點對點資訊分享系統在有線及無線網路上之設計與實作

（一）研究計畫之背景及目的

1、 研究計畫之背景

隨著資訊傳輸技術的快速發展，網際網路上的傳輸內容已逐漸由單純的純文字傳輸，轉變成為多媒體資料傳輸。這樣的改變不但代表網際網路內容提供者(Internet Content Provider)能夠提供更多樣的資訊給網際網路使用者，也代表著多媒體網路的世代已經來臨。事實上，整個多媒體相關產業已經因為網際網路傳輸技術的快速發展，產生龐大的商機。此商機主要來自於多媒體應用，以及通訊與網路服務之間的結合。兩者之間技術的整合到實際應用的過程，將帶動整個多媒體網路的發展，也將主導未來的技術發展與市場趨勢。換言之，在多媒體網路服務越來越普遍的現在，分散式多媒體系統技術的研究也變得日益重要。目前的分散式多媒體系統主要使用客戶端—伺服器架構。然而，隨著客戶端數目快速增加，伺服器容易負載過重，而產生延展性不佳的問題。為了避免伺服器成為系統瓶頸之所在，IP 群播成為一個非常熱門的研究課題。然而，經過一二十年的發展，目前 IP 群播仍然無法普遍應用，主要有下列三個因素：一，延展性不佳。群播路由器需要紀錄每個群播群組的狀態，使得其成為網路瓶頸所在。二，對傳輸層以及更上層功能的支援性不佳。如可靠群播，壅塞控制等等，目前都沒有具體有效的方案。三，網際網路服務提供者不支援 IP 群播。由於無法普遍應用於目前網際網路，且在收費機制方面的技術無法突破，使得目前只有少數的網際網路服務提供者支援 IP 群播。為了解決這個問題，近一兩年來許多研究提出應用層群播(Application Level Multicast)的概念，希望將群播服務由原先的網路層改為應用層來提供。隨著相關架構與技術的日益成熟，應用層群播技術已經成為目前最可能解決客戶端—伺服器架構延展性不佳問題的方案。因此，本人未來三年的研究，將以應用層群播為領域，以實作高延展性的多媒體網路傳輸平台為手段，以相關的應用層群播網路傳輸技術，以及點對點(Peer-to-Peer, P2P)網路分享技術為內容，進行連續性多年期計畫性之研究。以下先就相關技術之背景進行說明。

(1) 應用層群播網路技術之相關背景

為了使群播服務得以在現今網路架構下普遍被應用，應用層群播成為近兩年非常熱門的研究議題。在應用層群播網路中，資料是透過同一群播群組中節點與節點間建立的單播(Unicast)連線所形成樹狀結構的疊加網路(overlay network)來傳送，這樣的網路亦被稱之為疊加樹(overlay tree)。使用應用層群播架構主要有下面幾個好處：一，應用容易。應用層群播與現今網路架構相符合，不需更動現有的網路協定與硬體。二，延展性佳。應用層群播不需網路所有路由器都支援 IP multicast 功能，封包轉送的工作改由群組成員或特定伺服器負責，路

由器不需記錄群播群組的狀態，故不再成為網路瓶頸。目前有關應用層群播的研究主要針對兩個議題：

一、疊加樹的建立：此部份主要探討在不同的應用程式需求下(如高頻寬，低傳輸延遲等等)，新成員如何加入疊加樹的問題。

二、疊加樹的維護：此部份包括成員離開，斷線修復，疊加樹路徑最佳化等議題。

目前有關應用層群播的研究，依照疊加樹的形成方式，可分為節點間先形成網絡，然後再形成疊加樹的網絡式(Mesh First)，以及直接形成疊加樹的樹狀結構式(Tree First)兩類；而從疊加樹的節點組成來分類，則可分為由特定伺服器所組成的架構式(infrastructure based)，純粹由群播群組內的成員所組成的點對點式(Peer to Peer based)，以及由伺服器與群播群組內的成員共同組成的混合式(mix based)。表一為目前應用層群播較具代表性研究的分類表。

表一：服務疊加網路相關研究分類表

	網絡式	樹狀結構式
架構式	Scattercast	Overcast
點對點式	NARADA	ALMI，TBCP，TAG，P2Cast
混合式		CoopNet

Scattercast[1]是一個以代理伺服器(proxy server)所形成的疊加樹來提供群播服務的系統，客戶端則透過與代理伺服器的連線來取得資料。其疊加樹是由許多稱為 Scattercast 代理伺服器(SCattercast proXy，SCX)的節點所形成。其主要是針對如電子白板這類多個資料來源，大型群組(數千個客戶端以上)，低傳輸延遲 (Low latency)要求的多媒體應用程式而設計。在疊加網路的建立上，Scattercast 利用下面兩個步驟來對每個資料來源的建立各自的疊加樹(Overlay Tree)：

(A) 網絡(Mesh)建立：一個新加入的 SCX 透過一個進入點(Bootstrap node)知道所有 SCX 的資訊，並且隨機的與其他 K 個的 SCX 建立單播連線。K 為一系統預設值。當越來越多 SCX 加入疊加網路時，這些 SCX 就形成一個類似網絡的網路拓墣。

(B) 疊加樹建立：由於傳輸延遲是主要的效能考量，因此每一個資料來源在疊加網路上根據 SCX 相互之間量測的傳輸延遲利用 DVMRP 這類的群播路由協定計算出資料來源到每個 SCX 的最端路徑而形成疊加樹。

此外，每個 SCX 會定期的隨機選取一個 SCX 來進行傳輸延遲(propagation delay)測量，並透過一個最佳化演算法來保持網絡拓墣上 SCX 間路徑的最佳化。

相對於 Scattercast 利用代理伺服器建立起疊加網路， NARADA[2]利用純粹由群組成員所形成的疊加網路來提供群播服務。換句話說，NARADA 屬於一個點對點式疊加網路系統。其主要針對視訊會議這類多個資料來源，小型群組(約數十個客戶端)，低傳輸延遲要求的多媒體應用程式而設計。在疊加網路的建立上，NARADA 採用的方法與 Scattercast 大致相同，都是以網絡建立與疊加樹建立兩個步驟來建立每個資料來源的疊加樹。唯一的不同是由於 NARADA 應用於成員數少的多媒體應用程式，每個成員都會與所有其他成員建立單播連線。因此，當因為成員離開或成員因故無法繼續提供傳輸服務時，其疊加樹上的子成員可以迅速的找到替代的父成員來避免連線中斷，故在系統強度(Robustness)方面會有很好的表現。此外，每個成員會定期與隨機選出的數個成員進行傳輸延遲量測，根據這些量測結果利用一個最佳化演算法來保持疊加網路成員間路徑的最佳化。

與 NARARA 相同，ALMI[3]同樣應用於多個資料來源，小型群組，低傳輸延遲要求的多媒體應用程式。然而，ALMI 在疊加網路的建立則是採用中央控制機制。新成員先向一個稱為會議控制者(Session Controller)的主機送出加入訊息，由會議控制者來指定新成員在疊加樹中的父成員。此外，ALMI 疊加網路的一個特色是多個資料來源利用一個分享疊加樹(shared overlay tree)來傳送資料，而非每一個資料來源擁有各自的疊加樹。由於 ALMI 是使用分享樹的架構，因此其疊加樹是屬於斯坦那最小樹(Steiner Minimum Tree，SMT)，而之前所提多資料來源疊加網路系統所建立的疊加樹則是屬於最小擴張樹(Minimum Spanning Tree，MST)。

為因應寬頻網路時代的市場發展，許多廠商都不斷的找尋更有效率的網路多媒體傳輸方法。為了使得使用者不需下載整個媒體檔案，就可以即時觀賞媒體提供者所提供的網上節目，串流(streaming)技術成為各家廠商爭相投入研發的方向。目前主要的串流技術方案有 Microsoft 的 Microsoft Windows Media Technologies[4]，Real Network 所提出的 Real Player[5]，以及 Apple 所提出的 Apple QuickTime[6]等。隨著這股發展趨勢，許多研究也提出利用應用層群播網路提供串流技術的解決方案。

Overcast[7]是一個以代理伺服器所組成的疊加網路來提供群播服務的多媒體系統。其主要針對如隨選視訊這類單一資料來源，大型群組，高傳輸效能(High throughput)要求的多媒體應用程式而設計。為了達到高傳輸效能的要求，疊加網路的建立是以頻寬作為主要考量因素。整個 Overcast 架構可分為下面三個部份：

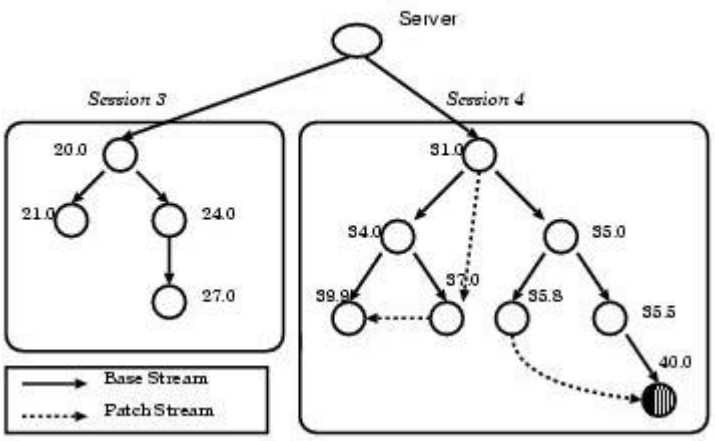
(A) 疊加網路的建立：在 Overcast 中，疊加網路是由許多具有快取功能的代理伺服器所組成的疊加樹來代表。因此，疊加網路中每個節點即代表一個代理伺服器。當一個新的節點欲加入疊加樹來提供服務時，由於頻寬是主要的考量因素，且為了

不使多媒體伺服器因為過多的連線而形成系統瓶頸，新節點會由多媒體伺服器開始，以遞迴方式往疊加樹下游方向，找尋一個距離多媒體伺服器最遠，且能夠提供最大頻寬的節點並與之連線來成為疊加樹的成員。

(B) 疊加網路的維護：在疊加樹的維護方面，每個節點會定期將其所有疊加樹上的子節點的連線狀態，所能提供的最大頻寬，以及所在位置(以 IP 位址表示)等資訊，整合之後回報給父節點。因此，多媒體伺服器擁有所有節點的資訊。此外，每個節點會定期與父節點以及祖父節點進行頻寬量測。當一節點發現其父節點無法提供服務時或無法提供足夠頻寬時，此節點會先嘗試與祖父節點連線，若祖父節點同樣無法提供服務時或無法提供足夠頻寬時，此節點會以遞迴方式往疊加樹上游方向尋找直到找到一個可提供服務的父節點為止。

(C) 客戶端連線：在 Overcast 中，一個疊加網路可由一個 URL 來表示。此 URL 主要包含多媒體伺服器位址以及媒體檔案位置。一個客戶端可利用瀏覽器送出一個帶有的疊加網路 URL 的 HTTP GET 訊息給多媒體伺服器要求開始接收媒體內容。多媒體伺服器會根據媒體檔案位置，客戶端位置，以及所有代理伺服器的目前狀態，指定一個疊加網路中的節點提供客戶端連線來取得媒體內容。由於多媒體伺服器擁有疊加網路中所有節點的狀態，因此可以很快的決定出可供客戶端連線的節點代理伺服器而不需要在網路上發送訊息給所有節點來詢問相關的狀態，如此就可以達到快速連線的目的。

另一方面，P2cast[8]則是一個以點對點式服務疊加網路來提供隨選視訊服務的多媒體系統。其疊加網路的建立與 Overcast 相同，新成員都是找尋能夠提供最大頻寬的成員作為疊加樹的父節點。為了使得加入時間點不同的各個成員能夠取得完整的視訊內容，P2cast 提出一個利用點對點間的「修補(patching)」技術來提供隨選視訊服務。在 P2cast 中，加入時間點相近的成員組成一個群播群組，而同一群播群組較晚加入的成員則找尋一個之前加入的成員作為「修補伺服器(patch server)」來取得加入時間點之前的媒體資料，如圖一所示。



圖一：P2cast **執行過程（引自**[8]**）**

圖一為 P2cast **在時間點** 40 **時的執行過程。圖一中每個圓圈代表一個成員，圓圈旁邊的數字則代表加入時間。此外，系統定義每個群組間的時間間隔為** 10 **個單位時間。因此，這個例子中包含兩個開始於時間點** 20 **的** session 3**，以及開始於時間點** 31 **的** session 4**。**P2cast **中的每個節點都可同時提供兩種串流的轉送，一個是直接由疊加樹中父成員接收到，包含完整視訊內容的基本串流**(base stream)**，另一個則是由修補伺服器所接收到，包含從群組開始時間點到加入時間點之間視訊內容的修補串流**(patch stream)**。以圖一右下角加入時間點為** 40 **的成員為例，其指定加入時間點為** 35.5 **的成員為疊加樹中的父成員來接收基本串流，並指定加入時間點為** 35.8 **的成員為其修補伺服器來接收時間點** 31 **到** 40 **之間的視訊內容。**

一般商業用的多媒體伺服器端應用程式，為了服務眾多的客戶端，通常設置在具有強大計算能力，且擁有充裕對外頻寬的伺服器上。CoopNet[9]**基於這樣的特性，提出一個結合架構式應用層群播與點對點式應用層群播特性的混合式應用層群播串流系統。**CoopNet **保留的傳統客戶端—伺服器架構，也就是仍然有一個伺服器直接傳送資料給客戶端，而只有在伺服器的負載過重無法提供更多的連線時，新加入的客戶端才由其他的客戶端取得資料。此外，為了增加系統強度，**CoopNet **將串流資料分為許多的訊框群組**(Group Of Frame，GOP)**，利用編碼技巧將訊框群組分割為一組封包，並將這一組封包透過不同的疊加樹傳送。客戶端可同時由不同的疊加樹接收串流資料。只要接收到一定比例的封包就能夠將還原整個訊框群組。使得客戶端不會因為一個疊加樹成員的離開而無法接收所有的串流資料。**

隨著技術發展日益成熟，許多廠商也推出以應用層群播為基礎的串流媒體網路平台。如 Allcast[10], vTrails[11], **以及** Bluefalcon[12]**。這些產品宣稱能夠提供實況串流以及隨選視訊等服務，由於沒有詳細的公開文件，我們無法詳細了解其運作的原理。**

