行政院國家科學委員會專題研究計畫　成果報告

Peer-to-Peer 架構下的匿名機制:分析與應用

計畫類別： 個別型計畫
計畫編號： NSC92-2213-E-009-070-
執行期間： 92 年 08 月 01 日至 93 年 07 月 31 日
執行單位： 國立交通大學資訊科學學系

計畫主持人： 楊武

計畫參與人員： 楊武、黃經緯

報告類型： 精簡報告
報告附件： 出席國際會議研究心得報告及發表論文
處理方式： 本計畫可公開查詢

中　華　民　國 93 年 11 月 1 日

行政院國家科學委員會補助專題研究計畫成果報告

※※※※※※※※※※※※※※※※※※※※※※※※※
※　　　　　　　　　　　　　　　　　　　　　　　※
※　　Peer-to-Peer 架構下的匿名機制：分析與應用　　※
※　　　　　　　　　　　　　　　　　　　　　　　※
※※※※※※※※※※※※※※※※※※※※※※※※※

計畫類別：個別型計畫
計畫編號： NSC 92－2213－E－009－070－
執行期間： 92 年 8 月 1 日至 93 年 7 月 31 日

計畫主持人：楊武
共同主持人：

本成果報告包括以下應繳交之附件：
□赴國外出差或研習心得報告一份
□赴大陸地區出差或研習心得報告一份
□出席國際學術會議心得報告及發表之論文各一份
□國際合作研究計畫國外研究報告書一份

執行單位：國立交通大學資訊科學學系

中　華　民　國　93　年　10　月　31　日

行政院國家科學委員會專題研究計畫成果報告

計畫編號：NSC 92-2213-E-009-070-
執行期限：92 年 8 月 1 日至 93 年 7 月 31 日
主持人：楊武　國立交通大學資訊科學學系
計畫參與人員： 黃經緯　交通大學資訊科學學系

一、中文摘要

在目前的 TCP/IP 通訊協定中，通訊的兩端的 IP ADDRESS 是可被檢視的，除了無法做到匿名通訊之外，還有可能會被惡意程式加以侵入。我們提出一個基於點對點系統下的匿名通訊機制(Anonymous Communication on Peer-to-Peer System)，本機制稱之為 EAS (Efficient Anonymous System)系統，在 EAS 中，訊息的路由(Routing)與頻寬耗用都運用了先進的演算法，使其更有效率。在 EAS 的實驗中，我們也發現雖然匿名通訊與點對點系統的自我組織(Self-Organization)兩者互相衝突，但是兩者依然可以互相合作來改進匿名通訊的效率。

關鍵詞：點對點、主從架構、匿名通訊機制

Abstract

In current TCP/IP based computer networks, the IP addresses of the communicating parties are still visible to outside observers. This creates a potential weakness of the network and facilitates attacks. We propose a mechanism for anonymous communication on peer-to-peer systems, called EAS. In EAS, we used several advanced algorithms to make communications more efficient. We also discovered potential conflicts between anonymous communications and self-organization of peer-to-peer systems.

Keywords: peer-to-peer, server-client, anonymous communication.

二、緣由與目的

匿名通訊在目前的日漸受到重視的網路隱私要求下越來越顯得重要，目前已經有眾多匿名通訊在各方面的應用與實作，例如 Gnutella 這個 Peer-to-Peer 的檔案交換系統，它提供了匿名搜尋 (但檔案傳輸的部份則則無此功能)，又如 Freenet 也是一個 Peer-to-Peer 的訊息交換系統，它提供了高度的匿名性(Anonymity)，訊息文件的發送與接受端都受到匿名保護，但是 Freenet 的缺點是效率不彰，一份文件的傳遞往往需要相當多的時間方能完成。另外，相關的研究還包含了 Tarzen 與 APFS，兩者都在 TCP/IP 的層次中進行 Packet 的匿名傳送，不過這兩者都需要一群可被信任的主機群來改變 與混亂化 Packet 的傳遞路徑。

我們針對 Gnutella 與 Freenet 進行研究，找出兩個可以加以改進其通訊效率的方面，第一為更好的自我組織能力，第二為更有效率的訊息路由路徑(Message Routing Path)。對於像是 Gnutella 和 Freenet 等的點對點系統而言，為了維持訊息傳遞的匿名性，節點(Node)與節點之間的連結(Connection)很少變動，而通訊則是憑藉著節點之間的訊息傳遞。這樣的方式容易造成通訊成本的大幅提高。

舉例來看一個極端的例子，在 England 的 John 想要與他的鄰居 Marry 通訊，但是卻得透過他們的共同朋友，位於美國的 Tom，這樣的通訊方式耗費了寶貴的頻

