

國立政治大學資訊科學系
Department of Computer Science
National Chengchi University

碩士論文
Master's Thesis

預算法全 IP 核心網路服務品質管理之路徑規劃
**Path Planning in Budget-Based QoS Management
for All-IP Core Networks**

研究生：李宗勳
指導教授：連耀南

中華民國九十二年九月

September 2003

內容目錄

摘要 (Abstract).....	9
第一章、簡介 (Introduction).....	9
1.1 電信產業的趨勢.....	10
1.1.1 整合性網路 (Network Convergence - All-IP Network).....	11
1.1.2 封包網路上乘載具時效性應用問題 (Real-time Application on packet-switching networks).....	12
1.2 網路服務品質定義 (Definition of QoS).....	13
1.2.1 分歧的品質保證期望 (Diversified QoS Expectations).....	13
1.3 全 IP 網路之服務品質保證 (QoS over All-IP Networks).....	14
1.4 UMTS 服務等級 (UMTS Services Class).....	14
1.5 網路服務品質管理方法 (QoS Management).....	17
1.5.1 Integrated Service.....	17
1.5.2 Differentiated Service.....	18
1.6 研究動機及目的 (Motivation and Research Objective).....	18
1.7 解決方案 (Solution Approaches).....	19
1.8 論文組織結構.....	20
第二章、相關研究 (Related Work).....	21
2.1 服務品質保證方案相關研究.....	21
2.1.1 端對端的服務品質保證.....	21
2.1.2 資源分配最佳化.....	21
2.2 服務品質網路架構.....	22
2.2.1 Integrated Service.....	22
2.2.2 Differentiated Service.....	23
2.2.3 Victor's System.....	26
2.2.4 TEQUILA.....	27
2.3 繞徑問題相關研究.....	29
2.3.1 傳統繞徑方法.....	29
2.3.1.1 (Flooding).....	29
2.3.1.2 最短路徑演算法 (Shorest Path Routing).....	29

2.3.1.3 訊務基礎路徑演算法 (Flow-based Routing).....	29
2.3.1.4 距離向量繞徑演算法 (Distance Vector Routing).....	30
2.3.1.5 鏈結狀態繞徑方法 (Link State Routing).....	30
2.3.2 具服務品質保證之繞徑方法 (Multi-Constrained Path).....	30
2.4 評論 (Summary).....	31
第三章、以預算為基礎之服務品質保證 (Budget-Based QoS).....	33
3.1 BBQ 架構 (Budget-Based QoS Framework).....	33
3.1.1 簡化的 All-IP 網路架構 (A Simplified All-IP Network Architecture).....	34
3.1.2 以預算為基礎之管理 (Budget-Based Management).....	35
3.1.3 路徑定義 (Paths Definitions)	36
3.1.4 承載服務架構 (Bearer Service Architecture).....	36
3.1.5 服務品質熵數 (Quality Entropy).....	37
3.1.6 即時資源分配與預先資源管理 (Pre-Planning vs. On Demand Allocation).....	38
3.1.7 集中式配置與分散式配置 (Centralized Allocation vs. Distributed Allocation).....	39
3.1.8 需求預測.....	40
3.2 管理系統架構 (Management System Architecture for BBQ).....	41
3.2.1 BBQ 管理系統假設 (BBQ System Assumptions).....	41
3.2.2 分散式分層管理系統 (Distributed Management System Hierarchy).....	41
3.2.3 管理系統軟體架構 (Management System Software Architecture).....	43
3.2.4 簡化的端對端服務品質建立流程 (A Simplified End-to-End Path Setup Procedure).....	43
3.3 BBQ 中的核心網路架構與 QoS 元件 (Core Network Architecture and QoS Components for BBQ).....	44
3.3.1 核心網路資源規劃方法.....	45
3.3.2 核心網路內的資源規劃元件.....	46
3.3.3 分散式資源規劃運作流程.....	48
3.3.4 資源調配可能產生之問題討論.....	51
3.3.4.1 Inter-Ingress 資源浪費.....	51
3.3.4.2 Intra-Ingress 資源浪費.....	53
第四章、核心網路之路徑規劃.....	55
4.1 BBQ 架構中各層級之路徑 (Path Definition).....	56
4.2 在 BBQ 架構中的路徑規劃元件 (Path Planning Agent in BBQ Architecture).....	57

4.3 路徑規劃的環境假設.....	58
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 (Cooperation between LPPA and Short Path Planning Agent).....	58
4.5 核心網路內路徑規劃運作流程 (Path Planning Procedure).....	59
4.5.1 路徑資源需求與鏈結頻寬需求之轉換.....	59
4.6 路徑規劃最佳化.....	61
4.7 最佳化模型 (Problem Model).....	62
4.8 Greedy Algorithm for Path Planning(G.P.P.A.).....	65
4.8.1 演算法虛擬碼 (Pseudo Code).....	67
4.8.2 演算法複雜度分析.....	69
4.9 小結(Summary).....	69
第五章、效能評估 (Performance Evaluation).....	70
5.1 效能評估指標 (Performance Evaluation Metrics).....	70
5.1.1 獲利指標 (Profit).....	70
5.1.2 鏈結使用率之標準差 (Link Utilization S.D.).....	70
5.1.3 獲利密度(Profit Density).....	71
5.2 實驗設計 (Design of Experiment).....	71
5.2.1 實驗工具.....	71
5.2.1.1 BBQ Computational Simulator.....	71
5.2.2 實驗測試組產生方式.....	72
5.2.2.1 網路拓樸之產生與設定.....	72
5.2.2.2 訊務之產生與設定.....	72
5.2.3 實驗流程.....	73
5.2.4 實驗組.....	75
5.3 模擬過程及結果分析.....	76
5.3.1 實驗一：演算法的行為測試.....	76
5.3.1.1 獲利趨勢.....	76
5.3.1.2 獲利密度趨勢.....	77
5.3.2 實驗二：節點多寡之影響 (Number of Nodes).....	78
5.3.2.1 拓樸中節點多寡對於獲利指標的影響.....	78
5.3.2.2 拓樸中節點多寡對於鏈結使用率標準差的影響.....	79
5.3.2.3 拓樸中節點多寡對於獲利密度的影響.....	80

5.3.3 實驗三：連接率之影響 (Connectivity).....	80
5.3.3.1 連接率對於獲利指標的影響.....	81
5.3.3.2 連接率對於鏈結使用率標準差的影響.....	81
5.3.3.3 連接率對於獲利密度的影響.....	82
5.3.4 實驗四：預測誤差之影響 (Forecast Error).....	83
5.3.4.1 預測誤差之定義.....	83
5.3.4.2 預測誤差對於獲利指標的影響.....	83
5.3.4.3 預測誤差對於獲利密度的影響.....	84
5.4 結論.....	85
第六章、總結 (Conclusion).....	87
參考文獻 (References).....	88

圖目錄

圖 1.1：各 UMTS 服務類別對三大品質指標之容忍度	11
圖 2.1：每個使用者所得到的服務品質取決於網路的負載	17
圖 2.2：DiffServ Domain 說明	19
圖 2.3：以 DiffServ 為基礎的分散式資源管理	22
圖 2.4：TEQUILA 架構	23
圖 3.1：簡化的全 IP 網路架構	30
圖 3.2：端對端承載服務	32
圖 3.3：服務品質熵數之對應	33
圖 3.4：BBQ 管理系統軟體元件架構	38
圖 3.5：簡化的端對端服務品質建立流程	39
圖 3.6：BBQ 架構之核心網路	40
圖 3.7：核心網路軟體架構	43
圖 3.8：核心網路中資源分配流程	45
圖 3.9：資源規劃流程	46
圖 3.10：不同 Ingress 之間資源無法彈性調配	48
圖 3.11：不同路徑之間的資源無法彈性調配	49
圖 4.1：路徑規劃轉換鏈結頻寬需求之說明	55
圖 4.2、演算法流程	61

表格目錄

表 1.1：各種新興的網路技術	6
表 1.2：傳統電信網路，網路網路和整合型網路特性比較表	7
表 1.3：circuit-switching 與 packet-switching 特性比較	8
表 1.4：3GPP UMTS 品質分及與各項特性	11
表 1.5：UMTS Bearer Service Attributes 的數值範圍	12
表 2.1：各種 PHB 的服務範例	20
表 3.1：BBQ 中分層分工架構	37
表 4.1：BBQ 架構中各種路徑之定義	51
表 4.2：BBQ 架構中各種 PPA 的比較	53
表 4.3：各階段路徑規劃所需之輸入與輸出參數	56

表 4.4：資源分配路徑規劃符號表 58

表 5.2：效能評估實驗 71

摘要 (Abstract)

在未來的整合性網路的趨勢下，各種新興的應用將在整合性 All-IP[1]網路上承載，但是利用 packet-switching 的網路來承載具時效性的應用有其天生的缺點[2, 3]。因此服務品質是近年的重要研究議題，各種具有服務品質保證的機制被提出以改善服務品質的問題，但都有其不足之處。因此行動通訊研究團隊提出 Budget-Based QoS Management 架構，企圖提供各式應用 End-to-End 的傳輸網路品質保證服務，而為了保證進入網路中的訊務傳輸服務品質，本文提出以路徑規劃的方式作為此架構中路由方法，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，且可以提供進入網路的訊務較佳的服务品質保證。

第一章、簡介 (Introduction)

整合型 All-IP 網路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[4, 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。本計畫旨在探討整合型 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 電信產業的趨勢

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網路以支援所有的應用服務。此種整合性網路將以一個單一傳輸平台提供固定網路及行動網路上所有的應用，包括語音、多媒體、資料等各類服務。

而推動此種趨勢的主要因素如下：

- 電信自由化的浪潮，刺激網路的大量建設、新技術的加速引進、與新服務的提供。
- 光纖技術的進步，使得以 DWDM 技術為基礎的高容量光纖可輕易地取代銅線網路，長途頻寬的供應大幅增加，價格降低。
- 近年來網際網路的蓬勃發展，刺激大量多媒體網路資訊的流通，使得頻寬需求大幅增加，網路的應用多樣化，對品質的要求亦隨之複雜化。
- 網際網路在全球蓬勃發展，間接使得 IP 技術變成全球網路的共同標準。
- VoIP 技術的發展，使得 packet switching 網路可以支援語音與即時影像服務。

由於以上的這些因素，網路服務正朝向多媒體化，多樣化的演進，而網路架構則朝向寬頻化、光纖化、扁平化、整合化發展。為因應這種趨勢，許多新興網路技術正被積極的研究(表 1.1)：

表 1.1：各種新興的網路技術

交換網路	VoIP, MPLS, Network Convergency
傳輸網路	DWDM, IP over SDH, IP over Fiber
接取網路	FTTx, xDSL, HFC, LMDS, 3G
智慧型網路	LND, Service Creation

1.1.1 整合性網路 (Network Convergence - All-IP Network)

這些新興技術中，對現有網路衝擊最大的當屬 Network Convergence，企圖將目前分離的 circuit switching 與 packet switching 傳統網路整合成單一的網路。為了打破以往不同製造商設備無法完全互通的問題，這個整合性網路將採用全球統一開放標準，而 IP 通訊協定正因為網際網路以全球使用率最高而成為唯一的選擇。此種革命性的整合型 All-IP 網路不但可以降低建置成本、營運管理成本，更重要者，還可提供一個新的服務平台，使得跨網路的應用成為可能。表 1.2 列出傳統電信網路及現在的網路網路和整合型網路的特點比較：

表 1.2：傳統電信網路，網路網路和整合型網路特性比較表

	PSTN/IN	Internet	Converged Network
多媒體服務承載	否	是	是
QoS-enabled	是(語音部分)	否	是
Network Intelligence	是	否	是
Intelligent CPE	否	是	是
底層傳輸媒介 Underlying transport network	Circuit-switching Network	Packet-switching Network	Packet-switching Network

	PSTN/IN	Internet	Converged Network
服務可信度 Service reliability	High	Low	High
Service creation	Complex	Ad-hoc	Systematic
服務易用度 Ease of use of service	Medium	High	High
Evolvability/modularity	Low	Medium	High
Time to market service	Long	Short	Short
Architecture openness	Low	High	High

1.1.2 封包網路上乘載具時效性應用問題 (Real-time Application on packet-switching networks)

傳統上，具有時效性的服務是由 circuit-switching 承載，而 packet-switching 網路大多用以傳送資料，表 1.3 簡單比較 packet-switching 和 circuit-switching 網路的特性。但是若要在未來的整合型 All-IP 網路上承載所有服務時，受限於 packet-switching 的天生特性，將會面臨以下三大問題：

- 封包傳送延遲時間過長 (Long Delay)：在 IP 網路中，資料是以封包的形式傳送，經過每一個 hop 時的處理將導致傳送時間容易增長。
- 封包傳送時間抖動 (Jitter)：封包是在一個一個的 Hop 中交遞傳送，在每一個 Hop 中容易受到其他的因素影響，如過多的封包在同一時間傳送、每一個封包的傳輸路徑不同，而造成在接收端收到封包的時間間隔不定，若是進行一個 Video On Demand 應用，則會造成影像的斷續。
- 封包遺失問題 (Packet Loss)：封包在傳送過程中，易因傳送媒介的問題，使得封包在傳送中被丟棄(Drop)，對於需要可靠資料傳送的應用如 E-mail 和 FTP 而言，可能會造成資料不完全的結果，而必須進行重新傳送，導致網路負擔增加。

表 1.3：circuit-switching 與 packet-switching 特性比較

比較項目	circuit-switching	packet-switching
指定傳送路徑	是	否
可使用頻寬	固定	動態調整
可靠性	良好	較差
循序傳送	是	否
儲存後傳送	是	否
傳送初始化	需要	不需要
阻塞可能發生時間	初始階段	每個封包傳送時
計費方式	以時間為單位	以傳送資料量或以時間為單位

1.2 網路服務品質定義 (Definition of QoS)

處理服務品質的問題，除了對網路品質問題要有深入的瞭解外，還要對使用者與網路營運者對品質保證的期望有所瞭解。不同的服務，不同的使用者不同的網路營運者對品質的定義不一定相同。例如 VoIP 注重傳送延遲時間和傳送延遲抖動(Delay and Jitter)，而 FTP 注重封包是否遺失。此外，他們對品質的期望也不盡相同。

1.2.1 分歧的品質保證期望 (Diversified QoS Expectations)

不同的使用者對網路所提供之品質服務保證，可能有不同的期望。例如，一般的使用者可能會有下列不同的期望：

1. 以最低的價格，買到最好的服務品質
2. 以最低的價格，買到需要的服務品質

3. 以可接受的價格，買到最好的服務品質
4. 以最低的價格，買到可忍受的服務品質

另一方面，網路營運者依據其策略上的目的，也可能有不同服務品質政策。例如，網路營運者可能：

1. 以可接受的售價，提供對使用者而言，最好的服務品質
2. 以最高的售價，提供使用者可接受的服務品質
3. 以最低的售價，提供使用者可忍受的服務品質

本研究之目的是設計一個有彈性的服務品質保證管理架構。網路營運者可根據自己的營運目標調整自己的網路管理系統，以達到營運者策略上的目的。

1.3 全 IP 網路之服務品質保證 (QoS over All-IP Networks)

All-IP 網路受限於封包交換網路原有的特性，有前述三大品質問題有待克服[7]。近年來雖然有極多的 QoS 研究，但是絕大多數著重於研究既有的數據服務 QoS，其品質管理偏重頻寬之管理，並不適用於欲提供全面性服務之整合性全 IP 網路。舉例而言，台灣連接到美國的通訊鏈路中，有海底電纜亦有衛星通訊，只要求頻寬的傳統數據服務並不排斥運用衛星鏈路。但是，對於封包延遲有嚴格要求的語音服務 (Voice Over IP) 而言，衛星鏈路即使可以提供充足的頻寬，但因傳輸距離所造成的 delay time 將嚴重影響通話品質。因此，一個好的全 IP 網路管理，應該針對各種應用服務的各種品質需求參數提供適當的資源分配與管理，而非只針對頻寬進行管理。

面對品質要求即時化、多樣化且負載極高的 All-IP 網路，其品質管理複雜度遠比單純的語音或數據網路複雜。猶如管理大小汽車與機車爭道的一般街道遠比車種單純的高速公路複雜一樣，All-IP 網路上的品質管理是一大技術挑戰。網路管理系統必須提供適當的品質資源管理機制，並讓管理者可輕易的調校網路，使得各類服務都可以獲得適當的品質服務。

1.4 UMTS 服務等級 (UMTS Services Class)

3GPP 在 1999 年初所開始制訂的 UMTS (Universal Mobile Telephony System) 系統上有提出完整的 QoS 階層式架構，其中包含對即時性訊務的服務考量。UMTS 在不同的網路上有不同的方式支援 QoS，主要分成 Radio Network、Radio Network – Core Network interface 與

Core Network。

Radio Network 主要是以 W-CDMA 為標準，其服務範圍包含所有類型的訊務，以 Dedicated Channel 與 Shared Channel 來滿足不同服務所需求的服務品質，並以 Soft Handover 與針對即時訊務所做的 Optimized Handover 來彌補在 handover 時品質上可能造成的損失。每個資料流有個別的 PDP (packet data protocol) contexts，配合 GTP (GPRS Tunnel Protocol) 建立從 Radio Network 起至 Core Network 有 QoS 功能的 tunnel，以 ATM 為基礎來傳送資料，IP 層也可配合使用 DiffServ。

UMTS 系統將網路上風行的應用依時效與品質需求分為四大類：交談式 (conversational class)，串流式 (stream class)，互動式 (interactive class)，背景式 (background class)。以下是簡要分析：

交談式主要用來支援人類雙向溝通，根據人類感官之經驗歸納，此種服務對 delay time 與 jitter 相當敏感，使用者在 delay time 超過 300 ms 時，就難以忍受其通話品質。串流式則要求持續穩定的 packet flow，因此對 jitter 相當敏感。互動式與背景式都屬於 data communication 的服務，可容忍較長的資料傳送時間，但是要求精準的資料傳送，因此幾乎無法忍受資料的遺失。圖 1.1 顯示每個類別品質指標之容忍度；表 1.4、1.5 表示各類別的特性與品質需求。

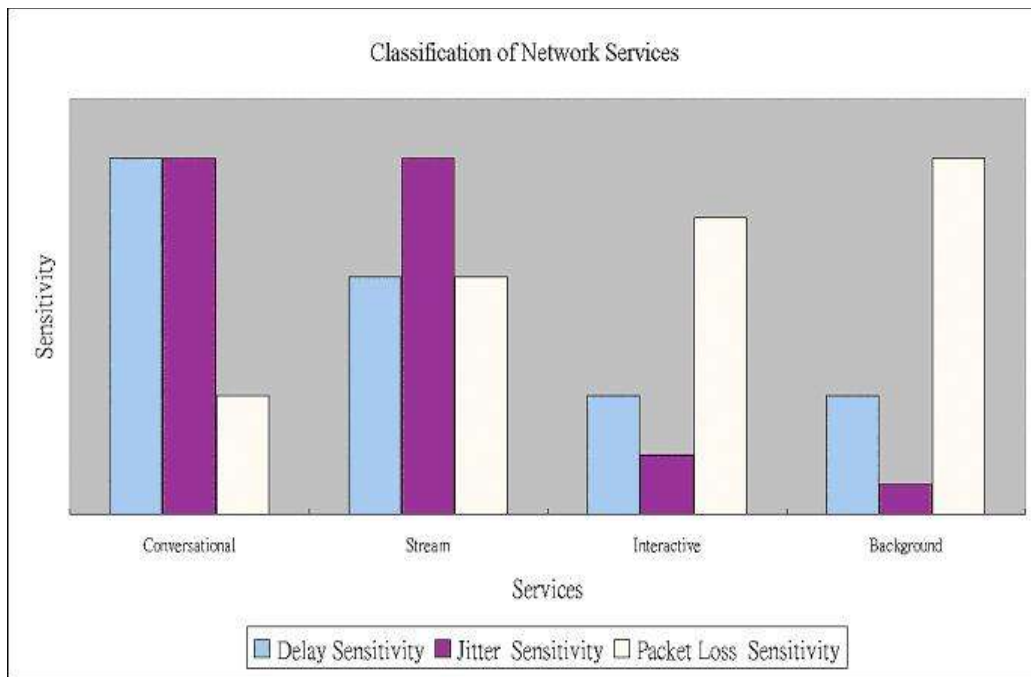


圖 1.1：各 UMTS 服務類別對三大品質指標之容忍度

表 1.4：3GPP UMTS 品質分及與各項特性

Traffic class	Max Bit	Transfer	Guaranteed	Traffic	SDU Error
	Rate	Delay	Bit Rate	Priority	Rate
Conversational	X	X	X		X
Streaming	X	X	X		X
Interactive	X			X	X
Background	X				X