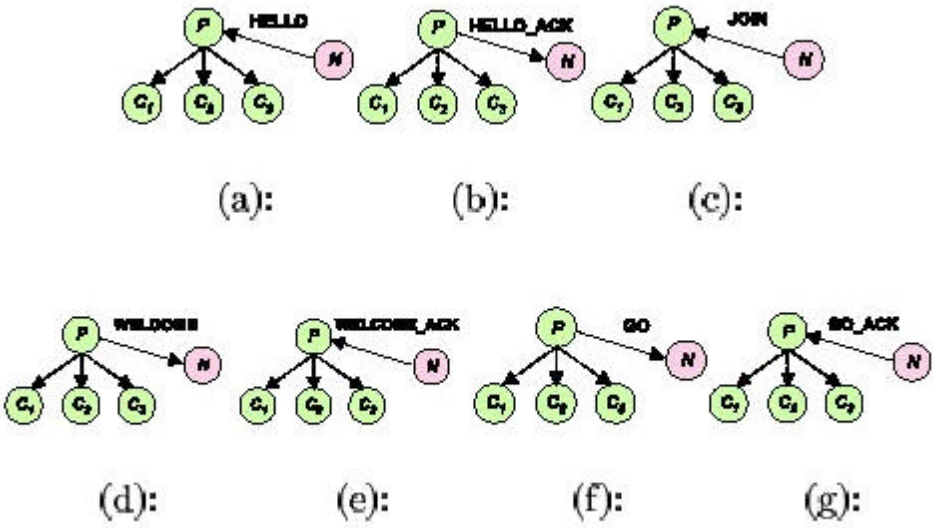
除了針對特定的多媒體應用程式所設計的應用層群播系統，一些研究也針對應用層群播疊加網路的建立提出通用的解決方案，如 TBCP[13]與 TAG[14]。TBCP 提出一個讓新加入成員依照加入時間點疊加樹的狀態來找到最佳父節點。其提出一個由伺服器開始，由上而下(Top-down)，以費用函數 (cost function)為考量因素的樹狀結構疊加網路建立方案。TBCP 的特色是為了達到高延展性，每一個節點只提供固定個數的連線。我們利用圖二來介紹新成員加入服務疊加網路的步驟：

(1) 新加入成員 N 送出一個 HELLO 訊息給候選父節點 P。一開始候選父節點設為伺服器。(圖二(a))
(2) 候選父節點 P 送出一個包含所有子節點的 HELLO_ACK 訊息給新成員 N。(圖二(b))
(3) 依照價值函數的定義，新成員 N 與候選父節點與其子節點進行頻寬或傳輸延遲的量測，並把結果利用 JOIN 訊息傳送回候選父節點。(圖二(c))
(4) 候選父節點 P 將所有包括 P，N，與其所有子節點所形成的服務疊加子樹形成方式，利用價值函數估算出 N 的最佳父節點。圖三為當每個節點的最大連線數為 4 時，所有可能的服務疊加子樹形成方式。
(5) 如果 P 決定成為 N 的父節點，P 送出 WELCOME 訊息給 N，當 N 送回 WELCOME_ACK 訊息作為回應之後，N 即成功加入此服務疊加網路。(圖二(d)與圖二(e))
(6) 否則，當 N 或或一個 P 的子節點(我們稱此點為 C_j)被指定為 P 的一個子節點之子節點時(我們稱此點為 C_k)，P 會送出一個 GO(C_k)的訊息給 C_j，C_j 則利用 GO_ACK 訊息回應此訊息。此時，C_k 成為新的候選父節點，而 C_j 成為新的新加入成員，再回到由步驟(1)開始重新執行。(圖二(f)與圖二(g))

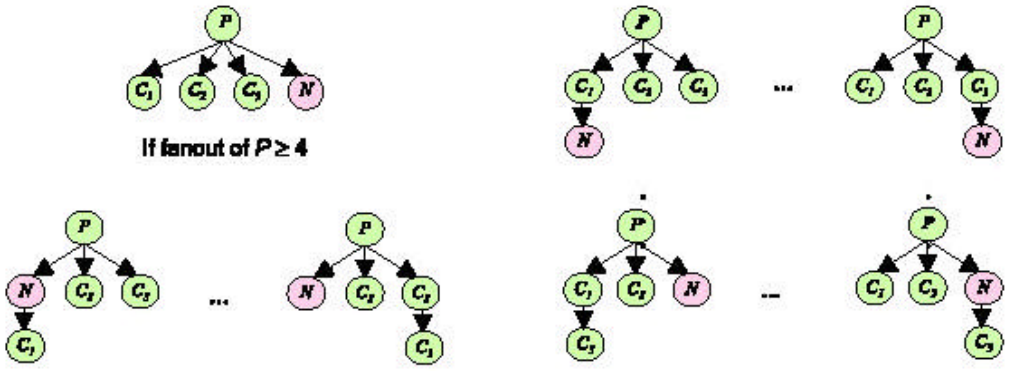
另一方面，TAG 則是利用資料來源與成員間的路由資訊來建立服務疊加網路，以達到傳送路徑的最佳化。當傳送延遲成為主要考量因素時，資料來源到服務疊加網路節點間路徑必須盡量與 IP 路由的路徑相同以達到最佳化，根據這樣的概念，TAG 在新成員加入時，資料來源必須利用 traceroute 這樣的工具來取得資料來源與新成員間路徑的路由資訊，並利用由上而下的方法往疊加樹下游方向找到路由資訊最相像的節點來當作父節點。

綜觀上述所提多媒體應用層群播技術，絕大部分的研究都屬於點對點，樹狀結構式的應用層群播系統。事實上，這也是未來應用層群播技術最具發展性的研究方向。在架構式系統中，由於必須在網路上事先架設負責資料轉送的伺服器，伺服器的數量與伺服器架設位置

就決定著整個系統的效能。然而，由於網路狀況的動態性，以及客戶端數量與客戶端加入系統時間的不確定性，要以固定數量與架設位置的伺服器達到最佳傳輸品質事實上非常困難。另一方面，網絡式系統也存在延展性不佳的問題。網絡式系統每個節點需要定期與其他全部或一定比例的節點交換控制訊息以了解其他節點的狀態，雖然系統強度上可以有最佳的表現，但在系統中節點個數逐漸增加時，網路上的控制訊息數量會急速成長，造成網路壅塞的可能性大增，對資料傳輸品質也會有不利的影響。此外，混合式架構中伺服器擁有強大的計算能力與充裕對外頻寬的假設，使得伺服器必須架設於 ISP 的機房中。如此一來，也相對降低了其應用的便利性。因此，高彈性，高延展性，架設容易的點對點樹狀結構多媒體應用層群播技術相信將是未來研究的主流。



圖二：TBCP **新成員加入步驟(引自**[13])



圖三：TBCP **疊加子樹形成之所有可能性範例(引自**[13])

由於多媒體資料通常包含大量資訊內容，頻寬就成為多媒體傳輸品質好壞的關鍵。儘管目前已有研究提出以頻寬為主要考量因素之點對點樹狀結構式應用層群播技術，但目前仍有下面兩個問題亟待解決：

1. 新節點加入疊加樹的時間過長。
2. 傳輸中斷之失敗修復成功機率過低或等待時間過長。

目前樹狀結構應用層群播技術都採用由上而下的演算法來找出新成員在疊加樹中的父節點。也就是從伺服器開始，往疊加樹下游方向找尋一個能比所有子節點提供更大頻寬的節點當做父節點。為了找到合適的父節點，新節點必須與多次節點進行頻寬量測，然而，網路上兩點間頻寬量測為了達到正確性往往十分耗時，造成節點需要等待很長的時間才能開始接收資料。另一方面，由於點對點應用層群播系統的節點成員通常以一般個人電腦為主，電腦使用者擁有很高的自主性。每個成員可能於任何時間點加入或離開疊加樹，因此不可能像一般路由器提供長期且穩定的網路服務。針對這種成員的不可預測性，點對點應用層群播系統必須有一套快速的失敗修復機制來解決傳輸中斷的問題。目前的點對點應用層群播系統主要採用下列兩種機制來進行失敗修復：

(1) 每個節點保留父節點與祖父節點的資訊。當無法從父節點接收到資料時，節點重新連線至祖父節點以避免傳輸中斷。

(2) 重新加入疊加樹。當無法從父節點接收到資料時，節點向伺服器送出傳新加入訊息，利用新成員加入疊加樹的方式由伺服器開始找尋新的父節點。

第一個方法事實上無法保證祖父節點能夠提供足夠的頻寬，可能造成失敗修復的成功率過低；第二個方法則是有等待時間過長的問題。故兩個方法都無法達到快速失敗修復的目標。因此，在我們的計畫中，如何提出一個高效率的方法能夠減少節點加入疊加樹與失敗修復的時間，以及提高失敗修復成功機率將是重點的研究項目。

(2) 點對點網路檔案系統之相關背景

隨著個人電腦計算能力以及網際網路的快速發展，如何去分享及運用網路上數以百萬計的個人電腦運算能力和其儲存之檔案已經成為目前非常熱門的研究話題。早在 1999 年由 Napster [15]帶動的檔案分享熱潮，已引起相當多的網路研究學者的注意。尤其在分散式點對點檔案系統上如何得知檔案位置的方法，所受到的注目更多，當然成果也相當多。

就以目前點對點檔案系統的架構來分類，可以分成無架構式點對點檔案系統 (Unstructured P2P File System)和架構式點對點檔案系統(Structured P2P File System)這兩種方

式。在無架構式點對點檔案系統方面較著名系統的有 Napster, KaZaA, Gnutella[16], Freenet[17]等。Napster 是以客戶端—伺服器架構，也就是節點向特定的檔案伺服器詢問來取得檔案所在位置。另一方面，Gnutella 與 Freenet 則是以廣播方式，亦即對所有系統中節點發出詢問訊息來達到檔案尋找的目的。然而，這兩種的檔案搜尋方式都會使得系統的延展性很差，造成搜尋效率隨著系統節點數的增加而急速下降。而在架構式點對點檔案系統下，所有節點被組織成一個邏輯性架構。檔案位置的搜尋則依據邏輯性架構的設計，利用不同的演算法來進行。由於檔案搜尋不需透過特定檔案伺服器或廣播方式，故架構式點對點檔案系統擁有較佳的延展性。此外，由於每個節點因為邏輯架構而擁有其他一些節點的相關資訊，因此架構式點對點檔案系統在檔案資訊搜尋速度，以及對於熱門檔案資訊查詢之負載平衡方面，也有較佳的效能表現。

為了達到高延展性與快速搜尋的目的，目前已有相當多的國外著名大學都投入架構式點對點檔案系統的研究領域。如麻省理工學院的 Chord[18,19]、AT&T 的 CAN[20]、加州柏克萊大學的 Tapestry[21-23]、史丹福大學的 YAPPERS[24,25]及微軟倫敦康橋軟體研究院的 Pastry[26,27]，在近兩年中有相當多的成果發表。而預計在往後的一、兩年將會有更多的大學和研究機構會投入相關研究領域。目前架構式點對點檔案系統，主要針對下面兩個議題進行研究：

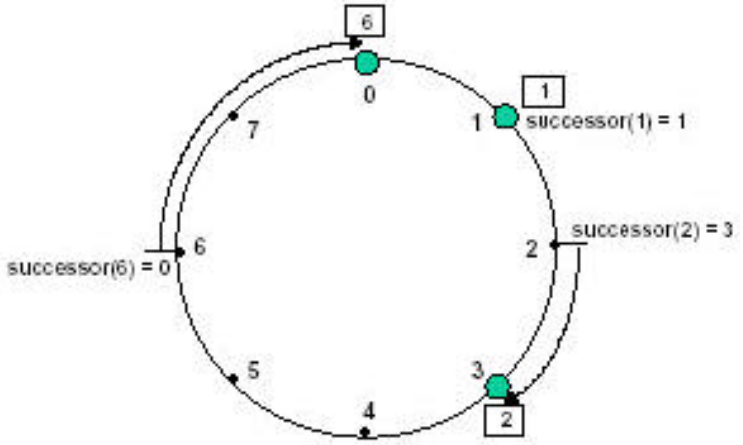
一、 邏輯架構的建立與維護。

二、 快速搜尋演算法。

Chord 提出一個延伸自 consistent hashing[28,29]，將所有節點與檔案對應到一個由 N 個整數所形成邏輯代碼環(identifier circle)的點對點檔案系統。每個節點以一個節點代碼(Node ID)來代表在代碼環中的位置，節點代碼是利用 IP 位址透過一個節點雜湊函數(Node Hash Function)計算得到。而每個檔案則利用一個物件代碼(Object ID)代表，物件代碼則為利用檔案名稱透過一個檔案雜湊函數(File Hash Function)計算得到。圖四為一包含節點 0，節點 1，以及節點 3 三個節點的架構概念圖。

當一個新檔案位置資訊加入 Chord 系統時，系統會利用檔案的物件代碼尋找在代碼環中的次節點(successor)來儲存檔案位置資訊。次節點為從物件代碼開始，沿著代碼環順時鐘方向行進所遇到的第一個節點。在圖四的例子中，物件代碼 1 的檔案資訊存在節點 1 (successor(1)=1)中；物件代碼 2 的檔案資訊存在節點 3 (successor(2)=3)中；而物件代碼 6 的檔案資訊則存在節點 0 (successor(6)=0)中。為了加快檔案搜尋速度，每個節點都會利用一個「指示表(finger table)」紀錄 successor$(n+2^{i-1})$，$1 \le i \le \log N$，這些物件代碼與節點間的對映關係，

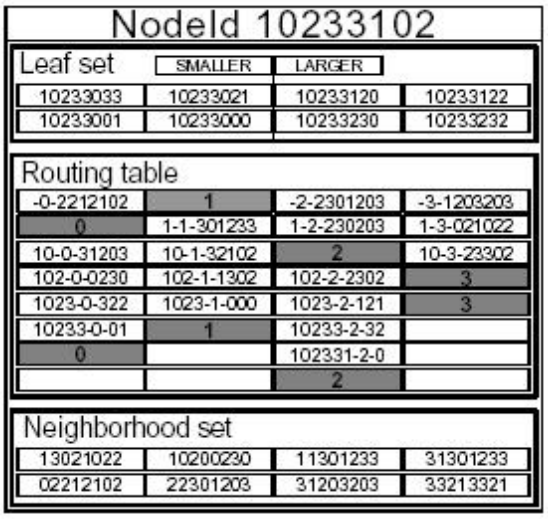
其中 n 為該節點之節點代碼。透過在各節點間指示表的查詢，在 Chord 系統搜尋檔案位置時可利用類似路由(routing)的方式進行跳躍式的搜尋, 而不需依照順時鐘方向循序的詢問所有節點。



