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(II)—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－009－003

執行期間：　88 **年** 08 **月** 01 **日至**　89 **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　89　**年**　　**月**　　**日**

目錄

Abstract **2**

中文摘要 **10**

第 一 章 前言 **13**

第 二 章 計劃緣起與目的 **16**

一、大容量路由器與訊務管制技術 16
二、寬頻網際網路之允入控制及排程技術 16
三、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 18
四、寬頻網際網路中服務品質路由之研究 19

第 三 章 研究方法及結果 **25**

（一）、使用於寬頻網際網路之 GIGABIT 路由器與訊務管制技術 25
A. 適用可變長度封包之 Crossbar 交換機 25
B. 訊務區分器 28
B.1 三級訊務區分演算法 28
B.2 搜尋程序 30
B.3 演算法分析 30
（二）、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 31
A. 流量量測 31
B. 無壅塞傳輸服務模型 31
（三）、寬頻網際網路中服務品質路由之研究 33
A. Hierarchical Routing 33
B. Cost Functions for Hierarchical QoS Routing 34
C. Update Poicies 35
D. Crankback Approaches 36
E. Simulation Results 37
（四）、整合服務與差別服務網路之相容運作技術 49
A. QOSPF with Overflowed Cache (PER-PAIR/OC) 49
B. QOSPF With Overflowed Cache With Two-Phase Routing (PER-PAIR/OCTP) ... 49
C. QOSPF Using Per-Class Routing Mark (PER-PAIR/PC) 50
D. Performance Evaluation 50
D.1 Network Model and Traffic Model 50

D.2 Granularity and Performance Metrics 51
（五）、寬頻網際網路之允入控制與訊務排程 53
A. The VQS System 53
B. Implementation Architecture 54
C. Results and Merit Review of the Project 54
第 四 章 結 論 56
參考文獻 59

LIST OF TABLE

Table **page**

1. Link capacity assignments……………………………………………………………. 44
2. The Traffic Model Of The Simulation………………………………………… 51
3. Cache Granularities Of The Simulation……………………………………….…51
4. Summary Of The Simulation Results……………………………………………52

LIST OF FIGURE

Figure **page**

1. 系統概觀………………………………………………………………………… 25
2. 系統架構………………………………………………………………………… 26
3. 埠控制器功能方塊圖……………………………………………………… 27
4. SLIP 交換機與我們的交換機在不同負載量之封包平均延遲與輸入佇列長度的比較：（a）平均延遲（b）平均輸入佇列長度。………………………………………… 29
5. 1999/8/25 凌晨 2:00、取樣時間為 100ms 的流量負載分配圖…………………… 32
6. 1999/8/25 凌晨 2:00、不同取樣區間(1sec, 100ms, 20ms)的流量負載分配圖……………………………………………………………………………… 33
7. 動態頻寬分配與緩衝區管理模型示意圖………………………………………… 33
8. Network topology……………………… ………………………………………… 38
9. Fractional reward loss…………………………………. ………..……….………… 39
10. Crankback per requested calls. ………………….……………………………... 40
11. Fractional reward loss of five update policies……………………………….....41
12. Average number of crankback per connection request ……………………………42
13. Average number of re-aggregation per unit of time..…………………………….. 42
14. Average number of re-aggregation per unit of time yielded by DCU, DCUH, and LRBU policies……………………………………………………………………… 43
15. Network topology.…………………………………………………………………..45
16. Fractional reward loss of different crankback schemes under PNNIU update policy…46
17. Average hops count of crankback of different crankback schemes under PNNIU update policy……………………………………………………………………………47
18. Fractional reward loss of different crankback schemes under LRBU update policy……48
19. Average hops count of crankback of different crankback schemes under LRBU update policy……………………………………………………………………………48
20. Overflowed Cache mechanism (Per-pair/OC)………………………………………49
21. VQS implementation architecture………………………………………….…..…..55
22. Mean delay under an increasing *S*1 weight………………………………………55

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 consisting of five sub-projects constantly improves the results in the first year and further investigates the following key technologies:

A. High-capacity (Gigabit) routers: Sub-project I investigated two key technologies of developing high-capacity routers: switch architecture and longest prefix matching with hardware (or hardware routing). We employed space-division architecture, such as the crossbar, to build a real large-capacity router. In addition, we designed queue management and fast scheduling algorithms to (partially) remove head-of-line blocking in an attempt to improve the router's throughput. This study mainly focuses on designing a switch for variable-length packets to reduce idle time on output port in achieving better performance in terms of throughput and packet delay. Through simulation, the result showed that switches designed for fixed-length packets are inadequate for network environment of variable-length packets on the fly. Besides, to efficiently classify arriving packets, this study proposed a three-phase packet classification algorithm based on destination/source IP addresses, destination/source port numbers and protocol ID fields and designed efficient hardware routing schemes to speed up routing decision.

B. Admission control/QoS scheduling: QoS scheduling for broadband Internet is aimed to provide bounded delay and fairness while retaining a minimum of computational complexity. Prevailing weight-based scheduling disciplines advocate the use of multiple

queues and engage in timestamp computation. These disciplines achieve either superior QoS performance at the expense of higher complexity or degraded performance in return for computational simplicity. In the sub-project IV of this year, we have designed a weight-based Versatile QoS Scheduler (VQS) and its feasible VLSI hardware implementation architecture. VQS is capable of being implemented in various network elements in broadband Internet facilitating proper trade-off balance between performance and complexity. Taking advantage of simpler single-queue management and lack of timestamp computation, VQS governs packet insertion in a shared data structure comprising a sequence of fixed-size *windows* based on weights. Within a given widow, the maximum number of packets from a session is proportional to the session weight and the Window Size (*WS*). Simulation results demonstrate that, applying a smaller *WS* for high-power network elements, VQS performs as superior as WF^{2}Q with respect to throughput fairness, mean delay, and worst-case delay fairness. Moreover, compatible to WF^{2}Q, VQS outperforms WFQ with respect to 99-percentile delay bound and jitter in the presence of traffic burstiness.

C. Traffic measurement and statistics collection/admission control: Sub-project III 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 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.” Simulation results show that in order to maintain a maximum packet loss rate, it is important that the system avoids operating at heavy loads, i.e. high link utilization. In a link sharing system, dynamic bandwidth allocation based on input loads can effectively avoid congestion. Furthermore, when combined with active queue management, it can further accommodate transient traffic bursts for non-self-similar traffic.

D. QoS routing: For reducing network information to achieve scalability in large ATM networks, ATM Private Network-to-Network Interface (PNNI) adopts hierarchical routing. Consequently, although routing complexity is significantly reduced, numerous issues in PNNI routing require further study to achieve more efficient, accurate, scalable, and QoS-aware routing. In this year, we proposed several methods to achieve efficient, scalable, and QoS-aware ATM PNNI routing. First, an efficient aggregation scheme, referred to as Asymmetric Simple, is proposed. The aggregated routing information includes available bandwidth, delay and cost. Second, two approaches for defining link costs are investigated, namely, the Markov Decision Process (MDP) approach and the Competitive On-Line (COL) routing approach, and these are compared with the Widest Path (WP) approach. Third, a dynamic update policy, referred to as the dynamic cost-based update (DCU) policy, is proposed to improve the accuracy of the aggregated information and the performance of hierarchical routing, while decreasing the frequency of re-aggregation and information distribution. Finally, we proposed CIS (Crankback Information Stack) and CT (Cost Threshold) approaches to reduce crankback overhead. Simulation results demonstrate that the proposed Asymmetric Simple aggregation scheme yields very good network utilization while significantly reducing the amount of advertised information. Between these two links cost functions, the MDP approach provides a systematic method of defining call admission function and yields better network utilization than the COL approach. The proposed DCU policy also yields an enhanced network utilization while significantly reducing the frequency of re-aggregation. Meanwhile, the proposed CIS and CT approaches reduce crankback overhead significantly. Especially, the combination of CIS and CT approach achieves further improvement.

E. RSVP (ReSource reserVation Protocol) to PHB (Per-Hop Behavior) mapping: Sub-project V investigates how granularity of routing decision significantly affects the scalability and blocking performance of QoS routing based on QoS routing extensions to OSPF. Three mechanisms, overflowed-cache, two-phase routing, and per-class routing mark, are also proposed to achieve computational and storage scalability as well as low blocking probability

in wire-speed packet-switching networks. Simulation results of various routing and forwarding granularities, including per-destination, per-pair, per-flow, per-pair/ overflowed-cache, per-pair/two-phase, per-pair/class, indicate that the proposed mechanisms can significantly lower blocking probability, increase fairness, as well as lower storage and computational overhead. Also, two or three classes are sufficient for per-class routing which is suitable for DiffServ core networks. Comparing flow driven mechanisms like per-flow and per-pair/overflowed-cache with topology driven mechanisms like per-destination, per-pair, per-pair/two-phase, and per-pair/class reveals that the former usually perform better in blocking probability, fairness, and state accuracy, while the latter result in less overhead.

Keywords: Broadband Internet, Quality of service, QoS routing, Gigabit router, Signaling, Policing, Scheduling, Network planning, Management

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，本計畫由五個子計劃整合，除了延續第一年的研究成果，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：子計劃（一）擴充交換機的功能，使之具有第三層硬體路由技術、第四層防火牆功能、支援群播功能、port trunking 以及八種優先權處理。交換機的架構主要採用 crossbar，並發展佇列管理和快速排程演算法來移除佇列前端擁塞問題，進而提昇路由器的傳輸率；特別是本交換機適用於可變長度封包的網路環境，以減小輸出埠閒置時間來提高效率降低封包延遲。此外，此子計畫設計了一個三級式多欄位訊務區分器演算法，透過 hashing、search tree、linear search 三個步驟，可以將 filter 集合有效分散開來，減少搜尋所需要的運算量，以達成高速訊務區分的目的。

B. 寬頻網際網路之允入控制及排程技術：寬頻網際網路上的 QoS (Quality-of-Service)排程必須提供有限延遲與公平性的考量且保持最小之計算複雜度。Prevailing weight-based 的排程原則提倡應用 multiple queues 和 engage 於 timestamp 的計算。這些原則在高複雜度的花費下，可以達到最佳之 QoS 效能，也可以為了計算簡化，達到次級之效能。子計畫（二）設計了一個 weight-based Versatile QoS Scheduler (VQS) 和其可行之 VLSI 硬體實作架構。為了促進較適當之效能與複雜度間的平衡，VQS 可以在寬頻網際網路內用不同之網路元件來實作。VQS 利用較簡單之 single-queue 與不需 timestamp 計算之優點，以 weight 來控制封包插入於一分享式之資料結構。此資料結構包含了一系列固定大小之 windows。在一給定的 window 內，由一個 session 來的最大封包個數，是和此 session 的 weight 和 window 大小成正比。實驗結果證明，應用較小之 window 於 high-power 的網路元件，VQS 可以在 throughput fairness、mean delay、與 worst-case delay

fairness 等項目上表現和 WF^{2}Q 一樣好。甚至在和 WF^{2}Q 相容與具備 traffic burstiness 的情況下，VQS 較 WFQ 優越約 99%之 delay bound 和 jitter。

C. 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壅塞避免控制：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動態調整資料的傳送，他們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，子計劃（三）之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。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。當網路流量的自我相似性不是很顯著時，配置適量的緩衝區更有助於封包遺失率的降低。

D. 寬頻網際網路中服務品質路由之研究：ATM Private Network-to-Network Interface (PNNI) 提出階層式選徑通信協定來減少網路資訊以實現一大型 ATM 網路的可擴展性，必然地，降低了選徑的複雜度，但是如何使 PNNI 網路提供一更高效率、正確資訊、擴展性與服務品質保證的選徑問題是需要許多的研究。子計劃（四）提出四項方法以實現一更高效率、正確資訊、擴展性與服務品質保證的 PNNI 選徑機制。第一、提出一高效率且低複雜度的聚集方法，稱之為 Asymmetric Simple 聚集法；第二、提出以馬可夫理論(MDP)為基礎的鏈路成本定義機制，並將此法與線上競爭法(COL)和最寬頻寬法(WP)作比較；第三、提出以成本為機礎的動態資訊更新政策(DCU)，以提高資訊的正確性同時減少資訊更新頻率；最後，以 CIS 與 CT 方法減少 crankback overhead。結果顯示 Asymmetric Simple 聚集法可有效提供良好的網路使用率同時減低網域劇集的複雜度。MDP 方法提供系統化的 call admission 機制且考慮 calld 到達率，因此與 COL 和 WP 方法相比，MDP 具有良好的網路使用率。再者，DCU 方法適用於動態網路流量因此可得良好網路使用率同時明顯減少網路資訊更新頻率。而所提出的 CIS 與 CT 方法明顯

降低 crankback overhead**，尤其以結合兩者時更為顯著。**

E. **整合服務與差別服務網路之相容運作技術：子計劃（五）探討路由決策的粗細度對服務品質路由的延展性及阻絕率的影響。並且提出三個機制--溢流快取、兩階段式路由、和路由記號分類轉送等技術，目的在線速封包網路上達到計算上及儲存上的延展性，並降低阻絕率。在各種粗細度的系統模擬中，結果顯示所提機制確有不錯的效果，尤其路由記號分類轉送技術使用二或三種分類很適合差別服務的核心網路中。總結比較，**flow driven **的路由技術如** per-flow **和** per-pair/overflowed-cache **會有較佳阻絕率、公平性、和狀態準確性，**data driven **的路由技術如** per-destination、per-pair、per-pair/two-phase **和** per-pair/class **會有較少的** overhead。

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、管制、排程、網路規劃與管理**

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。

本年度計劃延續第一年的研究成果並深入研究與發展相關的關鍵技術，包括有：

子計畫一（交通大學電信工程研究所李程輝教授主持）擴充交換機的功能，使之具

有第三層硬體路由技術、第四層防火牆功能、支援群播功能、port trunking 以及八種優先權處理。交換機的架構主要採用 crossbar，並發展佇列管理和快速排程演算法來移除佇列前端擁塞問題，進而提昇路由器的傳輸率；特別是本交換機適用於可變長度封包的網路環境，以減小輸出埠閒置時間來提高效率降低封包延遲。此外，此子計畫設計了一個三級式多欄位訊務區分器演算法，透過 hashing、search tree、linear search 三個步驟，可以將 filter 集合有效分散開來，減少搜尋所需要的運算量，以達成高速訊務區分的目的。

子計畫二（交通大學資訊工程研究所楊啟瑞教授主持）主要在研究服務排程技術與允入控制，以提供各種服務品質。寬頻網際網路上的 QoS (Quality-of-Service)排程必須提供有限延遲與公平性的考量且保持最小之計算複雜度。本年度的計畫設計了一個 weight-based Versatile QoS Scheduler (VQS) 和其可行之 VLSI 硬體實作架構。為了促進較適當之效能與複雜度間的平衡，VQS 可以在寬頻網際網路內用不同之網路元件來實作。

子計畫三(台灣大學資訊管理研究所孫雅麗教授主持)主要研究如何透過量測訊務來提升網路使用效率及避免擁塞發生；近來許多網路上的即時媒體應用對於傳輸服務品質的要求較為寬鬆，針對不需嚴格封包延遲、但需避免封包遺失的應用，有必要提供可保障封包遺失率的無壅塞傳輸服務。於此子計劃，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，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，但若要更精確地控制封包遺失率，則需要進一步的研究。

子計畫四（中正大學資訊工程研究所黃仁竑副教授主持）主要在研究如何選定 QoS 的 metrics 與 routing algorithm，以及搭配 RSVP 時，若資源保留要求失敗時，如何 re-routing；此子計劃提出四項方法以實現一更高效率、正確資訊、擴展性與服務品質保證的 PNNI 選徑機制。第一、提出一高效率且低複雜度的聚集方法，稱之為 Asymmetric Simple 聚集法；第二、提出以馬可夫理論(MDP)為基礎的鏈路成本定義機制，並將此法與線上競爭法(COL)和最寬頻寬法(WP)作效能比較；第三、提出以成本為機礎的動態資

