

第二章

相關研究(Related Work)

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。以下將就管理架構與管理方式討論目前著名的研究。

2.1. QoS 管理架構

2.1.1 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。

IntServ 之重大的缺點在於建立 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，然而實際網路上卻有數量龐大且壽命極短的資料流，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)上受到嚴重限制，不適用於大型網路。

2.1.2 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，執行相同 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 2.1，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，兩者可能執行不同的 PHB，對同類別的資料可有不同的 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。

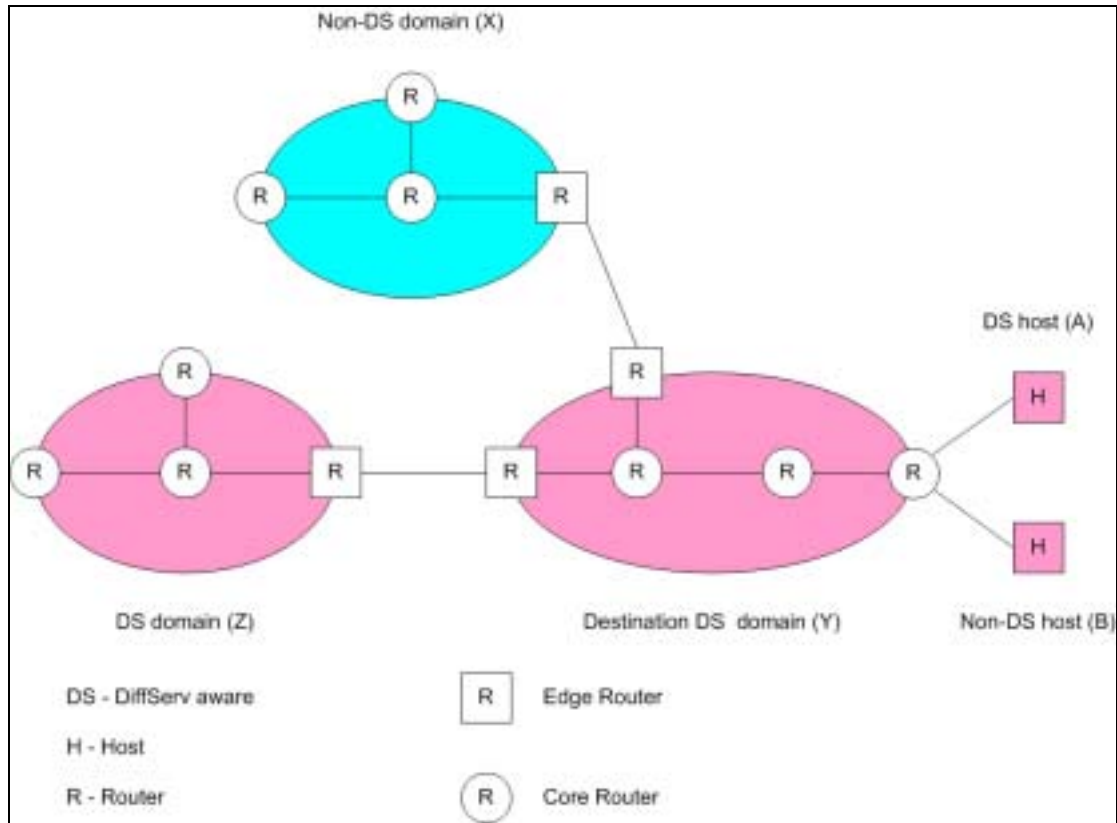


圖 2. 1 : DiffServ Domain & Non-DiffServ Domain

DiffServ 架構的設計主要有兩個機制來管理和控制網路上的資料傳遞：

Classifying

依據如來源和目的地的 IP 位址、應用程式、埠號(port number)、或是網路通訊協定等來分類。

Policing

Metering – 測量某一資料流的 flow rate 與 burst size 等參數，這些即時的數據可以提供給其他如 Marker、Shaper、Dropper 等控制網路流量或計費之參考

