

九十二年度電信國家型計畫執行成果

總計畫名稱：ALL-IP 網路上以預算為基礎之品質管理研究

子計畫名稱：ALL-IP 核心網路品質管理研究

國立政治大學資訊科學系

連耀南、陳宗勳

預算法全 IP 核心網路服務品質管理之路徑規劃

摘要

面對通訊與資訊科技的大幅進步、網際網路的蓬勃發展、以及電信自由化帶來的激烈競爭，通訊網路正在進行一個巨大的變革，企圖將原有 Circuit Switching 與 Packet Switching 網路整合成一個單一整合型網路 - All-IP 網路以支援所有的應用服務。All-IP 網路受限於封包交換網路原有的特性，有服務品質問題 (QoS) 有待克服，因此有必要在 All-IP 網路上提供服務品質管理機制以實現整合型網路的目標。而要提供適當的 QoS 管理，其成功之關鍵主要在於是否能提供一個簡單易行之架構。本論文先提出 BBQ(Budget-Based QoS)採用以預算為基礎之服務品質管理，BBQ 提供一個高適用性的管理架構和相關的管理工具，可適用於不同的下層網路架構和不同營運目標的網管政策。

本文中提出以路徑規劃(Path Planning)的方式作為此架構的路由方法。路徑規劃係以過去的歷史訊務資料為基礎，為未來可能進入網路中的訊務規劃出一組具有品質要求的路徑，而在系統運作時，只需根據訊務的需求，指定一條符合其品質要求的路徑即可。此方法不但可以給予進入網路的訊務絕對的服務品質保證，而且事前的運算可以免除在系統運作時大量的計算負擔。在本論文中，我們設計了一套路徑規劃的演算法，來驗證我們的方法之效能。而經過反覆的模擬測試，發現本路徑規劃方法的確較傳統 OSPF 演算法擁有較佳的網路效能。

Path Planning in Budget-Based QoS Management for All-IP Core Networks

Abstract

In response to the great progress of communications and computer technologies, aggressive deployment of broadband fiber optical network, advance of Internet technology, and the global standardization of IP technology, the telecommunication industry is moving toward a converged network, which uses a single global IP based packet-switching network to carry all types of network services. Diverse types of services demand diverse QoS requirements making it a great challenge to support potential services with guaranteed QoS on All-IP networks.

Our research group proposes a Budget-Based QoS (BBQ) management architecture to facilitate network operators of diversified networks. With BBQ management architecture, network operators can adjust their network architectures and management policies to support as many services as possible with end-to-end QoS guarantee. To reduce real-time resource reservation and computation overhead, BBQ utilizes preplanning approach to allocate resources and to plan paths (routing) for future incoming traffics.

In this thesis, based on BBQ QoS architecture, we propose a path planning methodology that integrates both centralized and distributed processes. We also developed a heuristic algorithm to solve the path planning problem (routing) under BBQ architecture.

To admit a service request during execution time, admission control component will assign a planned path that satisfy the QoS requirements to the admitted service. Under this method, absolute QoS will be guaranteed for admitted traffic.

The heuristic path-planning algorithm we designed, the GPPA algorithm, takes total profit as its maximization objective and with bandwidth and quality entropy as its constraints. Through a performance evaluation using simulation method, we demonstrate that the pre-planning approach with our GPPA path-planning algorithm might outperform the real-time resource allocation approach with the traditional OSPF routing algorithm.

內容目錄

第一章	6
簡介 (Introduction)	6
1.1 研究動機及目的 (Motivation and Research Objective)	6
1.2 解決方案 (Solution Approaches)	8
1.3 論文組織結構	8
第二章	10
相關研究 (Related Work)	10
2.1 繞徑問題相關研究	10
2.2 評論 (Summary)	12
第三章	14
以預算為基礎之服務品質保證 (Budget-Based QoS)	14
3.1 BBQ 中的核心網路架構與 QoS 元件 (Core Network Architecture and QoS Components for BBQ)	14
第四章	26
核心網路之路徑規劃	26
4.1 BBQ 架構中各層級之路徑 (Path Definition)	27
4.2 在 BBQ 架構中的路徑規劃元件 (Path Planning Agent in BBQ Architecture)	28
4.3 路徑規劃的環境假設	30
4.4 端對端規劃路徑元件與核心網路內路徑規劃元件之互動	30
4.5 核心網路內路徑規劃運作流程 (Path Planning Procedure)	31
4.6 路徑規劃最佳化	34
4.7 最佳化模型 (System Model)	35
4.8 Greedy Algorithm for Path Planning(G.P.P.A.)	37
4.9 小結(Summary)	41
第五章	42
效能評估 (Performance Evaluation)	42

5.1	效能評估指標 (Performance Evaluation Metrics).....	42
5.2	實驗設計 (Design of Experiments).....	43
5.3	模擬過程及結果分析.....	48
5.4	小結.....	60
第六章		62
總結 (Concluding Remarks).....		62
參考文獻 (References).....		63

第一章

簡介 (Introduction)

整合型 All-IP 網路[1]將以一個單一傳輸平台提供固定網路及行動網路上所有服務，包括語音、多媒體、資料等各類服務[2-5]。此種革命性網路不但可以降低建置成本與營運管理成本，更可以提供一個新的服務平台，供營運者建置跨網路的應用服務。但欲達到整合型網路的理想之前，我們仍須克服許多困難，其中最關鍵的問題之一即是品質問題[6]。All-IP 網路受限於 packet switching 原有的特性，有三大品質問題有待克服：long delay time, jitter 以及 packet loss。這些品質問題對某些諸如語音或多媒體等應用服務有關鍵性的影響。此外，由於未來的網路係由特性差異極大的異質性網路所組成，而欲在此種網路上支援品質要求差異極大的多樣應用服務，其品質管理變得異常複雜，難以引用現有的品質管理方案。本論文旨在探討整合型 All-IP 網路之品質管理各項問題，並提出適當的管理機制。

我們提出 Budget-Based QoS 架構，以簡化管理、追求效率、不增加管理複雜度為原則，利用預算分配和預先資源分配規劃，來達成 End-to-End 的網路服務品質保證。根據此簡化管理原則，我們提供一套服務品質的管理工具，採用分層分權的方式將 QoS 管理權責以預算的方式分散至每個網路元件，如此可以避免繁複的折衝協調和即興式(real time on demand)的資源管理。此套管理工具，可供網路營運業者依其需要調整，在有限資源下追求使用者之整體最大滿意度。

在本文中提出一個路徑規劃問題的最佳化模型，藉以讓網路營運業者也已獲取最大的利潤，在分析問題後，並提出一個 Heuristic 方法來解決路徑規劃的問題，並以模擬的方式來驗證所提的演算法之效能。

1.1 研究動機及目的 (Motivation and Research Objective)

對於使用者而言，真正獲得的服務品質保證，不是由單一網路所提供。而是由端對端所經過之所有網路共同提供。所以，使用者關心的服務品質保證，是端對端服務

品質保證，而非單一網路之服務品質保證。因此，在 All-IP 網路中，如果僅提供單一網路之服務品質保證將無法滿足使用者之需求。通訊網路事實上是由全球大大小小的電信公司所轄網路藉由彼此之間的網路互連協定連接而成一個四通八達無遠弗屆的通訊網，一個長途通訊需求(request/traffic flow/phone call) 可能必須橫跨數個不同營運者的網路，其端對端品質管理是一大技術挑戰。

由於 DiffServ 並沒有辦法提供每個資料流端對端的品質保證，也不對個別的資料流做品質管理，所以在核心網路的架構中，必須需要有另外的機制負責找出每個資料流適當的傳送路徑，並配置頻寬來滿足不同資料流的品質需求。

一個簡單的機制，是由一個中央元件負責繞徑並分配資源，但集中式的作法卻有許多缺點：

- 中央統籌元件附近的節點會因大量的資源配置與路徑規劃訊息交遞而造成擁塞。
- 中央統籌元件必須記錄所有網域上 traffic flow 的資源使用資訊，overhead 過大。
- 中央統籌元件如遇當機，會使得網路癱瘓，風險過高。

所以即時的由中央統籌元件管理個別資料流會造成不論是元件或是網域的負荷都過大，這樣的機制並不適合使用在大型網路上。

本論文提出一套資源規劃及實際執行的架構，供網路營運者運用以提供具端對端品質保證的各種網路服務予使用者。此架構以簡化管理、追求效率，不增加管理複雜度和具高度適用性為原則。除了提供端對端品質保證予使用者之外，本架構能協助網路營運者追求網路資源的有效利用，在所擁有的資源中，盡力提高服務滿足度以獲取最大利益。

而自封包網路建置以來，封包繞逕就是許多學者投入的議題，傳統封包網路其傳送的資料量不大，都是以文字的資料為主，因此已傳送節點數為距離記量單位的最短路徑繞逕演算法即可應付，但是當網路蓬勃發展後，日漸增多的網路應用，使得網路的負擔大大增加，而各種語音及多媒體的應用也要求網路具有一定的服務品質，因此如何使得繞逕具有一定的品質保證，是現在重要的研究議題。

在 BBQ 的架構中，如何為每一個進入網路中的訊務，提供所需的服務品質保證，並且提高系統的使用率，使得系統營運者可以允許更多的訊務進入以提高利潤，都需要一個良好的繞逕演算法配合。

1.2 解決方案 (Solution Approaches)

為了解決上述的問題，在路徑選擇與資源配置上，本研究採取事先規劃之法，先為一段時間內可能進入的訊務事先規劃路徑與分配資源。此外，並將部分的繞逕與資源分配決策分散於 Edge Router。

事先規劃的方式雖然可以彌補即時決策機制的缺點，但也有其困難必須克服，此即預測誤差問題。在資源配置的方法上，必需設法減少預測誤差，或建置彌補預測誤差之機制以降低資源錯置之機會，提高資源使用之效率。

論文將提出一個分散式與階層分權式的品質與資源管理架構，依訊務流集合需求的歷史資料預測未來可能的訊務流集合需求，再以含有預測誤差彌補機制的預算分配及預先資源規劃的方式，提供以預算為基礎之端對端服務品質保證。

另外，本文將在 BBQ 的架構下，提出一個關於路徑規劃方法的最佳化模型，並發展一個考慮服務品質有兼具可行性的路徑規劃演算法 Greedy Path Planning Algorithm，此演算法以歷史的訊務資料為基礎，為未來可能出現的訊務，規劃一組具有頻寬和服務品質考量的路徑，供未來可能進入網路內的訊務使用。我們以事先的路徑規劃取代即時的路徑計算，減少大量的計算負擔，且規劃出的路徑兼具頻寬和服務品質的考量。

且為了測試本文所提出之路徑規劃演算法的效能，我們以訊務通過網路所獲得的利益，以及總利益與網路中鏈結的使用率的比值作為評量指標，將路徑規劃的方法，和傳統的 OSPF 以模擬的方式做比較，其模擬的實驗成果將展示於第五章。

1.3 論文組織結構

本論文共有六個章節，第一章簡介網路技術的演進和關於網路傳輸服務品質所遭遇的各種問題，第二章中將各種提供服務品質保證的網路架構作簡介和評論；在第三章中，將介紹本研究團隊所提出的 BBQ 服務品質管理架構和運作流程，第四章進入本

研究所專注的路徑規劃問題、最佳化的模型，及本研究所提出的解決方案。而在第五章中以模擬的方法來評估本研究提出之演算法的效能，最後是本文結論和未來展望。

第二章

相關研究 (Related Work)

2.1 繞徑問題相關研究

自 1970 年代，網際網路的濫觴 ARPANet 建置以來，Internet 在此後的二十多年快速的蓬勃發展，全球每年上網的人數也日益增加，在國內 Internet 也漸漸成為人們生活不可或缺的一部份，就資料傳送的方式來說，Internet 屬於 packet-switching 網路不同於一般常見的 circuit-switching 電信網路，其中 packet-switching 和 circuit-switching 的不同在於 packet-switching 是將資料以封包的形式傳送，封包在一個個的路由器中以 hop 的方式達到目的地，不同於 circuit-switching network 是以專線的方式來傳送資料，因此封包的繞境問題，一直是學者所專注的議題。處理關於封包傳送的問題，稱之為繞徑方法，主要分為下列幾種：

2.1.1 傳統繞徑方法

2.1.1.1 廣播式繞徑方法 (Flooding)

Flooding 是一種簡單的繞徑方式，即是將從接收端所收的封包，在每一個外送端傳送出去，此種方法會複製大量的重複的封包在網路上傳送，事實上若沒有一個明確的終止命令，此複製的過程會一直持續下去。此種在一般的應用上說來並不實用，但是在某些特定的應用上卻有其價值，如在某些軍事應用上，大量路由器可能因為任何原因隨時故障停擺，因此 flooding 將可以產生強固的可靠性。

2.1.1.2 最短路徑演算法 (Shortest Path Routing)

最短路徑演算法是以建立最短路徑樹來達成繞徑得一種方法，著名的 Dijkstra 演

算法就是一個最短路徑演算法的例子，此演算法可以 $O(n \log n)$ 的時間內找出某一個節點至網路拓樸中其他節點的路徑。因為此演算法在計算時具有網路拓樸中所有的資訊，因此是一種集中式演算法，也由於此種集中式的特性，此演算法擁有無迴圈的特性(loop-free)。

2.1.1.3 訊務基礎路徑演算法 (Flow-based Routing)

Flow-based routing 與其他種類的繞徑演算法不同之處在於，Flow-based routing 同時考慮網路的拓樸和其上負載，其基本的假設如下，若一鏈結的傳輸能力和平均的流量已知的話，就可以根據以上資訊利用 Queueing 的理論來得知封包通過此鏈結所需的平均傳送延遲時間，依此可以很容易的計算出通過整個網路所需的時間，那麼就可以簡化繞徑的問題成為找出整個網路上最小傳送延遲。

2.1.1.4 距離向量繞徑演算法 (Distance Vector Routing)

距離向量的繞徑演算法利用某一些方法來估算從來源端到目的端的距離，一般稱距離向量方法為 Bell-Ford 的演算法，因為此方式是基於最短路徑計算方法，從早期 ARPANET 至今日的 Internet，距離向量的繞徑演算法就廣泛的被採用，如 hop count 就是一種常用的距離計算方式，RIP 繞徑方法就屬於此類，距離向量的演算法會定期的其鄰近的節點傳送資訊，每一個節點可以從其所經過的節點來計算至其節點的距離。若網路形狀固定，則可以得到一個收斂的結果，而繞徑的更新資訊只會傳送給鄰近的節點，此資訊並不會造成太大的網路負擔，距離向量的繞徑方法弱點在於其運算的收斂時間，其收斂的時間會被定期的繞徑更新資訊傳遞時間所影響，所以 RIP 的演算法限制所經過的節點最長不能超過 16 個，雖然如此，由於 RIP 的設定容易、管理成本低且易於實作的特性，使其在小型的網路還是廣泛的被運用。

2.1.1.5 鏈結狀態繞徑方法 (Link State Routing)

Link State 繞徑方法與距離向量的不同在於其注重的是鏈結上的資訊，Link State 的方法將 Link 上的狀態透過一些數學的轉換，然後根據此資訊來選擇適當的路徑 Open Shortest Path First(OSPF) 就是一種被大量使用的 Link State 繞徑方法。

2.1.2 具服務品質保證之繞徑方法 (Multi-Constrained Path)

在具有 QoS 保證的繞徑方法研究中，已經有學者提出不同的方法，但其中基本的構想都是在現有的繞徑協定中，如 Distant Vector 和 Link State，做適當的修改來符合 QoS 的要求，近年來有些研究將主題專注於此一方面，稱為 multi-constrained path(MCP) 的問題，而 MCP 的問題的目標，主要在找出符合數個 Constraints 的路徑，如 Gang Feng, Kia Makki 等在「An Efficient Approximate Algorithm for Delay-Cost-Constrained QoS Routing」[19]中所提出的方法為，將 Delay 和 Cost 結合成為一個單一項，然後再以此 Vector 作為 Dijkstra's Algorithm 的參數來計算路徑，其所得到的路徑將會 Cost 和 Delay bound 的需求。[20, 21]

然而以上的方法並沒有辦法滿足我們的需要，首先，我們的目標跟 MCP 是不一樣的，Path Planning 的最佳化目的在於為某一個 Core Network 上所可能發生的 Traffic 找出一組合適的 Path Set，此 Path Set 必須滿足某些 Traffic Demand 對於 QoS 的要求，讓 Run-time 時候的經過 Admission Control 所許可的進入 Core Network 的 Flow 達到其所需的品質要求，但是 MCP 所追求的卻是符合某些 Constrains 之下最佳的一條路徑。若將 MCP 中，前人所提出的方法套用在 Path Planning 中，可預期的結果是只能滿足某些 Request 的需求，而且可能造成整個 Core Network 資源利用的不平衡，因此 MCP 所提出的方法將不適用於我們 Path Planning 的問題。

2.1.3 繞徑方法之評論

繞徑的方法是達成服務品質的重要關鍵，可是在傳統的繞徑方法中，通常能對訊務以 Best Effort 的方式來送達，並無考慮其傳送服務品質的問題。而近年來對於提升服務品質保證的繞徑方法已經越來越受重視，許多相關的研究發別被提出，學者通常稱此類的問題為 Multi-Constrained Path Problem，許多學者也認為此為一個 NP-Complete 的問題，因此對於此也相繼提出不同的 Heuristic 的演算法來解決。

2.2 評論 (Summary)

如果對每一個訊務以 real-time on-demand 去要求資源建立路徑以提供端對端的品質保證，其 overhead 將極為可觀，例如 IntServ 使用 RSVP 的方式預留資源，必須針對

每個訊務的需求在核心網路內逐步搜尋，得到一條保留路徑，因其 overhead 太大，只能適用於小型網路。而 DiffServ 雖然以 per-aggregate 的方式減低管理的複雜度，但是這種 per-aggregate 的服務品質保證，卻沒有辦法對於單一個訊務提供絕對的端對端服務品質保證。