圖四: Chord 架構概念圖(引自[18])

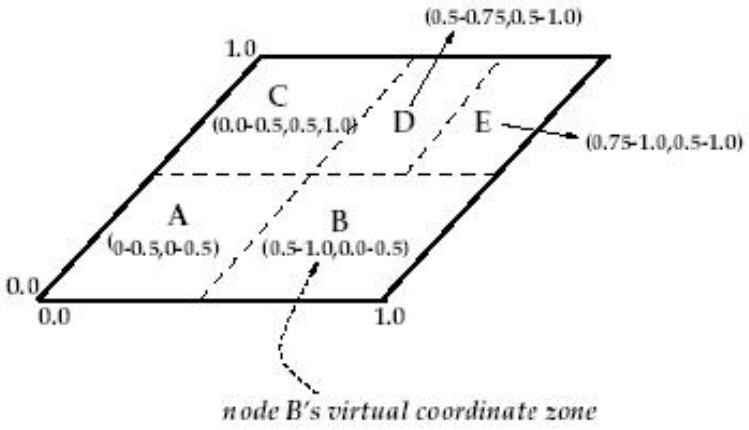
Pastry 則是另一個利用環狀邏輯架構來設計的點對點檔案系統。其同樣利用兩個雜湊函數分別利用節點的 IP 和檔案的名稱轉換成 128-bit 之節點代碼與物件代碼, 並利用一個「路由表(routing table)」來儲存一些節點代碼與物件代碼間的對應關係。為了達到快速搜尋, Pastry 中每個節點 *N* 會將路由表中所記錄的節點分為 Leaf set，Routing table，及 Neighborhood set 三個部份。其中 Leaf set 記錄節點代碼與 *N* 之節點代碼「相鄰」的節點；Neighborhood set 則記錄與 *N* 之網路距離相近的節點；Routing table 則記錄與 *N* 之節點代碼數個前置碼(prefix)相同的節點。圖五為一節點代碼為 10233102 的節點其路由表中的內容範例。Pastry 在做檔案搜尋時, 會先判斷所要尋找的 Peer 是否在 Leaf set 或 Neighborhood set 中, 若有即可直接尋找到, 若沒有時在從 Routing table 以比對前置碼的方式尋找最接近的節點，並從此節點繼續搜尋下去。

相對於 Chord 與 Pastry 使用環狀架構， CAN 則是利用多維座標空間的概念來建構檔案系統。觀念上，每一個節點擁有多維座標空間中的一部份空間。圖六為一個利用二維座標系統, 包含五個節點的 CAN 系統架構概念圖. 當一個新檔案位置資訊加入 CAN 系統時, CAN 系統會利用檔案名稱依照多個事先定義的雜湊函數計算出一個座標，並將資訊儲存在包含此座標的空間之節點中。另一方面，當一個新節點 *N* 欲加入 CAN 系統時，節點 *N* 透過一個起始點(Bootstrap)隨機選擇一個系統中的節點 *P*，並送出 JOIN 訊息給 *P*。當收到 *N* 的 JOIN 訊息後，*P* 會與 *N* 均分其所擁有的座標空間，並將應該儲存在 *N* 的檔案位置資訊傳送給 *N*。



圖五: Pastry **節點之路由表內容** (**引自**[27])

在檔案資訊搜尋技術方面，CAN **則是利用類似網路路由的方式來找尋檔案資訊所在的節點。在** CAN **系統中，每個節點利用一個座標路由表**(coordinate routing table)**記錄在座標空間中相鄰節點的** IP **位址資訊及其所擁有的空間資訊。當收到一個檔案資訊查詢的要求訊息時，起始點會先利用雜湊函數計算出此檔案所代表的座標，並將此座標資訊記錄在查詢要求訊息中接著轉送給隨機選擇的一個節點。當收到此座標資訊訊息時，節點首先查詢檔案資訊是否存在，若存在則回報此檔案資訊。否則，此節點會依照座標資訊利用座標路由表以貪婪演算法**(greedy algorithm)**找出一個與檔案所代表座標最相近的一個節點並轉送此查詢訊息。如此遞迴下去直到找到檔案資訊為止。**

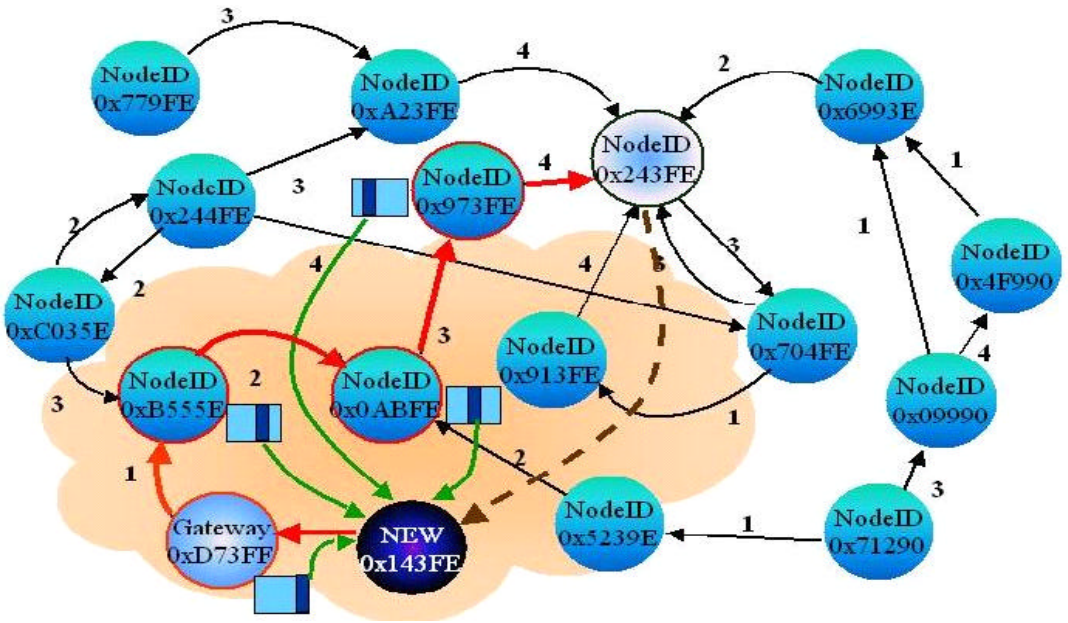


圖六: CAN **架構概念圖(引自**[20])

Tapestry **是由加州柏克萊大學所發展，延伸自** OceanStore[30]**的一個點對點檔案系統。其檔案系統架構是利用節點代碼之間字尾比對**(Suffix matching)**之結果所形成的一個網絡。節**

點代碼是使用節點雜湊函數與節點 IP 計算得到的，而這個節點代碼是由多個 16 進位數字組成，如 0xF1290。而每一個檔案也會有一個物件代碼(Object ID)。此物件代碼是由物件雜湊函數和檔案名稱計算取得的，同樣的其物件代碼也是由多個 16 進位的數字組成。每一個檔案的位址資訊則會被儲存在與其物件代碼和節點代碼有最相近的字尾比對的節點上。此外每一個節點會維護一個路由表, 紀錄著本身的節點代碼和其他節點代碼中最相近的字尾比對的節點。

在新節點加入方面，新節點利用本身節點代碼與鄰近節點的路由表中記錄的節點代碼做字尾比對路由，進而找到與其節點代碼最相近的節點，並依照自己的節點代碼跟此節點取得檔案位置資訊。新節點要加入時會先跟系統一個預設的匝道節點(Gateway node)取得其路由表，並利用本身節點代碼與路由表中節點代碼做字尾比對，依照比對的結果找到與一個節點代碼最相近的節點，再利用找到節點的路由資訊做字尾比對，遞迴往與本身節點代碼相近的節點逼近，直到找到最相近的節點代碼的節點後，由此節點取得其所負責管理的檔案位址資訊。圖七為加入一個節點代碼為 0x143FE 的新節點到 Tapestry 系統中，節點 0x143FE 會先找到匝道節點後, 取得其路由表後, 得知與其第五個數字, "E", 相同節點代碼的節點, 0xB555E。之後再由 0xB555E 的路由表得知與其第四個數字，"F"，相同的節點 0x0ABFE，一直找到與其節點代碼最相近字尾比對的節點 0x243FE，並從節點 0x243FE 取得檔案代碼為 0x043FE 到 0x143FE 的檔案位址資訊後即完成加入動作。



圖七: Tapestry 加入新 Node(引自 Ben Y. Zhao 在 Berkeley 的演講稿)

在檔案找尋方面，Tapestry 是利用檔案代碼與節點所紀錄的路由表中的節點代碼作字尾比對路由，進而找到檔案位址資訊。發出檔案位址查詢要求的節點會將檔案名稱和檔案雜

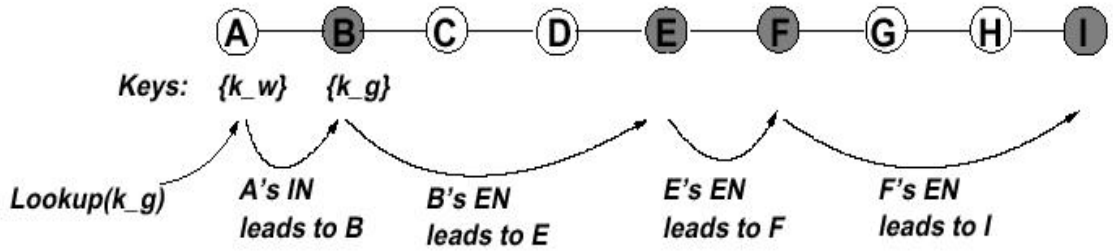
湊函數加以計算得知此檔案的檔案代碼，並依照本身擁有的路由表，利用剛剛描述的新節點加入方式尋找與檔案代碼最相近字尾比對的節點代碼的節點，即可取得檔案的位址資訊。

YAPPERS 是由史丹福大學在 2002 年底所提出的一個架構在 Gnutella 系統的分散式點對點檔案系統。與上述的系統最大的差別為，YAPPERS 混合了非架構式檔案系統與架構式檔案系統的方法，同時利用的分散雜湊函數和廣播技術達成檔案找尋的問題。

YAPPERS先利用一組特殊的分散雜湊函數將檔案名稱和節點IP分成多個盒子(Buckets), 例如是白或灰的顏色的兩個盒子，若有一屬於白色盒子節點想要新增檔案資訊，先利用分散雜湊函數的知其屬於哪一個顏色的盒子，如屬於白色盒子的檔案就將檔案資訊儲存在本身，若屬於灰色盒子時則將檔案資訊儲存在其知道的灰色盒子節點上，若無鄰近的屬於灰色盒子節點時，就以廣播的方式找尋。

爾後一個節點要尋找檔案時，先在利用雜湊函數取得檔案是屬於哪一個盒子，之後詢問自己附近的屬於該盒子的節點，若無屬於該盒子節點時，則利用廣播的方式向所有附近節點詢問是否知道屬於該盒子節點，一直詢問到找到檔案為止。也就是說YAPPERS是跳躍式的廣播方式，這比Gnutella方式有著更好的效能，而且YAPPERS系統容錯能力也相當不錯。但是YAPPERS相當於Gnutella的改進，所以當系統規模很大時，其效能還是有待商確的。

圖八為 YAPPERS 在做檔案找尋機制的概念圖，我們可以看到當白色節點 A 發出一個屬於灰色盒子的檔案找尋請求時，先與其鄰近的灰色節點 B 詢問，若節點 B 沒有此檔案時，B 點會跳過節點 C、節點 D，向節點 E 做詢問的動作，一直到節點 F 和節點 I，都是詢問屬於灰色盒子的節點。此方式大大減少了廣播訊息，並保留了 Gnutella 原先在系統強韌度上的表現。



圖八 YAPPERS 基本架構

綜觀上述所提點對點檔案系統，在延展性與詢問節點個數都有良好的表現，然而，尋找檔案位置資訊等待時間過長及系統容錯效能上的表現較差為點對點檔案系統急待解決的兩個問題。針對尋找檔案位置資訊等待時間過長的問題方面，俱備位置知覺的點對點檔案系統可以減少大幅度的傳輸延遲時間，並加快檔案搜尋時間。當一個節點在發出檔案尋找要求時，在沒有位置知覺的點對點檔案系統中，節點並無法依照實體網路架構詢問與其相鄰的節點，

而必須依照系統所維護的邏輯架構依序詢問，使得尋找檔案位置資訊的效率無法達到最佳化。

而在系統容錯效能表現較差的問題上，具備節點叢聚性的檔案系統可以加強系統在強韌度的表現。當系統在發生節點突然離線時，其鄰近節點可以快速得知其離線，並盡快的修復失敗節點所造成的錯誤。

最近才提出的 YAPPERS 並沒有相當大幅度的改善點對點檔案系統的效能，不過他是目前第一個真正做到位置知覺路由的點對點檔案系統。YAPPERS 會先在自己鄰近節點尋找檔案，若找不到時再擴展至較遠的節點，最後才擴散至整個點對點檔案系統。不過 YAPPERS 基本上還是屬於廣播的方式在找尋檔案，且為非架構式點對點檔案系統，這使得 YAPPERS 在系統延展性上有相當大的挑戰。

要將位置知覺的特性加入以分散式雜湊函數為基礎的架構式點對點檔案系統，有相當大的問題存在。而又 YAPPERS 這類非架構式點對點檔案系統在延展度的表現並不好，因此在我們的計畫中，我們將注重於如何建構一個俱有位置知覺特性的架構式點對點檔案系統為核心，並以減少檔案尋找時間及加強系統強韌度為主要的研究項目。

2、 研究計畫之目的

本計畫的目的是將結合新的點對點應用層群播以及點對點檔案系統技術，由寬頻網路延伸至無線網路，建置一個高延展性，高傳輸效能的多媒體傳輸網路平台，並在此一多媒體傳輸網路平台上進行進階的多媒體傳輸相關議題的研究。我們希望以新的研究成果為基礎，打造新的多媒體傳輸與資訊分享系統。

