

第一章

導論

ALL-IP Network (AIPN) [9,11]是一個開放的服務平台，相較於以前各自獨立的傳輸平台，這種網路的優點是可以容納多種不同的服務[14,20]，例如語音、多媒體、資料等各類服務。服務傳輸方式是基於封包交換網路 (Packet Switched Network)，以封包方式傳輸各種類型的訊務，此種網路具有封包交換網路的特性，在承載高時效性的服務 (Time Sensitive Service) 時其服務品質可能不符合要求，例如過長的延遲 (Delay)、劇烈的抖動 (Jitter) 與遺失過多封包遺失 (Packet Loss)。為了要保證從網路一端到另一端的服務品質一端對端服務品質 (End-to-End QoS)，AIPN 中各個獨立的管理架構內可能會採取保留資源給予特定服務之法以達成目標，在這種機制下，服務可能會發生失約或未依約充分使用保留下來的資源因而導致資源之浪費。前者例如預測誤差，後者例如語音訊務 (Voice Traffic)，就是一個很顯著的例子，使用者雙方在講話時不就是講話者說話給另一個人聽，就是講話者會斷斷續續說話，保留的資源會沒有充分被使用，導致資源使用率降低。根據 Brady 的研究，一般使用者平均只有 40% 的時間是在說話[3]。這時候網路營運者必須找出提升資源使用率與淨收益的方法，本研究提出超額允入法以提升資源使用的效能。

1.1 All-IP Networks

現有網路衝擊最大的當屬 Network Convergence，它將目前分離的 circuit-switched

與 packet-switched 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。AIPN 是單一的 IP 網路提供封包交換網路與迴路交換網路的服務，而 AIPN 的優點與缺點如下所述：

AIPN 的優點：

AIPN 提供了一個開放的服務平台，此平台可以容納與支援各種不同的服務，服務是以 Packet Switching 的技術承載服務，而 AIPN 也可以降低建置成本與營運管理成本。最後，又因為 AIPN 是一個開放的服務平台，所以可以支援跨網路的應用。

AIPN 的缺點：

AIPN 承襲原有封包交換網路的特性，有不指定傳送路徑、不可靠或不循序傳送等因素，這些因素將會影響到封包的延遲、抖動與封包遺失，所以不容易提供品質保證予高時效性的服務。

簡化的 AIPN 實際環境：

我們以簡化的 AIPN 實際環境做為代表，介紹 AIPN(圖 1.1)。小雲朵部分是接取網路 (Access Network)，接取網路可能是固定網路、GPRS、3G Radio Network、Local loops 及無線區域網路 (Wireless LAN) 等，接取網路裡面則是以 Access Gateway 與核心網路相連。

骨幹網路包含多個互連的核心網路，大雲朵代表一個核心網路 (Core Network)，核心網路包含 Edge Router 與 Interior Router，Interior Router 與 Interior Router 互相連結，而 Edge Router 則有 Border Gateway 與 Inter-Domain Gateway，Edge Router 與 Interior Router 連結共同形成核心網路，核心網路藉由 Edge Router 與接取網路連結或與另一個核心網路互相連結。而各個核心網路之間使用 interconnection links 相互連接成骨幹網路，稱為骨幹網路 (Backbone Network)。

當使用者要獲取一項服務時需要經由接取網路以連接到核心網路，然後藉由核心網

路與另一端的接取網路連接起來，其所連接的網路及所經過的路如圖 1.1 中之紅線部分，而傳輸的方式則是基於封包交換網路，以封包方式傳送到另一端，封包所經過的路並非是沿同一個路徑，也可以沿不同的路徑來走。服務從使用者端到另一端所經過的路徑在本研究環境 BBQ 內的意義是 End-to-End Path，骨幹網路的路徑稱為 Long Path，在一個核心網路內所經過的路徑是 Short-Path。

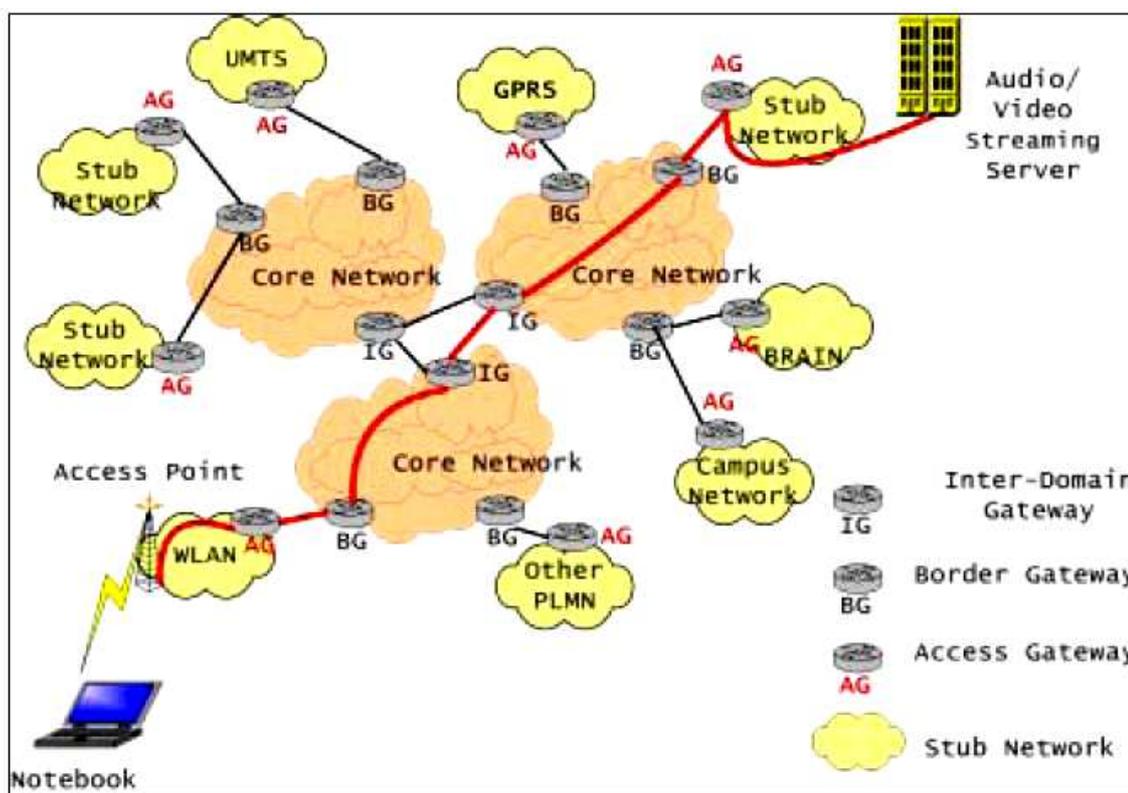


圖 1.1：簡化的 AIPN 實際環境

AIPN 是以封包交換網路承載各類服務，所以不容易保證端對端的服務品質 (Guaranteed End-to-End QoS)。為了保證端對端的服務品質，AIPN 內各個核心網路可能是獨立運作，各個核心網路可能使用不同的 QoS 管理架構，因此要很難管理端對端服務品質。這時服務可能會經過多個營運者的核心網路，所以需要多核心網路的 QoS 管理架構。

1.2 提供單一核心網路的 QoS 管理架構

在單一核心網路內的 QoS 管理架構中，知名的有 IntServ 及 DiffServ 這兩種網路架構，IntServ 使用保留資源之法以保證服務品質，而 DiffServ 則沒有保留資源，而是根據訊務的優先權提供服務品質。

1.2.1 IntServ (Integrated Service) 架構

網路架構 IntServ[6]是以控制端對端延遲來整合即時性服務，IntServ 將即時性的服務分成事先保證 (Predictive) 與即時保證 (Guaranteed)。為了達成整合即時性服務，在這種網路架構上各個路由器中分成控制與資料層。在控制層要有明確的控制信號 RSVP (Resource Reservation Protocol) 以溝通資源保留的情況，以此建立虛擬通道 (Virtual Circuit)，在資料層依照虛擬通道依次序送出封包。最後路由器可以控制線路分享 (Controlled Link-sharing)，以達成控制端對端延遲。但是為了控制端對端延遲，每個路由器要都要參與控制所以 overhead 太大，因此允入訊務數目有限制。

1.2.2 DiffServ (Differentiated Service) 架構

DiffServ[4,8,19]是將相似 QoS 需求的訊務合併在一起形成一類，對同類型的訊務提供一致與相對的服務保證。各類型服務依照各類優先權傳送封包，每一類型服務在 DiffServ 會有相對應的 PHB (Per-Hop Behavior)，並依照 PHB 來傳送。DiffServ 雖然沒有辦法達到像 IntServ 一樣的絕對性服務品質保證，但是可以避免 IntServ 所有的擴充性問題。DiffServ 中服務品質的需求基本定義了三種行為 (Behaviors) 為標準：

1. Expedited Forwarding (EF)：EF 是最高等級的服務品質，EF 必須提供足夠的資源，以降低可能的延遲。
2. Assured Forwarding (AF)：此服務提供較 EF 不穩定與寬鬆的 QoS 保證，AF 依

據需求的 drop precedence 決定哪些封包該被捨棄。

3. Best Effort (BE)：BE 不刻意保留資源，因此幾乎不提供 QoS 保證，採用在沒有時效性需求之服務。

1.3 提供多核心網路的 QoS 管理架構

由於 AIPN 可能有好幾個核心網路，服務會經過好幾個不同營運者的核心網路，因此有提供多核心網路的 QoS 管理架構，提供多核心網路的 QoS 的管理架構有 BBQ、TEQUILA 與 AQUILA 等，本研究以 BBQ 服務品質管理架構當作研究用架構。

1.3.1 TEQUILA (The Traffic Engineering for Quality of Service in the Internet at Large Scale) 管理架構

這個架構是歐洲電信業者 (ALCATEL、Algo Systems、France Telecom、Global Crossing) 共同贊助的一個計畫[16]，目標是研究網路服務的定義並提出一些 traffic engineering 的工具以達成兼具質與量的服務品質保證。由 Service Level Specifications Management (SLSs Management)、Traffic Engineering 與 Policy Management 所構成，底層則是以 Data plane 所構成。

1.3.2 AQUILA (Adaptive Resource Control for QoS Using an IP based Layered Architecture) 管理架構

這個架構也是歐洲電信業者共同合作的計畫[2] (Siemens、Arvato Systems、Polish Telecom、Telekom Austria、Q-Systems、CoRiTel and Elisa Communications)，目標是在具有 IP 的網際網路中，也就是基於具有 DiffServ Network Concept 的網路，這個架構要增強其所在網路的能力，使其同時具有成本效益與具有擴展性的 QoS 的特性，此架構將達成端對端優值的服務保證。此架構從使用端開始保留資源的方式並提供 toolkits 工具給使用者，並以動態方式提供 QoS 給使用者運用。

1.3.3 BBQ (Budget-Based QoS Network Management) 管理架構

BBQ 管理架構[12]以簡化管理複雜度、增進資源使用效率為原則，其主要目的是協助營運者靈活管理控制網路，以達到營運者的營運目標。BBQ 架構內的核心網路中有 ACA(Admission Control Agent)與 BB(Bandwidth Broker)，以此允入訊務與管理分配資源。

本研究假設系統環境是 BBQ 管理架構，它是以預算為基礎的 QoS 管理架構，預先分配各個網路元件需要承載的 QoS 責任，而每一個核心網路為了要承載各自的 QoS 責任，先由 BB 將鏈結資源分配到所有 ACA，而每一個 ACA 使用分配到的鏈結資源組成 short-path 資源，每個 ACA 所擁有的資源是 edge router 到 edge router 的 short-path，以承載這一段的 QoS 責任。第二章將會完整敘述 BBQ 管理架構。

1.4 資源保留機制的缺點

核心網路中的 ACA 是一個資源保留機制，訊務可能因失約或未依約充分運用所保留之資源而導致資源閒置。

前者例如：BB 根據預測訊務的需求來配置資源給各個 ACA，但預測會有誤差，導致有些地方資源過剩，有些地方資源短缺，造成某些 ACA 有剩餘資源而某些 ACA 必須拒絕服務。後者如語音訊務 (Voice Traffic) 是一個很顯著的例子，依據統計[3]，在通話過程當中，通話的雙方通常只有 40% 的時間有在講話。剩下的 60% 時間，通話者沒有在講話，例如停下來聽對方講話或是句子和句子之間以及字與字之間的停頓，都是不發出聲音的。不發出聲音時段，所保留的資源並未依照約定充分運用。保留的資源閒置下來沒有被其他訊務所使用，資源使用的效能就會降低。

1.5 解決資源閒置之方法

資源保留機制發生資源閒置情況，這種情況分成解決訊務預測誤差而失約的問題與解決訊務未依約使用資源的問題，這節將講述各個問題在 AIPN 內 BBQ 管理架構所提出的解決方法。

1.5.1 解決預測誤差而失約的問題

在 BBQ 管理架構內的規劃階段，BB 分配 link resource 給 ACA，BB 根據資源需求量事先規劃與分配資源，但預測的資源需求量可能有誤差故需要彌補，彌補的方法有下述三種：超額分配法、資源重新分配法、中央保留資源法。

- 中央保留資源法 (Central Pool)：

在 BB 先分配資源之前，BB 先保留一些資源以備不時之需，而核心網路的 BB 負責網路資源的統籌規劃，各 ACA 對剩餘的資源有設定一個門檻 (Threshold)，當剩餘資源低於此門檻時，ACA 將會向 BB 要求更多資源，BB 會將中央所保留之資源配置給 ACA。也就是小部分的資源被 BB 保留下來，直到 ACA 缺乏資源時，BB 再即時分配給 ACA 使用。

- 資源重新分配法 (Resource Reallocation)：

BB 將閒置的資源即時重新分配給需要使用資源的 ACA，或 ACA 主動繳回過剩之資源，而 BB 視使用情況調整配置資源。

- 超額分配法 (Overbooking)：

因為在規劃階段時 BB，由於預測資源需求量有誤差，這種情況會造成資源錯置的現象，所以網路內有些 ACA 資源會不足，而有些 ACA 資源會過剩，BB 可以事先超額分配較多一點資源給 ACA 以彌補誤差並提高資源使用率。

1.5.2 解決訊務未依約使用資源的問題

在 BBQ 管理架構內的允入階段，ACA 配置資源給訊務，但是訊務若是有未依約充分運用所保留之資源的性質，這將會造成資源閒置。

- 超額允入法：

在保留資源以保證服務品質的系統中，各種服務並不一定會經常滿載，造成資源之閒置。例如語音訊務，講話者有時講話或有時沒有講話，同一時間通常只有一方在講話，導致資源閒置，我們可以使用超額允入法讓超額的訊務進入網路之中以提升整體資源使用的效能。但若是過度超額允入訊務時，會使網路超載，造成溢流導致嚴重的封包遺失，因此有必要預先計算最佳允入量，使超額允入之同時避免嚴重的超載，而這就是本研究的重點。

ACA 允入很多種訊務，不同類型的訊務有不同的溢流機率，組合不同類型訊務的溢流機率是很困難，因此研究困難在於如何求出溢流機率，並且只考慮到語音訊務。假如服務需求的機率分佈為已知，我們可以預估溢流機率，以此算出合適的允入量。

最後，在資源保留架構內，允入控制機制超額允入比所擁有的資源量還要多的訊務量，以提升整體資源使用的效能。

1.6 研究目的與方法

本研究的目的是在研究超額允入法，以此方法提升整體資源使用的效能。但是 ACA 使用了超額允入法可能會過多允入訊務，所以系統的資源使用的效能將會下降。而本研究的重點在允入階段內計算最佳的超額允入量，以提升整體資源使用的效能。

由於目前 AIPN 比較盛行的訊務其中一種是語音訊務，預測未來將會是一種重要的應用服務，並且組合各類訊務的困難，所以目前只針對語音訊務作研究。然後假設語音訊務的機率分佈是 exponential on/off traffic，以此建立起分析模型，再求得最高淨收益時

的允入量。最後使用模擬以驗證模型與評估效能。

1.7 論文組織結構

第二章是相關研究，第三章是分析模型與分析模型內各個模型，第四章則是模型驗證與效能評估，最後則是研究總結。

第二章

相關研究

這一章將會簡介 AIPN 內的所用的大型 QoS 管理架構，與超額分配與允入概念相關的文獻。

2.1 提供多核心網路的 QoS 管理架構

在 AIPN 內的大型 QoS 管理架構除了 BBQ 管理架構，有名的管理架構各別有 TEQUILA 與 AQUILA 管理架構，這兩個管理架構提供 AIPN 端對端服務品質保證，現在介紹這兩個架構，之後介紹本研究的研究環境 BBQ 管理架構。