現階段的資源管理多以集中的方式配置資源，網域的允入控制由一元件統一執行。這樣的作法於資源易受限於核心網路大小與網路流量，當網域過大或者流量過多，除中央統籌元件負荷量重與網路 overhead 過大外，在實際執行上也有困難存在。

在上述的架構中，在路徑規畫與資源分配上，多半是以事先規畫的方式，減低即時運算的過量負荷。在 TEQUILA 的架構中，品質參數多，品質管理的模型較為複雜，其 Dynamic Route Management 的目標為平衡負荷(load balance)，Dynamic Resource Management 則負責管理與調整連線頻寬(link bandwidth)與暫存器空間(buffer space)，然而對於訊務預測的誤差並沒有辦法有效彌補，當預測誤差過大時，系統會有過多的即時運算，承擔過多的負荷。至於 Victor O.K. Li's system 以分散式的方法配置資源，可能會有較高的資源浪費，所以每隔一段時間便會依各個 Ingress Router 的使用情況重新調整，然而這樣的方式在突然進入大量訊務(burst)的情況下會無法處理。

鑑於提供端對端品質管理所面臨的問題，本論文之研究目標在於提出一簡單的端對端品質服務架構，以預算分配為基礎，使用簡化的單一品質參數，使得系統業者可以調整自己的品質管理策略，來增加系統效能，減低管理效率。除此之外，系統佈署(deployment)的容易與否也是我們考量的重點之一。

在本論文所提出的 BBQ 架構中，以分層管理的方式解決端對端的品質問題，將資源管理與允入控制分開進行。資源管理方面，核心網路內部頻寬資源由 BB 統一控管，依照網域各個 Ingress Router 需求不同進行適當的分配，再由 Ingress Router 內部元件事先進行路徑規畫，並依規畫結果 per-flow 進行允入控制。在本論文後面的章節，除了 BBQ 管理系統的介紹外，另將著重於 BBQ 架構中，核心網路上的路徑規畫，在我們所提出的資源規畫方案中，核心網路上的資源分配方案和路徑規畫息息相關，接下來本文將介紹整個 BBQ 的整體架構，接著說明 BBQ 中核心網路內的資源分配方法，然後分析路徑規畫的問題，並提出一個路徑規畫的獲利最佳化方案。

第三章

以預算為基礎之服務品質保證 (Budget-Based QoS)

在異質性極高的下一代網路上，要提供 per flow End-to-End QoS 是一項管理複雜度極高的工作，但唯有 per flow End-to-End QoS 才可提供使用者絕對的服務品質保證。

BBQ 之設計目的即為提供一個簡單易行的 per flow End-to-End QoS 管理架構。

要提供適當的 QoS 管理，其成功之關鍵主要在於是否能提供一個簡單易行之架構，再據此設計各種解決方案。以目前管理趨勢，由於 Per flow QoS 將造成大量的管理負擔，多使用訊務集合技術(Traffic aggregate)將許多 flow 歸併，減低管理負擔。另一技術則將網路上的資源依照服務品質優劣做等級區分，採用分級分流管理，分級收費的方式。目前最風行的 DiffServ，即基於此兩種理念，惟無法提供 per flow 的服務品質保證，也無法提供端對端服務品質保證，仍為極大缺憾。為了克服此問題，BBQ 之目標即在設計一個可提供 per flow 端對端服務品質保證的架構與工具。

BBQ(Budget-Based QoS)採用預算為基礎之服務品質管理，以簡化管理、追求效率，不增加管理複雜度為原則。根據這種簡化管理的原則，利用分層分權的方式將 QoS 管理權責以預算之方式分散到每個網路元件。並且，避免繁複的折衝協調，尤其是應盡量避免即興式 (real time on demand)的資源管理，盡量以預先規劃取代即興式資源分配。BBQ 提供一個高適用性的管理架構和相關的管理工具，可適用於不同的下層網路架構和不同營運目標的網管政策。以下將簡介 BBQ 的網路架構和管理系統架構。

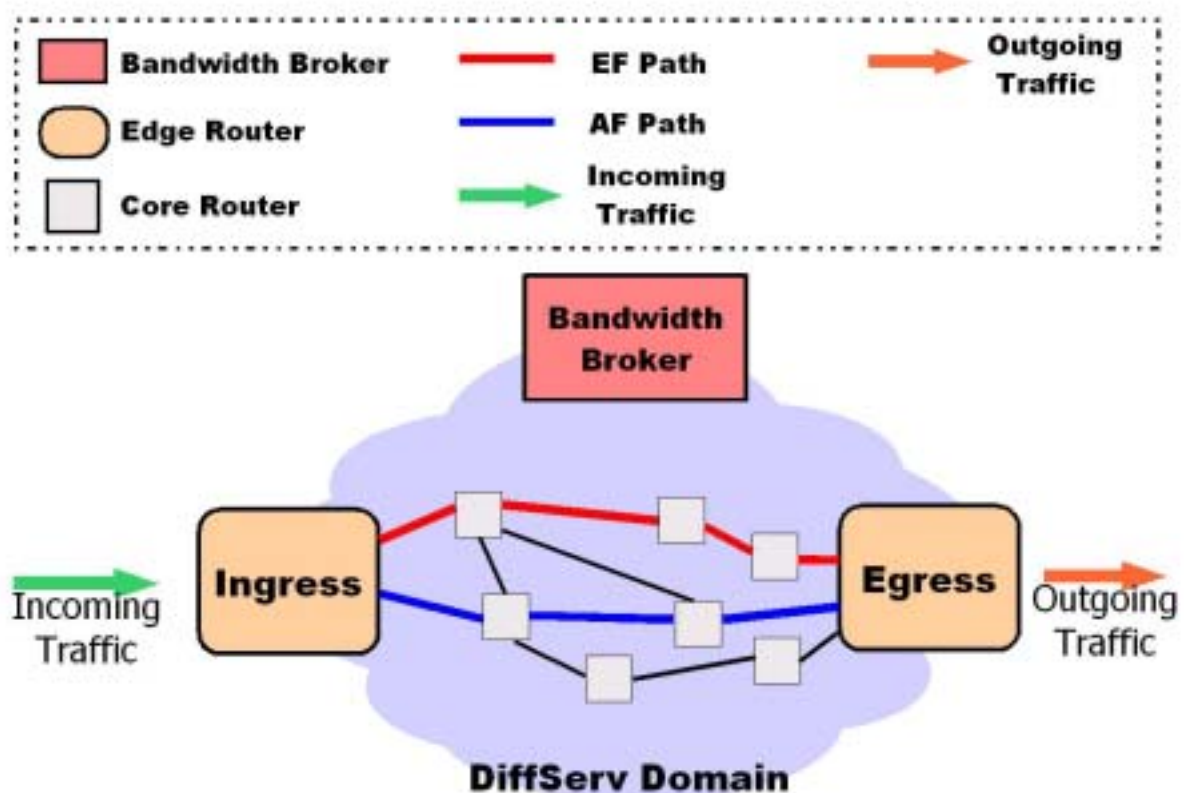
3.1 BBQ 中的核心網路架構與 QoS 元件 (Core Network Architecture and QoS Components for BBQ)

一個端對端的網路架構中包含許多的核心網路。在 BBQ 的架構之下，我們假設一個核心網路是由一個電信公司所獨自擁有的網路，而各個核心網路的營運者都有其管理自身的管理政策。在 BBQ 分層管理的概念下，一個端對端具服務品質保證的資料流

可能會由多個不同電信公司所管理的，當最上層端對端管理元件將 QoS budget 分配到至核心網路後，核心網路上執行 QoS 保證的元件就負責滿足品質要求，完成具品質保證的端對端服務 Ingress 至 Egress，如此分工可有效降低管理的複雜度。

而為了提供每個訊務通過核心網路的品質保證，在核心網路中，我們採用資源預先配置方式，訊務若被允許進入此網路中，則會獲得一定的資源，以保證訊務通過網路時，可以達成所需之服務品質保證。

在 BBQ 中的核心網路架構，將會在 DiffServ 的為基礎的網路架構中，加入我們所提出的資源管理和傳輸服務品質架構之機制，藉以提供網路營運業者一套簡單易行的管理工具，來達成 End-to-End 服務品質保證，讓網路營運者調整其品質政策以達成其最大的滿意度。



3.1.1 核心網路資源規劃方法

為了提供每個訊務通過核心網路的品質保證，我們建議指定每個訊務的傳送路徑。在路由方法上，主要可以分為集中式(centralized approach)與分散式(distributed approach)兩種，集中式是由一元件負責所有 Ingress Router 進入資料流的路徑規劃選擇；分散式則由各個負責允入的 Ingress Router 找出可以滿足資料流品質需求的路徑。

如 3.1.7 節所述，分散式的資源分配較能避免不公平的問題，而且也較能因應突增的訊務流，本研究採用分散式搭配集中式的方法路由，並輔以適當的資源配置法，規劃成一個個完整路徑，給 Ingress Router 允入使用。資源配置法考量時效性問題與管理複雜度，避免採用即時資源配置受到較多限制，而採用預先分配法，事先進行資源規劃。

3.1.2 核心網路內的資源規劃元件

在 BBQ 的架構之下，我們假設一個核心網路是由一個營運者所獨自擁有的網路，因此而一個端對端的網路架構中包含許多的核心網路。在 BBQ 分層管理的概念下，負責 long path 及 short path 的元件各自負責在指定的品質預算內規劃出一組組 long path 及 short path。

目前我們假設核心網路為一 DiffServ 網路，由數個 Edge Router 與 Core Router 組成，除了 Core Router 僅負責傳遞資料外，QoS 的管理主要分散在核心網路協調元件與各個 Edge Router 上。如圖 3.7 所示，核心網路 QoS 管理元件主要有核心網路協調元件與 Edge Router，Edge Router 又分為 Ingress Router 與 Egress Router，其中核心網路協調元件包含資源管理元件與核心網路路徑規劃元件，Ingress Router 則包含頻寬訂購代理人、允入控制代理人與路徑規劃元件，上述元件的功能分別如下：

在 BBQ 架構中，整個 BBQ 的架構中包含了幾個重要的元件，負責實際的 QoS 保證功能的執行，以下就網路的架構來說明 BBQ 中的元件及其功能：

- **核心網路協調元件(Core Network Coordinator, CNC)**

在每一個核心網路之中皆有一個核心網路協調元件(以下簡稱 CNC)，是核心網路主要控制元件，也是管理上核心網路對外的窗口，其內包含兩個元件：

(a) 資源管理元件(Bandwidth Broker, BB)：負責對核心網路內的資源做適當的分配，主要採用分層管理的精神，在系統初始的時候將核心網路內的頻寬資源交與各個入口路由器做利用。

(b) 核心網路路徑規劃元件(Short Path Planning Agent, SPPA)：主要負責將欲進入該核心網路之訊務預測，以中央集中式繞徑及資源分配方式計算出其路徑需求，再轉換成每一個鏈路(link)的需求，交由負責批購頻寬的元件，購買頻寬時的參考。

- **邊界路由器 (Edge Router)**

在網域最外圍連結其他網域的 router 稱為 Edge Router，資料流的進入的 Edge Router 稱為 Ingress Router，離開的 Edge Router 則為 Egress Router。

- (a) 入口路由器 (Ingress Router)

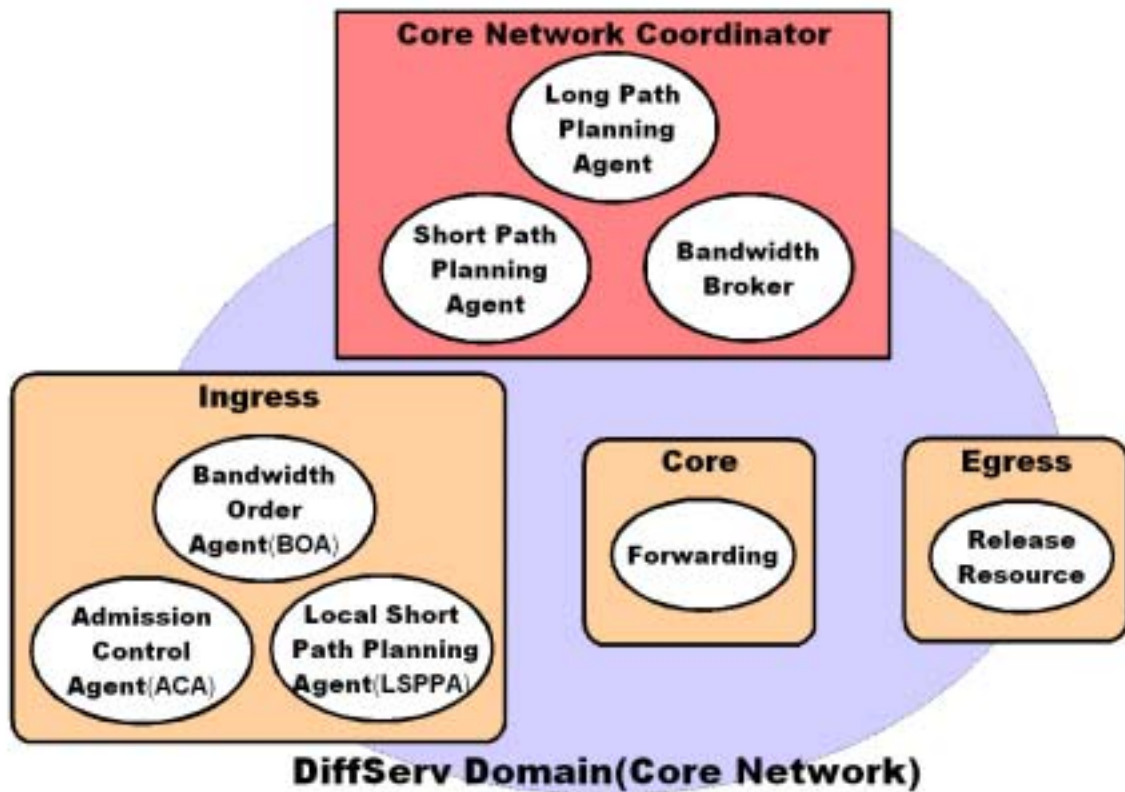
- (i) 頻寬訂購代理人(Bandwidth Order Agent, BOA)：根據以往的訊務統計由 short PPA 供給計算最佳批購量，向 CNC 元件訂購所需的頻寬交由 Local SPPA 規劃成可用路徑。

- (ii) 允入控制代理人(Admission Control Agent, ACA)：依據所掌握的路徑資源來決定是否可以滿足資料流的需求，若允許某一訊務流進入網域，就表示可以保證滿足此資料流對於傳送品質的要求。

- (iii) 路徑規劃元件(Local Short Path Planning Agent, Local SPPA)：此元件會將 BOA 所批購回來的資源，規劃成有各種品質的路徑所組成的路徑組，供 ACA 在系統執行時使用，各個 Local SPPA 可根據各個 Edge Router 的情況選擇規劃方法，不一定強求一致。

- (b) 出口路由器(Egress Router)

訊務離開的 Edge Router，當資料流結束傳送時，Egress Router 負責釋放原先配置的資源，以利之後進入的資料流使用。



3.1.3 集中-分散式資源規劃運作流程

BBQ 架構是以預測的方式規劃網域的資源，來應付未來訊務的資源需求，因此需要一套良好的資源規劃方法，在 BBQ 的核心網路中利用分層管理，權責區分的精神，採用批發零售的方式，來管理網域內的資源。每一個鏈結的頻寬皆由 CNC 中的 Bandwidth Broker 統一分配，BBQ 採用集中-分散式方法作資源規劃及分配。第一階段採用集中式，如果有必要，則進行第二階段分散式的資源分配，由 BOA 根據預測，向 BB 批購每一個鏈結的部份頻寬，批購來的各段鏈結頻寬由 Edge Router 自由使用，組成各種路徑分配給進來的資料流使用，整個規劃與分配流程如下：(圖 3.8)

首先將整個訊務型態的歷史資料分成幾個時段，挑選出與下個預測時段特性相似的一個參考時段(例如最近的參考時段)，將這一個時段的訊務輸入 SPPA 計算出一組最

佳的路徑。此為集中分配之結果，即可依路徑之起點，分配給所有 Edge Router。如果因為公平性的問題，或各 Edge Router 有特殊考量則可啟動第二階段分散式的資源規劃與分配。我們所設計的分散式規則，將考慮流量需求的統計特性，計算各個鏈結資源需求的機率，求得最佳的頻寬需求期望值。首先利用第一階段求得的路徑組，將每一個參考時段的訊務輸入這些路徑，求得每一個鏈結上的頻寬需求，當所有參考訊務逐一輸入計算後，即可求得每一個鏈結上訊務頻寬需求的統計分佈，以此分佈當作欲規劃時段該鏈結上訊務需求的機率分佈，BOA 則可根據這些需求之機率分佈計算出最佳批購值(成本期望值最低)。以下是詳細的資源規劃流程：

