具快速恢復特性之多源應用層群播影音串流系統之研製

廖梓彤、柯開維*、吳和庭、蔡文能
國立台北科技大學資訊工程所

摘要 —對等式網路技術已發展穩定， 對等式的特點在於資源共享的權力是在各個用戶手上而非單一網站，而技術的應用使得網路上的資源得以共享並得到充分利用。本系統是基於應用層群播技術，新成員加入群組後計算與每位成員之間的距離，建立樹狀的傳輸路徑，並有效的減少新的節點加入或舊節點的斷線與離開的調整時間[1]。

一、 簡介

近年來，網路的應用越來越頻繁，從雲端儲存、即時語音甚至到隨選多媒體等應用越來越廣泛。而隨著使用網路的人口數逐年攀升，網路上傳輸的資料量不斷提高，頻寬需求亦相當驚人，所有的傳輸都集中在骨幹網路上，使得骨幹網路上的負載相當重，並且在網路大量的資料傳輸中不難發現，相同的資料傳輸佔所有資料傳輸的比率非常高，如果將重複的傳輸降低將可以使骨幹網路的傳輸獲得舒緩。

了讓伺服器面對大量的存取之餘又不超過系統負荷量，提升與擴充系統處理能力成為了伺服器業者重要的研究方向，許多分散式與叢集式網路架構的概念紛紛被提出，用以改善大量傳輸時伺服器負擔的問題，提高伺服器資源的可用性。

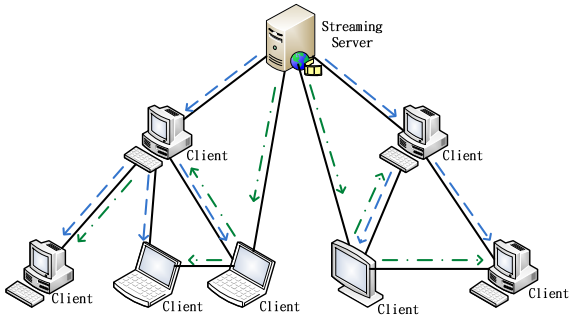
另一方面，對於網路系統業者而言，骨幹網路為了能夠處理大量的資料傳輸，必須花費大量成本在提升網路傳輸的能力以滿足顧客越來越大的傳輸需求，然而問題並不會因此獲得改善，一旦提高網路傳輸能力，網路傳輸的資料量將會不斷地攀升，又將發生傳輸瓶頸的問題，而重複發生不斷的相互影響。

在本系統實作使用應用層群播之演算法，建立串流傳輸的群播樹狀網路，達到降低重複傳輸對於網路骨幹的負擔，由群播的群組成員協助傳輸降低骨幹網路的重複性，此外還可達到分散存取與傳輸量的效果，降低伺服器的負擔，滿足多媒體傳輸上的伺服器與網路傳輸的問題，使聯網電視大資料量傳輸服務能夠更有效率。

二、 應用層群播傳輸路徑

樹狀的架構分為兩種[1]：一種是僅使用單一樹狀傳輸路徑；另一種是使用如圖一所示之多樹狀傳輸路徑，同樣是單一來源但分為兩種路徑傳輸，當某節點發生問題時可改由另一條路徑傳輸，因此可以得到較可靠的傳輸，具有節點變動快速恢復的容許能力。

[1] 本研究由國科會贊助，計畫編號 NSC102-2219-E-027-003。



圖一：多樹狀串流傳輸

在樹狀的傳輸架構下具有較佳的時間同步性，但樹狀網路相對於網狀網路的恢復要花費更多的成本，必須有良好具快速恢復能力的演算法進行樹狀路徑的維護，在本系統實作中，採用樹狀的架構與 Application Layer Multicast(ALM)[2]的方式傳輸影音多媒體內容，並參考論文應用層群播路由協定之設計與效能分析[3]，實作其 Distributed Multisource Forwarding Trees(DMFT) 與 Cluster Distributed Multisource Forwarding Trees(CDMFT) 兩個串流群播樹建構與維護的演算法。

在 DMFT 與 CDMFT 演算法中，每個成員建構自己的群播樹，因此具有多源之特性，相較於單一來源，本系統使用之演算法更具有彈性，單一來源僅會有一個來源端，其他成員只扮演接收與轉送的角色並無法提供資源傳輸，使得單一個網路中只會有一個串流傳輸，採用多源的方式可以由各個成員在同一個網路中發送串流，為使用此多源演算法之優點。