訊更新政策(DCU)，以提高資訊正確性的同時減少資訊更新頻率，降低了選徑的複雜度；最後，以 CIS 與 CT 方法減少 crankback overhead。

子計畫五(交通大學資訊科學研究所林盈達副教授主持)主要探討路由決策的粗細度對服務品質路由的延展性及阻絕率的影響；並且提出三個機制—溢流快取、兩階段路由、與路由記號分類轉送技術，目的在限速封包網路上達到計算上級儲存上的延展性，並降低阻絕率。

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：

高效能網路交換機採用 crossbar 交換結構作為交換核心漸漸成為一種趨勢。而 crossbar 交換機需要一套交換排程演算法（Switch scheduling algorithm）來決定輸出入埠配對, 目前存在的排程演算法, 例如 PIM (Parallel Iteration Matching)、RRM (Round-robin matching）與 SLIP，均是為固定長度封包所設計，雖然經由切割再重組的方式可以處理可變長度的封包，卻可能造成輸出、入埠所需的緩衝空間與封包交換延遲的增加。本研究針對可變長度封包設計一交換機架構，以減小輸出埠閒置時間來提高效率，並降低封包延遲。此外，高速的多欄位訊務區分器是現今發展服務保證網路最重要的核心技術之一。傳統的訊務區分器僅根據第三階層的標頭欄位來分類，並不適用於較高階層之表頭欄位分類器；另外，網路頻寬與使用者需求的指數倍數成長使得路由器的傳輸率要求相對提高。所以，為了支援多欄位分類與高傳輸率的需求，我們提出一個三級式多欄位訊務區分器演算法，透過 hashing、search tree、linear search 三個步驟，可以將 filter 集合有效分散開來，減少搜尋所需要的運算量，以達成高速訊務區分的目的。

二、寬頻網際網路之允入控制及排程技術：

Scheduling disciplines proposed in the literature have been either single-queue or multiple-queue-based. Single-queue- based disciplines advocate the maintenance of a single shared queue for each output link. Different-session packets destined to the same output link are inserted in the shared queue in accordance with, for instance, the deadlines or priorities of packets. Packets are then transmitted in a FIFO manner. Consequently, scheduling complexity completely resides in the enqueueing process. Examples of this class include Earliest Deadline First (EDF), Threshold Based Priority (TBP), and Precedence with Partial Push-Out (PPP). The EDF discipline was shown to successfully support tight delay bound. However, it undergoes two major limitations- a priori deadline assumption and high implementation complexity due to packet sorting. Although TBP and PPP were justified effective for switches supporting two priorities, they fail to provide bounded delay and throughput fairness in the presence of malicious sessions.

Multiple-queue-based disciplines, on the other hand, adopt multiple queues maintained at each output link, one for each session. Packets arriving from different sessions are simply placed at the end of their corresponding queues. Scheduling complexity in this class resides in the dequeueing process instead. Prevailing disciplines in this class, which are weight-based, include Weighted Fair Queueing (WFQ), Worst-case Fair Weighted Fair Queueing (WF^{2}Q), Self-Clocked Fair Queueing (SCFQ), and Frame-based Fair Queueing (FFQ).

In this project, we aim to design a weight-based, highly versatile QoS scheduler, referred to as VQS, capable of being implemented in diverse network elements facilitating proper trade-off balance between performance and complexity. Taking advantage of simpler single-queue management and lack of timestamp computation, VQS governs the insertion of packets belonging to the same output link in a shared data structure comprising a sequence of fixed-size *windows*. Within a given widow, the maximum number of packets from a session is proportional to the session weight and the Window Size (*WS*). Packets being placed at the same window are transmitted on a FIFO basis, limiting short-term unfairness to within a window interval. Packets being arranged outside of the window trigger new windows to be activated, enforcing weight-proportional service to be exerted.

三、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)以及差異化服務(Differentiated Service)。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)，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。

目前文獻上以量測為基礎(measurement-based)的流量控制主要應用在允諾控制(admission control)上。傳統的允諾控制以參數為基礎(parameter- based)，如的 equivalent capacity 及的 effective bandwidth，亦即必須事先(a priori)明確描述出既有流量及新進資料流的行為才能發揮作用，但目前在網路上的應用程式日益多元，很難精確而完整地得到允諾控制所需的相關參數。以量測為基礎的允諾控制以實際量測的方式了解網路流量目前的狀況，如此一來就可以破除需要事先知道流量參數的限制，使允諾控制在應用上有更大的彈性。

在諸多以量測為基礎的允諾控制演算法中，所提出的方法是使用量測窗(measurement window)估計並預測網路流量，並將此資訊應用於允諾控制的判斷，以提

供一個具有寬鬆延遲上限保證的預測服務(predictive service)**。另外，可以利用** equivalent capacity **的原理來描述流量，它以量測值來估計目前流量的平均到達率**(average arrival rate)**，利用** Hoeffding bound **計算出** equivalent capacity **用於判斷目前流量所需使用的頻寬，希望提供的是一種讓封包遺失率維持某一水準以下的服務。此法雖然亦著眼於提供無壅塞服務，但卻未考慮到緩衝區對減輕封包遺失的效益。**

四、寬頻網際網路中服務品質路由之研究：

The ATM PNNI standard adopts a source-based hierarchical routing for supporting scalability and security in a large network. The main advantage of the hierarchical routing is reducing large communication overhead while achieving efficient routing. The scalability and performance of hierarchical networks depend on various design schemes, such as the aggregation scheme for aggregating routing information, the cost functions for defining link and path costs, and update policies for advertising the aggregated information. Nevertheless, how to design these schemes remains an open issue. In this project, we propose several solutions, including the efficient Asymmetric Simple aggregation scheme, the QoS-capable COL and MDP link cost approaches, the dynamic cost-based update policy, and CIS and CT for reducing crankback overhead to achieve an efficient hierarchical QoS routing in large ATM networks. These methods are briefly described below.

A. Hierarical Routing

In this project, we study the PNNI standard, source-based hierarchical routing in ATM networks. The source-based hierarchical routing problem can be decomposed into two issues: how to aggregate routing information and how to perform hierarchical routing. For routing information aggregation, Iwata *et al.* proposed two aggregation schemes, star and simple node, with three aggregation versions, aggressive, conservative and simple no-aggregation. These schemes transform a non-linear programming problem into a linear problem for the corresponding QoS parameters. Meanwhile, Lee proposed a spanning tree aggregation scheme, and Awerbuch *et al.* compared the performances of several aggregation schemes, including a star with radius equal to half the cost of the network diameter (DIA), a star with radius equal to half the average cost between border nodes (AVE), Minimum Spanning Tree (MST), Random Spanning Tree (RST), and t-spanner. Our earlier work proposed a novel aggregation scheme, called Asymmetric Simple and compared it with two existing aggregation schemes (Simple Node and Full-Mesh) using various performance metrics, such as representation size and representation accuracy for routing information, and network revenues.

For hierarchical routing, Guo *et al.* applied probabilistically routing to hierarchical networks. Meanwhile, Mieghem presented the unicast hierarchical routing based on PNNI standard. Furthermore, , Montgomery *et al.* and Xie *et al.* applied the theory of reduced load approximation to analyze the blocking probability of PNNI hierarchical networks. Finally, Hao *et al.* investigated the call rejection probability for routing with crankback.

Our numerical results first demonstrate that an effective aggregation scheme reduces overhead for call set up while yielding high traffic throughput. The proposed Asymmetric Simple aggregation scheme can yield competitive performance compared to the Full Mesh aggregation scheme.

B. Cost Functions for Hierarchical QoS Routing

In this project, we define the objective of hierarchical QoS routing as to maximize network revenue under the constraint that each established connection is guaranteed with certain QoS requirements. This optimization problem is generally re-formulated as, for each new arriving connection, to find the path with minimum cost while satisfying certain QoS requirements. Two issues can be identified that differ from QoS routing in flat networks. First, owing to the inaccuracy of aggregated information, a chosen hierarchical path may not satisfy the end-to-end QoS requirement. In this case, the crankbank scheme can be employed to reroute the connection to an alternate path. The crankback scheme is further discussed in chapter 2. The second issue is how to define the link cost function and aggregate link cost functions in a peer group. A good link cost function should comprise two properties; maximizing network revenue by minimizing path cost, and providing a systematic call admission function.

In this project, we study two approaches for defining link cost functions in a hierarchical network, namely the Markov Decision Process (MDP) approach and the Competitive On-Line (COL) approach. The theory of Markov decision process is a pledge method in a lot of network-related issues. Various network control schemes have been developed based on the Markov decision process. For example, many MDP-based routing algorithms, which compute the link cost based on the MDP theory, have been proposed and demonstrated to perform very well. However, Gawlick, *et al.*, proposed an on-line optimal routing algorithm, referred to as the Competitive On-Line (COL) algorithm. This approach defines the link cost function as an exponential function of the residual bandwidth. They have shown good routing performance based on this cost function. We proposed MDP-based cost function for hierarchical QoS routing.

In this project, we compare the performance of the MDP-based and the COL-based cost functions with the Widest Path (WP) approach, which routes an incoming connection to the path with maximum residual bandwidth. The residual bandwidth of a path is defined as the minimum of the residual bandwidth of all links on the path. The simulation results show that the MDP and the COL approaches outperform the WP approach. The MDP approach yields the best network utilization. A further advantage of the MDP approach is that it provides a systematic call admission function.

C. Update Poicies

On the other hand, PNNI adopts time-based update policy which is inadequate to cope with dynamic network traffic. Furthermore, the accuracy of aggregated information is depended on the update interval, with a reduced update interval meaning more accurate aggregated information. However, in this situation the overhead of re-aggregation and information distribution increases. Awerbuch et al. proposed the logarithmic update approach, which is based on the residual bandwidth of a link, to reduce the computational overhead of re-aggregation.

An event-based update policy typically suffers from oscillation, which can be avoided by hystersis. The technique of hysteresis has been applied to various areas of high-speed networks. For example, Jong applied hysteresis to ATM rate control to enhance system stability, and Orda et al. proposed an adaptive virtual path allocation policy using hysteresis to prevent excessive processing of requests due to oscillations around thresholds. Meanwhile, Shun-Ping Chung and Jin-Chang Lee propose a dynamic reservation with hysteresis as CAC for cellular multiservice networks.

We proposed an event-driven update policy based on the link cost. Furthermore, to avoid oscillation, the hystersis technique was applied. The proposed policies are called the Dynamic Cost-based Update (DCU) policy and the DCU with hysteresis (DCUH) policy. The performance of the proposed policies is compared to other time-driven and event-driven update policies, including the PNNI time-based update approach (PNNIU), full update approach (FU), logarithm of residual bandwidth update approach (LRBU), and dynamic cost-based update policy without hysteresis. Simulation results show that the DCUH policy performs best among these update policies.

D. Crankback Approaches

Alternate path routing approaches can be used to achieve lower connection blocking probabilities and higher network throughput in ATM networks. Conventional alternate path routing techniques tend to fall into two classes: progressive control. Hwang, R.H. *etc.* noted that progressive control, as compared to the originating control, provides a higher blocking probability but smaller connection setup time The primary advantage of progressive control is the ability to provide fast connection setup, while the disadvantages of progressive control are the use of sub-optimal alternate path. Chung proposed an ICD (information about the crankback destination nodes) approach in a connection setup messages, hence they could predict the crankback destination node. Meanwhile, Spieqel *et al.* presented an approach to combine the features of progressive and originating control approaches, allowing for fast connection setup as well as near minimum cost paths. Felstaine *et al.* proposed to allocate a “quota” to the PGs along the message path and then to “sub-allocate” quota to the son PGs of these PGs.

The purpose of this section is to study mechanisms for reducing the crankback overhead in PNNI hierarchical networks. Two heuristics are proposed herein to reduce crankback overhead. First, if we can predict where to crank back, referred to as the destination node, the call setup message need not to crank back to the source node, thus the call set up overhead at the common nodes along the path can be reduced. Second, the aggregated path cost represents the expected cost for setting up the call on the path of O-D pair. Therefore, a call setup message on a path with high path cost is likely to be blocked. Consequently, avoiding setting up a call on a path with high path cost will reduce the crankback overhead. Based on these two heuristics, we proposed two approaches to reduce crankback overhead. The first approach adds additional information, referred to as CIS, to predict the crankback destination node. In this approach, we keep CIS in the call setup message while crossing the PG. When call setup message encounters a block link, the intermediate node using CIS to find the crankback destination node. Hence, the call setup message needs not to crank back to the ingress node. The second approach uses aggregated path cost, referred to as CT, to determine whether call setup on alternate paths should be tried. In PNNI hierarchical network, paths with smaller hierarchical path cost are tried first. If the aggregated path cost information is accurate and a call setup message failed on the path with minimum cost, then call set up on alternate paths with higher path cost will likely to be failed. Simulation results show that both of these two approaches reduce the crankback overhead significantly. Furthermore, when we combine CIS with CT, referred to as CIS_CT, the crankback overhead can be reduced more conspicuously.

五、整合服務與差別服務網路之相容運作技術：

To achieve both low blocking probability and high scalability, this study proposes three QoS routing extensions to OSPF, *overflowed-cache*, *two-phase routing*, and *per-class routing mark*. The *overflowed-cache* mechanism divides the packet-forwarding cache into a per-pair cache (*P-cache*) and an overflowed per-flow cache (*O-cache*). The flows that the P-cache indicated paths cannot satisfy with the required QoS are routed individually and their forwarding decisions are *overflowed* into the O-cache. The *two-phase routing* reserves a block of bandwidth that exceeds the bandwidth requirement of a flow when the first flow between an S-D pair is established. The *per-class routing* aggregates QoS flows into a number of classes via a marking technique. Therefore, flows with the same *mark* for the specific S-D pair are routed on the same path. This routing mechanism is suitable for DiffServ networks, where packets are marked at edge routers and fast-forwarded in the core network.