2.1.1 TEQUILA

TEQUILA (The Traffic Engineering for Quality of Service in the Internet at Large Scale)[17] 是歐洲的電信業者 (ALCATEL、Algo Systems、France Telecom、Global Crossing) 所共同贊助的一個計畫專案。這一個專案實行、驗證 Service 的定義與規劃 TE (Traffic Engineering) 的工具。以 Service 定義與 TE 工具達成可簡單動態控制的訊務管理，並且具有可擴展的性質，在具 DiffServ Concept 的網際網路中 (DiffServ、IP networks) 得到量化的端對端服務品質保證。

TEQUILA 的架構分成三個部分，如圖 2.1 所示，SLSM (Service Level Specifications

Management) 主要負責處理客戶的服務品質要求，SLSM 內的 SLSs 則有靜態、動態、inter-domain 與 intra-domain 的規格書。Traffic Engineering 提供了 schema 來進行 QoS 的協調。PM (Policy Management) 則是根據已有的 policy 扮演一個 Administrator 的角色，而底層的 Data Plane 則是負責實際資料的傳送。

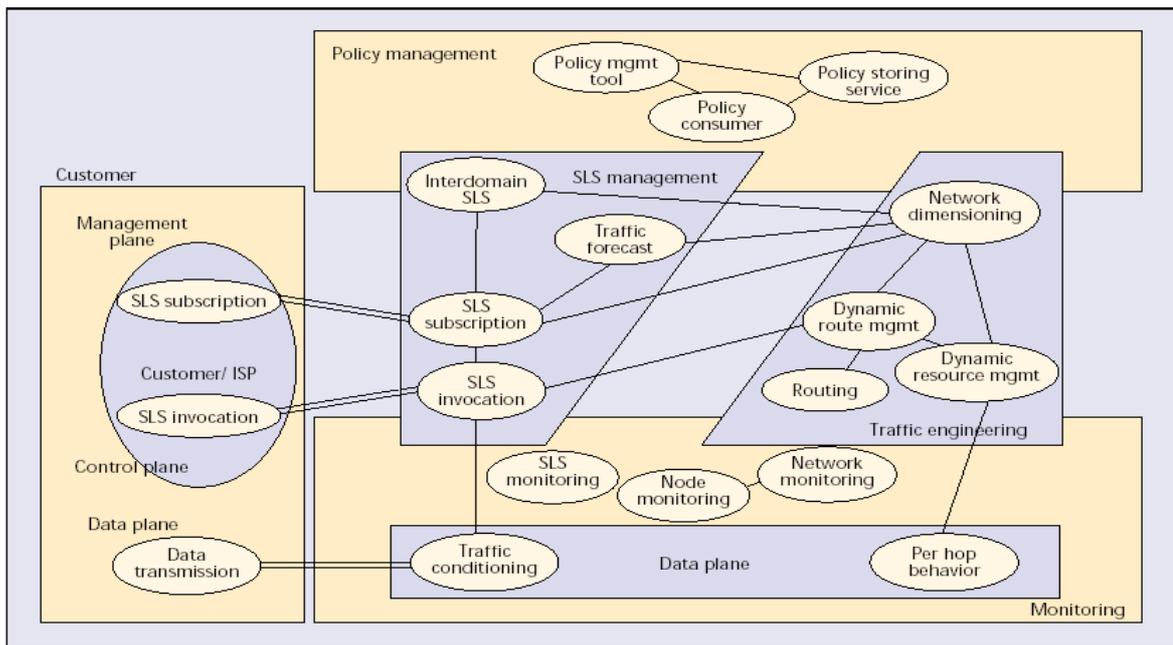


圖 2.1：TEQUILA 架構

使用者經由 SLSs 在 SLSM 之內與 AS 互相溝通。當資料將要傳送到網路時，TE 根據已經算好的部分路徑，依據網路狀態選擇合適的路徑給予授權後的使用者。最後在不同的網路狀態下，PM 依據狀態設定的 policy 管理資源。

2.1.2 AQUILA

AQUILA (Adaptive Resource Control for QoS Using an IP based Layered Architecture) [2] 是 IST (Information Society Technology) 在歐洲聯合各電信業者 (Siemens、Arvato Systems、Polish Telecom、Telekom Austria、Q-Systems、CoRiTel and Elisa Communications)

的專案之一，這專案在網際網路中定義、實行 QoS 的進階架構。這個專案內的架構依照成本效益與擴增性原則使用 IntServ、DiffServ 與 Label Switch 等科技，以此連結使用者與網路以達到動態的端對端服務品質。

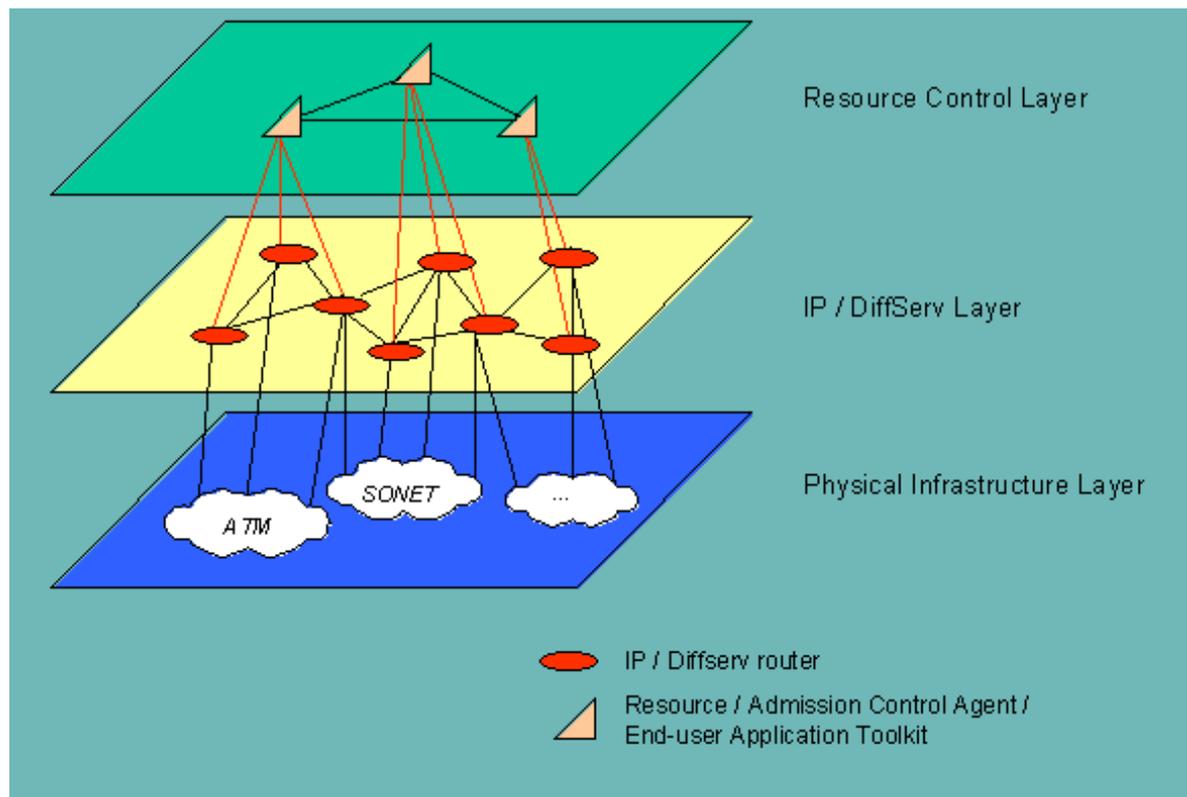


圖 2.2：AQUILA 架構

AQUILA 管理架構分成三層：RCL (Resource Control Layer)、IP/DiffServ Layer、Physical Infrastructure Layer，可由圖 2.2 看出這三層。這三層中比較重要的是 RCL，RCL 主要是用分散方式管理網路資源、終端管理與允入管理並提供 QoS，RCL 提供 QoS 的方式要先對服務進行測量網路狀態以提供或增進允入控制的能力。保證服務品質過程則是由使用者端開始傳送保留資源的訊號給 RCL，RCL 依據訊號保留資源以保證 QoS。

2.3.3 BBQ (Budget-Based QoS Network Management) 管理架構

BBQ 管理架構[12]以簡化管理複雜度、增進資源使用效率為原則，主要的目標是幫助營運者靈活管理控制網路，以達到營運者的營運目標。

為了提高網路效能與加速回應速度，BBQ 架構使用預算分配和事先規劃資源分配之法，避免採用即時性 on-demand 的方法，來達成端對端服務品質保證。

在 BBQ 管理架構內的各個網路元件，例如核心網路與接取網路，BBQ 根據各網路元件的能力事先規劃 QoS 責任，而各網路元件要盡力達到所指定的 QoS 責任，若是訊務所經過的各個網路元件都達到指定的 QoS 責任，就可達到端對端服務品質保證。

● 端對端承載服務品質

各個子網路可能以預留資源之方式，以保證達到這一段的 QoS 責任。在核心網路內 short-path 是基本的資源，各個 Edge Router 內的允入控制者(ACA)預留一批 short-path 資源並分配給允入的 requests 以達到這一段 QoS 的責任。Global ACA 需要挑選出 long path，long path 是由 short-paths 接連起來，ACA 需要找出一條合適的 long path 與預留 long path 來達成這一段的 QoS 責任。當各子網路可以成功達成各自負責的 QoS 責任時，將可以達成端對端服務品質。可以從圖 1.4 可以看得出以上所敘述。

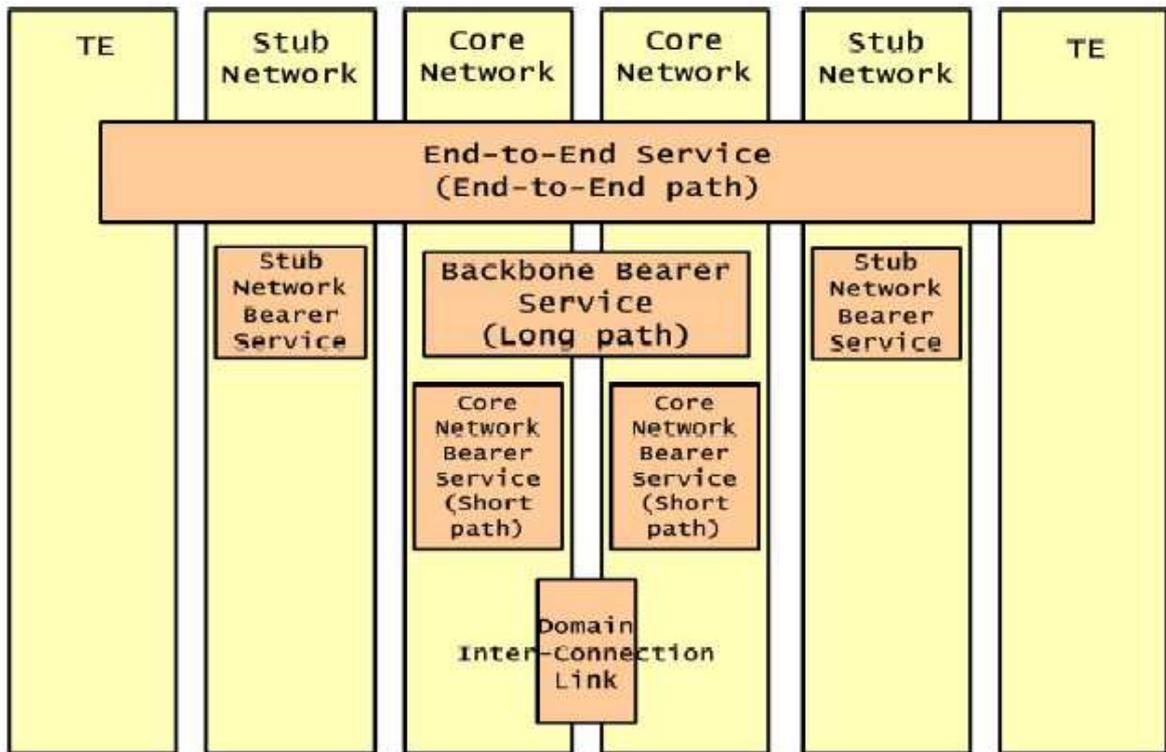


圖 2.3：端對端承載服務品質

- 核心網路管理架構與資源分配

在 BBQ 架構下，核心網路管理的架構如圖 1.5，假設核心網路為一個 DiffServ 網路，由數個 Edge Router、Core Router 與一個核心網路協調元件（Core Network Coordinator, CNC）組成，QoS 的管理分散在核心網路協調元件與各個 Edge Router 上，而 Core Router 僅負責傳遞資料，Edge Router 又分為 Ingress Router 與 Egress Router，各別讓訊務進入與流出核心網路。在 CNC 中有 BB(Bandwidth Broker)，在 Edge Router 上有 ACA。每個核心網路有一個 BB，BB 負責管理核心網路內部所有鏈結的頻寬，在規劃階段 BB 把鏈結的頻寬分配給各個 ACA。Edge router 上的 ACA 負責允入訊務與管理資源分配。在規劃階段各個 ACA 把所分配到的鏈結資源組成 short-path 資源，在允入階段 ACA 挑選出適當的 short-path，並保留給要進來的 requests 使用。

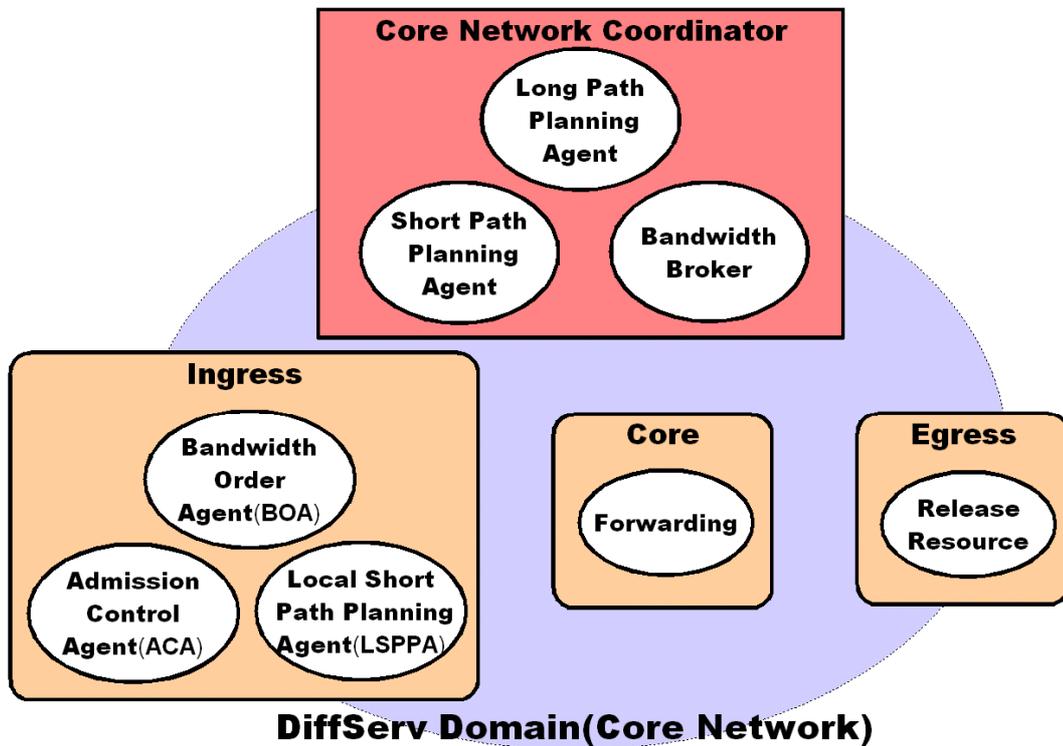


圖 2.4：核心網路的管理架構

- **BBQ 的前端允入控制機制**

當訊務要進入 BBQ 所管理之網路時，如圖 1.6 所示，首先訊務先接觸到的是在接取網路內的元件 global ACA（global Admission Controller Agent），global ACA 針對訊務的端對端路徑，對所經過的 Access Networks（圖中 Entrance Stub Network 與 Exit Stub Network）進行檢查與預留資源的動作。對所經過的核心網路將會即時性檢查與預留 short-paths 資源，ACA 即時預留這些資源。當各段都可以成功預留與檢查資源，global ACA 即可以允入訊務。

