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)　　　　※
※子計畫五: 整合服務與差別服務網路之相容運作技術 ※
※　　　　　　　　　　　　　　　　　　　　　　　　※
※※※※※※※※※※※※※※※※※※※※※※※※※

計畫類別：□個別型計畫　　■整合型計畫

計畫編號：NSC　－　89　－ 2219 －　E　－　009　－ 022

執行期間： 89 年　8　月　1　日至　90　年　7　月　31　日

計畫主持人：　　林盈達 教授

共同主持人：

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：交通大學資訊科學系

中　華　民　國 90　**年**　9　**月**　15　**日**

行政院國家科學委員會專題研究計畫成果報告
寬頻網際網路服務品質保證(III)
子計畫六：整合服務與差別服務網路之相容運作技術(III)／林盈達
計畫編號：NSC 89-2219-E-009-022
執行期限：89 年 8 月 1 日至 90 年 7 月 31 日

主持人：林盈達　交通大學資訊科學系

一、中文摘要

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：網路處理器，差別式服務，IXP1200，延展性，SRAM，SDRAM。

Abstract

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: Network Processor, DiffServ, IXP1200, scalability, SRAM, SDRAM

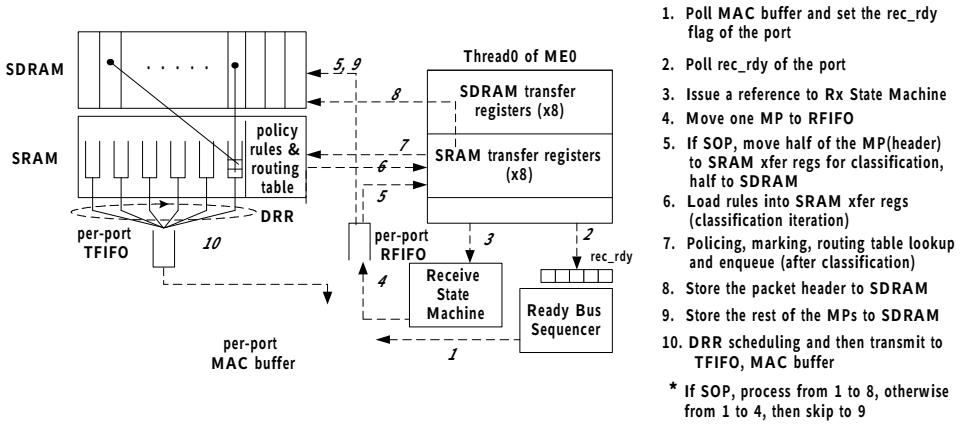

Fig. 2. Detailed DiffServ packet flow in IXP1200

二、緣由與目的

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 [1], VPN gateways [2], Layer 3 switches [3], DiffServ routers [4] and Web switches [5]. 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 [6] 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.

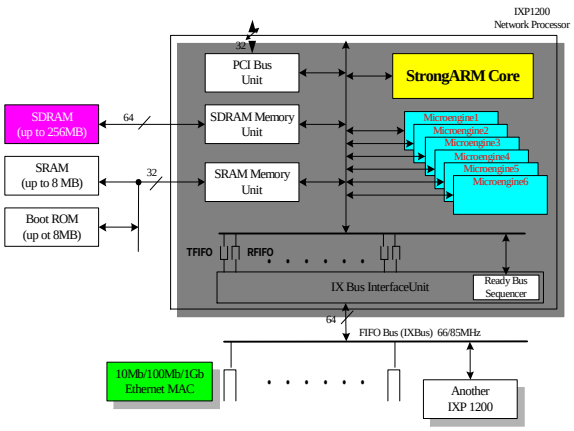

Fig. 1 Hardware architecture of IXP1200

三、結果與討論

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. 2. Then we detail the mapping of DiffServ onto IXP1200, as shown in Fig. 3. There are two most important modules in DiffServ, classifier and scheduler, which are implemented with Multi-dimensional Range Matching [7] and Deficit Round Robin [8]. Finally we have external and internal benchmarks in order to find the bottlenecks in our implementation and possible design pitfalls of IXP1200.