- 步驟一. 首先，我們會將整個訊務型態的歷史資料分成幾個時段，挑選出與下個預測時段相似的參考時段，根據這一個時段的訊務特性還有上次資源的使用情形，來為各個 Ingress 下個時段資源需求做預測。
- 步驟二. 此時 Short Path Planning Agent(SPPA)會根據此一參考時段內各個 Ingress 上的頻寬需求和此核心網路內的頻寬，計算出最適合此時段訊務特性的一組路徑，此路徑組是由從各個 Ingress 出發具有不同 QoS 品質的路徑所有組成，預料將可以滿足下一個時段中不同應用的品質要求。此為第一階段的資源分配及規劃，若採集中式的資源分配及規劃，則資源規劃程序在此結束。如啟動第二階段的分散式資源分配及規劃，則進入步驟三。
- 步驟三. 各 Ingress Router 中的 LPPA 將第一階段所得之路徑組，輸入不同參考時段的訊務，得到各鏈結上不同的訊務頻寬需求，形成一個頻寬需求的分佈，然後將此資訊交給 BOA(Bandwidth Order Agent)，BOA 會根據此一資訊向資源的管理者 BB 批購所需要的資源。
- 步驟四. 而 BB 則會根據每一個 BOA 的需求來決定是否給予其所需的資源，若 BOA 的需求不被允許的話，則 BOA 會依其考量與 BB 協調改變購買的資源。
- 步驟五. 當每個 Ingress 上的 BOA 購得所需要的資源之後，會將此 Link-base 的資源交給各個 Ingress 之中的 Local SPPA 來做路徑的規劃，Local SPPA 會利用這些資源規劃出適合的路徑組，儲存在各個 Ingress 的 Resource Database 中，供 Admission Control Agent 和 End-to-end 的 Agent 來查詢使用。

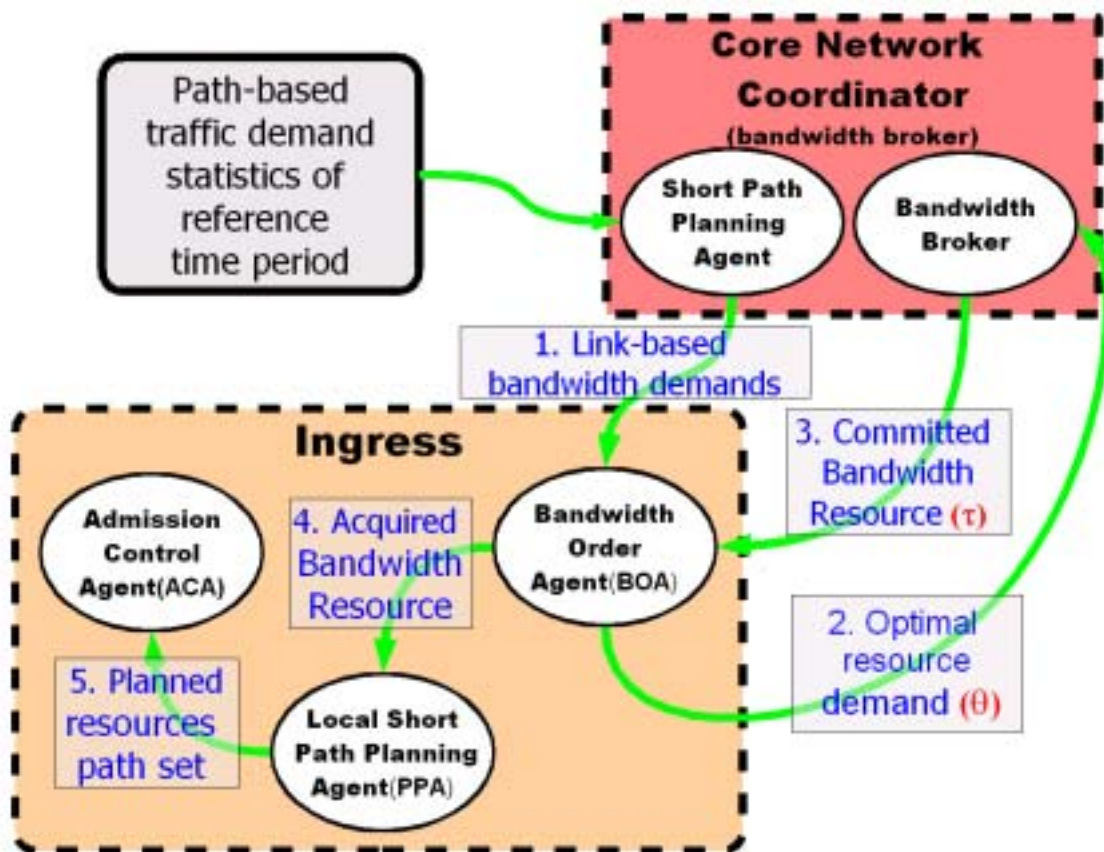
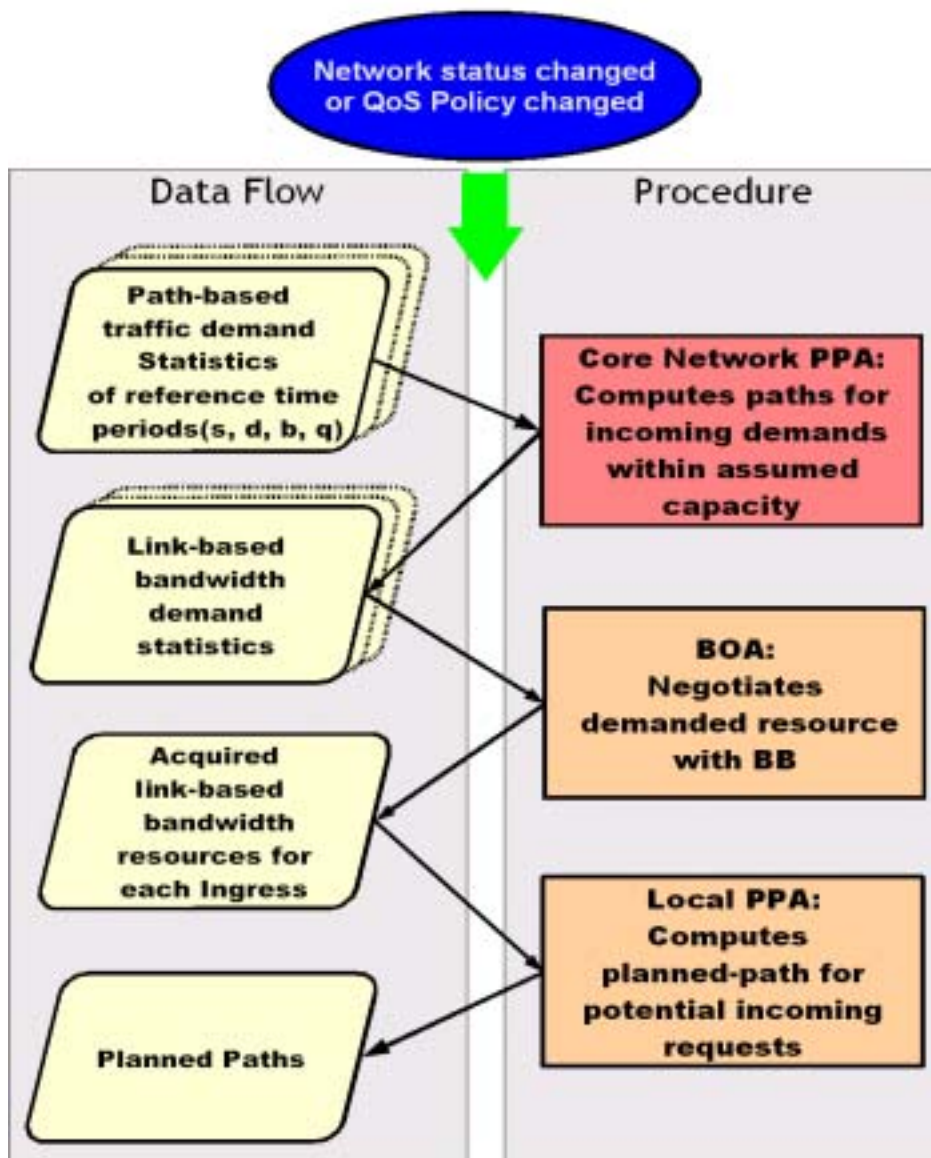


圖 3.9 是以資料流的流向來說明核心網路內的資源規劃程序：



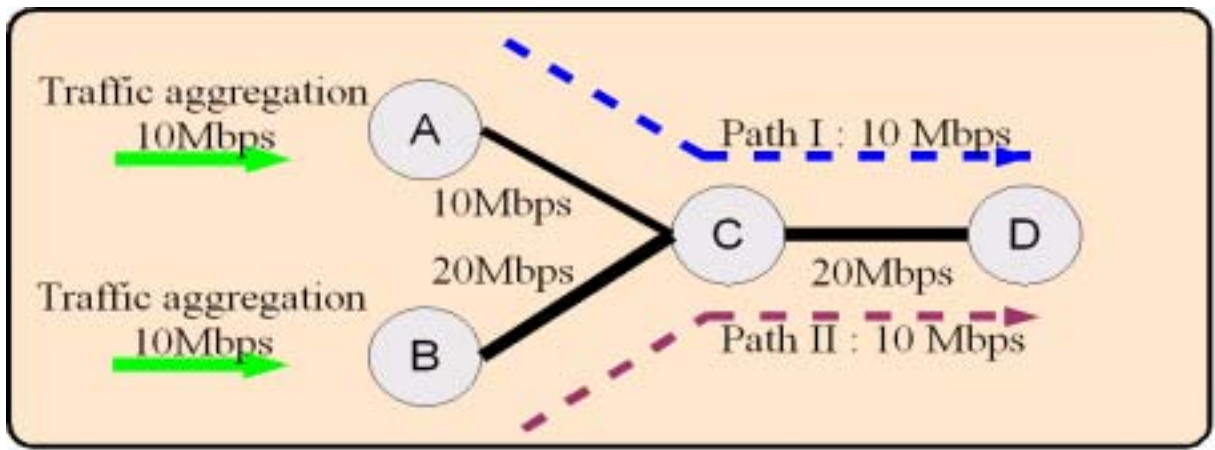
3.1.4 資源調配可能產生之問題討論

在上述的各種可能的資源調配方案中，由於資源是採預先分配的方式，在資源不足的情形下，可能會產生如下的情形，而無法提高系統的效率：

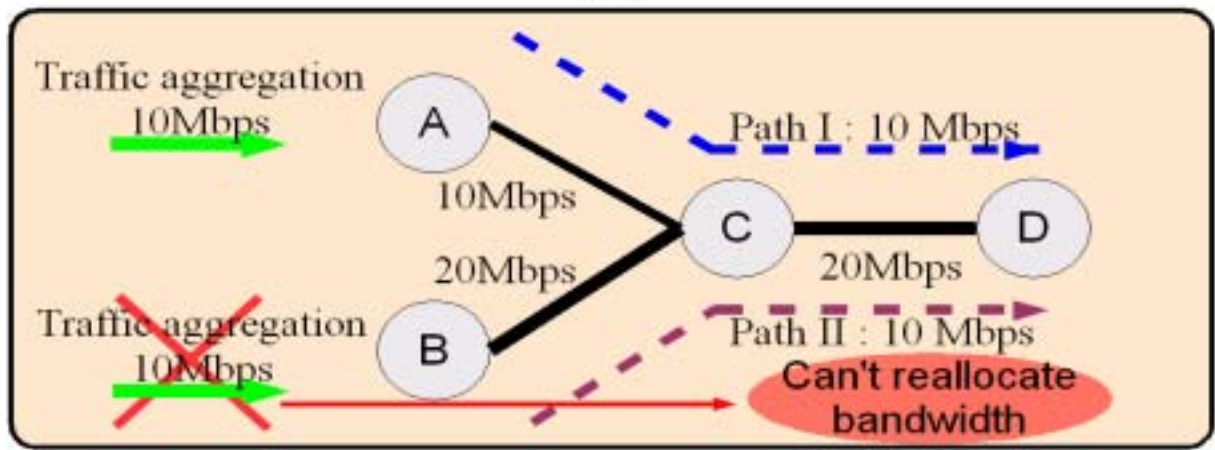
3.1.4.1 Inter-Ingress 資源浪費

為了達到一定的 QoS 保證，我們為每一種可能進入核心網路中的訊務，預先規劃傳送的路徑，並依歷史需求預先保留一定量的資源。因為所有的計算皆是預先進行，因此在系統運作時，可以免除大量計算的負擔。此種方法，雖然可以達到預期的品質保證，這些鏈結上的頻寬資源，實際上是經過兩個階段的預先資源分配後，才成為系統運行時，真正可以供 ACA 運用的一組具有服務品質保證的路徑，在第一階段中，每一個 Ingress 中的 BOA 元件先向 Bandwidth Broker 購買所需資源，一旦購買完成之後，核心網路中每個鏈結上的資源，可說就由 Bandwidth Broker 分配給了每一個 Ingress Router。而第二階段的資源分配則是在每一個 Ingress 中的 PPA 進行路徑規劃後，Ingress 自 Bandwidth Broker 所購得每個鏈結上的資源，就歸屬於所規劃出的每一條路徑所有。這些已經分配的鏈結資源即無法在這一個時段內進行重新分配的作業，但是系統運行時，可能進入核心網路內的訊務，卻有其不確定性，每一種訊務都可能有其獨特的需求，一個 Ingress 以預測的方式所規劃的路徑，可能跟系統運行時真正訊務的需求，有很大的出入。

在上述的兩個資源分配階段中，都有可能形成資源的浪費，在圖 3.10 和圖 3.11 中我們舉出二個頻寬浪費的例子，分別以兩個資源分配的角度切入說明可能形成的資源浪費。在一次路徑規劃中 A、B 的兩個 Ingress Router，分別有 Path I 經節點 A, C, D 和 Path II 經節點 B, C, D 兩條路徑到達 Router D，在共用的鏈結(C, D)中，Ingress A, B 所購得的頻寬分別為 10Mbps，在圖 3.10(A)中，若在系統運行的某一個瞬間，分別有總量為 10Mbps 的 traffic 進入 Router A 和 Router B，Path I 和 Path II 剛好符合 Traffic 的需求而被允許進入核心網路中；可是如情形如圖 3.10(B)中，當進入 Router A 和 Router B 中的 Traffic 其頻寬總量所需分別改變成為 5Mbps 和 15Mbps，則原本 Router B 所規劃的 Path I 就不在符合其要求，因為在鏈結(C,D)上 Path I 只被分配了 10Mbps 的頻寬，就算 Path I 經過鏈結(C, D)只需要 5Mbps 的頻寬，我們已無法適當的加以挪用，因而造成了鏈結(C, D)的頻寬無法被充分的利用，導致真正有需求訊務被拒絕，結果將會造成整個系統的效率低落。



(A)

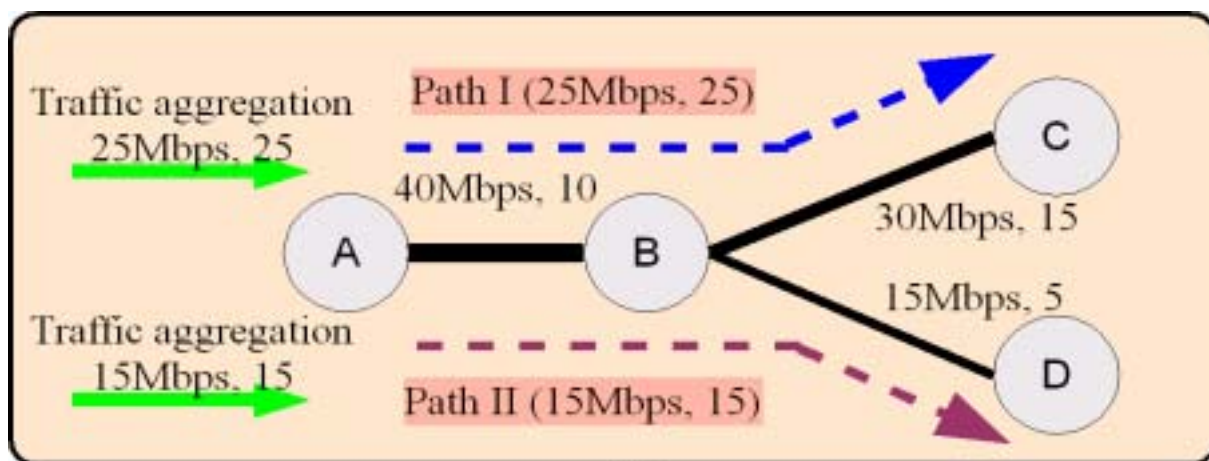


(B)

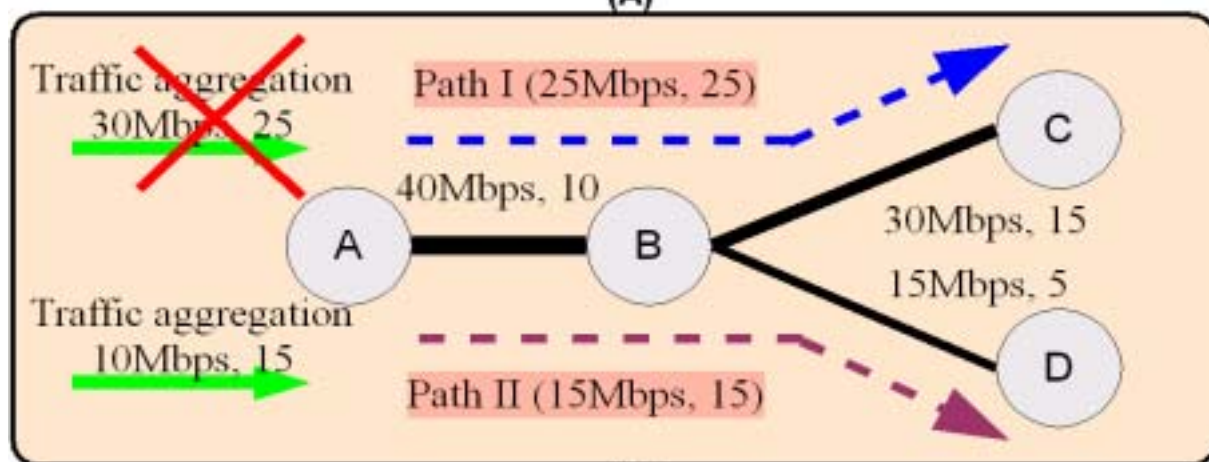
3.1.4.2 Intra-Ingress 資源浪費

而圖 3.11 則是另一種可能形成的資源浪費，Ingress A 向 Bandwidth Broker 購買了鏈結(A, B)、鏈結(B, C)和鏈結(B, D)上的資源，此三條鏈結分別具有如表 n 所示的服務品質參數，在第二階段的路徑規劃中，在 Ingress A 中的 PPA 將這資源規劃成兩條路徑 Path I 和 Path II，其上的所可以通過的頻寬和服務品質商數分別為 Path I : 25Mbps, 25、Path II : 15Mbps, 15。若在系統運作時，進入 Ingress A 之訊務分別為，服務品質需求為 25 的訊務量其頻寬需求總量為 25Mbps 和另一個服務品質需求為 15 的訊務量，其頻寬需求總量為 15Mbps，需要通過此網路，則規劃出的 Path I 和 Path II 則是恰好可以滿足