表 1.5：UMTS Bearer Service Attributes 的數值範圍

Traffic class	Conversational class	Streaming class	Interactive class	Background class
Maximum bitrate(kbps)	< 2048	< 2048	< 2048 – overhead	<2048 - overhead
Delivery order	Yes/No	Yes/No	Yes/No	Yes/No
Maximum SDU size (octets)	<= 1500 or 1502	<= 1500 or 1502	<= 1500 or 1502	<= 1500 or 1502
Transfer Delay	100 – maximum value	250 – maximum value		
Guaranteed bit rate (kbps)	< 2048	< 2048		

本架構於網路層提供服務品質之管理，本架構以提供一個具適應性的管理模型為目標，上層的使用者應用程式需求和下層之服務品質執行層將是可替換。所以，上方之使用者應用程式於服務品質的要求，不侷限於 UMTS 所分類之四種服務等級需求；所管理之服務品質參數也不用侷限於 UMTS 所考慮之 delay time、packet loss、jitter。同樣，底層之 QoS 執行層也不一定侷限使用 DiffServ。如此設計的好處為此一般化管理模型可處理其他的網路環境，而不會被上、下層之網路技術限制。

1.5 網路服務品質管理方法 (QoS Management)

針對以上在封包網路上容易發生的傳送品質問題，網路標準機構 IETF 制定了 Intserv (integrated service)與 DiffServ (differentiated service)兩種機制，對於異質性網路提供服務品質

保證的環境。

1.5.1 Integrated Service

IntServ 使用 RSVP(Resource Reservation Protocol)[8]針對各個訊務建立一保留頻寬的 virtual circuit 來滿足 QoS 上的需求。這樣的架構有其相當的好處，首先，它對使用者提供了絕對保證的 service，再者，每個使用者的資料流都可以輕易地監控管理，並且可以運用現有的 routing protocols。但因 overhead 過多，使得網路在擴充性(scalability)上受到極大的限制。

1.5.2 Differentiated Service

DiffServ [9]則是採用與 IntServ 不同的策略來提供服務品質保證。DiffServ 是將具有相似 QoS 需求的訊務合併一起處理，對同一類型的資料提供一致性的服務與相對性的保證，而不是針對各別的訊務提供保證。每一類型的資料會有其相對應的 Per-Hop Behavior 在 DiffServ domain 上傳送。這樣的方法雖然沒有辦法達到如 IntServ 般絕對的服務品質保證，但卻可以解決 IntServ 在擴充性和實作上的問題，因此 DiffServ 的架構漸漸取得其主流地位，但如何在 DiffServ 的架構上，提供各個的訊務的 End-to-End QoS 保證正是極待解決的主要問題。

綜觀以上兩種服務品質保證之方法，若是以 Intserv 方法來提供服務品質保證，則服務網域會因為訊務控制資訊量成長快速而受到限制，並且 RSVP 路徑尋找方法所產生的訊務量 (RSVP message)也會導致整個網路要負擔額外增加流量。若是採用 DiffServ 方法來提供服務品質保證，所有的訊務被歸納為三個不同的 QoS 服務等級，只在每個 DiffServ 網域內的節點之間提供相對的品質保證，所以在純粹 DiffServ network 環境下，我們無法針對每個 flow 做到真正的 End-to-End QoS。

1.6 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。

本研究之目的在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。並且，除了提供端對端品質保證予使用者之外，網路營運者更希望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，封包繞徑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徑演算法即可應付，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，因此如何使得繞徑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徑演算法配合。

1.7 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，並將部分的繞徑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。

本研究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。

1.8 論文組織結構

在本文中，共有六個章節，第一章將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。而在第五章中將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 服務品質保證方案相關研究

2.1.1 端對端的服務品質保證

在傳統封包網路上，封包的傳送彼此是獨立的，在封包傳送的過程之中，會因種種不同的因素，使得每一個封包在傳送的過程之中傳送的時間不定，甚至遺失，而使用者只能期望網路將資料盡力送達(Best Effort)。在網路的發展過程之中，許多方法被提出用以對付這種不穩定性，且有一些成果。可是自 1989 後 World Wide Web 的推波助瀾下，Internet 以驚人的速度成長，將世界變成了一個地球村，當我們以瀏覽器瀏覽位於 IETF 主機上的 RFC 時，我們要求網頁的封包可能透過不同的主機一站一站的送至對方的 Web 主機上，這些主機都是位於不同網路營運者的網域之中，不同於一般的核心網路大多由一個網路營運者所管理，而每一個網域中，可能因為營運者不同的經營策略，而對於網路的資源做不同的配置，因此當我們需要一個 End-to-End 的服務品質保證時，我們將面臨許多困難，BBQ 的架構試圖要為這個問題提出一個可能的解決方案。

2.1.2 資源分配最佳化

在網路服務品質保證的研究之中，一般皆認為給某個應用較多可用的頻寬(圖 2.1)[10]，代表著網路的傳輸品質較好，相對的也會提昇此應用的服務品質，如常見的 Voice over IP 的應用，當以較大的 bit rate 來傳送語音封包資料，將可以使得語音傳送品質更為清晰，因此網路上的頻寬是一個網路的重要資源，如何將網域內的資源做最佳化的處理，使系統可以承載更多使用的需求，是每一個網路營運業者的目標，在 BBQ 的架構中以各種不同任務的代理人(Agent)來達成此目標，希望可以提供一個最佳化的方法供各營運業者作為參考。

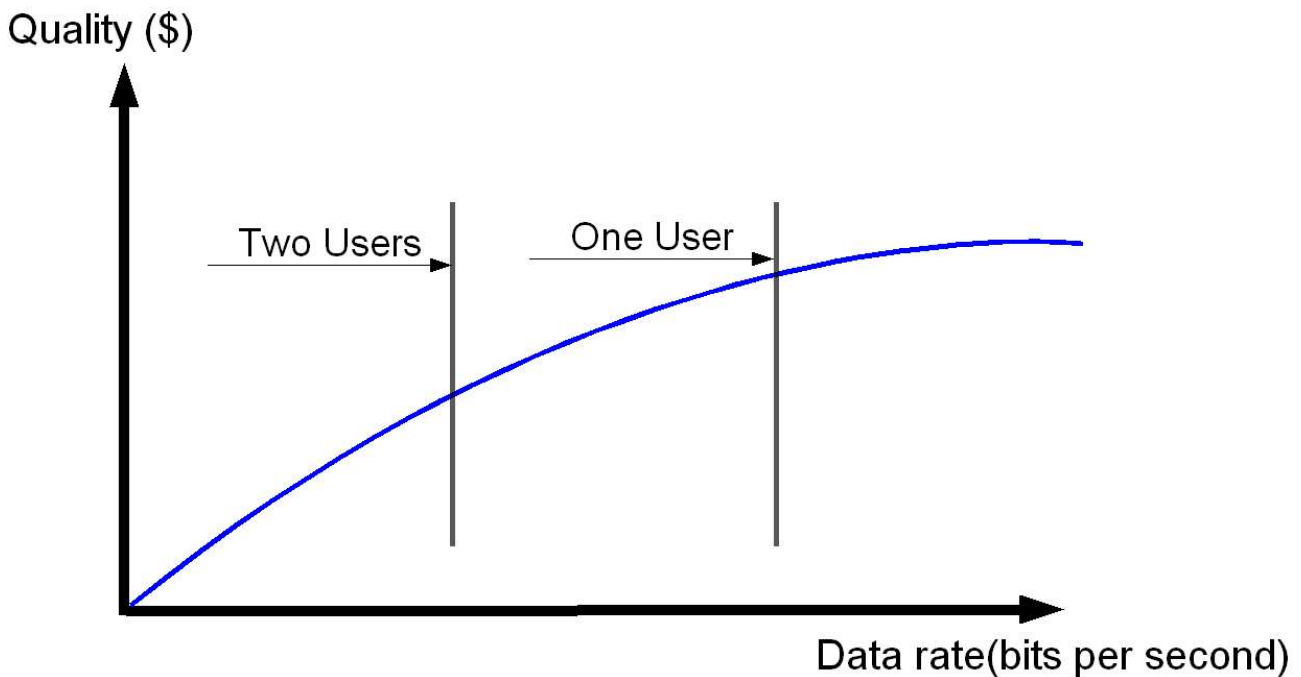


圖 2.1：每個使用者所得到的服務品質取決於網路的負載

2.2 服務品質網路架構

為了提供異質性網路上各種 QoS service，需要有一套完整的 QoS 架構與管理機制。目前的研究大多將整體 QoS service 分為 data plane 與 control plane。data plane 為分類服務，將 packet 依照其需求之服務等級進行區分，目前廣為採用的是 IETF 制定的 DiffServ。DiffServ 依照各種服務需求屬性，將封包分為 EF、AF、BE 三個等級。control plane 即是以 data plane 為基礎所衍伸出的管理架構，隨著管理架構不同，管理方式也就有所不同。

而近年來許多的學者提出了可以提供服務品質保證的網路架構，其中較為值得注意的是由 Spiridon Bakiras 和 Victor O.K. Li 在 2002, IEEE, 「Efficient Resource Management for End-to-End QoS Guarantees in DiffServ」[11] 所提出的在 DiffServ 原有的架構之中增加 End-to-End 的 QoS 機制來達到服務品質的保證。以下簡介幾種近年來學者所提出的服務品質保證的管理架構：

2.2.1 Integrated Service

IntServ 利用 RSVP(Resource Reservation Protocol) 為各個資料流做品質管理，保留資源建

立一個專屬的虛擬通道(virtual circuit)來滿足 QoS 上的需求。在建立通道時順便保留資源，傳送端每隔一段時間會傳送 PATH 的訊息至接受端，內容包含訊務的種類與需求的資源等訊息，接收端在收到此訊息後會傳送 RESV (reserve) 訊息，循著 PATH 訊息傳送的路徑回到傳送端，沿途每個節點會處理 RESV 訊息並保留資源，當 RESV 訊息回到接收端後，一個保留資源的虛擬通道便建立完成。

除了實行簡單外，這樣的架構還有其相當的好處，首先，它設計對使用者提供端對端品質上絕對的保證，由於從傳送端至接收端都執行 RSVP 來保留資源與建立通道，因此每個資料流都可以被監管，避免資料流消耗超過其請求保留的資源，再者，每個使用者的資料流都可以輕易地監控管理，並且建立資料流路徑可以運用現有的路由協定。

可是 IntServ 有項重大的缺點，在建立 virtual circuit 時，路徑中每個節點都要參與，並保留每個資料流的使用狀態和負擔許多在通道建立上計算的 overhead，然而網際網路上卻有數量龐大短時間使用的資料流，保留和管理每個資料流的使用資源與狀態，會對網路造成相當大的負荷，此特性使 IntServ 在擴充性(scalability)上有限制，不適用於大型網路。

2.2.2 Differentiated Service

DiffServ 是將具有相似 QoS 需求的訊務合併視為同一 aggregation 一起處理，對同一 aggregation 的資料提供一致性的服務與相對性的保證，然而在沒有指定資料傳送路徑的情況下，有時間敏感度的訊務並無法得到端對端品質保證。每一類型的資料會有一個固定的 DSCP (DiffServ codepoint) 讓網域內的節點來分辨，傳遞資料時，每一個 DiffServ 節點會根據此類型資料的 DSCP，依其相對應的 PHB (Per-Hop Behavior) 傳送。

一個 DiffServ Domain 是由許多個提供 DiffServ 服務，具有相同 PHB 且相連的節點所組成，這些節點主要可以分為 Edge Router 和 Core Router。如圖 2.2，X domain 為一沒有 DiffServ 功能的網域，Y 和 Z domain 為各別的兩個 DiffServ 網域，因為兩者可能執行不同的 PHB 或是對同類別的資料有不同的 DSCP。與其他網域連結的點統稱為 Edge Router，又分為 Ingress Router 和 Egress Router，分別表示訊務進入網域和離開網域的節點；沒有與其他 domain 相連接的節點稱為 Core Router。

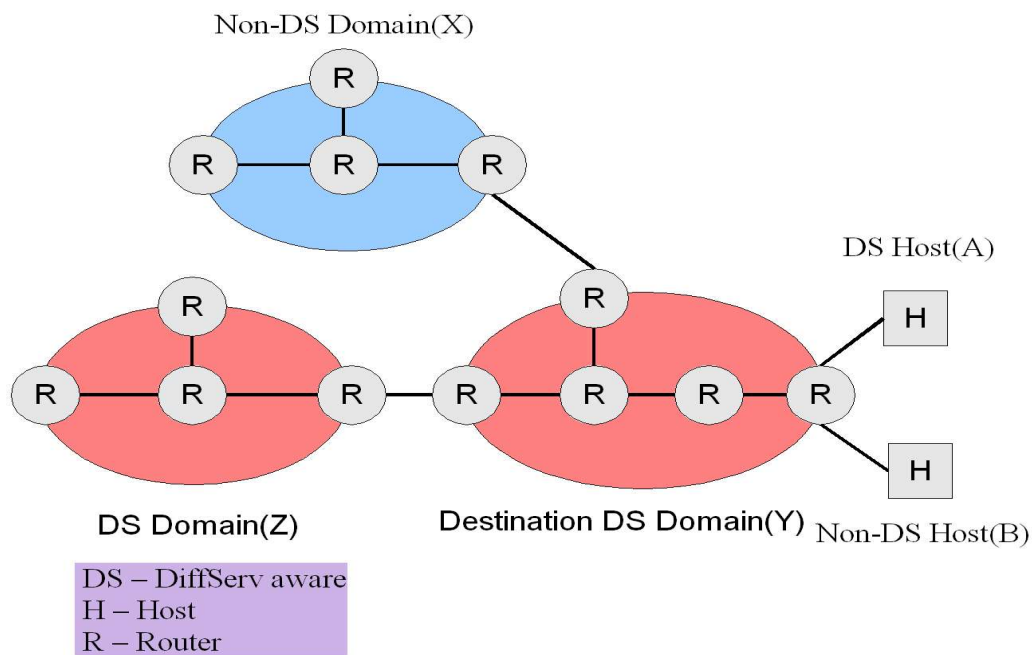


圖 2.2 : DiffServ Domain 說明

DiffServ 主要有兩個 function 來管理和控制網路上的資料 -

- Classifying :

依據如來源和目的地的 IP 位址、應用程式與 port number、或是 network protocol 等來進行分類。

- Policing :

- Metering – 測量某一 traffic stream 的 flow rate 與 burst size 等參數，這些及時的數據可以提供給其他如 marker、shaper、dropper 等或是計費等等參考使用。
- Shaping – 控制同一個 traffic stream 的封包傳送速度與整批傳送的數量，以符合傳送前所訂定的 traffic profile。
- Dropping – 根據上面的結果，將封包拋棄以減低網路負荷。

不同性質的應用所產生的資料有不同的特性。例如部份的資料有時效性，如果傳遞的延遲太長，即使接收方無誤的接收到資料，也是無效，反而造成網路頻寬的浪費；反之，部份的資料並沒有時效性的問題，但在傳送上不能有遺缺或錯誤。

在 DiffServ 的架構中，根據需求之不同提供不同等級的服務，識別方法是利用在 IP

header 上的 DSCP 來指定不同的傳送方式。針對訊務本身的特性，IETF 的 Network Working Group 定義了數種基本的 PHB 傳遞資料，其中包含 Assured Forwarding (AF)將資料分成數個不同的類別(class)來傳送與一個高品質的 PHB 稱為 Expedited Forwarding (EF)：

- Expedited Forwarding [12] – 降低任何網路壅塞時可能的延遲以支援需求低延遲的服務，如 Voice over IP(VOIP), Video on Demand(VOD)等應用即可使用此項服務乘載。
- Assured Forwarding [13] – 依據每一個封包的被拋棄的先後等級來決定當網路壅塞時，那一種封包將會被優先丟棄，藉以提供此等級的封包一種相對性的保證，如 Web Browsing, Telnet 等應用為此項服務的適用範圍。
- Best Effort [14] – 支援沒有特別需求與還沒有 DiffServ 時(backward compatibility)的網路 traffic，如 E-mail 和 FTP 等應用即可由此項服務來乘載。

表 2.1：各種 PHB 的服務範例

PHB	Examples
BE	E-mail, FTP
EF	Voice over IP(VOIP), Video on Demand(VOD)
AF	Web Browsing, Telnet

除了提供不同等級的服務，AF 每一個類別中的封包可以被指派數種不同的 drop precedence，超過的部份則會有較高的 drop precedence。當網路壅塞時，DiffServ 網域上的節點會優先捨棄較高 drop precedence 的封包來保護較低 drop precedence 的封包。由上述得知，在 DiffServ 網域中，不同的傳遞品質保證是受到配置給其所屬 AF 類別的資源、此 AF 類別的負荷與同一類別中封包的 drop precedence 所影響。

EF 是用來提供通過 DiffServ 網域，低 latency、低 loss、低 jitter，保證頻寬的高品質服務。不論是 packet loss、latency 或 jitter 都是由於資料傳送時在網域某節點的 queue 上等待的所造成的，所以要達成上述的目標，在網域的節點設定上，不論網域與其他 aggregation 忙碌與否，EF 的資料要有一個最小的離開速率。此外，必須對進入網域 EF 的資料嚴格控管，使得不論在網域任何一個節點，訊務離開的速率要比進入得快。

在資源配置上，每個 aggregation 都必須有基本的資源配置，對 AF 而言，部份網段的壅塞或許不會對高優先權的訊務產生影響，但會使得大量的低優先權訊務遺失，造成其他網段的資源浪費，甚至惡化網路壅塞的情況。EF 則以控制訊務進出節點的流速來提供高品質的服務，當網域某些地方訊務繁忙時，為了不降低服務品質，只好減少進入網域或通過節點的訊務，造成其他網段資源的浪費，整體資源的使用率與系統業者的收入也相形降低。

DiffServ 的缺點是沒有辦法如 IntServ 提供個別訊務絕對的端對端服務品質保證，僅能對同一個 aggregation 的資料提供相對的品質保證，在沒有指定路徑的情況下，對於有時間敏感度的訊務，並沒有辦法提供個別資料流端對端的品質保證。由於不必對個別資料流提供品質保證及管理，可大幅降低 overhead，避免了在擴充性和實作上的問題，所以 DiffServ 的架構已被廣泛接受採用。但欲在 All-IP 網路上使用 DiffServ，由於 DiffServ 的架構並沒有辦法對各別訊務提供端對端品質保證，必需有其他的網路管理架構來提供各個訊務端對端的品質保證。

由於上可知以 DiffServ 只能提供 per-aggregation 的保證，無法達到所需要的 Per-flow QoS 保證，雖然 BBQ 以 DiffServ 為底層的架構，但是會在 QoS 的執行上加以加強，以期達到 end-to-end per-flow QoS 保證。