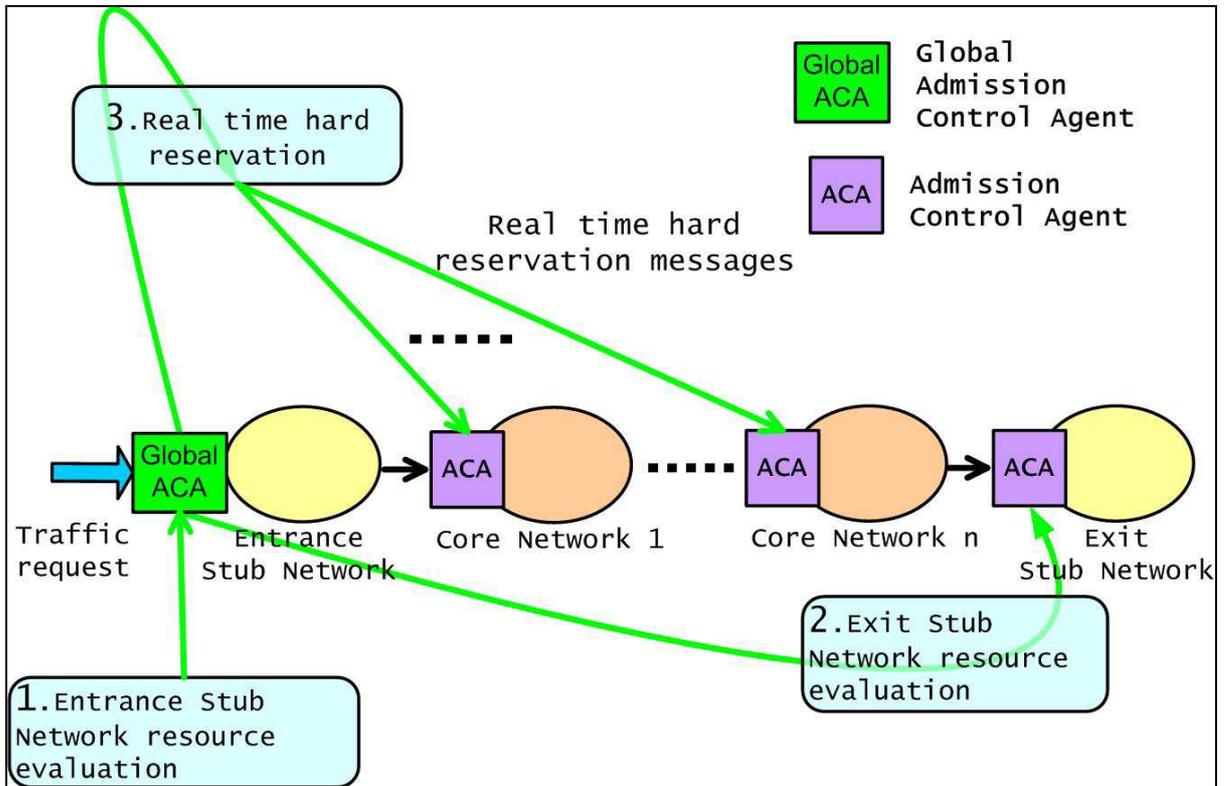


圖 2.5：在 Global ACA 中端對端服務品質的建立程序

2.2 超額分配與允入法的相關研究

在此先介紹超額分配法的相關研究，然後介紹超額允入法的相關研究。

● 超額分配法

超額分配法[13]可用以解決由於預測誤差所導致的資源錯置的問題。當系統以歷史資料預測需求，預先配置資源時，會因預測誤差導致資源錯置，造成部分可用的資源被保留在不需要的地方，而有些地方資源不足而必須拒絕進入的訊務。因此需要一個有效的方法以防範資源浪費的情況發生。

假設網路之中有一些 Edge Routers 因預測誤差而可能有一些未使用的頻寬，此時可以藉用超額分配以運用閒置的資源，如此 BB 可以使用超額分配法以分配比所擁有的資

源量還多的資源給予訊務，其設計重點則在於使一個鏈結的整體實際的頻寬消耗量不會超過此鏈結的容量。

BB 超額分配資源給 Ingress Router 以增加各個 Ingress Router 可進入網域的訊務量與降低資源浪費的機率，這便是超額分配法主要的精神。每個 Ingress Router 對 BB 提出的資源預購需求可能過多，也可能太少，超額分配法隱含資源重分配之精神，發揮截長補短之功效，讓餘裕的資源自動重分配給資源短絀的 Ingress Router。

此文獻提出的超額分配法首先預估超額上限，再依比例分配給每個提出需求的 Ingress Router，提高進入的資料流總量。這個方法的好處在於系統不必隨著各個 Ingress Router 實際資源需求的變化來調整資源配置，降低了資源配置的負荷；然而超額分配法仍須承擔超過系統負荷的風險，各個 Ingress Router 進入網路的資料如果超過負荷時，網路品質可能會快速惡化。超額分配雖有相當的好處，但必須經過周密設計以減少超過負荷的機會。

此份文獻的困難在於如何超額分配才可以達到較佳的使用頻寬需求與訊務溢流機率最小化，最後文中以最佳化模型求出超額分配程度，從上面討論可看出超額分配是為了要解決預測資源需求量誤差而失約的問題。

在此找出與超額分配相關的文獻有[15,10]，第一份與第二份文獻是在 ATM 的環境，這兩份文獻是超額分配法。第三份文獻[9]是在 server cluster 環境中的超額分配法。

文獻一是在 ATM switch 與 backbones 的環境，有 ABR (Available Bit Rate) 訊務與 VBR (Variable Bit Rate) 背景訊務。為了要讓資源使用率更高，switch 將會 aggregate 訊務，當 aggregate ABR 與 VBR 背景訊務時，若是 VBR 背景訊務 burst 的情況嚴重或 RTT (Round Trip Time) 相當高時，假如 VBR 的 priority 很高，系統會難以使用 ABR 追蹤資源使用率，所以可以在訊務中超額分配資源。最後以控制 ABR 的方式，以改進的 rate-based flow control 來解決。

文獻二是在 ATM switch 內的 ABR service，ABR 對服務有進行保證頻寬的功能。文

獻中主要討論的部分是在 Up-streaming scheduler 與 Down-streaming switch 之間。在 Up-streaming scheduler 部分由於 ABR 已經先行給定資源，但是資源可能會有閒置情況會發生，所以要重新分配資源給其他的 ABR service 使用。最後，可以在 Up-streaming 部分執行超額分配，本文獻則是在 Up-streaming scheduler 內使用動態排程來超額分配出資源。

文獻三是在提供服務的網路內每個 host 是 server cluster，server cluster 是一種 host，每個 host 都是由獨立架構的小 server 所組成，例如 blade server 是一種例子，每個 host 所能提供的服務數量都會根據小 server 的所能提供的服務而不定。Host 會有部分的小 server 會沒有使用資源的時候，此時可以讓別的連結連接到此小 server 服務。當 host 會有未使用資源的時間，可以讓出未使用的資源給要進來的 service 使用，但是需要在 server 端設置監視應用程式的 profile，用此 profile 可以得知何時會使用資源，何時不會使用資源，所以可以控制每個服務資源使用，因為可以控制資源使用，所以可以超額分配 host 的資源並且增加總體的利益，文中使用 CPU on/off 來分析，並使用 third-party 應用程式來判斷是否可以超額分配資源，而資源主要針對 CPU 以及網路資源這兩種來進行討論，最後可以由實驗看出增加利潤是原來的三倍到五倍。

● 超額允入法

超額允入法主要是在解決訊務未能依約使用資源的問題，也就是當 ACA 給定或保留資源給予訊務，但是訊務可能沒有充分使用所保留的資源造成資源閒置的現象，這時 ACA 可以超額允入比所擁有的資源量還多的訊務量以提升整體資源使用率。

超額允入法相關的文獻[5]，這份文獻是處於 DiffServ 的環境，在這環境內 ACA 使用分散式的允入控制，其中 Admission Controller 可以彼此分享、借用與偷用資源。當 ACA 偷用資源後可能會大量的允入訊務，訊務可能會有服務品質低落的風險，這時需要有一個方法去預防這種情況發生。文獻中根據網路中的保證傳輸率提出一個最高界

限。

2.3 文獻討論

由上述可以知道一些著名的大型 QoS 網路管理架構，另外超額分配與允入可以用在保留資源為基礎的系統以提高資源使用率。在 BBQ 內，超額分配與允入的差別在於：超額允入適用於資源配置中的允入階段，超額分配則是適用於資源配置中的規劃階段。為了要適用於允入即時性訊務，本研究在 ACA 上使用超額允入法。

第三章

分析模型

於 BBQ 管理架構內的核心網路中，ACA 預留資源給予訊務以達到所負責的 QoS 責任，由於某些訊務可能會未依約充分使用所預留的資源導致核心網路內的 ACA 握有資源卻無法允入訊務。對於這一種情況，本研究提出超額允入法以提升資源使用效率，但 ACA 卻可能過度允入訊務招致嚴重的溢流進而影響到系統的 QoS。

本研究要提出一個分析模型以算出最佳的允入量供 ACA 使用，本章提出簡化的最佳化模型與研究的系統環境。

3.1 假設條件與系統環境

本研究假設是在 BBQ 管理架構之下，進入到 BBQ 管理系統的訊務全部都是語音訊務，此架構如第二章所述，在一個核心網路中，ACA 使用由 BB 分配出的鏈結資源組成的 short-path 資源，以承載此段的 QoS 責任，在本分析模型中單位資源是一條 short-path。

假設一個 ACA 擁有 C 單位 short-path 資源，允入的訊務量為 Z ，我們以收益來代表允入的語音訊務對系統效益之提昇，並假設每個允入的訊務在每單位時間與每單位資源下之收益為 $C1$ ，我們用收益之損失來代表 Overflow 對系統的 QoS 之損害，並假設每個允入訊務在每單位時間與每單位資源下之損失為 $C2$ 。語音訊務的 Active 與 Inactive Time 假設是隨機變數 T 與 T' ，它們的機率分配是指數分佈，而平均值為 $1/\lambda$ 與 $1/\lambda'$ 。

本研究要在以上給定的條件內找出最佳允入量 Z ，使獲得最大淨收益。表 3.1 列出

本文內所會用到的符號：

表 3.1：超額允入模型符號表

| 符號 | 意義 |
|--------------|---|
| Z | 允入量 |
| C | ACA 能提供的資源量 |
| $C1$ | 訊務使用每單位資源每單位時間的收益 |
| $C2$ | 訊務因過多允入訊務，每單位資源每單位時間之損失 |
| T | 訊務 Active Time 的隨機變數 |
| T' | 訊務 Inactive Time 的隨機變數 |
| $P(T)$ | 訊務 Active Time 的機率分佈，是指數機率分布， $\lambda e^{-\lambda T}$ |
| $P(T')$ | 訊務 Inactive Time 的機率分佈，是指數機率分布， $\lambda' e^{-\lambda' T'}$ |
| $1/\lambda$ | 訊務平均 Active Time |
| $1/\lambda'$ | 訊務平均 Inactive Time |

3.2 分析模型之探討

淨收益是允入 Z 個訊務所收得的收益減掉 Z 個訊務造成溢流所產生的損失，分析模型如

Eq. 3-1 所表示：

$$\text{Max NET_PROFIT}(Z) = \text{PROFIT}(Z) - \text{PENALTY}(Z)$$

s.t.

$$Z \geq C \dots\dots\dots(3-1)$$

其中

NET_PROFIT (Z) 是淨收益

PROFIT (Z) 是收益

PENALTY (Z) 是損失

3.2 收益與損失之探討

收益是以允入訊務的數量(Z)與當每單位時間及每單位資源下使用資源的收益(CI)來計算，如 Eq. 3-2 所示：

$$\text{PROFIT}(Z) = CI * Z \dots\dots\dots (3-2)$$

我們以圖 3.1 來解釋損失，當 ACA 所允入的訊務其 active 的訊務量多於它所擁有的資源量 (C) 時，這種情況將會發生有些訊務沒有可用的資源引起溢流現象，溢流將會導致封包遺失進而產生損失。當溢流時，溢流量與它所造成的品質損害之間可能是非線性關係，但是非線性問題比較困難，超出本研究的研究範圍，此外在最佳允入點附近數值其溢流可能比較不嚴重，因此假設溢流與損失呈線性關係。

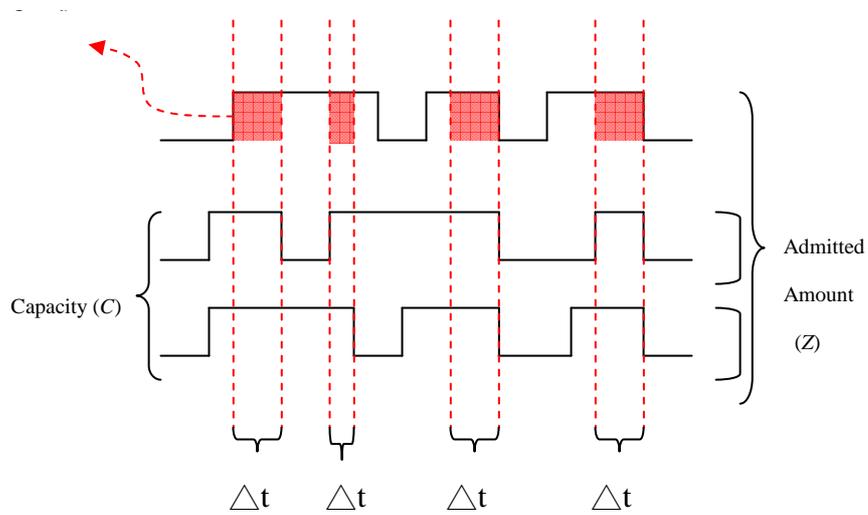


圖 3.1：允入訊務所造成的損失

所以損失是由溢流量 $(y-C)$ 、溢流機率 $(P(Z, y))$ 及在每單位時間與每單位資源下過度允入訊務的損失 $C2$ 總和而成，如表 3.2， $y-C$ 表示溢流數目， $y-C$ 會有對應的溢流機率。如 Eq. 3-3 所示：

$$PENALTY(Z) = C2 * \sum_{y=C+1}^Z \{(y-C) * P(Z, y)\} \dots\dots\dots (3-3)$$

表 3.2：Penalty 相關參數

| | |
|----|-----------------------|
| Z | 允入量 |
| y | Active 訊務量 |
| C2 | 每單位時間與每單位資源之過多允入訊務之損失 |

3.3 由語音訊務的特徵求溢流機率

在此藉由語音訊務之分析模型求出溢流機率，一個語音訊務會有 Active 與 Inactive 兩種狀態，本研究中的語音訊務如圖 3.2 所敘述， T 代表語音訊務中講話者講話的時間，假

設分佈是指數分佈，其平均值是 $1/\lambda$ 。 T' 代表語音訊務中講話者沒有講話的時間，也是假設是指數分佈，其平均值是 $1/\lambda'$ 。為了簡化模型，假設訊務 Active 與 Inactive 的機率在各時間點都相同，並且假設所進入的訊務會一直在網路中，所以溢流機率與 holding time 無關，這個假設所造成之誤差，將在實驗中探討。如此模型只要乘上語音訊務的 holding time 就可以此算出淨收益，接者我們以訊務狀態轉換的機率找出訊務 Active 的機率。

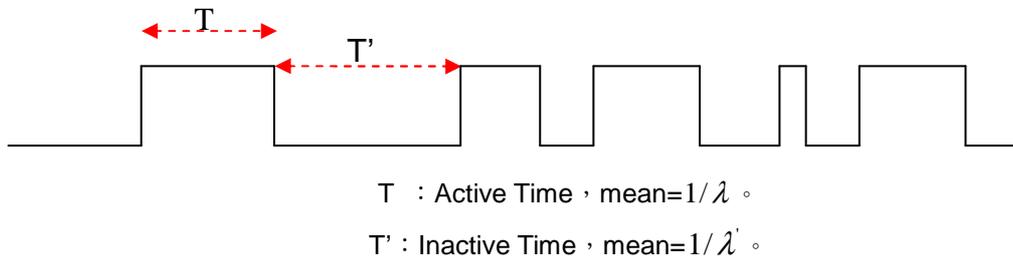


圖 3.2：語音訊務之機率模型

3.4 訊務 Active 機率

現以訊務的狀態轉換機率計算訊務的 Active 機率。訊務的狀態只有 Active 與 Inactive 兩種， Δt 為一個短暫的時間，訊務在 t 時的狀態與在 $t+\Delta t$ 之後的狀態各有兩種，而狀態轉換則有四種可能的組合：持續 Active、持續 Inactive 以及 Active 與 Inactive 互相轉換 [9]，如表 3.3 所示。

表 3.3：t 與 t+Δt 之間可能的狀態轉換

| 情況 | 狀態轉換 |
|----|--------------------|
| A | Active->Active |
| B | Inactive->Inactive |
| C | Active->Inactive |
| D | Inactive->Active |