本計畫預計以三年為期，第一年專注在發展點對點實況廣播 (P2P live broadcasting)技術、具位置知覺之點對點檔案系統(Location-Aware P2P file system)技術、有效量測技術分析與探討以及疊層服務網路(Service Overlay Network, SON)之服務品質提供等技術進行深入的理論探討，以期建立後面研究的理論根基。第二年我們將利用第一年的成果，包括點對點實況廣播技術、BGP 路由表資料等，進一步延伸到兩個新的主題上: 點對點視訊會議系統及以 Internet topology 為基礎的具網路知覺(Network-Aware)之點對點檔案系統。在點對點視訊會議系統的主題上，我們希望將第一年單一來源(single source)的點對點實況廣播成果延伸到多點來源(multiple sources)的即時視訊會議系統，也就是以 P2P 為架構的即時會議系統。與其他研究不同的是，此一系統將結合 unicast、multicast、location aware、bandwidth measurement、failure auto-recovery 等技術，實作出一高效率、高彈性、穩定的視訊會議系統。其次我們將利用 SON 的研究中，對於 Internet measurement 的成果及從 BGP routing table 所得之資料，進一步來發

掘 Internet topology，並以有效率的方式來提供此一資料來發展另一新的具網路知覺之點對點檔案系統。本計畫的第三年我們將以上之研究擴展到無線網路的環境。目前 P2P 的架構在無線網路的研究相當少，但在未來個人行動通訊將成為主流，利用 PDA、Tablet PC、Notebook 以 WLAN、Bluetooth、3G 等網路上網將愈來愈普及的時代，我們覺得 P2P 也將成為這些行動數位載具的一個新的 killer application，所以在第三年，我們將研究在無線網路(特別是 ad hoc 架構)的點對點資訊分享系統，並實作由不同載具、介接網路所形成的網路上的資訊分享系統。

3、 研究計畫之重要性

隨著 2001 年一月十八日美國線上(American On Line)與時代華納(Time Warner)宣布併購完成後，等於宣告了未來媒體，娛樂，傳播，以及網路產業間的整合將成為不可抵擋的潮流。在媒體內容數位化的趨勢發展下，高延展性，高傳輸品質的多媒體傳輸技術勢必成為未來網際網路技術發展的主流。當傳統客戶端—伺服器架構，以及 IP 群播已經無法提供多媒體系統所需要的傳輸品質時，如何利用現代個人電腦越來越強大的計算及儲存能力進行資料分享將是未來非常重要的議題。從近年來 P2P 的流量漸增，Web 流量相對漸減，許多人相信 P2P 架構將是下一波 killer application 的重要技術。其實如果不是著作權的問題，Napster、KaZaA 都是相當成功的例子。(當然我們還是相信所有技術都必須在正義公理、合法的前提下發展。)

4、 國內外有關本計畫之研究情況

不管在新的應用層群播技術與相關的點對點檔案系統，目前都是國內外相當重要的研究領域。如之前所述，目前相關研究主要針對應用層群播疊加網路的建立與維護，以及點對點檔案系統中邏輯架構建立，以及快速搜尋演算法等方面進行研究。但是，如何因應多媒體傳輸頻寬需求，快速建立起疊加網路，以及在點對點檔案系統中如何利用叢聚技術發展出更快速的檔案搜尋技術方面的研究則是較新的研究方向。在國內相關研究方面，台大、台科大、清華、元智等學校均有教授參與相關的研究，而在國外，從 Napster 到 KaZaA，相關的研究與應用更是不勝枚舉。此計畫的主要貢獻在於，將階層式叢聚技術應用在應用層群播技術以及點對點檔案系統中，以快速建立符合多媒體傳輸頻寬需求的疊加網路，並減少點對點檔案系統中的資料搜尋時間，進而發展出新一代具有高延展性、高效能、跨網路平台的多媒體傳輸與資訊分享系統。

5、 參考文獻

[1] Y. Chawathe, S. Mccanne, and E. Brewer, "An architecture for Internet Content distribution as an infrastructure service," http://www.cs.berkeley.edu/~yatin/papers/scattercast.ps

[2] Y. H. Chu, S. Rao, and H. Zhang, "A Case for End System Multicast," ACM SIGMETRIC 2000, pp. 1-12, June 2000.

[3] D. Pendarakis, S. Shi, D. Verma, and M. Waldvogel, "ALMI: An Application Level Multicast Infrastructure," 3rd USENIX Symposium on Internet Technologies, March 2001.

[4] Microsoft Windows Media Technologies, "http://www.microsoft.com/windows/windowsmedia/EN/default.asp"

[5] Real Player, "http://www.realnetworks.com/"

[6] Apple Quicktime, "http://www.apple.com/quicktime/authoring/"

[7] J. Jannotti, D. Gifford, K. Johnson, F. Kaashoek, and J. O'Toole, "Overcast: Reliable Multicasting with an Overlay Network," USENIX OSDI 2000, October 2000.

[8] Y. Guo, K. Suh, J. Jurose, and D. Towsley, "P2Cast: P2P Patching Scheme for VoD Service," submitted to IEEE INFOCOM 2003.

[9] V. N. Padmanahan, H. J. Wang, and P. A. Chou, "Distributed Streaming Media Content Using Cooperative Networking," Proceeding of NOSSDAV'02, May 2002.

[10] Allcast, "http://www.allcast.com"

[11] vTrails, "http://www.vtrails.com"

[12] Bluefalcon, "http://www.bluefalcon.com"

[13] L. Mathy, R. Canonico, and D. Hutchison, "An Overlay Tree Building Control Protocol," Proceeding of International Workshop on Networked Group Communication, November 2001.

[14] M. Kwon and S. Fahmy, "Topology-aware Overlay Networks for Group Communication," Proceeding of NOSSDAV'02.

[15] Napster. http://www.napster.com/

[16] Guatella. http://gnutella.wego.com/.

[17] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong, "Freenet: A distributed anonymous information storage and retrieval system," In the proceedings of the ICSI Workshop on Design issues in anonymity and unobservability , Berkeley,CA, June 2000. http://freenet.sourceforge.net

[18] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in Proceedings of ACM

SIGCOMM, San Diego, August 2001, pp. 160–177.

[19] F. Dabek, E. Brunskill, M. Frans Kaashoek, D. Karger, R Morris, I. Stoica, and H. Balakrishnan., "Building Peer-to-Peer Systems With Chord, a Distributed Lookup Service," 8th Workshop on Hot Topics in Operating Systems, Germany, May 2001.

[20] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," in Proceedings of ACM SIGCOMM, San Diego, August 2001, pp. 149–160.

[21] B. Y. Zhao, J. Kubiatowicz and A. Joseph, "Tapestry: An Infrastructure for Fault-tolerant Wide-area Location and Routing," UCB Tech. Report UCB/CSD-01-1141.

[22] B. Y. Zhao, Y. Duan, L. Huang, A. D. Joseph and J. D. Kubiatowicz, "Brocade: landmark routing on overlay networks," First International Workshop on Peer-to-Peer Systems (IPTPS) Cambridge, MA. March 2002.

[23] B. Y. Zhao, A. D. Joseph, and J. D. Kubiatowicz, "Locality-aware Mechanisms for Large-scale Networks, " Workshop on Future Directions in Distributed Computing Bertinoro, Italy. June 2002.

[24] P. Ganesan, Q. Sun, and H. Garcia-Molina, "YAPPERS: A Peer-to-Peer Lookup Service Over Arbitrary Topology," Infocom, 2003.

[25] B. Yang and H. Garcia-Molina, "Improving Search in Peer-to-Peer Systems," *ICDCS*, 2002.

[26] P. Druschel and A. Rowstron, "PAST: A large-scale, persistent peer-to-peer storage utility," HotOS VIII, Schoss Elmau, Germany, May 2001.

[27] A. Rowstron and P. Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems, " IFIP/ACM International Conference on Distributed Systems Platforms , Heidelberg, Germany, pages 329-350, November 2001.

[28] D. Karger, E. Lehman, F. Leighton, M. Levine, D. Lewin, and R. panigrahy, "Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the World Wide Web," in the proceedings of ACM symposium on Theory of Computing, 1997.

[29] D. lewin, "Consistent hashing and random trees: Algorithms for caching in distributed networks," Master's thesis, Department of EECS, MIT, 1998. Available at the MIT Library, http://thesis,mit.edu/

[30] J. Kubiatowicz, D. Bindel, Y. Chen, P. Eaton, D. Geels, R. Gumadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells and B. Y. Zhao., "OceanStore: An Architecture for Global-scale Persistent Storage," Proceedings of ACM ASPLOS, November, 2000.

[31] M. Jain and C. Dovrolis, "End-to-end available bandwidth: measurement methodology, dynamics, and relation with tcp throughput," Proceedings of ACM SIGCOMM, August 2002.

[32] Zhenhai Duan, Zhi-Li Zhang, and Yiwei Thomas Hou, "Service Overlay Networks: SLAs, QoS and Bandwidth Provision," submitted for publication.

[33] http://pma.nlanr.net/traces/long/auck2.html.

[34] L. Rabiner, "A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," Proceeding of the IEEE, vol. 77, pp.257-285, February 1989.

[35] S. Fine, Y. Singer, N. Tishby, "The Hierarchical Hidden Markov Model: Analysis and Applications," Machine Learning, vol. 32, pp. 41-62, 1998.

[36] D. G. Anderson, N. Feamster, S. Bauer, H. Balakrishnan, "Topology Inference from BGP Routing Dynamics," ACM SIGCOMM Internet Measurement Workshop, 2002.

[37] University of Orgeon, "RouteViews," http://www.routeviews.org/.

[38] T. Bu, L. Gao, D. Towsley, "On Characterizing BGP Routing Table Growth," IEEE GLOBECOM 2002.

[39] L. Gao and F. Wang, "The Extent of AS Path Inflation by Routing Policies," IEEE GLOBECOM 2002.

（二）研究方法、進行步驟及執行進度

本計畫預計以三年的時間，進行多媒體應用層群播的研究，包括點對點應用層即時群播技術與點對點資訊分享系統兩領域的研究。如上所述，在這三年中，我們預計第一年專注在發展點對點實況廣播技術、具位置知覺之點對點檔案系統技術以及疊層服務網路之服務品質提供等之基礎理論。第二年發展以 Internet BGP **路由為基礎的另一個具網路知覺之點對點檔案系統技術，以及實作點對點架構上之視訊會議系統。第三年則研究並實作在無線** ad hoc **網路下的點對點資訊分享系統。以下就這三年的研究工作的方法與步驟分年敘述。**

第一年:

這一年共有以下四項分項工作：點對點實況廣播技術設計，具位置知覺之點對點檔案系統技術設計，有效量測技術研究，以及如何提供疊層服務網路之服務品質之研究。

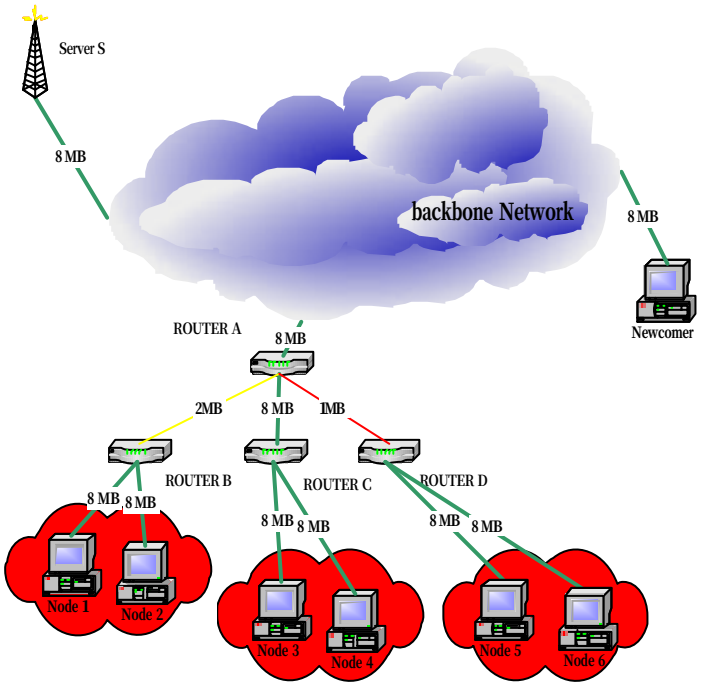
1、本計畫採用之研究方法與原因

點對點實況廣播技術:

為了提供最即時的訊息，許多電子媒體紛紛採用實況轉播將第一手的資訊以最快速的方式送到觀眾面前。隨著串流技術的快速發展，相信不久的將來媒體業者不再需

要負擔昂貴的衛星通訊費用，而可以利用相對便宜的網際網路來傳送即時資訊。因應這未來的發展趨勢，我們希望能夠發展出一套點對點實況轉播系統，使得媒體業者能夠迅速將最新的訊息傳送給閱聽大眾。由於實況轉播的資訊具有即時性與連續性，因此整個點對點實況轉播技術發展的重要關鍵，就在於能否解決之前所提新節點加入疊加樹時間過長，等待修復傳送中斷時間過長，以及傳輸中斷之失敗修復成功機率過低等問題。本項研究的目標是設計出一可以節省頻寬及具有高自我修復能力的點對點視訊實況轉播架構，並以模擬分析其最終之效能是否達成目標。

針對原有的點對點架構中，新節點加入時間過長及自我修復不易的問題，我們希望以節點叢聚技術加以解決。如之前所述，目前樹狀結構應用層群播技術都採用由上而下的演算法來找出新成員在疊加樹中的父節點。也就是從伺服器開始，往疊加樹下游方向找尋一個能比所有子節點提供更大頻寬的節點當做父節點。為了找到合適的父節點，新節點必須與多次節點進行頻寬量測，然而，網路上兩點間頻寬量測為了達到正確性往往十分耗時，造成節點需要等待很長的時間才能開始接收資料。事實上，如果能夠將提供頻寬相似的節點叢聚起來，新節點只需要與每個叢聚中的一個節點進行頻寬量測即可，如此就可以節省新節點加入的等待時間，我們利用圖九說明這樣的概念。



圖九: 節點叢聚概念圖

在圖九的例子中，一個新成員(Newcomer)欲加入如圖九所示包含一個伺服器 S 和六個節點(Node1 到 Node6)的群播群組。假設骨幹網路(backbone network)的頻寬大於 8MB，六個節點皆為伺服器 S 的子節點(即直接由伺服器提供服務)。在這個情況下，我們觀察 Newcomer 與這六個節點的關係，我們會發現 Node1 與 Node2、Node3 與 Node4、Node5 與 Node6 分別屬於三個 access network，很自然地分成了三個 cluster，而由 Newcomer 到這三個 cluster 的 bottleneck 很可能分別是 router B、C、D 到 router A 的 link 上。我們假設說 Node1 與 Node2 所能提供給 Newcomer 的頻寬都是 2MB；Node3 與 Node4 所能提供給 Newcomer 的頻寬都是 8MB；Node5 與 Node6 所能提供給 Newcomer 的頻寬都是 1MB。那麼，我們希望發展一項節點叢聚技術，將 Node1 與 Node2，Node3 與 Node4，以及 Node5 與 Node6 分別形成三個群組。Newcomer 只需要伺服器 S，以及從三個群組中各選其中的任何一個節點進行頻寬量測即可，而不需要與六個子節點進行頻寬量測，如此就可縮短新節點加入系統的時間。

