

# 非結構化對等式網路搜尋效率改進之研究

紀宗信 廖耕億

長庚大學資訊管理研究所

m9344010@stmail.cgu.edu.tw, gyliao@mail.cgu.edu.tw

## 摘要

近年來，對等式(Peer-to-Peer, P2P)網路的普遍應用，已經吸引了許多使用者的參與。檔案分享是 P2P 網路相當普遍的應用，而 Gnutella 則是其中一種常見的 P2P 協定。在檔案搜尋時，因為 Gnutella 採用廣播式的搜尋方法，所以會對整體網路產生非常大量的訊息洪流。儘管已有許多的文獻提出不同的改善方法，但分析其方法後，我們發現這些方法不是需要變更非結構化 P2P 網路的架構，就是需要事先將檔案或是檔案資源列表存放在其他的節點上，因此也形成了系統負荷的另一來源。在本研究中，我們利用索引機制改善非結構化 P2P 網路的搜尋效率。經過模擬實驗的驗證，索引機制在搜尋效率方面的確能夠產生較少的網路流量，所需付出的代價，則是少許的儲存成本；在儲存媒體成本大幅降低的今日，相信本研究的成果具有實務上的應用價值。

**關鍵詞：**非結構化對等式網路、搜尋效率、索引

## 1. 前言及相關文獻探討

近年來，拜電腦硬體價格不斷下降、連網環境日趨完善，使得網際網路上的使用者近年來有著快速的成長，也使得網際網路上的應用愈來愈多元化，其中最蓬勃發展的一塊就是對等式(Peer-to-Peer, P2P)網路的應用。

在結構化的 P2P 網路中，最大的特徵為在此網路中各節點所分享的物件皆有一唯一的識別碼，此識別碼通常是將欲分享的物件透過一雜湊函數的運算後，賦予此物件一獨一無二的 m-bit 的雜湊值。而在此網路中的各個節點皆會負責所有 m-bit 雜湊值中某一段雜湊值所對應的物件的儲存、查詢回應等動作。而此種的網路架構亦稱為分散式雜湊表(DHT, Distributed Hash Table)P2P 網路[4]。此特殊的網路架構具有一定的虛擬網路架構，可以保證使用者在固定的查詢步驟內找到所需求的物件。目前採用此架構的協定有 Chord[5]、CAN[10]以及 Pastry[2]等。

儘管結構化的 P2P 網路提供較佳的運作效能，但是卻存在著以下的缺點[13]：

1. 結構化 P2P 為了在資源搜尋時有較佳的表現，因此必須維持特殊的網路拓撲。而在節點來來去去的情形頻繁時，此部分需付

出大量的網路流量成本。

2. 結構化 P2P 網路較不適合使用關鍵字搜尋。

基於以上兩點缺陷，雖然結構化 P2P 網路在運作上有較佳的表現，但是目前使用較普遍的仍然是非結構化 P2P 網路。

相較於結構化 P2P 網路，在非結構化 P2P 網路上各節點所分享的物件並無一唯一的 m-bit 雜湊值所對應，各個節點所分享的物件皆由所分享的節點負責物件的儲存、查詢回應等動作。在此架構下的使用者並無法得知所需求資源的正確位置，因此，當使用者做資源搜尋時，通常透過關鍵字搜尋，以廣播方式對相鄰節點傳送「資源要求」的訊息，其相鄰節點會檢視是否有需求節點所要求的資源。若有，則循原路徑回傳給需求節點，若無，則繼續傳送給自己的相鄰節點。如果該資源存在於此網路中的話，則需求節點必定能找到想要的資源；但是如此一來可能會產生大量的訊息洪流（以下稱為 flooding），對整體網路頻寬有著極嚴重的影響。為了降低此種機制所產生的訊息洪流，目前大都採用 TCP/IP 中類似 TTL(Time To Live)的做法來改進此一缺點；但在這種改進機制下，卻有可能無法找到所需求的資源。目前採用此架構的協定有 Gnutella[12]，Limewire[3]，Kazaa[1]等。