其次我們將計算 A、B、C 與 D 的機率及訊務的 Active 機率。

- 情況 A：持續 Active(Active -> Active)

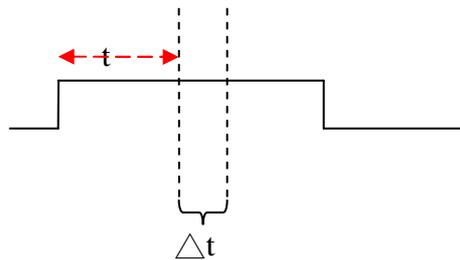


圖 3.3：情況 A(Active->Active)

如圖 3.3，當訊務在 t 時是 active 而且訊務持續 active 到 t + Δt 之後，如此訊務才會持續 active，Eq. 3-4 為其機率。

$$\begin{aligned}
 &P(T > t + \Delta t / T > t) \\
 &= \frac{p(T > t + \Delta t \cap T > t)}{p(T > t)} = \frac{p(T > t + \Delta t)}{p(T > t)} = \frac{e^{-\lambda(t+\Delta t)}}{e^{-\lambda t}} = e^{-\lambda \Delta t} = 1 - \lambda \Delta t + o(\Delta t) \dots\dots\dots (3-4)
 \end{aligned}$$

- 情況 B：持續 Inactive(Inactive -> Inactive)

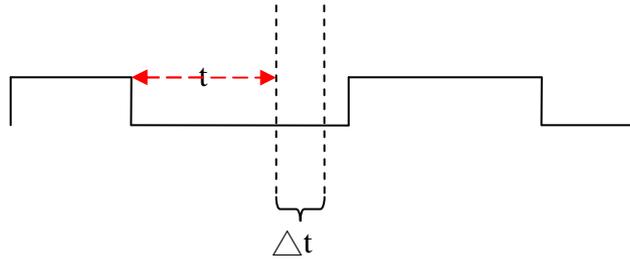


圖 3.4：情況 B(Inactive->Inactive)

如圖 3.4，當訊務在 t 時是 inactive 而且訊務持續 inactive 到 $t + \Delta t$ 之後，如此訊務才會持續 inactive，Eq. 3-5 為其機率。

$$\begin{aligned}
 &P(T' > t + \Delta t / T' > t) \\
 &= \frac{p(T' > t + \Delta t \cap T' > t)}{p(T' > t)} = \frac{p(T' > t + \Delta t)}{p(T' > t)} = \frac{e^{-\lambda(t + \Delta t)}}{e^{-\lambda t}} = e^{-\lambda \Delta t} = 1 - \lambda \Delta t + o(\Delta t) \dots \dots \dots (3-5)
 \end{aligned}$$

● 情況 C：從 Active 變成 Inactive(Active -> Inactive)

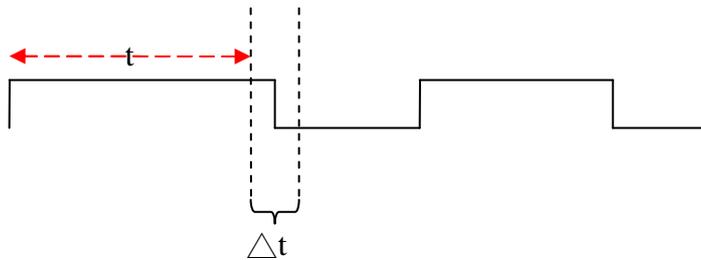


圖 3.5：情況 C(Active->Inactive)

如圖 3.5，當訊務在 t 時是 active 但是訊務卻會在 $t + \Delta t$ 之後持續 inactive，如此訊務才會從 active 轉換成 inactive，Eq. 3-6 為其機率。

$$P(t + \Delta t \geq T / T > t)$$

$$= \frac{p(t+\Delta t \geq T > t)}{p(T > t)} = \frac{p(t+\Delta t \geq T' > t)}{p(T' > t)} = \frac{(1-e^{-\lambda(t+\Delta t)})-(1-e^{-\lambda t})}{e^{-\lambda t}} = 1-e^{-\lambda \Delta t} = \lambda \Delta t + o(\Delta t) \dots \dots \dots (3-6)$$

● 情況 D：從 Inactive 變 Active(Inactive ->Active)

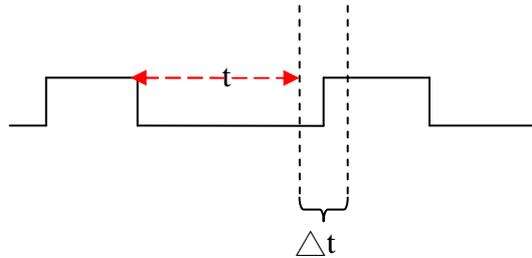


圖 3.6：情況 D(Inactive->Active)

如圖 3.6，當訊務在 t 時是 inactive 但是訊務卻會在 t + Δt 之後持續 active，如此訊務才會從 inactive 轉換成 active，Eq. 3-7 為其機率。

$$P(t + \Delta t \geq T' / T' > t)$$

$$= \frac{p(t+\Delta t \geq T' > t)}{p(T' > t)} = \frac{p(t+\Delta t \geq T' > t)}{p(T' > t)} = \frac{(1-e^{-\lambda(t+\Delta t)})-(1-e^{-\lambda t})}{e^{-\lambda t}} = 1-e^{-\lambda \Delta t} = \lambda \Delta t + o(\Delta t) \dots \dots \dots (3-7)$$

● 計算訊務狀態轉換的機率

以這一些狀態轉換的機率整理出在 t+Δt 時狀態是 active 的可能性有多高，以此得到已經 active 一段時間(t)的瞬間機率。如 Eq. 3-8 所示。

$$P_{active}(t+\Delta t) = P_{active}(t) * A + P_{inactive}(t) * D$$

$$P_{active}(t+\Delta t) = P_{active}(t) * (1 - \lambda \Delta t + o(\Delta t)) + P_{inactive}(t) * (\lambda \Delta t + o(\Delta t))$$

$$P_{active}(t+\Delta t) = P_{active}(t) * (1 - \lambda \Delta t + o(\Delta t)) + P_{inactive}(t) * (\lambda \Delta t + o(\Delta t))$$

$$\Rightarrow \lim_{\Delta t \rightarrow 0} \frac{P_{active}(t+\Delta t) - P_{active}(t)}{\Delta t} = \lim_{\Delta t \rightarrow 0} \{-\lambda P_{active}(t) + \lambda' P_{inactive}(t) + \frac{o(\Delta t)}{\Delta t} [P_{inactive}(t) + P_{active}(t)]\}$$

$$\Rightarrow \frac{dP_{active}(t)}{dt} = -\lambda P_{active}(t) + \lambda' P_{inactive}(t) \dots\dots\dots (3-8)$$

然後我們整理出在 $t+\Delta t$ 時狀態是 inactive 的可能性，以此得到已經 inactive 一段時間(t)的瞬間機率。如 Eq. 3-9 所示。

$$P_{inactive}(t+\Delta t) = P_{inactive}(t)*B + P_{active}(t)*C$$

$$P_{inactive}(t+\Delta t) = P_{inactive}(t)*(1-\lambda \Delta t + o(\Delta t)) + P_{active}(t)*(\lambda \Delta t + o(\Delta t))$$

$$P_{inactive}(t+\Delta t) = P_{inactive}(t)*(1-\lambda \Delta t + o(\Delta t)) + P_{active}(t)*(\lambda \Delta t + o(\Delta t))$$

$$\Rightarrow \lim_{\Delta t \rightarrow 0} \frac{P_{inactive}(t+\Delta t) - P_{inactive}(t)}{\Delta t} = \lim_{\Delta t \rightarrow 0} \{-\lambda' P_{inactive}(t) + \lambda P_{active}(t) + \frac{o(\Delta t)}{\Delta t} [P_{inactive}(t) + P_{active}(t)]\}$$

$$\Rightarrow \frac{dP_{inactive}(t)}{dt} = -\lambda P_{active}(t) + \lambda' P_{inactive}(t) \dots\dots\dots (3-9)$$

● 訊務 Active 的機率

由於之前已假設 active 與 inactive 機率在各時間點相同，所以 Eq. 3-8 與 Eq. 3-9 等號左邊部分的數值為 0，我們可以推出公式 Eq. 3-10， P_{active} 與 $P_{inactive}$ 各是 active 與 inactive 的機率。

$$0 = -\lambda P_{active} + \lambda' P_{inactive} \text{ 與 } 0 = -\lambda' P_{inactive} + \lambda P_{active} \dots\dots\dots (3-10)$$

然後平衡公式 Eq. 3-10 得到公式 Eq. 3-11。

$$\lambda' P_{inactive} = \lambda P_{active} \dots\dots\dots (3-11)$$

最後可以知道 active 機率與 inactive 機率的關係如公式 Eq. 3-12 所敘述。

$$P_{active} * \lambda = \lambda' * P_{inactive}$$

$$\Rightarrow P_{inactive} = (\lambda / \lambda') * P_{active} \dots\dots\dots (3-12)$$

因為訊務的狀態只有兩種，所以 active 與 inactive 的機率和是 1，我們可從由公式 Eq. 3-12 得到 active 與 inactive 的機率 Eq. 3-13。接下來以訊務 Active 機率來求出溢流機率。

$$P_{active} + P_{inactive} = 1$$

$$\Rightarrow P_{active} + (\lambda / \lambda') * P_{active} = 1$$

$$\Rightarrow P_{active} = \frac{1}{1 + \lambda / \lambda'} \text{ 與 } P_{inactive} = \frac{1}{1 + \lambda' / \lambda} \dots\dots\dots (3-13)$$

3.5 溢流機率 (Overflow Probability)

當 ACA 超額允入 Z 個訊務後，這些訊務中有 y 個訊務會 active 並且 y 的數量介於 C 與 Z 之間，這時網路可能會發生溢流。在此以圖 3.7 來解釋各種溢流情況與如何推導出溢流機率。

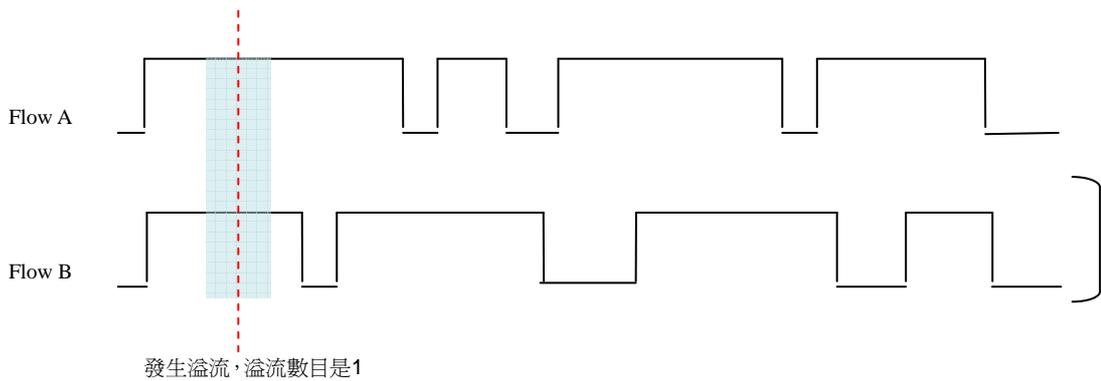
在 One Unit and Two Admitted Flows 的情況下，如圖 3.7(a)，由給定的參數可以知道訊務 active 的機率。兩個訊務都 active (p) 時會發生溢流，所以允入的訊務只要出現兩次 active 與零次 inactive 就會發生溢流，而機率是 $p^2(1-p)^0$ ，最後乘上全部的訊務都會 active 的組合量，以此得到溢流機率。

在 Two Units and Three Admitted Flows 的情況下，如圖 3.7(b)，由給定的參數可以知

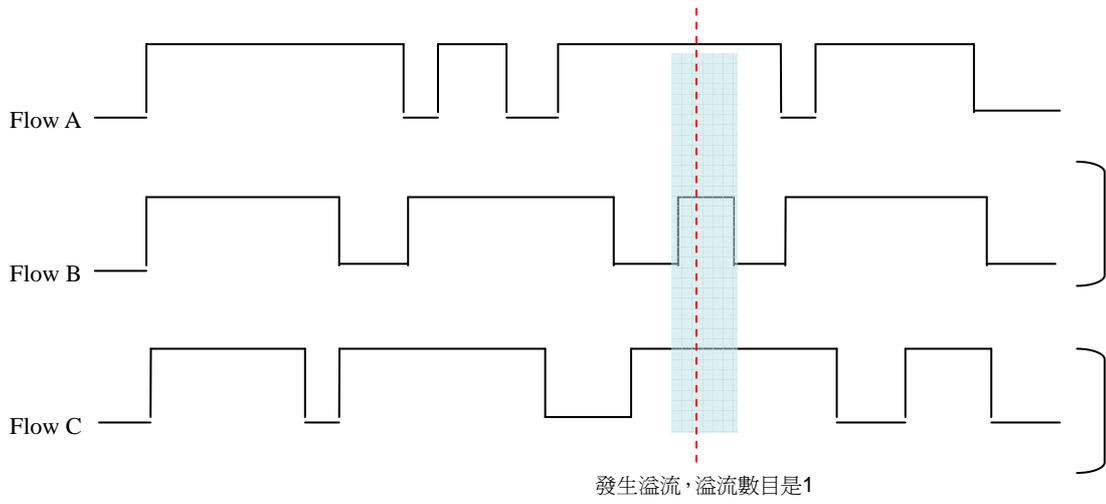
道訊務 active 的機率。三個訊務都會發生 active 時會發生溢流，再乘上三個訊務會發生 active 的組合量，以此得到溢流機率。

在 Two Units and Four Admitted Flows 的情況下，如圖 3.7(c)，由給定的參數可以知道訊務會 active 的機率，四個或三個訊務同時 active 時會發生溢流。其機率是 $p^4(1-p)^0$ 或 $p^3(1-p)^1$ 。最後將它乘上從四個訊務中有四個或三個會 active 的訊務的組合量，以得到溢流機率。

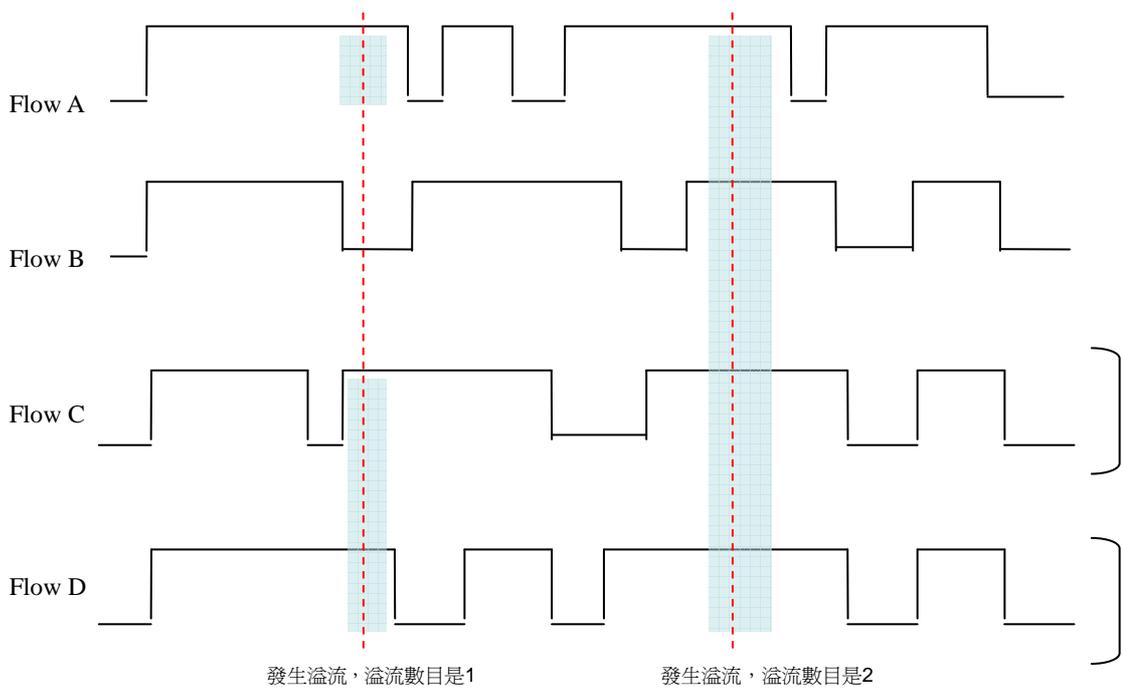
最後在 One Units and Four Admitted Flows 的情況下，如圖 3.7(d)，從給定的參數可以知道訊務會 active 的機率。四個、三個或兩個訊務同時 active 時會發生溢流，其機率是 $p^4(1-p)^0$ 、 $p^3(1-p)^1$ 或是 $p^2(1-p)^2$ ，最後也要乘上從四個訊務中取四個、三個或兩個訊務會 active 的組合量，以得到溢流機率。