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. 適用可變長度封包之 Crossbar 交換機

整個系統劃分成輸出入埠控制器與交換核心兩個子系統。所有網路介面都做成相同的模組，輸出入埠控制器位於這些獨立的介面卡上。模組的一端是網路介面，另一端則是與背版相連的匯流排插槽。交換核心則位於交換機背版，所有進入交換機的封包都經由交換機背版完成交換的動作。系統架構類似傳統電話交換機。交換機的組成如圖 3.1 所示，這樣的架構使得系統具有彈性，能夠依照不同流量需求增減介面卡數目，達到較好的成本效益比。此外，若使用具有熱抽換功能的匯流排介面（如 Compact PCI），則可在不關閉系統的情況下進行介面卡更換。這個特性對骨幹網路交換機是非常重要的。

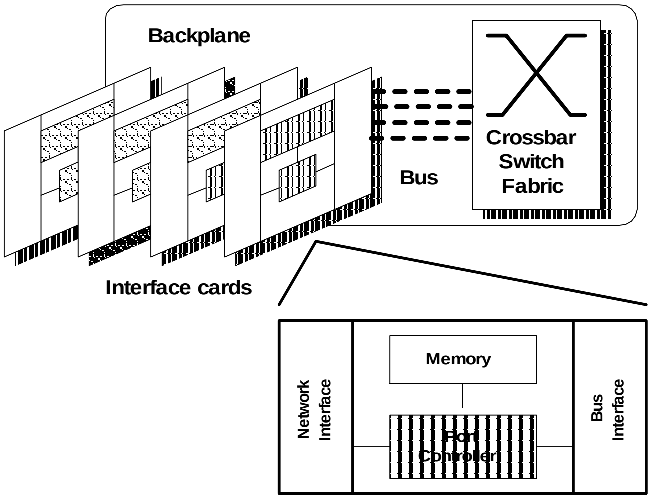

圖 3.1　系統概觀

A.1 系統架構

整個系統架構方塊圖顯示於圖 3.2。由網路進入交換機的封包首先由輸出入埠控制器（I/O Port Controller，如圖 3.3 所示）對標頭做處理，並將封包佇存在輸入埠的緩衝區（memory）內。每個輸出入埠控制器都有一組 P 通道（P_n in & P_n out）與交換核

心（如圖 3 所示）連接，此通道是對交換核心下達指令以及傳送封包的途徑。交換核心藉由串連所有埠控制器的回應通道(RSP)送出連接請求指令。核心內的仲裁器用來決定連結的建立與否，仲裁結果同樣透過此通道傳出。群播封包則經由專屬的群播通道（MC）傳送到各個輸出入埠控制器。此外，還有一個控制通道（CTRL）作為中央處理器與輸出入埠之間溝通的管道。

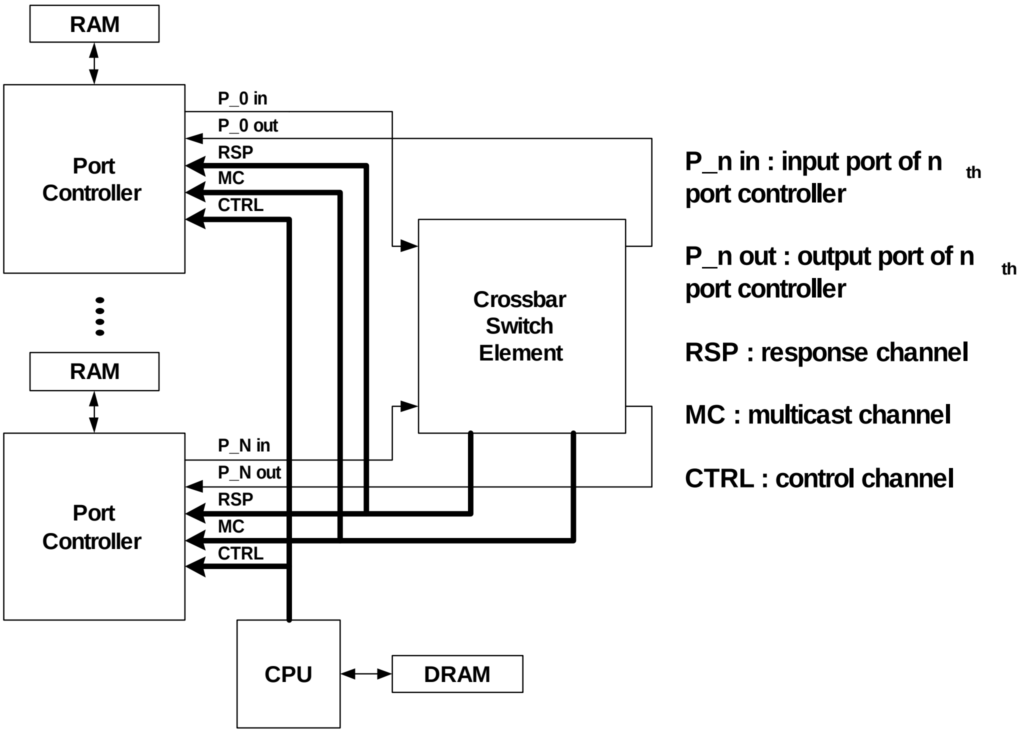

圖 3.2　系統架構

A.2 輸出入埠控制器

輸出入控制器負責封包標頭的解讀與轉換，以及對封包進行存取控制。它可以是單埠或多埠控制器。以多埠的共享緩衝式交換晶片作為埠控制器能組成大埠數交換機。目前商品化的單一晶片共享緩衝式交換機已可容納十六個 Fast Ethernet 網路介面。控制器內部的功能方塊如圖 3.3。其中 Data Path 單元以分時多功的方式處理由網路介面（PHY+MAC）送入的封包，並將封包標頭送至查表單元（Table Lookup Engine），封包內容則交由佇列管理單元（Queue Manager）處理。佇列管理單元根據查表結果，將送往不同輸出埠的封包分別放入 VOQ，並依照封包等級排序。佇列管理單元還負責將

各 VOQ 狀態及排頭封包的等級告知交換核心介面單元（Crossbar Interface）。封包的傳輸透過埠控制器與交換核心間的三個管道進行；交換核心介面單元負責送出封包，P_n out 通道接收交換核心送出的封包，群播封包則經由 MC 通道收送。查表單元根據封包標頭找出對應的輸出埠與該封包的等級，然後將結果告知佇列管理單元。此外，它還必須根據由中央控制器介面（CPU Interface）送入的指令進行路由表（routing table）與分級表(classification table)的更新。中央控制器也經由此介面向佇列管理單元蒐集網管所需的資訊。分開的控制通道可避免不同作用的資料發生壅塞，提高運作速率。與交換核心的介面由兩個單元組成。回應監視單元（RSP Monitor）負責 RSP 通道訊號的解碼，核心介面單元則根據解碼的結果，將對應的 VOQ 狀態或佇列內的封包，經由 P_n in 通道送出。

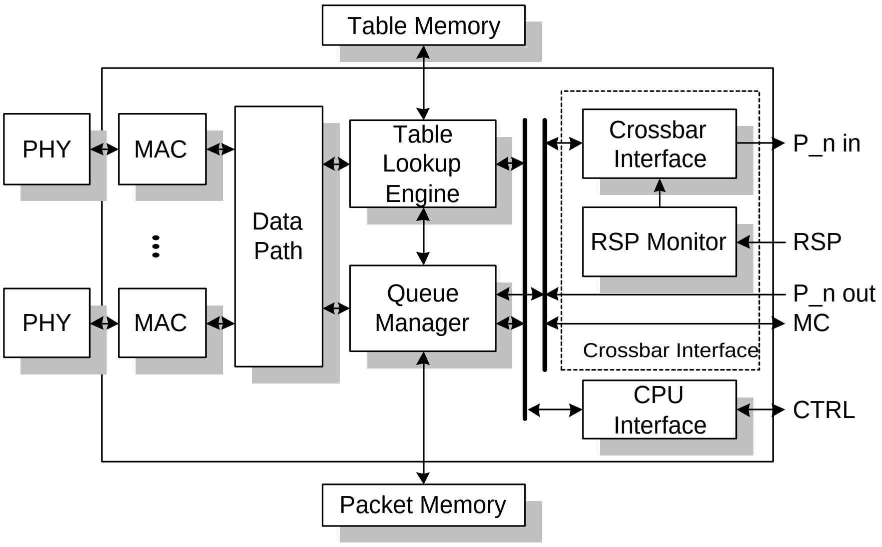

圖 3.3　埠控制器功能方塊圖

A.3 交換核心

交換核心包含了命令解碼器(Command Decoder)、仲裁器(Arbitration Logic)、回應編碼器（Response encoder）、狀態監視器（Status Monitor）、請求佇列（Request Queue）縱橫式交換陣列（Crossbar matrix）與點陣控制電路(Matrix controller)等數個部分。圖 3.4 顯示了這些功能區塊的配置。

命令解碼器跟據協定單元的標頭辨別其為發號或資料協定單元，並將協定單元所乘載的資料送至仲裁器或交換陣列。若協定單元的內容是 VOQ 的連接等級，則解碼器根據提出請求的輸出埠號，由 BUS1 將內容傳送至對應的仲裁器。若內容為封包資料，則直接送往交換陣列。仲裁器根據我們設計的方法決定連線建立與否，並將結果由 BUS2 送往陣列控制器與回應編碼器。陣列控制器依據仲裁結果控制縱橫式陣列內部交換元件的啟閉。回應編碼器則把仲裁結果做格式轉換後由 RSP 通道送出。交換陣列的輸出除了直接接出交換核心外，還透過 BUS3 送至狀態編碼器。狀態監視器則根據是否偵測到代表封包結束的字符(T+I)或暫停指令，將輸出埠忙碌或閒置狀態轉換成輸出埠請求指令，並記錄在請求佇列中。請求指令與回應指令以多工方式共用 RSP 通道。

為了使交換機有較高的運作效能，我們以輸出埠的角度來設計交換機；當輸出埠由忙碌變成閒置狀態時，立即向所有輸入埠要求送封包過來，除非都沒有以此輸出埠為目的的之封包進入交換機，或擁有送往此輸出埠封包的輸入埠正處於忙碌狀態，則輸出埠將經過一或數個仲裁週期後，開始送出封包。交換機以減少輸出埠閒置時間來提高較率。而仲裁器以找出具最高優先權輸入埠的方式使交換機具有差別服務功能，並使用輪轉法避免飢饉現象。

A.4 交換機效能模擬

以軟體模擬來分析比較 SLIP 交換機與本計劃所設計的交換機之封包延遲與輸入佇列長度，如圖 3.4（a）與（b）所示，相關參數設定為 s_size=16、MaxPktSize=1500、MinPktSize=64、PSN=5000、及 CellSize= 64。結果顯示根據固定長度封包所設計的交換機並不適用於可變封包長度的網路環境。

B. 訊務區分器

B.1　三級訊務區分演算法

■ 第一級：Hashing

選擇特定 bits 的排列將所有的 filter 區分開來。考慮於五個欄位共 104 個 bit 裡選 m 個 bit 出來，即可將 filters 所構成的 space 切割成 2^m部分；理想上，每個部分至多有

${}^{N}\!/_{2^m}$ 個 filter（其中 N是 filter 總數），依此方法，將可大幅減少需要找尋的 filter 數目。

我們採用的策略是於 Source IP address 與 Destination IP address 各取三個 bits, 取一個 bit

以

(a)

(b)

圖 3.4 SLIP 交換機與我們的交換機在不同負載量之封包平均延遲與輸入佇列長度的比較：(a）平均延遲　(b）平均輸入佇列長度。

以區分 UDP 與 TCP 的 protocol 欄位, 總共 7 個 bits 來進行 Hashing 的運作, 以期將 filter space 作最平均的分割。

- 第二級：Search Tree

根據各個filter的IP prefix建立search tree。建立順序為Source IP address> Destination Ip address>Protocol欄位；此外，為了提升速度，所建立的樹並非單純的二元樹，而是能夠一次檢查 m bits 所建立的 2^m-ary Search Tree。建構樹的過程中，為了避免建立整個search tree所浪費的記憶體空間，我們限定當節點的filter數超過一定值的時候，才建立下一層的節點，如此便可簡化建立search tree的複雜度。

- 第三級：Conflict Check and Sorting

將位於每個部分的filters根據其cost來排序，以加快整體封包search的速度。然而，由於IP address是採用longest prefix matching，即mask較少的filter擁有較高的priority，因此可能造成 mask 較長的 filter 永遠不會被 match；此為 conflict problem。為了防止conflict發生，於排序之前，尚需執行conflict check運作，定義如下：1)若mask較長之Filter i，其實際的優先權≤ mask較短之Filter j的優先權，則移除Filter i。2)若mask較長之Filter i的優先權> mask較短之Filter j的優先權，則將此兩者的costs對調。完成conflict check之後，依filters’ cost之高低排序，由於filter數目已非常少，一般的線性排序就可相當快速地完成，故此，我們採用氣泡排序法（bubble sort）。

B.2 搜尋程序

1） 根據Hashing特定位置的bits，找到對應search tree的根（root）。

2） 依照Source IP、Destination IP的順序，利用prefix尋找節點，當節點上的filter數目不為零時，表示路徑搜尋完成。

3） 以Linear Search的方式尋找符合條件的filter，第一個符合條件的filter即為best matching filter。

B.3 演算法分析

假設filter總數為 n、hashing key為 b bits、2^m-array search tree的depth為 $h(n)$，則平均每個節點所含的filter數目為 $n \big/ 2^{m h\left(n/2^b\right)}$，因此search一個filter所需的複雜度約為：

$$O\left(h\left(\frac{n}{2^b}\right)\right) + O\left(\frac{n}{2^{m h\left(n/2^b\right)}}\right)$$。

建構 2^m-ary search tree時，可能會根據IP prefix作複製的動作。假設depth=K時，

檢查的 prefix 恰小於 address mask，即 $m\cdot K< Address\ Mask<m\cdot (K+1)$，則 depth=$K$+1 時，必須複製$[(K+1)\cdot m\text{-}Address\ Mask]\cdot 2$ 個 filters。所以當最後 depth=P（$P>K+1$）時，總共需要複製$\left(2^m\right)^{P-(K+1)}[(K+1)\cdot m\text{-}Address\ Mask]\cdot 2=2^{m(P-K-1)+1}[(K+1)\cdot m\text{-}Address\ Mask]$。現假設 n 個 filters 中，有 i 個 filters 需要複製，其複製起始的 depth 以 K_i表示，address mask 為 $aMask_i$，則最後 search 所需的運算複雜度約為

$$O\left(h\left(n/2^b\right)\right)+O\left(\frac{n-i+\sum_{a=1}^{i}2^{m(h(n)-K_i-1)+1}\left[\left(K_i+1\right)\cdot m-aMask_i\right]}{2^{mh\left(n/2^b\right)}}\right)$$。

（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. 流量量測

為了以較精確的方式描述流量變化，我們希望找出流量負載(traffic load)是否遵循某種統計分配。經實際量測台大校園網路一段 oc3 link 的流量，發現在不同時段、不同取樣區間(sampling interval)下，當取樣數量夠多時，聚合(aggregate)流量負載均相當近似於常態分配(以量測到的流量負載之平均數及標準差為參數)，如圖 3.5 的機率密度函數(pdf, probability density function)所示。

需要注意的是，取樣區間的不同會影響量測結果。如圖 3.6 所示，同一時段，採用較小的取樣區間，由於可以捕捉到更多流量暴衝(burst)現象，也就是更能反應出流量的瞬間變化，因此流量負載的變異程度也較大，從 pdf 圖來看就形成略為扁平而兩端分佈較廣的鐘形(bell shape)；反之，當取樣區間較大時，就不容易看出流量的變異。

從這樣的結果可以發現，在量測流量負載時，必須謹慎選擇取樣區間，以免低估或高估流量的瞬間變化，對流量產生誤判，進而影響網路資源的配置。

B. 無壅塞傳輸服務模型

利用流量負載具有常態分配的特性及希望善用緩衝區的理念下，本研究設計了一個動態頻寬分配與緩衝區配置模型，如圖 3.7 所示。基本原理是將時間軸切割成數個量測

窗(大小為 T), 以 S 為取樣區間在一個量測窗內取樣 T/S 個流量負載, 給定一個比例 δ (0< δ <1), 依據常態分配的特性, 以該量測窗內流量負載的平均值 *AvgLoad* 及標準差 *StdLoad* 等式(1)計算出一個對應的頻寬使用量 *EC*，使得流量負載有 δ 的機會大於 *EC*，並以 *EC* 作為該量測窗的頻寬保留量, 其中等式(1)的 z 值是使累積機率等於 1- δ 所對應的標準常態變數值。由於流量負載具常態分配的特性，因此應該有(T/S)× δ 筆流量負載值會超過 *EC*，由於已保留了 *EC* 的頻寬，因此配置緩衝區時，只要考慮這(T/S)× δ 筆流量負載值超過 *EC* 的部份就可以控制封包遺失率，令量測窗內的最大流量負載值為 *MaxLoad*，若對每個高過 *EC* 的流量負載都保留(*MaxLoad-EC*)×*S* 的緩衝區容量, 理論上就不應有任何封包被丟棄(drop)，綜合上述說明，該量測窗所需配置的緩衝區即如等式(2)所示。

$$EC = AvgLoad + z \cdot StdevLoad \qquad (1)$$

$$\begin{aligned} BufferSize &= \frac{T}{S} \cdot \text{¢} \cdot [(MaxLoad - EC) \cdot S] \\ &= T \cdot \text{¢} \cdot (MaxLoad - EC) \qquad (2) \end{aligned}$$

模擬結果顯示，當 δ 、取樣區間及量測窗大小搭配得宜時，不管流量是否具有自我相似(self-similar)的特性，封包遺失率都可以控制在相當低的水準，甚至能夠讓封包遺失率維持在 0。但此法尚未找到 δ 、取樣區間、量測窗大小及目標封包遺失率 ε 之間的關係，亦即還無法預知在某種上述參數組合下所能達成的封包遺失比例，如何更精準地控制封包遺失率，有待更進一步的研究。