而在研讀相關改善 Gnutella 搜尋機制的文獻中，我們發現 YAPPERS[9]及 Percolation Search[11]是比較有效方法。

YAPPERS 在運作上，引用了結構化 P2P 網路 DHT 的概念，但與 DHT 不同的是，網路中的節點及被分享的資源並非被賦予一個一對一對應的雜湊值，而是將整個網路中的資源及節點分成特定幾個類別，為了解釋方便起見，假設只有白色及灰色兩種類別。因此，在做分類時，一可能的做法是將此資源或節點的編號做雜湊運算後再取 2 的模數。為了達到在搜尋時能有較好的效能表現及減少訊息洪流的現象，YAPPERS 中的每個節點在貢獻出欲提供的資源時，會先檢視此資源是屬於白色資源或是灰色資源；若分享此資源的節點屬於白色節點，且所享的資源亦是白色資源，則由本身負責；反之，若所分享的資源屬於灰色資源，則交由鄰近的灰色節點負責。而因為有對資源及節點做分類，因此，當一節點在搜尋時，若所欲搜尋的資源屬於白色資源，則此資源請求的訊息只會在同樣是白色的節點中被傳送；反之，若所欲搜尋的資源屬於灰色資源，則此資源請求的訊息只會在同樣是灰色的節點中被傳送。而為了達到上述搜尋機制的效果，

每個節點必須知道其附近一定範圍內節點的分佈狀況。對於一節點 A 來說，A 會將其附近節點分成「鄰近節點—IN(A)」、「邊境節點—F(A)」以及「遠方節點—EN(A)」。

IN(A)：在 A 節點 h hop 內的所有節點，包含 A 節點本身。

F(A)：若 N(v) 為與節點 v 直接相鄰的所有節點，則

$$F(A) = \bigcup_{v \in IN(A)} N(v) - IN(A)$$

EN(A)：F(A) 的鄰近節點。

$$EN(A) = \bigcup_{v \in F(A)} IN(v)$$

當 A 有一資源想在網路上分享時，會先決定此資源屬於那種顏色類別，然後將此資源存放在 IN(A) 中與此資源屬於相同顏色的一個節點上。而在搜尋時，A 會先找出一個在 IN(A) 中與欲搜尋資源屬於相同顏色的節點，在此假設為節點 B，待 B 接受到此資源請求時，會先傳回本身的查詢結果，之後，再將此資源請求傳送給 IN(B) ∪ F(B) 中與欲搜尋資源屬於相同顏色的節點。

Percolation Search 是運作在具有 Power Law 分佈特性的 Gnutella 網路上。其在改善搜尋效能的作法為以下三個步驟：

1. Content List Implantation：每個節點透過隨機漫步的方式，將自己所能分享的資源清單存放在隨機漫步過程中所經過的節點上。
2. Query Implantation：一節點在有資源需求時，會先透過隨機漫步的方式，將查詢條件存放在隨機漫步過程中所經過的節點上。
3. Bond Percolation：搜尋時，在步驟二中擁有搜尋條件的節點會使用隨機廣播搜尋的方法，將此搜尋條件向其鄰近節點廣播傳遞，但在傳遞給其鄰居節點前，會先計算一傳遞機率，只有在傳遞機率大於可傳遞的門檻值時，才會繼續傳遞下去。

在 Power Law 的網路中，以上的方法，經過分析推論，可以保證：

1. 在上述步驟一時，每個資源至少會被一有著高度連結的節點所得知，因此，
2. 在步驟二，至少會有一個有著高度連結的節點幫忙搜尋。

因此，在搜尋的結果上，如果欲搜尋的資源存在網路中，此機制可以找到至少一個擁有此資源的節點，相較於原始 Gnutella 的搜尋機制，減少非常可觀的訊息洪流。

## 2. 需求定義

根據研究[6]指出，目前以 Gnutella 協定為主的非結構化 P2P 網路上，若一個節點在檔案搜尋時，將 TTL 設為 8，且在訊息傳遞的過程中，每個

節點平均將此訊息傳送給其他的 8 個節點，在這種情況下，找到一個資源所需的耗費的流量為 1.2GB，由此可看出以 flooding 方式在檔案搜尋時所造成訊息洪流問題的嚴重性。因此，本論文的目的在於降低目前以 Gnutella 協定為主的非結構化 P2P 網路上，在檔案搜尋時所產生的訊息洪流現象。