2.2.3 Victor's System

在階層式管理架構，資源允入管理可分為集中式與分散式管理。分散式允入管理架構眾多，其中較具參考性由 Victor O.K. Li 等人於 2002 年 IEEE International Conference on Communications 發表之管理架構，以分散式資源管理由 Bandwidth Broker 將資源分配給所屬核心網路之 Ingress Router，由 Ingress Router 進行允入控制。

該管理架構中(圖 2.3)，基本網路分成多個核心網路，核心網路所採用的 QoS 機制為 DiffServ。

不同於過去以往對於資源集中管理，此架構以核心網路為管理單位，由 Bandwidth Broker(BB)將頻寬資源分配給核心網路之各個 Ingress Router，再由 Ingress Router 進行允入控制，每隔一段固定時間由 BB 檢視各個 Ingress Router 之頻寬使用效能，依照各個比例重新分

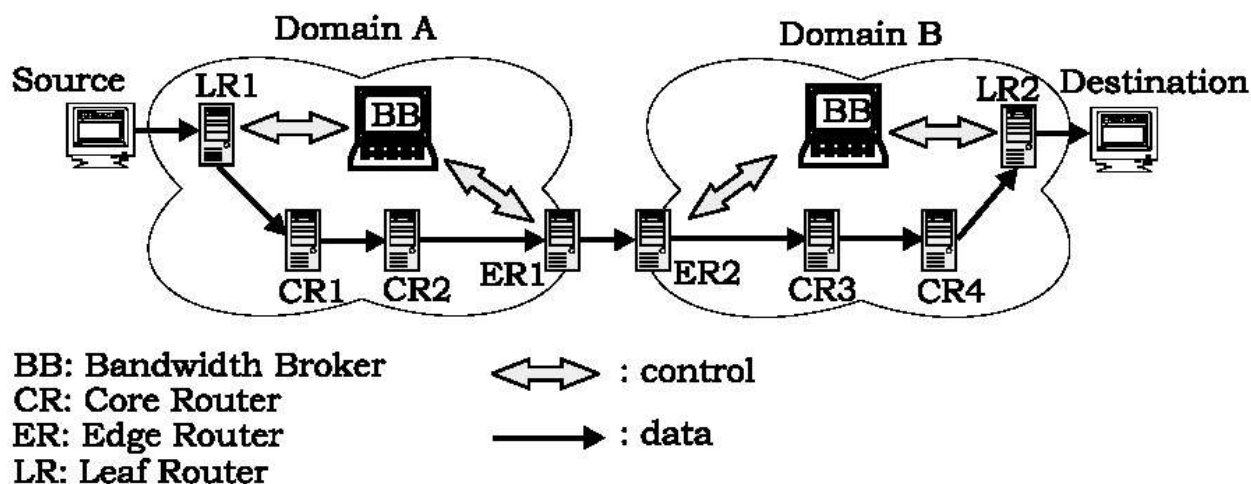


圖 2.3：以 DiffServ 為基礎的分散式資源管理

配頻寬資源，如此分配若是 BB 著重於整體的利益考量，就容易出現公平性(fairness)的問題，某些比較重要的 Ingress Router 分到較多資源，而有些 Ingress Router 分不到資源或是過少而使得允入時發生資源短缺無法順利接受訊務以提供服務。

Victor O.K. Li's system 以類似 RSVP 的方式進行允入控制，分為 inter-domain 與 intra-domain。Inter-domain 為跨核心網路之允入控制(如圖 2.3 之 ER1 到 ER2)；intra-domain 則為該核心網路之允入控制與路徑選擇(如圖 2.3 之 LR1 到 ER1)。在此架構下，由於 intra-domain 之資源分配是以固定時間方式分配，於進行分配時 BB 參考上個時段各個 Ingress Router 允入控制情形與進入流量，再加以重新配置資源。若是網路流量隨著時間不同而有所差異，此作法不易對當時情形作出反應而難以掌握突然大量出現的訊務；intra-domain 之允入控制為管理跨核心網路之允入情形，在此架構下目前只考慮 DiffServ 核心網路，對於其他類型的 QoS network(如 UMTS)尚未提及。由於不同類型的 QoS network 對於 QoS 服務等級有不同的定義，當跨 QoS network 時則必須考慮到服務等級對應的問題，必須以恰當的對應使得訊務在進行跨網路時依然能夠得到適當的 QoS 服務，因此在核心網路部分可引用其他類型的核心網路以此研究所提出之管理架構與方法為基礎作更進一步的研究。

2.2.4 TEQUILA

The Traffic Engineering for Quality of Service in the Internet at Large Scale (TEQUILA) [15] 的計畫是許多歐洲的電信業者所共同贊助的一個計畫，目標是研究網路服務的定義並提出一些 traffic engineering 的工具來達成兼具質和量的服務品質保證。在 2001 年，P. Trimintzios 等人於 IEEE Communication Magazine 上發表了「A Management and Control Architecture for Providing IP Differentiated Service in MPLS-Based Network」，就是 TEQUILA 計畫中的 QoS 管理架構。

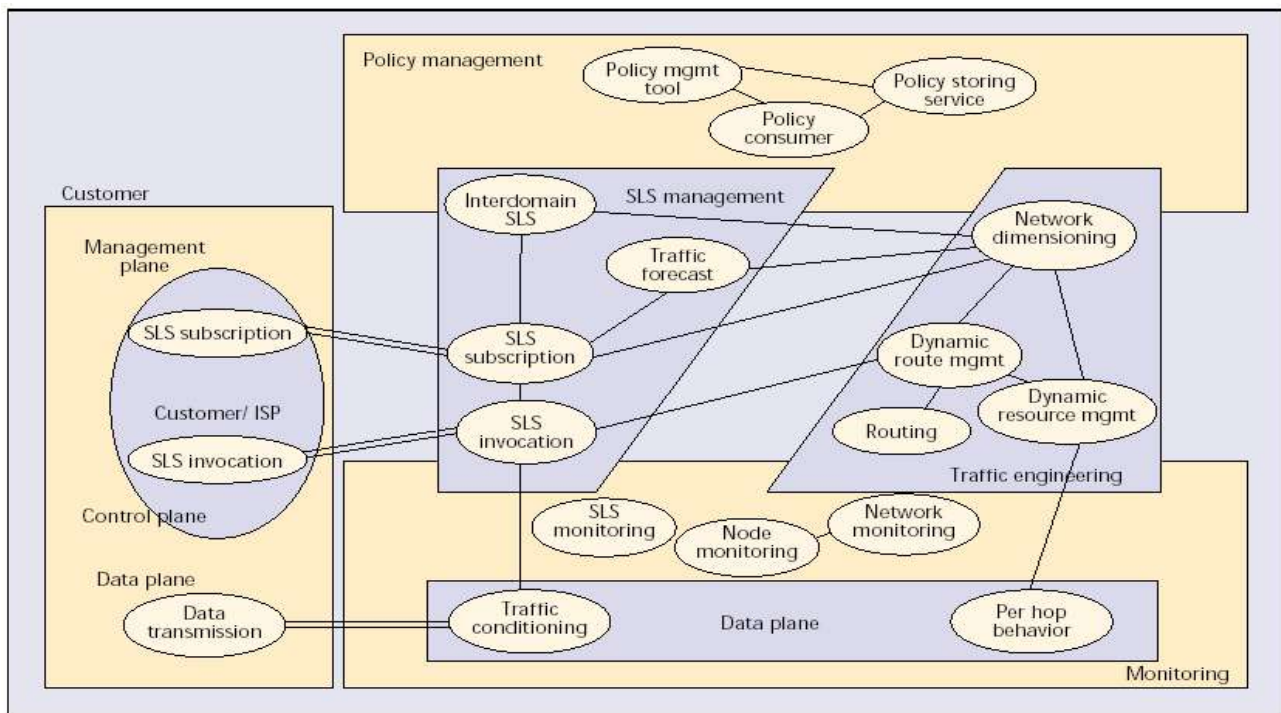


圖 2.4：TEQUILA 架構

在 TEQUILA 的架構之中(如圖 2.4)，可分為三個主要的部分，Service Level Specifications Management 主要負責處理客戶的服務品質要求，Traffic engineering 是網路中主要管理 QoS 的協調工作，而底層的 Data plane 則是負責實際資料的傳送，目前其假設的底層服務為 MPLS。

一個 End-to-End 的服務品質保證是由客戶的 Service Level Specifications (SLS)開始，客戶端將其對於服務品質的需求以 SLS 的形式與服務架構中的 SLS Management 進行協調，系統則根據目前的負荷能力，來決定是否接受此 SLS。若接受，則 Traffic Engineering 下的 Network Dimensioning 元件則會根據 SLS Management 和系統的管理政策制訂者 Policy

Management 所給予的資訊來協調網路上資源的運用，然後由 Dynamic Route Management 和 Dynamic Resource Management 等元件來執行真正的資源管理和封包的傳送。

目前 TEQUILA 計畫仍在進行當中，許多細部的功能元件尚在討論研究階段，P. Trimintzios 等人在 2002 發表「A Policy-Based Quality of Service Management System for IP DiffServ Networks」[16]說明了其中在 TEQUILA 架構中 Policy 的制訂方法，而 2003 的「Admission Control for Providing QoS in DiffServ IP Networks: The TEQUILA Approach」[17]中則說明關於允入控制等相關的議題。

2.3 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：

2.3.1 傳統繞徑方法

2.3.1.1 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。

2.3.1.2 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)。

2.3.1.3 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。

2.3.1.4 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。

2.3.1.5 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) [27] 就是一種被大量使用的 Link State 繞徑方法。

2.3.2 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」[18]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。[19, 20]

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。

2.3.3 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。

2.4 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

的做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，在路徑規化與資源分配上，多半是以事先規劃的方式，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，本研究的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。

在本研究所提出的 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劃，並依規劃結果 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劃，在我們所提出的資源規劃方案中，核心網路上的資源分配方案和路徑規劃息息相關，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，然後分析路徑規劃的問題，並提出一個路徑規劃的獲利最佳化方案。

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。所以，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，必須使用訊務集合技術 (Traffic aggregate) 將許多 flow 歸併，減低管理負擔。並且，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。

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 管理提供一個高適用性的管理模型和管理工具，可適用於不同的下層網路技術和上層營運者管理目標。以下將簡介 BBQ 的網路架構和管理系統架構。

BBQ 之設計目標為提供一個可於 All-IP Network 上之提供服務品質保證之管理架構，此架構可供網路營運者調整其品質政策，運用有限資源達成最大的營運滿意度，以下介紹 BBQ 之環境及架構。

3.1 BBQ 架構 (Budget-Based QoS Framework)

BBQ 的目標為在全 IP 網路架構上，提供一個簡單易行的服務品質管理機制，以提供 Per flow 之 QoS。

本章節將先介紹一個簡化之全 IP 網路架構，並且分析重要的網路節點，以釐清 BBQ 的 QoS 元件所需之功能和元件位置；以預算為基礎之管理為 BBQ 分散管理權責、化簡系統複雜度的概念；路徑定義和承載服務架構則將端對端的服務分解為各元件所能負擔之服務；服務品質熵數為 BBQ 所定義之品質指標，透過此單一數值指標，使 BBQ 可適應不同的服務品質定義；資源分配的時機和資源配置的方式是影響資源管理效率及實用性的一大重點，本研

究將先分析介紹，並且於不同的管理層中採用相異的方案；最後分析介紹 BBQ 的管理系統架構。

3.1.1 簡化的 All-IP 網路架構 (A Simplified All-IP Network Architecture)

一個遍及全世界的通訊網路，是由各個獨立營運的大小網路所共同組合而成。傳統上，多數的國家只有一個由政府營運的網路。但過去十年來，許多國家進行電信自由化，開放更多數量的商業網路執照。這些網路利用網路互連協定互相連接，組織成遍及全世界的通訊網路，如此便可將訊息傳遞到世界上大部分的地區。本研究假設一個全世界的全 IP 網路也是利用相同的方式部署，但為了使此網路架構簡明易懂，做了以下的假設。

一個獨立營運的網路包括了一個核心網路 (Core network) 和數個接取網路 (Stub Network)，一個核心網路涵蓋了整個營運網路所能服務的區域，而接取網路則是只涵蓋了較小的區域，例如一個城市。接取網路可能是固定或無線接取網路，例如 Wi-Fi WLAN 或 3G Radio Network。一般而言，兩個連接在同一個接取網路的終端設備，可以不透過核心網路直接通訊，但為了簡化問題，本研究假設一個 IP 封包將由本地接取網路經過核心網路傳遞，再送到遠端接取網路。本地接取網路將稱為入口接取網路 (Entrance Stub network)，遠端則稱為出口接取網路 (Exit Stub network)。所有核心網路的營運者利用 interconnection 鏈結相互連接成的網路，稱為骨幹網路 (Backbone Network)。當傳送一個封包到其他網路時，將先從入口接取網路出發，經過數個核心網路 (骨幹網路) 後，最後送到出口接取網路。核心網路中連接接取網路的邊緣路由器 (edge router)，稱為邊界入口閘道器 (Border Gateway, BG)，在接取網路與核心網路間執行閘道器的功能。以提供服務品質保證的目的而言，邊界入口閘道器也需要執行允入控制 (Admission control)。另一方面，接取網路連接核心網路的邊緣路由器，稱為接取閘道器 (Access Gateway, AG)。核心網路互連之路由器則稱為 Inter-Domain Gateway。下圖 3.1 為簡化之全 IP 網路架構。

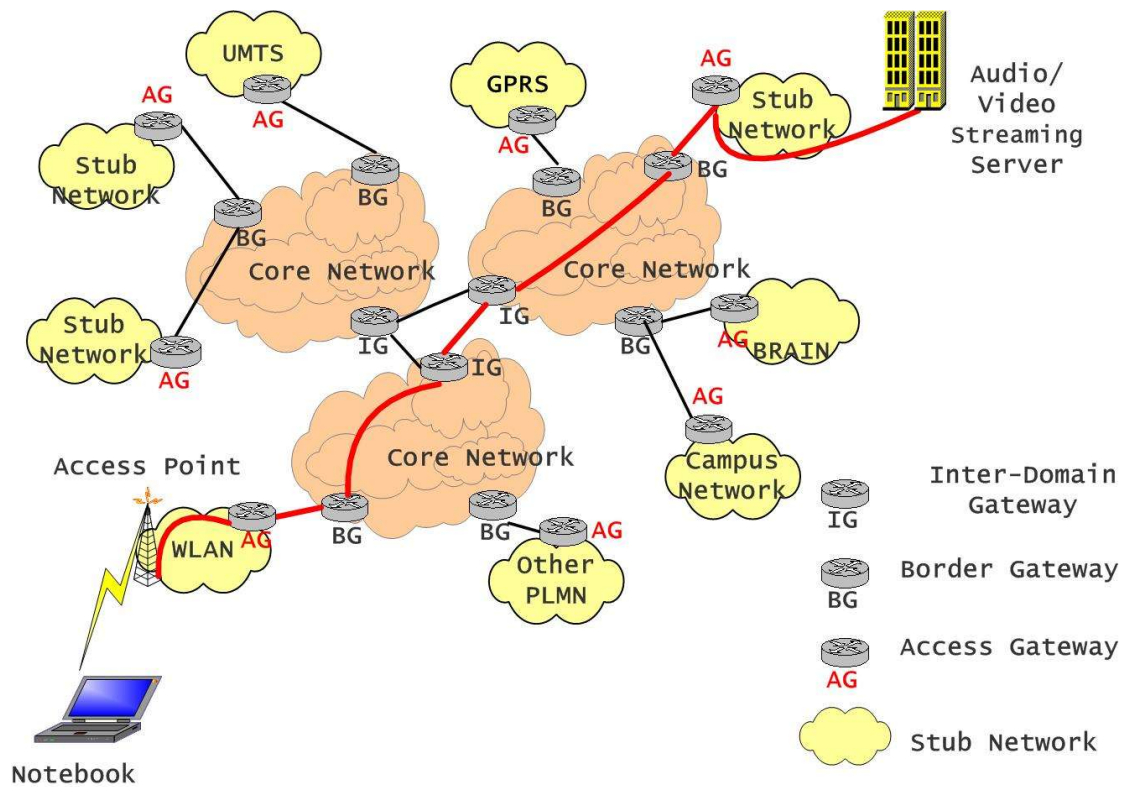


圖 3.1：簡化的全 IP 網路架構

3.1.2 以預算為基礎之管理 (Budget-Based Management)

為了提供端對端網路服務，網路必須提供端對端服務品質保證。在實際的通訊中，一個封包可能必須穿越數個不同營運者的網路，並非單純地在同一網路當中，我們將一個封包所經過的路程，稱為一條端對端路徑（End-to-End path），端對端路徑包含了許多網路元件，例如接取網路和核心網路。

以預算為基礎之概念就是，我們可以依據網路元件的能力，將使用者所要求之頻寬、服務品質以預算方式分配在這些網路元件上面。透過如此預算分配的方式，由網路元件負責提供部分的品質保證。此係一個實際網路能否成功運作之關鍵因素。

雖然以預算為基礎之管理架構無法達到最佳的整體資源運用效率，但是卻能大幅減低管理複雜度。所以，BBQ 架構將全面使用 budget 的方式，將管理權責以最佳方式分配到各個網路元件。

3.1.3 路徑定義 (Paths Definitions)

封包所行經一連串的節點和鏈結，就是所謂的路徑。一般來說，在封包交換網路當中，封包並無事先定義的路徑。不過，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根據由路由模組為路由器設計的靜態路徑，也可能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，將更容易確保服務品質保證。

因此，BBQ 將利用規劃具服務品質保證的路徑，以達到服務品質保證。BBQ 採用透過分層分權的方式，各層元件各自規劃該層級的資源成一路徑片段，提供給上層元件規劃成較長之路徑片段。Short Path 為一穿越某一個 Core Network 且提供服務品質保證之路徑，由核心網路路徑規劃元件，PPA (Path Planning Agent)負責規劃；Long Path 為一穿越 Backbone 且可提供服務品質保證之路徑，由由核心網路另一個路徑規劃元件，LPPA (Local Path Planning Agent)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，由接取網路元件 global ACA(Admission Control Agent)負責維護。如此，每一個核心網路負責兩個路徑規劃任務：一是負責將各自網路內鏈結(link)組成一條條附有品質保證的 short path。另一則是負責為連接到各自網路的接取網路所產生的訊務規劃 long path(規劃時需與其他核心網路協調)。

3.1.4 承載服務架構 (Bearer Service Architecture)

每當一個封包需要傳遞至目的地時，封包所行經的子網路均需要為該封包提供承載服務。一般來說，全 IP 網路需要三類基本的承載服務，分別為 Entrance Stub network，Exit Stub network，和 Backbone。一條 short path 由一個核心網路承載，一條 long path 由幾個核心網路共同承載，一條 End-to-End 則由以上這條子網路共同承載。