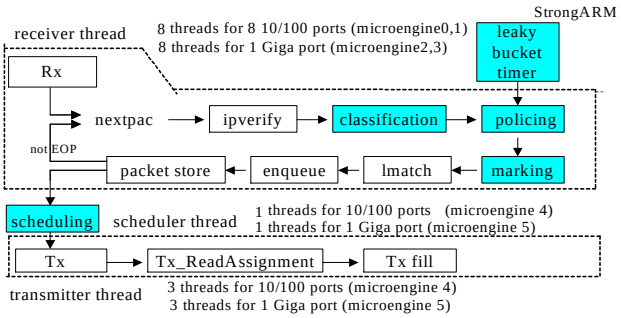

Fig. 3. Data-plane architecture of DiffServ edge router over IXP1200

The results of external benchmarks, as in Fig. 4, Fig. 5 and Fig. 6, 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. 7 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.

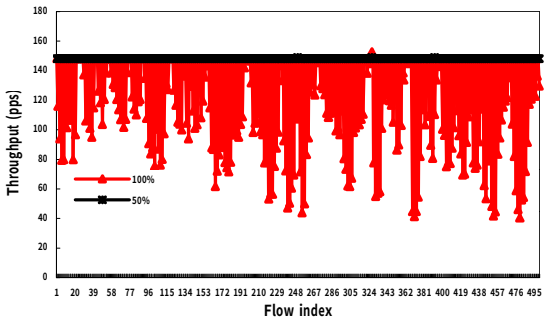

Fig. 4. Flow fairness test (Len=64bytes, 500 flows, BW=74400/500=148pps, normal case)

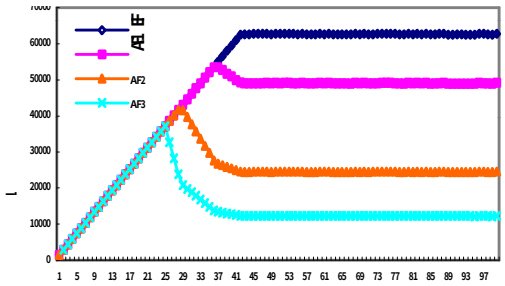

Fig. 5. Priority and bandwidth control test (Len=64byte, EF=62500pps)

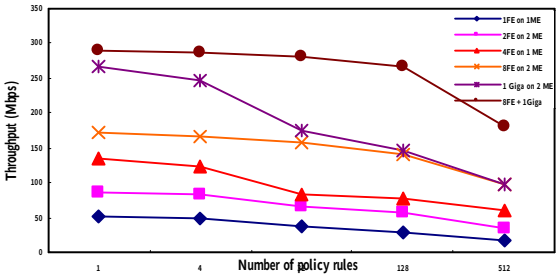

Fig. 6 Aggregated throughput (Len=64bytes, worst case)

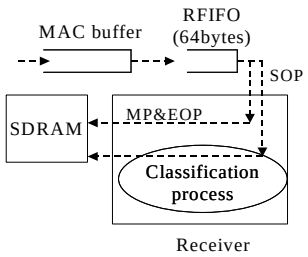

Fig. 7. Receiving process of a packet

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) [9] in IQ2000 [10] 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 [11], 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

四、計畫成果自評

This study investigates the feasibility of using network processors (IXP1200 in our study) as an alternative platform for DiffServ applications, compared with the traditional general-purpose processor and the ASIC's. From the external benchmark we can see that IXP1200 supports an aggregated of 1.8Gbps for simple forwarding service, although, it degrades to 200~300Mbps when performing DiffServ. However, it does outperform the general-purpose processor solution.

We had also submitted this paper to INFOCOM'02 in the hope of sharing the development experience in network processors.

五、參考文獻

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

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

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

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

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

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

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

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

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

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

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