而觀察目前在改進此一問題上最常見的做法，為節點在搜尋前，必須將自己所能分享的資源或是可供分享資源清單存放在其它節點上，但在這種做法下，若是節點提供新資源的頻率較高，或是在提供巨量的資源時，在將分享的資源或是可供分享資源清單存放在其他節點上時，仍會產生大量的網路流量。而因為此做法是為了達到在資源搜尋時有較佳的效能表現，因此本研究認為此前置步驟所產生的流量也應納入在衡量搜尋效能表現時的因子之一。

為了產生相對於原始 Gnutella 或是其他學者的方法有較佳的搜尋效能，本研究將問題分析為下列三點需求：

1. 對現行的非結構化對等式網路架構不做任何的變動。
2. 在搜尋前，節點不需將自己所能分享的資源或是可供分享資源清單存放在其他節點上。
3. 在搜尋時，透過相對少數節點的幫忙即可找到所需求的資源；此點也意味著，在搜尋時能減少一定程度的整體網路流量。

## 3. 方法設計—Index

目前 Gnutella 在搜尋機制的運作上如圖 1 所示：

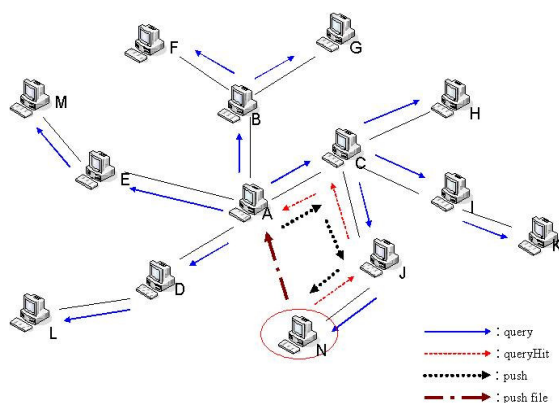


圖 1 Gnutella 搜尋機制(資料來源：本研究)

節點 A 向其鄰近節點 B、C、D、E 發出資源請求訊息(query)，B、C、D、E 會在接收到此資源請求訊息後，會先檢查本身是否有符合的資源，若無，則 B、C、D、E 會繼續向其鄰近節點轉送此 query 訊息；若在此 query 的傳遞過程中有某一節點有符合 query 訊息所請求的資源，在此例中為節點 N，

N 會循原 query 路徑回應 queryHit 給 A，而後，A 再依原 query、queryHit 的傳送路徑傳送 push 給 N，要求傳遞所需求的資源，待 N 收到後，即會傳送 A 所需的資源給 A。

本研究假定每一節點皆會貢獻自己的儲存空間來記錄曾傳遞過的 push 訊息，而此儲存空間所記錄的訊息將做為之後幫助其他節點搜尋時的索引，因此，本研究將此儲存空間稱為 index。

本研究所設計用來改善搜尋機制的作法如圖 2(承接圖 1)所示，讓幫忙傳送 push 的節點，此例中為節點 C 及節點 J，記錄 A 的連線資訊(IP、埠號)以及 A 將得到的資源。

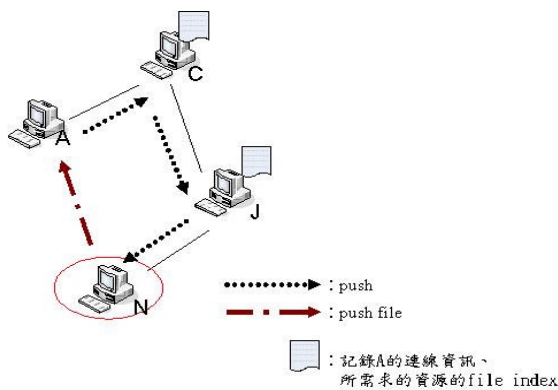


圖 2 本研究 Index 機制(資料來源：本研究)

因此，如圖 3(承接圖 2)，若之後節點 O 有與之前節點 A 有同樣的資源請求時，在傳送 query 給其鄰居節點 J 時，J 會先檢查此 query 的搜尋條件是否存在於自己的 Index 中，因為 J 此時有 A 的連線資訊，因此，J 不會將此 query 廣播給其鄰居節點，而是直接將此 query 訊息傳送給 A，若 A 在之前有得到此新資源，且願意將此資源貢獻給其他的節點，則 O 即可得到此資源。

