

# 非集中式點對點環境下之快速與低頻寬需求的 服務搜尋方法設計\*

## A Fast Searching Method for Low Bandwidth Requirement in Decentralized P2P Networks

陳青文\*\*，朝陽科技大學資訊工程學系

楊朝翔，朝陽科技大學資訊工程學系

顏佩詩，朝陽科技大學資訊工程學系

---

### 摘要

隨著網際網路的發展與普及，在網路上搜尋所需的檔案或服務已經是一個普遍的現象。在廣闊網路環境中尋找所需的服務大都是透過集中式伺服器來取得服務，但集中式伺服器有潛在的缺點，如擴充性、額外的費用等等，但在點對點架構中因不具集中式伺服器所以無上述問題產生。儘管如此，點對點仍有許多問題存在，如提供服務者之所在位置和提供哪些服務等問題。在本篇我們採用非集中式的架構，經由網域名稱伺服器（Domain Name Server）原有的階層式架構，利用此特性來輔助尋找提供服務者，以達到快速搜尋目的。不僅如此，階層式的架構在融入點對點系統後，將能保證提供服務者是距離較近的點（peer）。在實際的模擬中可以確定，在點對點系統中，透過網域名稱伺服器的輔助，在整體的搜尋上，所需執行時間確實有明顯的縮短，對整體網路品質而言，少量封包的傳輸將不會造成網路的擁塞。由此可知，執行時間的縮短和降低大量封包的傳輸將可以讓點對點系統能運作更暢，資料傳輸更快速。

**關鍵詞：**點對點（peer-to-peer, P2P）、網域名稱伺服器（Domain Name Server, DNS）、非集中式點對點網路（decentralized peer-to-peer networks）、服務搜尋（service discovery）

---

\* This research was supported by the National Science Council NSC-92-2213-E-324-006-

\*\* Corresponding author. Tel: +886-4-23323000 Ext. 4534 Fax: +886-4-23742375

## 1. 背景介紹

在廣闊的網際網路中使用者在尋找所需資源，需經由特定集中式伺服器指引才能達成，這種方法稱為主從式（Client-Server）架構。使用主從式架構雖然能在最短時間內取得所需的資訊，但對於伺服器而言是一個極大的負擔；當提供服務者和需求服務者的數量到達某種程度時，將面臨擴充和費用上等等問題。由於目前網際網路中潛在著這些問題，所以當點對點架構被提出時引起廣泛的討論[10]。

現今的網際網路環境中，電腦與電腦之間雖然有 Internet Protocol Address（簡稱為 IP Address）來做區別，但是不經由主從式查詢方式則無法得知提供服務者實際所在位置。然而在點對點系統中，可以不經由集中式伺服器的指引，即可輕鬆且隨意的尋找到所需資源，目前常見架構有集中式和非集中式二種類型。這二種類型各有其優缺點也有許多相關論文做這方面的研究[3, 6, 11, 12]。由此可知，若需在網路環境中快速找到有提供檔案分享或服務的提供者和提供者所在位置都是一件不容易的事；如使用集中式架構，可能需要花大量費用在設備擴充；使用非集中式架構雖不需花費額外費用在設備擴充，但常需要大量時間和空間記錄搜尋的資訊，而且這些記錄資訊是需要時常被更新的（因為提供服務者會加入和離開點對點系統），如此一來將可能形成極大的網路流量造成網路的癱患。

為了改善非集中式常見的缺點，可取其集中式和非集中式的優點結合出新的架構，但仍有技術方面問題有待克服，如提供服務者在哪？提供什麼服務？本篇我們將提出利用現有的網域名稱伺服器架構來輔助點對點的環境。我們提出的架構中，最主要目的是尋找提供服務者在哪和提供什麼服務，在第一部份我們利用網域名稱伺服器的查詢，經由過濾後取得所需資訊即可找出提供服務者在哪，第二部份則是向提供者詢問有無符合的資源或服務。經由以上兩個程序後即可明確得知提供服務者在哪裡和提供什麼服務，在取得所需檔案或服務後即達成任務。