Shaping – 控制同一個資料流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 – 根據上面的結果放棄封包來減低網路負荷。

不同性質的應用所產生的資料有不同的品質特性。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，部份的資料並不要求時效，但不能忍受遺缺或錯誤。

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)[9]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)[10]：

Best Effort – 支援沒有特別需求與還沒有 DiffServ(backward compatibility)的網路訊務。 [11]

Assured Forwarding – 依據不同類別資料的 drop precedence 來決定當網路壅塞時可捨棄之封包，並不以優先權區別不同的訊務。

Expedited Forwarding – 降低任何網路壅塞時可能的延遲，以支援高時效性的服務。

表 2.1 列出了各個 PHB 適於使用的各種不同應用。

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，如欲在 All-IP 網路上使用 DiffServ，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

2.1.3 TEQUILA

The Traffic Engineering for Quality of Service in the Internet at Large Scale (TEQUILA) 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」 [12]，就是 TEQUILA 計畫中的 QoS 管理架構。

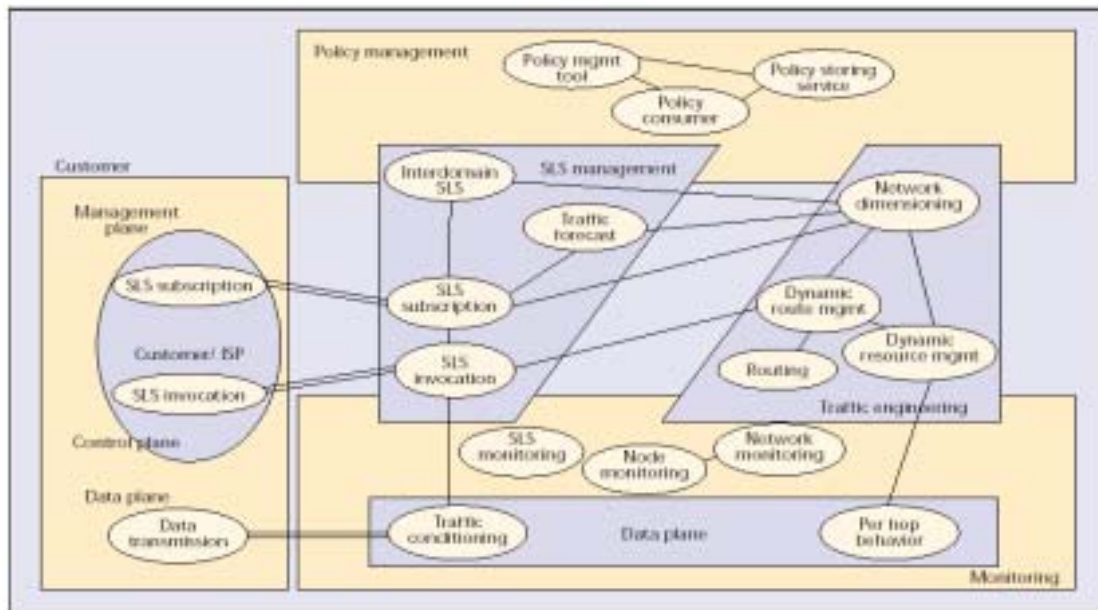


圖 2.2 : TEQUILA 架構

TEQUILA 的架構(如圖 2.2)可分為三個主要的部分，Service Level Specifications Management (SLS Management)主要負責處理客戶的服務品質要求，Traffic Engineering 是網路中主要負責 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」[13]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」 [14]中則說明關於允入控制等相關的議題。

2.1.4 Victor O.K. Li'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[15],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。

