、Gigabit **路由器、信號、排程、訊務量測、網路處理器，差別式服務**

目錄

第一章　前言

由於傳統電信網路與數據的網路服務對象不同，而分別採用了電路交換（Circuit switching）與分封交換（Packet switching）為多工與交換的技術。雖然是針對服務對象設計，使其效率很高。但是卻有重複投資線路與設備、以及不同網管的缺點，增加基礎建設的成本。而目前多媒體應用的普及化，使用者需求的頻寬急速增加，且要求的服務品質（Quality of Service）亦不同，促使網路朝寬頻整合服務的方向演進。此外，隨著網際網路的使用人口與訊務呈指數倍數成長，加上高速乙太網路（Fast Ethernet，100Mbps）與乙太交換器（Ethernet Switch）的問世，網際網路極可能成為未來整合服務網路。 然而現有網際網路並無法提供即時應用。它不能保證即時應用的延遲與延遲變異等服務品質。同時，若即時應用的訊務造成擁塞，非即時應用的運作可能會被中止。換言之，目前的網際網路也無法有效保護非即時性應用的連線。為滿足未來整合服務的需求，於是最近產生了寬頻網際網路。

寬頻網際網路無疑的將會面臨各類不同服務品質需求的應用，在面對訊務急速增加的情況下，為了有效保證服務品質，它需要:

(1) 大容量路由器：提升網路容量。
(2) QoS 路由：改善網路使用效率。
(3) 信號：提供訊務源特性與資源保留。
(4) 排程（Scheduling）：提供各種不同的服務品質。
(5) 允入控制（Admission Control）：限制網路資源之使用。
(6) 訊務管制（Traffic Policing）：確保連線之服務品質。
(7) 網路規劃與管理（Network Planning and Management）：監控網路效能與規劃。

本年度計劃延續前兩年的研究成果並深入研究與發展相關的關鍵技術。

第 二 章　計劃緣起與目的

寬頻網際網路的產生是為了解決目前 Internet 頻寬不足且無法保證服務品質（Quality of Service）的缺點；但由於 QoS control 牽涉的技術很廣，並非一人得以獨力完成，透過群體計畫可收集思廣益之效，提出整體性的解決方案。相關子計劃的研究項目著重在提昇網路頻寬與保證服務品質的關鍵技術，敘述如下。

一、大容量路由器與訊務管制技術：

未來的網路服務除了滿足頻寬需求外，網路更必須保證服務品質(Quality of Service, QoS)，因此在路由器中除了要能把封包加以分類之外，還要對於不同的封包由排程器來處理，以符合不同使用者各自的要求。在排程的技術上，複雜度與效能一直是很難兼顧的，如 GPS、WFQ 雖然擁有好的公平性與 delay bound，但是，要維持準確的系統時間的需要複雜度高的硬體才能達成；SCFQ 降低了系統時間的複雜度，卻付出了 delay bound 作為代價，其系統表現與系統複雜度成正比。因此我們選擇了 FFQ 作為硬體實踐的演算法，因為它保證的效能與 WFQ 相同，但是大大的減低了實踐的複雜度。

長久以來，因為前端壅塞限制傳輸率的關係，交換系統的設計大都避免採用輸入佇列(input-queued)架構。但是輸出佇列架構或共享記憶體架構的交換系統容量受限於記憶體頻寬，而目前記憶體頻寬尚無法與傳輸速率匹配，因此，隨訊務量急速增加的結果，使得輸入佇列架構受到重視。降低前端壅塞的方法主要可從兩方面下手：(1)在輸入埠中，不再採用先進先出(FIFO)的架構，而是到相同輸出埠的封包形成一個佇列，然後搭配 maximal 配對方式，可以達到接近 100%傳輸率的效果；(2)將縱橫式架構加速(speedup)，使其速度超過輸入/輸出線(input/output link)，因為加速的關係，所以輸出埠也需要緩衝器(buffer)，因此變成一個 combined input and output-queued (CIOQ)的架構。在本計劃中，我們證明了 CIOQ 能夠完全仿效一個輸出佇列的架構，並探討了它在輸出緩衝器有限的情況下的考量。

二、支援寬頻網際網路「負載控制服務」以量測為基礎的允諾控制與擁塞避免機制：

為了因應即時媒體應用對傳輸服務品質(QoS, quality of service)的需求，IETF 制訂了兩個可提供傳輸服務品質的架構，分別是整合服務(Integrated Service)[19]以及差異化服務(Differentiated Service) [12]。傳統的即時媒體服務研究主張對每個封包提供絕對(absolute)的延遲上限(delay bound)，而且不允許有任何封包在排隊(queueing)時被丟棄，這樣的服務被稱為保證服務(Guaranteed Service)[14] [18]，但並不是所有即時媒體應用都需要這麼高水準的服務品質，目前已經發展出一些應用程式如 *vat*、*nv*、*vic* 等，能夠視封包遭遇到的延遲動態調整資料的傳送，對這些應用程式而言，並不需要絕對可靠的封包延遲上限，盡可能避免封包在傳輸過程中被丟棄而順利抵達目的地，反而是較為必要的傳輸服務品質。

寬鬆的傳輸服務品質保證對於網路資源的管理提供了更大的彈性，也為路由器在允諾控制(admission control)、封包排程(packet scheduling)及緩衝區管理(buffer management)等各項功能設計上提供了一個新的思考面向。我們認為提供低封包遺失率且有最小傳輸率保證的無壅塞服務，對現存以及將來可能出現的應用，應會具有相當的實用性。而透過以量測為基礎的方法即時了網路流量特性，以從事允諾控制及動態配置資源，相信將非常有助於促成無壅塞服務的實現。

為了讓預測更精確，我們發展了一套稱為 AppMeasure 的量測工具。這套工具可以被動地接收所有網路上流動的封包，辨識出此封包所屬的資料流，並更新此資料流的的相關資料。因此，我們可以獲得網路上目前所有的資料流的相關資訊，不但可以提升預測的準確度，也方便效能評估。

三、寬頻網際網路中服務品質路由之研究：

Blocking probability and route granularity are important issues for QoS routing protocols. In general, finer granularity yields higher blocking probability but also higher computational complexity. A new algorithm proposed in [24] to show us a new thought of QoS routing after integrated the researches mentioned above. This QoS routing protocol, with a novel idea of per-class forwarding with routing marks, is designed in order to achieve the high scalability and low blocking probability. The mechanism of the routing is based on per-pair granularity. It is shown by simulation that, with a small number of routing marks, the routing algorithm yields competitive blocking probability as compared to the routing algorithm that routes flows independently.

In this sub-project, we try to implement the class-based routing algorithm as an extension to a routing daemon, named Zebra. When a router receives a QoS request, the router may trigger following actions: path calculation, marking, and forwarding. In this project, we will only focus on the issue of routing, i.e., the path calculation task. The path calculation will find a new route with the least cost and abundant residual bandwidth based on Dijkstra's algorithm, which is also adopted by OSPF

五、整合服務與差別服務網路之相容運作技術：

The increasing link bandwidth demands even faster nodal processing especially for the data-plane traffic. The nodal data-plane processing may range from routing table lookup to various classifications for firewall, DiffServ and Web switching. The traditional general-purpose processor architecture is no longer scalable enough for wire-speed processing so that some ASIC components or co-processors are commonly used to *offload* the data-plane processing, while leaving only control-plane processing to the original processor.