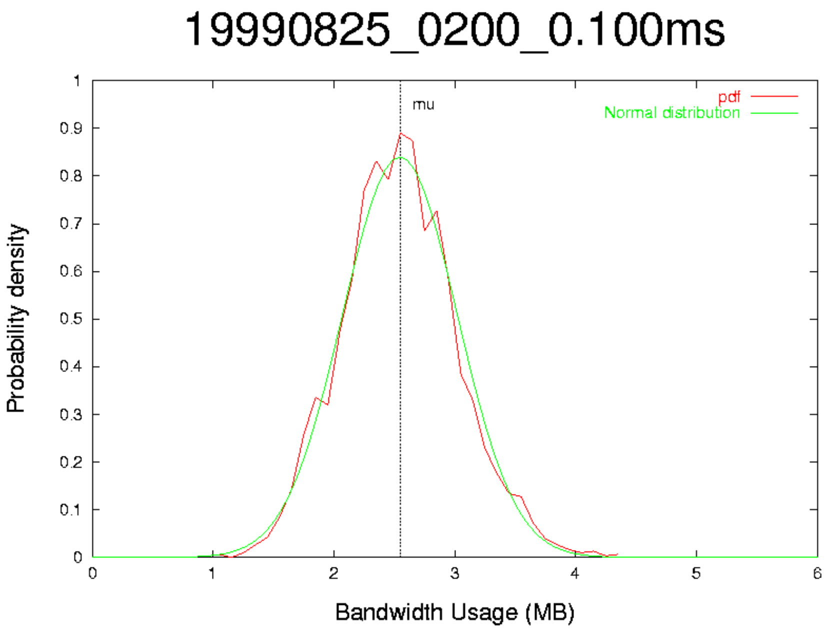

圖 3.5. 1999/8/25 凌晨 2:00、取樣時間為 100ms 的流量負載分配圖

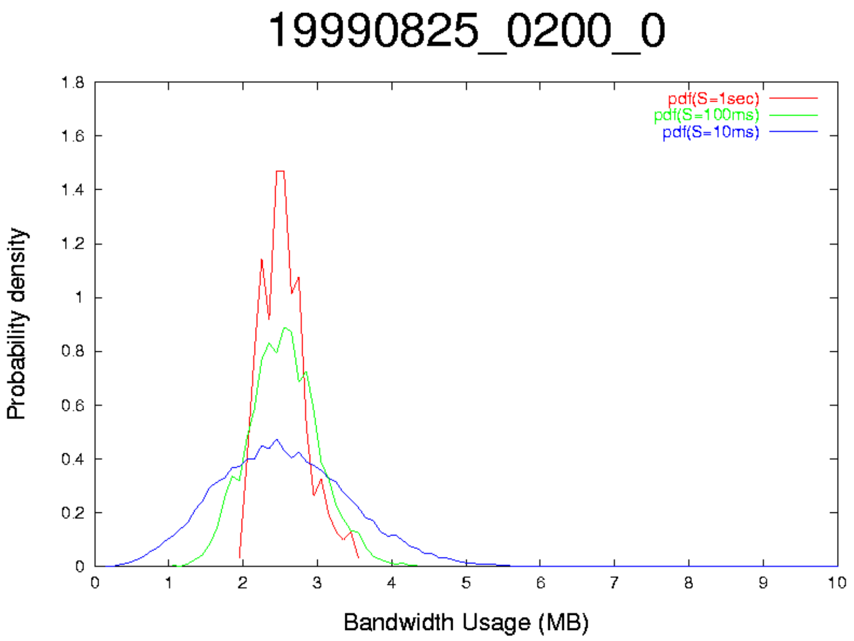

圖 3.6. 1999/8/25 淩晨 2:00、不同取樣區間(1sec, 100ms, 20ms)的流量負載分配圖

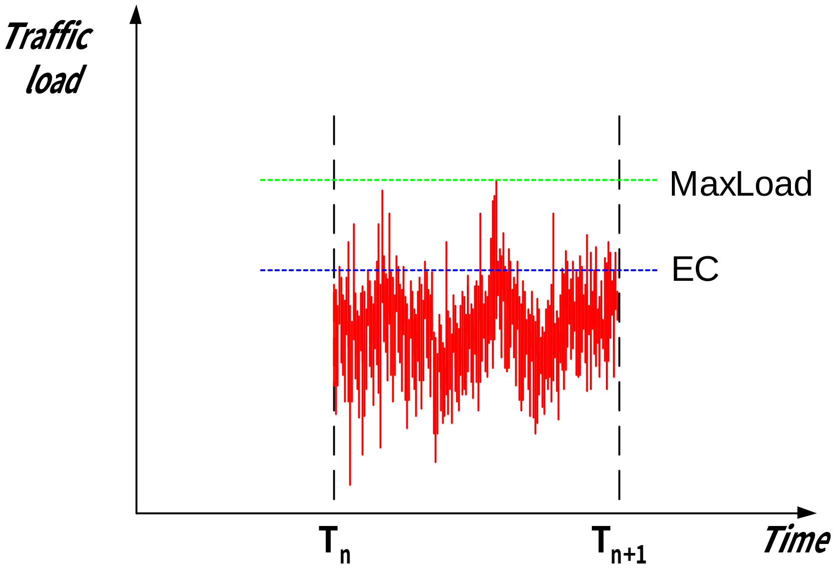

圖 3.7 動態頻寬分配與緩衝區管理模型示意圖

（三）、寬頻網際網路中服務品質路由之研究

A. Hierarchical Routing

In this project, we study the PNNI standard, source-based hierarchical routing in ATM networks. The source-based hierarchical routing problem can be decomposed into two issues: how to aggregate routing information and how to perform hierarchical routing. For routing information aggregation, Iwata et al. proposed two aggregation schemes, star and simple node, with three aggregation versions, aggressive, conservative and simple no-aggregation. These schemes transform a non-linear programming problem into a linear problem for the corresponding QoS parameters. Meanwhile, Lee proposed a spanning tree aggregation scheme, and Awerbuch et al. compared the performances of several aggregation schemes, including a star with radius equal to half the cost of the network diameter (DIA), a star with radius equal to half the average cost between border nodes (AVE), Minimum Spanning Tree (MST), Random Spanning Tree (RST), and t-spanner. Our earlier work proposed a novel aggregation scheme, called Asymmetric Simple and compared it with two existing aggregation schemes (Simple Node and Full-Mesh) using various performance metrics, such as representation size and representation accuracy for routing information, and network revenues.

For hierarchical routing, Guo et al. applied probabilistically routing to hierarchical networks. Meanwhile, Mieghem presented the unicast hierarchical routing based on PNNI standard. Furthermore, Montgomery et al. and Xie et al. applied the theory of reduced load approximation to analyze the blocking probability of PNNI hierarchical networks. Finally, Hao et al. investigated the call rejection probability for routing with crankback.

Our numerical results first demonstrate that an effective aggregation scheme reduces overhead for call set up while yielding high traffic throughput. The proposed Asymmetric Simple aggregation scheme can yield competitive performance compared to the Full Mesh aggregation scheme.

B. Cost Functions for Hierarchical QoS Routing

In this project, we define the objective of hierarchical QoS routing as to maximize network revenue under the constraint that each established connection is guaranteed with certain QoS requirements. This optimization problem is generally re-formulated as, for each new arriving connection, to find the path with minimum cost while satisfying certain QoS requirements. Two issues can be identified that differ from QoS routing in flat networks. First, owing to the inaccuracy of aggregated information, a chosen hierarchical path may not satisfy the end-to-end QoS requirement. In this case, the crankbank scheme can be employed to reroute the connection to an alternate path. The crankback scheme is further discussed in section 2. The second issue is how to define the link cost function and aggregate link cost functions in a peer group. A good link cost function should comprise two properties; maximizing network revenue by minimizing path cost, and providing a systematic call admission function.

In this project, we study two approaches for defining link cost functions in a hierarchical network, namely the Markov Decision Process (MDP) approach and the Competitive On-Line (COL) approach. The theory of Markov decision process is a pledge method in a lot of network-related issues. Various network control schemes have been developed based on the Markov decision process. For example, many MDP-based routing algorithms, which compute the link cost based on the MDP theory, have been proposed and demonstrated to perform very well.   In this project, we adopt the idea for its simplicity and efficiency. However, Gawlick, et al., proposed an on-line optimal routing algorithm, referred to as the Competitive On-Line (COL) algorithm. This approach defines the link cost function as an exponential function of the residual bandwidth. They have shown good routing performance based on this cost function. We proposed MDP-based cost function for hierarchical QoS routing.

In this project, we compare the performance of the MDP-based and the COL-based cost functions with the Widest Path (WP) approach, which routes an incoming connection to the path with maximum residual bandwidth. The residual bandwidth of a path is defined as the minimum of the residual bandwidth of all links on the path. The simulation results show that the MDP and the COL approaches outperform the WP approach. The MDP approach yields the best network utilization. A further advantage of the MDP approach is that it provides a systematic call admission function

C. Update Poicies

On the other hand, PNNI adopts time-based update policy which is inadequate to cope with dynamic network traffic. Furthermore, the accuracy of aggregated information is depended on the update interval, with a reduced update interval meaning more accurate aggregated information. However, in this situation the overhead of re-aggregation and information distribution increases. Awerbuch et al. proposed the logarithmic update approach, which is based on the residual bandwidth of a link, to reduce the computational overhead of re-aggregation.

An event-based update policy typically suffers from oscillation, which can be avoided by hystersis. The technique of hysteresis has been applied to various areas of high-speed networks. For example, Jong applied hysteresis to ATM rate control to enhance system stability, and Orda et al. proposed an adaptive virtual path allocation policy using hysteresis to prevent excessive processing of requests due to oscillations around thresholds.

We proposed an event-driven update policy based on the link cost. Furthermore, to avoid oscillation, the hystersis technique was applied. The proposed policies are called the Dynamic Cost-based Update (DCU) policy and the DCU with hysteresis (DCUH) policy. The performance of the proposed policies is compared to other time-driven and event-driven update policies, including the PNNI time-based update approach (PNNIU), full update approach (FU), logarithm of residual bandwidth update approach (LRBU), and dynamic cost-based update policy without hysteresis. Simulation results show that the DCUH policy performs best among these update policies.

D. Crankback Approaches

We further study mechanisms for reducing the crankback overhead in PNNI hierarchical networks. Two heuristics are proposed to reduce crankback overhead. First, if we can predict where to crank back, referred to as the destination node, the call setup message need not to crank back to the source node, thus the call set up overhead at the common nodes along the path can be reduced. Second, the aggregated path cost represents the expected cost for setting up the call on the path of O-D pair. Therefore, a call setup message on a path with high path cost is likely to be blocked. Consequently, avoiding setting up a call on a path with high path cost will reduce the crankback overhead. Based on these two heuristics, we proposed two approaches to reduce crankback overhead. The first approach adds an addition information, referred to as CIS, to predict the crankback destination node. In this approach, we keep CIS in the call setup message while crossing the PG. When call setup message encounters a block link, the intermediate node using CIS to find the crankback destination node. Hence, the call setup message needs not to crank back to the ingress node. The second approach uses aggregated path cost to determine whether call setup on alternate paths should be tried. In PNNI hierarchical network, paths with smaller hierarchical path cost are tried first. If the aggregated path cost information is accurate and a call setup message failed on the path with minimum cost, then call set up on alternate paths with higher path cost will likely to be failed.

E. Simulation Results

This section evaluates the performance of hierarchical routing with different aggregation schemes and cost functions via simulations. Figure 3.8 shows the network topology of the simulations. The capacity of each link is 155 Mbps, except for link(A.3.2-B.1.3) and link(B.1.3-A.3.2), which are 622 Mbps. Each link has a delay of 1ms.

In the simulations, the network supports two classes of traffic. The class 1 traffic has a bandwidth requirement of $b_1=1$ Mbps, and an end-to-end delay bound of $d_1=15$ ms. Class 1 traffic is assumed to arrive at any O-D pair with the same arrival rate $a^{\mu}_{1} = a$, while the class 2 traffic is assumed to have a bandwidth requirement of $b_2=5$ Mbps, an end-to-end delay bound of $d_2=15$ ms, and an arrival rate of $a^{\mu}_{2} = a$ / b_2 between each O-D pair. The average holding times of these two classes of traffic are normalized to unity.

The 95% confidence intervals of the simulation results shown in the following figures are obtained from 25 independent runs. For each run, the simulated time is 1100 units of mean

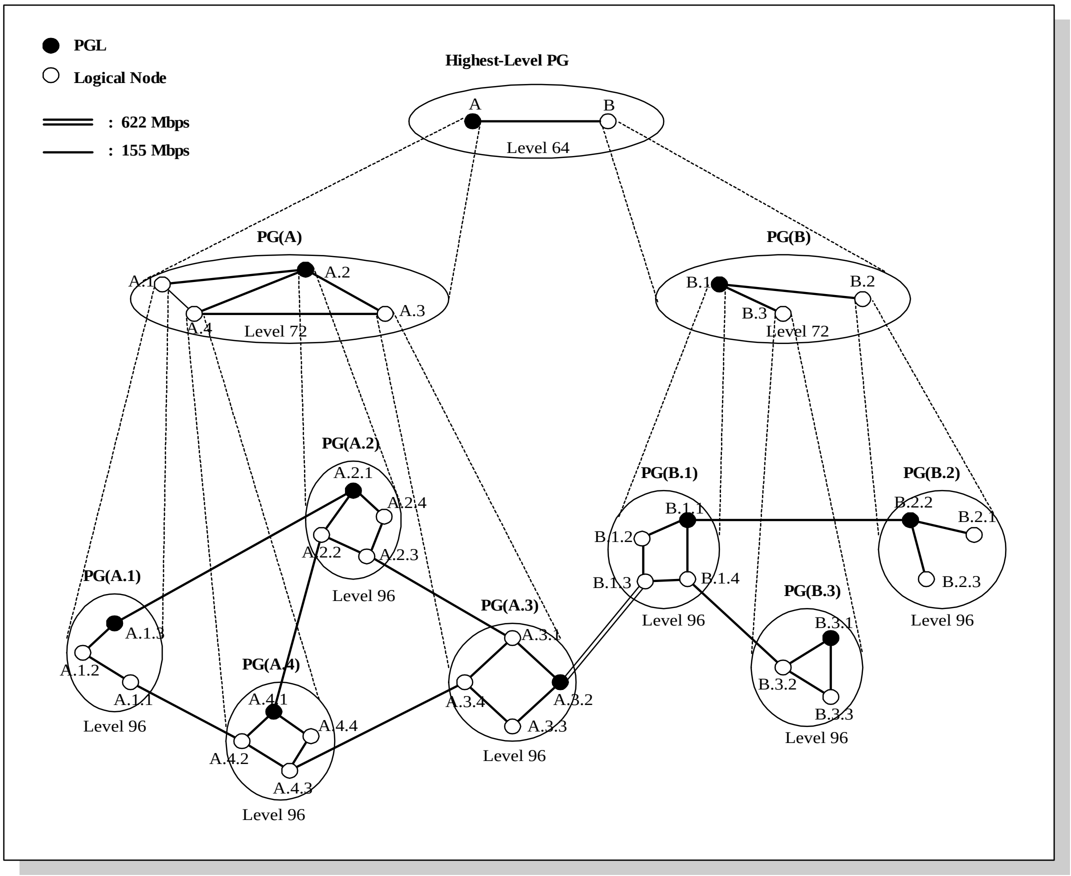

Figure 3.8 Network topology

call holding time. The initial 100 time units are estimated as the transient period and, thus, performance samples are discarded.