在以下的第二節相關研究中，將探討目前相關研究、使用方法和架構。第三節為研究方法，主要是說明整個研究方向與研究方法。第四節實驗與模擬中，將以實際的模擬來驗證，我們提出的方法確實能在較短時間內完成搜尋且不佔用極大的網路頻寬。第五節為本論文的總結。

## 2. 相關研究

許多學者在探討和定義著點對點計算（peer to peer computing）的問題，這

些議題可以從目前趨勢和相關資訊中發現[1]。點對點計算引起人們的注意，最主要原因在於網路檔案分享系統，包括像是 Freenet[2]、Ohaha[8 和 OceanStore[5] 等等。點對點的架構目前大致可分為集中式、非集中式。在集中式架構中主要的優點是將所需資訊統一管理在伺服器之中，當有需求者向伺服器尋問時，將可以快速取得所需資訊；而主要缺點在於當使用者數量增加到一定程度時，伺服器將面臨擴充性的問題。在非集中式架構中主要優點在於沒有集中式伺服器，在要求服務過程中，可以隨意尋找所需的服務也不需要知道提供者的所在位置；而主要缺點在於需要許多時間來搜尋或是利用儲存體的空間記錄先前搜尋的結果，所以在取得檔案或服務的速度將不理想。所以若能將集中式和非集中式的優點結合在一起，將能克服上述的問題。

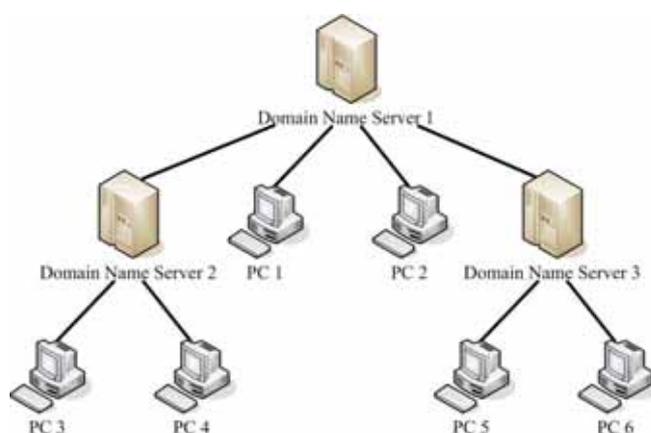
點對點系統中有各式各樣搜尋方法，而這類方法研究議題大都著重於資料交換。在點對點環境中，目前常見的實際應用像是 Napster[7]和 Gnutella[4]，在現今已經變成非常普遍的一種資料交換方式。每一種架構都具有其獨特優點和缺點，雖然如此他們最終目的總是在為提供需求者和要求需求者之間進行資料的傳輸。

在一個單純點對點系統中，不需經由集中式伺服器即可直接的分享檔案或服務。目前多數可用的非集中式點對點系統，實際上效能並不如預期，或者在系統方面會產生過多的額外負載，而且在使用的過程中可能會引導至已損壞或已離線的提供者，因此造成整體效能的大符降低。舉例來說，一個單純點對點系統如 Gnutella，其運作方式首先會完整搜尋所有分散的提供者，在每次查詢後會再傳播給每個提供者，所以使用者搜尋方式為在網路上利用這種洪流式（flooding）查詢方式；使用這類複雜查詢方式，大量的封包傳輸於網路上將可能造成網路的擁塞等不良影響。Napster 的機制雖然可以快速且有效找到可用資源，但嚴格來說不能算是一個點對點系統，因為在這個架構中使用集中式伺服器來儲存可用資源清單和提供服務者所在位置。

對於目前常見的架構而言，透過部份集中式的方式形成混合式的非集中式點對點系統，在效能上大都能優於單純點對點系統。可是，在點對點或單純點對點系統中，對於提供服務者所在位置有變動和所提供之服務有所增減時，將會有變動性和有效性等問題。有鑑於此，常見的問題在於網路傳輸量和點的有效性，在本文中我們提出非集中式的架構，透過網域名稱伺服器的輔助來完成點對點系統。經由網域名稱伺服器的輔助將能在短時間內搜尋完畢，而且也不會有大量封包傳輸於網路中，以下將說明研究方法和完整架構。