Many ASIC-driven products have been announced in the market, such as the acceleration cards for encryption/decryption [29], VPN gateways [30], Layer 3 switches [31], DiffServ routers [32] and Web switches [33]. While these ASICs indeed speedup the data-plane packet processing with special hardware blocks, much wider memory buses, and faster execution process, they lack flexibility in *reprogrammability* and have a long development cycle which is usually months or even years.

Network processors are emerging as an alternative solution to ASICs for providing scalability for data-plane packet processing while retaining reprogrammability. In this study, we adopt Intel IXP1200 [34] network processor shown in Fig.1 which consists of one StrongARM core and six co-processors referred as microengines, so that developers can embed the control-plane and data-plane traffic management modules into the StrongARM core and microengines, respectively. Scalability concern in data-plane packet processing could be satisfied with the four zero context switching overhead hardware contexts in each of the six microengines and the instructions specialized for networking.

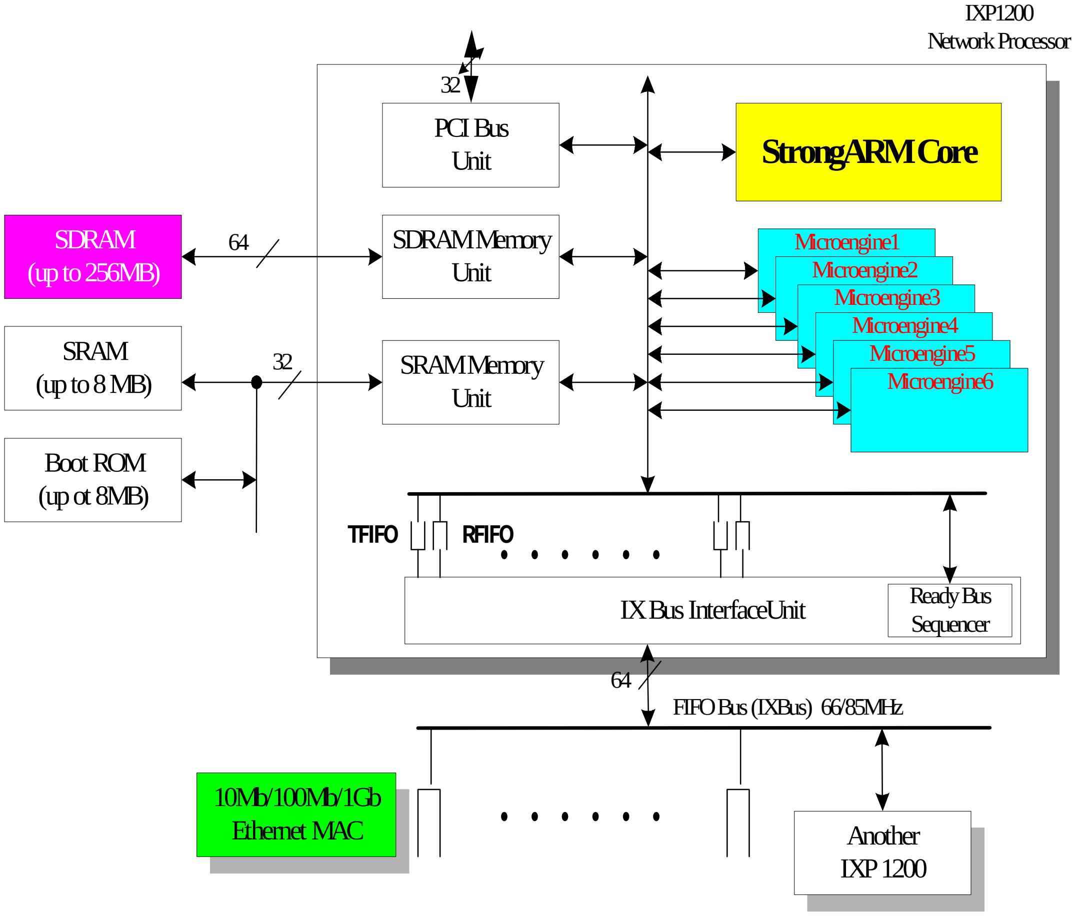


Fig. 1 Hardware architecture of IXP1200

第 三 章　研究方法及結果

(一)、使用於寬頻網際網路之 Gigabit 路由器與訊務管制技術

A. FQ 排程技術的實現

首先我們分析固定長度的 FFQ 演算法加以分析，其方塊圖如圖二所示有四個主要的工作區塊：主要控制單元、時間戳記的計算單元、frame 的控制單元、以及時間戳記的排序單元。當封包進出公平排程器是由主要控制單元來控制，frame 的控制單元則是依照封包進出的狀況以及 frame counter 的變化來決定是否更新系統時間或者是作 frame 的更新。時間戳記的計算單元則是依據現在的系統時間、類別的結束時間以及依據進來的封包的比重計算出它的結束伺服時間，並且判別此封包的結束伺服時間是否超越了現在的 frame，並加以標記。算完了結束伺服時間的封包便被送到時間戳記的排序單元，依據時間戳記的排序單元加以排序並等待被伺服。

圖二、功能方塊圖

對於一些參數在硬體的實現上我們必須加以設計，如時間標籤格式、frame 大小之決定、比重之決定、與矩陣計數的簡化，之後，將系統的 state machine 與 I/Q 介面定義出來。

在可變封包長度的環境下，結束伺服時間(finish time)時間可能不是整數，因此需要應用到浮點運算除法，但浮點運算是非常複雜的動作，所以比較好的解決方式是對算法加以簡化，以降低實踐上的複雜度。簡化型可變長度 FFQ 算法概念在於利用單位量化(quantization)的方式來表示浮點數字，也就是說以下的所有時間戳記(time stamp)的表示法再也不是絕對時間表示法，而是以某個量來作基本單位的相對表示法。我們訂定 64 個位元組的傳送時間為其時間基本單位。因此在封包長度也需要以 64 位元組為單位來表示，因此我們可以利用固定長度 FFQ 的硬體來實現可變長度的 FFQ 如圖三所示。

圖三、可變長度 FFQ 演算法的架構

B. CIOQ 交換架構

利用 CIOQ 的架構，在 corssbar 加速兩倍的情況下，我們不僅可以消除 HOL 的困擾，並發展出 LCF/MUF 演算法來仿效一個 output queueing 交換機。這裡仿效的意思是對相同的輸入訊務，每個封包離開交換機的時間及順序與輸出佇列架構相同，如此就可達到與輸出佇列交換系統相同的效能，也就是說只要這個被仿效的輸出佇列交換系統能達到的服務品質，這個 CIOQ 交換系統也能達到。此外，這個被仿效的輸出佇列交換系統並不限定使用何種服務排程(service scheduling)方法。

圖四所示是 CIOQ 的架構圖，中間是一個 crossbar switch，輸入與輸出均有 buffer，而每個 phase，corssbar 可以將輸入埠的資料送到輸出埠，而究竟是哪個輸入部會送到哪個輸出埠，則由配對器來決定，因此，為了增加 CIOQ 交換系統的效能，如何有效的配對是一個關鍵。本計劃提出新的配對法與架構，其中使用的配對方式稱為 Least Cushion First / Most Urgent First (LCF/MUF)，定義出緊急的程度，然後將最緊急的資料優先配對。經過證明，在 crossbar 加速兩倍的情況下，CIOQ 可以仿效任意的輸出佇列交換系