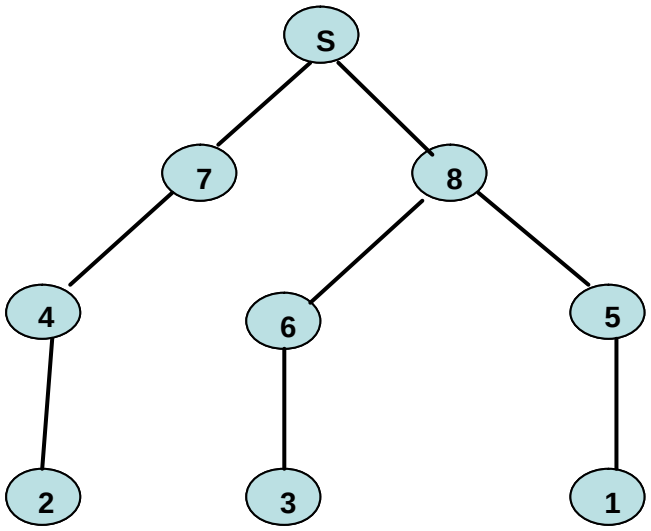
另一方面，這樣的叢聚技術也可提高失敗修復成功率與縮短失敗修復時間。在新節點找到頻寬最大的節點作為父節點後，若新節點能夠得到此群組其他成員的資訊，當因為傳輸中斷需要進行失敗修復時，由於群組中節點都能提供比其他群組中的節點更多的頻寬，因此只需找尋群組中其他節點當做替代的父節點即可。如此一來，既可以提高失敗修復成功率，又可縮短失敗修復時間。

因此，在整個點對點實況廣播系統的設計上，我們的重點工作在於提出一個基於所能提供新節點之頻寬為基礎，將系統中節點適當分群的叢聚技術。但要測量頻兩點之間的頻寬是相當費時的，例如[31]所提出的方法。所以在這部份我們初步的想法是利用 delay 的量測來做為頻寬大小的預估根據。為了初步了解 delay 與頻寬間是否有關聯性存在，我們利用在中正大學資工所內的一台機器，與中正大學計算機中心，國家高速電腦中心，成功大學，美國加州大學柏克萊分校，以及美國加州大學洛杉磯分校中的機器進行頻寬與 delay 的量測，其結果如表二。由表二可以看出對中正大學資工所的機器而言，delay 相近的機器所量測出來的頻寬也都非常接近。因此，我們想藉著更多的量測來揭發 delay 與 bandwidth 的關係，並以此做為快速預估頻寬及建立適當的叢聚及其階層(hierarchy)的依據。

表二：delay **與頻寬之量測結果**

	delay	**頻寬**
中正大學計算機中心	<10ms	2Mbps – 3Mpbs
國家高速電腦中心	30 – 50 ms	800 – 890 kbps
成功大學	30 – 50 ms	800 – 890 kbps
美國加州大學柏克萊分校	200 – 250ms	100 – 300kbps
美國加州大學洛杉磯分校	200 – 250ms	100 – 200kbps

另外在建立階層式的叢聚時，我們需要考量讓一個新節點加入系統時，能夠盡快接收到即時的媒體內容，並且減少因為系統中節點停止提供服務或無預警故障所產生傳輸中斷的影響，所以我們認為資料傳送樹(data distribution tree)**必須是屬於所謂的寬短樹**(short-and-wide tree)**。要形成寬短樹的架構，我們初步的想法是在每個節點可以提供不同個數連線的情況下，能夠提供較多連線的節點，應該距離資料傳送樹的資料來源越近。圖十為一包含一個資料來源** S**，與八個能夠提供不同個數連線的節點，所形成的一個資料傳送樹。圓圈內的數字代表該節點所能夠提供的最大連線個數。**



圖十: **短寬樹概念圖**

在此項研究中，我們將提出一個架構及其協定，其中包括以下之運作:

(1) **新成員加入。當一個新成員加入系統時，主要包括兩個步驟：**

(a) **叢聚搜尋：利用與各叢聚中節點的** delay **來預估與叢聚中節點的頻寬，藉此決定所加入的叢聚。**

(b) **父節點搜尋：利用所設計的演算法來找尋在所加入的叢聚中，接收資料的**

節點，也就是在資料傳送樹中的父節點，並符合寬短樹的原則。

(2) 成員離開。當一成員離開時，其子節點可以迅速的找到替代父節點。

(3) 斷線修復。當一成員無預警斷線時，其子節點可以迅速的找到替代父節點，以繼續收到 streaming 的資料。

效能評估:

最後，為了證明我們設計的點對點實況廣播系統能夠達到預期的設計目標，我們在系統實作之前先利用現有的網路模擬程式，如 Network Simulator 2 (ns2)，設計一系列實驗來驗證系統效能。在點對點實況廣播系統方面，我們將與其他同類型的點對點多媒體系統做下列因素的比較：

1. 新節點加入系統成功率。
2. 新節點加入系統所需時間。
3. 失敗修復成功率。
4. 失敗修復所需時間。

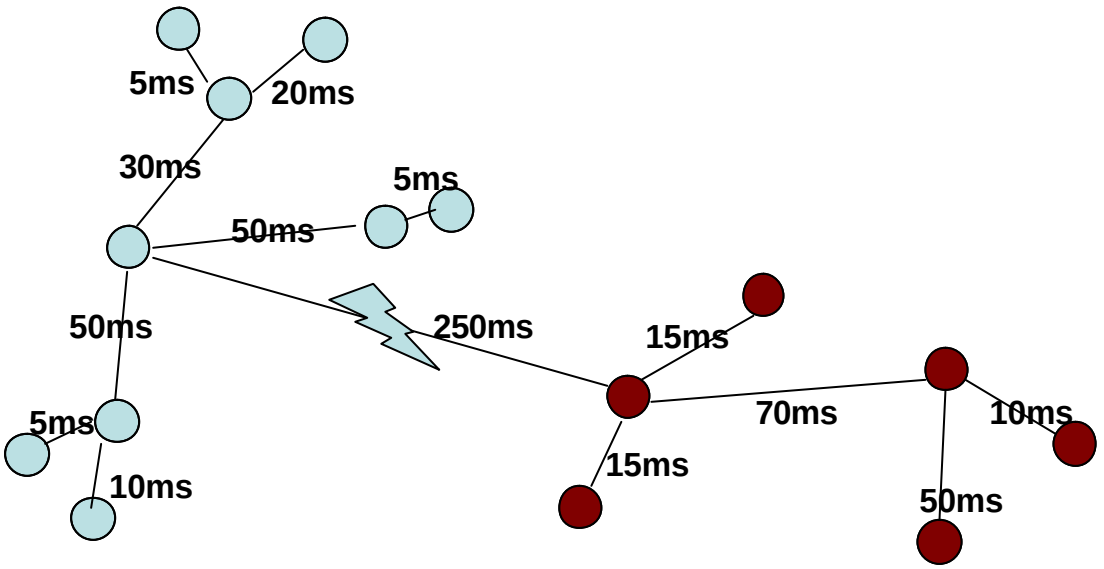
為了證明我們設計的點對點實況廣播系統具有高延展性，我們將進行新節點加入系統成功率在我們的系統與其他點對點多媒體系統之比較。新節點加入系統成功率越高，代表系統在傳輸品質不變的情況下可以容納越多的節點加入，也代表具有較高的延展性。另一方面，透過新節點加入系統所需時間的比較，可以驗證我們的系統是否能夠以較短的時間開始接收媒體資料，以達到實況廣播系統即時性的要求。另外，透過失敗修復成功率的比較，將可驗證我們的系統在系統強度方面的表現是否比其他多媒體系統有較佳的表現。最後，透過失敗修復所需時間的比較，則可以驗證我們所提的叢聚技術是否降低傳輸中斷的機率。

具位置知覺之點對點檔案系統:

因為電腦及網際網路技術的發展，人們可以很容易在全球資訊網(WWW)中找到想要找到的知識。進一步地，隨著網路頻寬越來越大，使得人們開始對多媒體資料傳輸及檔案分享的需求也越來越多。而要完成檔案分享的功能，產生了很多需要研究解決的問題，如重複儲存檔案、尋找最近的檔案提供者、匿名使用者、找尋檔案位置、認證檔案及使用者、檔案命名等問題。然而其中最重要的核心問題是如何有效而快速的找尋檔案所被儲存的位置。目前已有很多國外知名大學提出這個問題的解決方法，

其系統在延展性和搜尋檔案的效能上都有不錯的表現，但是在強韌性上的表現都還是不足。另一方面目前所發表的系統幾乎都是利用分散式雜湊函式形成一個邏輯性的架構，並要求節點要維護此架構的一致性。但分散式雜湊函式先天的缺點為無法讓節點知道自己在此檔案系統中所在的位置，進而詢問與自己較近的節點，減少網路傳輸延遲。而由上一項研究議題的分析中，我們也可以知道網路傳輸延遲和可用頻寬有一定的關聯性，所以不具位置知覺的邏輯性架構，往往造成不在位置鄰近、具高頻寬的節點上取得資料，反而在邏輯性架構上鄰近但真實位置相距甚遠、頻寬甚低的節點上去取資料。

為了讓節點具有位置知覺性質，我們希望以節點叢聚技術來達到目的。目前提出的系統大部份並未考慮到節點叢聚性對系統效能的影響，或是試圖想使節點有此特性但因為目前無特殊的雜湊函式提供此功能。而在我們提出的系統中，我們希望能利用節點與節點之間的傳輸延遲將節點劃分為幾個叢聚的區域，進一步利用此劃分出來的區域，使節點在尋找檔案位置資訊時能在其區域內找到，不必橫越兩個區域。圖十一為利用節點與節點之間的傳輸延遲形成的兩個黑色和灰色的區域。我們可以看到在黑色的區域中任何節點與在黑色區域中的節點傳輸延遲都會比其與灰色區域中的任何節點小。而在檔案搜尋期間若節點只會對其所在的區域中節點發出檔案位置詢問要求，將會大大減少檔案查詢時間，因為發出要求的節點和回應節點皆在同一區域中。



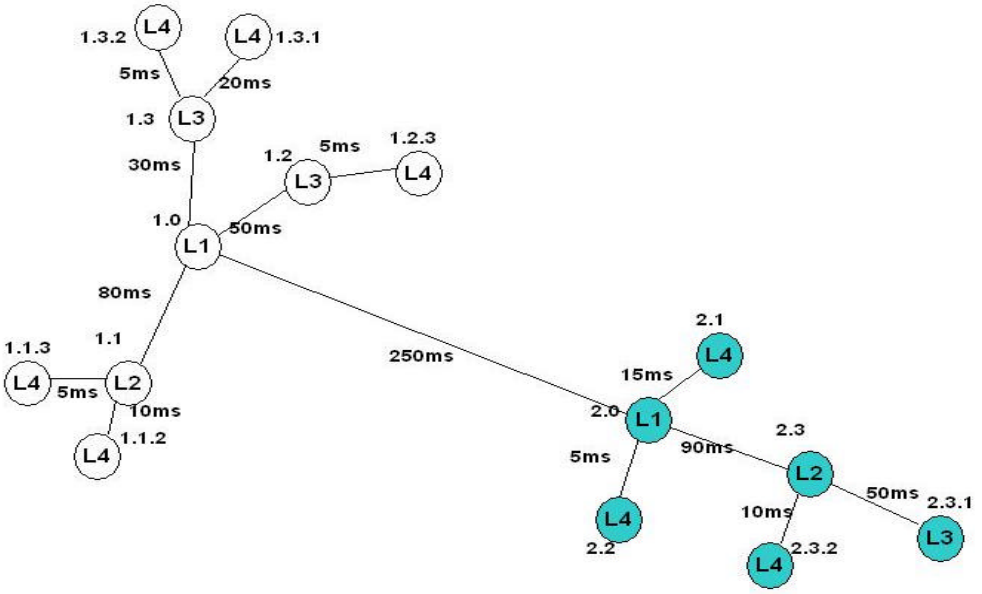
圖十一: 節點叢聚概念圖

節點叢聚性對系統的強韌性也有一定的優勢。當有節點加入、離開或突然的斷線離開時，所需要的控制訊息傳遞只會在其區域中傳遞，並不會影響到其他區域的運作。

所以當一個區域內發生節點頻繁加入或離開時，其他的區域的節點還是可以正常的查詢檔案位置資訊，並不會受到干擾。

在分散式點對點檔案系統的設計上，我們將會提出利用傳輸延遲為叢聚依據的具位置知覺之點對點檔案系統。使得節點在查詢檔案位置資訊時能在其區域內快速詢問到所要求的資訊，並提高系統的強韌度，減少節點在查詢檔案位置資訊時所等待的時間。我們預計當完成的工作包括：

(1) 以延遲為依據，建立多層次具位置知覺的階層式叢聚架構。初步的想法是利用節點加入時測量出的延遲資訊來讓各個在系統中的節點能找到並加入與其最臨近的節點，而此新節點會依照與最鄰近節點延遲資訊得到一個標籤(位置編碼)。使我們可以利用此標籤來定義階層式叢聚架構。我們稱之為具位置知覺的階層式叢聚架構，因為每一個節點可以經由自己的標籤很容易知道自己在系統中的相對位置。圖十二表現了將圖十一中的節點，利用延遲資訊來區分其階層式的叢聚架構。我們先將每一節點與其最鄰近節點的延遲分成幾個區段，例如延遲為 0ms~25ms 為區段四(L4)、25ms~75ms 為區段三(L3)、75ms~175ms 為區段二(L2)、175ms 以上為區段一(L1)。



圖十二: 階層式叢聚架構圖

若新節點要加入此系統時，會先找出與其有最小延遲的節點，並依照區分的延遲區段找到其所屬的階層。而此階層有從屬關係，區段一節點可有區段一、區段二、區段三和區段四等子節點，區段二節點可有區段三和區段四