Figure 3.9 shows the fractional reward loss of calls under different arrival rates, when two aggregation schemes are used for the WP and the MDP and the COL link cost functions. The figure shows that the fractional reward loss increases as the arrival rate increases. Meanwhile, the fractional reward loss of the Asymmetric Simple scheme of the COL link cost is slightly higher than for the Full Mesh scheme. The fractional reward loss of Asymmetric Simple scheme of MDP link cost is almost the same as Full Mesh scheme. Meanwhile, the MDP approach yields a better performance than the COL in both the Asymmetric Simple and Full Mesh schemes. This superiority is because the MDP approach considers the arrival rate, and so can provide more accurate cost information and call admission function. However, the MDP and COL cost-based approaches significantly outperform the WP approach. The main reason is that the bandwidth-based approach only considers the maximum residual bandwidth

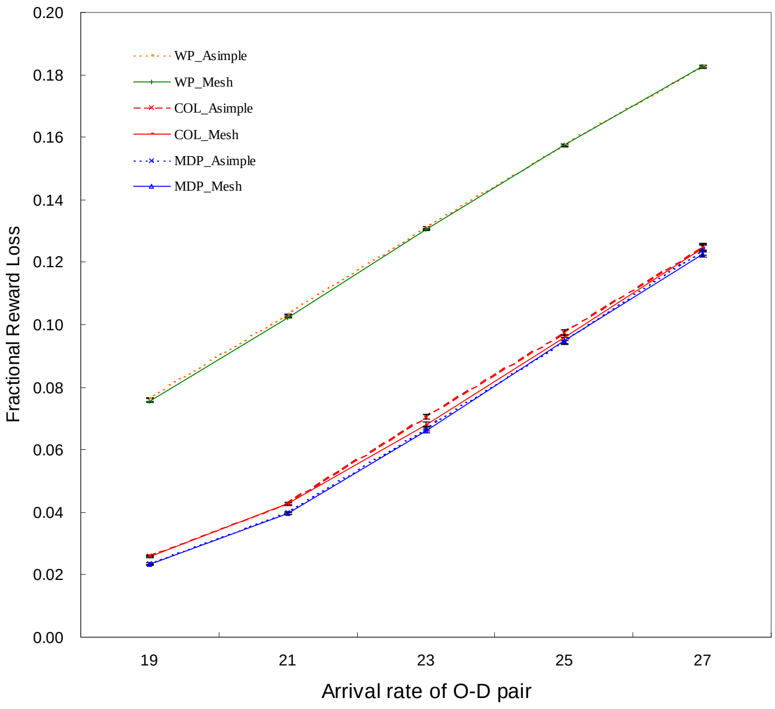

Figure 3.9 Fractional reward loss.

of bottleneck link among all candidate paths. However, the cost-based approaches first transfer the residual bandwidth of link into a reasonable cost and then route the minimum path cost.

Figure 3.10 shows the average number of crankbacks per connection request under different arrival rates, when two aggregation schemes are used. The figure shows that the average number of crankbacks increases as the arrival rate increases. The average number of crankbacks of the Asymmetric Simple of the COL link cost function is higher than for the Full Mesh scheme. Meanwhile, the average number of crankbacks of the Asymmetric Simple of MDP link cost function is almost the same as that of the Full Mesh scheme. The COL approach of all schemes requires less crankbacks than the MDP. The reasons for this phenomenon are that the cost function is less accurate and the call admission of COL approach is less conscientious than that of the MDP approach. Consequently, a call may be routed to non-optimal path that requires more network resources and thus increases call blocking probability. The call admission function rejects a call if the aggregated path cost exceeds the admission threshold, which also results in less paths (crankbacks) being tried and

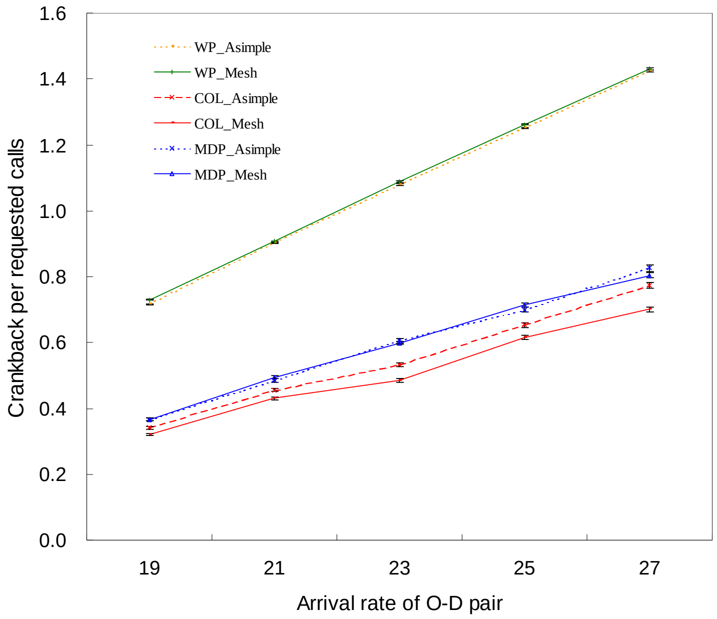

Figure 3.10 Crankback per requested calls.

higher call blocking probability. However, the MDP approach provides more accurate aggregated cost and precise call admission policy because the link cost is computed based on Markov decision theory. This approach results in less calls being blocked due to call admission at the hierarchical routing procedure, but more blocking due to the call admission at the physical network level. Hence, the MDP approach results in more crankbacks. Additionally, the number of crankbacks of the WP approach is more than that of the COL and MDP approaches. Clearly, the reason is that the WP approach is not a good approach for hierarchical routing.

Figure 3.11 shows the fractional reward loss yielded by the five update policies under different arrival rates. As figure 3.11 illustrates, the FU policy yields the lowest fractional reward loss, because it provides the most accurate routing updates. However, the LRBU, DCU, and DCUH policies all yield very competitive fractional reward loss as compared to the FU policy. Figure 3.11 shows that the DCUH policy yields slightly lower fractional reward loss than the LRBU and DCU policies. Meanwhile, figure 3.11 also illustrates that PNNIU policies yield worse performance than the other four policies. The performance of PNNIU

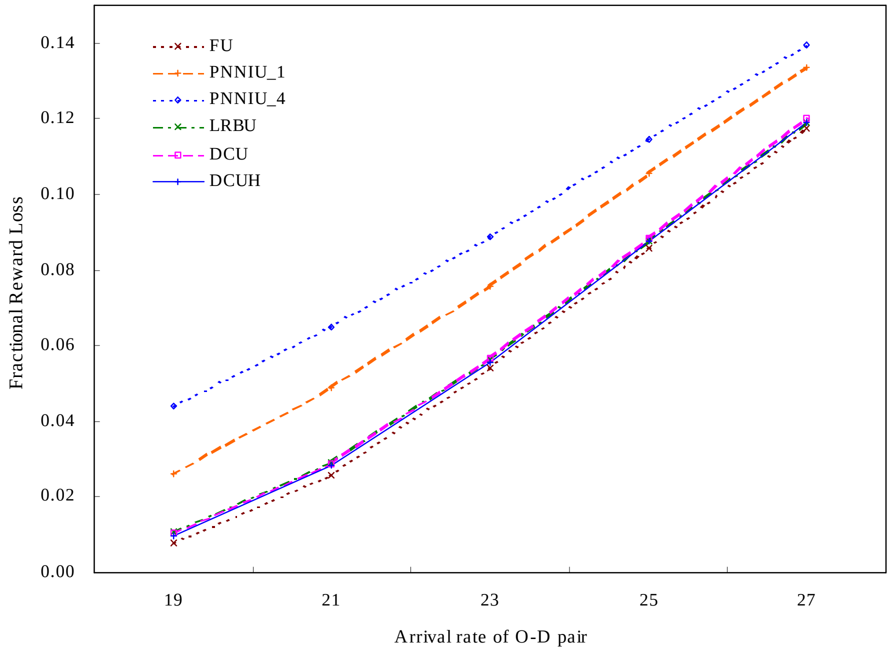

Figure 3.11 Fractional reward loss of five update policies.

policy can be improved if the update interval reduces. For example, the PNNIU_1 policy outperforms the PNNIU_4 policy.

Figure 3.12 shows the average number of crankbacks per connection request under different arrival rates. Intuitively, the more accurate the aggregated routing information, the less the average number of crankbacks. Hence, the average number of crankbacks per connection request is an important performance metric for update policies. Figure 3.12 illustrates that FU, LRBU, DCU, and DCUH have almost the same number of crankbacks per connection while the PNNIU policies have a much higher number of crankbacks.

Figures 3.13 and 3.14 show the number of information re-aggregations and distributions per unit of time under different arrival rates. Intuitively, the FU policy should have the highest re-aggregation frequency, and the two figures confirm this. On the other hand, adjusting the update interval gives the PNNU policy the least update frequency. Comparing the LRBU, DCU, and DCUH policies reveals that the DCU and DCUH policies yield less overhead re-aggregation than the LRBU policy. However, the DCU policy with hysteresis does not significantly reduce the update frequency. One possible reason for this phenomenon is that the

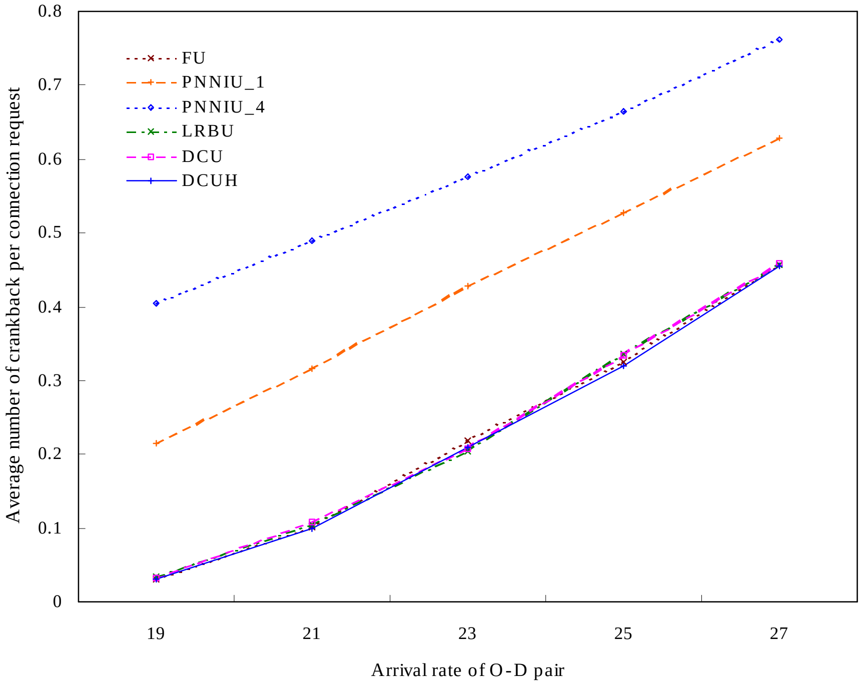

Figure 3.12 Average number of crankback per connection request.

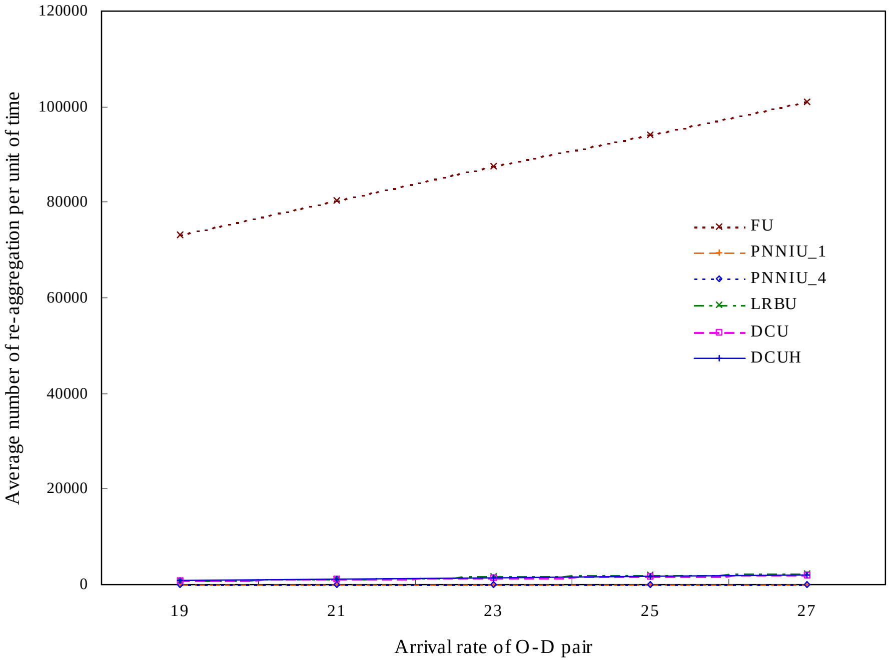

Figure 3.13 Average number of re-aggregation per unit of time.

DCU policy does not suffer much from oscillation. The effect of hysteresis requires further investigation.

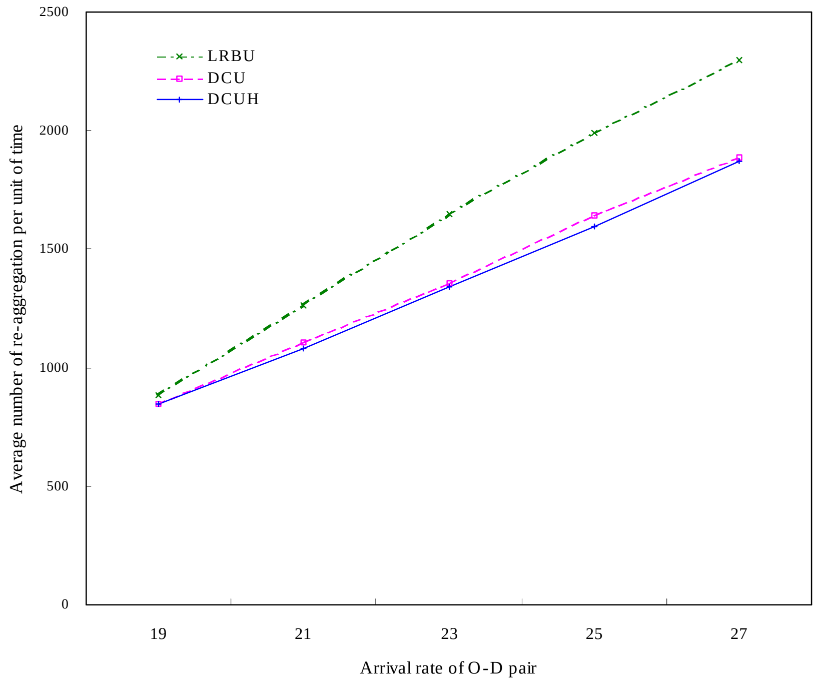

Figure 3.14 Average number of re-aggregation per unit of time yielded by DCU, DCUH, and LRBU policies.

The above figures reveal that the Asymmetric Simple and Full Mesh aggregation scheme of hierarchical routing perform very competitively under different simulation parameters. However, the complexity of advertised information of Asymmetric Simple is O(n) while that of Full Mesh is O(n^2). The MDP link cost function outperforms the COL link cost function due to more accurate cost information and more systematic call admission function. We also observe that FU, LRBU, DCU, and DCUH perform competitively in fractional reward loss and average number of crankbacks per connection request. However, the DCU and DCUH policies require the least re-aggregation frequency. Furthermore, the time-based PNNIU policies yield the highest fractional reward loss and number of crankbacks, but require significantly less update frequency. Meanwhile, the LRBU performs very competitively performance compared to the cost-based update policies, and thus can be considered a simple but efficient policy. Finally, DCU appears not to suffer from the oscillation problem, while the DCUH policy does not significantly reduce re-aggregation frequency.

Finally, the performance of PNNI routing with the proposed crankback schemes, CIS and CT, is evaluated via simulations and compared with two baseline crankback schemes, NAIVE and ABL. When crankback procedure is initiated, the NAIVE scheme tries to route the call setup message on alternate paths one by one. ABL(Avoids Block Link) is similar to NAIVE but will avoid routing on alternate paths with blocked link. Both schemes crank back the blocked setup message to the ingress node within a PG or the first PG of the same level. The blocked link information is required in the ABL scheme. Figure 3.15 shows the network topology for our simulations. The detail capacity of each link is shown in Table 3.11 and each link has 1 ms delay.