圖四、CIOQ 架構圖

統。在 CIOQ 架構下，每個輸出埠都有自己的 buffer，因此我們需要了解在有限 buffer 的情況下，效能降低的情形。經過模擬實驗，對於 uniform 的訊務，就算是 16X16 的交換機，輸出的 buffer 只要有八個 cell 就可以達到很好的效果。至於 correlated 的訊務，假設其 burst length 為 L，在負載為 0.8 的情況下，則需要 3L 至 4L 的 buffer 才可以達到不錯的效果。

（二）、支援寬頻網際網路「負載控制服務」以量測為基礎的允諾控制與擁塞避免機制

A. 相關研究

NetFlow 是目前 Cisco 公司所研發的網路流量分析系統，在 Cisco 公司所出產的路由器和交換器中都可以見到其蹤影。而我們自行開發的工具也是一套類似的網路流量擷取分析的系統. 由於目前已經許多應用可以用來分析 NetFlow 的輸出資料, 為了相容性, 我們特地將 AppMeasure 的輸出報告格式設計成和 NetFlow 相容。

NetFlow 中是以 IP 來源位址、IP 目的位址、IP 來源埠號、IP 目的埠號、IP 協定類別以及 TOS 欄位來定義出一個資料流. 而 AppMeasure 不僅可以檢查上述封包傳輸層的標頭（header）資訊，藉此取得 IP 資料流；還可以由使用者設定封包內容擷取長度，檢

查封包所攜帶的資料內容（content），因而提供了更精確的封包分類，取得應用流（Application flow）的資訊。

B. AppMeasure 的功能及架構

AppMeasure 主要提供了三項功能：(1) IPFlow：類似 NetFlow，針對每一個網路 IP 資料流紀錄成檔，且檔案格式和 NetFlow 相容，故可使用現成分析 Cisco NetFlow 的工具程式來分析所得資料。(2) AnyTrace：使用者可以自行設定封包擷取的資料長度。因此，我們可以獲得封包所攜帶的內容，藉此將封包做更精細的分類。(3) AppFlow：使用者可以任意指定想要監測的網路應用類別，如：FTP，TELNET，WWW 等等。目前是使用埠號來代表想要監測的 application。AppMeasure 系統架構圖如圖四所示，分成 AppMeasure 和 AppCollect 兩部分；如果機器的運算能力足夠，這兩部分也可以在同一部機器上。AppCollect 位於某個量測路徑上，所有流經此路徑的封包都被它攔截並送往 AppMeasure 記錄。藉由這些網路資料流相關資訊，我們可以對網路特性作更精確的預測，也可以清楚瞭解某特定資料流所受的待遇，進而校調系統相關參數，讓整個網路的效能更加提升。此外，藉由 AppMeasure 所提供的資訊，我們可以運用深度網路分析技巧和資料探勘（Data Mining）技術來瞭解網路的一些特性，讓網路規劃和服務品質管理做得更好。而 AppMeasure 也可以擔任網路監視器（Monitor）的角色，使得網路也可以提供使用者計費的功能，讓網路管理更為方便。

圖五. AppMeasure 的架構圖。

圖六. AppMeasure 程式流程圖。

圖七.測試環境

圖八.測試結果

C. 計畫進行進度及狀態

為了增進 AppMeasure 的整體系統效率，目前的測試環境是將 AppMeasure 和 AppCollect 這兩支程式放在不同的兩部機器上。AppMeasure 和 AppCollect 目前都是實作在使用者層（user level）的應用。如此的實作的方式是為了要能夠跨平台使用。為了更加提升效率，將來我們會將它們實作在作業系統核心層（kernel level）。AppCollect 目前的實作方式是使用 Pcap 這個函式庫。整個系統實作架構圖請參考圖六。Pcap 會將抓到的封包傳送至位於使用者層的 AppCollect，而 AppCollect 會定期（一秒鐘）或是緩衝區滿載時，將這些擷取的封包以資料報(datagram) 透過網路傳給 AppMeasure。我們對 AppMeasure 進行了一些初步測試, 以觀察在不同封包大小的情況之下, 啟動 AppMeasure 對原本作業系統的額外負擔（overhead）有多大。測試環境請參閱圖七。測試結果如圖八。由實驗結果可知，在小封包為主的網路中，啟動 AppMeasure 還是會對系統效能造成一定程度的影響。所以在下個版本中，AppMeasure 將實作在核心層，以提昇效率。此外，我們也將會增加 AppMeasure 所能夠辨識的應用程式數目。

（三)、寬頻網際網路中服務品質路由之研究

A. Implementation Issue

A.1 . Design Objectives and Scope

Our objective is to put the class-based QoS routing protocol into practice, and we try to limit the implementation complexity. There are some important assumptions which affect the design choices. These assumptions include:

- Support for on-demand path recalculation.
- Exchange QoS parameter using the metric field of LSAs.
- Support per-class routing.
- Interface to RSVP.

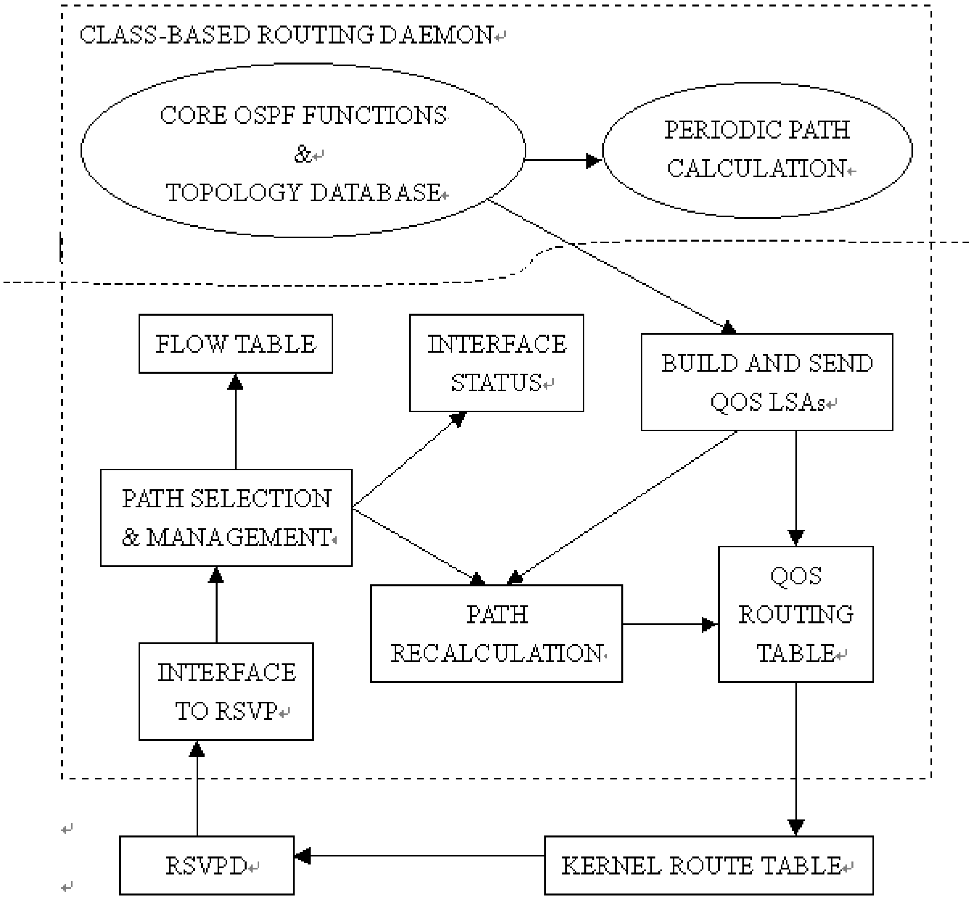
In addition, our implementation also relies on the following simplifying assumptions made in [23].