其需求。但若今天服務品質需求為 25 的訊務量增加，變成其總頻寬需求為 30Mbps，而另一個頻寬需求為 15 的訊務量，其總頻寬需求只有 10Mbps，於由路徑規劃的不可變性，Path II 中有剩餘的頻寬資源，而 Path I 卻面臨頻寬不足的情況，使得 Path I 依然只能容許 25Mbps 的訊務進入網路中，因而造成鏈結(A,B)中的頻寬浪費。



(A)



(B)

3.1.5 資源調配問題之解決方案

對於上述討論中，資源配置時所可能產生的浪費情形，在資源充足的情況之下，並不會對系統效能造成嚴重的影響。但是當系統資源不足，卻有部分元件掌握剩餘頻

寬造成浪費時，就大有改進空間。以下是一些解決的方案：

對於 Intra-Ingress 之資源錯置問題，可由 Ingress Router 內的 ACA 元件負責，因為在 Run-time 時，ACA 掌握了這個 Ingress 內所有路徑的資源和其使用情形，當指定給某訊務的路徑不足以滿足訊務所需時，ACA 可以彈性的從其他未被完全使用的路徑調配資源給予有需求的訊務。

解決 Inter-Ingress 之資源錯置問題，則需要 BB、PPA 和 ACA 共同協調，簡單的作法之一，即是採用臨時批購鏈結頻寬的作法[23,24]。在 run-time 時，不能冒然將已分配給其他 Ingress Router 的資源挪用。

第四章

核心網路之路徑規劃

封包所行經一連串的節點和鏈結，就是所謂的路徑(path)。在傳統的封包網路(datagram)中，封包是以 hop 的方式傳送，每個封包之間是彼此獨立的，並不像 circuit-switching 網路中資料的傳送需要事先建立連線，因此在封包網路中，路徑的概念與 circuit-switching 網路中稍有不同。

為了某些目的，網路營運者可能會為一個封包指定一條路徑，此條路徑可能是預先靜態指定或是動態指派。為一個封包找一條路徑的程序稱為路由。某些路由執行方法是由網路營運者依據路由的結果為所有的路由器設定路由表。封包的行進路徑可能是根據由路由模組為路由器設計的靜態路徑，也可能是根據由路由模組為封包規劃的動態路徑。如果封包能依規劃的路徑行進，確保服務品質之保證將更容易。

但在傳統的封包網路中，對於服務品質的保證並沒有太多的機制來確保。封包在傳送的過程中，在 hop 之中都需要經過檢驗封包的動作，查看其傳送目的地的動作，這處理封包的動作，在傳送經過的每一個節點都需要進行一次，而同一 flow 中的封包，傳送到達目的地的時間不定，另外也可因為網路擁擠或其他因素導致封包遺失的現象，網路只能盡力(Best Effort)將封包送達。此種 Best Effort 的網路在傳統的應用中(如 email)，只需負責靜態資料傳遞的封包網路上是足以應付所需，但若要以 packet-switching 來傳送即時性資料，則會出現嚴重的問題。例如一個即時的 VoIP 應用，若傳遞語音資料的封包到達時間不一，甚至資料封包遺失，則會造成語音斷斷續續，無法讓人舒服的進行一個交談的過程。

如前所述，關於服務品質保證的研究中，多以傳送延遲時間、傳輸延遲抖動和封包遺失率作為服務品質評量的標準，而許多服務品質保證的相關研究被提出以解決上述的問題，如 IntServ、DiffServ 和 MPLS。 IntServ 使用 RSVP 來即時保留資源給訊

務，給予訊務 per-flow 的保證；DiffServ 將訊務在 Ingress 預先分類，而在其內的 core router 則會依據所每一個分類的 PHB(Per-Hop Behavior)來處理封包；而近來十分熱門的 MPLS 技術則是使用 label 將封包在 Ingress 標記，然後在 MPLS Domain 中循預先指定的路徑傳送，各個節點只需要根據封包標記加以轉送，大幅降低封包處理的時間，進而改進封包傳送延遲過長的問題。

可是只提高封包傳送的速度，並不足以保證傳送品質。以常見的服務品質指標 – 傳送延遲來說，在 DiffServ 中，每一個相同分類的封包，在每個 Router 中的轉送服務時，雖然受到某種程度的資源保證(PHB)，但是封包經過幾個節點的轉送後，通過此一 DiffServ Domain 後的總傳送延遲(Total Delay)，卻可能遠超過訊務要求的傳送延遲，對於即時性應用之品質而言，將造成嚴重傷害。且在 DiffServ 的服務中，服務品質的保證是以 Class 為基礎，一旦網路的負擔較重時，在同一 Class 會被丟棄的封包並不知道是屬於何個 flow，無法對 flow 提供絕對的 QoS 保證。因此，如欲提供核心網路內的 End-to-End QoS 予某一應用，應在 DiffServ 機制上加上路徑規劃，引導封包循特定路徑轉送，才能提供 End-to-End QoS。但路徑規劃耗時較久，使用簡單的即時路徑規劃容易導致資源浪費。

反之預先的路徑規劃可以有充裕的時間作較佳的規劃，提高網路效率，加速封包處理速率。本研究以歷史資料預測將來可能發生的訊務，然後預先規劃出符合訊務頻寬和傳輸品質要求的路徑組。在系統運行時透過 source routing 的方法，讓封包在一進入網路的時候，就在指定的路徑上傳送，此法類似 MPLS 在事先指定傳送之路徑，但 MPLS 僅以 IP 為依據，所指定之傳送路徑未必符合每個訊務流之特定品質要求。本法則根據訊務流之品質要求及 IP 位址挑選路徑，由於每個訊務所傳送的路徑經過適當的規劃，不但可以將流量分散，也由於封包所經過的路徑已知，所以封包經過整個網域時的傳輸品質是可以獲得控制。

4.1 BBQ 架構中各層級之路徑 (Path Definition)

在 BBQ 的架構中我們以服務品質保證的路徑，來達到服務品質保證，而透過分層分權的方式，各層元件各自規劃該層級的資源組成所需之路徑，提供給上層元件規劃之用。Short Path 為一條穿越核心網路且提供服務品質保證之路徑，由 Ingress PPA 負

責規劃；Long Path 為一穿越 Backbone 且可提供服務品質保證之路徑，由 LPPA 負責規劃。End-to-End Path 則為 end user 到 end user 且實際提供端對端服務品質保證之路徑，由 GACA 負責取得。表 4.1 列出各種在 BBQ 中路徑的名詞定義，及其負責規劃之元件。

表 4.1：BBQ 架構中各種路徑之定義

規劃路徑	管理元件	路徑能力
端對端路徑 (End-to-End)	接取網路之 Global ACA	End-to-End path 由通過接取網路之路徑和 long path 相連而成，同樣可提供服務品質保證。
長路徑 (Long Path)	各核心網路之 LPPA	Long path 為可通過骨幹網路之路徑，並且可提供服務品質保證。
短路徑 (Short Path)	各核心網路之 SPPA	Short path 為通過核心網路之路徑，並且可提供服務品質保證。

4.2 在 BBQ 架構中的路徑規劃元件 (Path Planning Agent in BBQ Architecture)

在 BBQ 的架構中，負責規劃某個核心網路的路徑之元件稱為 Path Planning Agent(PPA)，每個核心網路中總共有三種 QoS 路徑規劃元件，一為 LPPA，主要是負責 end-to-end 的路徑安排和協調，而在核心網路內的路徑規劃，分別由兩個元件負責，一個為位於 Core Network Coordinator 中的 Short Path Planning Agent(SPPA)，另一則為在每一個 Ingress Router 中的 Local SPPA(表 4.1)。

SPPA 位於 Core Network Coordinator 之中，主要的功能是依據各個 Ingress 過去統計的訊務歷史資料計算出一組最能符合未來可能出現訊務需求的路徑，此一組路徑係針對網域內所有資源考量而提供給所有可能進入該核心網路的訊務使用。其次將此一組路徑依其所屬的 Ingress 分割，分配給各 Edge Router。如果是採用集中分配法，則路徑規劃的程序在此結束，將結果放入各 Edge Router 之路徑資料庫中備用。如果分配的結果不公平，或各 Edge Router 有不同的考慮，則可採用集中-分散混合式進行第二階段資源分配與路徑規劃。

第二階段的規劃，首先各 Edge Router 將第一階段分配所得之路徑分解，計算出每一個 Ingress 在每一個鏈結上的頻寬需求。再將此資訊將交給位於每一個 Ingress 內的 BOA，BOA 根據某些原則向 BB 重新批購各鏈結之頻寬。

而每個 Ingress 之中另外還有 Ingress PPA 的元件，負責將 BOA 經過協調後所購得位於每一條鏈結上的頻寬資源，組成一組一組的具有服務品質的路徑，儲存在 Ingress 之中的路徑資料庫之中，供負責 end-to-end 路徑規劃的 LPPA 作為規劃 end-to-end 路徑的參考和核心網路內之訊務支援。在系統運作的時候，允入控制元件(ACA)也會根據此一路徑的資料表來為每一個欲進入網域中的訊務挑選符合其頻寬和傳輸服務品質的路徑並指派給該訊務。

因此 PPA 的目標就是依據預計會發生的需求，找出一組可以滿足最多需求的路徑組合，而且此一組路徑必須可以滿足訊務所要求的頻寬和傳輸品質上的要求。在傳輸品質上，我們將以本研究所提出的 Quality Entropy 作為品質評定的指數。

表 4.2：BBQ 架構中各種 PPA 的比較

名稱	位置	功能
Long PPA(LPPA)	位於 Core Network Coordinator 中	1. 為各自所屬 Stub Network 發生之訊務需求規劃 Long Path。 2. 提供 Short Path 給其他 LPPA 規劃 Long Path 之用。
SPPA	位於 Core Network Coordinator 中	負責規劃核心網路的 Short Path。

名稱	位置	功能
Local SPPA	位於每一個 Ingress Router	規劃屬於該 Edge Router 之 Short Path。

4.3 路徑規劃的環境假設

在 BBQ 中我們以一個單一的 Quality Entropy 來評量傳送品質的好壞，此一參數是由系統業者依其所需制訂，現階段此一參數的設定並非本研究所著重的部分。為了規劃工具所需，我們假設 Quality Entropy 具有可加性(additive)。例如，傳輸延遲(Delay)，即為一種可加性的服務品質參數。若一條由 A 至 C 傳送路徑經過 ABC 三個點，則封包從 A 傳送到 C 的傳輸延遲可由 A 至 B 的傳輸延遲加上 B 至 C 的傳送延遲而得。

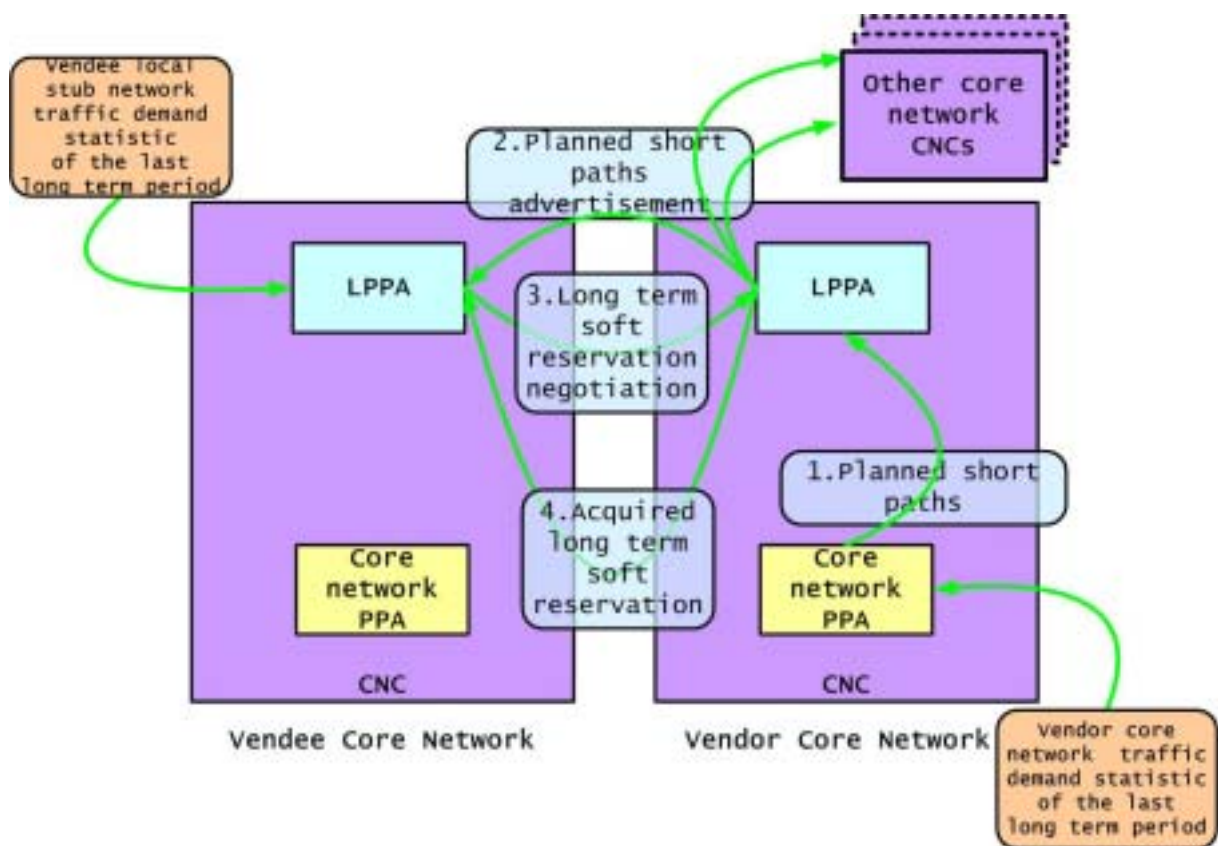
而在每一個鏈結上，如果在此鏈結上的所使用的頻寬沒有超過鏈結上的頻寬容量，我們假設其下屬網路(如 DiffServ)可以保證所設定的 Quality Entropy[22]。

4.4 端對端規劃路徑元件與核心網路內路徑規劃元件之互動

在 BBQ 架構中，核心網路只是整個 End-to-End 路徑的一個部分，因此核心網路內的資源規劃也要配合整個 End-to-End 路徑的需求。在 BBQ 中，負責 End-to-End 路徑規劃的元件為 Long Path Planning Agent(LPPA)，此元件位於每一個核心網路的 CNC 內，是核心網路的對外協調窗口。

在 End-to-End 路徑規劃階段時，各個核心網路之 LPPA 會先以每一個核心網路所可能提供的路徑選擇來為本身核心網路負責處理的訊務規劃最佳 End-to-End 路徑，所規劃的 End-to-End 路徑將作為網路營運者建置或調整網路架構之依據。

當核心網路內的完成資源規劃的程序後，每一個 Ingress 會擁有真正可以運用的資源，並以路徑的方式儲存在一個路徑的資料庫中，此路徑的資料庫將是 LPPA 在 Runtime 時可以運用來規劃 End-to-End 路徑的資源，圖 4.1 為此一個過程的流程圖：



4.5 核心網路內路徑規劃運作流程 (Path Planning Procedure)

在核心網路內的路徑規劃，分為兩個不同的階段，並由不同的元件所執行，第一個階段是由 SPPA 以預測的資料來做路徑規劃，而第二階段則由位於每一個 Ingress Router 中的 Ingress PPA 負責，依據實際所獲得的資源來做規劃，兩個階段的詳細流程如下。

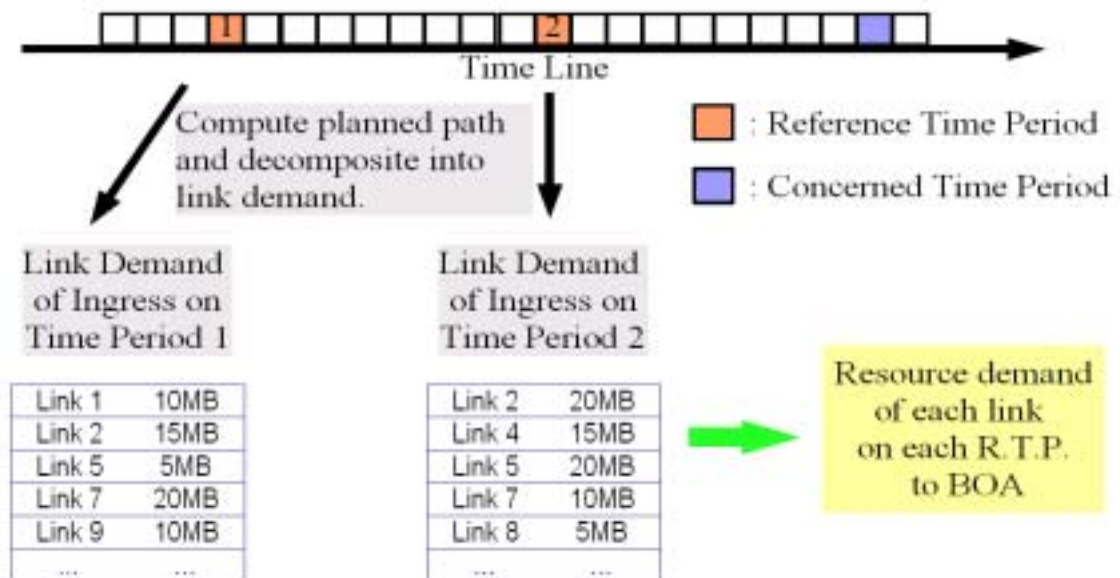
在第一階段中，SPPA 會依據兩個輸入來做路徑的規劃，一為過去的整個核心網路中某一參考時段的所出現的訊務流量與型態。此一資訊包含每一個 flow 的資訊，其中包含四個項目，訊務的來源、訊務的目的地、訊務所要求的頻寬和此一訊務所要求的服務品質參數(s, d, b, q)。第二個輸入是整個網域之中的拓樸結構和組成網域中每個鏈結上的資源資訊，包括其上可以使用的頻寬，以及所能保證的服務品質熵數。有了以上兩項參數，SPPA 會以本研究所提的路徑規劃方法，為整個核心網路某一個未來的時