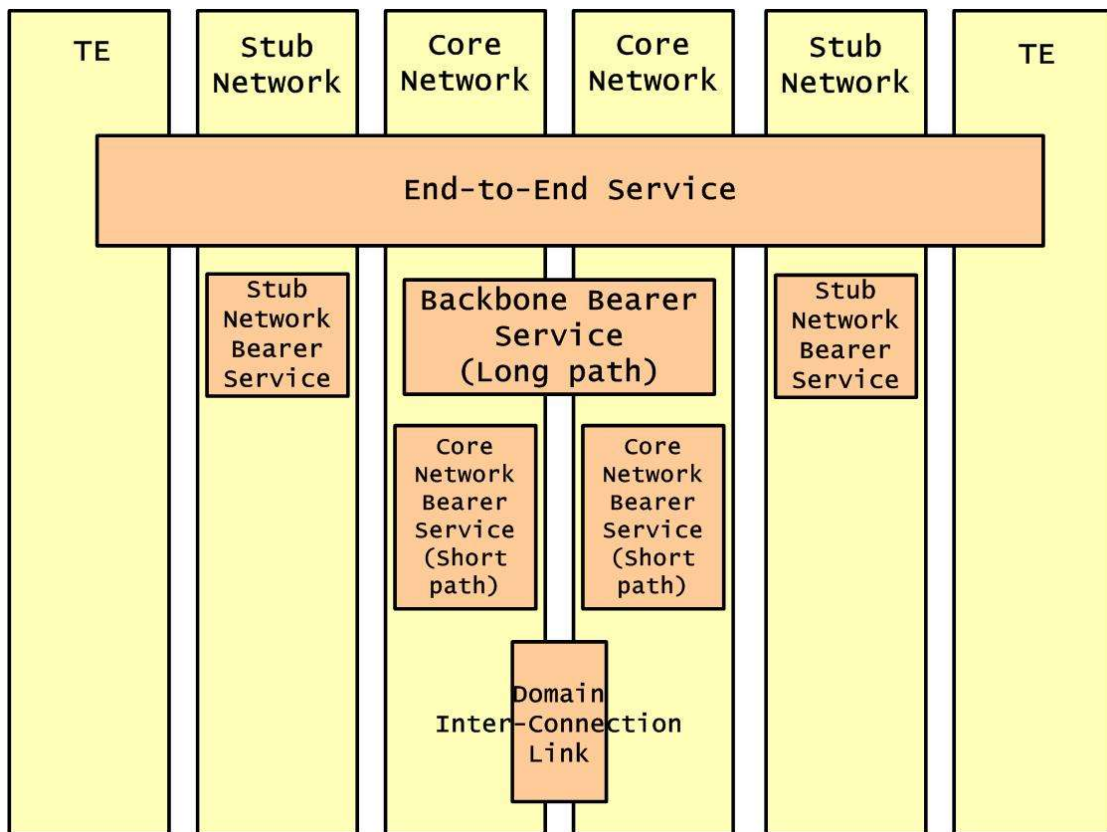


圖 3.2：端對端承載服務

由於規劃之需求，我們需要一個單一數值具可加性的品質量度，此量度由使用者之品質期望轉換而成，也可轉換成低層網路之品質參數。我們假這網路營運者可以自行定義轉換之公式。

3.1.5 服務品質熵數 (Quality Entropy)

服務品質熵數 (Quality Entropy) 為一與服務品質高低相反之數，此值愈小則服務品質愈高。服務品質熵數可由營運者根據自身需求定義，通常可包含 delay time、jitter、packet loss 等品質參數，最簡單者可將此熵數訂為 delay time。本研究的各種品質相關機制均假設網路營運者應提供品質熵數之定義。

使用者對服務品質之定義與網路提供服務品質之定義不一定一致，例如，UMTS 為使用者所定的 class 與 DiffServ 所提供的 class 就不相同，於網路規劃時莫所適從。本研究所定義的服務品質熵數可作為規劃時的一個通用指標，使得規劃方法及工具具高度適用性，適用於各種不同的服務要求。

在使用本研究提出的方法，須將使用者需求轉換成服務品質熵數，再利用本研究所提供

之方法規劃品質政策。如圖 3.3：

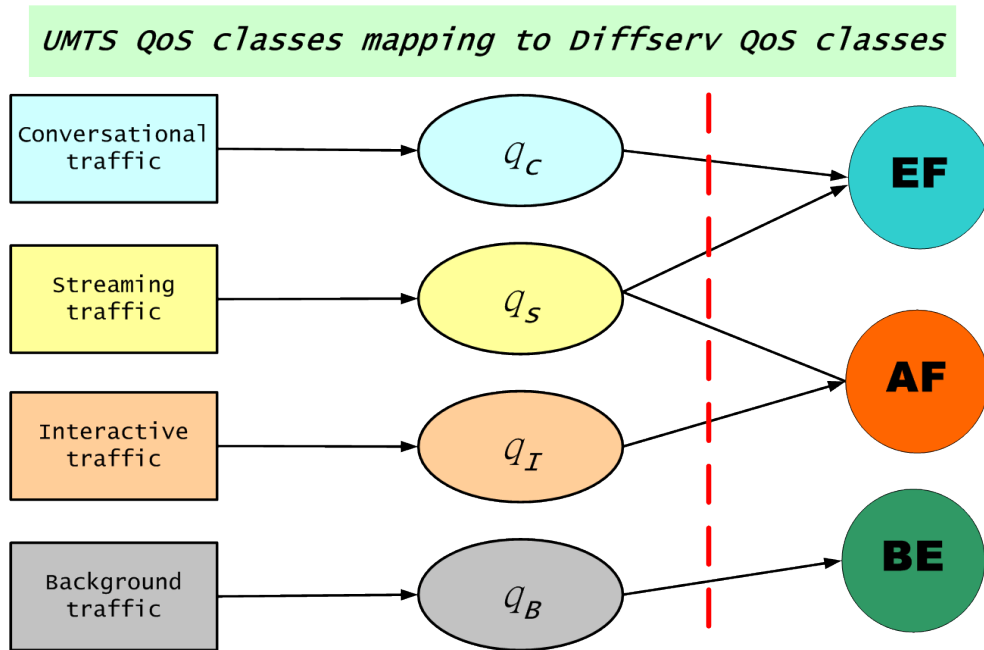


圖 3.3：服務品質熵數之對應

左方矩形代表服務等級（不一定為四類，可依照 operator 所欲提供的服務類型增減），中間橢圓形表示服務品質熵數（Quality Entropy），每類的服務均對應到一個服務品質熵數的範圍，而右方的橢圓型則表示 network layer 的 QoS 管理技術。舉例來說，Conversational class 所對應之服務品質熵數為 q_c ，代表該 class 所需要之服務品質保證，又對應到 Network Layer 的 DiffServ 之 EF class 來管理。所以，我們提供能滿足 q_c 之資源，作為供給此 class demand 之資源。

3.1.6 即時資源分配與預先資源管理 (Pre-Planning vs. On Demand Allocation)

資源之管理可使用預先分配法或即時管理法。即時管理用於 QoS 中較著名的協定為 RSVP。在即時管理的方式之下，允入控制元件(Admission Controller)對於可用資源的掌握較少，只在訊務要進入網域時才臨時去向資源配置元件即時提出資源配置要求並選擇繞徑，以此做允入控制。這樣方式之好處是不會有資源配置過量造成浪費，且進入網域的訊務都可以得到一定的品質保證，但是當網路上的流量逐漸增大，繞徑及資源管理訊息會隨著成長，漸漸成為 router 管理上的一大負擔，並且尋找路徑的 overhead 會對網路造成相當的影響。對於路徑尋找，必須耗費相當的運算時間才能求得好的路徑，因此對於時間敏感度高的訊務並

不適用即時的繞徑運算。綜合以上，即時管理在實際執行上易受到較多限制，不適合大型網路。相較於即時管理，預先分配可容忍較高的計算時間，但受限於預測不準確而不易與現實情況結合，如果訊務可預測，那麼可以設計一套好的預測方式以求預先分配最佳化。

在BBQ架構下的核心網路及核心網路所組成的骨幹網路，資源分配採用事先規劃的策略進行分配，事先規劃不同於即時資源分配，可容許較複雜之計算程序以及耗費較高時間以達到資源分配最佳化。網域的資源現僅涵蓋每個link的頻寬。每個Ingress Router紀錄過去各時間點各種需求的累積流量分布，利用這些歷史資料來預測各不同時段所需頻寬，並且經過計算整理成可使用資源，接下來等到實際執行階段依照實際流量做適當分配。

網路上的流量需求並不一定非常規律，預測的頻寬需求可能與實際情況出現誤差。對於預測形成的誤差，除盡可能提高預測之準確度之外，可以採取適當方案以減少因估計誤差所導致的資源浪費。

3.1.7 集中式配置與分散式配置 (Centralized Allocation vs. Distributed Allocation)

本研究歸納了幾種不同的資源配置實作方式，分為集中式配置與分散式配置和混合式的資源配置方法。前者由資源配置元件統一配置網域資源，後者則是將整個網域資源先分配給網域內的各個Ingress Router自行運用。

集中式的資源配置由資源配置元件統一配置網域資源，例如：由路徑規劃的元件參考歷史的訊務資料，規劃出附有品質保證及頻寬上限的路徑後，將路徑視為資源即依照路徑規之起點分配給各個Ingress Router。

這樣的方法最大的優點是簡單且容易實行，且資源配置時的考量是整個網域中的資源，可以獲得較高效率的規劃。但頻繁的資源配置由單一元件統籌也會有許多缺點：

- 集中式的規劃方式易造成解決公平性的問題(fairness problem)。系統為了達成最佳化目標(如獲利最高)，可能不能顧及公平性造成部分Ingress Router沒有獲得公平的對待。
- 中央集中式資源分配雖然可以因為擁有全部網路資訊作統一規劃而有較好的效率，但也因為問題較大通常無法得到最佳解。

- 由於對訊務的預測有其不準度，需要即時資源配置來輔助，在執行時期，即時的資源配置對中央資源分配元件造成可觀負荷，難以擺脫即時分配法之原有缺點。

因此集中式資源配置的架構因應訊務流量異常變化的能力較差。

分散式的方法則將資源配置交由幾個不同的元件負責。例如：由 Ingress Router 預測某一段時間內可能的資源需求量，依據預測的結果對資源配置元件要求資源配置，資源配置元件綜合所有需求後將資源分配給各個 Ingress Router 自行運用。當有新的資料流欲進入網域時，Ingress Router 只需檢視自己手上所擁有的資源，再決定是否讓訊務進入網路，不用再由資源配置元件處理訊務的資源配置。

分散式路由方法是將路徑規劃的工作分散至各別 Ingress Router，每個 Ingress Router 負責規劃由自己進入的資料流路徑。分散式的方法減輕了中央統籌元件的負荷，解決在規劃路徑上所造成的公平性問題，而且在發生訊務異常時，可以由 Edge Router 作小幅度的內部調整，可以增加應付訊務異常之能力。但分散式的架構較為複雜，且整體的資源使用率會較低，此點可藉混合式彌補。

混合式的方法是以集中式的方法作初步規劃，再以分散式的方法做細部調整，在執行時段也採用分散資源分配。如此，可增加集中式資源規劃的效率提高最佳化能力，減低公平性問題，又可避免即時資源分配的問題。

在大型網路的資源配置需要穩定，且為了減少資源的浪費並使用的最佳化的方法妥善分配資源，應減少即時配置資源的比例，並將資源配置的責任分散到各個 Ingress Router 上，雖然方法複雜且消耗許多運算資源，但事先規劃的配置方法可以承受較高的運算負荷。

3.1.8 需求預測

BBQ 架構規劃網域的資源必須依靠精準之需求預測，以過去具有相關性的歷史資料預測未來某個時段的訊務需求。在規劃的時間點(planning time，例如每天晚上 12:00)、規劃的時間長度(length of time period，例如以一小時為單位)與規劃範圍(planning cycle)都有彈性可供選擇。在時間長度部份，根據 BBQ 網路在不同環境下，對於流量變異性較小之網路，我們可以取較長的規劃時間長度，例如以每小時為一個單位，將一天分成 24 個時段，再以一天為一個規劃範圍。對於流量變異性較大之網路，則可以縮短規劃時間長度，例如以半小時

為一個單位，將一天分成 48 個時段進行規劃。當然也可以採用不規則的時間長，尖峰時段較短而離峰時段較長。

規劃時間長度的調整在 BBQ 架構下為需求預測提供一個富有彈性的選擇，讓系統業者根據該網路狀況進行調整，較長的規劃時間長度可以減少事先進行預測與配置資源的流程次數，適合用在流量相對於時間的變異性較小之網路；而較短的規劃時間會耗費較多次的預測與資源配置流程，但是在流量相對於時間的變異性較大之網路可以較為精準的預測需求。

決定了規劃時間長度(length of time period)後，應在適當的時機對於下個規劃範圍(planning cycle)，針對規劃範圍內各個 Concerned Time Period (CTP，欲規劃資源需求的時段)進行需求分析並進行資源配置(例如前天晚上進行隔天每個規劃時間的需求預測與資源配置)。對於每一個 CTP 都可以定義一組 Reference Time Period (RTP，預測 CTP 資源需求之參考時段)，RTP 為 Ingress Router 紀錄的歷史資料中，與該 CTP 有相關性的歷史資料，舉例而言，如果 CTP 為本週週一早上 8 - 9AM，則預測中所參考的 RTP 可設為之前所有週一早上 8 - 9AM。RTP 的流量需求紀錄，可作為 CTP 需求預測的參考，根據需求分佈的情況來調整資源批購方針。

3.2 管理系統架構 (Management System Architecture for BBQ)

3.2.1 BBQ 管理系統假設 (BBQ System Assumptions)

BBQ 管理架構為了簡化設計的複雜度，因此我們將有以下的假設。第一，quality entropy 為由營運者根據自身需求定義之函數計算的單一數值評估指標。第二，本研究假設 quality entropy 為一具可加性之數值，可透過預算分配的方式，將服務品質熵數分配在各個網路元件之上。第三，BBQ 假設於核心網路等大型網路當中，具有 periodical traffic pattern。第四，BBQ 將不同服務等級之資源事為獨立的網路資源分開規劃。

3.2.2 分散式分層管理系統 (Distributed Management System Hierarchy)

為了降低管理的複雜度，BBQ 採用分散式階層式相當於 3.1.7 所述的混合式管理系統，負責規劃資源的使用以及提供服務品質的路徑規劃。其主要目標是要讓網路管理者在所擁有的網路資源下，提供最多符合品質的服務。以預先資源規劃的方式在各元件之間事先協調資

源分配，允入控制元件獲得資源後，再整合各項資源形成符合各項 QoS 條件的訊務路徑留待執行時期提供給允入的訊務使用。

BBQ 將服務品質管理依照階層架構做分類。由上而下，端對端服務品質保證協調層 (End-to-End Network QoS Coordination Layer) 負責提供端對端服務品質保證，利用下層元件所提供之資源，規劃 long path 和 End-to-End Path；核心網路資源管理層 (Core Network Resource Management Layer) 負責核心網路之資源管理分配；核心網路資源控制層 (Core Network QoS Control Layer) 負責執行服務品質保證之策略以提供服務品質保證，例如允入控制等。DiffServ 或其他 IP 層網路則負責執行上層元件所規劃出來服務品質管理策略，屬於下層網路技術。BBQ 為具適用行之管理架構，可更換底層之網路技術，目前 BBQ 採用以 IETF 所制定的 DiffServ 為代表。

表 3.1：BBQ 中分層分工架構

層級	作用
End-to-End Network QoS 協調 (每一個 operator 都有一個)	End-to-End 的服務品質控制，End-to-End 的資源和路徑規劃。
核心網路資源管理 (Bandwidth Broker)	Allocate resources of a core-network to the QoS Manager of all resource mediators, e.g. Ingress Router in a Core Network or RRM in a 3G Access Network.
核心網路資源控制 (在每個 Ingress 內) Core-Network QoS Control (One for each Ingress)	Execute QoS policy of core network, e.g. admission control, load control, routing and path selection, packet scheduler, etc.

層級	作用
DiffServ	執行上層資源管理所設定策略。 (Execute the QoS policy designated by the upper layer manager)

3.2.3 管理系統軟體架構 (Management System Software Architecture)

每個核心網路均有一個軟體元件為 Core Network Coordinator, CNC, 內含 Long Path Planning Agent, LPPA, 負責 long path 規劃。而接取網路的網路接取伺服器上則有一個軟體元件為 global ACA, 負責端對端服務品質保證之允入控制。圖 3.4 顯示各元件之關係位置。

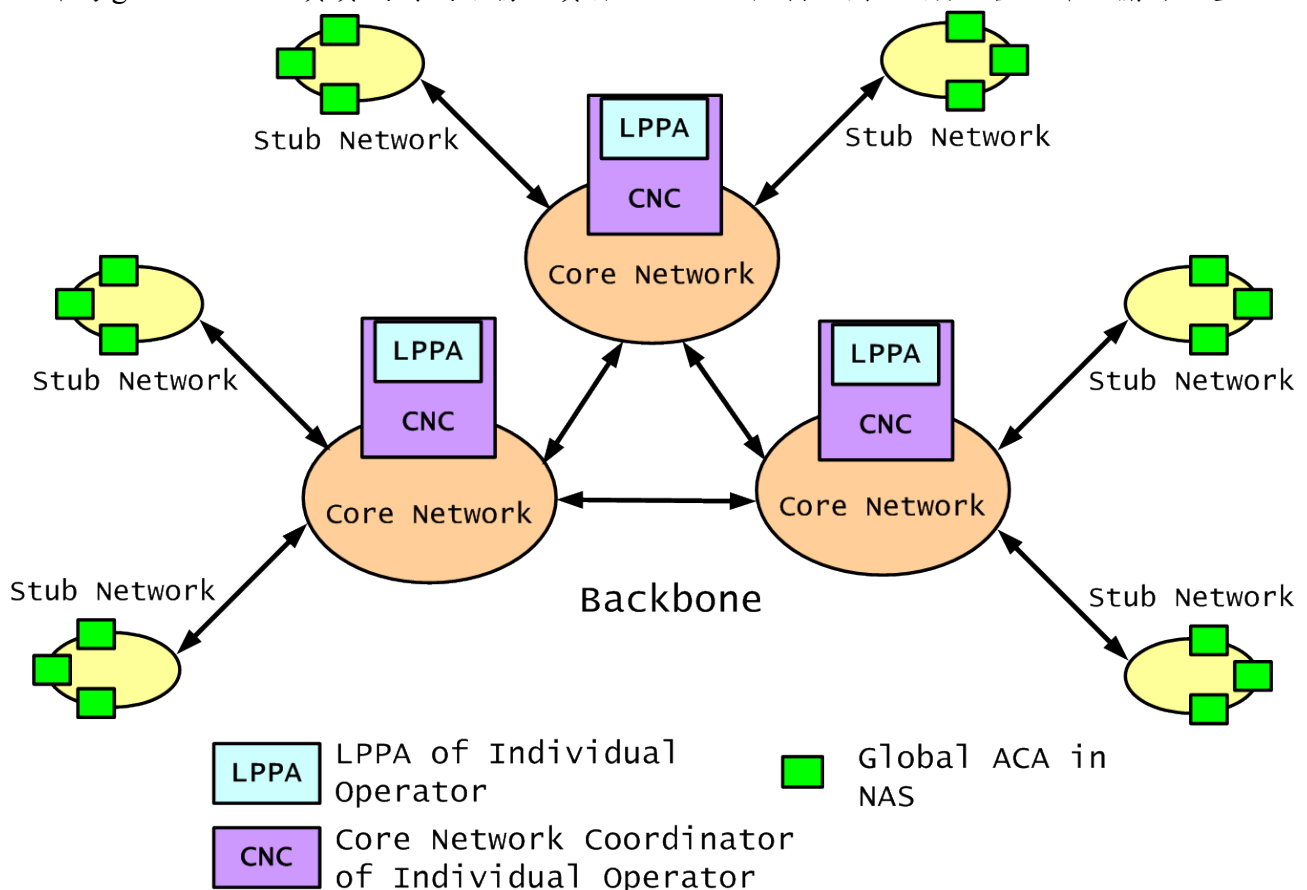


圖 3.4 : BBQ 管理系統軟體元件架構

3.2.4 簡化的端對端服務品質建立流程 (A Simplified End-to-End Path Setup Procedure)