- The scope of QoS route computation is limited in a single area.
- All routers in the area run the class-based QoS route daemon.
- All interfaces on a router are QoS capable.
- Support hop-by-hop routing only.

3.2. Software Architecture

Figure 9 shows the architecture of the class-based QoS routing daemon. The modules of QoS routing consists of following modules:

- **Interface status module** maintain parameters of network status from the original OSPF. The interface status records the residual bandwidth on the interface.
- **QoS routing table module** records route cost and residual bandwidth along the path as QoS parameter in its route entry. the module stores the class-based routing table into kernel routing table using the data structure made for ECMP.
- **QoS LSAs module** creates QoS LSAs by collecting status of all interfaces on the router and updates the QoS parameters, route cost and residual bandwidth, recorded in the routing table when a new QoS LSA arrives.
- **Flow table (with traffic characteristics) module** records the QoS request of flows. The entry of the flow table contains flow informantion and traffic characteristics. The QoS requirements are part of RSVP messages.
- **Path selection and management module** records the request of flows into the flow table and trigger path recalculation if necessary.
- **Path Recalculation module** finds the least cost route according to Dijkstra algorithm.

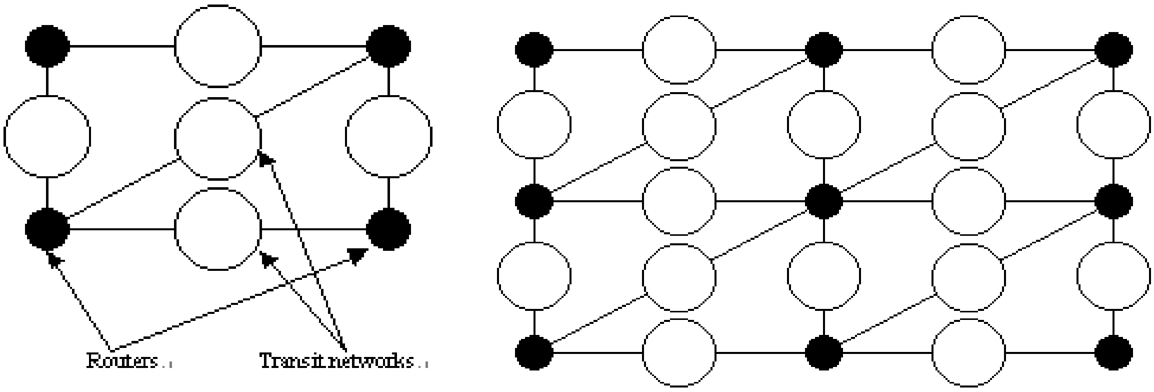


B. Performance Evaluation

B.1. Methodology

In this section, we will evaluate the performance of our implemetation. We explore three different dimensions in our comparisons: a) processing cost, b) memory requirement, and c) message generation and reception cost.

We construct a network topology to accomplish the experiment. The topology can be expanded by repeating a basic building block. The basic building block consists of 4 routers and 5 transit networks and is shown in Figure 10. The N x N mesh topology is constructed by repeating the basic block along two dimensions. We can control the network size by simply changing N. We set up the mesh topology by creating pseudo router LSAs in our implementation.



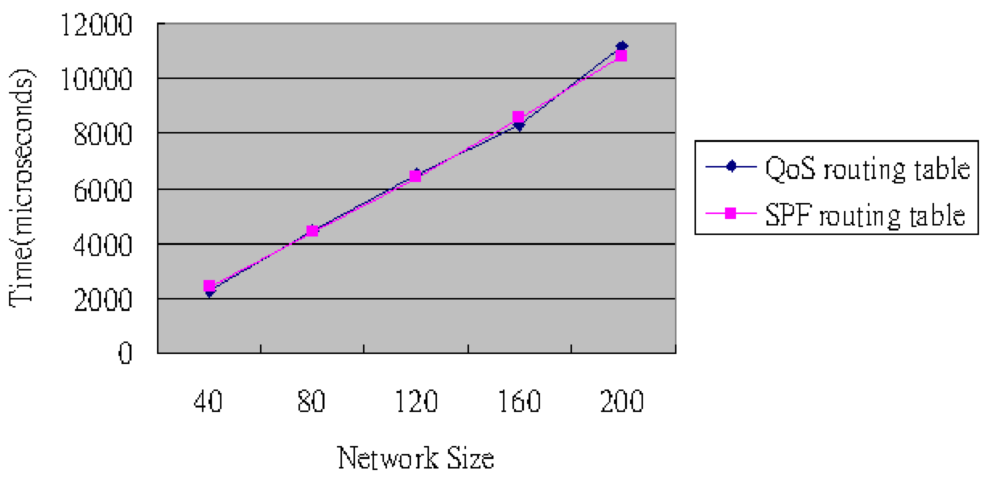


Fig. 3. Processing time for path computation

B.2. Stand-Alone Cost

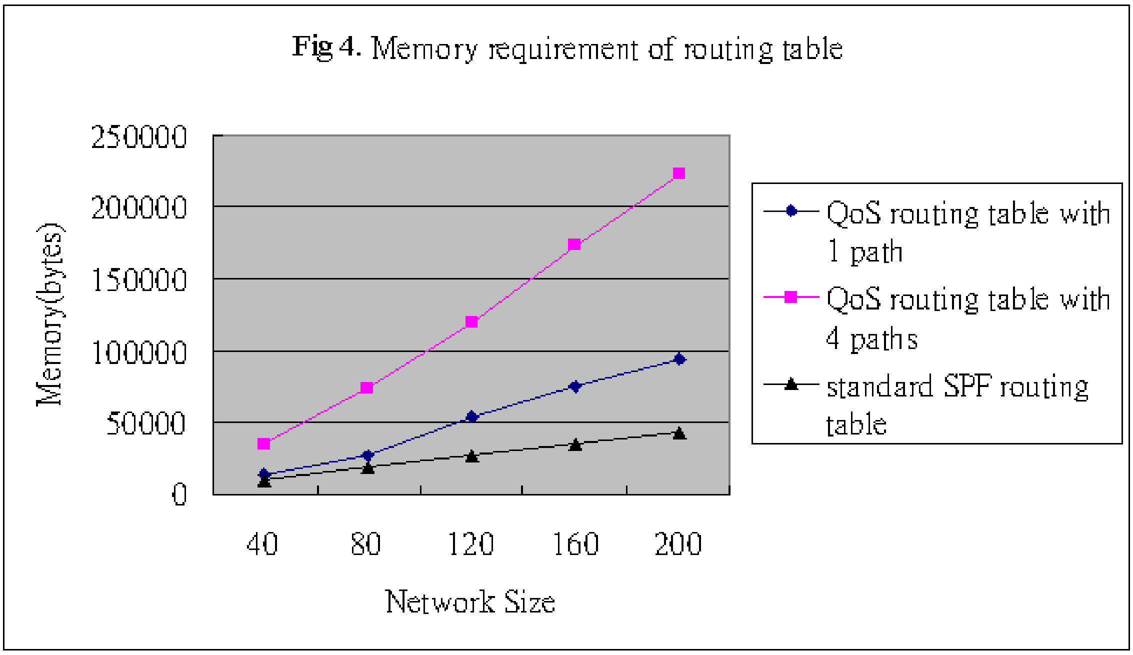
It is important to measure the cost of our implementation. Among the router cost, processing time and routing table size can be measured on a single router. The cost measured on a single router is called stand-alone cost. To measure the traffic amount of LSAs exchange, we need at least two routers and monitor the packets of LSAs on the link connected to the two routers.