段中可能出現的訊務流量規劃出一組路徑(Planned Path Set)，此一組路徑預計可以符合未來運作時段中的訊務需求。如果因為公平性的問題，或各 Edge Router 有特殊考量則可啟動第二階段分散式的資源規劃與分配。

4.5.1 路徑資源需求與鏈結頻寬需求之轉換

歷史資料中記錄流量需求的進入點、離開點、頻寬及品質需求，但資源規劃是以鏈結上的頻寬為單位，在第二階段由 BOA 向 Bandwidth Broker 批購鏈結頻寬資源時，必須將以路徑為基礎之流量需求轉換成以鏈結為基礎之流量需求。因此 SPPA 的主要任務即為規劃出一組路徑，將原始需求依照第一階段所求得之路徑分解成每一個鏈結上的頻寬需求，交給 BOA 作為批購頻寬的依據。

由於訊務的機率特性，BOA 在批購頻寬的時候，最好參考多次訊務的流量情形，以需求量的統計分佈作預測才可能降低預測誤差。因此我們以第一階段求得之路徑作為基礎，輸入歷次 Reference Time Period 的訊務需求量經過這些路徑，得到鏈結上的資源需求量之統計，此即為 BOA 計算最佳頻寬訂購量的基礎，以上為第二階段的分散式路徑規劃之準備工作。



當 BOA 將真正購得的頻寬交給在每一個 Ingress 中的 Local SPPA 元件後，則進入 Path Planning 的第二階段，由於所批購得到的資源可能不如預期或者有不同的規劃目標，位於各個 Ingress 中的 Local SPPA 元件必須利用所購得的資源來作第二次的路徑規劃，計算時仍然需要參考此一 Ingress 過去的訊務需求來規劃此一 Ingress 在系統運行時所真正可以利用的路徑組，不同於第一階段的路徑規劃所參考的是某一次 Reference Time Period 的路徑需求統計資料，在 Local SPPA 中採用的是過去幾次 Reference Time Period 的路徑需求平均值，做為路徑規劃的依據，所規劃得出的此組路徑具有頻寬和傳輸品質保證，然後將儲存在各 Ingress 中的路徑 Database 中，供 LPPA 作為規劃 End-to-End 路徑時選擇和系統運作時允入控制元件以此路徑資訊來對欲進入網域中的訊務做允入控制。

表 4.3：各階段路徑規劃所需之輸入與輸出參數

路徑規劃階段	輸入資訊	輸出資訊
第一階段： SPPA 執行	<ol style="list-style-type: none"> 1. 核心網路之拓樸和每個鏈結上的頻寬和傳輸服務品質參數 2. 整個核心網路內的歷史訊務統計 	一組最佳(或次佳)的路徑組，並會依路徑的出發點分類，然後分解成每一個 Ingress 在每一個鏈結上所需要的頻寬。
第二階段：由每一個 Ingress 中的 Local SPPA 元件所執行	<ol style="list-style-type: none"> 1. 在核心網路中每一個鏈結上所可以運用的頻寬資源和鏈結的傳輸服務品質參數。 2. 此一個 Ingress 在過去 Reference Time Period 的路徑需求訊務統計 	一組路徑組由許多不同目的地和傳輸服務品質參數的路徑所組成。

4.6 路徑規劃最佳化

路徑規劃的目的在於提供訊務一定的服務品質保證，充分利用網路的頻寬並追求營運者所定義之最大利益。在資源保留路徑規劃方法中，規劃所得之路徑將附有頻寬限制，換言之各鏈結之頻寬將分配並保留予規劃所得路徑。在系統運作時可以保證路徑所擁有頻寬，不會與其他經過相同鏈結的訊務競爭，一條鏈結上所有的訊務流量皆經過一定的規劃控制，而不會超出其上的傳輸能力。

但是要如何評量我們所規劃出來的路徑之好壞是一個重要的議題，因此在本研究中以網路營運者的角度來思考，為我們所規劃出的路徑來作一個衡量，一般說來一個網路營運者莫不希望在既有的網路設備上提供更多傳輸服務，因此若我們可能以一定的網路資源來服務更多的使用者，即表示網路營運者的獲益將會越高，但是每一個訊務所要求的頻寬和傳輸服務品質並不同，越高的傳輸品質代表的是系統保留更多的頻寬來服務訊務的需求。

因此，我們可以為要求進入核心網路中的訊務，依其資源需求定出一個值，此值越高則代表系統需要以更多的資源來滿足訊務的需求，但是同時也是表示若可以滿足的此訊務的需求，我們將可向此訊務收取較高的費用，因此當我們依整個核心網路中過去的訊務流量歷史統計資料規劃出一組路徑後，我們可以依此路徑組可以容許通過之所有訊務的價值加總，而整個絕對資源保留路徑規劃的最佳化目標，就是要盡量提

高此值。

4.7 最佳化模型 (System Model)

鑒於上述的因素，我們對於 Path Planning 的問題以 Integer Programming 方式呈現，並求最佳解，表 4.4 列出路徑規劃相關之符號說明。

表 4.4：資源分配路徑規劃符號表

Symbol	Definition
$G(V, E)$	A directed graph, G , containing $ V $ nodes and $ E $ directed links; V denotes the set of nodes, and E denotes the set of links
v	A node; $v \in V$
e_k	A directed link $e_k = (v_x, v_y) \in E$, v_x is the start node, and v_y is the end node of link e_k ; also denoted as e_{xy}
R	Incoming traffic set.
r_i	A request i , consist of (s_i, d_i, p_i, q_i) ; $r_i \in R$
s_i	Source node of request i
d_i	Destinaiton node of request i
b_i	Bandwidth requirement of request i
q_i	Quality entropy of request i
m_i	Profit earned for admitting request i
P_i	All feasible paths of request i
p_{ij}	A path; $p_{ij} \in P_i$
ω_i	Selected feasible path set by algorithm GPPA for request i
Ω	The path set selected by algorithm GPPA
$B_e(p_{ij})$	$= b_e(p_{ij})$, if links of p_{ij} contain link e $= 0$, otherwise
$L(p_{ij})$	The satisfied ratio of request i with p_{ij}
$q(e)$	Quality entropy of link e

路徑規劃問題描述如下：給定一組網路的拓樸 $G(V, E)$ 及 Request(R)，找出一組路徑 Ω 使得獲利最大。任一個 link e_i ，附有兩個屬性，一為此鏈結所能負擔的最大頻寬 b_i ，另一為此 link 所能提供的服務熵數 ε_i ；Requests (R)，由許多的 request 所組成，對於每一個 r_i 以下列四個參數來描述，起點(s_i)、終點(d_i)、所需頻寬(b_i)和我們定義的服

務品質熵數(q_i)。我們定義獲利 m_i 是允許某一個 request i 進入核心網路後所獲得的利益，獲利 m_i 是由 b_i 除以 q_i 所得(公式 4.1)，其中 b_i 為 request i 所要求之頻寬，而 q_i 為 request i 所要求之服務品質熵數。若一個 Request 所要求的品質(即頻寬 b_i 或服務品質熵數 q_i)越嚴苛，則其所獲得的利益越大，所以定義 m_i 與要求的頻寬成正比，並與服務品質參數 q 成反比。

$$m_i = \frac{b_i}{q_i} \quad (4.1)$$

當某一個訊務通過允入控制元件進入核心網路，即表示我們可以滿足其對於頻寬和服務品質的要求，所以路徑規劃的最佳化目標即是求得一組可以使我們獲得最大利益的路徑組。我們利用 Ω 來表示此一個路徑組，則路徑規劃的最佳化問題描述如下：對於一組給定的拓樸 G ， G 是由一組節點 V 和一組鏈結 E 所組成，另外給定一組 Request R ， R 是由 n 個 request 所組成，每一個 request 由四個項所表示起點 s 、終點 d 、此 request 的頻寬需求量 b 和此 request 的服務品質熵數 q 。以數學模型表示如下：

$$\begin{aligned} & \text{find } \Omega \\ & \max \sum_{i=1}^n m_i \sum_{p_{ij} \in \omega} L(p_{ij}) = \max \sum_{i=1}^n \frac{b_i}{q_i} \sum_{p_{ij} \in \omega_i} L(p_{ij}) \end{aligned} \quad (4.2)$$

such that

$$\sum_{e \in \Omega} q(e) \leq q_i, \text{ for all } p_{ij} \in \Omega \quad (4.3)$$

$$\sum_{p_{ij} \in \Omega} b_e(p_{ij}) \leq B(e), \text{ for all } e \in E \quad (4.4)$$

此一最佳化的函式的兩個限制式為不等式 4.3 和 4.4：

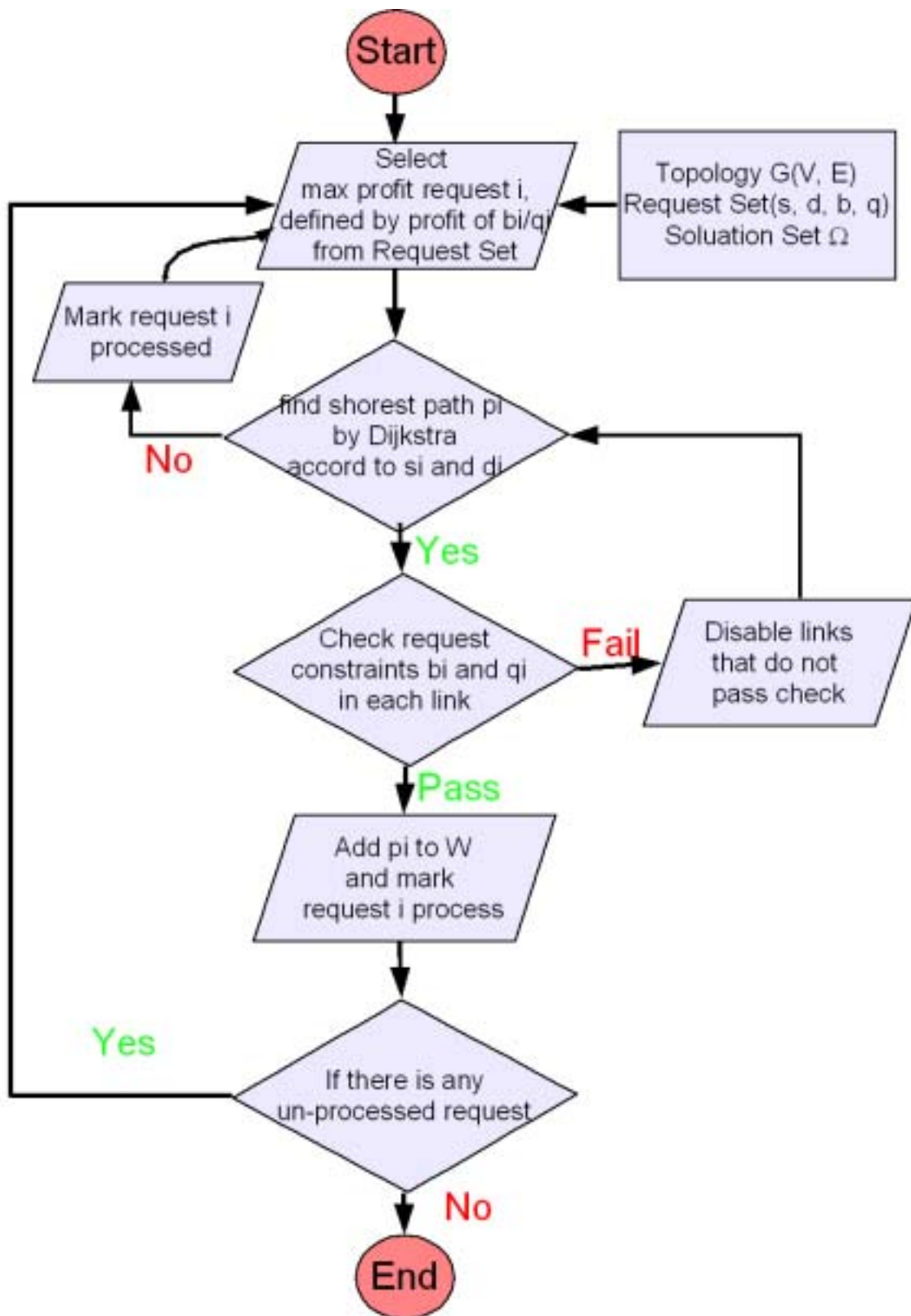
一、不等式 4.3 表示在所求出的路徑組中，每條路徑所使用的頻寬，不能超過組成網路的每一個 edge(e)所提供的總頻寬。

二、不等式 4.4 表示而對於某一個被滿足的 request r_i ，所挑選出的路徑 P_i ，其所帶 QoS 熵數，至少必須滿足此一個 request 的服務品質熵數的限制。

4.8 Greedy Algorithm for Path Planning(G.P.P.A.)

對於上述的問題，可得知路徑規劃的主要目的是找出一組最符合利益的路徑，而且組成此路徑組中的每一條路徑皆需要符合頻寬和品質兩項限制式。如第二章所提及的，在繞徑研究中，在一個網路拓樸中，找出一組符合一個以上限制式的問題，被稱為是 Multi-Constraints Path(MCP)的問題，而此為一個 NP-Complete 的問題[20]。而對於 NP-Complete 的問題我們無法找到快速的最佳解，我們提出一個 Heuristic 的演算法來解這個問題。此種方法是 greedy method，以下是演算法的步驟：

- 輸入：一個網路拓樸 $G(V, E)$ ，一個拓樸是由一組節點 V 和一組鏈結 E 所組成，和一組 Request Set R ， R 為 n 個 request 所組成，每一個 request 有四個部分，起點 s 、終點 d 、此 request 的頻寬需求 b 和此 request 的服務品質熵數 q 。
 - 輸出：一組符合最大利益的路徑組 Ω 。 Ω 由一組帶有頻寬限制和服務熵數的路徑所組成，此組路徑的獲利總和為最大。
1. 從 Request Set 中，選出 Profit 最大的一個 request i ，profit 由 b_i/q_i 計算而得。
 2. 利用 Dijkstra Algorithm 為 request i 的 s_i, d_i ，找出一條路徑，若我們無法找出一條路徑，則將此 request i 標示為已處理。
 3. 檢查由步驟二所找出的路徑中，其組成路徑的所有鏈結是否符合該 request i 之頻寬 b_i 和 quality entropy q_i 之限制。
 4. 若有鏈結 m 不符 request i 之 b_i 和 q_i 的限制，則暫時拿掉鏈結 m ，得到一個暫時的網路拓樸 $G'(V', E')$ ，回到步驟二；若此 path 合乎 request i 之限制，則將此 path 加入 W 中，並將 path 標示為已處理。
 5. 若還有未處理的 request 在 request set 中，則回到步驟一，直到所有 request 皆為已處理。



4.8.1 演算法虛擬碼 (Pseudo Code)

```
// Check Path Constraints
CheckPathConstraints( path  $p_i$  , request  $r_i$ )
{
    BWCheck <- TRUE;
    DECheck <- FALSE;

    TOTAL_DELAY <- 0;
    // Check link capacity
    for all  $e_k$  in  $p_i$  {
        if  $B(e_k) < B(r_i)$ 
            BWCheck <- FALSE;

        TOTAL_DELAY <- TOTAL_DELAY +  $D(e_k)$ ;
    }
    // Check Total Delay
    if TOTAL_DELAY <  $D(r_i)$ 
        DELAYCheck <- TRUE;

    if BWCheck and DELAYCheck
        return TRUE;
    else
        return FALSE;
}
```

```

PlanPlanning(G, R)
{
  for k <- 1 to |R|
  {
    // Process Request k
    P(k) <- NIL;
    process <- false;
    While (!processed){
      // Find shortest path for request k
      S(k) <- Dijkstra.shortest.path(G, rk);

      // Check Constraints
      if (P(k)i != NUL){
        Granted <- CheckPathConstraints(S(k)i, );
        if Granted
          P(k) <- P(k) ∪ S(k)i;
        else
          DisableUnqualifiedLink(G, S(k)i);
      }
      else
        processed <- TRUE;
    }
  }
}

```

4.8.2 演算法複雜度分析

在路徑規劃的演算法之中，我們根據歷史的訊務資料，對可能會進入網域內之訊務尋找一個符合其要求的路徑，因此需要對每一個訊務都進行一次最短路徑演算法，在本研究中，我們採用的最短路徑演算法為 Dijkstra's shortest path 演算法，因此對一組需有 M 個預測訊務的輸入和 N 個節點之拓樸的路徑規劃而言，其時間的複雜度為 $O(M * N^2 \log N)$ 。而在處理每一個訊務的過程中，我們以 shortest path 演算法所找出的路徑不一定可以符合訊務的需求，因此在每一個尋找的過程之中，可能需要執行超過一次以上的 shortest path 演算法，假設一次的處理訊務過程最多需要額外執行 k 次的 shortest path，則我們的時間複雜在最差的情形下，將會被限制在 $(M+k) * N^2 \log N$ 之下。