### 3. 研究方法

在點對點環境中，若要達到搜尋和取得服務的目的，首先需要知道兩個重要的資訊：提供服務的點在哪裡和提供什麼服務或資源。所以在我們的方法中採用現有的網域名稱伺服器來記錄點的所在位置資訊，對於網域名稱伺服器而言，單純記錄這類簡短的資訊外無需處理其他的資料，所以基本上無需做擴充的考量。其中網域名稱伺服器本身具有網域名稱（domain name）且可對應至實際主機的所在位置（IP Address），所以在我們提出的方法中，將利用現有網域名稱伺服器和它先天快速搜尋的特性，輔助點對點系統以此快速搜尋出有提供服務之點，其架構圖如圖一所示：



圖一 階層式的網路架構

目前現有的網域名稱伺服器為階層式架構，所以在搜尋時更能確保搜尋到的點是距離較近的點。基於網域名稱伺服器具有以上的特質，因此我們在網域名稱伺服器上額外增加一個服務來輔助點對點網路環境。在我們架構中主要分成兩個部份：網域名稱伺服器群組和點的叢集，以下將依序說明各部份的結構。

#### 3.1. 網域名稱伺服器群組：

在這裡，網域名稱伺服器主要目的在於提供一個自動化服務，讓提供服務的點和所分享資源類別依序記錄在網域名稱伺服器之中；每個網域名稱伺服器皆控管著所屬網域中的電腦，以此建構出階層式架構和組成網域名稱伺服器群組。所以當點向網域名稱伺服器查詢提供服務者的所在位置，當所屬網域名稱伺服器查詢完畢後，將會向同一層或其上一層或下一層的網域名稱伺服器來查詢。

由於傳統網域名稱伺服器在新增或刪除網域名稱需由手動方式來做增加或

刪除，所以我們在網域名稱伺服器上面增加了一個服務，以此來自動新增或刪除網域名稱的記錄，我們將此項服務命名為 Register P2P 簡稱為 RP2P。RP2P 目的為當點向網域名稱伺服器提出申請註冊網域名稱（domain name）時，RP2P 會自動向網域名稱伺服器註冊，而當點要離開時則會向網域名稱伺服器取消註冊。為了要區別一般的網域名稱和具有特殊服務的網域名稱，當點向網域名稱伺服器提出申請註冊時，RP2P 會將提出申請的點，先以本機名稱（Host Name）做為命名且在該名稱後註記”\_p2p”（Ex：提出申請的點本機名稱為”pc1”，則註冊的網域名稱即為”pc1\_p2p”）以此來區別；當提供服務的點具有範圍內之服務欲提供時（Ex：Application Software(as), Operation System(op), Music, Other 等），RP2P 會以別名（Alias）的方式註冊在網域名稱伺服器內，別名名稱則是將提供服務的縮寫後註記”\_p2p”（Ex：提供服務的點原先在網域名稱伺服器上註冊名稱為 pc1\_p2p，所以若該點有提供 Application Software 的服務時，RP2P 會在網域名稱伺服器上以別名方式命名為”as\_p2p”，且將別名對應至”pc1\_p2p”）以此來區別。所以在本篇中，網域名稱伺服器群組的實際運作方式如下：

#### (一)網域名稱伺服器收到點的要求註冊或取消註冊

- 收到點的要求註冊
  - i. 以點的本機名稱加上”\_p2p”做為網域名稱來註冊
  - ii. 若點具有範圍內之服務欲提供時，將提供服務名稱之縮寫加上”\_p2p”，接著用別名來註冊網域名稱
  - iii. 切斷與網域名稱伺服器的連線
- 收到點的取消註冊
  - i. 將此點所註冊的網域名稱和所註冊的別名全部刪除
  - ii. 切斷與網域名稱伺服器的連線