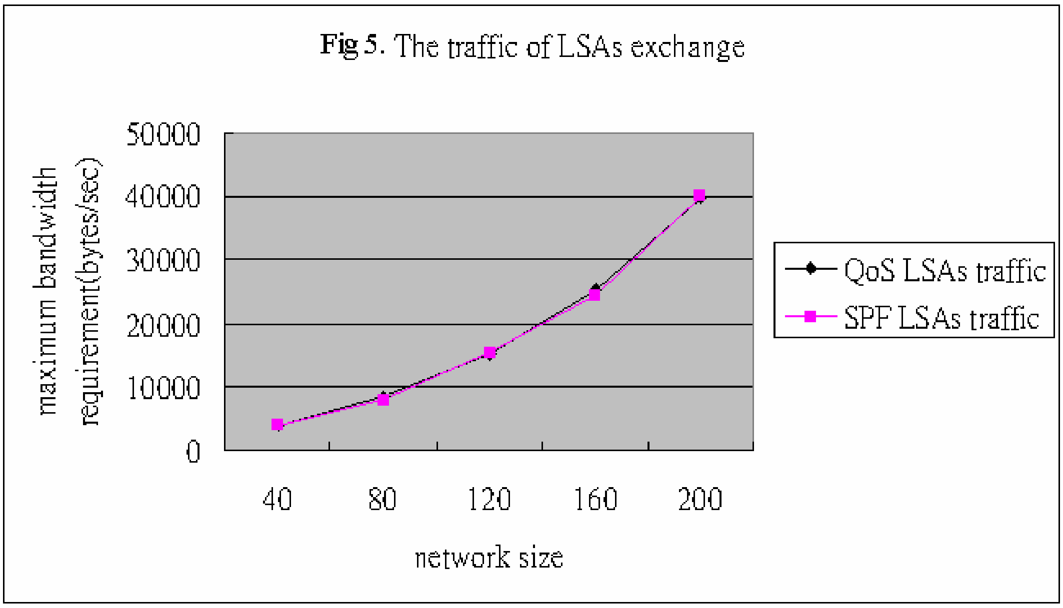
B.2.1 Processing time of Routing Table Computation

Figure 11 shows the comparison of processing time used by QoS routing and standard SPF tree calculation. The standard SPF routing algorithm runs Dijkstra to find the SPF tree, and so does the class-based QoS routing algorithm. The result provides that the sum of reciprocal bandwidth is good to be the cost function of the class-based QoS routing algorithm.

B.2.2. Memory Requirements of the QoS Routing Table

Figure 12 shows the comparison of memory requirements of QoS routing table with full paths, QoS routing table within one path and standard SPF routing table. To provide accuracy of QoS routing, we maintain the lists of nexthops in each route entry. It results that the memory used to store the QoS routing table is much greater than the standard SPF routing table. It's necessary to record the nexthops to provide accuracy.





B.2.3. Link State Advertisements Generation and Receiption

As seen in figure 13, the curves of the QoS LSAs traffic and the standard SPF LSAs traffic are closed to each other. The major difference between the two kind of traffic is their frequency. We always monitor the QoS LSAs traffic on the link but the SPF LSAs traffic is seen for two or three times. Thus, the frequency of the QoS LSAs generation is much higher than the SPF LSAs. However, even the maximum LSAs traffic amount is still a little fragment of the whole bandwidth of a link. The influence of the LSAs traffic can be ignored.

(四)、整合服務與差別服務網路之相容運作技術

In this work, we first explain the need of network processors for today's complex applications, and introduce the architecture and packet flow in IXP1200 shown in Fig. 14. Then we detail the mapping of DiffServ onto IXP1200, as shown in Fig. 15. There are two most important modules in DiffServ, classifier and scheduler, which are implemented with Multi-dimensional Range Matching [35] and Deficit Round Robin [36]. Finally we have external and internal benchmarks in order to find the bottlenecks in our implementation and possible design pitfalls of IXP1200.

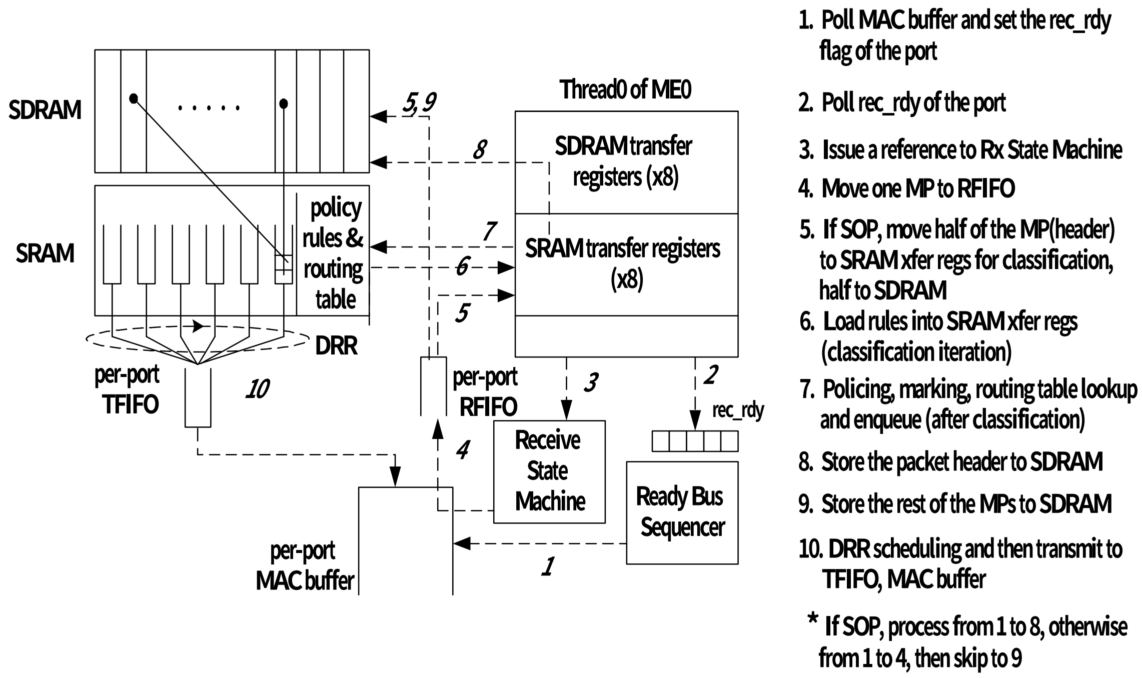


Fig. 14. Detailed DiffServ packet flow in IXP1200.