4.9 小結(Summary)

本章 4.1 節和 4.2 節介紹了核心網路中路徑規劃的問題和路徑規劃的元件，4.3 節中說明路徑規劃的環境假設，4.4 節和 4.5 節中，介紹核心網路中路徑規劃的程序，在 4.6 節中，我們將探討核心網路中路徑規劃的問題，本研究提出一個路徑規劃的最佳化模型則在 4.7 節和中說明，並在 4.8 節提出一個 Heuristic 的方法來解決路徑規劃的問題。接下來在第五章中，我們將以模擬的方式來測試此路徑規劃演算法和 OSPF 的效能比較。

第五章

效能評估 (Performance Evaluation)

本研究採用幾組模擬來評估我們所提出的路徑規劃解決方案之效能，並藉由觀察系統的獲利指標和獲利與鏈結使用率之比來瞭解演算法的效能。我們會以模擬實際網路的運作情形，比較本研究所提出的演算法和傳統的 OSPF 之差異，來說明 BBQ 架構的優點及其效能的改善。

5.1 效能評估指標 (Performance Evaluation Metrics)

以下是模擬實驗所觀察的效能指標。

1. 獲利指標：網路所允入的每一個訊務都帶有一定的獲利，藉由模擬一個參考時段所產生的訊務需求，觀察本路徑規劃演算法滿足訊務需求的能力，然後可由這些訊務所產生的利潤，來評量本演算法之好壞。
2. 鏈結使用率之標準差：藉由觀察鏈結使用率之標準差，可得知在拓樸中，每一個鏈結使用情形的差異。
3. 獲利與鏈結使用率之比：獲利除以平均的鏈結使用率即為獲利與鏈結使用率之比。由獲利與鏈結使用率之比我們可以得知，每單位鏈結使用率的獲利情形。

5.1.1 獲利指標 (Profit)

在路徑規劃演算法中，我們以訊務所要求的頻寬和服務品質作為評量訊務價值的方法，而一個訊務所要求的頻寬除以 Quaility Entropy 即為此訊務的價值，在我們的路徑規劃方法中，可以容許更多的訊務獲得服務即表示所得到的獲益越大，因此我們在改變不同的實驗變因時也觀察此一獲益指標來評估演算的效能。而在彈性的資源保留路徑規劃方案中，我們提出的以每一個鏈結的被使用率之標準差做為我們最佳化的目標，因此此一標準差指標也是我們在變化實驗變因時的觀察對象。

5.1.2 鏈結使用率之標準差 (Link Utilization S.D.)

鏈結的使用率，是由鏈結上的負載除以鏈結的能力所得，鏈結的使用率標準差代表各個鏈結使用情形的離散度，當鏈結使用率標準差的數值越大，代表各個鏈結的使用率差異越大。在同樣訊務負擔的情形下，一個路徑演算法使得鏈結使用率之標準差越大，可能的情形即為，以其演算法封包所走的路徑，將集中於幾個鏈結之上，而標準差值越小，則代表各鏈結的資源使用較平均。

5.1.3 獲利與鏈結使用率之比 (P.U. Ratio)

網路中的資源，即為每個鏈結上的頻寬，我們將總獲利除以鏈結使用率作一個觀察的指標，稱為獲利與鏈結使用率之比 (P.U. Ratio)。鏈結使用率計算如下：將計算單一鏈結上的被使用的頻寬除此鏈結的總頻寬，得到一個單一鏈結上的資源使用率，然後對模擬網路中所得鏈結使用率求平均，此為此組路徑所使用的鏈結使用率。因為若經過路徑規劃的結果，我們可以較少的頻寬，就能獲得較高的使用率，即每單位的鏈結頻寬獲益較高，則表示其方法較佳。

5.2 實驗設計 (Design of Experiments)

由於本研究主要是在 BBQ 的整體架構中，利用預先規劃路徑的方法來模擬整體的實驗架構，因此本研究所進行的實驗中除了觀察演算法的效能外，也利用一個簡單的模擬環境來比較經過路徑規劃和傳統沒有經過路徑規劃之網路效能。

5.2.1 實驗工具

本實驗以 FreeBSD 為作業平台，在其上以 GNU C++ 3.2 實作一個本文所提出之路徑規劃演算法和一個 Simplify BBQ Computational Environment。前者用來產生利用 PPA 元件所規劃的路徑，後者為一個簡化的 BBQ 架構 Computational Simulator，內含有簡單的 BBQ 功能元件，用來模擬 BBQ 架構的運作流程，但是為了簡化實驗流程，我們目前只有實作模擬系統運作時，訊務進入 Ingress 時所進行之允入控制和路徑指定的程序，以觀察我們路徑規劃的效能。

5.2.1.1 BBQ Computational Simulator

BBQ Computational Simulator 是本團隊以 C++ 所實作的一套模擬程式，主要的功能在於模擬網路系統的運作流程和簡單的 BBQ 功能，此一模擬系統以兩種繞徑模式進行模擬，一為傳統的 OSPF 演算法，在此種模式之中，進來的訊務會依照 OSPF 的演算法來進行繞徑，另一種則會依照 BBQ 的運作程序，將欲進入系統內的訊務，經由此 Simulator 的允入控制功能，從 Ingress 的路徑資料庫中尋找一條符合訊務需求的路徑並指定給此一訊務，此一 Simulator 會詳細記錄每一個訊務所獲得的獲利及頻寬的使用，以分析整理。

5.2.2 實驗測試組產生方式

實驗組的產生包含網路拓樸和訊務兩部分。在網路型態方面，其設定的參數包括，節點數量、節點的鏈結率、鏈結上的頻寬、鏈結的傳輸延遲設定。而對於訊務的設定參數包括訊務進入點、離開點、需求的頻寬和所需要的服務品質參數。各參數的設定數值及範圍如表 5.1 所示。

5.2.2.1 網路拓樸之產生與設定

網路節點的數目範圍從 10 到 50，而節點的組成分為 Edge Node 和 Core Node，訊務會從某個 Edge Node 出發，並從其他 Edge 離開，而 Core Node 只有轉送封包的功能，Edge 和 Core 的比例為一比一。而一個節點鏈結率的定義如下，假設一個拓樸內有 N 個節點，若一個節點與其他 $N-1$ 個節點皆有鏈結連接，則此點的鏈結率定義為 100%，因此一個節點的連結率為 $L / (N-1)$ 。

而實驗變因所指之連結率，則是將實驗網路拓樸中的每一個節點其鏈結率求平均值的結果，我們設定的鏈結率從 20% 至 100%。

5.2.2.2 訊務之產生與設定

對於路徑規劃所需的歷史訊務資料，我們假設其是以一個集合的方式呈現，每一個節點可能都有一組訊務流從此節點傳遞至其他網路內的任何一個節點。

本實驗中所產生的每一個 request 所要求的頻寬則是假設呈 normal 分佈，而每一組 request 所要求的總頻寬約為一個固定值。

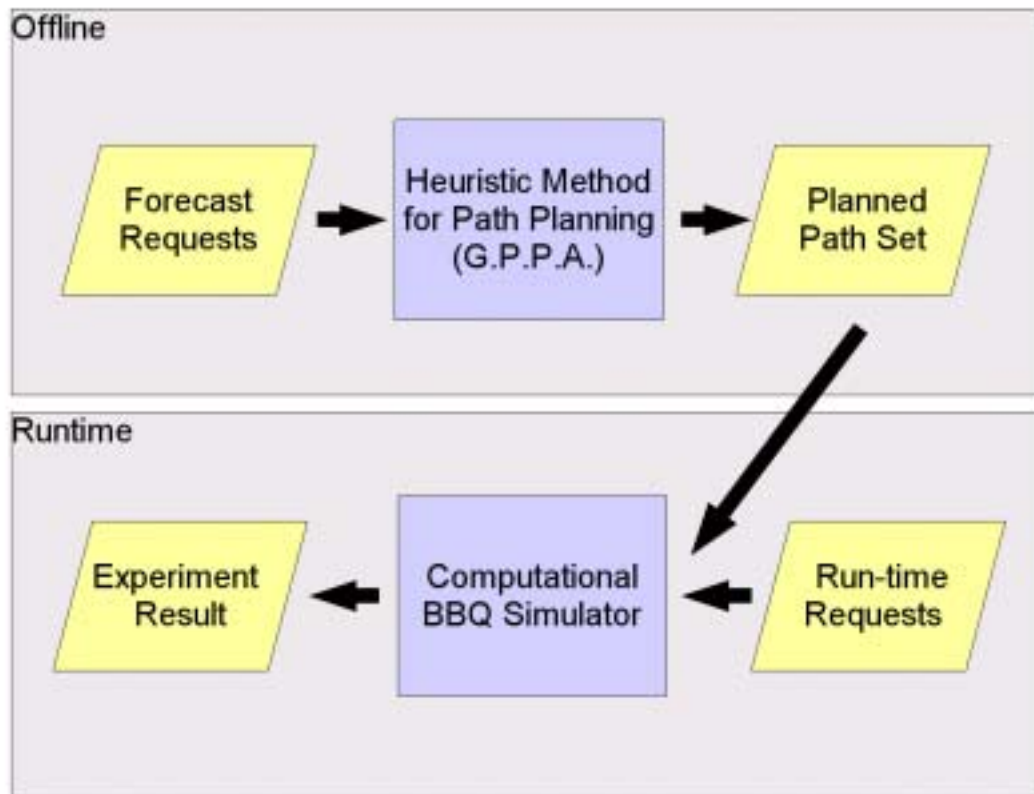
表 5.1：實驗組參數設定及其範圍

參數	數值範圍
節點數目	10, 20, ... , 50
Edge 節點和 Core 節點之比	1:1
節點鏈結率	20%, 40%, 60%, 80%, 100%
鏈結頻寬	1G bps
鏈結延遲	5 ~ 25 ms
訊務所需頻寬	0 ~ 100 Mbps
訊務之傳送延遲限制	20 ~ 100 ms

5.2.3 實驗流程

首先將此一個 Reference Time Period 的一組歷史訊務統計資料，及一組實驗的網路拓樸，輸入路徑規劃的演算法中，經過計算得到一組規劃路徑，然後利用 NS2 產生 Run-time 訊務流。然後，將此上述的三個資料輸入 BBQ Real-Time Simulator 之中，模擬實際的 BBQ 的運作流程，藉以觀察此演算法之效能。

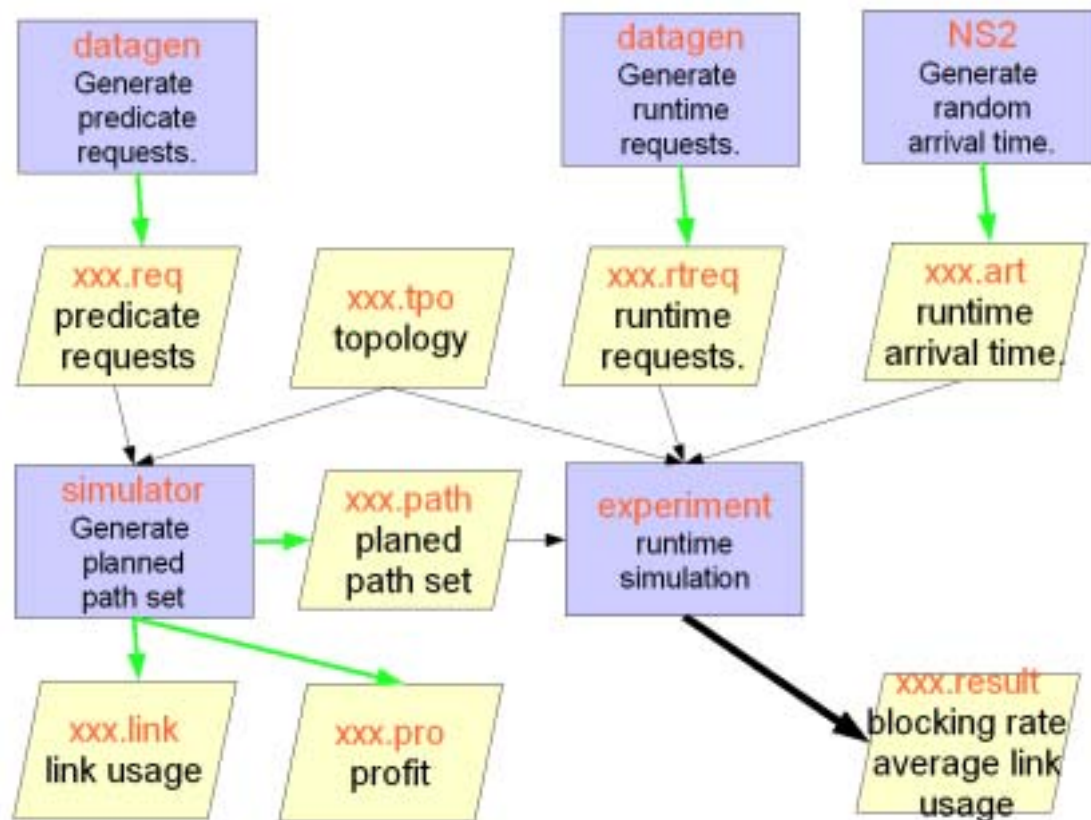
本實驗的對照組是為未經訊務預測和路徑規劃的 OSPF 演算法，藉以比較經過路徑規劃程序和無路徑規劃方法的優劣。



5.2.4 實驗組

以下是本研究所進行之實驗：

- 實驗一的目的在於關於本演算法的行為，我們以一個小型的網路為環境，觀察本演算法與最佳解之差距。
- 實驗二的目的在了解網路的節點數對於本演算法效率的影響。
- 實驗三的目的在了解網路的鏈結率對於本演算法效率的影響。



- 實驗四的目的在了解預測誤差對於本演算法效率的影響。

表 5.2：效能評估實驗

實驗	目標
實驗一：演算法行為測試	1. 觀察本演算法之行為 2. 衡量本演算法之效能與最佳化結果之差距
實驗二：網路節點多寡影響測試	以二個指標來觀察本演算法之效能 1. 獲利 (Profit) 2. 鏈結使用率標準差(Link Utilization S.D.) 3. 獲利與鏈結使用率之比 (P.U. Ratio)
實驗三：網路鏈結率影響測試	
實驗四：預測誤差影響測試	

5.3 模擬過程及結果分析

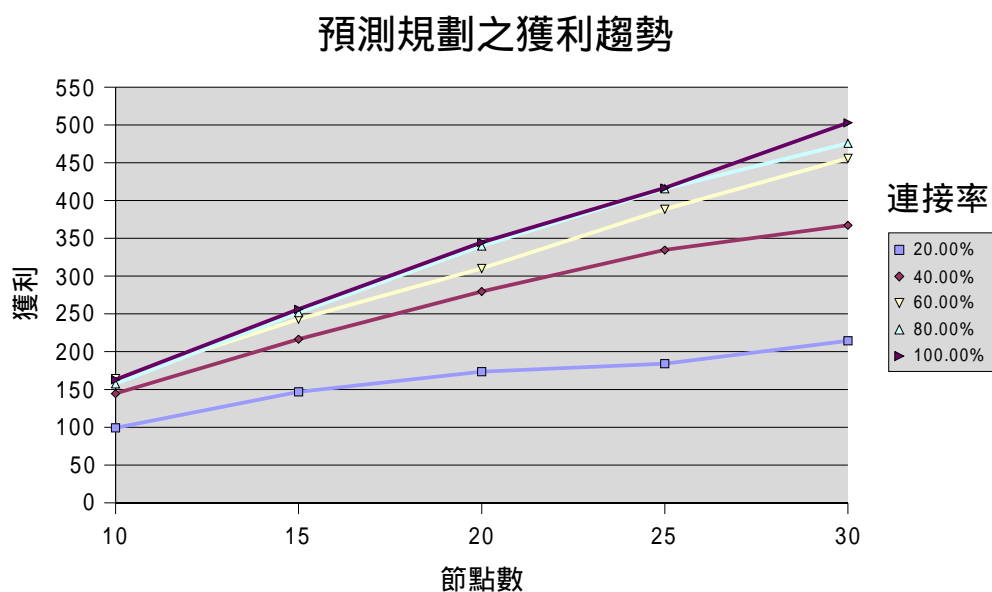
5.3.1 實驗一：演算法的行為測試

在實驗一中，我們以路徑規劃時所獲得的利潤和利潤密度來觀察演算法的行為。

5.3.1.1 獲利趨勢

本實驗以拓樸中的節點數量為實驗變因，探討當固定網路鏈結率時，拓樸中的節點數量對獲利之影響。同時，測試當不相同的網路鏈結率，觀察是否有相同的趨勢。

圖 5.3 中，橫軸為節點的數量，直軸為獲利，由圖中可看出，當節點增加時，因為可以選擇的路徑增加，使得獲利也有提升。而當鏈結率的增加時，代表鏈結數目增加，同時也會增加可供選擇的繞徑因此也會提高獲利。而在網路鏈結率提升至在 80% 至 100% 時，可以發現，兩條線的趨勢幾乎是一樣的，這是因為進入的訊務數量並未跟著提昇，所以獲利並未再增加。