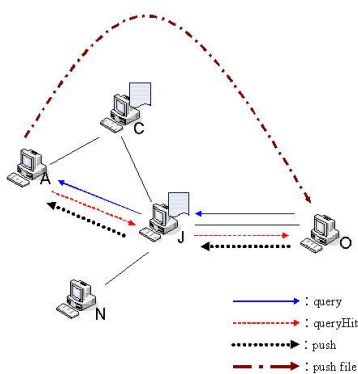


圖 3 Index 機制下的搜尋示意圖(資料來源：本研究)

#### 4. 實驗設計

本研究將以 Java 技術模擬基於 Gnutella 協定的非結構化 P2P 網路環境，讓實驗的搜尋評估機制

得以在較可控制的範圍下，進行模擬實驗。

而在搜尋機制的比較上，本研究的模擬實驗中，將讓每個節點具有以下四種搜尋功能：

1. 原始 Gnutella 協定中的搜尋功能。
2. YAPPERS 機制中的搜尋功能。
3. Percolation Search 的搜尋功能。
4. 本研究 Index 機制的搜尋功能。

在 P2P 網路的拓撲設計上，根據研究[7][11]指出，Gnutella 所形成的網路拓撲具有 Power Law 分佈的特性，因此，本研究在節點的連結配置上將參考點。而在節點的初始檔案分佈上，並無文獻指出 P2P 網路初始的檔案分佈為何，因此，本研究假定初始時檔案的分佈也是符合 Power Law 的分佈。

在節點數量方面，因為受限於可取得的 API 緣故，本研究目前的模擬實驗中只模擬出 100 個節點，但此 100 個節點的連結數分配符合 Power Law。

在資源數量以及分佈方面，為了使資源的分佈也能符合 Power Law 的分配，因此，系統中的不同資源數為 200 個。

模擬實驗中一次 trial 的實驗步驟為：

1. 隨機挑選一節點發出資源請求。
2. 執行資源請求的節點依四種不同的搜尋機制發出相同的資源請求。
3. 在每種搜尋機制結束後，收集不同收搜尋機制所產生的相關數據。

一次模擬實驗會在所有的 peer 得到系統中所有的檔案後結束。圖 4 為模擬實驗的執行步驟

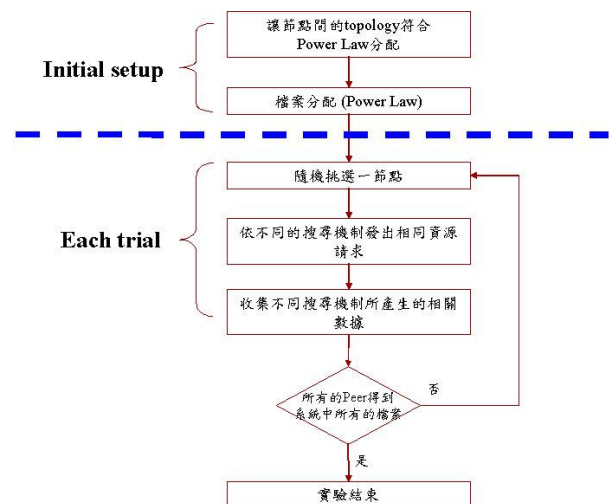


圖 4 模擬實驗執行步驟(資料來源：本研究)

實驗後所要探討的數據將有以下幾項：

1. Coverage：搜尋時，所經過的節點數/總節點數。
2. Query Hits：搜尋後所找到可提供所需求檔案的 peer 數。
3. Query Message：搜尋時所產生的流量。
4. Success Rate：搜尋成功的機率 = 成功的次數/總次數。

5. Success Rate Coverage: 搜尋覆盖率本益比 = 每 1% 的 Coverage 可以貢獻多少百分比的 Success Rate
6. Min. Response Time: 搜尋到第一個可提供檔案的 peer 所需經過的最短 hop 數。
7. 節點回應檔案的平均次數。
8. 節點儲存空間的負荷量: YAPPERS 中, 為節點 insert 到 proxy node 的資源數; Percolation 中, 為節點放在其他節點的 content list; Index 中, 為節點儲存的 index 數。

Percolation Index	33.69 2.55%
-------------------	----------------

由以上的結果可看出, Percolation、Index 機制在搜尋所產生的訊息量方面的確有很大的改進。而 YAPPERS 的流量之所以會超過 Gnutella, 是因為我們將 YAPPERS 的節點在把所分享的資源存放在其它節點上時所產生的訊息量加上的緣故。