圖 3.5 為簡化的端對端服務品質建立流程。當即時的訊務流要求進入全 IP 網路時，接取網路 Network Access Network 上的 Global ACA 則負責允入控制。由於 BBQ 將採用資源預先規劃，即時分配的管理方式，因此，允入訊務流之前需要先建立具有服務品質保證之端對端路徑。簡單分為三階段：

- 入口接取網路承載服務資源取得
- 出口接取網路承載服務資源取得
- Long path 資源使用權取得。Long path 由數條 short path 所組成，所以此階段將需要取得所需之 short path 資源使用權。

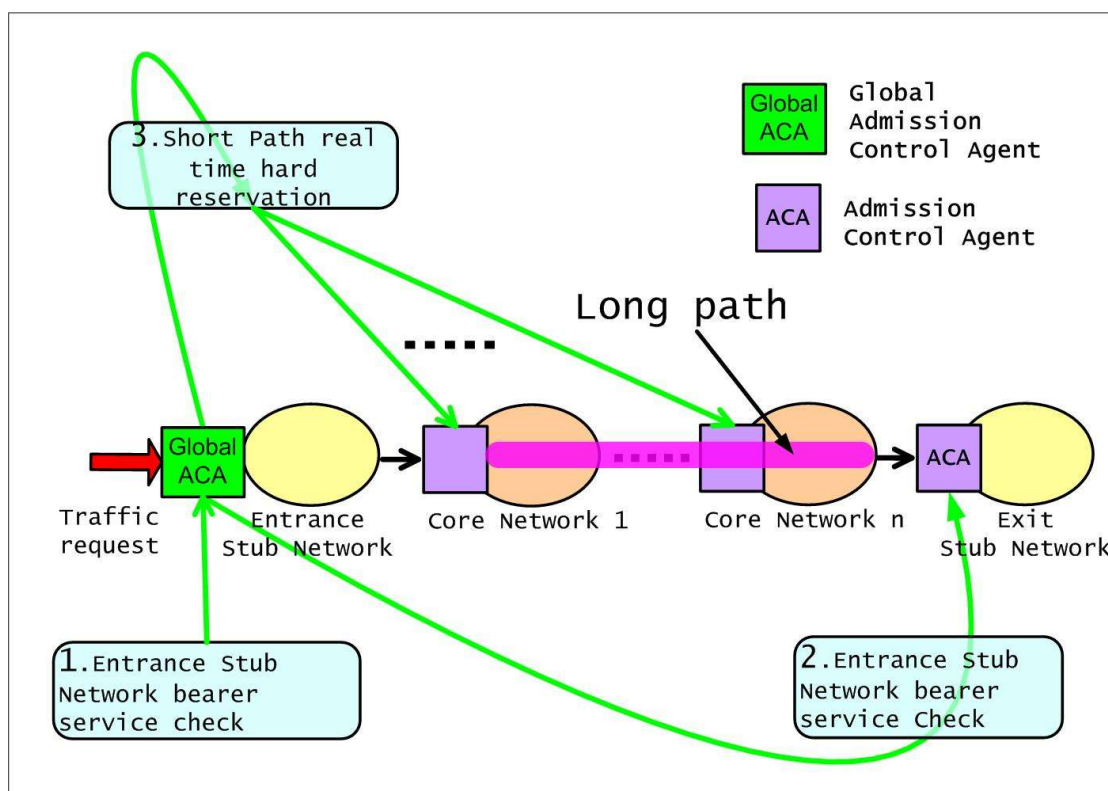


圖 3.5：簡化的端對端服務品質建立流程

3.3 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。

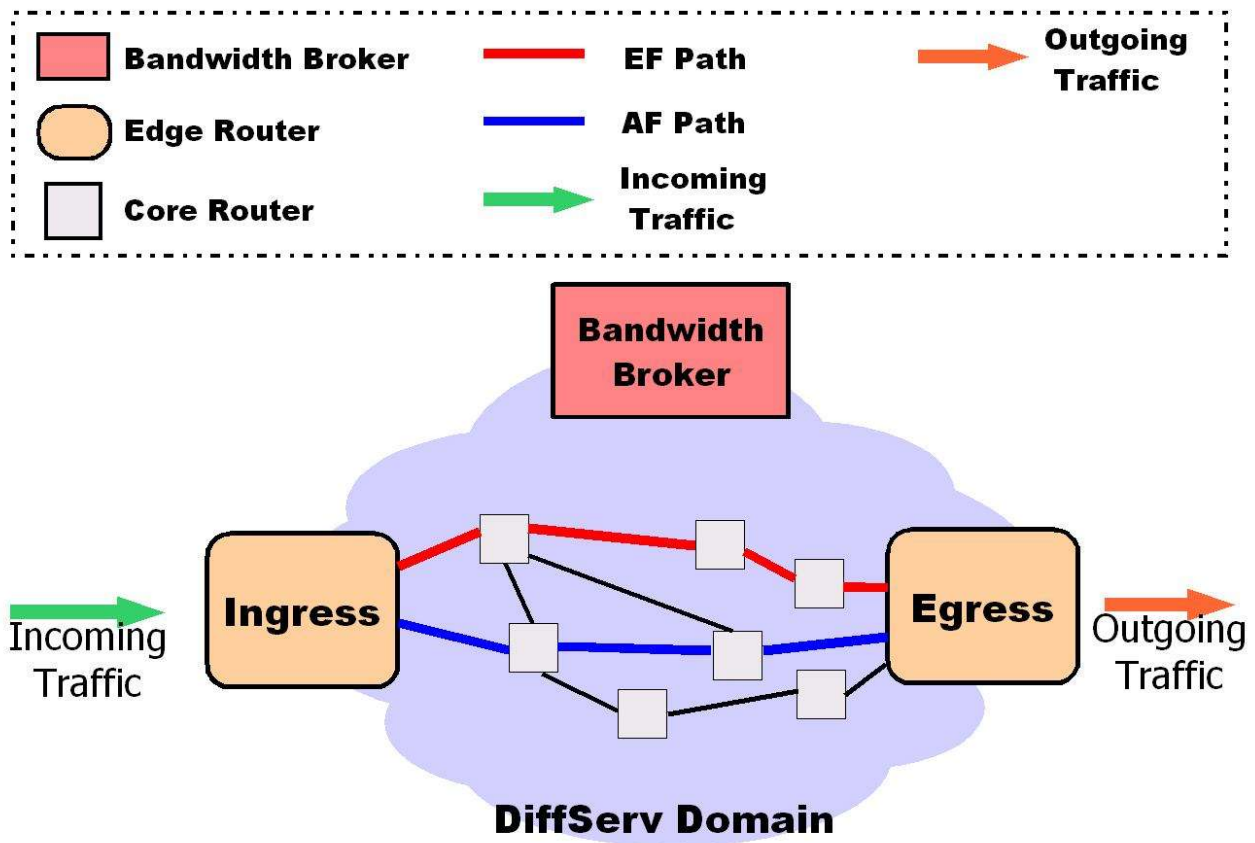


圖 3.6：BBQ 架構之核心網路

3.3.1 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。

本研究考量網路資源的使用率與網域內各 Ingress Router 的公平性問題，以分散式搭配集中式的方法路由，並配合上適當的資源配置法，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。

3.3.2 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 LSPPA 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，若允許進入網域，就表示可以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, LSPPA)：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 LSPPA 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。

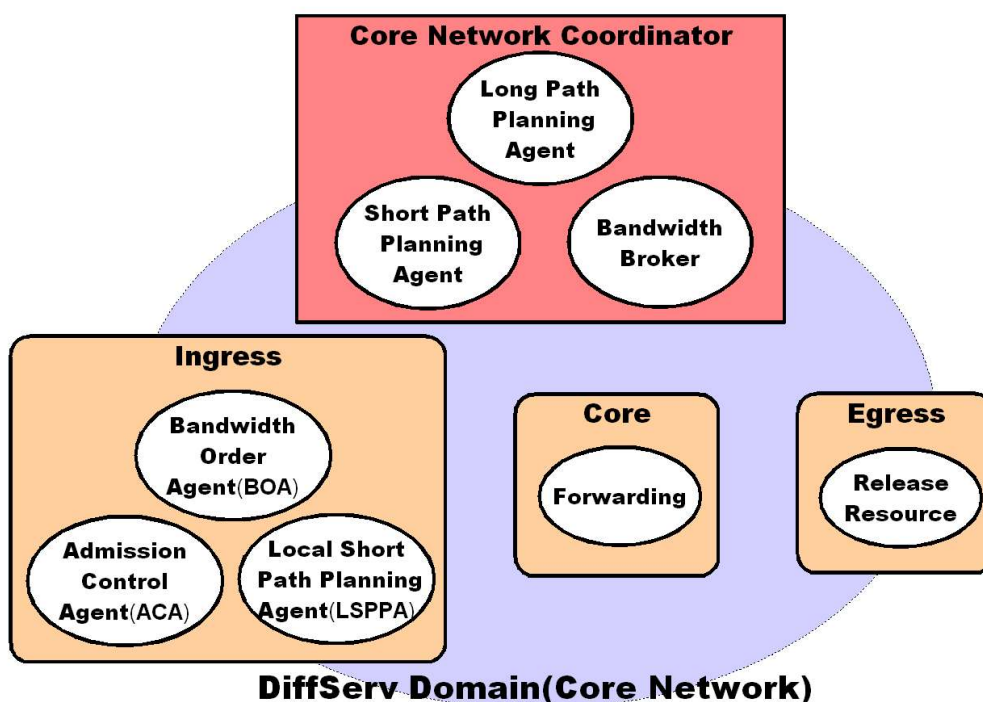


圖 3.7：核心網路軟體架構

3.3.3 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。每一個鏈結的頻寬均由 CNC 中的 Resource Manager 統一分配，BOA 根據預測，預先向 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，整個批發零售的過程如下(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相似的參考時段，根據這一個時段的訊務特性還有先前資源的使用情形，為各個 Ingress Router 某個未來時段資源需求做預測，此一個預測的結果為一個資源需求的分佈圖。此時 Short Path Planning Agent (SPPA)會根據各個 Ingress Router 上的需求分佈圖和核心網路內的頻寬，計算出最適合此時段訊務的一組路徑組，此路徑組是由從各個 Ingress Router 出發具有不同 QoS 品質的路徑所組成，預料將可以滿足該時段中不同應用的品質要求。

- 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，此一個預測的結果為一個資源需求的分佈圖。
- 步驟二. 此時 Global Path Planning(GPPA)會根據各個 Ingress 上的需求分佈圖和 Core Network 內的頻寬，計算出最適合此時段訊務的一組路徑組，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。
- 步驟三. GPPA 會在將此整體路徑組轉換成每一個 Ingress 在各 Link 上的資源需求，其結果也是一個各個 Ingress 的資源需求分佈圖，然後此資訊交給 BOA(Bandwidth Order Agent)，BOA 會根據此一資訊向資源的管理者 BB 中的 Resource Manager 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 會降低其要求直到 BB 給予所須的資源。

- 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 PPA 來做路徑的規劃，PPA 會利用這些資源規劃出適合的路徑組，

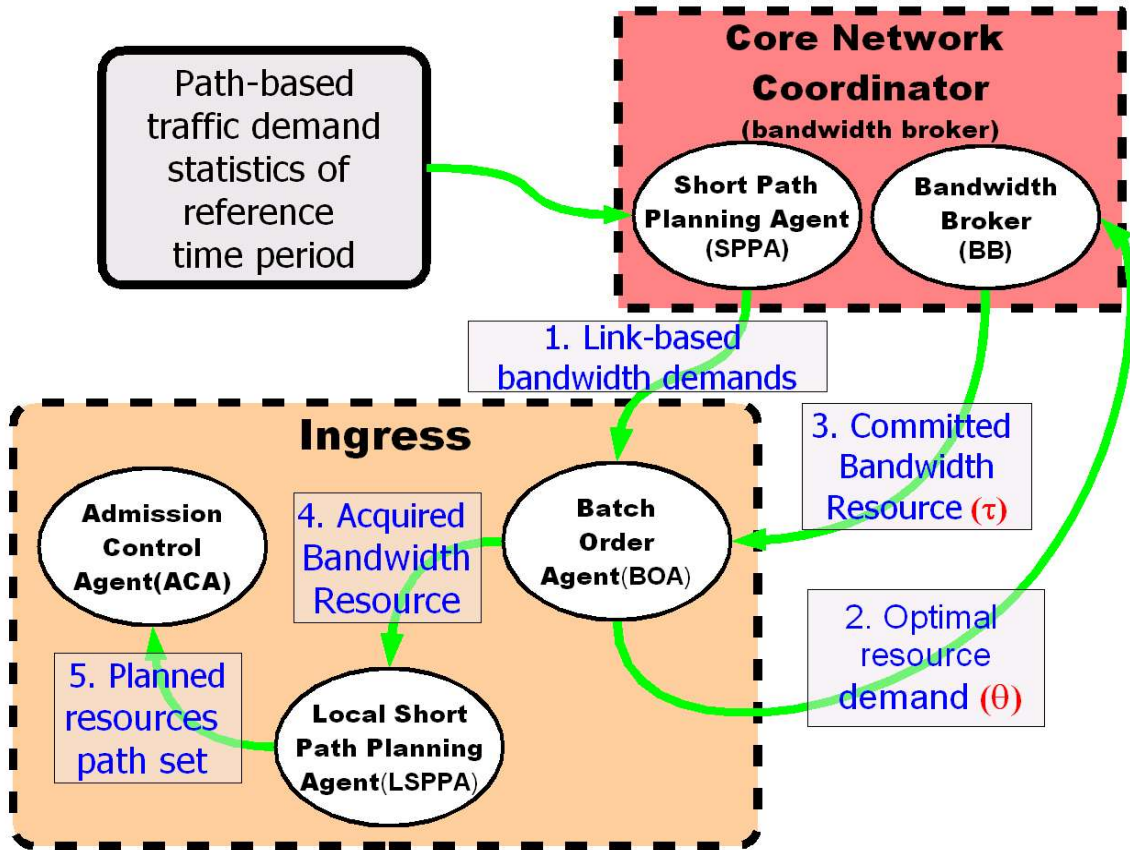


圖 3.8：核心網路中資源分配流程

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：

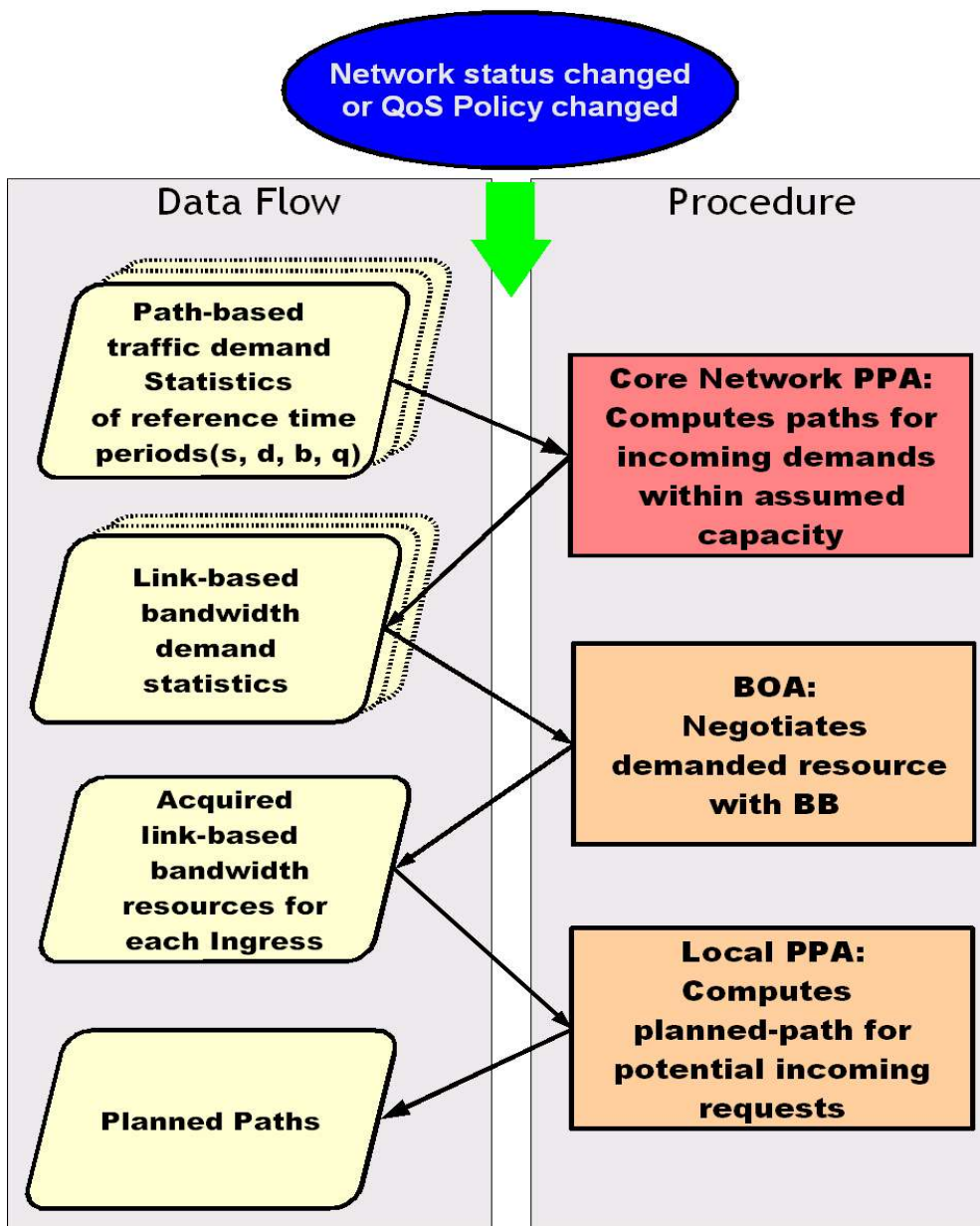


圖 3.9：資源規劃流程

3.3.4 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：

3.3.4.1 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。

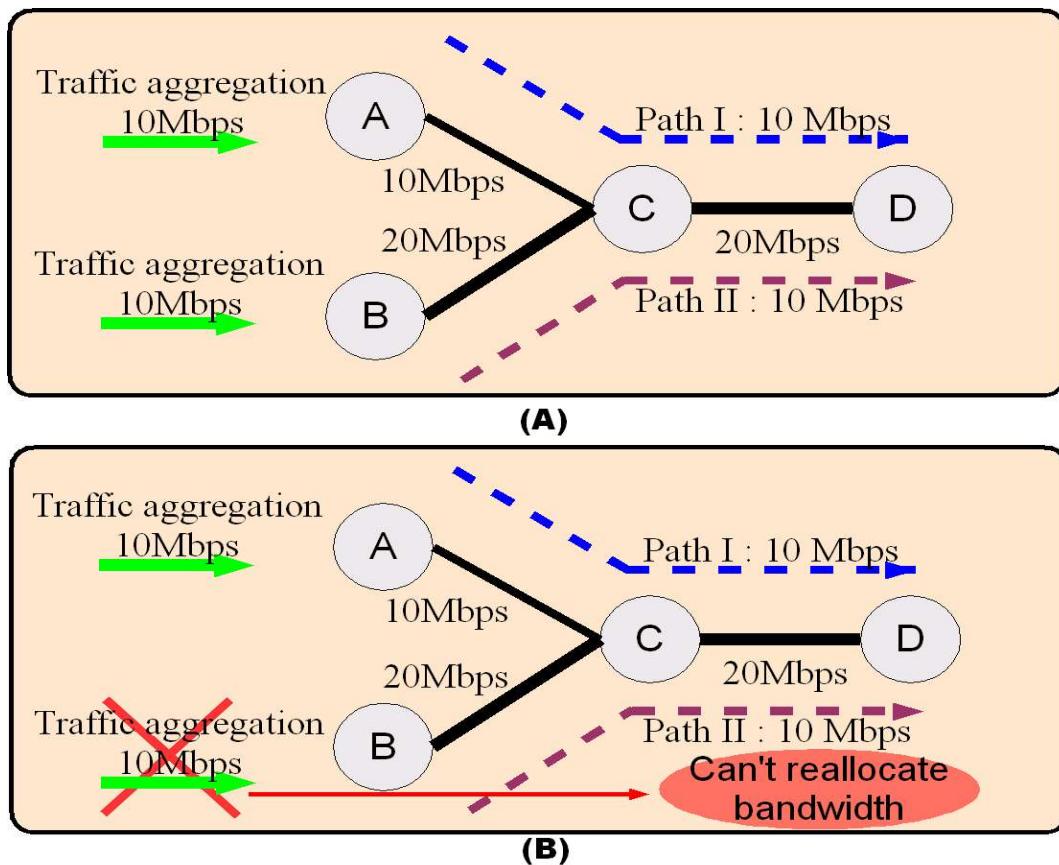


圖 3.10：不同 Ingress 之間資源無法彈性調配

3.3.4.2 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，我們無法將 Path II 中沒有使用到的頻寬資源，重新分配給 Path I 使用，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。