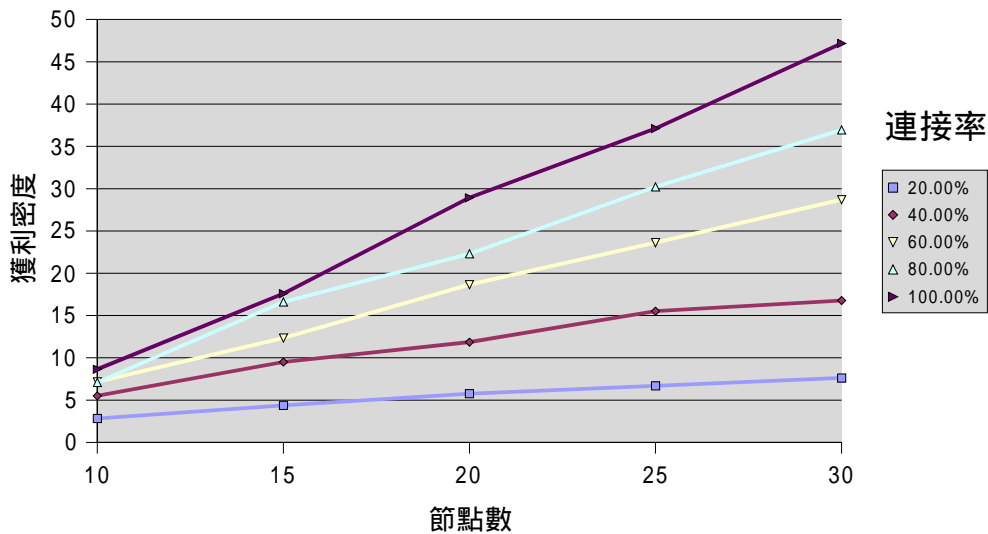


5.3.1.2 獲利與鏈結使用率之比趨勢

如同上，本實驗以拓樸中的節點數量為實驗變因，探討當固定網路鏈結率時之影響，不同的是，我們將觀察拓樸中的節點數量對獲利與鏈結使用率之比之影響。同時，測試在不相同的網路鏈結率下是否有相同的趨勢。

在圖 5.4 中，在我們所觀察的是與預測規劃之利潤密度趨勢，由圖中可以看出，當節點數增加時，獲利會隨之上昇。而當鏈結率增加時，獲利也會隨之上昇。

預測規劃之利潤密度趨勢



5.3.2 實驗二：節點多寡之影響 (Sensitivity to Number of Nodes)

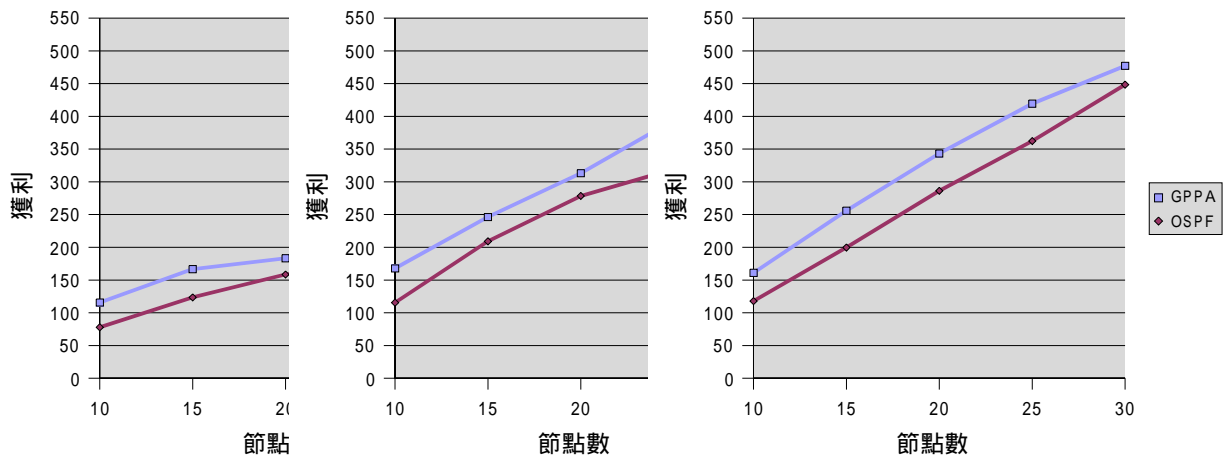
在此實驗中我們將觀察一個網路拓樸中節點的多寡對於演算法效能的影響。在 BBQ 所假設的網路底層架構中，是以 DiffServ 為底層的網路架構，而在 DiffServ 中，網路中的節點分為邊緣節點(Edge Router)和中間轉送的節點(Core Router)，邊緣節點是訊務進入或離開網路的節點，所以一個訊務在描述訊務時所使用的進入點和離開點，皆為網路中的邊緣節點。在本實驗中所設定的網路拓樸其 Core 和 Edge 節點的比例為一比一。

5.3.2.1 拓樸中節點多寡對於獲利指標的影響

本實驗以網路拓樸節點的多寡為實驗變因，探討當網路拓樸節點的多寡變化時對獲利之影響，本實驗設置了一個對照組為傳統的 OSPF 繞徑演算法，實驗過程中會將此兩套演算法透過 BBQ Computational Simulator 進行模擬。本實驗並同時測試當不相同的鏈結率時，觀察是否有相同的趨勢。

在圖 5.5(A)和圖 5.5(B)，其橫軸為節點數目的變化，直軸為獲利的變化，在圖 5.5(A)為鏈結率固定為 20%時節點數目變化時，獲利情形的變化，圖 5.5(B)為鏈結率固

定為 40% 時的變化情形。我們可以從所獲得的利率來看使用本演算法在經過 BBQ Computational Simulator 的模擬運作之後，可以發現在節點數增加後，所獲得的獲利比 OSPF 高出許多。



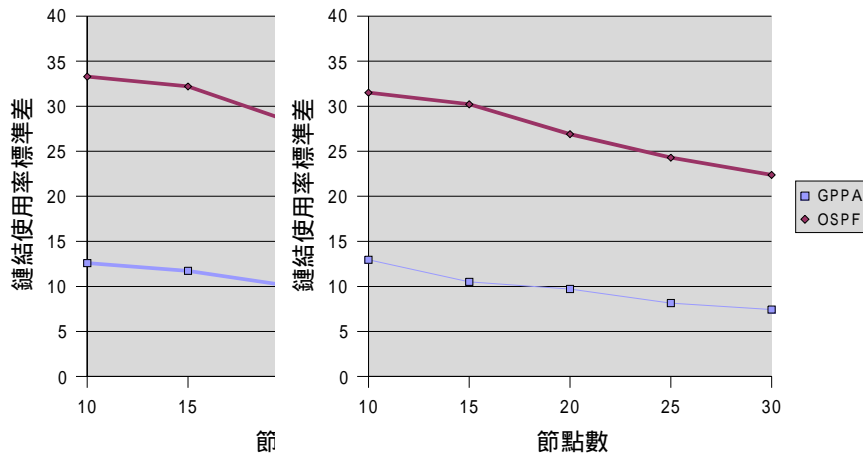
- (C) 鏈結率為 60%
- (D) 鏈結率為 80%

圖 5.5：節點多寡對於獲利指標的影響

5.3.2.2 拓樸中節點多寡對於鏈結使用率標準差的影響

本實驗依然以網路拓樸節點的多寡為實驗變因，不過觀察的是當網路拓樸節點的多寡變化時對鏈結使用率標準差之影響，本實驗也設置了一個對照組為傳統的 OSPF 繞徑演算法，實驗過程中會將此兩套演算法透過 BBQ Computational Simulator 進行模擬。本實驗並同時測試當不相同的鏈結率時，觀察是否有相同的趨勢。

在圖 5.6(A)和圖 5.6(B)，其橫軸為節點數目的變化，直軸為鏈結使用率標準差的變化，在圖 5.6(A)為鏈結率固定為 20%時節點數目變化時，鏈結使用率標準差的變化，圖 5.6(B)為鏈結率固定為 40%時的變化情形，圖 5.6(C)和圖 5.6(D)則分別為鏈結率 60% 和 80%的情形。我們可以從所獲得的利率來看使用本演算法在經過 BBQ Computational Simulator 的模擬運作之後，可以發現在節點數增加後，所獲得的獲利比 OSPF 高出許多。



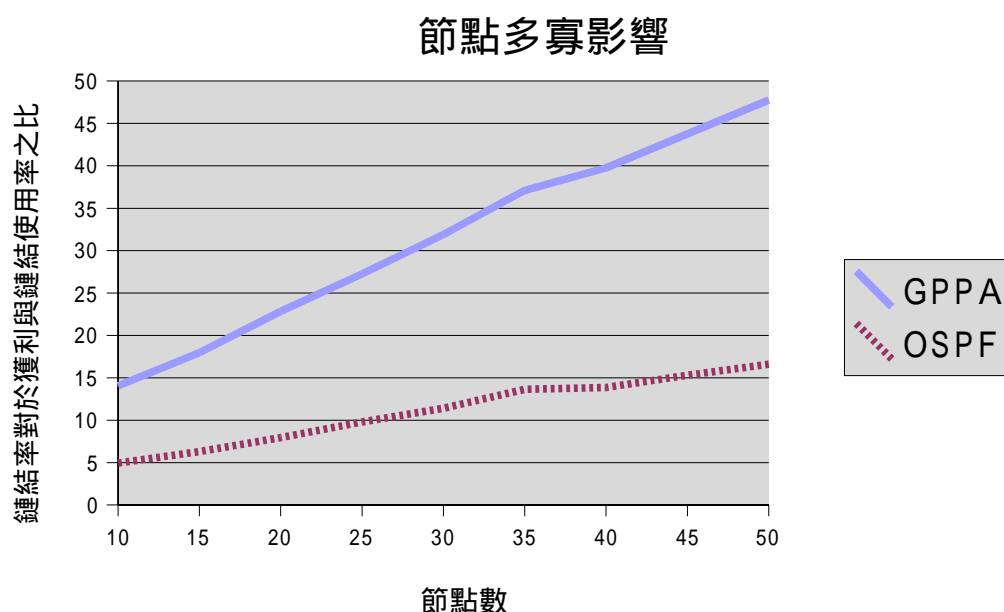
- (C) 鏈結率為 60%
- (D) 鏈結率為 80%

圖 5.6：節點多寡對於鏈結使用率標準差的影響

5.3.2.3 拓樸中節點多寡對於獲利與鏈結使用率之比的影響

在圖 5.7，我們觀察的是節點數目增加時，對於獲利與鏈結使用率之比的影響，當網路節點增加時，代表我們繞徑演算時所可以使用的節點越多，則可以規劃出的路徑也會相對的較多。由圖可以看出，經過路徑規劃後的獲利與鏈結使用率之比將比 OSPF 為佳，即表示本路徑規劃方法之獲利和網路使用率之比較高，本方法可以較低的網路

使用率來獲得與 OSPF 相同的獲利。



5.3.3 實驗三：鏈結率之影響 (Sensitivity to Connectivity)

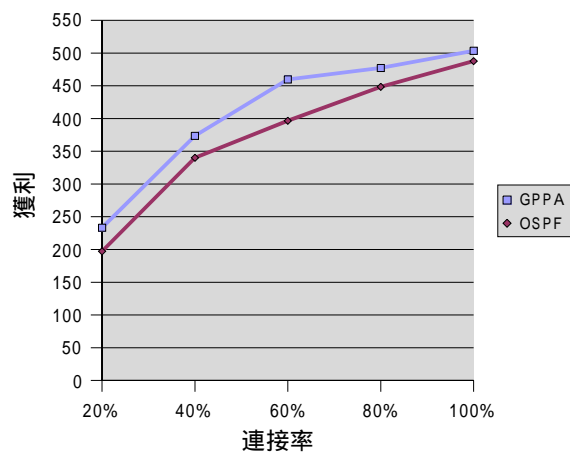
在此實驗中我們將觀察一個網路鏈結率大小對於演算法效能的影響。

5.3.3.1 鏈結率對於獲利指標的影響

本實驗以網路連結率之大小為實驗變因，探討當網路連結率之大小變化時對於獲利之影響，本實驗設置了一個對照組為傳統的 OSPF 繞徑演算法，實驗過程中會將此兩套演算法透過 BBQ Computational Simulator 進行模擬。本實驗並同時測試當不相同的節點數時，觀察是否有相同的趨勢。

在圖 5.8(A)和圖 5.8(B)，其橫軸為網路連結率的變化，直軸為獲利大小，在圖 5.8(A)為網路節點數目固定為 20 時網路連結率改變時，獲利情形之變化，圖 5.8(B)為網路節點數目固定為 25 時的變化情形。鏈結率的增加表示鏈結數目增加，而路徑計算時中可行的路徑也是相對增加。我們可以從獲利的情形來看使用本演算法在經過 BBQ Computational Simulator 的模擬運作之後，在圖 5.8 中，我們可以看出，在較低的

鏈結率時，OSPF 與本演算法的效能相差不多，但在鏈結率增加時，本方法獲利成長的速度是遠超過 OSPF。



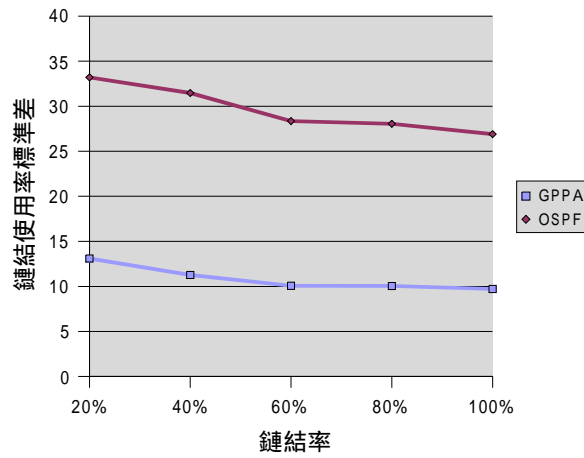
- (C) 網路節點數目為 25
- (D) 網路節點數目為 30

圖 5.8：鏈結率對於獲利指標的影響

5.3.3.2 鏈結率對於鏈結使用率標準差的影響

本實驗仍以網路連結率之大小為實驗變因，探討當網路連結率之大小變化時對於鏈結使用率標準差之影響，本實驗也設置了一個對照組為傳統的 OSPF 繞徑演算法，實驗過程中會將此兩套演算法透過 BBQ Computational Simulator 進行模擬。本實驗並同時測試當不相同的節點數時，觀察是否有相同的趨勢。

在圖 5.9 中其橫軸為網路連結率之大小，直軸為鏈結使用率標準差的變化，其中圖 5.9(A)為網路節點數目固定為 15 時網路連結率改變時，鏈結使用率標準差之變化，圖 5.9(B)為網路節點數目固定為 20 時的變化情形，而圖 5.9(C)和圖 5.9(D)則分別為節點數目為 25 和 30 時的情形。鏈結率的增加表示鏈結數目增加，而路徑計算時中可行的路徑也是相對增加。我們可以從鏈結使用率標準差來看使用本演算法在經過 BBQ Computational Simulator 的模擬運作之後，在圖 5.9 中，我們可以看出，在鏈結率增加時，GPPA 演算法和 OSPF 的標準差都會下降，因為可以使用的連結增加了，不過 GPPA 的標準差都與 OSPF 有一段差距，顯示 GPPA 在對於相同數量的訊務輸入時，對於網路拓樸中每個鏈結的使用率較為平均。



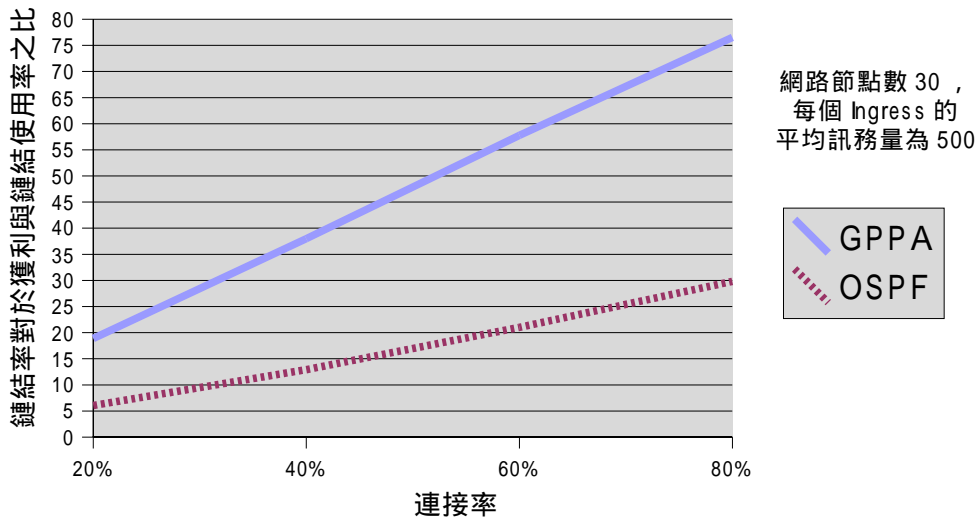
- (C) 網路節點數目為 25
- (D) 網路節點數目為 30

圖 5.9：鏈結率對於鏈結使用率標準差的影響

5.3.3.3 鏈結率對於獲利與鏈結使用率之比的影響

在圖 5.10 中其橫軸為鏈結率的變化，直軸為獲利與鏈結使用率之比的變化，從圖中可以觀察到在獲利與鏈結使用率之比上，本方法在鏈結率增加後，其比值增加的幅度是遠超的 OSPF，也代表路徑規劃演算法在連結率增加後，有較佳的網路利用率。

連接率影響



5.3.4 實驗四：預測誤差之影響 (Sensitivity to Forecast Error)

5.3.4.1 預測誤差之定義

由於 BBQ 中的路徑規劃方法，是以過去的歷史資料為基礎，對於未來可能出現的訊務作預先的路徑規劃，但是在系統運行的時候，會出現的訊務需求將會有很大的不確定性，因此必然會產生一定的預測誤差出現，實驗三的目的在測試當預測誤差出現時，對於本方法效能之影響。