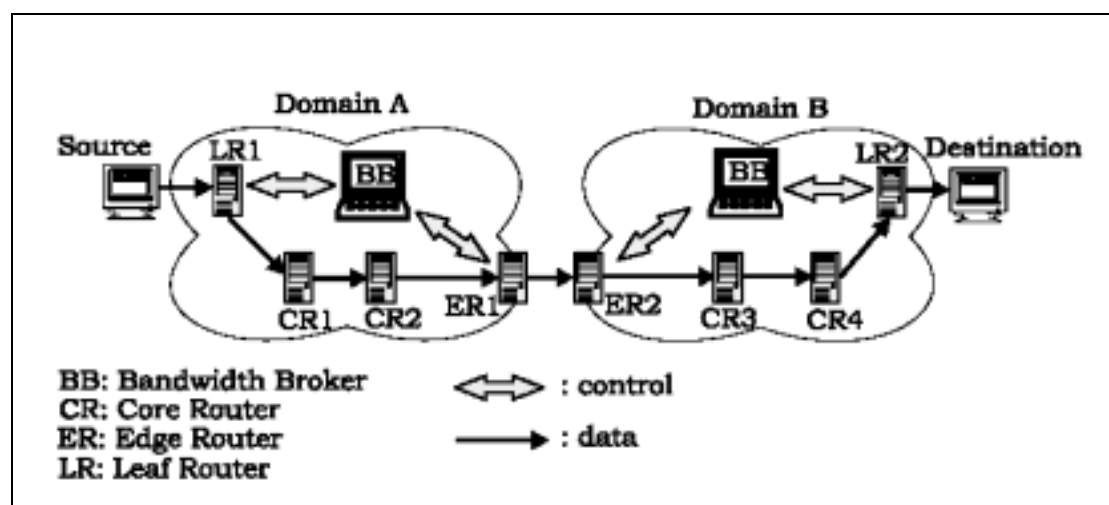


圖 2.3 : 分散式管理架構

不同於以往集中管理資源之作法,此架構以核心網路為管理單位,核心網路之各個 Ingress Router 與各個 Egress Router 間有數條預先計算好的路徑,路徑的頻寬資源由 BB 分配,再由 Ingress Router 進行允入控制。每隔一段固定時間, BB 檢視各個 Ingress Router 之路徑資源使用效能,依據平均資源使用情況與最新資源使用情況調配路徑資源。分配資源時,若 BB 僅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 Edge Router 允入量差異過大。

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 網路(如 UMTS)尚未提及。由於不同類型的 QoS 網路對於 QoS 服務類別有不同的定義，當跨 QoS 網路時則必須考慮到服務類別對映(mapping)的問題，必須以恰當的對映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 評論(Summary)

如果對每一個訊務以 real-time on-demand 去要求資源建立路徑以提供端對端的品質保證，其 overhead 將極為可觀，例如 IntServ 使用 RSVP 的方式預留資源，必須針對每個訊務的需求在核心網路內逐步搜尋，得到一條保留路徑，因其 overhead 太大，只能適用於小型網路。而 DiffServ 雖然以 per-aggregate 的方式減低管理的複雜度，但是這種 per-aggregate 的服務品質保證，卻沒有辦法對於單一個訊務提供絕對的端對端服務品質保證。

現階段的資源管理多以集中的方式配置資源，網域的允入控制由一元件統一執行。這樣的作法於資源易受限於核心網路大小與網路流量，當網域過大或者流

量過多，除中央統籌元件負荷量重與網路 overhead 過大外，在實際執行上也有困難存在。

在上述的架構中，在路徑規化與資源分配上，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，本論文之研究目標在於提出一簡單的端對端品質服務架構，以預算分配為基礎，使用簡化的單一品質參數，使得系統業者可以調整自己的品質管理策略，來增加系統效能，減低管理效率。除此之外，系統佈署(deployment)的容易與否也是我們考量的重點之一。

在本研究架構，我們以分層管理的方式解決問題，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，依照核心網路各個 Ingress Router 需求不同進行適當的分配，由 Ingress Router 內部元件事先進行路徑規劃，接下來再依照規劃結果對於訊務進行 per-flow 允入控制。在下面章節先簡介 BBQ 管理架構，以 Ingress Router 作為研究重點，提出兩套資源管理方法於 Ingress Router 執行，分別是預先頻寬批購與執行時段頻寬管理。