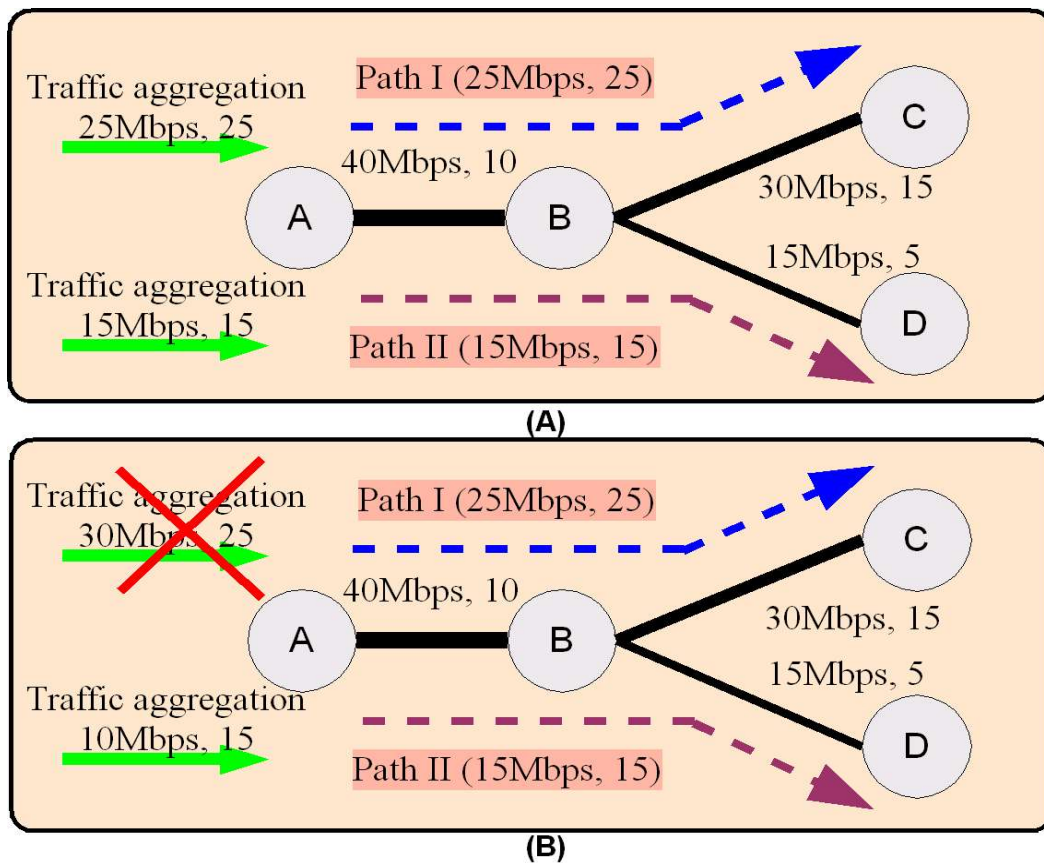


圖 3.11：不同路徑之間的資源無法彈性調配

集中式的方法簡單易行，卻可能衍生網域內各個不同 Ingress Router 間公平性的問題；分散式的方法中，也可能產生如此的問題，但卻較容易解決，可是相對的也較複雜。本研究考量網路資源的使用率與網域內各 Ingress Router 的公平性問題，以分散式為主，並配合上適當的資源配置法，規劃成一個個完整路徑，給 Ingress Router 允入使用。

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 中資料的傳送需要建立連線，因此在封包網路中，並無路徑的概念。

不過，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根據由路由模組為路由器設計的靜態路徑，也可能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，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，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。

規劃路徑	管理元件	路徑能力
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，在據以計算出每一個 Ingress 在每一個鏈結上的頻寬需求。最後將此資訊將交給位於每一個 Ingress 內的 BOA，作為向 BB 批購資源時的依據。

而每個 Ingress 之中另外還有 Ingress PPA 的元件，其作用則是把 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 作為品質評定的標準。

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	協調具有 End-to-End 的服務品質保證的 Path。
SPPA	位於 Core Network Coordinator 中	負責將規劃出來的路徑頻寬需求，轉換成鏈結上的頻寬需求。
Local SPPA	位於每一個 Ingress Router	主要的功能在於將所擁有的頻寬資源，規劃成不同服務等級的路徑，供應 Run-time 時的 Request 需求。

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，為了簡化起見，我們假設 Quality Entropy 具有可加性(additive)，如常見的服務品質保證參數 - 傳輸延遲 (Delay)而言，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，我們假設其可以達到所設定的 Quality Entropy，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[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 (Cooperation between LPPA and Short Path Planning Agent)

在 BBQ 架構中，核心網路只是整個 End-to-End 的一個部分，因此核心網路內的資源規劃也需要考慮到整個 End-to-End 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規劃 End-to-End 本身核心網路負責處理之訊務的最佳路徑，此預訂的頻寬會成為 Short Path Planning Agent 計算所需頻寬的另一個參考依據(主要的依據是核心網路內的歷史統計資料)，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：

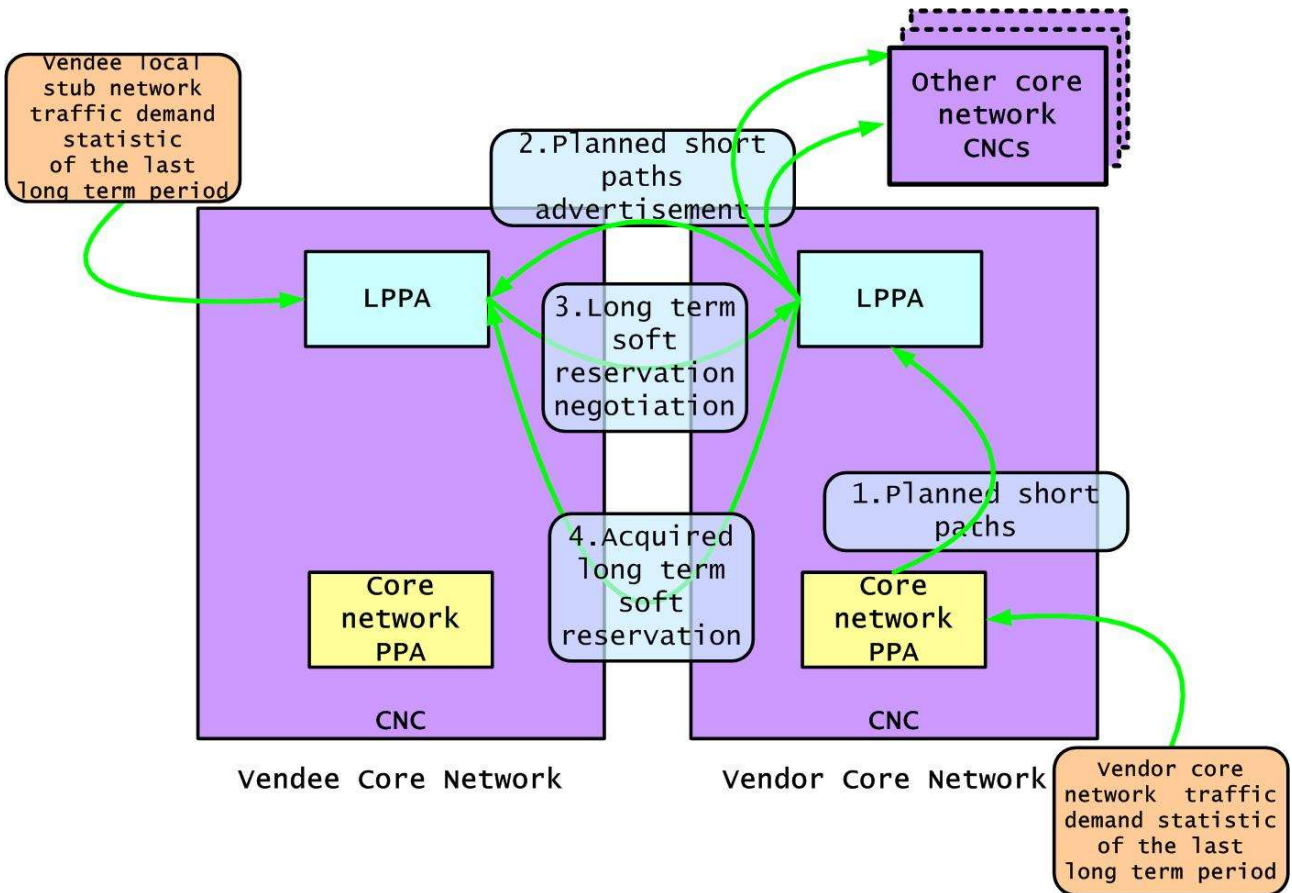


圖 4.1、路徑規劃元件之互動程序

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同一參考時段的所出現的訊務流量與型態。此統計資料是以過去此網域內，同一個參考時段中所出現過的訊務統計之後所得的結果，會記錄每一個 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，鏈結上的資源除了其上可以使用的頻寬以外，還要知道此一個鏈結所可提供的服務品質熵數為何，以本研究目前所著重的傳輸延遲而言，在 SPPA 規劃路徑的時候，需要知道通過每一個鏈結時，所可能遭遇的傳輸延遲為何，這樣規劃出的路徑才可以準確的提供訊務所要求的品質。有了以上兩項參數，SPPA 會以本研究所提的路徑規劃方法，為整個核心網路下一個即將來臨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。

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，由 BOA 向 Bandwidth Broker 批購鏈結頻寬資源，所以統計資料中以進入點、離開點和服務品質需求為分類的流量需求必須轉換成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。

而 BOA 在批購頻寬的時候，不能只參考一個時段的流量，必須參考多次訊務的流量情形，以需求量的統計分佈作預測才會較為精確，我們從所有 Reference Time Period 中挑出最近一次的訊務流量統計，讓 SPPA 執行計算，求得一組路徑作為基礎，然後我們會輸入以往相同 Reference Time Period 中不同的訊務資源需求量經過這些路徑，得到一個鏈結上的資源需求量之統計，此即為 BOA 計算最佳的頻寬訂購量的基礎，以上為 Path Planning 第一階段的任務。

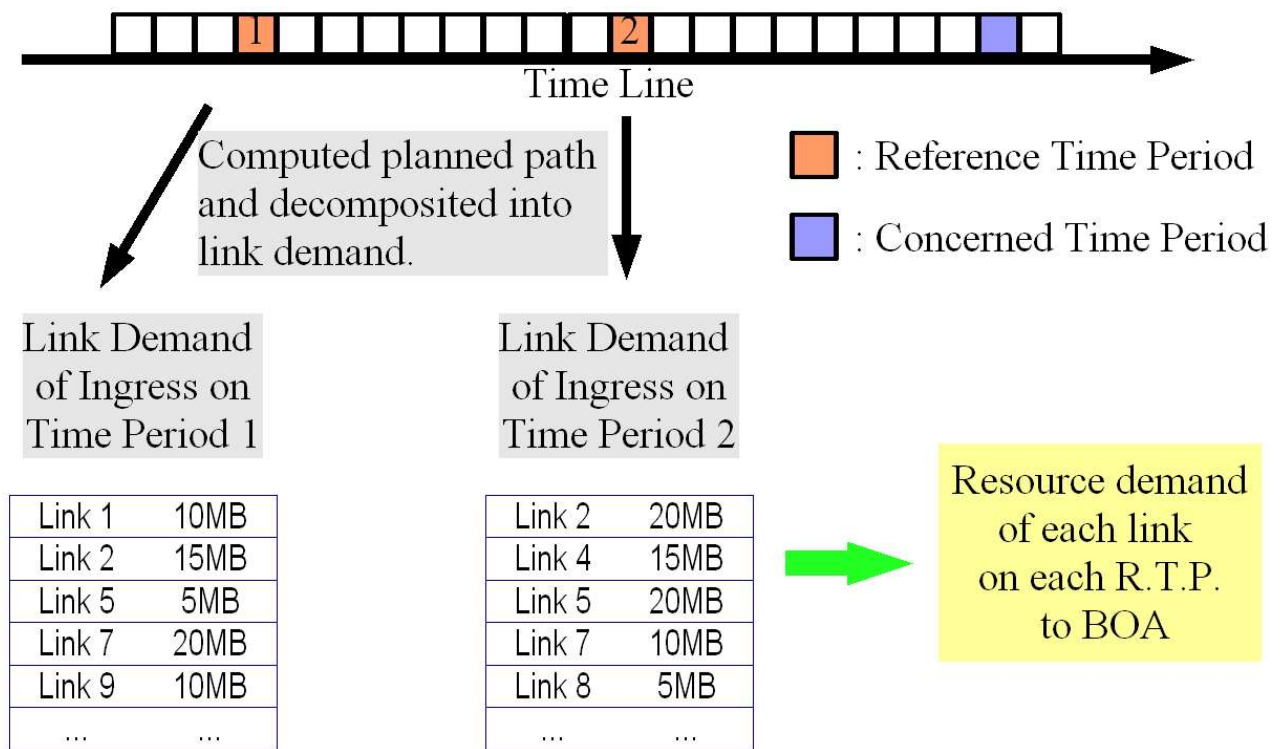


圖 4.2：路徑規劃轉換鏈結頻寬需求之說明

當 BOA 將真正購得的頻寬交給在每一個 Ingress 中的 LSPPA 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，因此位於各個 Ingress 中的 LSPPA 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經規劃所參考的是最近一次 Reference Time Period 的統計資料，在 LSPPA 中採用的是過去幾次 Reference Time Period 的平均值，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 	<p>一組符合最符合網路狀態的路徑組，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。</p>
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 的統計訊務量 	<p>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。</p>

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，來衡量此組規劃路徑是否有充分的使用到網路中資源，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 (Problem 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 為路徑規劃所需之符號表說明。

表 4.4：資源分配路徑規劃符號表

Symbols	Definition
$G(V, E)$	A directed graph, G , containing $ V $ nodes and $ E $ directed links; V denotes the set of all nodes, and E denotes the set of all links
v_i	A node; $v_i \in V$
e_k	A directed link $e_k = (v_x, v_y) \in E$, v_x is the start node, 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 a request i
d_i	Destinaiton node of a request i
b_i	Bandwidth requirement of a request i
q_i	Quality entropy of a request i
m_i	Profit earn of a admitted request i
P_i	All possible path satisfied constraints of request i
p_i	A path; $p_i \in P_i$
ω_i	The path set satisfied constraints of request i and selected by our algorithm
Ω	The final path set selected by our algorithm
$B_e(P_{ij})$	$= b_e(P_{ij})$, if links of P_{ij} contains link e $= 0$, otherwise
$L(P_{ij})$	The satisfied ratio of request i with P_{ij}
$q(e)$	Quality entropy of link e

絕對資源保留路徑規劃問題描述如下：給定一組網路的拓樸 G ，由 $\text{node}(V)$ 和 $\text{link}(E)$ 所組成，對於某一個 $\text{link } e_i$ ，會帶有兩個屬性，一個為此鏈結所能負擔的最大頻寬 b_i ，一個為此 link 所能提供的服務熵數 ε_i ；另外有一組 Requests (R)，由許多的 request 所組成，對於每一個 r_i 以下列四個項來描述，起點(s)、終點(d)、所需頻寬(b)和我們定義的服務品質熵數 (q)。我們定義 Profit m_i 是允許某一個 request i 進入核心網路後所獲得的利益，Profit m_i 是由 b_i 除以 q_i 所得(公式 3.1)，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 Profit 是越大的，所以 Profit m_i 可以說是和要求的頻寬成正比，和服務品質參數 ε 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 W 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 各個 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，路徑規劃的最佳化目標即是找一組路徑 Ω ，此組路徑可以使得滿足 request 之 profit 總量為最大(公式 3.2)：

$$\max \sum_{i=1}^n m_i \sum_{P_{ij} \in \omega} L(P_{ij}) = \max \sum_{i=1}^n \frac{b_i}{q_i} \sum_{P_{ij} \in \omega_i} L(P_{ij}) \quad (4.2)$$

此一最佳化的函式有兩個限制式：

一、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，即公式 3.3：

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

二、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，公式 3.4：

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

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。如本文在第二章所提及的，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，而一般皆認為此問題是一個 NP-Complete 的問題，所以對於路徑規劃的問題，我們也假設為 NP-Complete。而對於 NP-Complete 的問題我們可知將無法找到最佳解，因此我們在此提出一個 Heuristic 的演算法來解決路徑規劃的問題。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 各 request 所組成，每一個 request 有四個項，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 。
 - 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 Profit 加總為最大。
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