Table 3.1: Link capacity assignments

Link id	Cap.	Link id	Cap.	Link id	Cap.
e(A11,A12)	250	e(A42,A47)	400	e(B21,B22)	155
e(A11,A13)	250	e(A43,A44)	350	e(B21,B24)	155
e(A12,A21)	350	e(A43,A46)	350	e(B22,B23)	155
e(A13,A51)	350	e(A44,A45)	350	e(B22,B24)	155
e(A21,A22)	155	e(A46,A47)	400	e(B23,B24)	155
e(A21,A24)	155	e(A47,B41)	622	e(B23,B34)	350
e(A22,A23)	155	e(A51,A52)	155	e(B31,B32)	350
e(A22,A24)	155	e(A51,A56)	155	e(B32,B33)	155
e(A23,A34)	350	e(A52,A53)	155	e(B32,B36)	155
e(A23,A24)	155	e(A52,A54)	155	e(B33,B34)	350
e(A31,A32)	350	e(A53,A54)	155	e(B33,B35)	155
e(A31,A53)	350	e(A54,A56)	155	e(B35,B36)	155
e(A32,A33)	155	e(A55,A56)	350	e(B36,B37)	350
e(A32,A36)	155	e(B11,B12)	155	e(B37,B45)	350
e(A33,A34)	350	e(B11,B17)	155	e(B41,B42)	400
e(A33,A35)	155	e(B12,B13)	155	e(B41,B46)	400
e(A35,A36)	155	e(B12,B15)	155	e(B42,B43)	350
e(A36,A37)	350	e(B13,B14)	350	e(B42,B47)	350
e(A37,A45)	350	e(B14,B21)	350	e(B43,B44)	350
e(A41,A42)	400	e(B15,B17)	155	e(B44,B46)	400
e(A41,A55)	350	e(B16,B17)	350	e(B44,B47)	350
e(A42,A43)	350	e(B16,B31)	350	e(B45,B46)	400

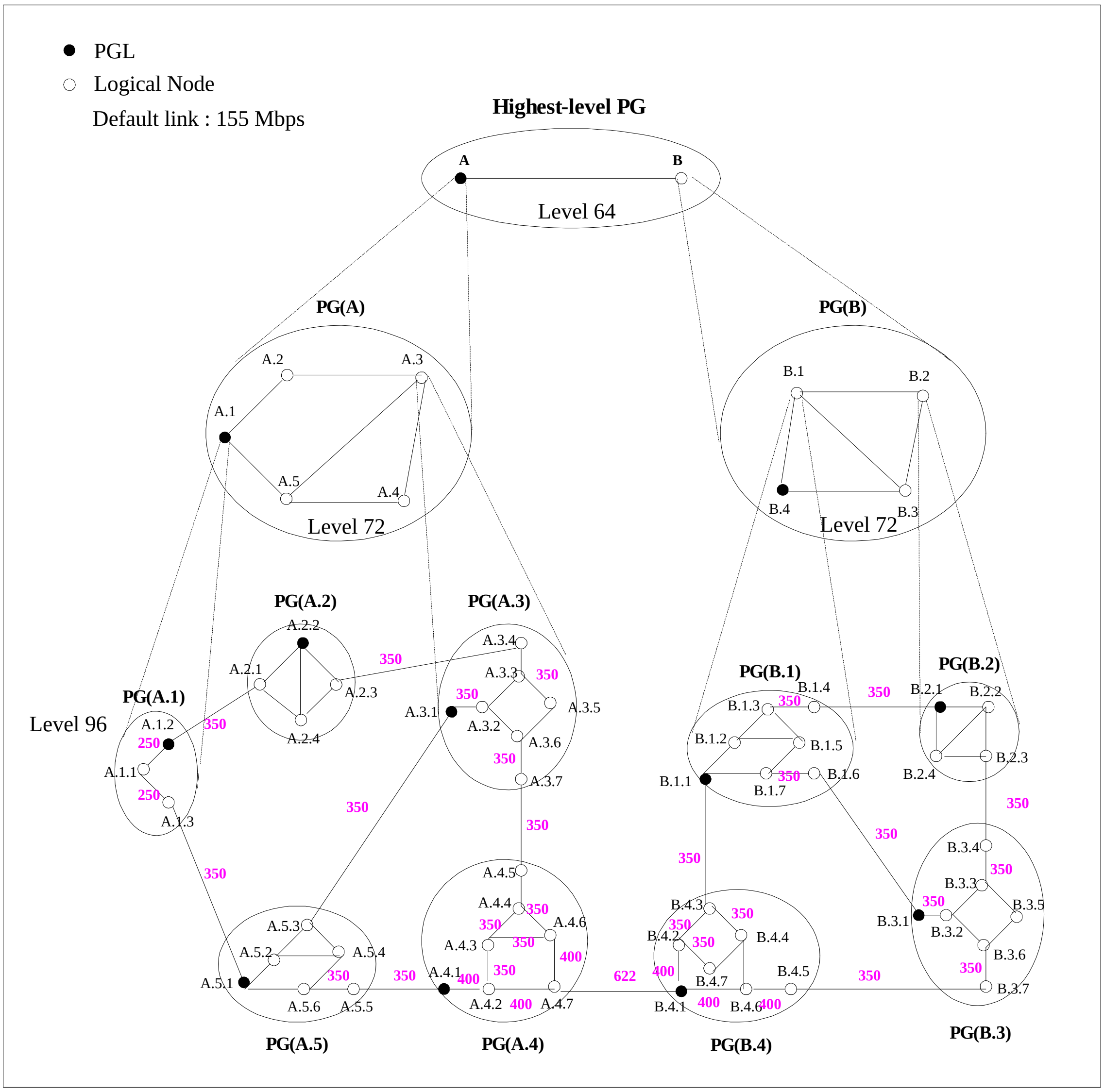

Figure 3.15 Network topology.

In this simulations, the network supports two classes of traffic. The class 1 traffic has a bandwidth requirement of b_1=1 Mbps, and an end-to-end delay of d_1=30 ms, with arrival rate $a_1^w = a$. The class 2 traffic has a bandwidth requirement of b_2=5 Mbps, and an end-to-end delay of d_2=30 ms, with arrival rate $a_2^w = a / b_2$. The average holding time of these two classes of traffic is normalized to unity. The PNNI protocol adopts time-based PTSE update policy, referred to as the PNNIU policy. We assume that the PTSE update interval is 0.2 times of mean call holding time. Each simulation result is observed over 10 independent runs. The length of each run is 1100 units of mean call holding time. For each run, the initial 10% of the samples were discarded. Crankback overhead is measured by the average number of nodes that a call setup message traversed during the crankback procedure.

Note that, NAIVE, ABL, and CIS, use the same call admission control policy. The difference of these schemes is where to crank back, referred as crankback destination node. Therefore, these three schemes have the same fractional reward loss. The CT scheme predicts that an alternate path will be blocked if its aggregated path cost is larger than a cost threshold. In our simulations, the threshold is set such that crankback overhead can be reduced as much as possible without increasing the fractional reward loss. According to our experiments, the best cost threshold parameter is set to $1.25 * D_{k,\min}$. Figure 3.16 shows that these schemes have the same fractional reward loss. The vertical lines about each point in the figure indicate 95 percent confidence interval. Figure 3.17 shows that both of CIS and CT schemes yield much lower crankback overhead than the NAIVE and ABL schemes. Furthermore, when we combine CIS with CT, referred to as the CIS_CT scheme, the crankback overhead can be reduced further. The reduction on crankback overhead becomes more significant as traffic load goes higher.

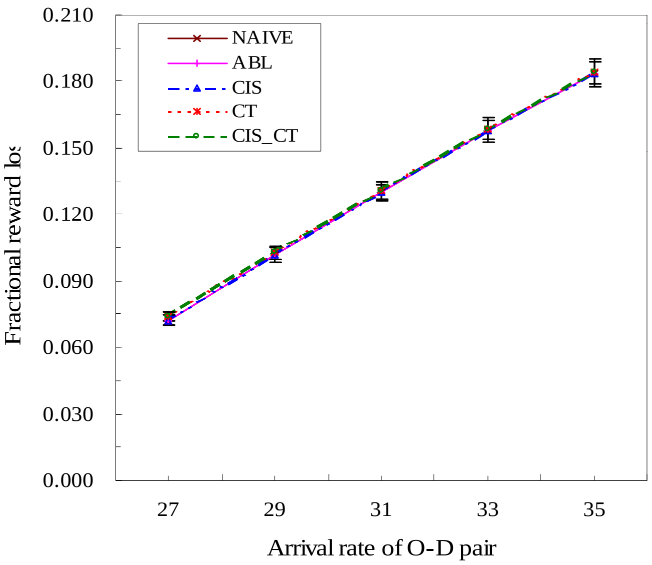

Figure 3.16 Fractional reward loss of different crankback schemes under PNNIU update policy.

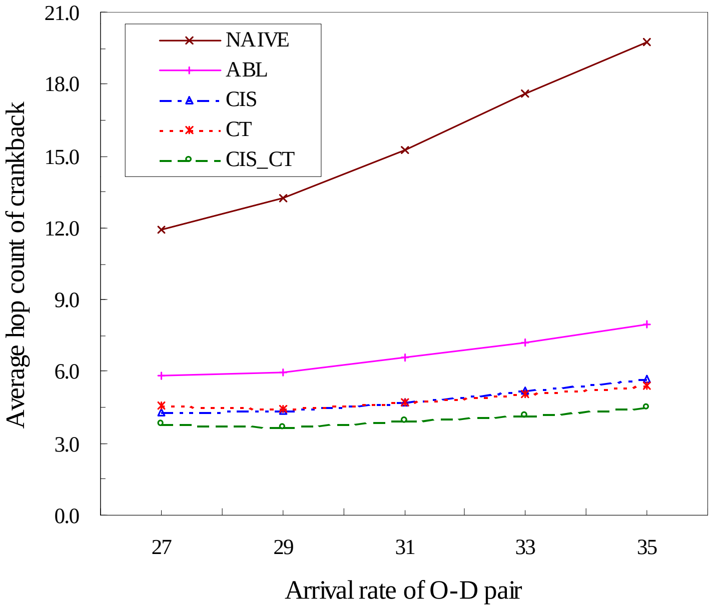

Figure 3.17 Average hops count of crankback of different crankback schemes under PNNIU update policy.

Chang et al. show that PTSE update interval will affect the accuracy of routing information. Therefore, in following simulations, we adopt LRBU (logarithm of residual bandwidth update), as the PTSE update policy. In LRBU, the bandwidth of link l, is divided into $\lfloor \log_2 Cap(l) \rfloor + 1$ states. For example, a link with capacity Cap(l)=8 has four states with residual bandwidth of $[0,1)$, $[1,2)$, $[2,4)$, $[4,8]$, respectively. The routing information will be re-aggregated and distributed when the link state changes. Figure 3.18 shows that CT yields the same fractional reward loss as compared to the NAIVE, ABL, and CIS schemes under the LRBU update policy. The CT scheme yields much lower crankback overhead than those schemes, as shown in figure 3.19. The crankback overhead yielded by the CT scheme is only half of that of the CIS scheme.

From the above figures, we observe that CIS and CT schemes reduce the crankback overhead significantly. Furthermore, combining CIS and CT schemes can further reduce the crankback overhead.

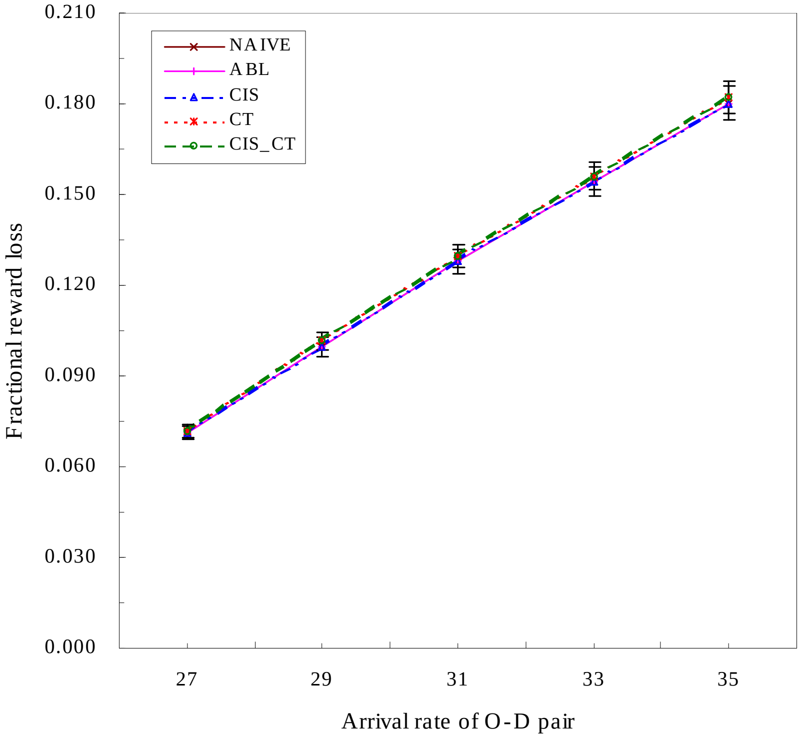

Figure 3.18 Fractional reward loss of different crankback schemes under LRBU update policy.

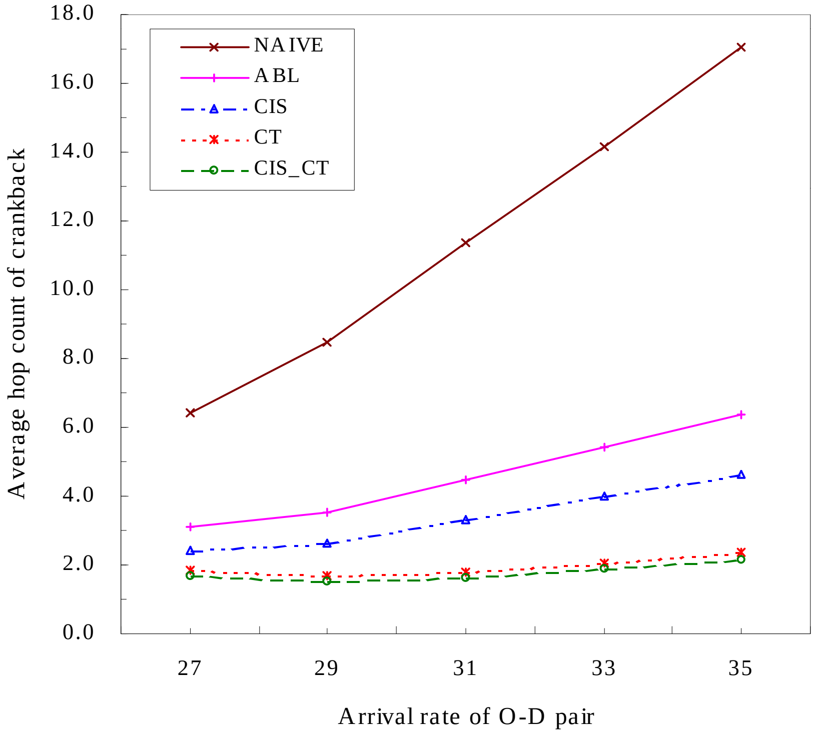

Figure 3.19 Average hops count of crankback of different crankback schemes under LRBU update policy.

（四）、整合服務與差別服務網路之相容運作技術

A. QOSPF with Overflowed Cache (PER-PAIR/OC)

According to Fig. 3.20, the overflowed-cache (OC) mechanism divides the forwarding cache into a per-destination cache (*D-cache*), a per-pair cache (*P-cache*) and an overflowed per-flow cache (*O-cache*). D-cache entries are looked up for the best effort packets, and the O-cache entry is created when a new flow arrives and cannot find sufficient bandwidth on the path of P-cache.