等子節點，區段三節點可有區段三和區段四等子節點，而區段四只能有區段四的子節點。在加入的過程中，我們會給每一個節點一個位置編碼。編碼的規則為初始節點為 1.0，而區段一節點為 X.0，X 為初始節點所擁有的區段一節點個數加一，區段二、三、四節點則為其父節點的位置編碼加上和其兄弟節點個數加一後所合成的邊碼。如圖十一中位置編碼為 2.3.2 節點其位置編碼是由父節點的 2.3 和其兄弟節點個數加一後所合成的。

(2) 檔案之分佈與儲存及雜湊函式之設計。我們將設計一雜湊函式來將檔案名稱轉換成一特殊標籤其格式如上述節點的位置編碼，再依照此特殊標籤將檔案對應到與節點位置編碼相對的階層式叢聚的節點上，並設計出新節點加入時，如何取得並儲存該負責儲存的檔案。我們簡單描述將檔案位置資訊分散至各個節點的運作過程，系統一開始假設初始節點 1.0 擁有所有的檔案位置資訊，當有區段一節點加入時，新加入的節點會將所有的檔案位置資訊下載下來。而區段二、三、四節點加入時，每一節點會依據其本身的位置邊碼下載儲存其應該保管的檔案位置資訊。由於我們並不知道整個樹的完整架構，所以有可能有些 logic addresses 是沒有對應的節點存在。我們預計的做法是，當有一檔案位置資訊所對應的節點未加入時，則此檔案位置資訊會被儲存在對應節點的父節點上，也就是若今有一檔案位置資訊應該被儲存於位置邊碼為 1.2.3.4 的節點上，但此節點未出現在系統中，則其父節點 1.2.3 必須幫忙儲存其檔案位置資訊，等到此節點加入時再轉由此節點儲存。

(3) 檔案找尋(routing)演算法之設計。我們初步的想法是利用階層式叢聚架構和節點本身的具位置知覺特性來做加快檔案找尋的速度，並保證節點在做檔案找尋時並不會跨出自己所在的叢聚區域(即所有資料至少可以從本身的 L1 的 root(x.0)取得)。當節點要查詢某一檔案的位置資訊時，只要先與本身的位置編碼和檔案的特殊標籤做字前比對(Prefix Match)後即可得知保管此檔案位置資訊的節點與本身是否在同一叢聚區域中，若是在同一叢聚中時，查詢者只須在此叢聚區域中找尋保管此檔案位置資訊的節點即可，若是在不同的叢聚區域時，查詢者只要先找到此叢聚區域後再深入找尋此節點即可。以這樣的檔案找尋演算法搜尋檔案，我們可以保證每次的查詢都會在本身所在的區段一(L1)叢聚區域中，不會跨出本身所在的區段一叢聚區域。我們將會在每一節點上設計一 routing table (cache)，儲存 root、parent、children 的基本資料外，

再加上以前曾查過的節點的資料(如其 IP 位址)。我們將會對此 cache 設計 cache replacement policy，如要 replace 時應先 replace 最久沒有用過的 L4 的資料，因 L2、L3、L4 所能提供的節點資料依序漸減，所以 L4 的資料最沒有價值。利用此一 routing table，當 1.1.2 要取得 1.2.3 的資料時，它會先比對 logic address 的 prefix，發現它可以跟 1.2, 1.0, 1.1 來詢問 1.2.3 的 IP 位址。如果 routing cache 中有 1.2 的資料，那它就直接問 1.2。否則，它會向 1.0 或 1.1 來詢問(default 它至少要認識 root 及它的 parent、children)。

(4) 節點離開: 我們將設計預知(graceful)與突然(abort)兩種可能下，節點離開時的演算法。我們目前初步的想法是，當節點的離開是比較屬於預知的離開時，那可告知父節點後離開並把其原先所保管的檔案位置資訊傳交給父節點，而其父節點則會選擇離開節點中區段等級最大(L2<L3<L4)的子節點來取代該離開節點。若是節點突然離開的話，其子節點將會是第一個知道其突然離開。由於節點是有叢聚的關係，突然的離開會有一些修復叢聚的工作要做。而且這個工作會因離開節點本身的區段等級有所不同有不一樣的動作。我們目前的想法是當離開節點為區段一的節點時，我們讓其所有子節點會與初始節點 1.0 聯繫並互相競爭成為新的區段一節點。之後由初始節點 1.0 依某一標準，如區段等級最大的爭競節點，選一節點成為新的區段一節點。若離開節點為其他區段節點時，我們就讓其子節點互相競爭取代離開節點，由此離開節點的父節點選擇一節點，如區段等級最大的爭競節點，取代該離開節點。

效能評估:

在具位置知覺之點對點檔案系統方面，我們將針對下面的因素與其他同類型檔案系統做比較：

1. 尋找檔案位置資訊等待時間。
2. 詢問路徑上的節點數。
3. 詢問路徑在實體網路上經過節點數。
4. 有節點失敗的情形下檔案詢問等待時間。

為了證明我們提出的分散式點對點檔案系統具有位置知覺性質和高容錯性，我們會與其他分散式檔案系統比較「尋找檔案等待時間」、「詢問路徑上的節點數」及「詢

問路徑在實體網路上經過節點數」，當這兩者越小，其代表發出詢問節點在其鄰近節點就找到其檔案，也就是詢問節點具有位置知覺性質，知道其在整個檔案系統的相對位置，使得詢問節點只會在自己座落的區域尋找檔案，不會跨出自己的區域找尋檔案。在使用者來看，能越快找到檔案，當然系統效能就越高。另外我們也會與其他檔案系統比較「有節點失敗的情形下檔案詢問等待時間」，同樣的其值越小代表系統容錯性較高。不會因為節點失敗造成系統效能下降太多，這也代表節點叢聚性對點對點檔案系統有一定程度的幫助。

有效量測技術分析：

在上述兩項研究中，我們都需要用到量測網路上兩主機間的延遲的技術。通常我們可以使用 ping 這種 active-probe-based 的方法，由想量測的主機主動送出 ICMP 封包給受測端，再收集其回饋回來的封包的時間，即可算出兩點之間的延遲。但我們也知道網路流量常常變動，每一次的量測所得的數據不見得都一樣，那該以那一次的量測為準呢?由於我們最常需要知道的是在一些 candidates 中，到底與那一個受測端有最小的延遲，所以在這個研究議題中，我們想要探討的問題是如何使用最少的 probes 得到最有信心的答案，或者是給定一個固定的 probes 個數，該如何使用這些 probes 使得我們知道離那一個受測端最近的這個答案我們最有信心。這個議題的研究步驟初步規劃如下:

(1) 我們需要先知道量測延遲的 distribution。當我們對同一個受測點發出 n 個 probes 後，我們可以得到 n 個延遲的樣本。我們是否可以從這些樣本中，推出其最可能的 distribution。例如我們可能可以發現在較近的距離(LAN)時，因變因不大，延遲可能是 constant 或 uniform distribution。而在較遠的距離(WAN)時，可能是 normal distribution。我們可以在不同的時間(白夾/晚上)就不同的距離、延遲、網路、國家進行測量與觀察，以推測不同的情形下可能 distribution。

(2) 假設我們可以推出可能的 distribution，那麼我們該如何安排 probe 以知道在一些 candidates 中，到那一個受測端有最小的延遲。例如，我們知道所有 distribution 都是 normal distribution，那我們先對 n 個受測端各發出一個 probe，得到各自的延遲。此時我們可以將這些受測端依延遲大小排序。那麼我們是否可以說下一次發出 probes 時，可以只發給前 x%的受測端或與最小

的延遲差異在 x 倍的標準差內呢?或者我們該先對所有受測端發三個 probe，求其平均，再做下一步的判斷呢?如果 distribution 是知道的，那麼我們相信是可以推導出一個較節省、較有效安排 probes 的演算法，而且我們知道我們的答案可以在某個信心程度內(如 95% confidence)。

(3) 同樣的，如果是因時間的限制，我們不能做到 95%的信心程度，那麼給定一個限定的 probe 數，我們該如何安排這些 probes，我們對所得到的答案又有多少信心呢?

(4) 以上的演算法可以用來和一個最基本的方法做比較。我們可以說對每個受測端都發 a 個 probes，然後再求各受測端的延遲平均，以此做為選最小延遲的受測端。或者，我們只能用 b 個 probes，那麼我們就對 n 個受測端平均來分 probes，每個可分得 b/n 個 probes，得到的延遲再求平均，以做為挑選的依據。我們可以比較要達到同樣的信心程度，我們的演算法可以節省多少 probes。或者我們可以比較在用同樣多的 probes 下，我們對答案的信心程度可以比基本的方法提高多少。

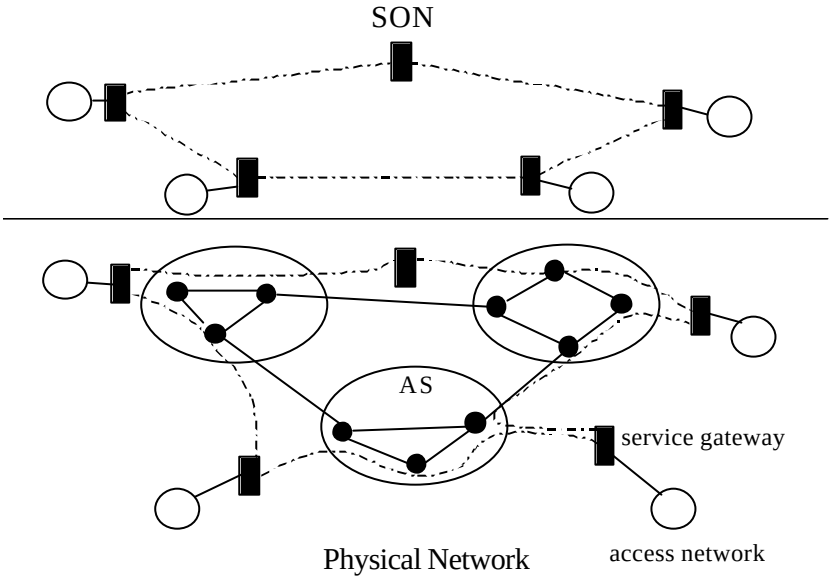
疊層服務網路之服務品質之研究:

網路上許多應用都需要有服務品質的保證，像是 VoIP、VOD 等。跟群播一樣，QoS 的議題十多年來一直停留在研究上，未能有效率地全面實施，所以從 ATM、RSVP、到 DiffServ，幾乎每隔幾年就會遇到實施上的瓶頸而被迫放棄。究其原因不難發現要提供 end-to-end QoS 所需要配合的，不僅僅是技術與協定，更重要的是各 ISP 的政策。所以，就如同群播一樣，在眾多關係複雜的 ISP 間想要達成提供不同 end-to-end QoS 的理想，恐怕也是很難實現。

但是 VoIP 等應用仍需有一定的服務品質保證，所以就有了 SON (Service Overlay Network)[32]的構想被提出。SON 的想法如圖十三所示，是要在現有的 Internet 架構上，再架一層由 SON Service Gateway 所形成的邏輯網路。為了提供服務品質保證，我們必須在 end-to-end 的客戶間的路徑上提供足夠的頻寬。這在 SON 的網路上是分兩部份完成。首先是 client 到 Service Gateway 間，這部份是由 client 跟 ISP 所租的頻寬，屬於 local loop 的 bandwidth provision 問題。另一部份是兩個 Service Gateway 間的頻寬，因可能跨 ISP，所以需要在一個 ISP 的 backbone 上租頻寬給特定的 service 用。以 VoIP 而言，client 在 access network 上的頻寬因目前大多是 100Mbps 以上之高速乙太網路，

所以不太會有問題。VoIP 的業者所要確保的是 client 的 access network 到 service gateway 及 service gateway 間的頻寬足夠。

向 ISP 租頻寬不管是目前以頻寬計費的消費模式或 DiffServ 的服務品質提供模式都是十分可行且容易計費的。但 SON 比 DiffServ 在頻寬或服務品質的提供上更容易做到更精準因為 SON 是屬於 service aggregation，也就是將同一種 service 的 traffic aggregate 起來，一起來 allocate 頻寬，以提供 service 的服務品質。因 service gateway 完全了解其 service 的 traffic 特性及所需要的服務品質，所以對該如何保留頻寬是可以做的比 DiffServ 更好的。



圖十三：SON 網路架構圖

在此計畫中，我們有興趣的是如何保留 SON 網路上的頻寬，以使 service provider 的收入可以最大化。因為服務品質保證是一個複雜的問題，但足夠的頻寬通常是最根本的解決方法，所以我們假設為了有服務品質保證，一條 SON link 上的總流量必須小於所保留的頻寬的某一比率以下，以數學符號表示的話，如果 link 頻寬是 c_l，服務品質的參數是 h_l (就是前面所說的比率值)，則此條 link 上的流量必須小於 $h_l \cdot c_l$，否則就無法保證所需要的服務品質。Service provider 的頻寬保留策略可以分成兩部份：靜態的頻寬保留及動態的頻寬保留。靜態的頻寬保留是假設 service provider 可以以長期固定租用頻寬的方式向 ISP 以較低廉的價錢租用頻寬。但租用固定頻寬時，需付固定的費用，不管是否真的有這麼多的流量。因流量隨時間而變化，所以固定頻寬不能租用太多。當流量大時，可以以動態方式租用額外的頻寬來保證 service 的服務品質。但每天每時的流量是未知的，我們不可能完全準確地預知下一時刻的流量，所以當動態保留