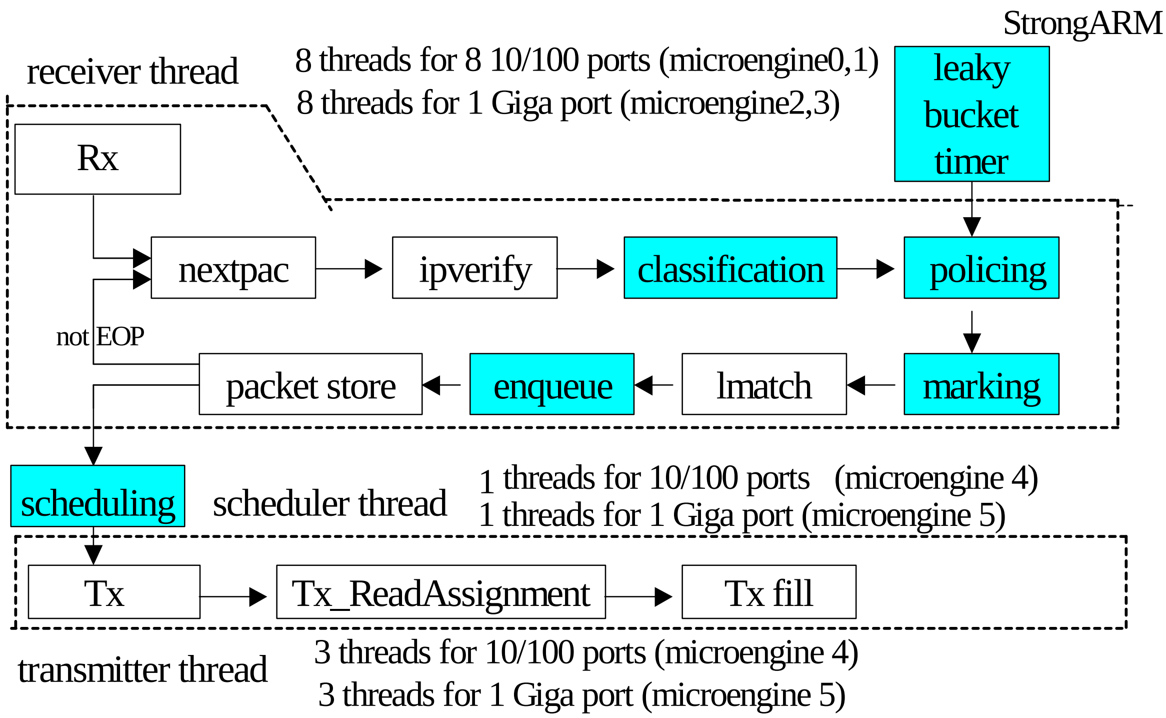


Fig. 15. Data-plane architecture of DiffServ edge router over IXP1200.

The results of external benchmarks, as in Fig. 16, Fig. 17 and Fig. 18, have shown that our implementation can support well the PHBs in DiffServ at an aggregated throughput of 290Mbps. .We also identify the *MAC buffer overflow* which is described below. Fig. 19 shows a diagram of packet reception. As we can see in Fig. 1, the rest of MPs, which are basic data units in IXP1200, are transferred from MAC buffer, RFIFO to SDRAM after the SOP (Start Of Packet) is classified. However, if SOP cannot be processed in time and the buffer is not large enough, the incoming MPs of the same packet could fill up the whole buffer and thus result in a packet drop, and then 100% packet loss.

Since both the *slow classification* and *small buffer* contribute to the MAC buffer overflow, we propose three solutions to avoid the two necessary conditions. They are (1). faster classification, (2). larger MAC buffer size, and (3). move the MPs into SDRAM before classification.

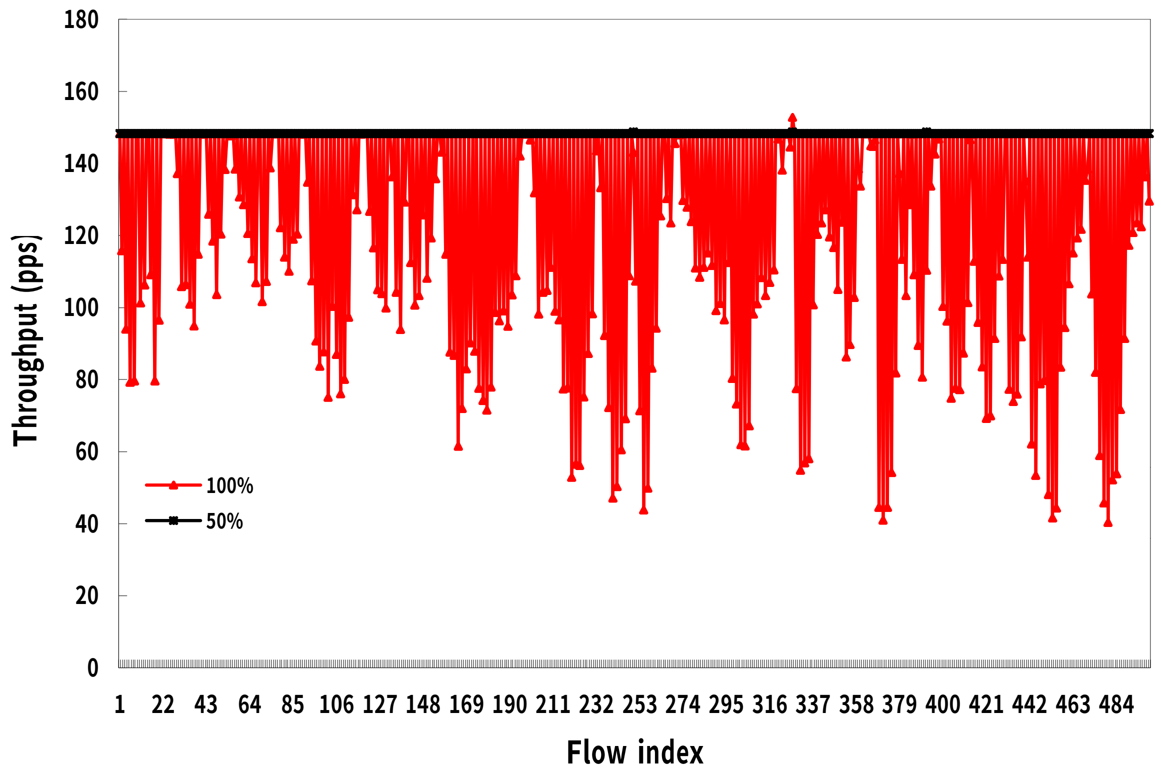


Fig. 16. Flow fairness test (Len=64bytes, 500 flows, BW=74400/500=148pps, normal case).

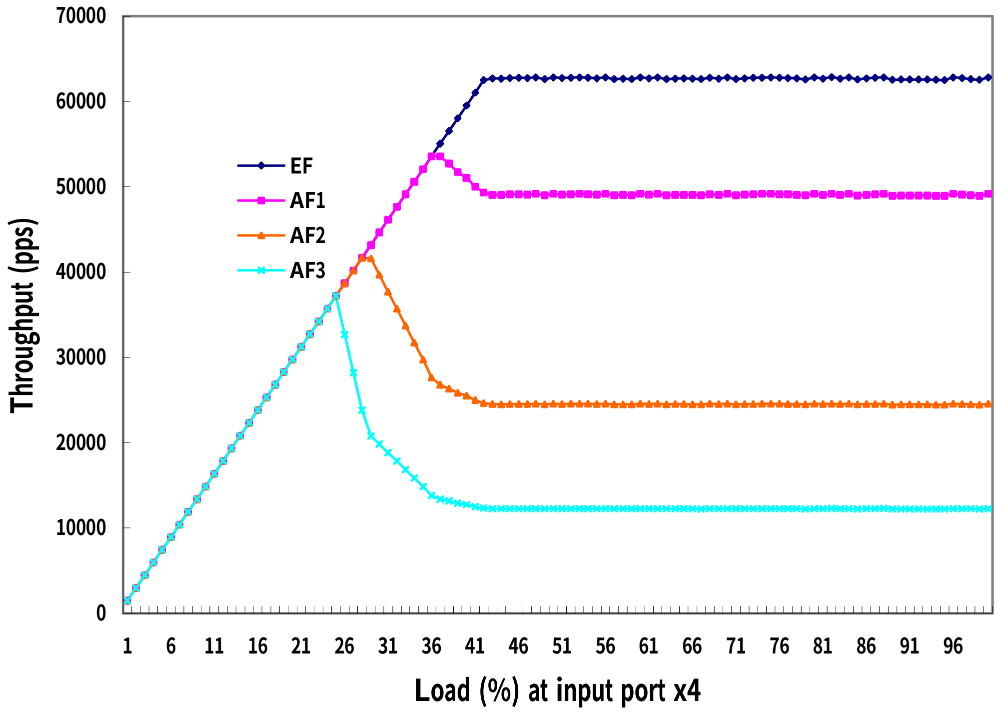


Fig. 17. Priority and bandwidth control test (Len=64byte, EF=62500pps)