各機制 Coverage 與 Success Rate 之間的關係。圖 7、圖 8 為實驗結果。

## 5. 實驗分析

在 Percolation threshold 為 0.3, YAPPERS 分類數為 3 的情況下, 四種搜尋機制在訊息流量方面的比較, 圖 5、圖 6 為實驗結果。

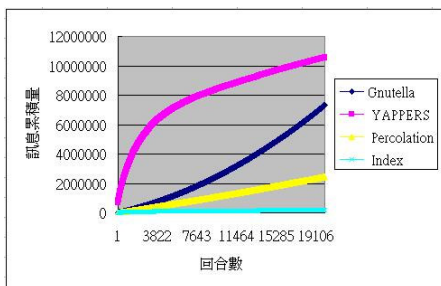


圖 5 各搜尋機制搜尋訊息累積量比較圖之一 (資料來源: 本研究)

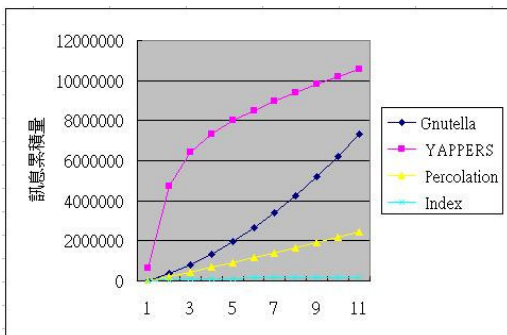


圖 6 各搜尋機制搜尋訊息累積量比較圖之二 (資料來源: 本研究)

圖 6 是從圖 5 中截取十一個點所畫出來的圖。若將 Gnutella 模擬結束後所產生的訊息累積量訂為 100%, 從其它三個搜尋機制的訊息累積量如下表 1 所示。

表 1 各搜尋機制搜尋訊息累積量比較表 (資料來源: 本研究)

	訊息累積量
Gnutella	100%
YAPPERS	145.3%

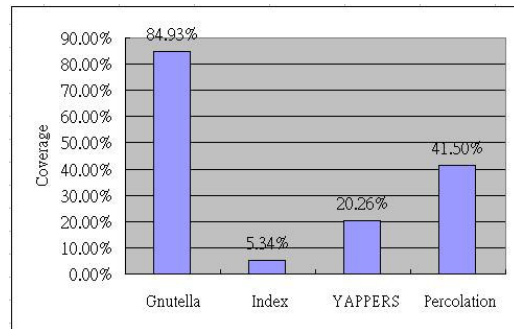


圖 7 各搜尋機制 Coverage 比較圖 (資料來源: 本研究)

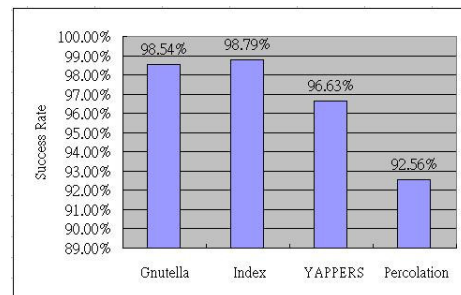


圖 8 各搜尋機制 Success Rate 比較圖 (資料來源: 本研究)

由以上結果可看出, Index 僅需要經過系統中 5% 左右的節點, 就可達到高達 98% 的搜尋成功率。而各個搜尋機制每 1% 的 Coverage 所貢獻的 Success Rate 如下圖 9 所示。由此可看出 Index 的本益比是相當高的, 而 YAPPERS、Percolation 在此方面也遠優於 Gnutella。

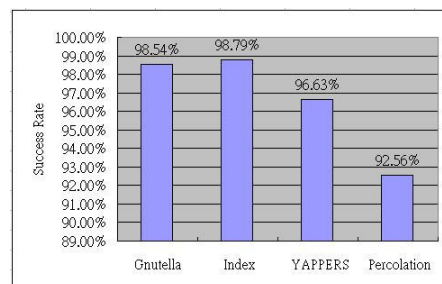


圖 9 搜尋機制 SuccessRate/Coverage 比較圖 (資料來源: 本研究)