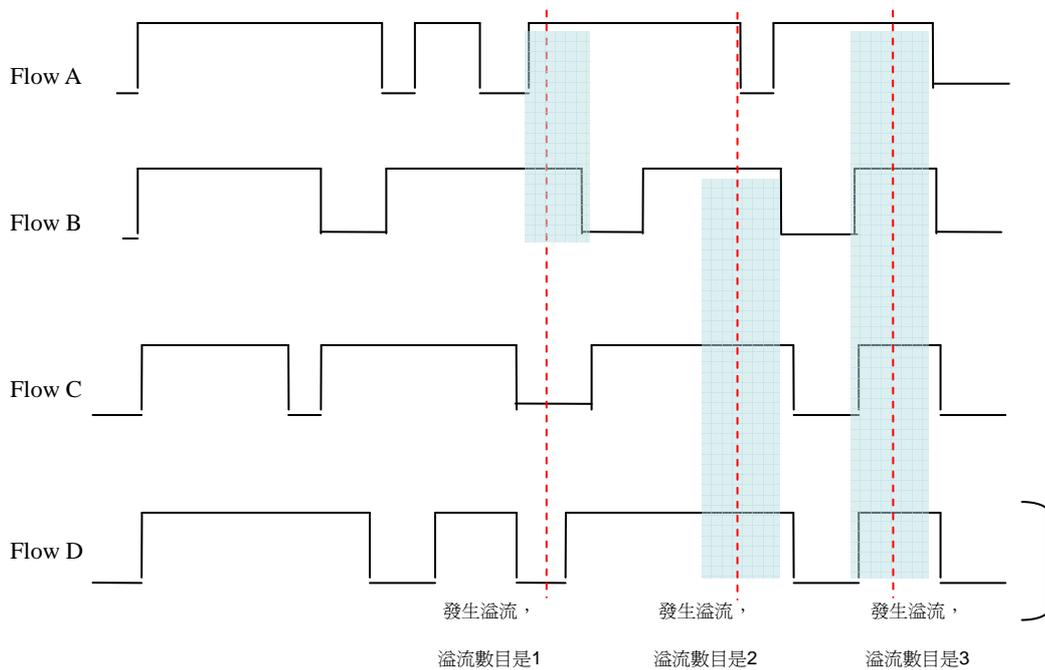
(a)



(b)



(c)



(d)

圖 3.10：溢流的狀況在(a) One Unit and Two Admitted Flows，(b) Two Units and Three Admitted Flows，(c) Two Units and Four Admitted Flows，(d) One Unit and Four Admitted Flows

本研究中 y 代表 active 的訊務數目， Z 是總允入量， p 是訊務會 active 的機率，如 Eq. 3-14 所示。

$$P(Z, y) = \binom{Z}{y} (p)^y (1-p)^{Z-y}, p = \frac{1}{(1 + \lambda / \lambda')}, p > 0 \dots\dots\dots (3-14)$$

Eq. 3-14 使用之參數如表 3.4 所示。

表 3.4：溢流機率之參數

| | |
|---|---------------|
| Z | 允入量 |
| y | Active 訊務量 |
| P | 訊務 active 的機率 |

最後整理成如 Eq. 3-15。而 ACA 可以使用本模型，以線性搜尋的方式找出最高淨收益時的允入量，ACA 可依照最佳的超額允入量允入訊務。

$$NET_PROFIT(Z) = C1 * Z - C2 * \sum_{y=C+1}^Z \left\{ (y-C) * \binom{Z}{y} (p)^y (1-p)^{Z-y} \right\}, p = \frac{1}{(1+\lambda/\lambda')}, p > 0 \dots\dots\dots (3-15)$$

表 3.5：淨收益模型之參數

| | |
|----|-------------------------|
| Z | 允入量 |
| Y | Active 訊務量 |
| P | 訊務 Active 的機率 |
| C1 | 訊務使用每單位資源每單位時間的收益 |
| C2 | 訊務因過多允入訊務，每單位資源每單位時間之損失 |

第四章

模型驗證與效能評估

我們用 NS2[18]模擬器來驗證我們的分析模型的準確度並評估超額允入法的效能。實驗分成三部分，第一部份是驗證模型的準確度，第二部分是超額允入法的效能評估，第三部分是超額允入法的 robustness 測試，最後下一個實驗的總結。

4.1 實驗目的與方法

1. 驗證模型準確度的實驗目的是使用模擬的方法以驗證模型的準確度。
2. 在超額允入法的效能評估方面，則是評估超額允入法的效能。
3. 在超額允入法的 robustness 方面，其實驗目的是在探討 ACA 無法精確掌握資源量時，ACA 使用超額允入法後系統效能所受到的影響。例如，當 ACA 同時使用超額分配與超額允入時，部分訊務可能不會失約，因此有些保留的資源就會被使用，所以保留下來的資源量會比預期的還要少。

在驗證模型中，我們所用的指標是淨收益增量與允入量、超額允入比例(Optimal Increased Admission Ratio, $(Z-C)/C$)、封包遺失比率 (packet drop/total packet) 與損失，而超額允入比例是允入增量與資源量的比例。效能評估所使用的指標是淨收益增量比率、超額允入比例、封包遺失比率 (packet drop/total packet) 與損失。在 robustness 方面的實驗中所使用的指標是淨收益增量比率與允入量。

4.2 收益與損失的定義

為了簡化實驗環境，我們假設所有語音訊務都有相同的 pps (packets per second) 與封包大小。

在分析模型中，收益與損失的意義如下：

- 收益： $C1 * \text{總通話時間} * \text{pps} = C1 * \text{允入量} * \text{平均通話時間}(\text{Mean Holding Time}) * \text{pps}$
- 損失： $C2 * \text{封包遺失數目} = C2 * \text{溢流的訊務量} * \text{平均通話時間} * \text{pps}$

在模擬系統，收益與損失有一樣的意義，而計算方式如下：

- 收益： $C1 * \text{總通話時間} * \text{pps} = C1 * \sum (\text{per call}) \text{通話時間} * \text{pps}$
- 損失： $C2 * \text{封包遺失數目}$

4.3 實驗拓撲、變因與參數

實驗的拓撲如圖 4.1 所示，這個拓撲表達出 ACA 有 C 單位的資源量以允入 Z 個訊務，當允入量增加時兩端同時增加 node 數目，在圖內左邊的 node 是 source 端，而右邊則是 destination 端，實驗在 destination 端收集封包遺失數目以計算淨收益。

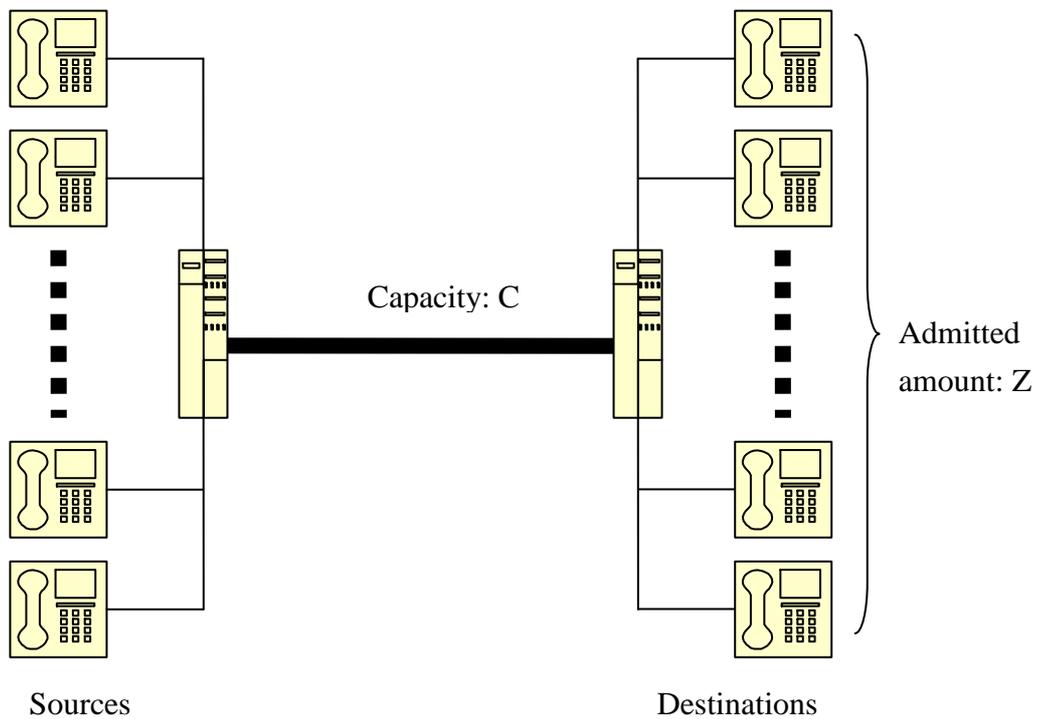


圖 4.1：實驗拓撲

在實驗內有四個變因如表 4.1 與 4.2，依次序各是 active ratio、mean holding time(HT)、capacity 與 C1:C2 比值。

表 4.1：實驗目的與變因

| 實驗目的 | 變因 |
|------|--|
| 模型驗證 | a. Active ratio b. Mean holding time (HT) |
| 效能評估 | c. Capacity d. C1:C2 |

表 4.2：實驗參數與參數範圍

| 參數 (Parameters) | 範圍 (Range) |
|------------------------|--------------|
| Active ratio | 0.1~0.9 |
| Mean holding time (HT) | 2mins, 5mins |
| Capacity | 10~30 |
| C1:C2 | 1:2,1:3,1:4 |

4.4 模型驗證

這一個實驗是要觀察模型準確度，分成兩個部分，第一部分是觀察各種參數對淨收益的誤差之影響，第二部分則是觀察各種參數對最佳允入量誤差之影響。

4.4.1 Mean Holding Time 對溢流機率之影響

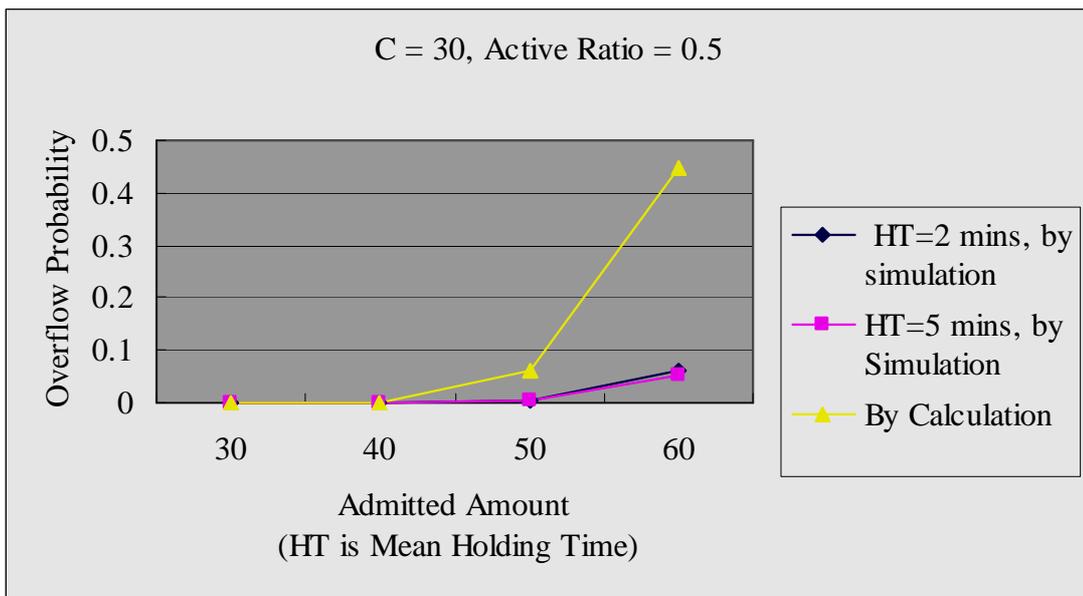


圖 4.2：Mean Holding Time 變化時允入量與溢流機率的關係

不同的 Mean Holding Time 產生的收益不同，在比較的基準不一樣之下，不能直接比較淨收益，所以比較溢流機率，我們可從圖 4.2 看出當 capacity 為 30 而允入量是 50 之前，ACA 會有準確度很高的溢流機率。在允入量超過 50 之後，誤差較大，但因預測的溢流機率偏高將使 ACA 不容易允入過多的訊務，不致於造成負面的影響。

4.4.2 Active Ratio 對淨收益誤差之影響

我們可由圖 4.3 看出 ACA 在可以收到淨收益之前，ACA 在 Active Ratio 高時使用模型會有低淨收益的誤差，而在 Active Ratio 低時其誤差也是很小。在 Active Ratio 變化的情況下我們可看出淨收益的準確度高，而淨收益誤差如表 4.3 所示。

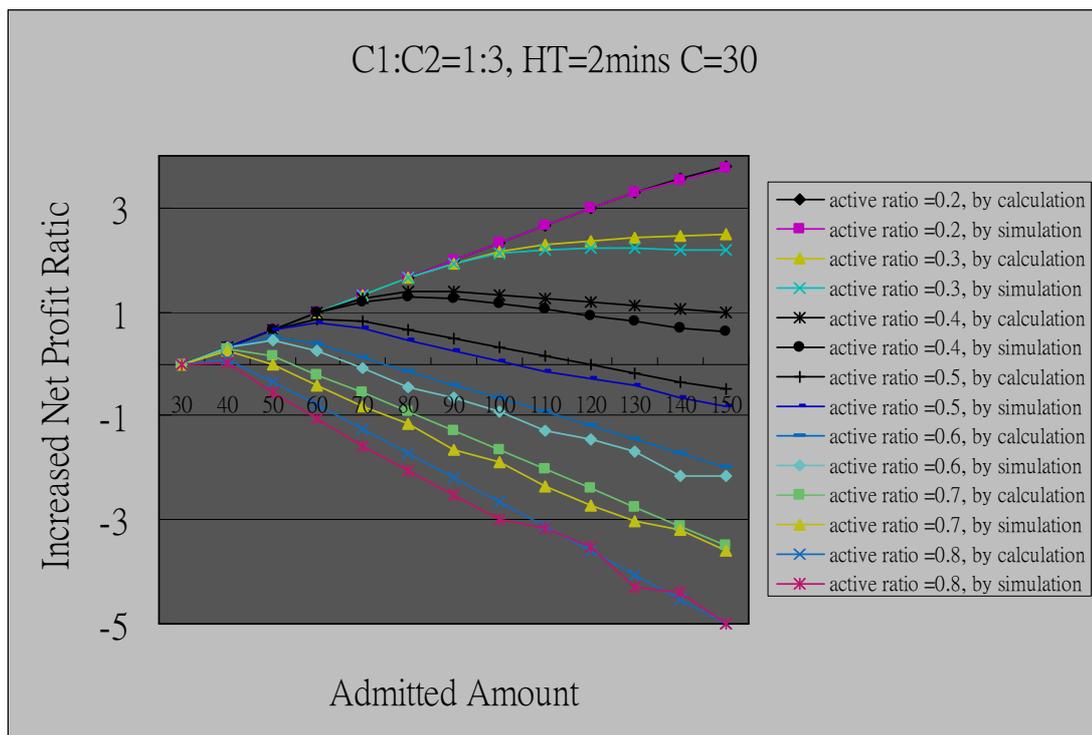


圖 4.3：Active Ratio 變化時允入量與淨收益增量的關係

表 4.3：Active ratio 變化時的誤差百分比

| | Active Ratio | | | | | | |
|---------|--------------|-------|-------|-------|-------|-------|-----|
| | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 |
| 平均誤差 | 0.09% | 1% | 2% | 5% | 5% | 6% | 8% |
| 最小誤差 | 0% | 0.00% | 0.00% | 0.01% | 0.00% | 0.60% | 1% |
| 最大誤差 | 0.50% | 4% | 6% | 8% | 13% | 12% | 12% |
| 最佳允入量誤差 | 3% | 3% | 2% | 2% | 2% | 1% | 1% |
| 最佳淨收益誤差 | 6% | 5% | 5% | 4% | 2% | 1% | 0% |