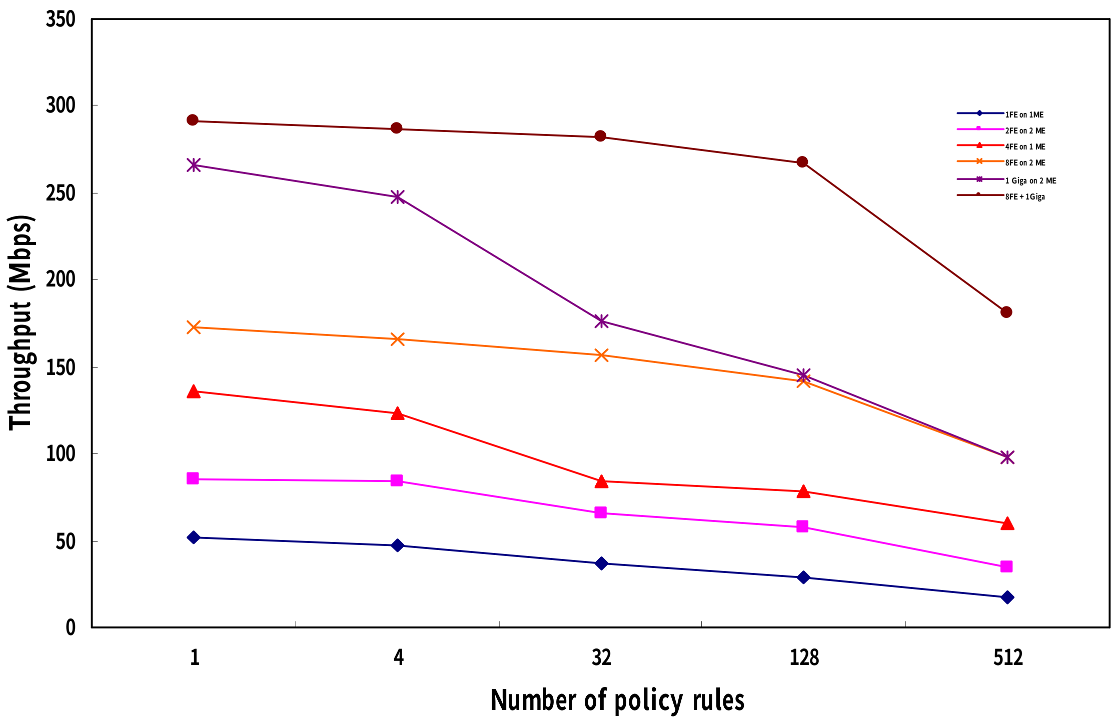


Fig. 18 Aggregated throughput (Len=64bytes, worst case)

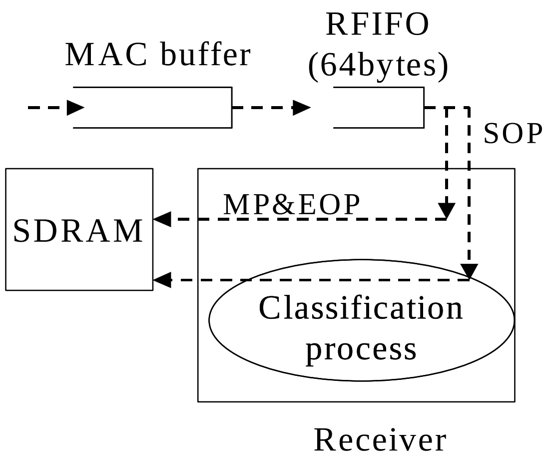


Fig. 19. Receiving process of a packet

In both external and internal benchmarks, we identify the *double bottleneck* of both exclusive SRAM access and the lack of computing power in microengines inside the Range Matching DiffServ, as shown in Table 1. That is, the Range Matching DiffServ could still suffer from the other bottleneck after one of them is solved. Three methods are proposed to solve the bottleneck of SRAM accesses that leads to the low utilization of receiver microengines. First is to divide one large SRAM into many smaller *banks* at different interfaces. This could shorten the queuing delay of requests in the command queue if the requested addresses are in different memory banks. Second, we may adopt a new memory architecture, for example, RAMBUS DRAM (RDRAM) [37] in IQ2000 [38] that has a peak bandwidth of up to 1.6GBps which is two to three times of what SRAM supports. Third, an additional *cache* can be used to reduce the number of memory accesses because the traffic in the same time period usually shows locality in lookups of policy and routing tables.

While the SDRAM is the bottleneck in IP forwarding [39], we observe that the bottleneck may shift from one functional unit to another depending on the specific service, algorithm and the way input traffic is allocated to threads, as shown in Table 1. We also find that the SRAM bottleneck does not necessarily occur at 100% utilization, it could even occur at *55%* when the access is *bursty*.

Table 1. Bottlenecks in DiffServs of two algorithms

Service or traffic allocation	Bottleneck
Linear search	SRAM
Range matching :	
Single input port	SRAM
8x100M input ports	ME
1 gigabit port	ME
8x100M and 1 gigabit	SRAM

第四章　結論

本年度計劃延續前兩年的研究成果並深入研究與發展相關的關鍵技術；在使用於寬頻網際網路之 Gigabit 路由器與訊務管制的計劃中，首先我們改進了排程模組，利用實現 FFQ 演算法的方式來改進舊有的 SCFQ 機制，使得公平性與延遲的保證都有改善。在可變封包長度的環境下，利用量化的概念，達到硬體重複使用，使得開發的時間得以降低。此外，我們進行了交換架構的研究，針對 CIOQ 架構加以分析，並提出 LCF/MUF 演算法，證明出它可以使 CIOQ 交換機仿效一個 output queueing 交換機，並分析了它在有限輸出 buffer 的情況下，其效能降低的情況。

在支援寬頻網際網路「負載控制服務」以量測為基礎的允諾控制與擁塞避免機制這個子計劃中， 主要研究為對於傳輸服務品質的要求較為寬鬆的一些網路上的即時媒體應用，這些應用不需嚴格封包延遲、但需避免封包遺失，因此，我們認為有必要提供可保障封包遺失率的無壅塞傳輸服務。本計劃首先實地量測台大校園網路的流量，確認流量負載具有常態分配的特性，利用此特性及量測窗方式，我們設計了一套動態頻寬分配與緩衝區管理模型，可有效降低封包遺失率。為了提升預測的精確度以及瞭解實際的成效，我們實作了 AppMeasure 來作為研究的輔助 AppMeasure。初步結果令人滿意，我們將持續發展以增進其效能。

在寬頻網際網路中服務品質路由之研究的子計劃中，主要研究分類服務品質保證的路由協定，它是以等級作為服務品質保證的依據，可在追求低阻絕率的同時，維持一定程度的計算負擔。我們設計了它的軟體架構，共分為幾個模組，並成功地在 GNU Zebra 平台上發展。

在整合服務與差別服務網路之相容運作技術的子計劃中，我們探討了利用網路處理器來發展 DiffServ 應用的技術，並與傳統的一般性處理器或是 ASIC 作比較。我們採用 LXP1200 網路處理器來發展，如果是單一種服務的情況下，它可以達到 1.8Gps 的處理量，如果是 DiffServ 的訊務，則有 200~300Mbps 的處理量，這樣的處理能力超越了一般性的處理器。

參考文獻

[1.] Hui Ahang, "Service Discipline for Guaranteed Performance Service in Packet -Switching Networks," Proceedings of IEEE, vol. 83, no.10, Oct., 1995.