拓撲 $G'(V', E')$ ，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。

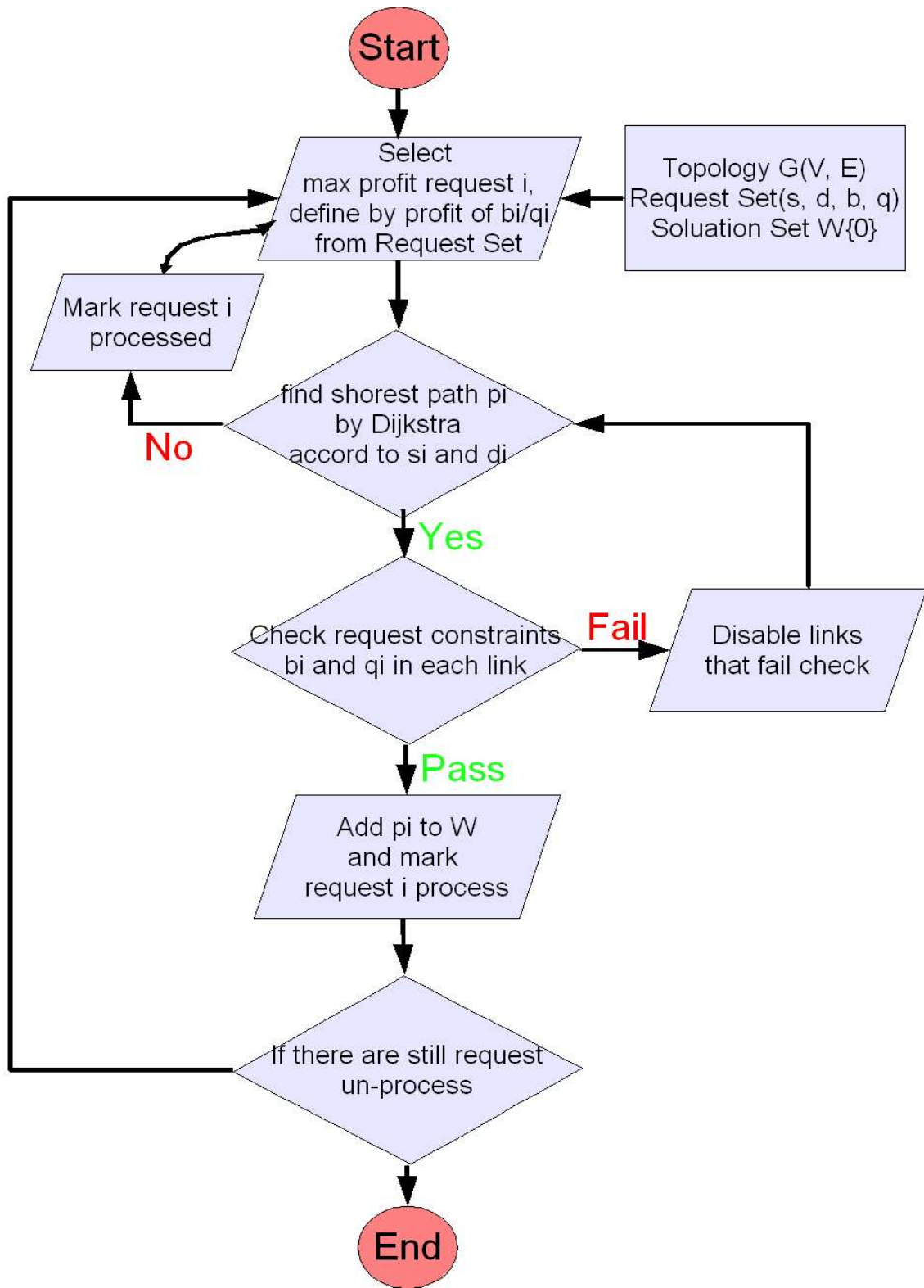


圖 4.3：演算法流程

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 $M * N^2 \log N$ 。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糟糕的情形下，將會被限制在 $(M+k) * N^2 \log N$ 之下，一般說來本演算法的時間複雜度為 $O(M * N^2 \log N)$ 。

4.9 小結(Summary)

在本章中，4.1 節和 4.2 節中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密度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. 獲利密度：獲利除以平均的鏈結使用率即為獲利密度。由獲利密度我們可以得知，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 Quality 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 獲利密度(Profit Density)

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，稱為獲利密度，鏈結使用率計算如下：將計算單一個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，我們可以在越少的頻寬裡，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 Experiment)

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 是本研究以 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、節點的連接率、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。而一個節點連接率的定義如下，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，則此點的連接率定義為 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連接率求平均值的結果，我們設定的連接率從 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至其他網路內的任何一個節點。

為求模擬資料的真實性，我們採用 Network Simulator 2(NS2)，來產生 Run-time 的 Request Data，NS-2 (Network Simulator Version 2) [4,5] 是一個用來提供網路相關研究的物件導向模擬器，可提供完整、趨於真實的網路模擬環境，是個具有擴充性、可程式化事件驅動模擬引擎、支援各種網路通訊協定與路由排程演算法的網路模擬器。發展至今，已經是一個很成熟的網路模擬環境。

對於 traffic 的產生我們採用 NS2 的 exponential on/off 來模擬 traffic 產生的型態，而每一個 request 所要求的頻寬則是呈現 normal 分佈。

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
節點連接率	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。

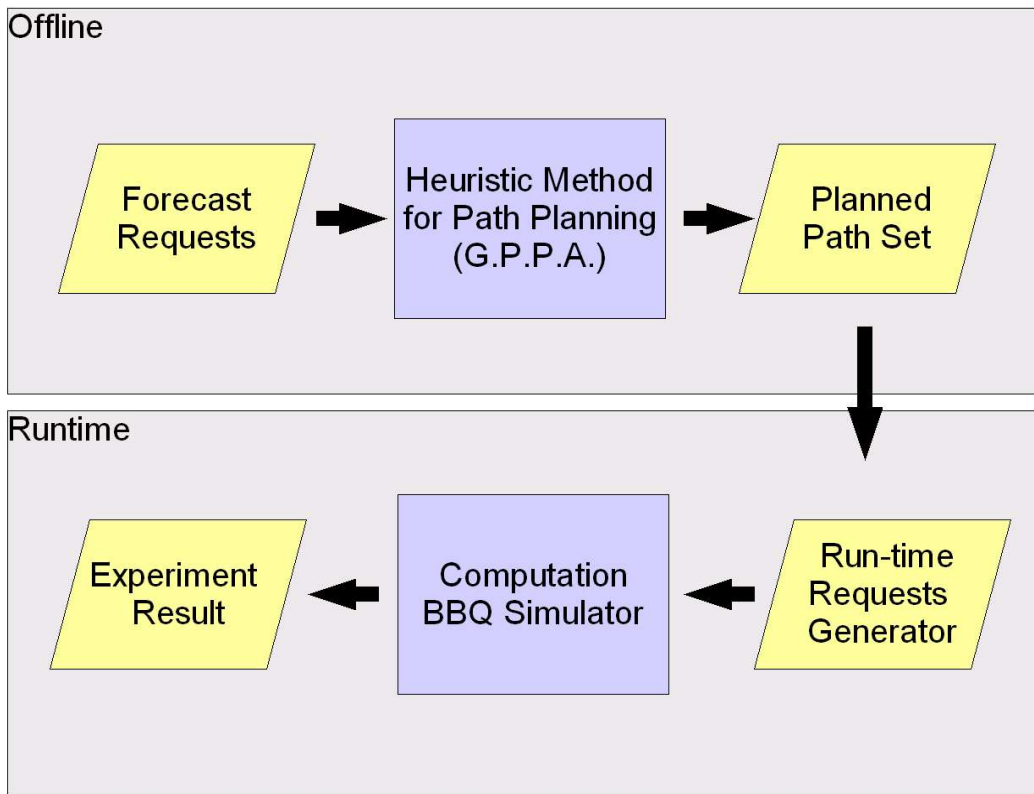


圖 5.1：實驗流程圖

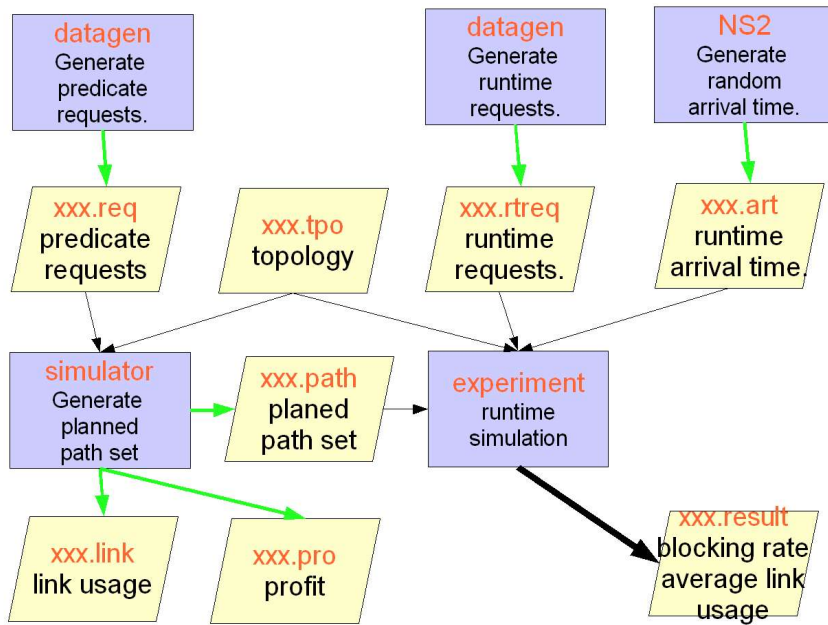


圖 5.2：實驗步驟及資料流向

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
實驗三：網路連接率影響測試	1. 獲利 (Profit)
實驗四：預測誤差影響測試	2. 鏈結使用率標準差(Link Utilization S.D.) 3. 獲利密度 (Profit Density)

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，探討當固定網路連接率時，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連接率，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。而當連接率的增加時，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。而在網路連接率提升至在 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。

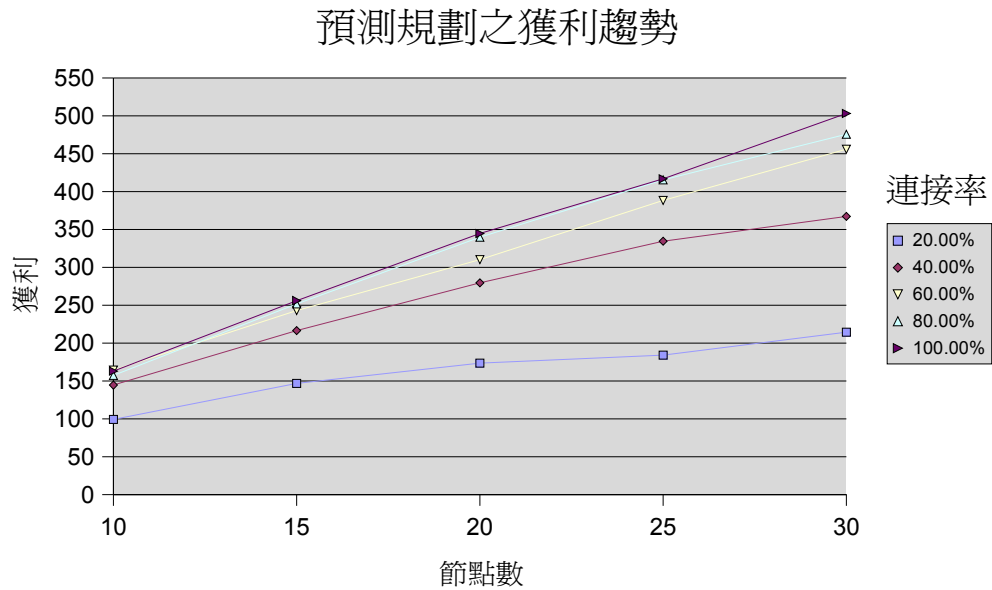


圖 5.3：預測規劃的獲利趨勢

5.3.1.2 獲利密度趨勢

如同上，本實驗以拓樸中的節點數量為實驗變因，探討當固定網路連接率時之影響，不同的是，我們將觀察拓樸中的節點數量對獲利密度之影響。同時，測試當不相同的網路連接率，觀察是否有相同的趨勢。

在圖 5.4中顯示，在我們所觀察的是與預測規劃之利潤密度趨勢，

預測規劃之利潤密度趨勢

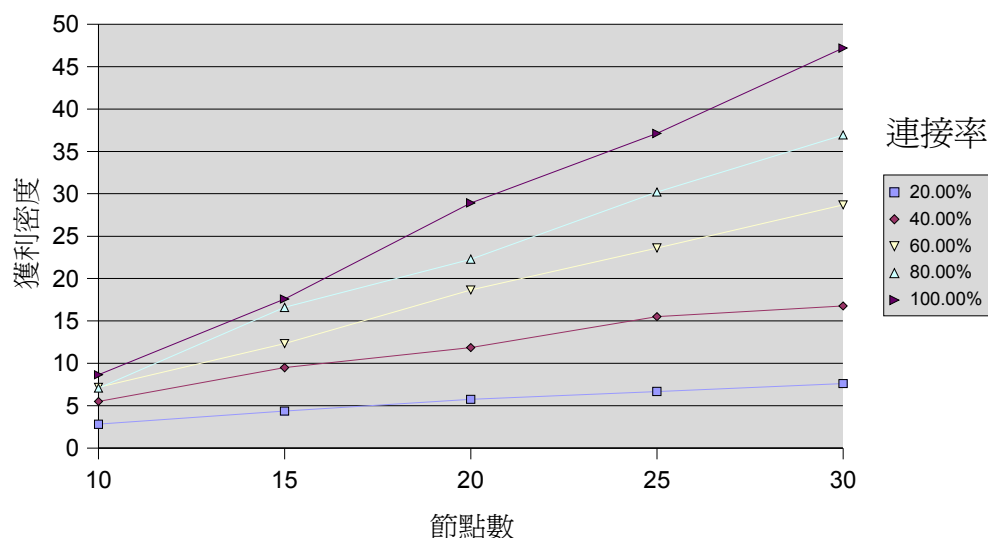


圖 5.4：預測規劃與利潤密度趨勢

5.3.2 實驗二：節點多寡之影響 (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连接率時，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)為连接率固定為 20%時節點數目變化時，獲利情形的變化，圖 5.5(B)為连接率固定為 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。

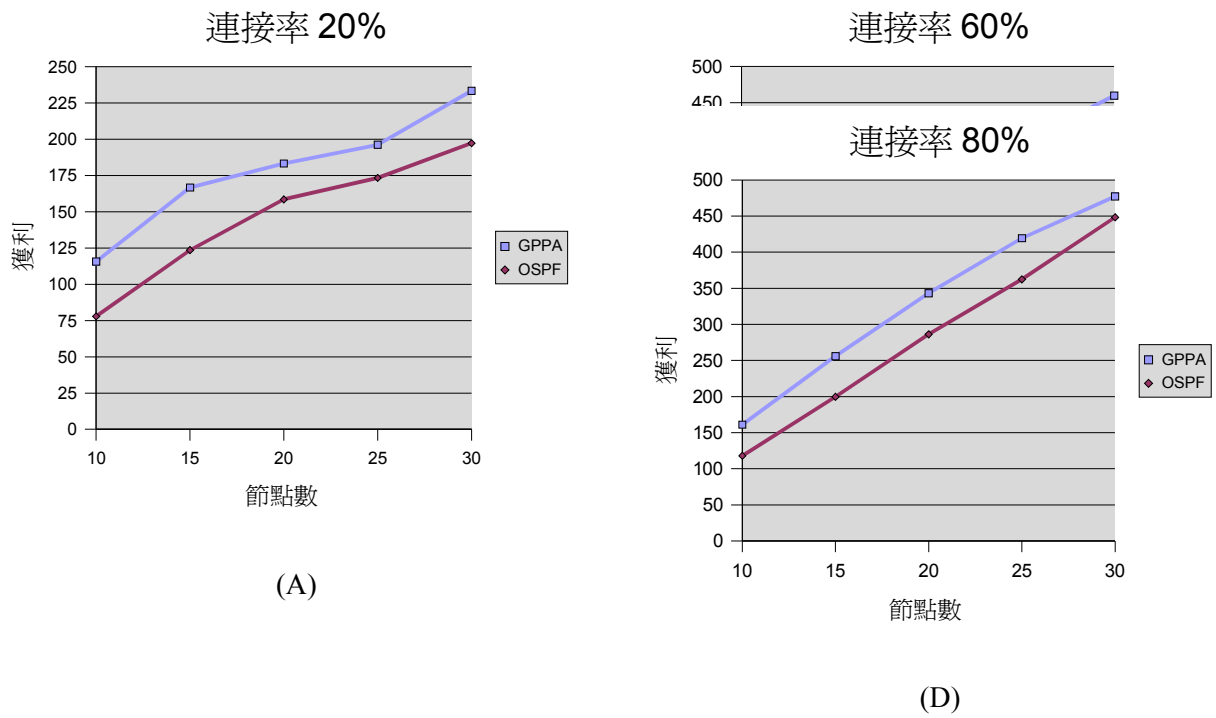


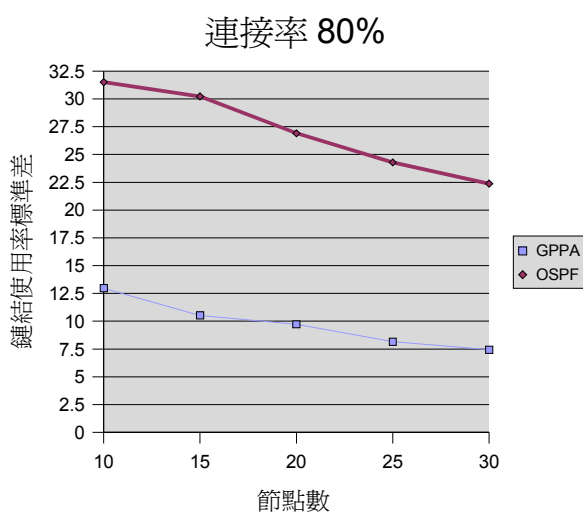
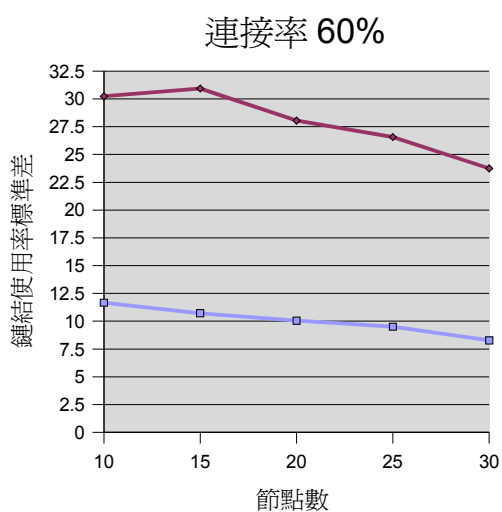
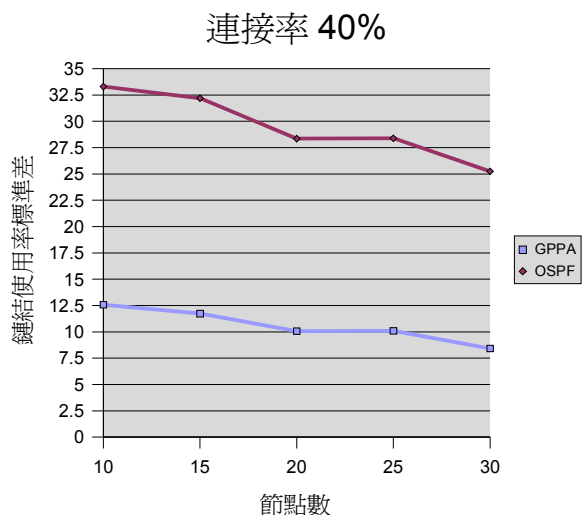
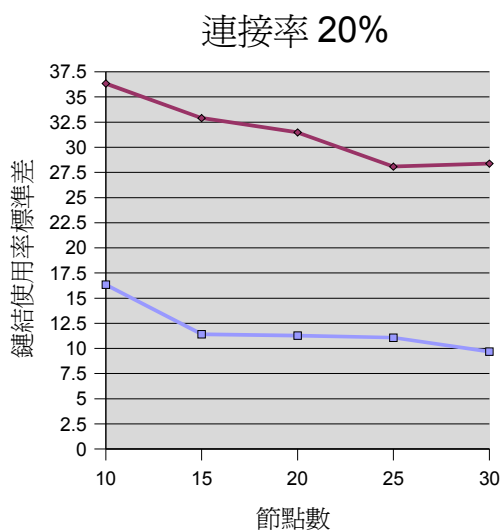
圖 5.5：節點多寡對於獲利指標的影響

(A) 連接率為 20%, (B) 連接率為 40%,

(C) 連接率為 60%, (D) 連接率為 80%

5.3.2.2 拓樸中節點多寡對於鏈結使用率標準差的影響

圖 5.6：節點多寡對於鏈結使用率標準差的影響



(C)

(D)

(A) 連接率為 20%, (B) 連接率為 40%,

(C) 連接率為 60%, (D) 連接率為 80%

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連接率時，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)為連接率固定為 20%時節點數目變化時，鏈結使用率標準差的變化，圖 5.6(B)為連接率固定為 40%時的變化情形，圖 5.6(C)和圖 5.6(D)則分別為連接率 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。