#### (二)網域名稱伺服器收到點的要求查詢

- 收到點的要求查詢
  - i. 將所有已註冊的網域名稱回傳
  - ii. 切斷與網域名稱伺服器的連線

### 3.2. 點的叢集

目前在網際網路環境中，對於不同的電腦只能以主機名稱（Host Name）和位置（IP Address）來區別。在點對點的環境中，通常將提供服務者和需求者統

稱為點 (peer)，原因是通常提供服務者也會是需求服務者。所以在本篇中點具有以下三種功能：

- I. 點對網域名稱伺服器要求註冊或取消註冊
- II. 點對網域名稱伺服器查詢現有的網域名稱
- III. 點向有提供服務的點搜尋檔案或服務

為了讓點達到上述的目的，利用我們所撰寫的程式，在這裡簡稱為 P2P\_Client，來輔助點達到此功能。以下為三種功能實際運作原理：

#### (一) 點對網域名稱伺服器要求註冊或取消註冊：

- I. 點加入點對點的環境
  - a. 啟用 P2P\_Client 後，將本機名稱傳送給本機所指定的網域名稱伺服器
  - b. 若有特定的服務提供時，將所要提供的服務選取後傳送給本機所指定的網域名稱伺服器
  - c. 註冊完成後立刻與網域名稱伺服器斷線，以降低對網域名稱伺服器佔有率
  - d. 設定最多可搜尋幾層網域名稱伺服器
- II. 點離開點對點環境時，向網域名稱伺服器取消所有註冊的資訊和服務之後，即可離開點對點環境

#### (二) 點對網域名稱伺服器查詢現有的網域名稱：

- I. 利用網域名稱查詢方式，使用 AXFR[9]參數來取得網域名稱伺服器上所有已註冊主機清單
- II. 將清單中具有”\_p2p”的該行存放在本機的暫存檔中依序向其他的網域名稱伺服器詢問並重覆前一步驟，直到符合先前設定的搜尋層數

#### (三) 點向有提供服務的點搜尋檔案或服務：

- I. 從本機已過濾並存放在暫存檔中的點依序詢問是否有所需的檔案或服務
- II. 若有，則要求檔案，若沒有回到前一步驟

由此可知，當每個點啟動 P2P\_Client 後會自動向 DNS 註冊，如此一來，有需求的使用者在尋找某個檔案或服務時，即可快速查詢出有效的點。其中，在設定層數的目的在於讓使用者自行決定搜尋的範圍，因為當層數越大所代表的是範圍越大和距離越遠，若無此設定將可能發生浪費太多時間與搜尋較無效率的點。

透過網域名稱伺服器的輔助，利用此架構我們可以在任何時間、任何地點快速的取得周遭附近的有用資源，不會如同大海撈針，也不會取出已失效或不實際的點來向他要求檔案或服務。下面的章節，我們將以實際的模擬來與直覺式窮舉法來做比較。

## 4. 實驗與模擬：

我們模擬的環境是由三台網域名稱伺服器和六台電腦架設在高速網路（Fast Ethernet）上如圖一所示，其中三台網域名稱伺服器分別控管一個 C 級網段的 IP Address（192.168.X.1~192.168.X.254），每台網域名稱伺服器控管著兩台電腦。

實際模擬部份我們將分別測試：一個 C 級網段（由一台網域名稱伺服器和二台電腦所組成）、二個 C 級網段（由二台網域名稱伺服器和四台電腦所組成）和三個 C 級網段（由三台網域名稱伺服器和六台電腦所組成）。在這三種環境中搜尋有提供服務的點，我們將測量所需花費間和網路上傳量和下載量。所使用測試方法將採用直覺式窮舉法來與我們所提出方法做比較；在窮舉法方面，將以不同線程（Thread）個數分別做搜尋測試；線程個數我們分別以 20, 40, 50, 100, 150, 200 和 254 個線程依序測試；在這個三種模擬環境中，在同一時間只有一台電腦對整個模擬環境做搜尋，以下我們在 4.1, 4.2, 4.3 段落中將會以三種不同環境所測得的數據和圖表顯示出來；在 4.4 的實驗結果與討論中將會探討 4.1, 4.2, 4.3 中所測量出的數據與現象。在以下的段落中，所使用的代號意義分別如下：