4.4.3 Capacity 對淨收益誤差之影響

我們可從圖 4.4 觀察到當 Capacity 變化時，在 ACA 仍可收到淨收益之前，ACA 可以在低 Capacity 時有較低的誤差。在高 Capacity 時，ACA 可以在最佳點之前有較低的誤差，如表 4.4 所示，最佳淨收益誤差在 2% 之內、最佳允入量誤差在 6% 之內。

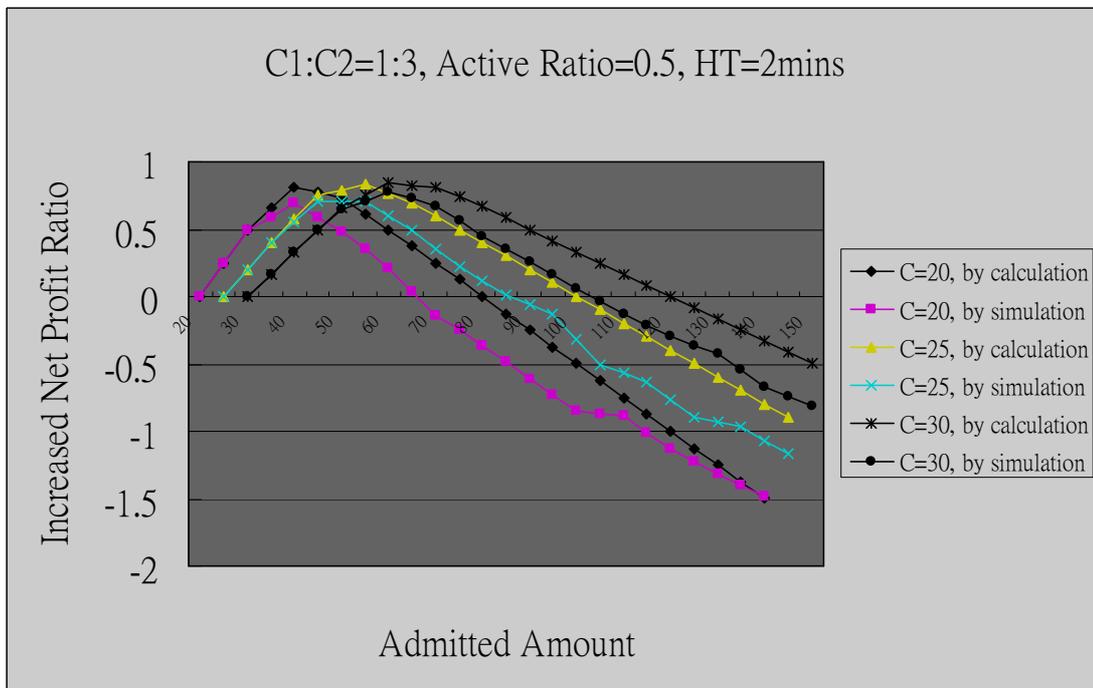


圖 4.4：Capacity 變化時允入量與淨收益增量的關係

表 4.4：Capacity 變化時的誤差百分比

| | Capacity | | |
|---------|----------|-------|-------|
| | 20 | 25 | 30 |
| 平均誤差 | 3% | 4% | 5% |
| 最小誤差 | 0.05% | 0.01% | 0.01% |
| 最大誤差 | 6% | 8% | 8% |
| 最佳允入量誤差 | 6% | 5% | 2% |
| 最佳淨收益誤差 | 2% | 3% | 4% |

● 最大淨收益之誤差

在這一實驗內，我們觀察當允入量為最高時在各個 capacity 下所收得的淨收益誤差，我們可以從圖 4.5 看出在最佳允入點的淨收益，當變化 capacity 數值時，淨收益誤差的變化幅度(<4%)比較小，如表 4.6 所示。

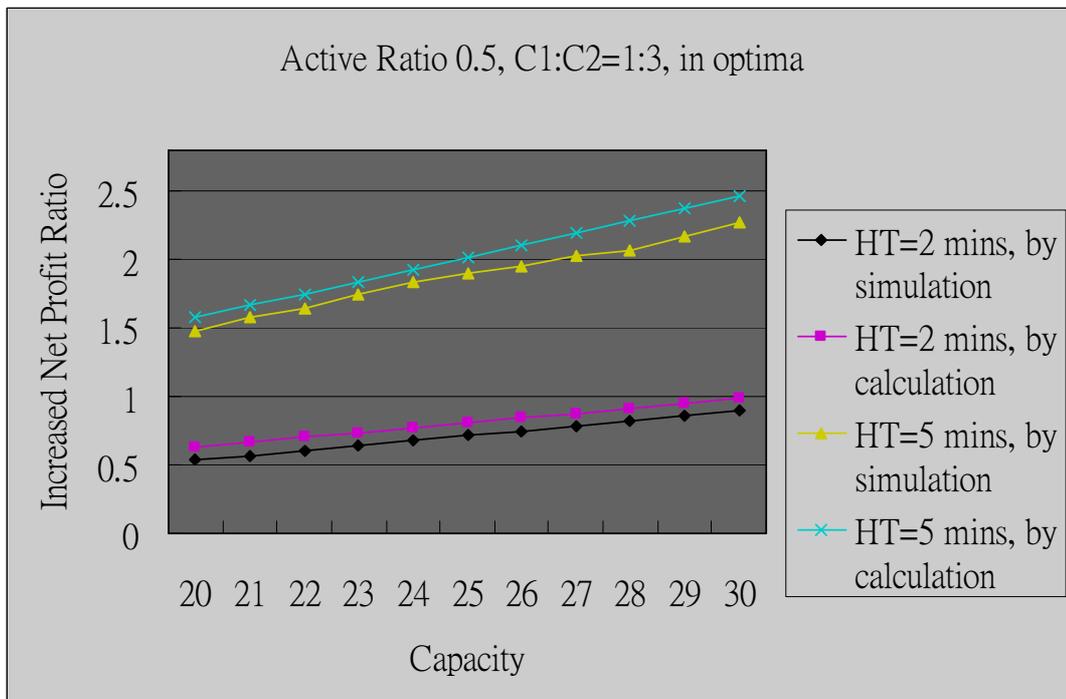


圖 4.5：Capacity 變化時最佳允入量上之淨收益增量的關係

表 4.5：Capacity 變化時在最佳允入量上的誤差百分比

| Capacity | Mean Holding Time | |
|----------|-------------------|-------|
| | 2mins | 5mins |
| 20 | 2% | 2% |
| 25 | 3% | 3% |
| 30 | 4% | 3% |

4.4.4 C1:C2 對淨收益誤差之影響

我們由圖 4.6 可看出淨收益為正值之前，隨者 C2 增加，最佳點之前誤差均很小，如表 4.6 所示最佳允入量誤差是在 3%之內、最佳淨收益誤差是在 5%之內。

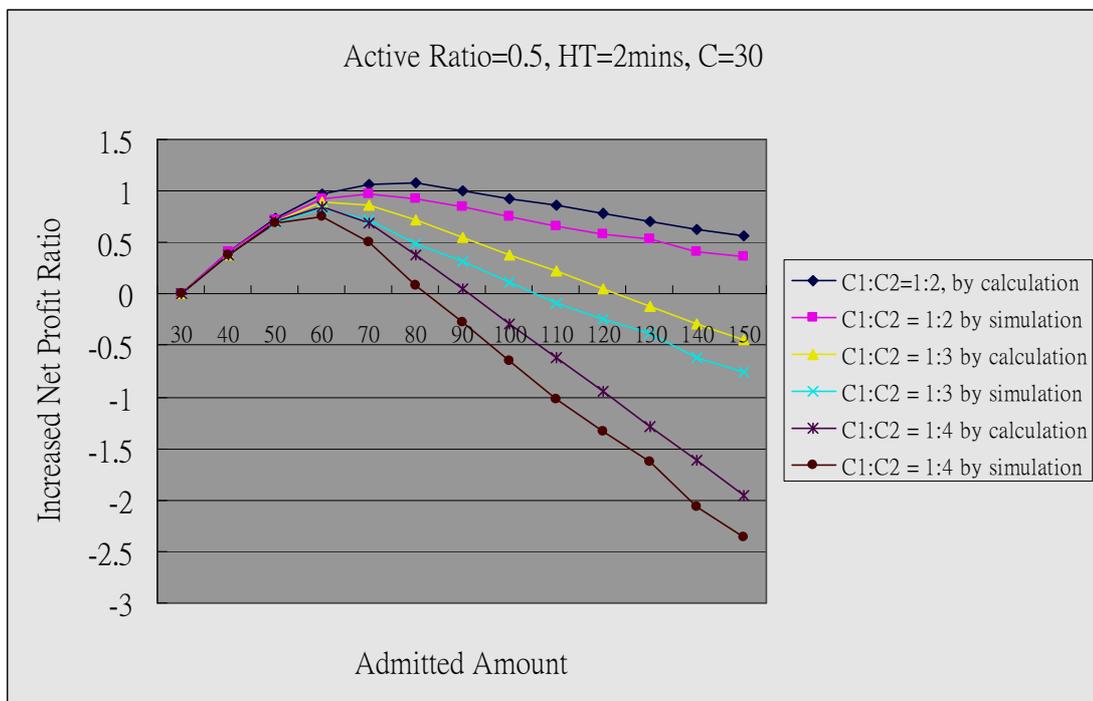


圖 4.6：C1:C2 變化時允入量與淨收益增量的關係

表 4.6：C1:C2 變化時各種誤差百分比

| | C1:C2 | | |
|---------|-------|-------|-------|
| | 1:2 | 1:3 | 1:4 |
| 平均誤差 | 4% | 5% | 6% |
| 最小誤差 | 0.00% | 0.01% | 0.01% |
| 最大誤差 | 2% | 4% | 7% |
| 最佳允入量誤差 | 1% | 2% | 3% |
| 最佳淨收益誤差 | 3% | 4% | 5% |

4.4.5 Active Ratio 對最佳超額允入比例誤差之影響

從圖 4.7 我們可看出當 Active Ratio 較低時，誤差較大 (<4%)，其原因是本模型的誤差是來自於溢流的機率，因為在簡化模型時的假設條件，假設 Active 與 Inactive 與時間無關，意即，訊務 Active/Inactive 的機率在各時間點都相同，與假設所進入的訊務會一直在網路中，這個假設所造成之誤差，導致溢流機率的估計難免有誤差，大量允入訊務會增加模型的誤差，ACA 在 Active Ratio 變化的情況下可看出本模型之誤差在 4% 之內，如表 4.7 所示。

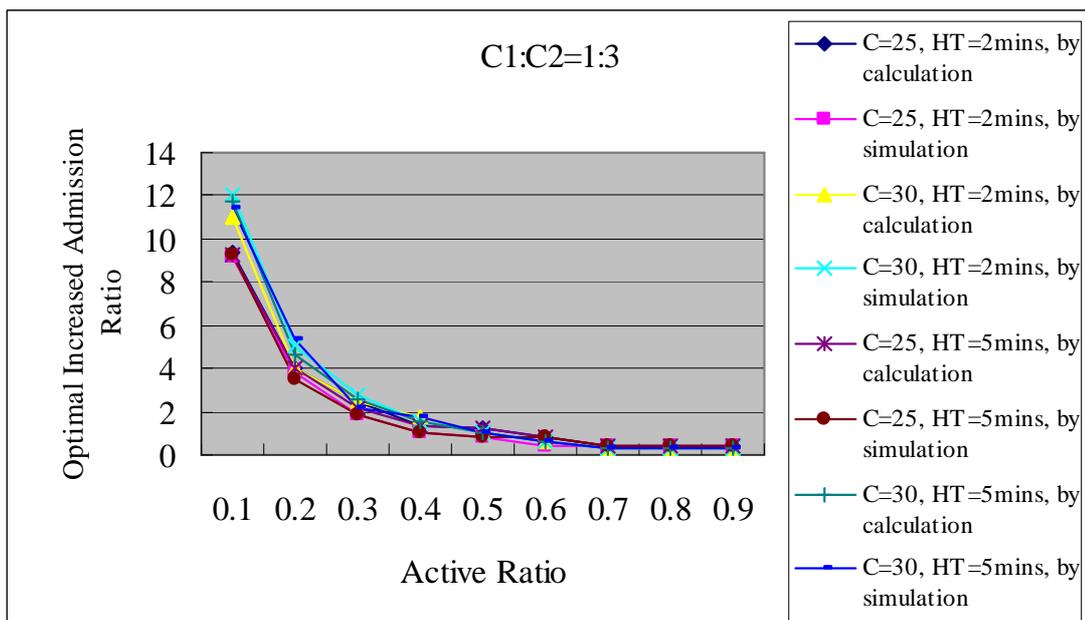


圖 4.7：Active Ratio 與超額允入比例的關係

表 4.7：Active Ratio 變化時各種誤差百分比

| Mean Holding Time | Active Ratio | | |
|-------------------|--------------|-----|-----|
| | 0.1 | 0.5 | 0.9 |
| 2 mins | 4% | 3% | 0% |
| 5 mins | 3% | 2% | 0% |

4.4.6 Capacity 對最佳超額允入比例誤差之影響

從圖 4.8 可看出 Active Ratio 變化時，彼此的最佳允入比例會有一定的差距，而在 Capacity 變化時，誤差的變化會很平穩，我們可看出在 Capacity 變化的情況下本模型的誤差是在 6% 之內，如表 4.8 所示。

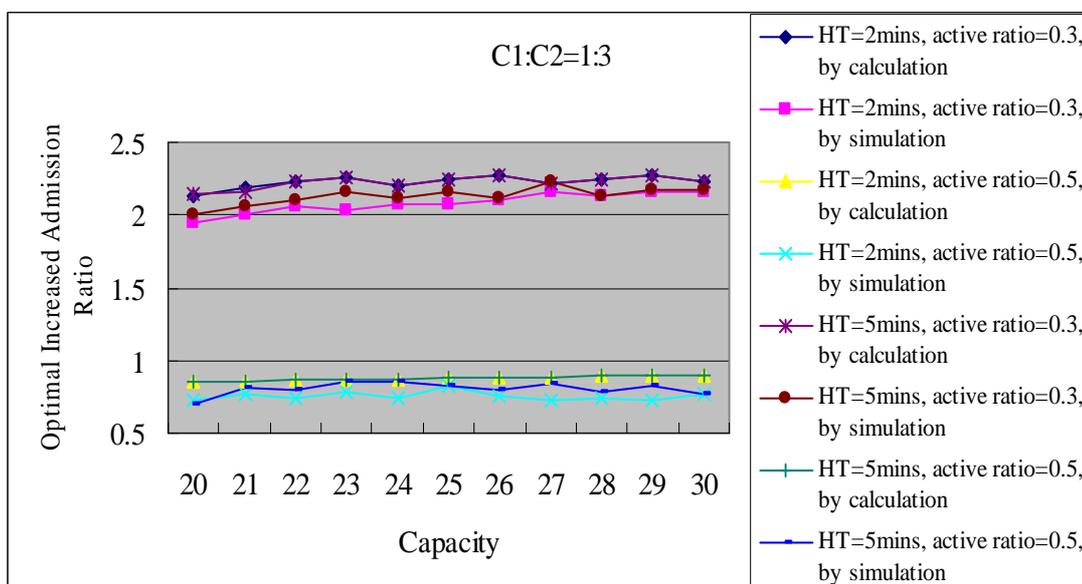


圖 4.8：Capacity 與超額允入比例的關係

表 4.8：Capacity 變化時各種誤差百分比

| Mean Holding Time | Capacity | | |
|-------------------|----------|----|----|
| | 20 | 25 | 30 |
| 2 mins | 6% | 5% | 6% |
| 5 mins | 5% | 4% | 4% |

4.4.7 C1:C2 對最佳超額允入比例誤差之影響

從圖 4.9 中可看出最佳超額允入比例對 Active Ratio 變化有比較高的敏感性，隨者 C2 增加，誤差會微幅增加，如表 4.9 所示 C2=4 時誤差到達 7%。