5.3.2.3 拓樸中節點多寡對於獲利密度的影響

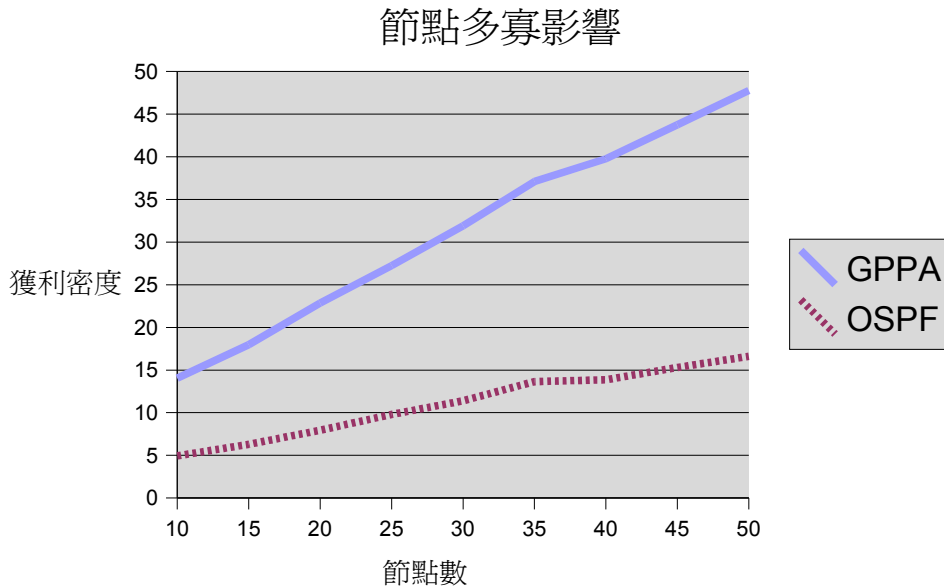


圖 5.7：節點多寡對於獲利密度的影響

在圖 5.7，我們觀察的是節點數目增加時，對於獲利密度的影響，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密度將比 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：連接率之影響 (Connectivity)

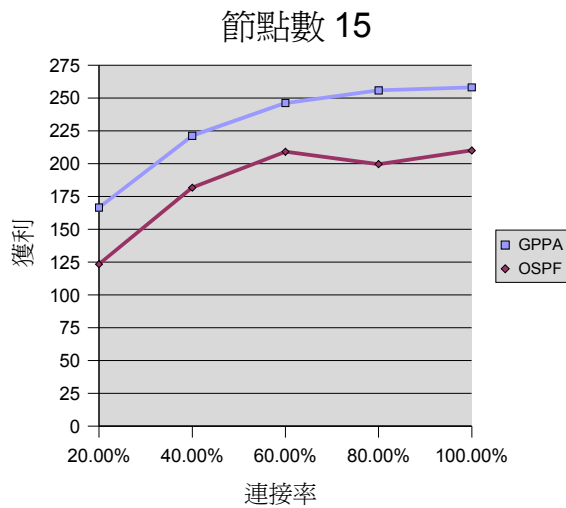
在此實驗中我們將觀察一個網路連接率大小對於演算法效能的影響。

5.3.3.1 連接率對於獲利指標的影響

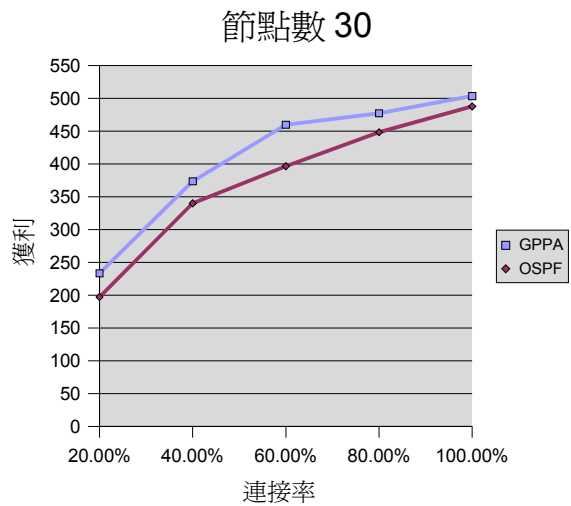
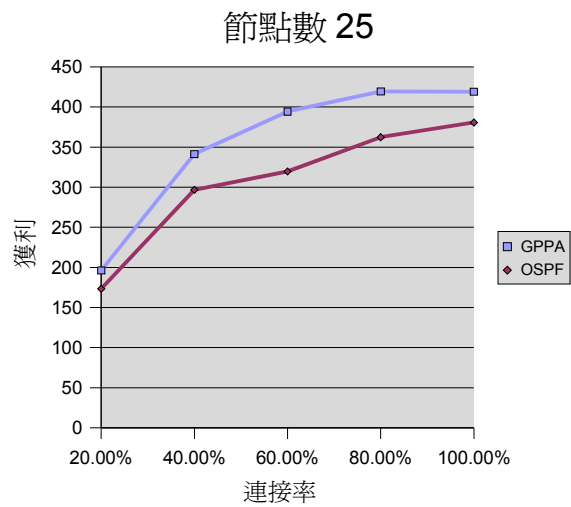
本實驗以網路連結率之大小為實驗變因，探討當網路連結率之大小變化時對於獲利之影響，本實驗設置了一個對照組為傳統的 OSPF 繞徑演算法，實驗過程中會將此兩套演算法透

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。連接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，在較低的連接率時，OSPF 與本演算法的效能相差不多，但在連接率增加時，本方法獲利成長的速度是遠超過 OSPF。



(A)



(D)

圖 5.8：連接率對於獲利指標的影響

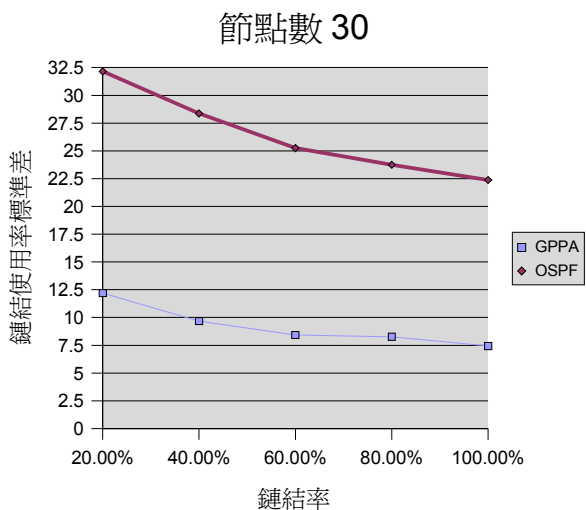
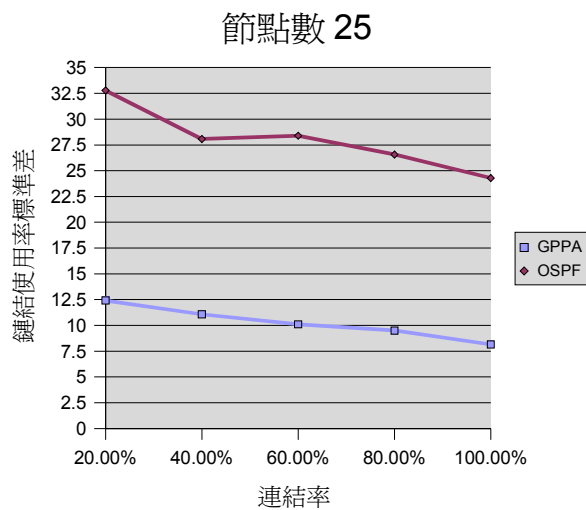
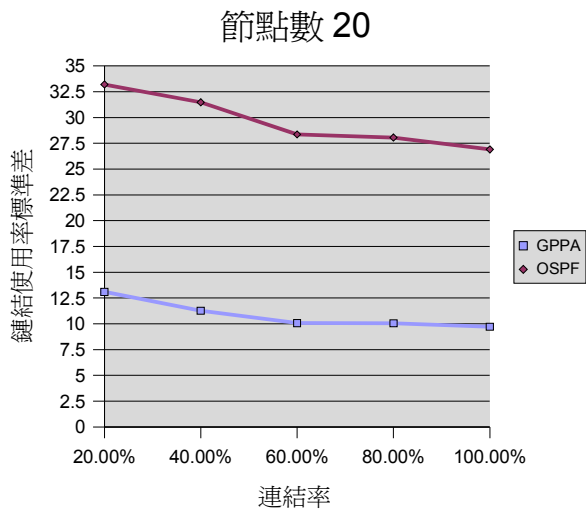
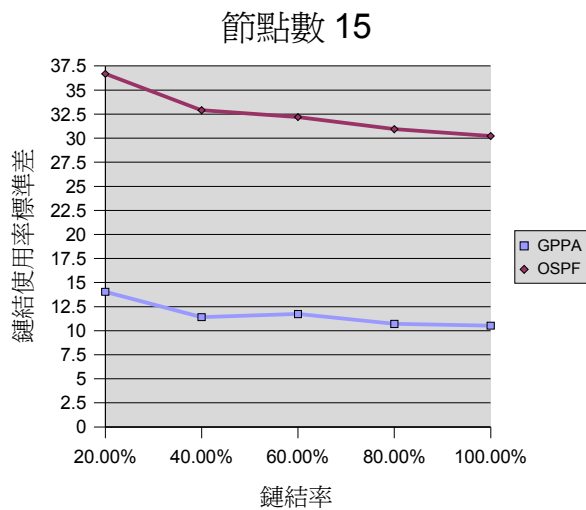
- (A) 網路節點數目為 15, (B) 網路節點數目為 20,
(C) 網路節點數目為 25, (D) 網路節點數目為 30

5.3.3.2 連接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(A)和圖 5.9(B)，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，在圖 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。連接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，在連接率增加時，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。

圖 5.9：鏈結率對於鏈結使用率標準差的影響



(C)

(D)

(A) 網路節點數目為 15, (B) 網路節點數目為 20,

(C) 網路節點數目為 25, (D) 網路節點數目為 30

5.3.3.3 連接率對於獲利密度的影響

在圖 5.10 中其橫軸為連接率的變化，直軸為獲利密度的變化，從圖中可以觀察到在獲利密度上，本方法在連接率增加後，其比值增加的幅度是遠超的 OSPF，也代表本套路徑規劃的方法在連結率增加後，有較佳的網路利用率。

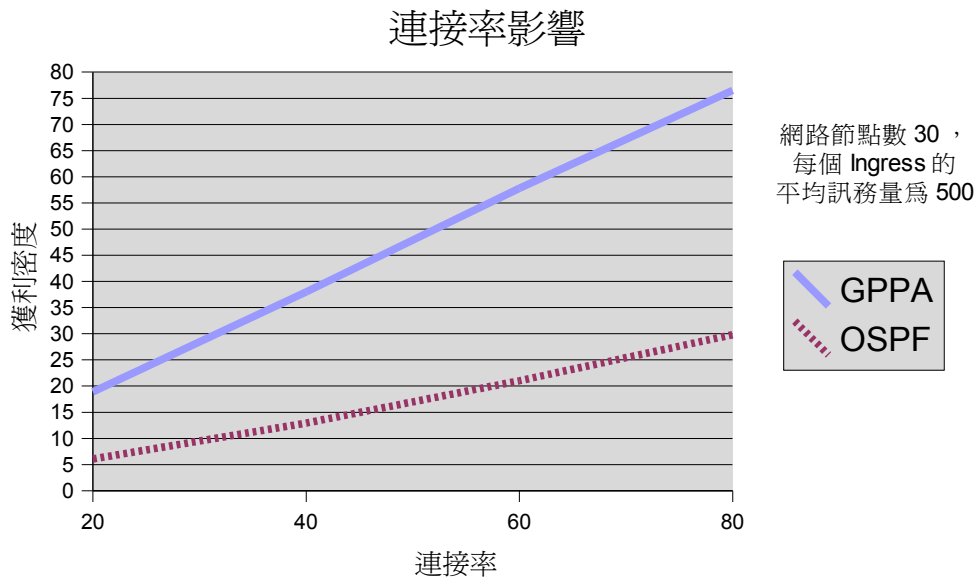


圖 5.10：連接率對於獲利密度的影響

5.3.4 實驗四：預測誤差之影響 (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 GPPA 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。

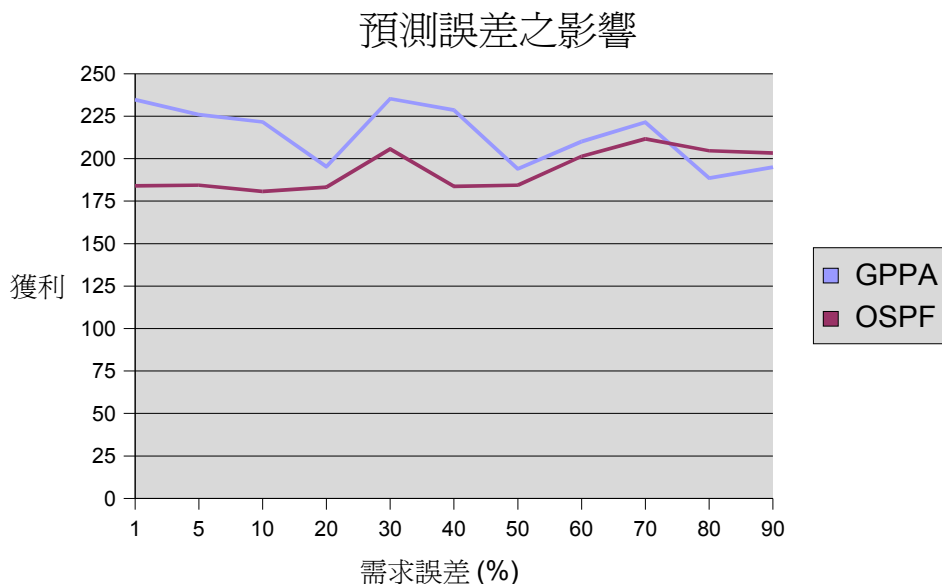


圖 5.11：預測誤差對於獲利指標的影響

5.3.4.3 預測誤差對於獲利密度的影響

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密度之影響，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，直軸為獲利密度大小。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，其獲利密度自然較高。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，因此對於獲利密度而言並未早成太大的變化。

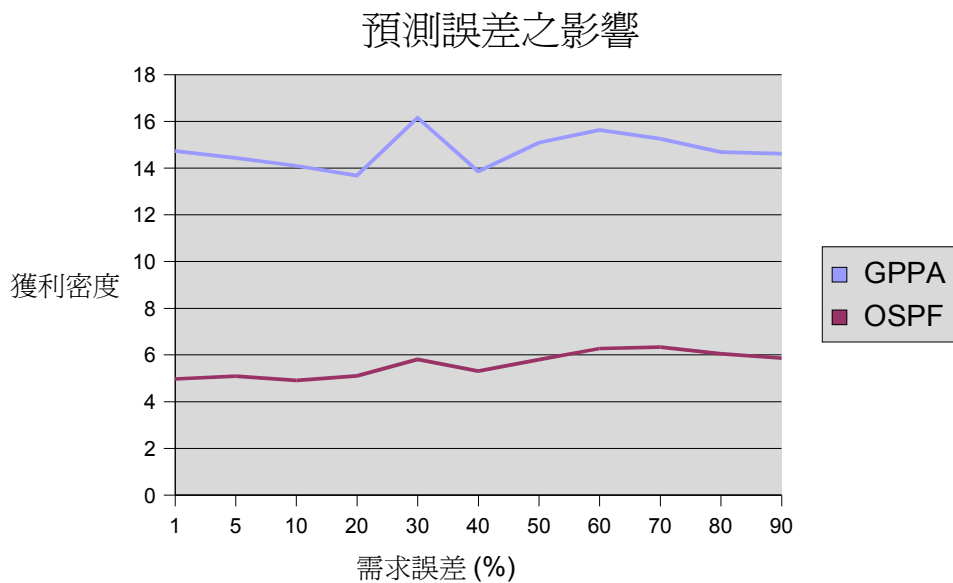


圖 5.12：預測誤差對於獲利密度的影響

5.4 結論

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密度作為效能評量的指標，並觀察節點多寡，連接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連接率對於系統獲利的影響，結果顯示，當節點數目及連接率增加時，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。而在獲利密度上，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，我們變化拓樸中的連接率的大小，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，因此連接率的大小變化，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 (Conclusion)

在本研究中提出了一個具有 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，路經規劃是 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 Profit 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。

致謝詞 (Acknowledgments)

本研究係國科會計劃「All-IP 核心網路品質管理研究」研究成果，編號 NSC 91-2219-E-004-001，特此感謝。另外感謝政治大學應用數學系陸行老師的協助，與資訊科學系行動計算實驗室所有老師與同學的大力幫助。

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] [Mahbubul 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] Uyles Black, "QoS in Wide Area Networks", *Prentice Hall PTR*, 2000.
- [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] 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.
- [16] 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.
- [17] Eleni Mykoniati et al., "Admission Control for Providing QoS in DiffServ IP Networks: The TEQUILA Approach," *IEEE Commun. Mag.* Jan 2003, pp. 38-44.
- [18] 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.
- [19] 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.
- [20] Turgay Korkmaz, and Marwan Krunz, "Multi-Constrained Optimal Path Selection," *IEEE INFOCOM 2001*, pp. 834-843.
- [21] 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.
- [22] Nicolas Christin and Jörg Liebeherr, "A QoS Architecture for Quantitative Service Differentiation", *IEEE Communications Magazine*, June 2003.
- [23] A. Demers, S. Keshav and S. Shenker, "Design and Analysis of a Fair Queueing Algorithm", *Proc. SIGCOMM'89*, ACM, September 1989, pp. 1-12.
- [24] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell and J. McManus, "Requirements for Traffic Engineering Over MPLS", *RFC 2702*, September 1999.
- [25] 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.
- [26] E. Rosen and Y. Rekhter, "BGP/MPLS VPN", *RFC 2547*, March 1999.
- [27] A. S. Tanenbaum, "Computer Networks, Third Edition", Prentice Hall, March 1996, pp. 345-366.
- [28] Dijkstra, E.W., "A Note on Two Problems in Connection with Graphs", *Numerische Math*, vol. 1, March 1959, pp. 269-271.
- [29] C. Hedrick, "Routing Information Protocol", *RFC 1058*, June 1988.
- [30] J. Moy, "OSPF version 2", *RFC 1583*, March 1994.
- [31] Christophe Beaujean, "Delay-Based Routing Issues in IP Networks", *contact GRADIENT CR/98/148*, May 2000.
- [32] Douglas S.Reeves and Hussein F. Salama, "A Distributed Algorithm for Delay-Constrained Unicast Routing", *IEEE Transaction on Network*, April 2000.

- [33] R. Braden, L. Zhang, S. Berson, S. Herzog and S. Jamin, "Resource Reservation Protocol (RSVP) – Version 1 Functional Specification", *RFC 2205*, September 1997.
- [34] K. Chan, R. Sahita, S. Hahn and K. McCloghrie, "Differentiated Services Quality of Service Policy Information Base ", *RFC 3317*, March 2003.
- [35] Bill Goodman, "Internet Telephony and Modem Delay", *IEEE Network*, May 1999, pp. 8-16.
- [36] J. Garcia-Luna-Aceves and J. Behrens, "Distributed scalable routing based on vectors of link states", *IEEE J. Select on Communication*, October 1995.
- [37] Jon Postel, "Internet Protocol", *RFC 791*, September 1981.
- [38] Mark A. Sportack, "IP Routing Fundamentals", *Cisco ISBN: I-57870-071-x*, May 1999.
- [39] 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.
- [40] 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.
- [41] 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.
- [42] Z. Wang and J. Crowcroft, "Quality of Service Routing for Supporting Multimedia Applications", *IEEE Select on Communication*, September 1996.