T254：窮舉法搜尋使用 254 個 Thread，T200：窮舉法搜尋使用 200 個 Thread

T150：窮舉法搜尋使用 150 個 Thread，T100：窮舉法搜尋使用 100 個 Thread

T50：窮舉法搜尋使用 50 個 Thread，T40：窮舉法搜尋使用 40 個 Thread

T20：窮舉法搜尋使用 20 個 Thread，DNS Query：網域名稱伺服器輔助搜尋

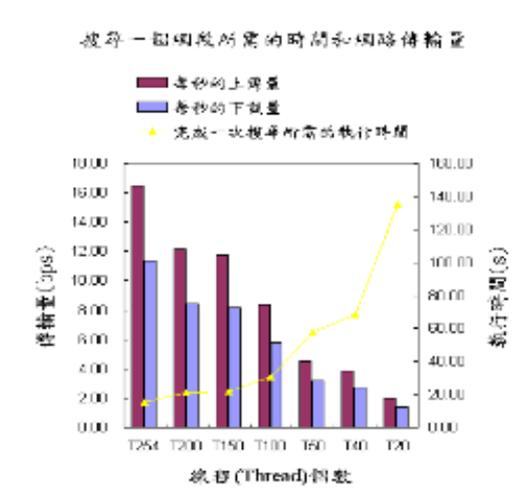
### 4.1. 搜尋一個 C 級網段範圍內的可用點：

在這次模擬中共有三台主機分別為一台網域名稱伺服器命名為 DNS1，二台電腦分別命名為 PC1 和 PC2，我們定義這一個 C 級網段的範圍為：192.168.0.1~192.168.0.254。在這裡，我們將測試在單一個 C 級網段中共有 254 個 IP Address 其中有二台電腦提供服務，對於這類環境搜尋完成一次所需時間和網路傳輸量。第一部份將測量使用窮舉法以不同線程個數其間差異性，測量結果如表一和圖二所示。由上述結果可知，當使用窮舉法來搜尋一個 C 級網段時，若以時間為考量需在最短時間內完成一次搜尋則所需平均成本為：執行時間為 15.5 秒，每秒所需上傳量為 16.47bps，每秒所需下載量為 11.33bps；若以網路傳輸量為考量，

希望暫用最少網路頻寬則所需成本為：執行時間為 135.10 秒，每秒所需上傳量為 2.00bps，每秒所需下載量為 1.42bps。第二部份將使用我們所提出方法來測量在這個環境搜尋一次所需付出的平均成本為：執行時間為 34.92 毫秒(ms)，總上傳量為 11.2Kbit，總下傳量為 2.4Kbit。

表一 窮舉法搜尋一個 C 級網段

搜尋一個 C 級網段所需的時間和網路傳輸量			
	每秒上傳量 (bps)	每秒下載量 (bps)	執行時間 (s)
T254	16.47	11.33	15.50
T200	12.16	8.44	21.20
T150	11.79	8.15	22.00
T100	8.38	5.79	30.90
T50	4.54	3.18	58.00
T40	3.87	2.72	68.40
T20	2.00	1.42	135.10



圖二 窮舉法搜尋一個 C 級網段

## 4.2. 搜尋二個 C 級網段範圍內的可用點：

在此次模擬中共有六台主機分別為二台網域名稱伺服器命名為 DNS1 和 DNS2，四台電腦命名為 PC1、PC2、PC3 和 PC4，我們定義這二個 C 級網段的範圍分別為：192.168.0.1~192.168.0.254 和 192.168.1.1~192.168.1.254。在第二次的模擬中，我們要測試在範圍為二個 C 級網段，其中共有 254\*2 個 IP Address，其中四台電腦為有提供服務的點，對於這類環境我們將要測量完成一次的搜尋所需花的時間和網路傳輸量。第一部份我們一樣先測量使用窮舉法以不同線程個數其間的差異性，其測量結果如表二和圖三所示。在第二部份的測量，使用我們所提出的方法運用在搜尋二個 C 級網段而且為階層式的架構中，所需的平均執行時間、網路的流量，結果如下所示。行時間為 35.01 毫秒 (ms)，總上傳量為 16.80Kbit，總下傳量為 3.20Kbit。由上述的結果與 4.1 比較可以得知對於不同的範圍利用窮舉法完成一次搜尋，所造成的差異有多少，在這裡我們用成長率來做時間和流量的評估。在本篇中成長率定義為：