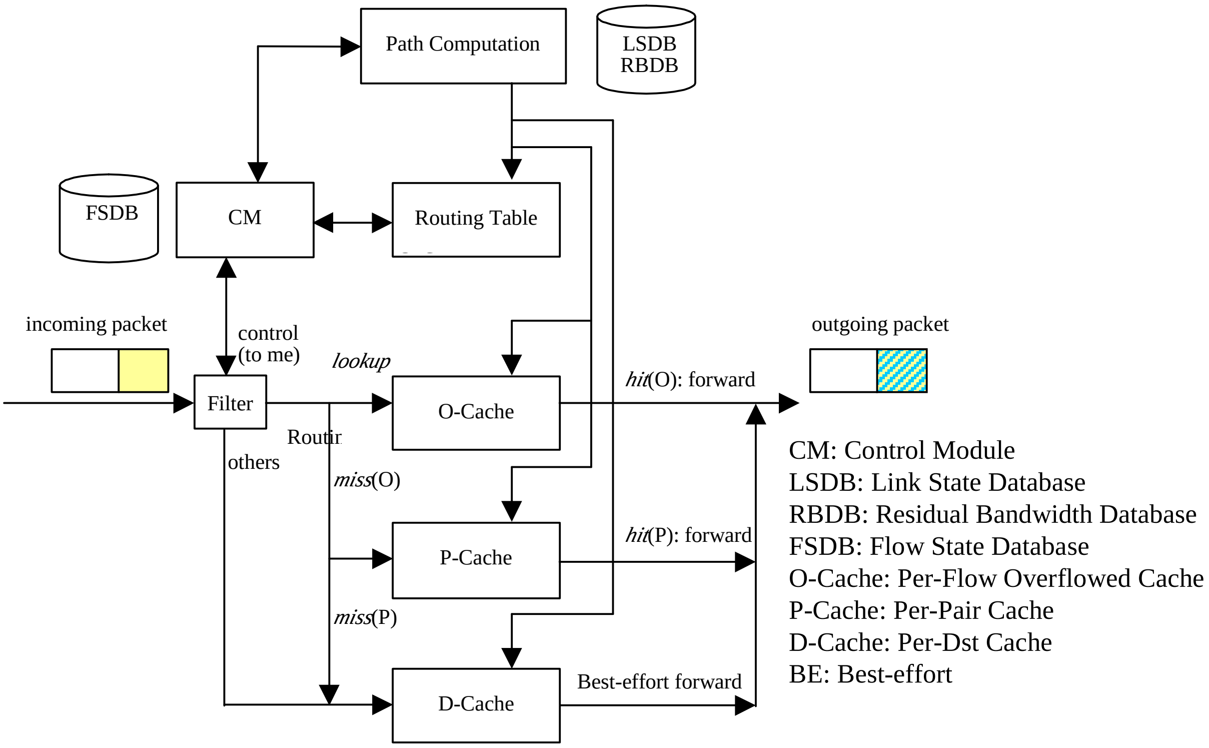

Figure 3.20 Overflowed Cache mechanism (Per-pair/OC)

B. QOSPF With Overflowed Cache With Two-Phase Routing (PER-PAIR/OCTP)

Intuitively, QoS flow locality exists between node pairs as in circuit-switched networks. This section extends OC to OCTP, the Overflowed Cache with Two-Phase routing, by replacing the function **FindRouteLeastCost** with the function **FindRoute_TP**. TP can work independently of OC, known as Per-pair/TP in this study. OCTP uses three types of caches, exactly the same as the OC scheme. However, a two-phase routing concept is used for finding the QoS path s with bandwidth requirement *b*. In phase I, referred to as *soft-reservation*, OCTP tries to find a path s_1 with more bandwidth than *b*, i.e. where *width*(s_1) ‡ b+b_{more}. Consequently, the subsequent incoming flows of the same S-D pair will be more likely to successfully reserve bandwidth on the path. Less misleading will increase the likelihood of success. If a soft QoS path s_1 cannot be found, OCTP will attempt to find a path s_2 with bandwidth *b*, i.e., *width*(s_2) ‡ *b*, referred to as *hard-reservation* because it takes the actual required bandwidth into account. This study refers to the database that reflects the link state of soft-reservation as soft-RBDB and that of actual reservation as hard-RBDB.

C. QOSPF Using Per-Class Routing Mark (PER-PAIR/PC)

When a new flow request with QoS requirement, Per-pair/PC first checks the forwarding cache (C-cache) if the number of sub-entries of the desired S-D pair, say |P(*s*, *d*)|, is zero. If yes, the C-cache will be missed and Per-pair/PC attempts to find the least costly feasible path, termed s. If s is found, Per-pair/PC assigns a new *mark* to s, inserts it into the cache, and will forward packets of the flow through s. The link cost function could be defined according to the need of network administrators. In this paper, we simply make the cost function the inverse of path width. If P(*s*, *d*) is full, Per-pair/PC simply finds the next hop p of the least costly feasible path among the existing |P(*s*, *d*)| paths, where p˛ P(*s*, *d*). If p is found, the algorithm marks the flow and forwards it to p, otherwise it blocks the flow. If P(*s*, *d*) is neither empty nor full, Per-pair/PC can either forward it to the p led by the cache, or route it through a newly computed path s, whichever costs less. Consequently, flows between an S-D pair may be routed on a maximum of *m* different paths where *m* is the maximum number of routing classes.

D. Performance Evaluation

D.1 Network Model and Traffic Model

Simulations are run on a 40-node random graph based on the Waxman's model. In our

TABLE 3.2: The Traffic Model Of The Simulation

Class	*Application*	*ratio*	*Bandwidth requirement*
GS_1	Video, e.g. H.260	20%	128Kbps
GS_2	Voice, e.g. I-	80%	16Kbps

TABLE 3.3: Cache Granularities Of The Simulation

Granularity	*Scheme*	*Feature*	*Path computation*
Per-destination	OSPF	Lookup next-hop by destination	Topology driven
Per-flow	QOSPF/G*	Route each individual flow	Flow driven
Per-pair	QOSPF/Z	Same route between a src-dst	Topology driven
	Per-pair/TP	Two-phase routing	Topology driven
Per-pair with overflowed cache	Per-pair/OC	Dual caches	Flow driven
	Per-pair/OCTP	Dual caches, two-phase routing	Flow driven
Per-pair/class	Per-pair/PC	Diff. route for diff. class between a src-dst	Topology driven

* QoS routing table indexed by (dst, hop_count).

* Only on-demand path-computation is used in our simulations

simulations assume that the token rate from *TSpec* of an RSVP PATH message is used as the bandwidth requirement of the flow. Furthermore, as Table 1 shows, this study assumes that there are two types of QoS traffic: GS_1 and GS_2. GS_1 models video sources where the bit rate is set to 128Kbps, for example videoconference, while GS_2 models voice source where the rate is set to 16Kbps.

D.2 Granularity and Performance Metrics

In our simulations, five different granularities of forwarding caches used in various QoS routing schemes are studied, as shown in Table 3.2. Seven performance metrics are interesting here: (1) Request blocking probability, P_{req}, (2) Cache misleading probability, P_{misl}, (3) Fractional reward loss, L_{rwd}, (4) Forwarding cache size, or N_{cache}, is the total storage overhead for a caching scheme. (5) Number of path computations, or N_{comp}, is the total number of path computations in the simulated network. (6) Path length distribution is the histogram of the path length of the admitted flows. (7) Fairness of traffic requirement.

TABLE 3.4: Summary Of The Simulation Results

Granularity	Mechanism	computation overhead	storage overhead	Blocking	misleading	fairness	path length
Per-dest	topology driven	low	low	high	high	poor	prefer short
Per-pair	topology driven	low	low	high	high	poor	prefer short
Per-flow	flow driven	very high*	very high	low**	no	fair	no discrimination
Per-pair /OC	flow driven	medium	medium	low**	no	fair	no discrimination
Per-pair /OCTP	flow driven	medium	medium	low**	no	fair	no discrimination
Per-pair /TP	topology driven	Low	low	medium	medium	medium	medium
Per-pair /PC	topology driven	medium	low	low	low	medium	medium

* scalable if pre-computation is used.

** except heavy loading.

D.3 Results

We has investigated the QoS routing extensions to the OSPF (QOSPF) and has proposed three mechanisms to achieve scalability with low blocking probability, overflowed cache (OC), two-phase routing (TP), and per-class routing mark (PC). OC divides the forwarding cache into a P-cache and an O-cache, and thus prevents the cache misleading effect. OC can be extended to OCTP with two-phase routing. Phase I soft-reserves more bandwidth for subsequent flows of the same S-D pair, while phase II hard-reserves actual bandwidth requirement if a flow is blocked in phase I. TP also can work independently of OC. PC aggregates the flows into several paths using routing marks, thus allowing packets to be fast forwarded in DiffServ core networks.

Extensive simulations using various routing and forwarding mechanisms found that per-destination routing has the worst blocking probability. This is because a coarser granularity is used, which reduces the accuracy of the network state. TP results in more flows running through their shortest paths than purely Per-pair. OC strengthens the path-finding ability as Per-flow scheme. OCTP combines the above two mechanisms and performs better than the alternatives. Note that, under heavy loading, the blocking probabilities performed by the flow driven mechanisms, including Per-flow, Per-pair/OC, and Per-pair/OCTP are as high as Per-dest and Per-pair. This is because many flows are traveling through longer paths which consume more resources per flow. Imposing a hop count limit *H*, where *H* can be either the value of network diameter or can be set explicitly, might solve the above unexpected behavior of flow driven mechanisms, which requires future studies. Additionally, Per-pair/PC has moderate blocking probability, fractional reward loss, with small forwarding cache. Per-pair/PC is suitable for the DiffServ networks because only 2 or 3 routing classes, i.e. marks, are needed.

Table 3.4 summarizes the simulation results. *Flow driven* mechanisms perform better in blocking probability, fairness, and state accuracy, while *topology driven* mechanisms result in less overheads. QoS routing and forwarding in a wire-speed core network, may use coarser granularity to achieve cheaper computation and storage cost, and forward packets faster.

Results presented herein can hopefully be applied to evaluate the overheads and performance of real network topologies. Moreover, we plan to extend the scalability issues studied in this paper to multicast QoS-based routing in IntServ and DiffServ networks.

（五）、寬頻網際網路之允入控制與訊務排程

A. The VQS System

VQS is assumed non-cut-through and non-preemptive. In other words, a packet is not served until its last bit has arrived, and once a packet is being served, no interruption is permitted until the whole packet is completely served. It is also work conserving in the sense that the server remains busy as long as there are packets in the queue. Packets are served under a normalized service rate of one cell/slot. Given a backlogged session, *i*, assigned with weight w_i, VQS allocates the session a minimum service rate of w_i / W (cells/slot), where $W = \sum_{i=1}^{N} w_i$ and N is the total number of sessions in the system. For ease of description, we assume the packet size is fixed (=*L* cells). The VQS algorithm, as will be shown, requires little modification for supporting variable-size packets.

For generalization, we consider two different VQS systems- a standalone VQS and an embedded VQS. While the former directly accepts input traffic from each session, the latter exerts a leaky-bucket regulator between each session's input traffic and VQS. First, the input traffic from each session is generally modeled by a discrete-time Switched Bernoulli Process (SBP). The process alternates between the High and the Low states. Second, the leaky-bucket regulator for session i is defined by ($^{\circ}_{i}, \pm_{i}$), where $^{\circ}_{i}$ (cells/slot) is the token generation rate and $\pm_{i}$ (cells) is the maximum token bucket size. Thus, under the embedded system, traffic from session i exhibits a mean arrival rate of $^{\circ}_{i} / L$ (packets/slot) and burstiness which increases with $\pm_{i}$.

B. Implementation Architecture

The architecture (see Figure 3.21) includes a VLSI chip, called the Sequencer, as a key component. The Sequencer is essentially a sorting-memory chip. By cascading multiple Sequencer chips in series or in parallel, a large linked list type packet pool could be implemented. As depicted in Figure 3.22, when a packet arrives, the packet is stored in the packet pool at the address provided by the Idle-Address FIFO, which contains the addresses of unused space in the packet pool. Before the packet is written into the packet pool, its session identifier is extracted and used as an index into the Session (S)-cache. The S-cache maintains a separate entry for each session, including the normalized weight and credit. Notice that since we assume WS=1 in this architecture, the sum of normalized weights of all sessions is equal to 1. The Session Controller is responsible for determining the index of the window in which this arriving packet can be placed, based on the normalized weight of the session to which the packet belongs.

C. Results and Merit Review of the Project

The performance of VQS is evaluated via simulation. Simulation results demonstrate that, applying a smaller WS for network elements with sufficient computation power, VQS performs as superior as the optimal scheme, WF^{2}Q, with respect to mean delay, throughput fairness, and worst-case delay fairness (see Figure 3.22). Moreover, compatible to WF^{2}Q, VQS outperforms WFQ with respect to 99-percentile delay bound and jitter in the presence of traffic burstiness. For network elements with limited power, VQS provides the best possible

QoS using a larger window size. The design and experimental results have been presented and demonstrated in various conferences and meetings, including IEEE ICC’00. Moreover, we have designed several networking control systems making use of the mechanism, which has been submitted to IEEE ICC 2001.

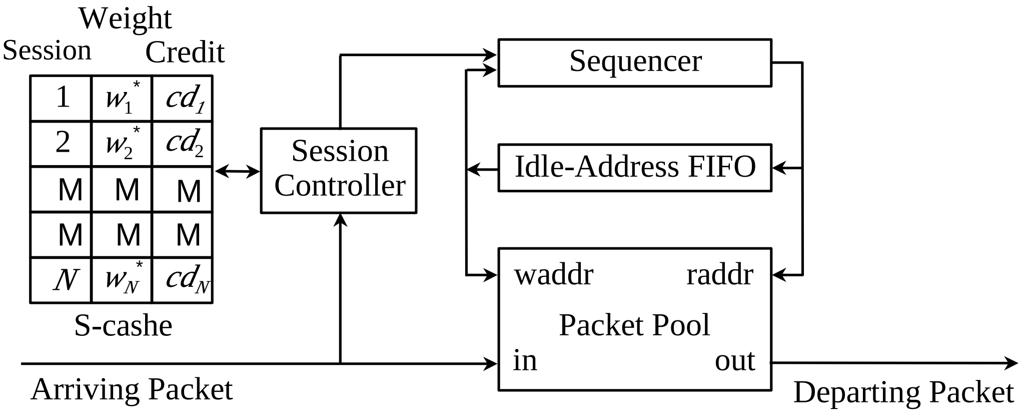

Figure 3.21 VQS implementation architecture.

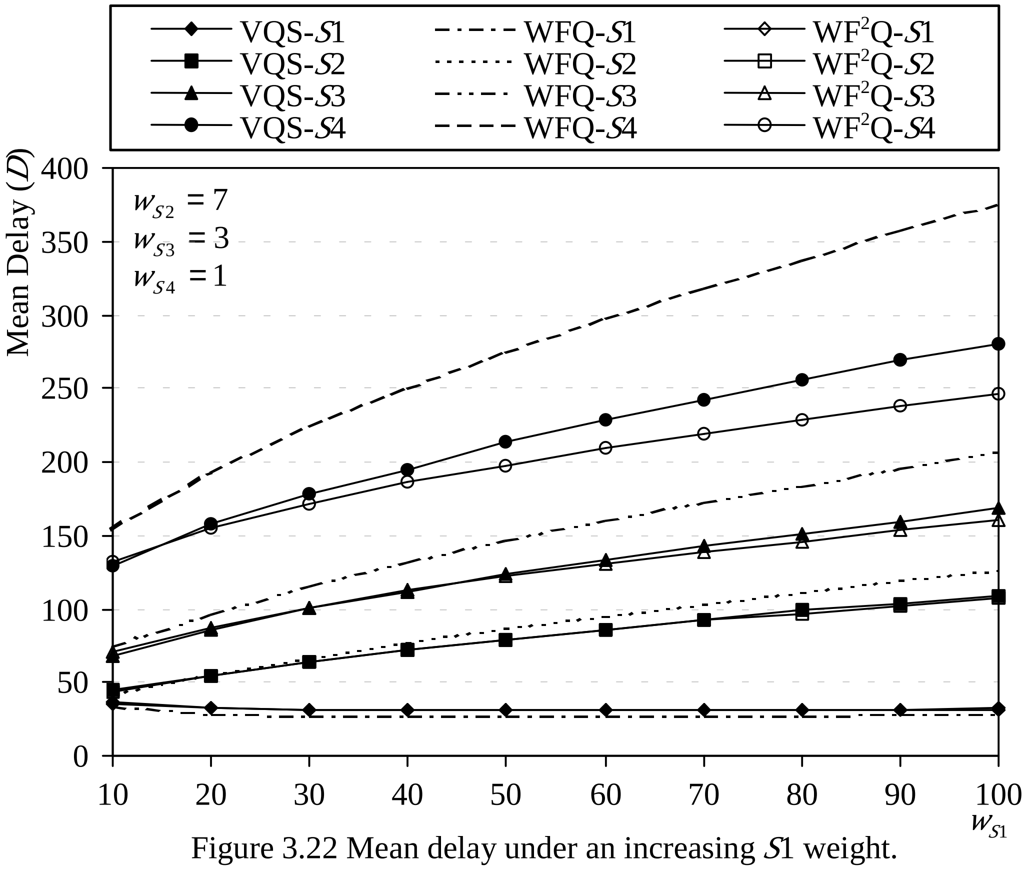

Figure 3.22 Mean delay under an increasing S1 weight.