由於 Gnutella 的搜尋機制所採取的是全系統的 flooding 搜尋，所以搜尋的 Success Rate 高於 YAPPERS、Percolation 是可預期的，但為何本研究的 Index 的機制還能略高於 Gnutella 呢？分析原因，即在於本研究的 Index 方法中，每個節點都有一份 index，在 TTL=0 時，若 index 中有符合的記錄存在，會比 Gnutella 多傳送一個節點，且此節點可提供被要求的檔案。

一節點每一次成功的搜尋平均可得到幾個 query hit，下圖 10 為實驗的結果。

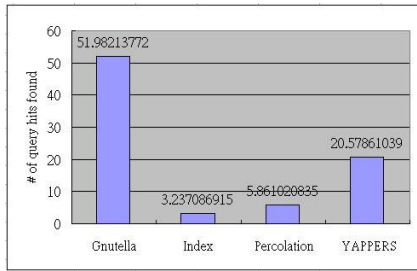


圖 10 各搜尋機制 # of query hits found 比較圖之一(資料來源：本研究)

由圖 10 可觀察到，不意外的，Gnutella 因為所使用的是 flooding 的搜尋，所以所得到的 query hit 是最多的。而在本研究的 Index 機制中，節點若在 index 中發現符合的記錄，即不會繼續往下傳，故所得到的 query hit 是最少的。

在設計 Index 機制後，我們曾提假設：Index 機制可減少熱點(hop spot)現象，亦即，大部分的 queryHit 不會由少數的節點所回應。圖 11 為實驗結果。

由圖 11 可看出，Index 機制在減少熱點現象方面的確有很大的改進。減少熱點現象，所代表的意義在於系統中各個節點的貢獻是較為平均的，不會造成某些節點因為必須負擔大量的回應而不願繼續待在 P2P 網路上。

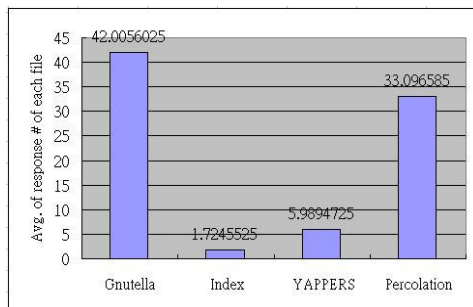


圖 11 各搜尋機制各節點對檔案回應平均次數比較圖(資料來源：本研究)

在 YAPPERS、Percolation，以及本研究的 Index 機制中，為了達到較好的搜尋效率，各個節點所付

出的儲存容量負擔。圖 12 為實驗結果。

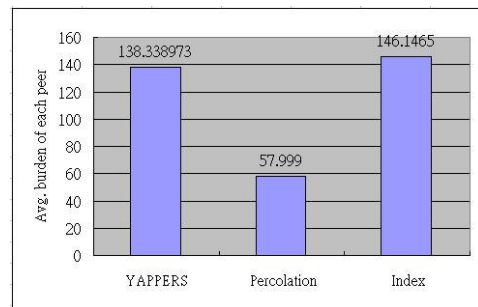


圖 12 各搜尋機制節點額外儲存空間負擔比較圖(資料來源：本研究)

在圖 12 中，若 YAPPERS 採用會使得搜尋所產生的流量最少的改進方法，平均只會有 17 個節點擔任 Proxy node 的角色，但每個角色所需額外負擔的儲存容量單位為 138 個左右，其儲存容量大小為此 138 個單位再乘上其平均大小，在本研究之前的實驗曾提及，我們對於每一個儲存單位的大小在傳送時所會造成的流量假設為 3000，其在搜尋流量的產生相對於 Gnutella 來說為 145%，若是其大小更大，對於流量的產生也就愈大。

而在 Percolation、Index 中，每個儲存單位的形式可看做是<key, value>的形式，其中，key 為檔案名稱，value 為可提供此檔案的節點資訊，這樣的一筆資訊，相對於 YAPPERS 的檔案來說是非常小的。而節點的負擔，會隨著網路大小而變化；即便如此，對於 Percolation、Index 來說，其負擔還是非常小的。

## 6. 研究結論

在 P2P 網路環境日益盛行的今日，固然提供了我們在檔案取得上相當大的便利性，但是相對的，也對目前的網路設備及整體網路環境造成了相當大的負擔。雖然在目前的發展上，P2P 網路除了非結構的架構之外，也另外有結構化的 P2P 網路架構提供了較佳的搜尋機制，但受限於其較特殊的架構，目前在應用上仍不普及。