寬。EAS 的做法是讓每一個節點認識最多 O(logN) 個鄰近節點，在犧牲一點匿名性的情形下, 讓訊息能夠快速的到達目的地。

另一方面，Gnutella 和 Freenet 都利用了 return path 的方法來讓搜尋要求(Search Request)的回覆訊息(Reply Message)沿著搜尋相反的路徑傳回發起該搜尋要求的節點。這樣的做法增加了頻寬的耗用，以及所有節點的負擔，因為每一個節點必須比對數千或數萬個先前訊息在該節點上留下的指標，確認出目前的訊息時，將該訊息傳遞給路徑上的前一個節點。許多研究 [] 指出 Cache 可以降低頻寬的使用與節點的負擔, 但是另一項研究 [] 則指出 Cache 會降低 P2P 網路的 Small-World 效果, 而該效果卻是快速訊息傳遞的重要基礎。EAS 的做法是對每一個節點都配置一個 Proxy，節點與節點之間的通訊都透過彼此 Proxy 傳遞，而非原本的搜尋路徑，這個做法的好處是路徑上的頻寬耗用不再出現。

EAS 根基於我們先前所作的研究 – DSE (Distributed Search Engine)，在 DSE 中的 Self-Organization 機制中，每個節點不斷的更新鄰近節點(Neighbor)，使得 IP Address 最接近的節點能連結在一起。在 DSE 中，每個節點在其鄰近節點中選擇一個成為他的 Proxy 來負責為其接收訊息。

三、結果與討論

系統架構

Neighbor Clustering Control

在 DSE 中，每個節點都有一個獨特的 ID 以及一個 IP Address，DSE 中連結各個節點的機制稱之為 Neighbor Clustering Control (NCC)，NCC 讓 IP Address 相近的節點互相連結，除了相近的節點之後，每個節點產生少許連結到較遠的節點，最終整個系統成為一個階層式的叢集結構，相近的節點連結成為一叢，各叢之間則有少數的連結。換言之，NCC 讓整個系統出現 Small-World 效應，而使得訊息傳遞能夠迅速完成。

匿名的定義

在英文中，匿名 : Anonymity 節制目前為止並無一個明確的定義，在本計畫當中，我們給予自行的定義 : 對於一個系統的兩個節點 X 與 Y, 若 X 知道 Y 的 IP Address 與 Y 的 ID 之間的對應, 則我們說 X recognizes Y (X 辨識 Y)。若系統符合下列兩個條件，則該系統稱之為 Anonymous : (1) 每個節點最多辨識 O(1) 個節點 (2)所有訊息的發出者無法被辨識。

另一方面，訊息(Message)是系統中最容易被破壞者用來破解系統匿名性的下手目標，訊息可以被追蹤，或者被動手腳使得其來源被辨識出來。對 EAS 中的訊息來說，除了發送者本身之外，若該訊息的發送者無法被辨識，則我們說該訊息為 Anonymous。

匿名的基本規則

通常來說，匿名與系統的自我組織 (Self-Organization, 簡稱為 SO)是相互衝突的，因為 SO 必須讓節點彼此互相認識 (Recognition), 以使得整個系統能夠分散式的形成特定的 Topology，然而匿名則是希望節點彼此之間儘可能不認識對方，我們的 EAS 就是希望能夠在兩者之間能夠達到一個平衡與妥協。在發展 EAS 之前，我們先給予一些基本的假設:

1. 對於節點 A 與 B，若 A 認得 B，兩者之間必定 “目前” 或 “曾經” 存在過連結(Connection)
2. 對於節點 A 與 B，若 A 認得 B，則 B 認得 A
3. 對於不認識的 A 與 B，任一其他(正常未被破解的)節點 C 不能告知 A 其 B 的位址，反之亦然
4. 對於不認識的 A 與 B，A (正常未被破解的節點)無法破壞由 B 所傳送的訊息的匿名性

假設 1 任一個節點無法認識除了它的 Neighbor 之外的節點，假設 2 說明了認識(Recognition)關係是對稱的，假設 3 保證沒有任何一個節點可以破壞其他節點的匿名性，第 4 點則確保沒有節點能破壞訊息的匿名性。

顧及匿名性質的自我組織

在 DSE 中，NCC 機制不斷的根據遇到的節點的遠近來覺得是否該節點可成為它的 Neighbor，因此必須存在一個方式來決定任一兩節點之間的距離，我們採用的方式是給定一個 Distance 函數，對於兩個節點 A 與 B，A 的位址為 a1.a2.a3.a4，B 的位址是 b1.b2.b3.b4，則兩者的距離定為:

$$\sum_{i=1}^{4} |a_i - b_i| * K^{(4-i)}, \quad where\, K\, is\, a\, constant$$

很明顯的，這個 Distance 公式並不能非常精確的反應出任兩個節點之間的真實距離(比如說 routing path 上的 gateway 個數)，但是它卻可以大致上反應出是否兩個節點屬於同一個 AS (Autonomous System)。NCC 機制根據節點彼此之間的距離來決定節點之間的認識(Recognition)，其目的是要讓訊息傳遞更有效率，比如說讓在進行 search request 時，flooding messages 能夠盡量減少，或者是訊息的傳遞能夠盡量集中在同一個 AS 中(也就是較高的 message locality)。