三、 DMFT 演算法

在 DMFT 演算法中，定義鄰近成員資訊表(Neighbor Information Table, NIT)與資料轉送表(Data Forwarding Table, DFT)並使用這兩份資料表，建構每個成員自己的群播樹，維護的這兩個表格功能分別為：

1. 成員資訊表

   這個表格欄位包含網際網路位址(Internet Protocol Address, IP)及中繼段個數(Hop Count, HC)，記錄鄰近成員與自己距離的 HC 值，HC 值越大代表距離越遠，越小代表距離越近。中繼段個數即為經過路由器的跳躍值，HC 量測方法可使用 IP Header 的 Time-to-Live (TTL)欄位值與初始值的差計算得到。群組中每個成員都必須要有自己的 NIT，表中包含群組中所有的成員與自己的距離，用以計算路徑。

Corresponding author:kwk@csie.ntut.edu.tw
DOI:10.6159/IJSE.2014.(4-2).56

2. 資料轉送表

每個成員都將 NIT 中從小到大依序排列，因此可以取得 HC 最小成員，即為距離最近的成員，並將其加入至 DFT 中成為路徑中傳遞的下一個成員，而除了 HC 最小的成員以外，將其他的成員清單交給路徑的下一個成員，當收到清單後重複相同的動作找尋 HC 最小的成員並加入 DFT 以此類推完成樹狀路徑下所有成員的 DFT。在 DFT 記錄著資料來源端的網際網路位址(Source Internet Protocol Address, Source IP)及其所有子節點成員的網際網路位址(Next Hop Member Internet Protocol Address, Next Hop Member IP)，來源端 IP 欄位為串流的資料來源的位址，子節點成員 IP 欄位為一個或多個成員的位址，DFT 這個資料表決定資料下一個轉送的路徑。

此演算法中定義五個運作機制，分別為成員加入(Member Join)、建構樹狀路徑(Build Tree)、修剪重複路徑(Prune)、成員離開恢復(Recovery)與定時回報，以下為各機制的詳細說明：

1. 新成員加入(Member Join)步驟

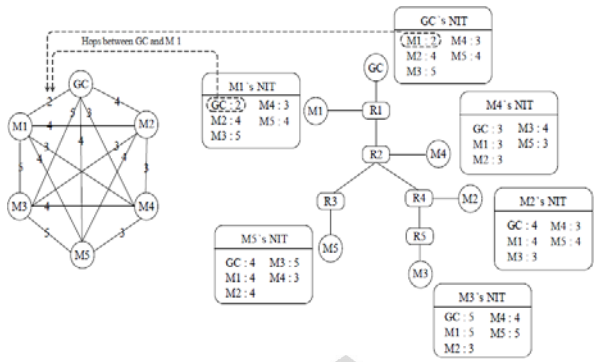
(1) 在建立群播群組之前，每個新成員都必須成為群組創造者(Group Creator, GC)的角色，並且被動的等待其他成員發送請求加入。若無其他成員加入可自行發送加入請求至步驟(2)否則繼續在步驟(1)等待。

(2) 若有已知的 GC 可以加入群組時，新成員成為群組成員的角色，並主動發送加入群組的請求，當 GC 收到該請求後，會回應 NIT 中所有成員的清單給新成員

(3) 新成員收到清單後發送新加入的訊息給群組中的所有成員，通知並取得各成員的 HC 值。

(4) 其他成員也同步更新 NIT 即完成成員的加入。

在 NIT 完成更新之後，如圖二所示，所有的節點皆會更新屬於自己的 DFT，即建構樹狀路徑。



圖二：完成 NIT 更新後的網路拓樸

2. 建構樹狀路徑(Build Tree)

群組內的所有成員皆擁有屬於自己的樹狀路徑，每個成員都會成為樹狀路徑中負責轉送的一員。在資料來源的成員要建構樹狀路徑步驟：

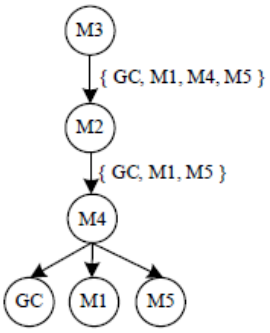
(1) 成員會從 NIT 中挑選 HC 最小的成員成為子成員，如圖三中 M3 所建立的樹狀路徑，M3 從 NIT 中找到 M2 為 HC 最小的成員。