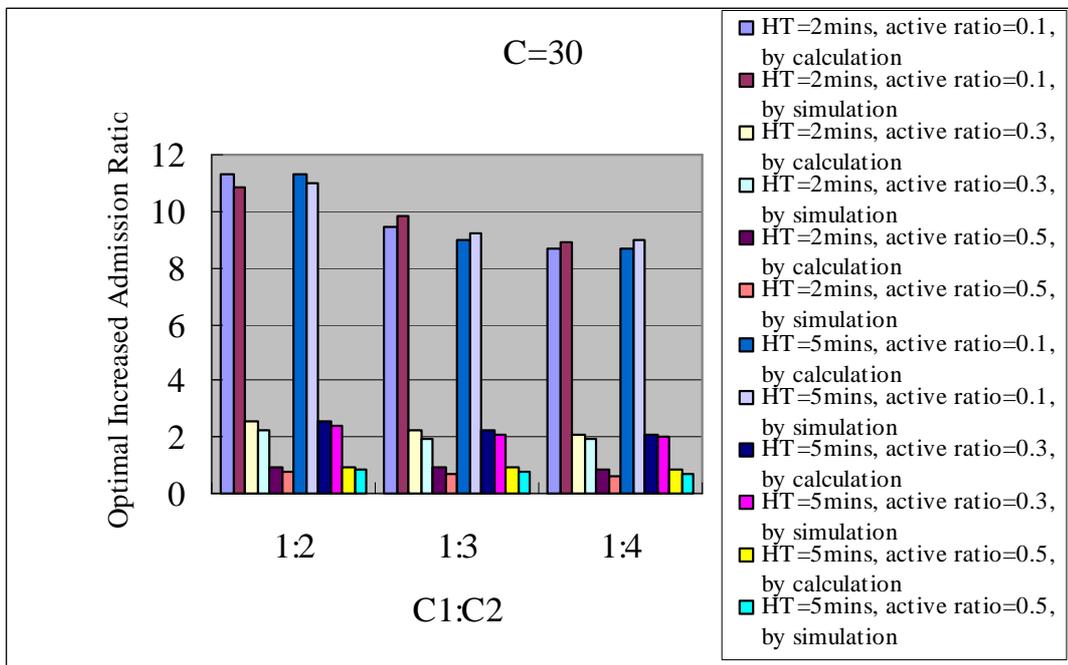


圖 4.9：C1:C2 與超額允入比例的關係

表 4.9：C1:C2 變化時各種誤差百分比

| Mean holding time | C1:C2 | | |
|-------------------|-------|-----|-----|
| | 1:2 | 1:3 | 1:4 |
| 2 mins | 4% | 6% | 7% |
| 5 mins | 3% | 4% | 5% |

● **模型驗證的結論**

實驗顯示出在大部分情況下分析模型的準確度是很高，在實驗中，最佳允入比例誤差最高是 7% 而最佳淨收益誤差最高是 7%。即使在準確度低的情況下，ACA 可以使用最佳允入量之前的量值以提升淨收益與準確度。

4.5 參數變化對淨收益之影響

本章節將使用模擬以觀察出超額允入法的好處。

4.5.1 Active Ratio 變化對淨收益之影響

我們從圖 4.10、4.11 與表 4.10 可發現 Active Ratio 與淨收益的關係，Active Ratio 越低越能提升淨收益，而封包遺失的情況並不嚴重。當 Active Ratio 越高時淨收益提升至多一倍，封包遺失的情況也不嚴重。大致上，超額允入法能提升淨收益的範圍從 1 倍到 5.8 倍，我們可以知道模型可以有效的提升淨收益。

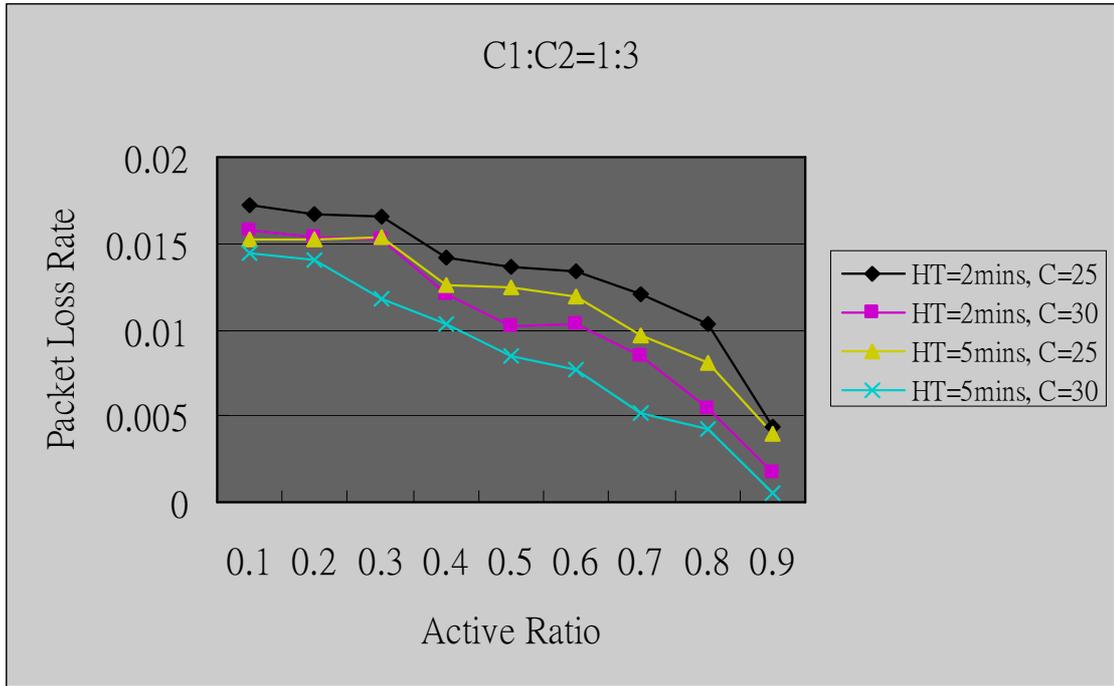


圖 4.10：Active Ratio 與封包遺失率的關係

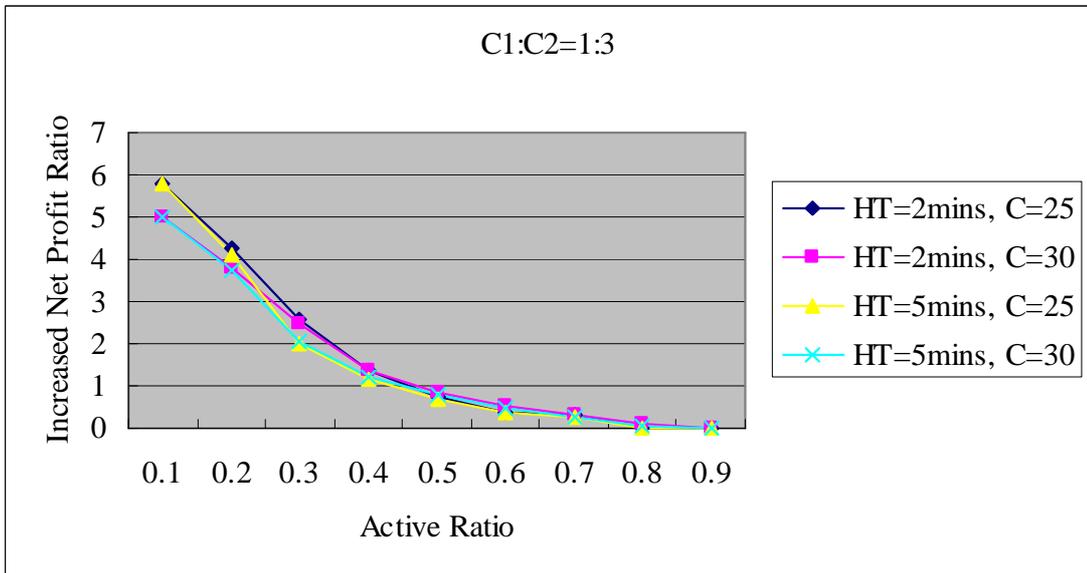


圖 4.11：Active Ratio 與淨收益增量比率的關係

表 4.10：Active Ratio 變化時的提升百分比

| Mean Holding Time | Capacity | Active Ratio=0.4 | Active Ratio=0.6 | Active Ratio=0.8 |
|-------------------|----------|------------------|------------------|------------------|
| 2mins | 25 | 114% | 44% | 24% |
| 2mins | 30 | 118% | 72% | 10% |
| 5mins | 25 | 117% | 39% | 1% |
| 5mins | 30 | 119% | 48% | 3% |

4.5.2 Capacity 變化對淨收益之影響

從圖 4.12、4.13 與表 4.11 可看出 Capacity 與淨收益之關係，Capacity 的變化會影響到淨收益提升的速度，ACA 仍然會有效的提升淨收益，而封包遺失數目會隨者 Capacity 升高而下降。

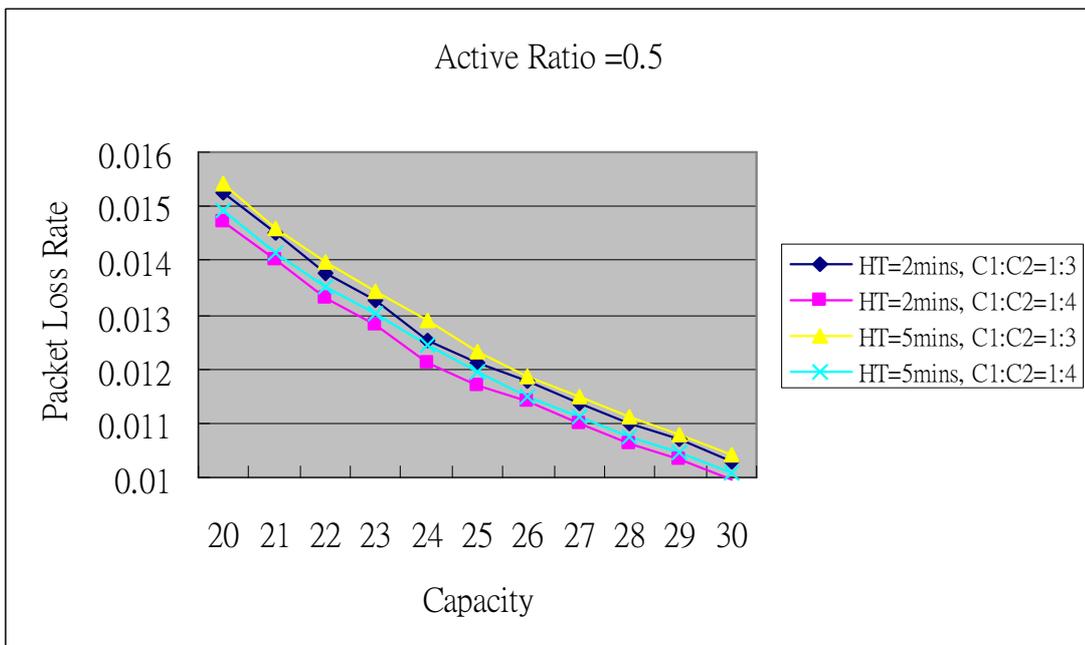


圖 4.12：Capacity 與封包遺失率的關係

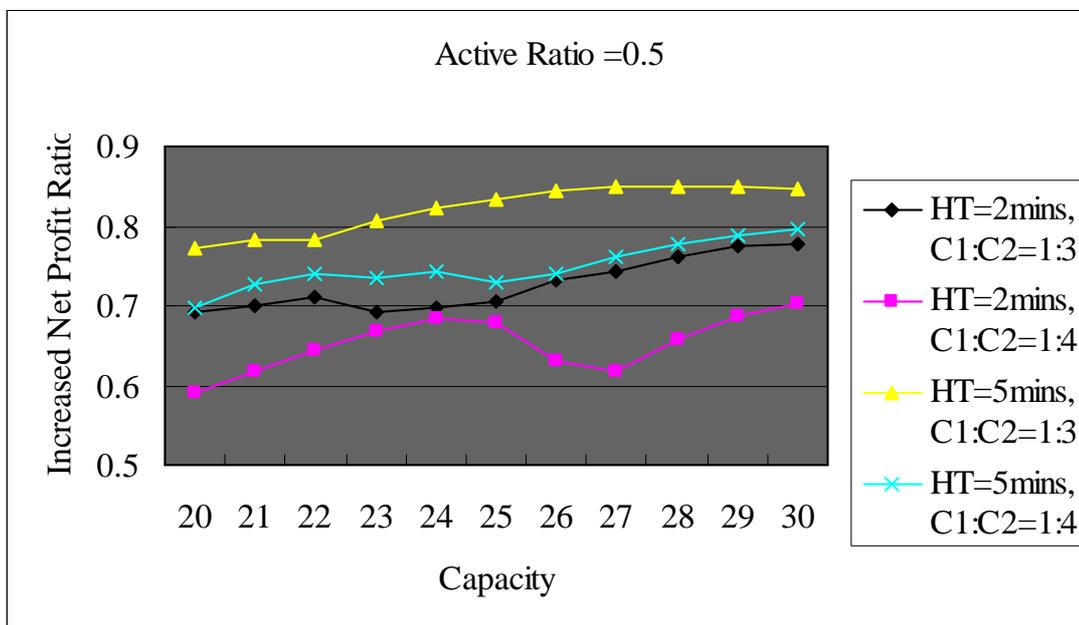


圖 4.13：Capacity 與淨收益增量比率的關係

表 4.11：Capacity 變化時的提升百分比

| Mean Holding Time | C1:C2 | Capacity=20 | Capacity=25 | Capacity=30 |
|-------------------|-------|-------------|-------------|-------------|
| 2mins | 1:3 | 69% | 70% | 77% |
| 2mins | 1:4 | 59% | 67% | 70% |
| 5mins | 1:3 | 78% | 83% | 84% |
| 5mins | 1:4 | 72% | 73% | 77% |

4.5.3 C1:C2 變化對淨收益之影響

從圖 4.14、4.15 與表 4.12 可看出，隨者 C2 增加時，淨收益增量的比率其減少幅度很小，而封包遺失的情形並不嚴重，超額允入仍然會有效的提升淨收益。

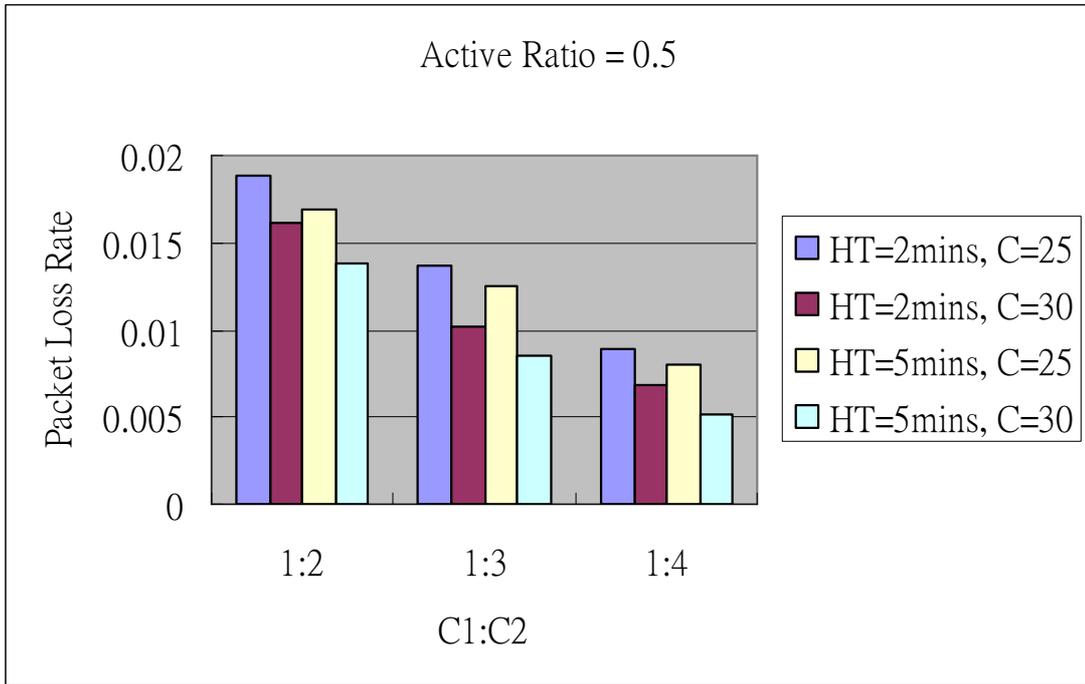


圖 4.14 : C1:C2 與封包遺失率的關係

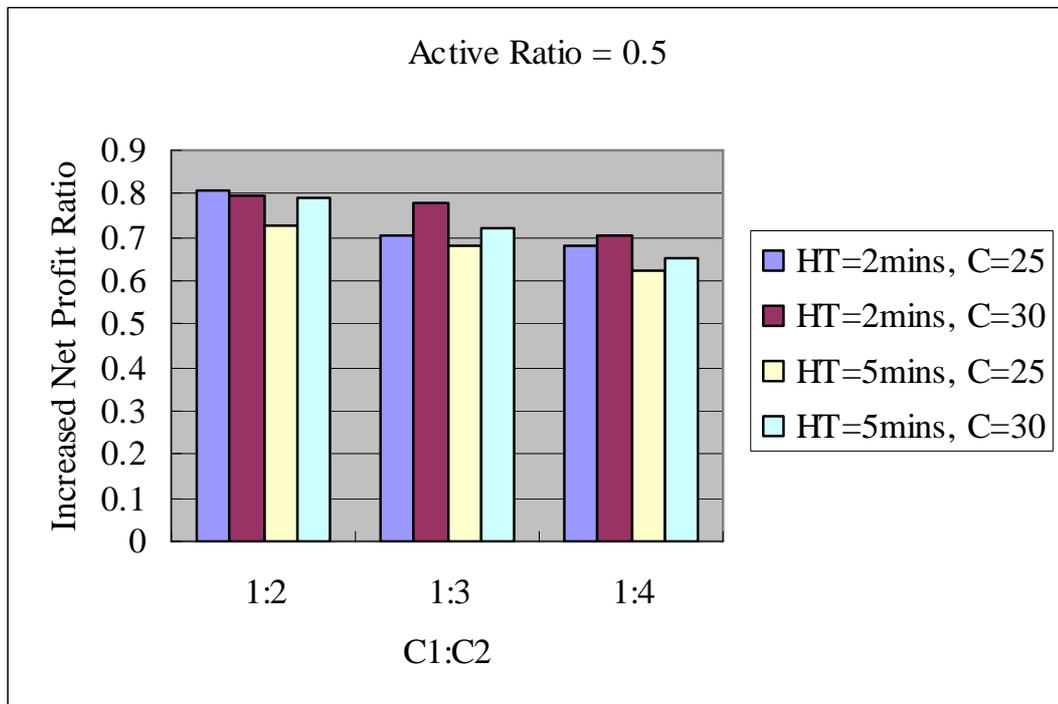


圖 4.15 : C1:C2 與增加淨收益增量比率的關係

表 4.12：Active ratio 變化時的提升百分比

| C1:C2 | 提升百分比 |
|-------|-------|
| 1:2 | 77% |
| 1:3 | 75% |
| 1:4 | 71% |

- **參數變化對淨收益之影響的結論**

實驗顯示出超額允入可以有效的提升淨收益，尤其在 Capacity 大與 Active Ratio 小時超額允入可以提升較佳的淨收益，由實驗顯示淨收益最高可以提升高達 5 倍，即使在其他情況下，ACA 使用超額允入法至少也能增加淨收益 1 倍。當提升淨收益時，封包遺失的情況並不嚴重。