[2.] A.K. Parekh and R.G. Gallager, "A Generalized Processor Sharing Approach to Flow Control – The Single Node case," Proc. INFOCOM '92, vol. 2, May 1992, pp. 915-24.

[3.] J.C.R. Bennett and Hui Zhang, "WF^2Q: Worst-case Fair Weighted Fair Queueing," in Proc. IEEEINFOCOM '96, San Francisco, CA, Mar. 1996, pp. 120-128.

[4.] S.Jamaloddin Golestani, "A self-clocked Fair Queueing Scheme for Broadband Applications," in Proc. IEEE INFOCOM '94, Toronto, CA, June 1994, pp. 636-646.

[5.] Dimitrios Stiliadis and Anujan Varma, "Frame-based Fair Queueing: A New Traffic Scheduling algorithm for Packet-Switch Networks," Tech. Rep.UCSC-CRL-95-39, July 18, 1995.

[6.] Dimitrios Stiliadis and Anujan Varma, "Latency-rate servers: A general model for analysis of traffic scheduler algorithms," Tech. Rep. UCSC-CRL-95-38, U.C. Santa Cruz, Dept. of computer Engineering, July 1995.

[7.] David A. Patterson, John L. Hennessy, "Computer Organization & Design, The Hardware/Software Interface," 2nd Edition, Morgan Kaufmann, 1998.

[8.] A. Varma and D. Stiliadis, "Hardware Implementation of Fair Queuing Algorithms for Asynchronous Traffic Mode Networks," IEEE Communications Magazine, vol. 35, no. 12, Dec. 1997, pp. 54-68.

[9.] B. Prabhakar and N. McKeown, "On the Speedup Required for Combined Input and Output Queued Switching," Computer Systems Lab, Technical Report CSL-TR-97-738, Stanford University.

[10.] Karol, M. Hluchyj, and S. Morgan, "Input Versus Output Queueing on a Space Division Switch," IEEE Trans. Commun., vol. 35, pp. 1347-1763, Dec.

[11.] N. McKeown, V. Anantharam, and J. Walrand, "Achieving 100% Throughput in an Input-queued Switch," Proc. IEEE INFOCOM'96, pp. 296-302.

[12.] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, "An Architecture for Differentiated Services", IETF RFC 2475, Dec. 1998.

[13.] C.S. Chang and J.A. Thomas, "Effective Bandwidth in High-Speed Digital Networks," IEEE Journal on Selected Areas in Communications, Vol. 13, No. 6, pp. 1091-1100, August, 1995.

[14.]D. D. Clark, S. Shenker, L. Zhang, "Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanism", SIGCOMM'92, 1992.

[15.]S. Floyd, "Comments on Measurement- Based Admissions Control for Controlled- Load Services", Technical Report, 1996. http://www.aciri.org/floyd/admit.html

[16.]R. Guerin, H. Ahmadi, and M. Naghshineh, "Equivalent Capacity and Its Application to Bandwidth Allocation in High-Speed Networks", IEEE Journal on Selected Areas in Communications, Vol. 9, No. 7, pp.968-981, Sep. 1991.

[17.]S. Jamin, P. B. Danzig, S. J. Shenker, and L. Zhang, "A Measurement- Based Admission Control Algorithm for Integrated Service Packet Networks", IEEE/ACM Transactions on Networking, Vol. 5, No. 1, pp.56-70, Feb. 1997.

[18.]S. Shenker, C. Partridge, and R. Guerin, "Specification of Guaranteed Quality of Service", IETF RFC2212, Sep. 1997.

[19.]J. Wroclawski, "The Use of RSVP with IETF Integrated Services", IETF RFC 2210, Sep. 1997.

[20.] "Cisco IOS NetFlow Technical Documents", http://www.cisco.com/warp/public/732/Tech/netflow/netflow_techdoc.shtml

[21.]The RSVP project, http://www.isi.edu/div7/rsvp/

[22.]The Zebra project, http://www.zebra.org/

[23.]G.apostolopoulos, R. Guerin, and S. Kamat, "Implementation and Performance Measurements of QoS Routing Extensions to OSPF," Proceedings of IEEE INFOCOM'99, New York, NY, March 1999.

[24.]Yun-Wen Chen, Ren-Hung Hwang, and Ying-Dar Lin. "Multipath QoS Routing with Bandwidth Guarantee,"

[25.]RFC 2676, "QoS Routing Mechanisms and OSPF Extensions," Jan 1998.

[26.]RFC 2205, R.Braden Ed, L.Zhang, S.Berson, S.Herzog, and S.Jamin, "Resource ReSerVation Protocol (RSVP)," Sep. 1997.

[27.]RFC 2328, J.Moy, "OSPF Version 2," April 1998.

[28.]Internet Draft, R.Braden, D.Hoffman, "RAPI -- An RSVP Application Programming Interface Version 5," August 1998.

[29.]NetScreen Appliances, http://www.netscreen.com/international/products/appliances.html#ns5

[30.]Intel NetStructure VPN Gateway Family, http://www.intel.com/network/idc/products/vpn_gateway.htm.

[31.]Intel Layer 3 Switching, "High speed LAN routing in an affordable switching solution," http://www.intel.com/network/tech_brief/layer_3_switching.htm.

[32.]eQoS Solutions for Service Providers using Riverstone Networks' Switch Routers, http://www.riverstonenet.com/technology/eqos.shtml.

[33.]Technical report on Hardware-Based Layer5 load balancer, http://www.nwfusion.com/research/2000/0501feat2.html.

[34.]Intel Electronic Design Kit, http://developer.intel.com/design/edk/product/ixp1200_edk.htm.

[35.]T.V. Lakshman, and D. Stiliadis, "High-Speed Policy-based Packet Forwarding Using Efficient Multi-dimensional Range Matching," ACM SIGCOMM'98.

[36.]M. Shreedhar, and G. Varghese, "Efficient Fair Queuing Using Deficit Round-Robin," IEEE/ACM Transactions on Networking, June 1996, vol. 4, no. 3, pp. 375-385.

[37.]Data Sheets of RDRAM, http://www.rambus.com/developer/support_rdram.html

[38.]IQ2000 Network Processor, VITESSE Corp, http://www.vitesse.com/products/categories.cfm?family_id=5&category_id=16

[39.]T. Spalink, S. Karlin, L. Peterson, "Evaluating Network Processors in IP Forwarding," *Technical Report TR-626-00*, Computer Science, Princeton University, Nov. 1999.