(2) NIT 中除了子成員以外的成員組成接收成員集合的訊息(Receiver Set, RS)，並發送給所有子成員。

(3) 當子成員收到 RS 訊息後，將 RS 訊息中所有成員比照查詢自己的 NIT，找到 HC 最小的一個或多個成員成為自己的子成員，如圖三中 M3 為資料來源時 M2 收到 RS 後找到 M4 為 HC 最小的成員並將 M4 從 RS 中刪除並發送給 M4，當 NIT 中發現成員有相同的 HC 值時將其加為樹狀路徑中同一階層，如圖三中的 GC、M1、M5。

(4) 重複步驟(3)直到 RS 訊息中無成員資訊可以取出為止。

迭代的由所有成員完成建構樹狀路徑的方式，能夠將建構所耗費的負擔分攤給所有的成員。



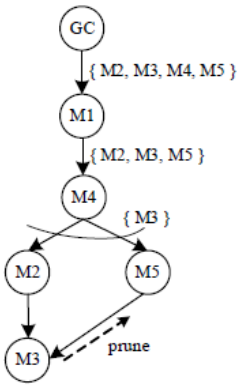
圖三：M3 完成 NIT 更新後的網路拓樸

3. 修剪重複路徑(Prune)

在建構樹狀路徑的機制中，資料會依序的從來源端發送到樹狀路徑中的葉節點成員，然而發送資料的過程中可能會因為 HC 值相同而產生路徑分歧，即一個成員有多個子成員的傳輸路徑。產生多個子成員的原因是在成員比對 RS 及 NIT 中的成員後，找到不只一個 HC 為最小值的成員，這讓多個子成員都會被建立在樹狀路徑的同一層，而同一層的子成員們會得到相同的 RS，從 RS 中相同的成員清單找尋子成員繼續往下建立樹狀路徑，但路徑的建立從 RS 的清單中找尋適合子成員建立路徑，在分歧後多個成員得到相同的 RS 從相同的清單中找尋子成員會產生重複的路徑，如此一來就會發生重複轉送資料的問題。此修剪機制正是用以解決重複分支的問題，運作的流程為：

(1) 當成員收到的重複的訊息，即從同一個樹狀路徑的來源發送的訊息卻又是不同復節點成員時，判定為重複的路徑。

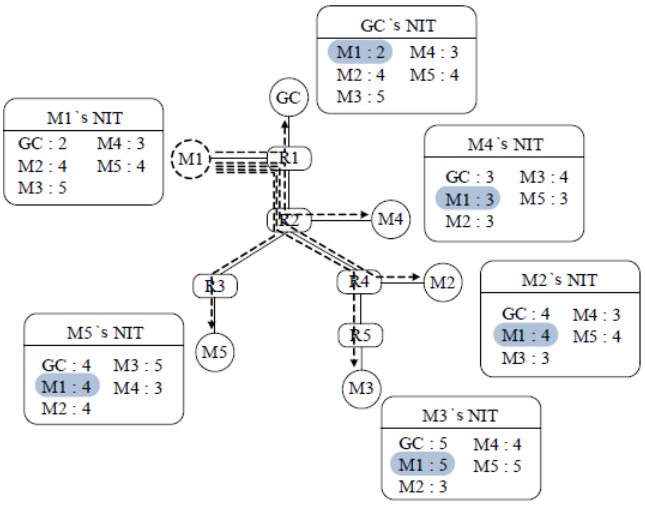
(2) 當發現重複時成員會根據自己的 NIT 保留重複的父節點成員中 HC 值最小者，並告知其餘的復節點成員不需要這個傳輸路徑，即完成修剪，如圖四所示 M3 收到 M2 與 M5 的訊息並從 NIT 中找到 M2 有較小的 HC，將發送 Prune 訊息給 M5 完成修剪。



圖四：GC 完成 NIT 更新後的網路拓樸並修剪重複的傳輸

4. 成員離開恢復(Recovery)

成員離開分為正常與非正常的方式離開，正常的方式離開，是一種有禮貌的離開方式，即在成員要離開的時候發送離開的訊息，這種方式可以有效的迅速恢復網路狀態；另一種則是非正常的方式離開，是一種網路異常的情況，即成員斷線或強制關閉等情況發生，這種方式並無法快速的恢復，僅能在測試傳輸如心跳封包(Heart Beat, HB)或是實際傳輸卻無回應時才會發現斷線必須要調整恢復網路的連線。不管是哪一種方式，當成員發現有子成員離開群組時，會將離開的成員從 NIT 與 DFT 中刪除，並且重新根據 Receiver Set 重新建立樹狀路徑。