4.6 超額允入法的 robustness

此部份的實驗目的是探討當 ACA 無法精確掌握資源量時，ACA 使用超額允入法後，系統效能所受到的影響，圖 4.16 係這一個實驗的拓撲，Assumed Capacity 的意義是假定的資源量，Allocated Capacity 的意義是 ACA 實際配給的資源量。假設 ACA 以為有 C 單位的資源，但 ACA 實際配置的資源量可能有短少，而 ACA 卻以 C 為基礎超額允入 Z 個訊務，本節將探討其對系統的衝擊。

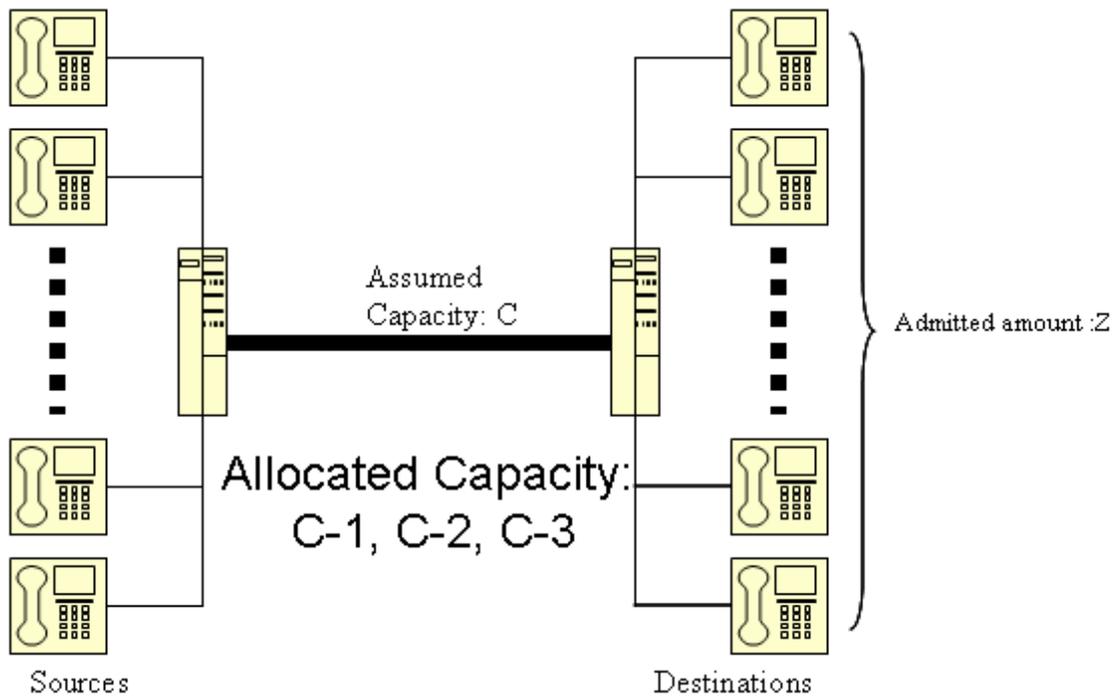


圖 4.16：資源無法精確掌握時的實驗拓撲

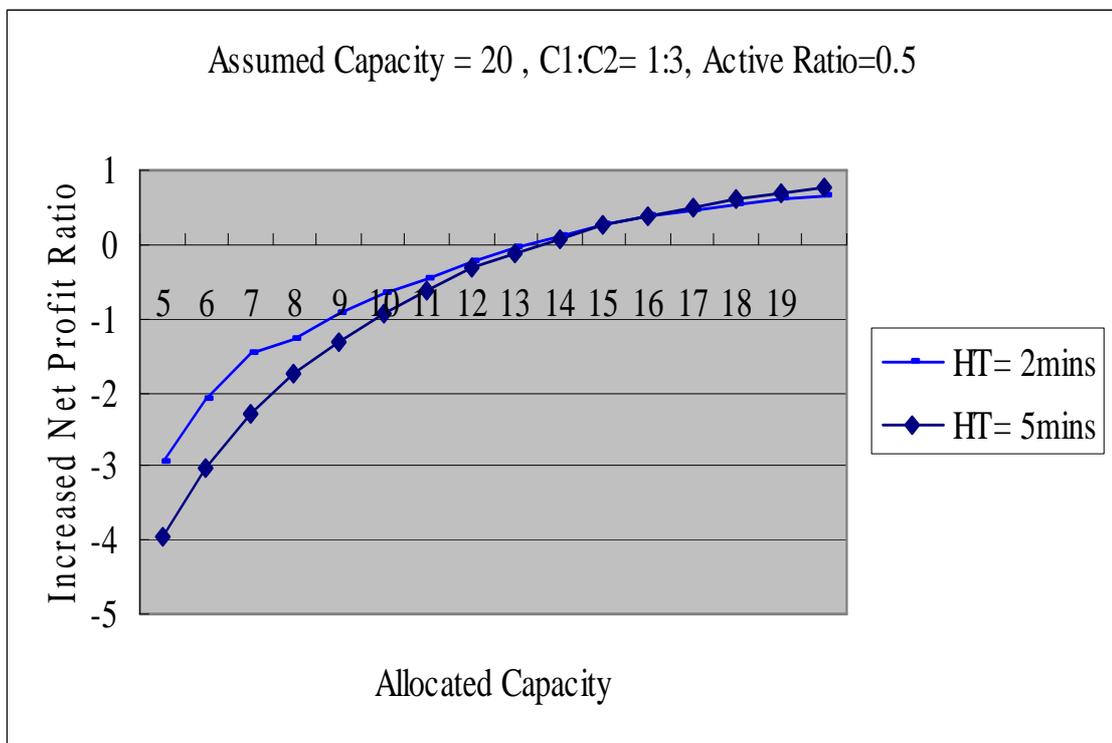


圖 4.17：資源短缺時對淨收益增加比率所受影響(Capacity=20)

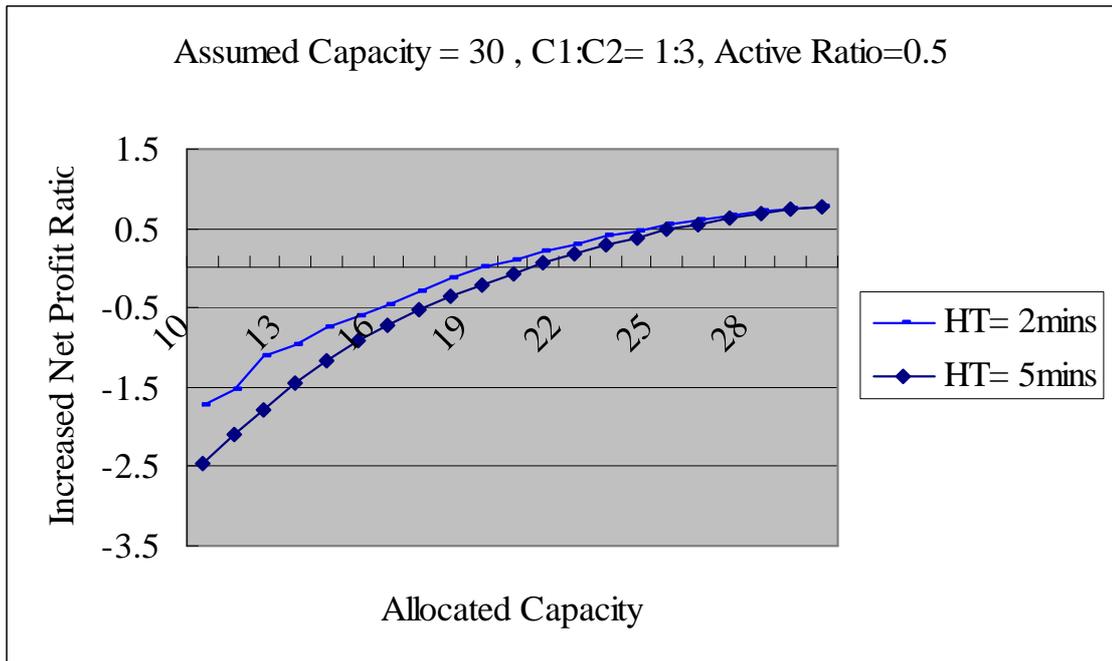


圖 4.18：資源短缺時對淨收益增加比率所受影響(Capacity=30)

由圖 4.17 與 4.18 可以看出當 Assumed Capacity 是 20 時，Allocated Capacity 最多可以不足 7 單位，本法仍可增加淨收益。而當 Assumed Capacity 是 30 時，Allocated Capacity 最多可以不足 11 單位，本法仍可增加淨收益。

- 資源無法精確掌握時對系統效能之影響的結論

實驗顯示，Allocated 與 Assumed Capacity 差距在一定範圍之內，ACA 仍可使用超額允入法以提升系統效能。

4.7 實驗結論

這一個實驗有三個主要的結論，在模型驗證方面，在大部分情況下，ACA 使用模型的準確度高，雖然在有些情況下我們可以看到模型的準確度不高，但是 ACA 可以使用最佳超額允入量以提升模型準確度與淨收益。在超額允入法的效能評估方面，ACA 可以

使用超額允入法以提升淨收益，當 Capacity 大與 Active Ratio 小時，ACA 最高可以提升 5.8 倍的淨收益，ACA 可以提高淨收益從 1 倍到 5.8 倍而封包遺失情況並不嚴重。在資源無法精確掌握時的效能評估方面下，當 Allocated 與 Assumed Capacity 在一定的資源單位範圍之內，ACA 仍可使用超額允入法以增加淨收益。

第五章

結論

在 AIPN 中因為使用者可能未依約充分使用保留之資源而導致資源使用率低落，尤其是語音訊務，根據 Brady 的研究，一般使用者平均只有 40% 的時間是在說話[3]，針對這一種情況，網路的營運者需要找出一個簡易的方法以提升整體資源的使用率，本研究使用超額允入法在資源保留為基礎的系統中以提高淨收益。

在本研究內，我們先以數個假設簡化了實際的環境，並以語音訊務的行為建立起分析模型，分析模型包括有收益與損失，收益是允入訊務對系統效益之提升的程度，而損失則是因過度允入訊務所產生溢流對訊務品質之損害，本研究以此計算出適當的允入量，並以 NS2 模擬方式驗證模型與評估本法之效能。

實驗顯示，本模型可以有相當高的準確度，雖然部分情況的準確度不高，但是 ACA 可以使用最佳允入量之前的量值以提高準確度與淨收益。在提升淨收益時，有些網路可能無法精確知道資源使用的情況，因此即使當資源無法精確掌握時，本法仍可容忍在一定範圍之內的資源量的短缺。

本研究提出的分析模型仍有需要改進之處，因為假設了 Active Ratio 與 Inactive Ratio 在各時間點都一樣，並假設所進入的訊務會一直在網路中，所以溢流機率與 holding time 無關，所以 ACA 使用超額允入法會產生誤差，不過誤差所帶來的影響不大，未來希望可以去除此假設以符合實際的環境。另外本研究只討論單一訊務，並未研究在 AIPN 中各類訊務的超額允入法，未來希望能夠結合不同的訊務型態以運用在實際的環境之下。

參考文獻

- [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.*, Oct. 1999.
- [2] AQUILA project, <http://www-st.inf.tu-dresden.de/aquila/>.
- [3] Brady, P., “a model for on-off speech patterns in two-way conversation,” *Bell System Technical Journal*, Sep. 1966.
- [4] D. Black, M. Carlson, E. Davies, Z. Wang, “An Architecture for Differentiated Services,” *RFC 2475*, Dec. 1998.
- [5] Coskun Cetinkaya, Vikram Kanodia, and Edward W. Knightly, “Scalable Services via Egress Admission Control,” *IEEE Transactions on Multimedia*, vol. 3, no. 1, Mar. 2001.
- [6] E. Crawley, Editor, L. Berger, S. Berson, “A Framework for Integrated Services and RSVP over ATM,” *RFC 2382*, Aug. 1998.
- [7] 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, Apr. 2002 pp. 139-154.
- [8] Heinanen, J., F. Baker, W. Weiss, and J. Wroclawski, “Assured Forwarding PHB Group,” *RFC 2597*, Jun. 1999.
- [9] Leonard Kleinrock, *Queueing systems vol I: theory*, pp. 347, 1974.
- [10] Dong Lin, “Constant-Time Dynamic ATM Bandwidth Scheduling for Guaranteed and

- Best Effort Services with Overbooking,” *Proc. IEEE INFOCOM’97*, 1997.
- [11] Pascal Lorenz, “Quality of service and new architectures for future telecommunications networks,” *MILCOM 2000 - IEEE Military Communications Conference*, no. 1, Oct. 2000 pp. 695-698.
- [12] Yao-Nan Lien, Chien-Tung Chen, Ming-Chih Chen, Tsung-Hsung Li and Yi-Min Chen, 2005, “QoS Management for All-IP Core Networks,” *Proceedings of The 11-th Mobile Computing Workshop*, Mar. 31, 2005.
- [13] Yao-Nan Lien and Yi-Min Chen, 2005, “Forecasting Error Tolerable Resource Allocation for All-IP Networks,” *3rd International Conference on Information Technology: Research and Education*, Jun. 27-30, 2005.
- [14] Miras, D., “Network QoS Needs of Advanced Internet Applications,” *Internet2 - QoS Working Group*, Nov. 2002.
- [15] Carlos Pazos and Mario Gerla, “Improving Internet Traffic Transport over ABR Backbones Through Bandwidth Overbooking,” *Proc. IEEE Globecom ’98*, 1998.
- [16] TEQUILA project, <http://www.ist-tequila.org/>.
- [17] Bhuvan Urgaonkar, Prashant Shenoy, and Timothy Roscoe, “Resource Overbooking and Application Profiling in Shared Hosting Platforms,” *Proceedings of the Fifth Symposium on Operating Systems Design and Implementation (OSDI)*, Dec. 2002.
- [18] K., K. Varadhan, The ns2 Manual, <http://www.isi.edu/nsnam/ns/>, Jun. 2005.
- [19] Jacobson, V., K. Nichols, K. Poduri, “An Expedited Forwarding PHB,” *RFC 2598*, Jun. 1999.
- [20] Xiao, X., L. -M. Ni, “Internet QoS: A Big Picture,” *IEEE Network*, vol.13, pp.8-18, Mar.-Apr. 1999.