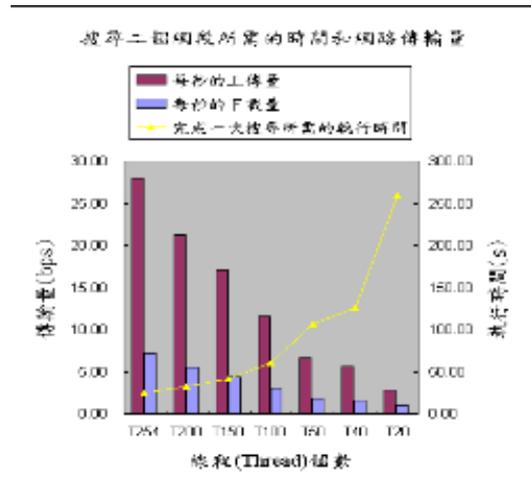
$$\text{執行時間成長率} = \frac{\text{搜尋一個網段以上所需執行時間}}{\text{搜尋一個網段所需執行時間}}, \text{路流量成長率} = \frac{\text{搜尋一個網段以上所需網路流量}}{\text{搜尋一個網段所需網路流量}}$$

對於成長率而言，當計算出的成長率為 1 時所表示意義為在這二次搜尋的過程中

，所付出之搜尋成本是相同的；成長率為 2 時表示所需之成本為原來的二倍，成長率為 3 時表示所需之成本為原來的三倍成長以此類推。所以針對以上二種 C 級網段的模擬（一次搜尋一個 C 級網段和一次搜尋二個 C 級網段），所得到的成長率為表三所示。由表三可以發現，當搜尋範圍當增加一倍時，窮舉法執行時間成長率為 1.55~1.95；對於網路流量而言上傳量成長率為 1.39~1.75，下載量成長率為 0.52~0.71。對於我們所提出的方法而言，執行時間成長率為 1，上傳量成長率為 1.5，下載量成長率為 1.33。

表二 窮舉法搜尋二個 C 級網段

搜尋二個 C 級網段所需的時間和網路傳輸量			
	每秒上傳量 (bps)	每秒下載量 (bps)	執行時間 (s)
T254	28.00	7.22	25.00
T200	21.28	5.49	32.90
T150	17.03	4.39	41.10
T100	11.62	2.99	60.30
T50	6.59	1.76	107.00
T40	5.67	1.61	126.00
T20	2.85	1.01	259.00



圖三 窮舉法搜尋二個 C 級網段

表三 執行時間和網路流量的成長率

完成一次搜尋所需成本之成長率								
	T254	T200	T150	T100	T50	T40	T20	DNS Query
上傳量	1.70	1.75	1.44	1.39	1.45	1.47	1.43	1.50
下載量	0.64	0.65	0.54	0.52	0.55	0.59	0.71	1.33
執行時間	1.61	1.55	1.87	1.95	1.84	1.84	1.92	1.00

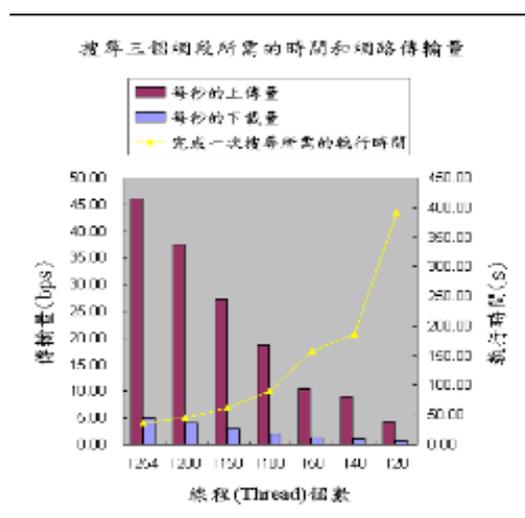
### 4.3. 在三個 Class C 的 IP Address :