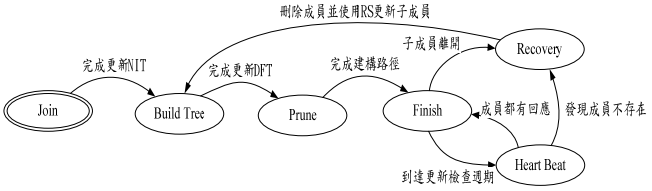


圖五：成員 M1 離開，通知其他成員將其刪除之程序

5. 定時輪詢

定時輪詢使用心跳封包(Heart Beat, HB)是用來檢查連線的情況，每隔一段時間會發送一次 HB，用來檢查成員的狀態，不用等到傳送資料時才發現成員不存在而緊急調整傳輸的路徑。HB 可以加快網路的狀態的更新，週期性的更新，一旦發現成員不存在則進入網路恢復狀態。

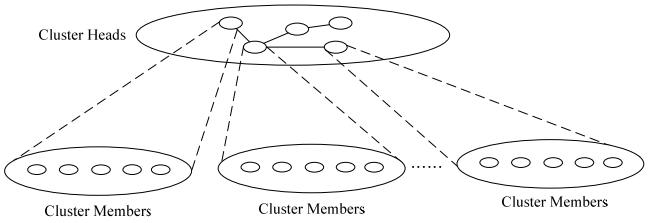
這五個機制的互動狀態如圖六在 Join 完成之後即維護此樹狀架構，並透過 HB 定時的更新確保使用最新的路徑傳輸。



圖六：樹狀網路維護機制的狀態圖

四、 CDMFT 演算法

叢集是指將原本大的群組劃分為數個獨立的小群組，這樣的方式可以避免過度集中在同一個群組中，而能夠區分成許多個小群組獨立的運作，因此能夠由許多個叢集組成原本大的群組，並由叢集中的一個或數個成員與其他叢集交換訊息，使其負載分散至各個叢集獨立運作，如圖七。在此演算法中繼承 DMFT 的演算法，並增加叢集代理(Cluster Head, CH)與備用叢集代理(Backup Cluster Head, BCH)兩個角色，CH 和 BCH 會維護叢集代理表(Cluster Head Table, CHT)與叢集成員表(Cluster Member Table, CMT)兩張表，而產生許多叢集分散於網路的效果。



圖七：CDMFT 示意圖

針對 DMFT 角色修改的部分， CH 為叢集的代理負責管理叢集中所有成員對叢集內外的傳輸，而後來加入的成員會依照所有 CH 的 HC 找尋最近的成員加入其叢集成為叢集成員；BCH 為 CH 離開時，重建 CH 之間的網路的依據，由於叢集中的成員都是由 CH 管理，一旦 CH 離開將無法找到叢集中的所有成員，使 BCH 成為 CH 並從叢集成員中找尋 HC 最小的成員成為 BCH。

CDMFT 將 NIT 以 CH 和 CM 區分，新增叢集領頭表(Cluster Head Table, CHT)與叢集成員表(Cluster Member Table, CMT)，CHT 由 DMFT 演算法維護，CMT 則由修改的機制進行維護。機制修改如下：

1. 成員加入：當沒有叢集存在或是沒有加入叢集時為新的 CH；當有叢集存在時，加入叢集成為叢集中的成員。如果成為新的 CH，則與群組中的其它 CH 使用 DMFT 的方式建立樹狀的路徑並更新 CHT；如果成為新的叢集成員，則與叢集中的其他成員使用 DMFT 的方式建立樹狀的路徑並更新 CMT。
2. 建構樹狀路徑：(a) 叢集代理的樹狀路徑，使用 CHT 中的 CH 建構轉送叢集代理清單(Forwarding Cluster Head List, FCHL)，FCHL 將所有 CH 依序迭代的轉送給網路中的 CH 直到得到空的清單為

止，沿用 DMFT 中 RS 訊息傳遞之方式，並在 CH 收到 FCHL 之後由 CH 通知叢集中的成員，直到 CH 的樹狀路徑建構完成。(b)叢集成員直接與叢集代理連接如同主從式的架構，並由叢集代理負責交換傳輸。