第四章　結論

本年度計劃延續第一年的研究成果並深入研究與發展相關的關鍵技術；其中子計畫一（交通大學電信工程研究所李程輝教授主持）擴充交換機的功能，使之具有第三層硬體路由技術、第四層防火牆功能、支援群播功能、port trunking 以及八種優先權處理。交換機的架構主要採用 crossbar，並發展佇列管理和快速排程演算法來移除佇列前端擁塞問題，進而提昇路由器的傳輸率。特別是本交換機適用於可變長度封包的網路環境，以減小輸出埠閒置時間來提高效率降低封包延遲。並經由分析結果驗證固定長度封包設計的交換機並不適用於可變封包長度的網路環境。此外，此子計畫設計了一個三級式多欄位訊務區分器演算法，透過 hashing、search tree、linear search 三個步驟，可以將 filter 集合有效分散開來，減少搜尋所需要的運算量，以達成高速訊務區分的目的；並分析了此演算法的複雜度。

子計畫二（交通大學資訊工程研究所楊啟瑞教授主持）主要在研究服務排程技術與允入控制，以提供各種服務品質。寬頻網際網路上的 QoS (Quality-of-Service)排程必須提供有限延遲與公平性的考量且保持最小之計算複雜度。Prevailing weight-based 的排程原則提倡應用 multiple queues 和 engage 於 timestamp 的計算。這些原則在高複雜度的花費下，可以達到最佳之 QoS 效能，也可以為了計算簡化，達到次級之效能。本年度的計畫設計了一個 weight-based Versatile QoS Scheduler (VQS) 和其可行之 VLSI 硬體實作架構。為了促進較適當之效能與複雜度間的平衡，VQS 可以在寬頻網際網路內用不同之網路元件來實作。VQS 利用較簡單之 single-queue 與不需 timestamp 計算之優點，以 weight 來控制封包插入於一分享式之資料結構。此資料結構包含了一系列固定大小之 windows。在一給定的 window 內，由一個 session 來的最大封包個數，是和此 session 的 weight 和 window 大小成正比。實驗結果證明，應用較小之 window 於 high-power 的網路元件，VQS 可以在 throughput fairness、mean delay、與 worst-case delay fairness 等項目上表現和 WF^2Q 一樣好。甚至在和 WF^2Q 相容與具備 traffic burstiness 的情況下，VQS 較 WFQ 優越約 99%之 delay bound 和 jitter。

子計畫三(台灣大學資訊管理研究所孫雅麗教授主持)主要研究如何透過量測訊務來提升網路使用效率及避免擁塞發生；近來許多網路上的即時媒體應用對於傳輸服務品質的要求較為寬鬆，針對不需嚴格封包延遲、但需避免封包遺失的應用，有必要提供可保障封包遺失率的無壅塞傳輸服務。於此子計劃，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，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，但若要更精確地控制封包遺失率，則需要進一步的研究。

子計畫四（中正大學資訊工程研究所黃仁竑副教授主持）主要在研究如何選定 QoS 的 metrics 與 routing algorithm，以及搭配 RSVP 時，若資源保留要求失敗時，如何 re-routing；此子計劃提出四項方法以實現一更高效率、正確資訊、擴展性與服務品質保證的 PNNI 選徑機制。第一、提出一高效率且低複雜度的聚集方法，稱之為 Asymmetric Simple 聚集法；第二、提出以馬可夫理論(MDP)為基礎的鏈路成本定義機制，並將此法與線上競爭法(COL)和最寬頻寬法(WP)作效能比較；第三、提出以成本為機礎的動態資訊更新政策(DCU), 以提高資訊正確性的同時減少資訊更新頻率, 降低了選徑的複雜度；最後，以 CIS 與 CT 方法減少 crankback overhead。結果顯示 Asymmetric Simple 聚集法可有效提供良好的網路使用率，同時減低網域劇集的複雜度。MDP 方法提供系統化的 call admission 機制且考慮 call 到達率，因此與 COL 和 WP 方法相比，MDP 具有良好的網路使用率與高效率。再者，DCU 方法適用於動態網路流量，因此可得良好網路使用率同時明顯減少網路資訊更新頻率。而所提出的 CIS 與 CT 方法明顯降低 crankback overhead，尤其以結合兩者時更為顯著。

子計畫五(交通大學資訊科學研究所林盈達副教授主持)主要探討路由決策的粗細度對服務品質路由的延展性及阻絕率的影響；並且提出三個機制—溢流快取、兩階段路由、與路由記號分類轉送技術，目的在限速封包網路上達到計算上級儲存上的延展性，並降低阻絕率。在各種粗細度的系統模擬中，結果顯示所提機制確有不錯的效果，尤其路由記憶分類轉送技術使用二或三種分類很適合差別服務的核心網路中。總結比較，flow driven 的路由技術如 per-flow 與 per-pair/overflowed-cache 會有較佳的阻絕率、公平

性、與狀態準確性，data driven **的路由技術如** per-destination、per-pair、per-pair/two-phase **與** per-pair/class **會有較少的** overhead。

參考文獻

[1] ATM Forum Technical Committee, “PNNI Specification Ver 1.0,” March, 1996.

[2] A. Iwata, H. Suzuki, R. Izmailov, and B. Sengupta, “QOS Aggregation Algorithms in Hierarchical ATM Networks,” *IEEE ICC’98*, 1998.

[3] B. Awerbuch, Y. Du, B. Khan, and Y. Shavit, “Routing Through Networks with Hierarchical Topology Aggregation,” *IEEE Symposium on Computer and Communication*, pp. 406-412, 1998.

[4] B. Awerbuch, Y. Du, and Y. Shavit, “The Effect of Network Hierarchy Structure on Performance of ATM PNNI Hierarchical Routing,” *IEEE Computer Communications and Networks*, pp. 406-412, 1998.

[5] Ben-Jye Chang, Hsien-Kang Chung, and Ren-Hung Hwang, “Hierarchical QoS Routing in ATM Networks,” *The 14th International Conference on Information Networking*, Jan. 2000.

[6] M. Montgomery and G. D. Veciana, “Hierarchical Source Routing Through Clouds,” *IEEE INFOCOM’98*, 1998.

[7] F. Hao, E. W. Zegura, and S. Bhatt, “Performance of the PNNI Protocol in Large Networks,” *IEEE ATM Workshop*, pp. 315-323, 1998.

[8] Ren-Hung Hwang and Youn-Chen Sun, “Adaptive Multicast Routing in Broadband Networks,” *SPIE Conference on Performance and Control of Network Systems*, Nov. 1998.

[9] R. A. Howard, Dynamic programming and Markov processes. John Wiley & Sons, Inc., 1960.

[10] Ren-Hung Hwang, James F. Kurose, and Don Towsley, “State Dependent Routing for Multirate Loss Network,” *Globecom’92*, pp. 565-570, 1992.

[11] Ren-Hung Hwang, May 1993,“Routing in High-Speed Networks”, Ph. D. Thesis, also available from Computer Science Dept. Technical Report 93-43, Univ. of Mass.

[12] Borodin Allan and Ran El-Yaniv, “Online computation and competitive analysis,” Cambridge University Press, 1998.

[13] Ren-Hung Hwang and Youn-Chen Sun, “Effect of Link Cost Function and Call Admission Control on Multicast Routing in Broadband Networks,” submit for publication.

[14] Ben-Jye Chang and Ren-Hung Hwang, “Hierarchical QoS Routing in ATM Networks Based on MDP Cost Function,” to appear in *IEEE ICON 2000.*

[15] B. Awerbuch, Y. Du, B. Khan, Y. Shavit, “Routing Through Networks with Hierarchical Topology Aggregation,” *IEEE Symposium on Computer and Communication*, pp. 406-412, 1998.

[16] Shun-Ping Chung and Jin-Chang Lee, “Dynamic Reservation with Hysteresis in Cellular Multiservice Networks,” *The 14th International Conference on Information Networking*, Jan. 2000.

[17] Ben-Jye Chang and Ren-Hung Hwang, “Dynamic Update of Aggregated Information for Hierarchical QoS Routing in ATM Networks,” submitted to *IEEE GLOBECOM 2000.*

[18] Jae-Yeul Chung, “A Predictive Alternate Path Routing Approach Supporting the Best QOS in ATM Networks”, *ICCT’98*, pp. 22-24, October 1998.

[19] Felstaine, E., Cohen, R., and Hadar, O., “Crankback Prediction in Hierarchical ATM Networks”, *INFOCOM '99*, pp. 671–679, 1999.

[20] J. Moy, “OSPF Version 2,” *RFC 2328*, April 1998.

[21] E. Crawley, R. Nair, B. Rajagopalan, H. Sandick, “A Framework for QoS-based Routing in the Internet,” *RFC 2386*, August 1998.

[22] R. Guerin, A. Orda and D. Williams, “QoS Routing Mechanisms and OSPF extensions,” *Internet Draft*, draft-guerin-qos-routing-ospf-03.txt, January 1998.

[23] G. Apostolopoulos, D. Williams, S. Kamat, R. Guerin, A. Orda and T. Przygienda, “QoS Routing Mechanisms and OSPF extensions,” *RFC 2676*, August 1999.

[24] G. Apostolopoulos, R. Guerin, S. Kamat, and S. Tripathi, “Quality of Service Based Routing: A Performance Perspective,” *SIGCOMM'98*, September 1998.

[25] G. Apostolopoulos, S. Kamat, and R. Guerin, “Implementation and Performance Measurements of QoS Routing Extensions to OSPF,” *INFOCOM'99*, July 1999.

[26] Z. Zhang, C. Sanchez, B. Salkewicz, and E. Crawley, “QoS Extensions to OSPF,” *Internet Draft*, draft-zhang-qos-ospf-01.txt, September 1997.

[27] D. Black, S. Blake, M. Carlson, E. Davies, Z. Wang, and W. Weiss, “An Architecture for Differentiated Services,” *RFC 2475,* December 1998.

[28] R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin, “Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Spec.,” *RFC 2205*, September 1997.

[29] P. Chemouil, J. Filipiak, and P. Gauthier, “Analysis and Control of Traffic Routing in Circuit-Switched Networks,” *Computer Networks and ISDN Systems*, 11, pp. 203-217, 1986.

[30] B. M. Waxman, “Routing of Multipoint Connections,” *IEEE JSAC*, Vol. 6, No. 9, pp. 1617-1622, December 1988.

[31] McKeown, N. “The iSLIP Scheduling Algorithm for Input-queued Switches,” in *IEEE/ACM Transactions on Networkin*g, Vol. 7 , April 1999.

[32] Karol, M.; Hluchyj, M.; and Morgan, S. “Input versus Output Queueing on a Space Division Switch,” in *IEEE Transactions of Communications*, vol. 35, pp. 1347-1356, 1987.

[33] V. Srinivasan, S. Suri, and G. Varghese. “Packet Classification Using Tuple Space Search,” in *Proceedings of ACM SIGCOMM’99*, Sept. 1999.

[34] V. Srinivasan, G. Varghese, S. Suri, and M. Waldvogel. “Fast and Scalable Layer Four Switching,” in *Proceedings of ACM SIGCOMM’ 99*, Sept. 1999.

[35] P. Gupta and N. Mckeown. “Packet Classification on Multiple Fields,” in *Proceedings of ACM SIGCOMM’99*, Sept. 1999.

[36] Thomas Y.C. Woo. “A Modular Approach to Packet Classification Algorithms and Results,” in *Proceedings of INFOCOM 2000*.

[37] Wei-Che Chen. “Design and Implemen- tation of Traffic Classification for Layer3 Switch,” July, 1999.

[38] M.Buddhikot, S. Suri, and M. Waldvogel. “Space Decomposition Techniques for Fast Layer-4 Switching,” in *Proceedings of IFIP Workshop on Protocols for High Speed Networks*, Salem, Massachusetts, August 1999.

[39] S. T. Chuang, A. Goel, N. McKeown, and B. Prabhakar, “Matching Output Queueing witch a Combined Input Output Queued Switch,” *Technical Report*, CSL-TR-98-758, April 1998.

[40] W. Doeringer, G. Karjoth, and M.Nassehi, “Routing on Longest Matching Prefixes,” in IEEE/ACM Transactions on Networking, vol. 4, Feb. 1996.

[41] D. C. Stephens and H. Zhang, “Exact Emulation of an Output Queueing Switch by a Combined Input Output Queueing Switch,” in *Proceedings of IWQOS’98*.

[42] P. Krishna, N. Patel, A. Charney, and R. Simcoe, “On the Speedup Required for Work-Conserving Crossbar Switches,” in *Proceedings of IWQOS'98*.

[43] N. McKeown, “Scheduling Algorithms for Input-queued Cell Switches,” *Ph.D. dissertation*, Univ. California, Berkeley, May 1995.

[44] A. Charny, P. Krishna, N. Patel, and R.simcoe, “Algorithms for Providing Bandwidth and Delay Guarantees in Input-Buffered Crossbar with Speedup,” in *Proceedings of IWQOS'98*.

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

[46] 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.

[47] D. D. Clark, S. Shenker, L. Zhang, “Supporting Real-Time Applications in an Integrated Services Packet Network: Architecture and Mechanism”, *SIGCOMM’92*, 1992.

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

[49] R. Guerin, H. Ahmadi, 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.

[50] S. Jamin, P. B. Danzig, S. J. Shenker, 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.

[51] S. Shenker, C. Partridge, R. Guerin, “Specification of Guaranteed Quality of Service”, *IETF RFC2212*, Sep. 1997.

[52] J. Wroclawski, “The Use of RSVP with IETF Integrated Services”, *IETF RFC 2210,* Sep. 1997.

[53] J. Liebeherr, D. Wrege, and D. Ferrari, “Exact Admission Control for Networks with a Bounded Delay Service,” *IEEE/ACM Trans. on Networking*, vol. 4, no. 6, Dec. 1996, pp. 885-901.

[54] D. Lee, and B. Sengupta, “Queueing Analysis of a Threshold Based Priority Scheme For ATM Networks,” *IEEE/ACM Trans. on Networking*, vol. 1, no. 6, Dec. 1993, pp. 709-717.

[55] J. Hah, and M. Yuang, “A Delay and Loss Versatile Scheduling Discipline in ATM Switches,” *Proc. IEEE INFOCOM*, 1998, pp. 939-946.

[56] A. Demers, S. Keshav, and S. Shenker, “Analysis and Simulation of a Fair Queueing Algorithm,” *Proc. SIGCOMM*, 1989.

[57] J. Bennett, and H. Zhang, “WF^{2}Q: Worst-case Fair Weighted Fair Queueing,” *IEEE INFOCOM*, 1996, pp. 120-128.

[58] S. Golestani, “A Self-Clocked Fair Queueing Scheme for Broadband Applications,” *IEEE INFOCOM*, 1994, pp. 636-646.

[59] D. Stiliadis, and A. Varma, “Efficient Fair Queueing Algorithms for Packet-Switched Networks,” *IEEE JSAC*, vol. 6, no. 2, April 1998, pp. 175-185.

[60] R. Cruz, “A Calculus for Network Delay, Part I: Network Elements in Isolation,” *IEEE Trans. on Information Theory* , vol. 37, no. 1, Jan. 1991, pp. 114-131.

[61] H. Saito, *Teletraffic Technologies in ATM Networks*, Artech House, 1994.

[62] H. Chao, and N. Uzun, “An ATM Queue Manager Handling Multiple Delay and Loss Priorities,” *IEEE/ACM Trans. on Networking*, vol. 13, no. 6, Dec. 1995, pp. 652-659.