在這個環境中共有九台主機分別為三台網域名稱伺服器命名為 DNS1、DNS2、DNS3，六台電腦命名為 PC1、PC2、PC3、PC4、PC5 和 PC6，我們定義這三個 C 級網段的範圍為：192.168.0.1~192.168.0.254，192.168.1.1~192.168.1.254，192.168.2.1~192.168.2.254。在第三次的模擬中，我們要測試範圍為三個 C 級網段，其中共有 254\*3 個 IP Address，而且經由三台網域名稱的組成階層式的架構，其中的六台電腦為有提供服務的點，在這個環境中將要測量完成一次的搜尋所需的時間和網路傳輸量。第一部份先測量使用窮舉法以不同線程個數其間的差

異性，其測量結果如表四和圖四所示。在第二部份的測量，使用我們所提出的方法運用在搜尋三個 C 級網段而且為階層式的架構中，所需的平均執行時間、網路的流量如下所示：執行時間為 37.01 毫秒 (ms)，總上傳量為 23.20Kbit，總下載量為 4.80Kbit。

表四 窮舉法搜尋三個 C 級網段

搜尋三個 C 級網段所需的時間和網路傳輸量			
	每秒上傳量 (bps)	每秒下載量 (bps)	執行時間 (s)
T254	46.02	5.03	36.80
T200	37.55	4.10	45.10
T150	27.27	2.98	62.10
T100	18.59	2.03	91.10
T50	10.39	1.23	157.70
T40	8.88	1.13	185.90
T20	4.31	0.81	391.00



圖四 窮舉法搜尋三個 C 級網段

由上述搜尋三個 C 級網段的結果與 4.1 搜尋一個 C 級網段比較可以發現，當搜尋的範圍大二倍時執行時間和網路流量成長率如表五所示。所以從表五可以得知當搜尋範圍當增加二倍時，窮舉法執行時間成長率為 2.13~2.95；對於網路流量而言上傳量成長率 2.16~3.09，下載量成長率 0.35~0.57。而我們所提出的方法，執行時間成長率為 1.06，上傳量成長率為 2.07，下載量成長率為 2。

表五 執行時間和網路流量的成長率

完成一次搜尋所需成本之成長率								
	T254	T200	T150	T100	T50	T40	T20	DNS Query
上傳量	2.79	3.09	2.31	2.22	2.29	2.30	2.16	2.07
下載量	0.44	0.49	0.37	0.35	0.39	0.42	0.57	2.00
執行時間	2.37	2.13	2.82	2.95	2.72	2.72	2.89	1.06

#### 4.4. 實驗結果與討論：

在 4.1、4.2、4.3 中可以明顯的得知，利用窮舉法來搜尋，對於執行時間和所需上傳量而言，最差的情況將近似倍數成長，而使用我們所提出的方式其執行時間的成長率微乎其微，雖然網路傳輸量約成長一半，但實際上在搜尋過程中，若需在最短的時間內完成一次搜尋(範圍為一個 C 級網段)的總傳輸量為 13.6Kbit

，所以最差前情況下若總傳輸量成長一半，在實際的使用環境為一個 B 級的網段（共 254 個 C 級的網段）又有數以千計的使用者同時使用時，對整個網路運作而言不會有不良的影響，因為完成一次搜尋所需時間僅需數十毫秒；但就窮舉法而言，完成一次搜尋（範圍為一個 C 級網段）的瞬間網路流量為 27.8bps（上傳量加下傳量），而實際使用的環境為一個 B 級的網段又有數以千計的使用者同時使用時，以高速網路而言對於整個網路的瞬間總流量，將有可能佔用 7/10 的網路頻寬例一。

例一：使用窮舉法計算一個 B 級網段同時有一千人做查詢時的網路流量

$$\begin{aligned}\text{瞬間流量} &= 27.8 \text{ (bps)} * 254 \text{ (C 級網段數)} * 1000 \\ &= 7061.2 \text{ Kbps}\end{aligned}$$