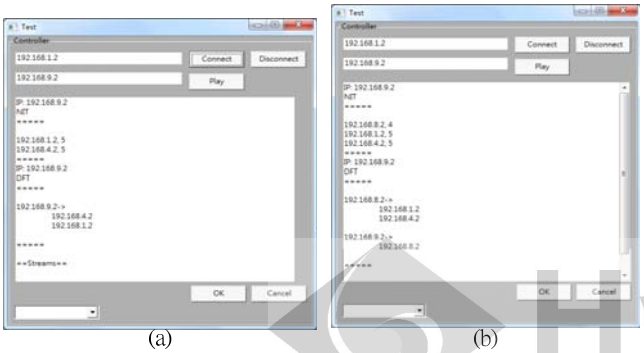
3. 修剪重複路徑：使用 DMFT 相同的修剪方法，但同樣分為對 CH 的樹狀路徑與叢集成員的樹狀路徑兩部份進行修剪。
4. 成員離開恢復：成員離開依離開者的角色分為兩個部份，當離開者是叢集中的成員時，此時使用 DMFT 的方法調整並由 CH 通知其他 CH 離開的事件發生；當離開者是 CH 時，此時必須由 BCH 成為 CH 找到其叢集成員重新建立 CH 的樹狀路徑。
5. 定時回報：如同 DMFT，用以檢查成員是否存在，每個叢集各別檢查，當發現有成員不存在時經由觸發恢復機制。

表 I
DMFT 與 CDMFT 比較表

比較項目	DMFT	CDMFT
空間使用率	僅使用 NIT 與 DFT	使用較多表進行管理。
訊息交換量	多，所有的節點都會重複發送路徑建構的。	中，叢集內會自行交換。
節點恢復速度	快，可以快速的針對特定部位更新。	中，需要更新所有的叢集。
路徑長度	可能會產生串列的樹狀。	使用叢集將樹拆成兩部份，可以降低樹的高度。

五、 實作結果

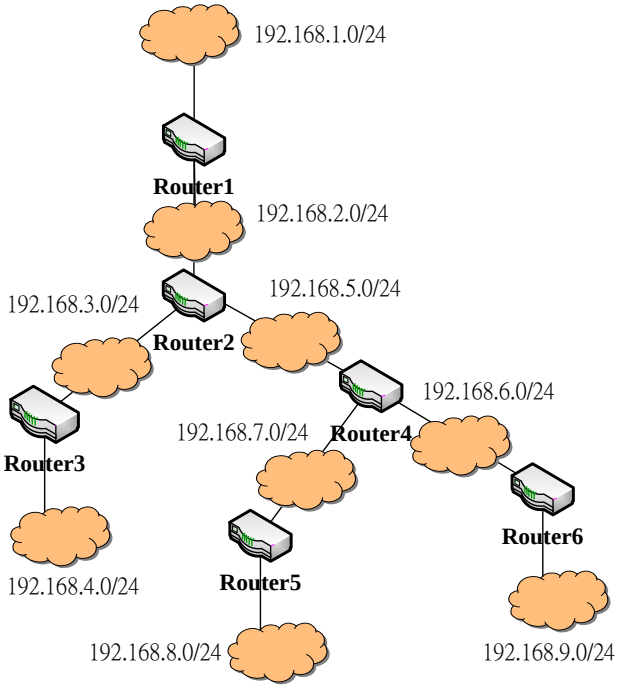
本系統實作加入群組之後可由程式看到路由的樹狀路徑規劃，如圖八所示。



(a) (b)

圖八：路由表顯示

本實作之驗證之測試環境使用 6 個路由器建立 9 個網路區段進行測試，透過本系統進行串流之傳輸，如圖九所示，圖十則為實際系統連接與實驗量測，進行傳輸測試。



圖九：網路拓樸

圖十：實際系統連接

六、結論

本系統實作完成 DMFT 與 CDMFT 兩個 ALM 的演算法並加以修改符合實作上之需求並完成應用層群播之演算法，透過本系統在同一個網路的群組內可以同時有多個不同來源之影音內容的串流，並可由使用者選擇播放來源，為多對多之串流系統，如此一來每個使用者都可以成為串流的來源提供者，將其擁有的資源分享，達到群組間互助合作的優勢。

七、參考文獻

[1] 張文 趙子銘, P2P網路技術原理與C++開發案例，人民郵電出版社，2008。
[2] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, “Scalable application layer multicast,” in Proc. ACM SIGCOMM’02, Pittsburgh, PA, Aug. 2002
[3] 黃家煇，應用層群播路由協定之設計與效能分析，博士論文，台北科技大學 資訊工程所，台北，2013。