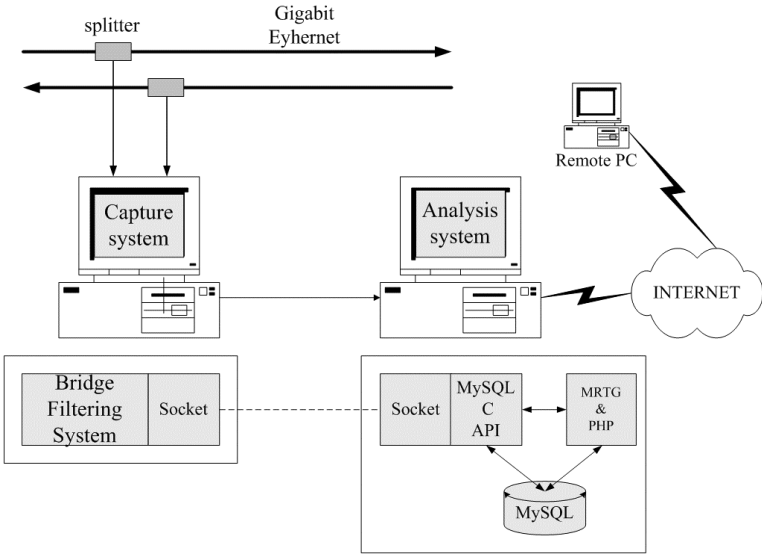
的頻寬加上靜態保留的頻寬仍不足保證服務品質所需的頻寬時，service 就必須付出一筆懲罰性賠償給消費者。如果以數學公式來表示的話，我們可以得到以下的式子:假設靜態頻寬保留了 c_l，其費用函數為 $\Phi_l(c_l)$；動態頻寬保留了 Δc_l，其費用函數為 $\Phi_l^{'}(\Delta c_l)$；所有路徑(assume one path per O-D pair)的集合為 R，假設每條路徑 r 上的流量是 r_r，路徑 r 上每單位流量每單位時間懲罰性賠償為 p_r，路徑 r 上每單位流量每單位時間收入為 e_r，則給定一組路徑之流量($\{r_r, r \in R\}$)下，我們要最大化的收入值 V 為

$$V\{r_r, r \in R\} = \sum_{r \in R} e_r r_r - \sum_{l \in L} \Phi_l(c_l) - \sum_{l \in L} \Phi_l^{'}(\Delta c_l) - \sum_{r \in R} p_r r_r 1\{r_l / C_l > h_l; l \in r\}$$

其中 $C_l = c_l + \Delta c_l$) 是總保留的頻寬，L 是所有 link 的集合，1{}函數表示括號中的條件成立的話為 1，否則為 0 。

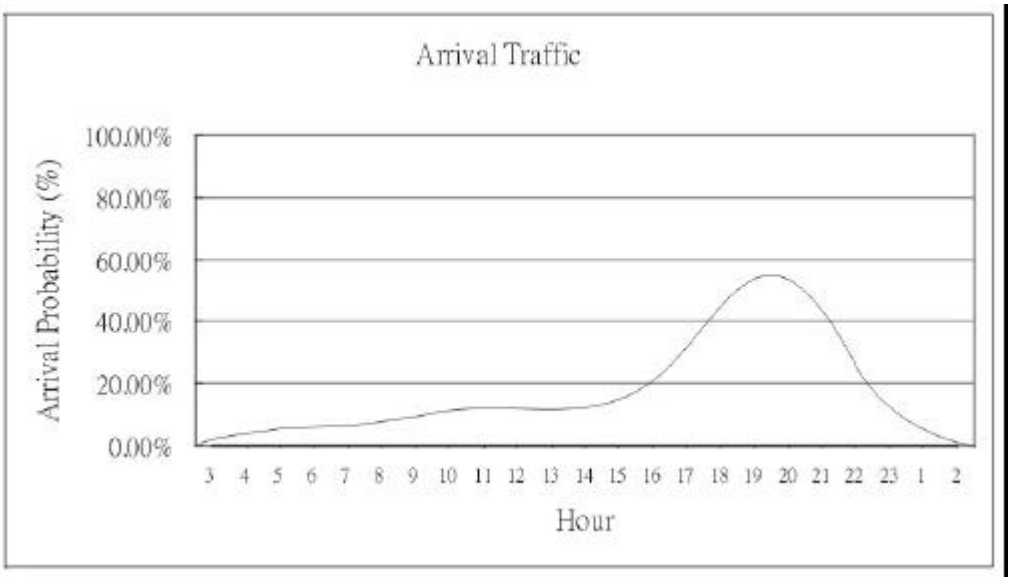
對於此一研究主題，我們最終的目標是發展一線上的動態頻寬保留機制，以達到收入值的最佳化。我們假設有過去的流量記錄可以觀察，做為流量預測的根據，先決定靜態該保留的頻寬，再決定動態該保留的頻寬。我們的研究步驟為:

(1) 收集流量資料，發展流量預測機制。流量的資料可以由一些知名的網站取得，如 University of Auckland [33]。但我們想自己收集系上及校內的資料。由於申請人曾任電算中心研發組組長，以往曾主持過統計中正 GigaPoP 流量的 NBEN 計畫，所以有能力收集流量資料。圖十四是去年計畫與中正電機系侯廷昭教授合作量測 TANet/I2 的 GE 介面的架構圖。對於流量的觀察，以往只是統計各種協定的流量分佈情形。我們希望將新收集的資料能進一步利用數學模式來分析，以建立預測機制。事實上我們可以從初步的觀察中發現，不同的時間的流量均不太一樣，但其每天 24 小時的變化情形卻是有跡可尋的。例如每天早上十點到十二點的流量在週一到週五事實上是差不多的量及變化。有此一規律變化的話，那流量預測機制就可以嚐試應用 Hidden Markov Model [34,35]、linear regression model (autoregression analysis, AR)、non-linear model (neural network)等技術。當然量測資料必須先進行 stationary 分析，對於不同時間下的平均值所得到的 data 可能會是 stationary 或 non-stationary 的數列。要對流量做預測，我們需先對 data 是否是 stationary 做討論。大部份的 model，如 HMM、AR，是不適合 model non-stationary data sequence 的。通常取長時間的平均值較可能得到 stationary 數列，有時因白天與晚上流量差異太大，需將一天分成幾個時段來分析，才可能有 stationary 數列。

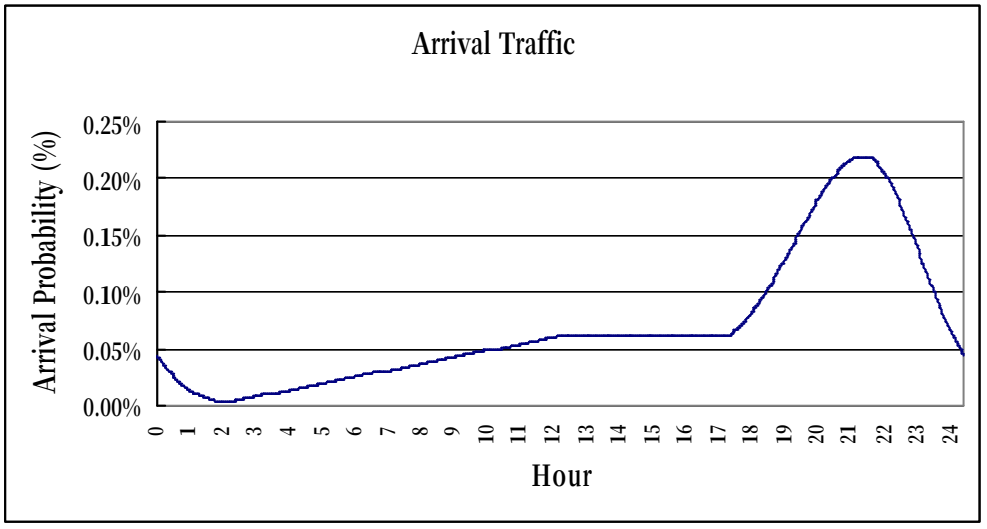


圖十四：Gigabit Ethernet **介面流量量測架構圖**

(2) **解決靜態頻寬該保留多少問題。以數學模式分析一理論上的最佳解。由於** traffic demand **是隨時間而變化，所以其** distribution **不可能以簡單的** well-known stochastic process **來** model**。但以往我們曾分析過** VOD **系統的** user request rate **也是隨時間而變化的，當時我們以** non-homogeneous Poisson process **來** model **其隨時間變化的特性, 以** normal distribution **配合不同的** mean **與** variance **及線性內插等方法，來得到如圖十五** (b)**的流量逼近圖。另一方面, 我們也需要將問題分成不同的時段分開解決, 因靜態頻寬是固定保留的，非常沒有彈性。所以我們至少可以依所觀察的流量變化分成白天和晚上，或尖峰時段與非尖峰時段。像圖十五** (a)**，我們很明顯可以看出在尖峰時段(晚上七點到十二點左右)的** distribution **與非尖峰時段的** distribution **非常的不同, 前者可以用** truncated normal distribution **來逼近，後者可以用線性函數來逼近。當然也許我們收集到的流量分佈是非常不規律的，那麼我們可以以較不準確的分析(因資訊不足)來求得該保留的靜態頻寬。例如我們至少可以從歷史的觀察中，量測出流量的平均值與變異數。依此資訊，我們將研究是否可以因費用函數是** non-decreasing function **而算出較好的靜態頻寬保留值。**



(a) VOD user request rate **之量測值**



(b)**以** non-homogeneous Poisson process **來** model VOD user request rate

圖十五: Non-homogeneous Poisson process model time-variant traffic demand **例子**

(3) **設計線上動態頻寬保留演算法。我們將考慮不同的預測機制下如何動態保留頻寬、頻寬的租用如有最小單位的限制下如何調整、如何避免太大的變化造成系統不穩定等等問題。此演算法的重點在於如何利用歷史量測資料發展出流量預測之** model**，並動態將新的量測資料回饋回數學** model**，來對下一時刻做出更精準的預測。當然其前提是給定一正確的流量需求，我們可以用數學模式準確算出所需要保留的頻寬。**

(4) **評估與比較不同之線上動態頻寬保留機制與流量預測機制。 我們可利用別人或自己收集的流量資料來比較不同的機制的** performance(**即** revenue)**。**

2、預計可能遭遇之困難及解決途徑

點對點實況廣播技術與點對點檔案系統技術是目前相當熱門的研究題目，論文集中在 2001 年以後大量發表，這一、兩年將是一個競爭激烈的熱門領域。但因此部份研究已在申請人的研究規劃中，所以在應用層群播技術、串流技術、點對點檔案系統，網路系統模擬等理論的背景，申請人及研究人員均已早在 2002 年暑假即開始研讀相關的論文與書籍。

3、重要儀器之配合使用情形

此一年度之計畫，部份研究主題需要進行網路流量量測及 BGP routing table 之收集，另也需要大量的模擬程式撰寫與系統模擬分析評估，所以預估需要三台個人電腦。至於研究人員的開發平台，因申請人已執行國科會計畫多年，有足夠的個人電腦提供他們使用。

第二年:

這一年將以發展 Internet BGP 路由為基礎的另一個具網路知覺之點對點檔案系統技術，以及實作點對點架構上之視訊會議系統為主。

1、本計畫採用之研究方法與原因

具網路知覺之點對點檔案系統:

在第一年中，我們依賴使用 active probe 所得到的 delay 來做為形成 cluster 的依據。Active probe 通常使用 ping、traceroute 等程式，其底層大部份是使用 ICMP 封包。由於這類的 probe 多多少少需要 router 的處理, 所以常會有被 router 限制每秒可接受的封包數而被 drop 掉。所以使用 active probe 雖方便，但有增加網路負載及失真的疑慮。所以在第二年中，我們預計改以收集一些 routing 資料，例如 BGP routing table 或 BGP routing messages 來推測網路相關位置。再利用此一推測出來的網路拓樸(Internet topology)，做為形成階層叢聚關係的依據。目前已有美國 MIT 學者發表利用 BGP 資訊來推論 Internet topology 的初步成果[36]。

與[36]不同的是，我們將利用 BGP routing table 為主來推論 Internet topology。基本上目前在 Internet backbone 上的 router 的 BGP routing table 相當的大，約有十二萬筆左右。但如果仔細觀察，許多不同的 IP prefix 其實都屬於同一個 ISP 所 service，所有有共同的 destination AS number。如果我們有辦法有效地將各 IP prefixes 與 AS number 做一對照表，那麼要形成階層叢聚關係時，當然同一 AS 要在同一 branch 下。類似的做法，在同一 AS 中，我們也許有可以得到更進一步的 routing 資訊或輔以其他 active probe 的資訊，來做更細的 cluster 的形成。也就是說，我們先利用 BGP routing table 的

資料，是否可以形成一 Internet topology 的資料庫。當進行 P2P 的架構時，可以先 query 這資料庫，得到它在 Internet topology 的位置後，再決定形成 cluster 關係的策略。所以我們稱此種方法為具網路位置知覺的點對點檔案系統。

對於此一研究主題，我們的研究步驟大致如下：

(5) 對 BGP routing table 進行收集分析，以推論出合適的 Internet topology 資訊。目前研究人員取得 Internet BGP routing table 的管道大多是使用 University of Orgeon, 的"RouteViews"計畫[37]。由於申請人曾任電算中心研發組組長，所以之前也曾取得並分析中正大學連接 TANet 的 router 的 BGP routing table 之資料，但因環境單純，所以筆數不多。後來也曾透過與成大合作，取得成大跟 ISP 另外租頻寬出國的那顆 router 的 BGP routing table 約有十二萬多筆。我們預計使用這些資料，進行不同的 IP prefix 是位於那一 AS 或更詳細之資料，並建立一個 logic Internet topology。以往在這部份的研究大都以觀察或統計一些數據，如 routing table size 成長趨勢[38]、AS policy 對路徑長度分佈的影響[39]等等。我們所要 explore 的是 logic Internet topology，也就是從網路建置、連接的關係來看各主機的距離關係。(兩台地理位置相近的主機可能因連接不同的 ISP 而在 Internet 上的距離很遠。這裏的 Internet 距離指的是 delay 或 bandwidth 的關係。距離短的意思可以是說 delay 小或 available bandwidth 大。)由於 BGP routing table 的資訊可能有限，如只能 explore 到 IP prefix 與 AS 的關係，我們還預計利用 SNMP 或其他 active probe 的方式來進一步建構出更精準的 Internet topology。

(6) 研究對於建構出的 Internet topology 如何使用在建置 P2P 的階層叢聚關係。如何有效率地分散 Internet topology，取得或分析兩台 Internet 主機的關係，是需要進一步設計出一個查詢協定的。再者，除了利用 Internet topology 所提供的資料外，是否還需配合少量的 active probe 也需要進一步評估。

(7) 修改第一年的成果，將以 delay 為依據形成的階層叢聚關係改以 Internet topology 的資訊為依據。此時，點對點檔案系統該如何建構，其相關的議題，如檔案分配、查詢、路由等，均需相對地調整。

(8) 研究是否也可將 Internet topology 應用於點對點實況廣播系統上。

點對點架構上之視訊會議系統:

在探討了單一來源的點對點即時廣播問題後，我們將研究延伸到多來源的群組通訊(group communication)上。我們的動機源自於目前我們所開發的網路教學平台上，學生與老師即時互動最常用的工具:文字與語音聊天室。雖然我們系統提供了多人語音聊天室，但使用的人不多。究其原因，當然使用者沒有錄音設備也不熟悉語音操作有關，但有許多的時候是源於 client/server 架構的關係。例如，許多國中小老師上網上課時，是使用學校的電腦教室，頻寬是共用一條 64K 專線或 ADSL。所以當多個老師修同一門課時，就會發生頻寬不足的問題，因每個 client 都由 server 取得即時的語音資料。而當 client 多時 server 的 load 也相對很重。所以 client/server 架構就顯得非常不 scalable。為了解決 scalability 的問題，我們目前是改由各個聊天室的主席(通常是老師)的主機當 server，原 e-Learning server 只是提供 membership 登錄及 client 找到主席主機的管道。但這還是沒有解決 client 端頻寬的問題，而 server 仍是 single point of failure。

在設計以點對點架構的視訊會議系統時，我們將分以下議題來分析與研究:

(1) 成員管理問題: 我們將先以網路教學系統的需求來設計這個系統。所以在成員管理上, 我們可以採 hybrid 的模式, 即會員資訊可以由網路教學系統 server 做收集，再由點對點的協定來做成員間動態的管理。基本上成員可能來來去去，但只要透過 server 給的 list 中找到一個還存在的成員，即可由我們設計的協定完成加入的動作。當然，我們也會一併考量當沒有 server 時，我們如何利用類似 boot server 的作法來解決找尋組群成員的問題。

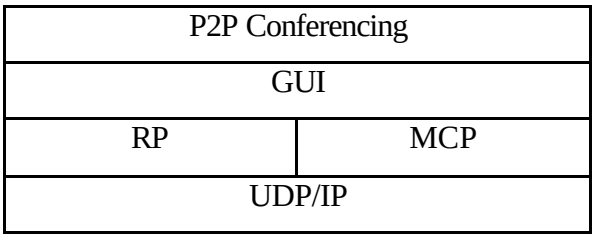
(2) 階層式叢聚關係的建立: 我們可利用第一年的成果來動態建造群組成員間階層式叢聚的關係。但我們需要考慮到多來源的問題, 所以在形成叢聚階層時，需從新的角度來設計。如何將成員組成叢聚倒是可以沿用第一年的成果。

(3) 資料傳輸問題: 系統將提供使用者可以用 text、audio、video 等三種 media 的互動工具。如何將其中一個成員送出的資料傳給所有其他人需靠階層式叢聚的關係及中間節點的 relay。另我們也需對不同的 media 考慮不同的傳輸策略。例如，當頻寬不足時，應以 text、audio、video 之順序來 drop packets。成員可以依其頻寬大小來選擇其中幾種 media，而不是全部都收。由於傳 text、audio、video 的難度也依此順序遞增，所以我們實作時，也會以此順序來完成。即先有 text 的聊天室後，再加入 audio，最後再加入 video。

(4) LAN 中 multicast 的使用: 如上面所舉的問題，在同一 LAN 下的成員該很有效率地使用頻寬否則對外的頻寬會不足。所以我們想將 LAN 下所有成員

group 成一虛擬的節點，由其中一個節點代表 LAN 下所有其他節點，我們稱之為 LAN agent。在同一 LAN 下的節點間的資料以 multicast 真接傳送，以節省頻寬。(LAN 下的 multicast 完全不需 router 的支援。)

(5) 系統平台與架構: 沿襲我們目前網路教學系統平台的做法，我們將以 Windows 為開發平台, 完成後的程式再包裝成 ActiveX 元件方式嵌入網頁中。使用者只要第一次瀏覽網頁時下載程式，以後就可以直接由 client 端直接執行了。系統架構初步構想如圖十六所示，使用者透過 GUI 介面，收送 text/audio/video 資料及取得群組成員名單。GUI 部份可如目前我們的聊天室加入額外的功能，如開闢公開/私人聊天室、私人通訊、加表情動作…等等。上層的各種工具則透過 Relay Protocol (RP)及 Member Control Protocol(MCP) 進行資料的傳輸、階層式叢聚關係的建及動態群組成員管理等等功能之實現。下層則以 UDP/IP 為網路層傳輸協定。



圖十六: P2P Conferencing protocol stack

2、預計可能遭遇之困難及解決途徑

在系統實作的部分主要需要 Windows network programming, 以及 Windows Inter-Process Communication 兩方面的技術。目前我們已有人力在這兩個方面擁有相關程式撰寫的經驗，所以估計不會有太大的問題。

3、重要儀器之配合使用情形

在具位置知覺之點對點檔案系統技術發展上，我們沿用第一年設備即可。在實作點對點視訊會議系統時，我們希望測試的網路可以跨不同的網路，所以規劃六台多媒體個人電腦來進行實作與實驗。實驗環境將包括系內 LAN 環境、校內不同系的 LAN、TANet 上不同學校間(預計有成大及交大合作進行)、嘉義縣市國中小(測多台 peer 在同一 ADSL 下與中正的 peer 連接)、家裏 ADSL 上 Hinet 等 ISP 網路連回中正的 peer 等等。其目的在於測試有極大差異的頻寬及骨幹網路間 peer adapt 到不同架構的情形。

第三年:

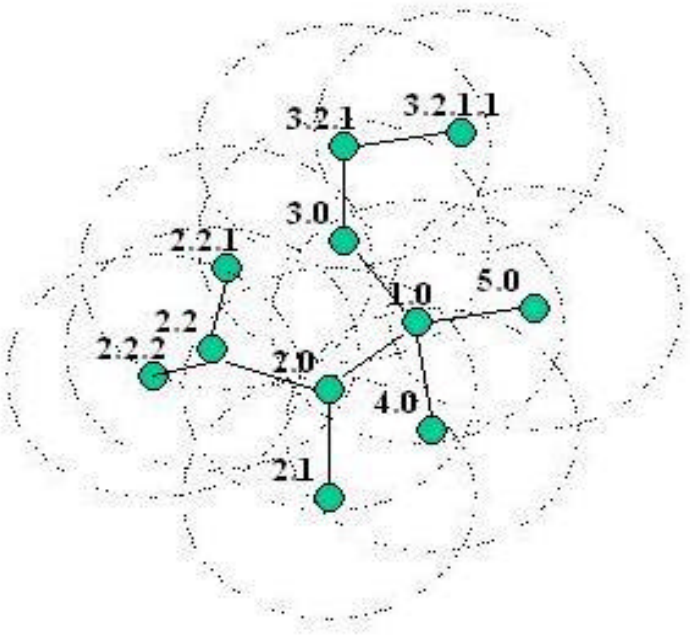
這一年我們將之前之研究延伸到無線網路(特別是 ad hoc **架構)的點對點資訊分享系統，並實作由不同載具、介接網路所形成的網路上的資訊分享系統。**

無線網路之點對點資訊分享系統:

想像在未來個人行動通訊普及的時代，在一列高速火車上或一架飛機上，大家都拿出個人行動載具，從事數位學習或各種娛樂(聆聽 mp3**、看電子報、看** DVD **等)。此時如果有人想要取得某份資料，如找今天某家的電子報或某一首** mp3**，那他可付費透過第三代行動通訊系統上網取得。或者，他也可以透過** ad hoc **網路，在火車/飛機上其他人願意分享出來的資料中找尋。後者顯然是較快速而經濟的方法。所以，在第三年，我們希望將前兩年的成果應用到這樣一個** ad hoc **的無線網路上。**

我們的目標是研究一個點對點架構的無線網路環境的資訊分享系統而不是檔案分享系統。因為在移動頻繁的無線網路中，我們所需要的不是完整的分散式檔案系統，而是一個有效率的資訊分享與索尋系統。例如，我們只是想在現有的 ad hoc **網路中看看能不能找到一首剛好想聽的歌，但不見得一定要找到。所以，在設計這樣的系統時，我們需要考慮的是** ad hoc **的網路傳輸特性、節點的移動性、資訊的找尋與路由的關係等等議題。我們的研究步驟大致可分為以下幾點**:

(1) **利用** ad hoc **廣播傳輸的特性建造類似第一年的具位置知覺的檔案分享系統的階層叢聚架構。例如當一個節點開機後，可以發出** hello message**。如果沒有其他節點回覆，那此節點可自行將** logic address **設為** 1.0**。如果有其他人回，則依某一標準選其中之一為其父節點(目前的想法是可從以下參數中取一或組合其中數項來研究**: power **最大、**single **最強、**mobility **最低、在** hierarchy **中階層最高等等等)，並請求父節點給予一個單一的** logic address**。例如父節點如果是** 1.0**，那可能會指定** 1.1 **給此子節點。靠著各節點傳輸的** power **及** broadcast **半徑的不同，會形成如圖十七的一個階層叢聚架構(虛線外圈為其** broadcast **半徑)。值得日後我們詳細研究的是，此一** logic **的階層叢聚架構的層階性與可直接通訊的鄰居有**(1) parent **與** child **一定可以互通**; (2) **不同** logic address **的** prefix **可能可以互通。在以下我們考慮** routing **時，必須注意此兩個特性。其中第一個可做為我們的基本** routing **策略，第二個則可以以** routing cache **的觀念來增進** routing **的效率。**



圖十七:一個階層叢聚架構。

(2) **設計一雜湊函式來將資訊對應到階層式叢聚架構的一個節點的** logic address **上。此處的資訊可能是一個檔名，也可能是一份電子報的名稱，也可能是某一個人的資訊。當一個節點加入此系統時，它就利用此一雜湊函式將它想要分享出來的資訊登錄到相對應的** logic address **的節點上。因為** ad hoc **網路頻寬是相當珍貴的資源，所以我們不能像檔案系統那樣直接把檔案或資訊儲存在相對應的節點上。我們只讓相對應的那個節點記錄下來這個資訊在那個節點可以找到。**

(3) **當要找尋某一資料時，我們利用同一個雜湊函式找到一個節點。送出** query message **給那個節點，再由那個節點根據所記錄的資料將** query message forward **到真正有那份資訊的節點上。再由此節點直接把資訊傳回給需要此資訊的節點。**

(4) **我們也需設計一個有效率的路由策略。初步的想法是利用** logic address **的** prefix **來決定路由。最基本的做法是判斷** logic address **是不是一樣，不是的話，就往** parent **送。例如** 1.2.2 **要到** 1.3.1**，則其路徑是** 1.2.2->1.2->1.0->1.3->1.3.1**。這是因為我們在指定** logic address **時，**parent **與** child **一定可以直接通訊，所以只要沿著** hierarchy **的** path **往上走到最上層的共同** parent **後再往下走，就可**

以到達 destination 了。但這個路由方式不是很有效率，因一個節點不只是可以直接和它的 parent 通訊，還可能有其他節點在其廣播的範圍中。所以我們將設計一個利用節點聆聽其他節點的廣播訊息及收到/轉送封包的資訊中，建立一個 local 的 routing table。當要傳資料到某一 destination 時，除利用上述基本方法外，可以靠這個 routing table 找 short cut 出來，讓路由更有效率。例如上述的例子中，如果 1.2.2 與 1.3 是在彼此的傳輸半徑中，那 1.2.2 可以從其 routing table 的記錄中知道可以直接傳給 1.3，配合基本方法，1.2.2 要到 1.3.1，就可以直接透過 1.3 來傳送。

(5) 我們將設計預知與突然這兩種節點離開情形的演算法。當節點移動時，比較屬於預知的離開，那可告知 parent 後離開。等移動到新的位置取得新的 logic address 時，再將資訊做更新登錄的動作，就是告知相對應的節點其 logic address 已更新。突然離開的話，會造成 query message 無法送達及路由錯誤。此時要靠 soft-state refresh、monitoring and update 等機制來更正。當 message 無法送達時，我們也會設計一 feedback message 將相對應的節點上原本登錄的資料或路由表更正。

(6) 如前兩年的研究，我們在完成系統設計時，也將進行系統模擬以評估效能。

點對點資訊分享系統在無線網路之實作:

當我們充分利用 ad hoc 網路的特性設計好一個點對點架構的無線網路環境的資訊分享系統後，我們將實作在具有 WLAN 介面的 PDA 及 Tablet PC 上。我們將以名片分享及 mp3 分享為 demo 的例子。也就是說透過我們的系統，不同的人可以很自由地在一個 WLAN 的 ad hoc 網路上，找尋某人的名片(通常就是本人，所以也就是可以利用此一機制來找看看在這個 ad hoc 網路中這個人有沒有上來)或一首 mp3 來聆聽。

2、預計可能遭遇之困難及解決途徑

在系統實作的部分需要在 PDA 上的 WinCE 作業系統下撰寫程式，並架構 ad hoc 網路。Tablet PC 的作業系統可以直接用 windows。這兩方面的技術我們將利用前兩年的時間先進行相關的技術訓練及資料研讀。由於本系陳裕賢教授及許政穆教授也有相關的經驗，所以應可順利取得相關的技術。

3、重要儀器之配合使用情形

因要在 PDA 及 Tablet PC 上實作，所以需要此種平台各兩台並各自配搭無線網路卡。

（三）預期完成之工作項目及成果

1. 預期完成之工作項目

第一年：

1. 完成點對點實況廣播技術的設計及其效能評估。
2. 完成具位置知覺之點對點檔案系統的設計及其效能評估。
3. 完成量測技術分析，得知如何做到最有效的量測。
4. 完成 SON 網路的線上即時動態頻寬保留演算法設計及其效能評估。

第二年：

1. 完成具網路知覺之點對點檔案系統的設計及其效能評估。
2. 完成 P2P 視訊會議系統之實作與實地測試。

第三年：

1. 完成 ad hoc 架構的無線網路的點對點資訊分享系統設計。
2. 在 PDA、Tablet PC 上實作無線網路的點對點資訊分享系統。

2．對於學術研究、國家發展及其他應用方面預期之貢獻

應用層群播與點對點檔案系統是目前在網路研究重要的研究領域。此計畫的進行，對我國在這兩方面的研究均有幫助。本計畫理論與實作並重，其成果相信不管在學術上或在應用上，均會有相當大的貢獻。

3．對於參與之工作人員，預期可獲之訓練

◎ 所有參與研究人員將學會如何搜集資料及懂得回顧研究領域的重要性。

◎ 所有參與研究人員將學會新的應用層群播技術並獲得相關實作經驗。

◎ 所有參與研究人員將學會新的點對點檔案系統技術並獲得相關實作經驗。

◎ 所有參與研究人員將學會數學模式分析能力、網路系統實作及視窗網路程式設計技巧。