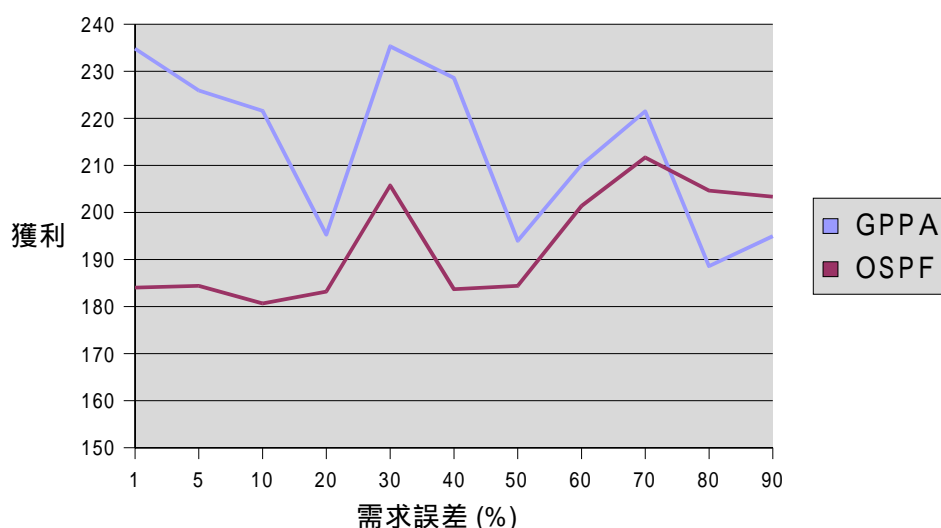
我們以如下的方法來定義預測誤差的產生，若過去的歷史資料和系統運行時所進入網路中的訊務完全一樣的話，則預測誤差為零，若在系統運行時，欲進入網域之總訊務數量中的 10% 統計資料，和歷史的資料不合，則我們定義預測誤差為 10%，而這些欲進入網路中的訊務所要求的頻寬總和則會維持在一定範圍。

5.3.4.2 預測誤差對於獲利指標的影響

本實驗以預測誤差之大小為實驗變因，探討當預測誤差之大小變化時對於獲利之影響，本實驗也設置了一個對照組為傳統的 OSPF 繞徑演算法，實驗過程中會將此兩套演算法透過 BBQ Computational Simulator 進行模擬。

在圖 5.11，橫軸為需求預測誤差，直軸為獲利大小。如圖所示在此實驗中，我們可以看出，當預測誤差在較小的時候，本方法仍較 OSPF 為佳，但當誤差漸漸拉大時，其影響開始出現，本方法所獲得的利益已被 OSPF 趕過，但是這情形是發生在預測誤差已在 80%、90% 以上時所產生的情形。

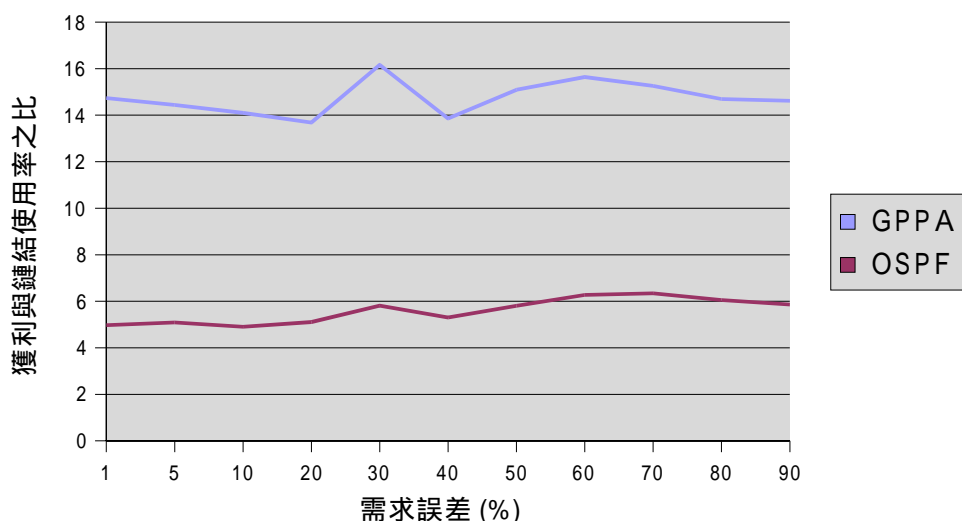
推測其可能的成因為，當預測誤差增加時，我們所事先規劃的路徑皆和系統運行時的訊務需求不合，因此，造成無法滿足訊務的實際需求，且會浪費的那先已經預先配置的頻寬資源，而 OSPF 採用即時的路徑規劃，因此其獲利會大概的維持一定的水準，但是在欲進入網路中的訊務個數不變時獲利並無法提升，但是本演算法的獲利卻會逐漸下降，因此造成兩者的獲利逐漸接近，甚至出現 GPA 演算法被超過的情形，因此我們需要有預測誤差的彌補方案來減低所造成的影響。



5.3.4.3 預測誤差對於獲利與鏈結使用率之比的影響

本實驗仍以預測誤差之大小為實驗變因，探討當預測誤差之大小變化時對於獲利與鏈結使用率之比之影響，本實驗也設置了一個對照組為傳統的 OSPF 繞徑演算法，實驗過程中會將此兩套演算法透過 BBQ Computational Simulator 進行模擬。

在圖 5.12 中，橫軸為需求預測誤差，直軸為獲利與鏈結使用率之比大小。如圖所示，預測誤差雖然使得獲得的利益可能漸漸不如 OSPF，可是就網路的使用率來說，本方法仍比 OSPF 為佳。推測其原因可能有二，其一為本演算法的將會較為分散訊務的路徑，使得鏈結的平均使用率較低，而在獲利相差不多的情形下，其獲利與鏈結使用率之比自然較高。而另一個原因為在預測誤差增加時，本演算法所能允入的訊務數較少，導致獲利下降，但是其鏈結的使用率也跟者下降，因此對於獲利與鏈結使用率之比而言並未早成太大的變化。



5.4 小結

在此章中，我們以一個 BBQ Computational Simulator 來對我們的演算法作效能測試，並和傳統的 OSPF 作比較，我們以系統獲利、鏈結使用率之標準差及獲利與鏈結使用率之比作為效能評量的指標，並觀察節點多寡，鏈結率和預測誤差三種變因對於本方法之影響。

在實驗一中，我們觀察 GPPA 路徑規劃演算法在預先規劃時，拓樸中的節點數目

多寡及拓樸的鏈結率對於系統獲利的影響，結果顯示，當節點數目及鏈結率增加時，皆會增加其獲利，其原因是因為當拓樸中可以使用的節點數及鏈結增加時，演算法所提供的路徑也增加，可以滿足更多的訊務需求，因此獲利增加。

在實驗二中，我們變化拓樸中的節點數目，來觀察節點數目對於本演算法效能的影響並和 OSPF 作比較，結果顯示在獲利方面，GPPA 演算法在節點數目增加時的獲利比 OSPF 高出許多，且在鏈結使用率標準差上，可以發現 GPPA 演算法會較分散訊務的對於鏈結的使用量，使得此值較 OSPF 為低。而在獲利與鏈結使用率之比上，也顯示 GPPA 在每單位的鏈結使用量的獲利會比 OSPF 為佳。

在實驗三中，我們變化拓樸中的鏈結率的大小，來觀察節點數目對於本演算法效能的影響並和 OSPF 作比較，結果顯示和實驗二中的結果大致相同，原因應該跟節點數目變化時相同，因此鏈結率的大小變化，跟節點數目的多寡對於本演算的影響是大致相同的。

在實驗四中，我們探討預測誤差對於本演算法的影響，其結果顯示，當預測誤差增加時，本演算法的弱點將顯現出來，就獲利看來，本演算法會因預測誤差的逐漸增加而導致獲利逐漸下降，甚至低於 OSPF。

從以上四個實驗中，我們可以歸納出，GPPA 的演算法的確要 OSPF 為佳，可以增加系統的獲利，但是在預測出現時，則會讓本演算法的效能退步，因此對於若要使用本演算法則要對可能出現的預測誤差有所因應。

第六章

總結 (Concluding Remarks)

本研究提出了一個具有 End-to-End 服務品質保證的網路架構，以分層管理和與預算分配，和資源的預先規劃來減輕系統即時運算的負擔達成 End-to-End 服務品質保證的目的。而在本文中，將研究重點放在核心網路上的路徑規劃上，路徑規劃是 BBQ 架構中核心網路達成絕對的服務品質保證之方法，本文在說明了分析了網路服務品質保證之重要性後，在相關研究中，對於可能提供 QoS 保證的網路架構作了一些探討，和研究現在提供 QoS 保證的一些繞徑演算法，然後在 BBQ 的整體架構下提出路徑規劃方式，來對通過核心網路中的訊務作絕對的 QoS 保證，其後我們對於路徑規劃的問題提出一個最佳化模型，並以一個 Heuristic 的演算法來解決路徑規劃的問題，透過實驗模擬的結果顯示，路徑規劃的確較傳統的 OSPF 為佳，對於系統營運業者來說，以較少的系統使用率就可能得到比傳統繞徑方法更佳獲利。

但是本路徑規劃的方法仍有其不足之處，如硬性保留的不可變動性，導致的結果就是可能資源嚴重的被浪費，當某一個 Ingress 對於資源分配不滿意的時候，並沒法在 Run-time 作變更，缺少調配彈性，而雖然我們提出了幾種彌補的方案希望可以減低資源浪費的程度，可是卻也增加系統的複雜度，如分散式的資源調配方案中，就需要多個元件互相溝通，才能達到系統所需之功能，對於實際的營運業者來說，無疑增加了佈建網路和錯誤檢修的困難度，而路徑規劃時只有考慮獲利一項因素也可能導致部分 Ingress 無法獲得所需之資源，因此在本研究未來展望中，希望可以在提出一個更精緻的路徑規劃方法，改善上述本研究所提出的路徑規劃之缺點，並以較為逼近真實網路的模擬環境來作為實驗模擬對象，評估路徑規劃的可行性，作為系統營運業者佈建網路的參考。

參考文獻 (References)

- [1] 3rd Generation Partnership Project, "Technical Specification Group Services and Systems Aspects: Architecture for an All IP network", 3GPP TR 23.922 version 1.0.0., October 1999.
- [2] Xiao, X., L. -M. Ni, "Internet QoS: A Big Picture", *IEEE Network*, 13(2):8-18, March-April 1999.
- [3] Miras, D., "Network QoS Needs of Advanced Internet Applications", *Internet2 - QoS Working Group*, November 2002.
- [4] Pascal Lorenz, "Quality of service and new architectures for future telecommunications networks", *MILCOM 2000 - IEEE Military Communications Conference, no.1*, October 2000 pp.695-698.
- [5] D. Goderis, S. Van den Bosch, Y. T'Joens, P. Georgatsos, D. Griffin, G. Pavlou, P. Trimintzios, G. Memenios, E. Mykoniati, C. Jacquenet, "A service-centric IP quality of service architecture for next generation networks", *NOMS 2002 - IEEE/IFIP Network Operations and Management Symposium, no.1*, April 2002 pp. 139-154.
- [6] Mahbulul Alam, Ramjee Prasad, John R. Farserotu, "Quality of service among IP-based heterogeneous networks", *IEEE Personal Communications, no.6, December 2001* pp.18-24.
- [7] Vijay K. Garg, Oliver T. W. Yu, "Integrated QoS support in 3G UMTS networks", *WCNC 2000 - IEEE Wireless Communications and Networking Conference, no.1*, September 2000 pp.1187-1192.
- [8] E. Crawley, Editor, L. Berger, S. Berson, "A Framework for Integrated Services and RSVP over ATM", *RFC 2382*, August 1998.
- [9] D. Black, M. Carlson, E. Davies, Z. Wang, "An Architecture for Differentiated Services", *RFC 2475*, December 1998.
- [10] Christian Huitema, "Routing in the Internet", *Prentice Hall PTR*, 2000, pp.356.
- [11] Spiridon Bakiras and Victor O.K. Li, "Efficient Resource Management for End-to-End QoS Guarantees in DiffServ Networks", *IEEE International Conference on Communications*, 2002.
- [12] Jacobson, V., K. Nichols, K. Poduri, "An Expedited Forwarding PHB", *RFC 2598*, June 1999.
- [13] Heinanen, J., F. Baker, W. Weiss, and J. Wroclawski, "Assured Forwarding PHB Group", *RFC 2597*, June 1999.

- [14] Clark, D., W. Fang, "Explicit Allocation of Best Effort packet Delivery Service", *IEEE/ACM Transactions on Networking*, 6(4):364-373, August 1998.
- [15] Tetsuya Takine et al., "Performance Evaluation of the Architecture for end-to-end Quality-of-Service Provisioning", *IEEE Communications Magazine*, April 2000.
- [16] P. Trimintzios et al., "A Management and Control Architecture for Providing IP Differentiated Services in MPLS-based Networks," *IEEE Commun. Mag.*, vol. 39, no. 5, May 2001, pp. 80-88.
- [17] P. Trimintzios et al., "A Policy-Based Quality of Service Management System for IP DiffServ Networks," *IEEE Network.*, vol. 16, no. 2, Mar 2002, pp. 50-56.
- [18] Eleni Mykoniati et al., "Admission Control for Providing QoS in DiffServ IP Networks: The TEQUILA Approach," *IEEE Commun. Mag.* Jan 2003, pp. 38-44.
- [19] G. Feng, K. Makki, N. Pissinou, C. Douligeris, "An efficient heuristic for delay-cost-constrained QoS routing," *IEEE International Conference on Communications*, 2001, ICC 2001, vol. 8, pp. 2603-2607.
- [20] Q. Ma and P. Steenkiste, "On path selection for traffic with bandwidth guarantees," *In Proceedings of IEEE International Conference on Network Protocols*, Atlanta, GA, October 1997.
- [21] Turgay Korkmaz, and Marwan Krunz, "Multi-Constrained Optimal Path Selection," *IEEE INFOCOM 2001*, pp. 834-843.
- [22] Yao-Nan Lien, Chien-Tung Chen, "Budget-Based End-to-End QoS Management for All-IP Networks", *Master Thesis*, September 2003.
- [23] Yao-Nan Lien, Ming-Chin Chen, "Distributed Resource Management and Admission Control in Budget-Based QoS Management for All-IP Core Networks ", *Master Thesis*, September 2003.
- [24] Yao-Nan Lien, Yi-Ming Chen, "Forecasting Error Tolerable Resource Allocation in Budget-Based QoS Management for All-IP Core Networks", *Master Thesis*, September 2003.
- [25] S. Floyd, and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance", *IEEE/ACM Transactions on Networking*, vol. 1, no. 4, August 1993, pp. 397-413.
- [26] Nicolas Christin and Jörg Liebeherr, "A QoS Architecture for Quantitative Service Differentiation ", *IEEE Communications Magazine*, June 2003.
- [27] A. Demers, S. Keshav and S. Shenker, "Design and Analysis of a Fair Queueing Algorithm", *Proc. SIGCOMM'89, ACM*, September 1989, pp. 1-12.
- [28] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell and J. McManus, "Requirements for

- Traffic Engineering Over MPLS", *RFC 2702*, September 1999.
- [29] D. Ooms, B. Sales, W. Livens, A. Acharya, F. Griffoul and F. Ansari, "Overview of IP Multicast in a Multi-Protocol Label Switching (MPLS) Environment", *RFC 3553*, August 2002.
- [30] E. Rosen and Y. Rekhter, "BGP/MPLS VPN", *RFC 2547*, March 1999.
- [31] A. S. Tanenbaum, "Computer Networks, Third Edition", *Prentice Hall*, March 1996, pp. 345-366.
- [32] Dijkstra, E.W., "A Note on Two Problems in Connection with Graphs", *Numerische Math*, vol. 1, March 1959, pp. 269-271.
- [33] C. Hedrick, "Routing Information Protocol", *RFC 1058*, June 1988.
- [34] J. Moy, "OSPF version 2", *RFC 1583*, March 1994.
- [35] Christophe Beaujean, "Delay-Based Routing Issues in IP Networks", *contact GRADIENT CR/98/148*, May 2000.
- [36] Douglas S.Reeves and Hussein F. Salama, "A Distributed Algorithm for Delay-Constrained Unicast Routing", *IEEE Transaction on Network*, April 2000.
- [37] R. Braden, L. Zhang, S. Berson, S. Herzog and S. Jamin, "Resource Reservation Protocol (RSVP) – Version 1 Functional Specification", *RFC 2205*, September 1997.
- [38] K. Chan, R. Sahita, S. Hahn and K. McCloghrie, "Differentiated Services Quality of Service Policy Information Base ", *RFC 3317*, March 2003.
- [39] Bill Goodman, "Internet Telephony and Modem Delay", *IEEE Network*, May 1999, pp. 8-16.
- [40] J. Garcia-Luna-Aceves and J. Behrens, "Distributed scalable routing based on vectors of link states", *IEEE J. Select on Communication*, October 1995.
- [41] Jon Postel, "Internet Protocol", *RFC 791*, September 1981.
- [42] Mark A. Sportack, "IP Routing Fundamentals", *Cisco ISBN: I-57870-071-x*, May 1999.
- [43] R. Wideyono, "The Design and Evaluation of Routing Algorithms for Real-Time Channels", *International Computer Science Institute, Univ. of California at Berkeley, Tech Rep. ISCI TR-94-024*, June 1994.
- [44] S. Rampal and D. Reeves, "An evaluation of routing and admission control algorithms for multimedia traffic", *Proc. of the 5th IFIP Conf. on High Performance Networks*, October 1995.
- [45] S. Lavenberg, "Mean Value Analysis of Closed Multichain Queuing Networks", *Journal of the Association for Computing Machinery*, vol. 27, no. 2, April 1980, pp. 313-322.
- [46] Z. Wang and J. Crowcroft, "Quality of Service Routing for Supporting Multimedia Applications", *IEEE Select on Communication*, September 1996.