在這裡我們可以發現，使用查詢方式測量單位是以總流量而非瞬間流量，因為對於查詢而言，完成一次的搜尋（範圍為一個 C 級網段）只需約 35ms，而瞬間流量的單位確是以秒為基底，因此不適用而改用總流量來評估。雖然看似可能佔用部份網路頻寬，但實際上當網段成倍數成長時，執行時間成長率趨近於 1，所以網段的大小對執行時間影響不大。由此可知查詢法完成一次搜尋在 1 秒鐘內即可完成，所以當以 1 秒鐘為單位時間時，一個單位時間內即可完成搜尋數千人次的搜尋，而窮舉法則需以數千或數萬以上的單位時間才能完成。由上述推論可以證實，使用窮舉法來搜尋將會長時間佔用大量網路頻寬，因此造成網路擁塞。

由以上的實驗數據可以很明確的了解到，在網際網路中要尋找提供服務者並不容易，當我們想要尋找提供服務者時，直覺的想法就是一個一個問也就是所謂的窮舉法，而這種方式從上面的數據可以證實，速度慢且會佔用極大的網路頻寬而形成網路的擁塞。雖然使用網域名稱伺服器來輔助點對點環境時和窮舉法一樣是屬於非集中式架構，但利用額外技術來架構出具有階層式的型態，不僅在執行時間和網路傳輸上佔有優勢而且更進一步的確保所尋找到的點必定是距離較近，就距離而言將有助於檔案傳輸或服務的提供能更快速完成。如此一來能快速完成使用者所提出的需求，更能降低網路頻寬的佔有率和碰撞等問題。

## 5. 結論：

在本文中，我們利用網域名稱伺服器的輔助實際建置一個點對點網路環境，同時我們也與窮舉法來作比較與測量。經由分析後的數據我們可以確定，透過網域名稱伺服器的輔助不僅不需考慮使用者增加時伺服器擴充性的問題，也不需考慮搜尋所需花費的時間，因為完成一次搜尋在單位時間內即可完成；在網路品質方面，向網域名稱伺服器查詢後回傳的結果，所需佔用的頻寬更是微乎其微，亦

不會造成網路的擁塞。目前網際網路傳輸協定為 IPv4，但在不久的將來 IPv6 將會成為主流，所以可用 IP 位址數量將會極速成長。雖然如此，我們所提出的架構在此環境依然適用，而且階層式架構更能保證所搜尋到的點是距離較近，所以基於上述之特性以此方法建構出點對點環境，將對此系統有莫大的幫助。

## 6. 參考文獻

1. B. Yang, H. Garcia-Molina, Efficient Search in Peer-to-Peer Networks. Proc. Int. Conf. On Distributed Computing Systems, 2002.
2. I. Clarke, et al., "Freenet: A Distributed Anonymous Information Storage and Retrieval System," in Designing Privacy Enhancing Technologies, Lecture Notes in Computer Science 2009, H. Federrath, ed., Springer-Verlag, Berlin, 2001, pp. 46-66.
3. G. Moro, A. M. Ouksel & C. Sartori, "Agents and Peer-to-Peer Computing: A Promising Combination of Paradigms", International Workshop on Agents and Peer-to-Peer Computing (AP2PC 2002), LNAI 2530, pp. 1-14, 2003.
4. Gnutella homepage, <http://www.gnutella.wego.com>.
5. John Kubiawicz, "OceanStore: An Architecture for Global-Scale Persistent Storage", ACM Cambridge, Massachusetts, United States, pp. 190-201, Nov 2000.
6. Munindar, P.S., "Peering at Peer-to-Peer Computing", IEEE Internet Computing 2001, Jan/Feb 2001.
7. Napster, <http://www.napster.com>.
8. Ohaha. <http://www.ohaha.com/design.html>.
9. P. Mockapetris, "Domain names-Implementation and specification", Internet Request for Comments (RFC 1035), February 1987.
10. Peer-to-peer working group, <http://www.peer-to-peerwg.org/>.
11. Schollmeier, R., "A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications", IEEE Conference Peer-to-Peer 2001, pp. 27-39, August 2001.
12. Simon.S., "Peer-to-Peer Network Management in an IBM SNA Network", IEEE Network Magazine, Vol. 5, pp. 30-34, March 1991.