在本研究中，我們研究了不同在改善目前基於 Gnutella 協定的非結構化 P2P 網路的搜尋機制，我們介紹了成效較好的 YAPPERS、Percolation，也另外提出了一個改善的搜尋機制—Index，在經過模擬實驗後，也驗證了本研究的 Index 機制，除了在搜尋流量方面相較於 Gnutella 有相當程度的改善之外，在 Coverage、Success Rate 等方面相較於 YAPPERS、Percolation 也毫不遜色。

本研究在 3.1 節曾提及希望所設計的搜尋機制能符合下述的需求：

1. 對現行的非結構化對等式網路架構不做任何的變動。
2. 在搜尋前，節點不需將自己所能分享的資源

- 或是可供分享資源清單存放在其他節點上。
3. 在搜尋時，透過相對少數節點的幫忙即可找到所需求的資源；此點也意謂著，在搜尋時能減少一定程度的整體網路流量。

在本研究方法設計時，即已讓所設計的方法符點上述的 1、2 兩點，在經過模擬實驗的驗證後，也驗證了上述的第 3 點。

而透過模擬實驗的結果，可以發現 Index 機制尚有以下三點貢獻：

1. 對於非結構化 P2P 網路中，在資源搜尋時所產生的訊息洪流有一定程度的減少。
2. 透過相對較少的節點的幫忙，即可找到所欲搜尋的資源。
3. 可減少熱點(hop spot)現象，亦即，大部分的 queryHit 不會由少數的節點所回應。

## 參考文獻

- [1]. "How Peer-To-Peer (P2P) and Kazaa Media Desktop Work", Kazaa Website: [http://www.kazaa.com/us/help/guide\\_aboutp2p.htm](http://www.kazaa.com/us/help/guide_aboutp2p.htm), 2003.
- [2]. A. Rowstron and P. Druschel, "Pastry: Scalable, Distributed Object Location and Routing for Large-scale Peer-To-Peer Systems," Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms Heidelberg, Vol. 2218, pp.329-350, 2001.
- [3]. C. Rohrs, "LimeWire Design", LimeWire documents: <http://www.limewire.org/techdocs/design.html>, 2001.
- [4]. H. Balakrishnan, M. F. Kaashoek, D. Karger, R. Morris, I. Stoica, "Looking Up Data in P2P Systems," Communications of the ACM, Vol. 46, No. 2, pp.43-48, February 2003.
- [5]. I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan, "Chord: A Scalable Peer-To-Peer Lookup Service for Internet Applications," Proceedings of the ACM SIGCOMM 2001 Technical Conference on San Diego, USA, August 2001.
- [6]. J Ritter, "Why Gnutella Can't Scale. No, Really." <http://www.darkridge.com/~jpr5/doc/gnutella.html>
- [7]. M. Ripeanu, I. Foster, and A. Iamnitchi, "Mapping the Gnutella Network: Properties of Large-Scale Peer-To-Peer Systems and Implications for System Design," IEEE Internet Computing, Vol. 6, No. 1, pp.50-57, January-February 2002.
- [8]. N. Sarshar, P. O. Boykin, V. P. Roychowdhury, "Percolation Search in Power Law Networks: Making Unstructured Peer-To-Peer Networks Scalable," Proceedings of the IEEE 2004 Peer-To-Peer Computing, pp.2-9, 25-27 Aug. 2004
- [9]. P. Ganesan, Q. Sun, H. Garcia-Molina, "YAPPERS: A Peer-To-Peer Lookup Service Over Arbitrary Topology," INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE Vol. 2, pp.1250-1260, 30 March-3 April 2003.
- [10]. S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A Scalable Content-Addressable Network," SIGCOMM'01 on San Diego, California, USA, pp.161-172, August 27-31, 2001.
- [11]. S. Saroiu, P. K. Gummadi, S. D. Gribble, "A Measurement Study of Peer-To-Peer File Sharing Systems," MMCN'02 on San Jose, California, USA., pp.161-172, August 27-31, 2001.
- [12]. T. Klingberg and R. Manfredi, Rfc-gnutella, <http://rfc-gnutella.sourceforge.net>.
- [13]. Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker. "Making Gnutella-like P2P systems scalable," Proceedings of SIGCOMM 2003 on Karlsruhe, Germany, pp.407-418, 2003.