NCC 的運作機制如下，(1) 每個訊息都內含發出該訊息的節點資訊(節點 ID 與 Address)，(2) 每個節點收到訊息後，取出內含的發出者與本身之間的距離，(3) 每個節點持續地連接距離本身更近的節點，直到數量足夠為止。

NCC 的機制與匿名性可說是互相違背的，我們為此找出一個平衡的做法，這個機制稱之為 Passive Neighbor update (簡稱之為 PNU)，PNU 機制讓節點透過盡量減少的 recognition 次數便可認識足夠的 Neighbor，使得 NCC 機制仍能有效運作。

Passive Neighbor Update 機制

在說明 PNU 機制之前，先定義一些會用到的項目:

RECID : 對於一個節點 H 來說，RECID 是 H 累積所認識的 Neighbor 的集合(包含之前的與目前的)

AVDS : 整個系統中，一個 connection 的平均距離，即所有 connection 的距離總和除以所有 connection 個數

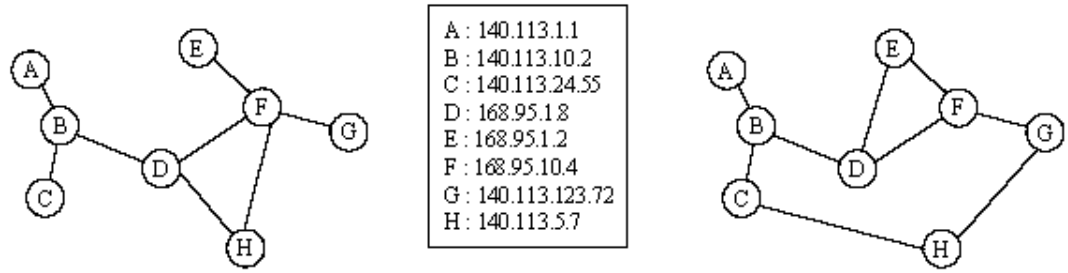
AVIS : 整個系統中，平均每一個 node 累

積所認識的 Neighbor 個數(包含之前的與目前的)

PNU 的運作機制很簡單，假設一個節點 S 與它的 Neighbor 節點 N，S 持續地在它所認識的節點(先前與目前的 Neighbors)中找出與 N 距離最近的節點 M，並且將 M 介紹給 N 認識。PNU 運作機制將會逐漸的降低 AVDS 並且增加 AVIS 一直到所有的節點都連接到與它最接近的節點。

另外，節點之間的 Introduction 必須是互相信任的，也就是說任一節點僅能接受它的 Neighbor 的命令來與其他的節點互相認識。PNU 機制的運作基礎基於先前提到的

以圖例來說明，下列的例子中，A 是原始的系統，B 是經過 PNU 機制轉換過後的系統，AVDS 從原本的 235,477,713 降低到 104,662,271，而整個轉換的過程為: F 介紹 D,E 認識、F 介紹 H,G 認識、F,H 的連結中斷、D 介紹 B,H 認識、B 介紹 C,H 認識、B,H 的連結中斷。



(a) AVDS = 235,477,713

(b) AVDS = 104,662,271

架設 3，若沒有這項假設，任一節點將可以認識無限多個其他節點。

PNU 機制的演算法列表如下:

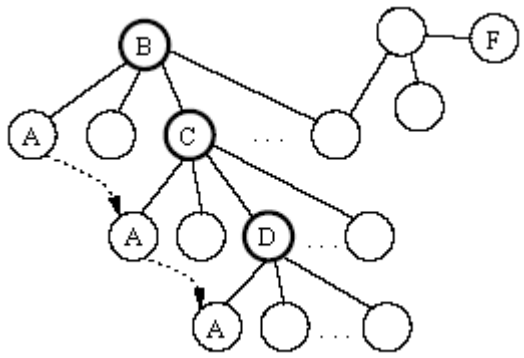
```
function PassiveNeighborUpdate() {
    // Assume this function is running on node S
    // NBRQ : the set of neighbors of node S,
    // RECID : the set of current and past neighbors, NBRQ ⊆RECID

    for each node N in RECID {
        find node K such that K∈NBRQ and distance(K,N)<=distance(P,N)
                for any P∈RECID;
        inform nodes N and K to recognize each other;
        };
    };
```

PNU 機制事實上非常類似將一個 node 加入(join)一個 Ordered, Threaded 的 K-ary Tree，以下面的例子來看，節點 A 加入了一個 Tree，它的連近節點 B 將它介紹給了 C，因為 A 與 C 的距離近，而 C 又將 A 介紹給了 D，這樣的介紹持續到 A 連接到與它最近的節點為止。假設 Tree 上的每個 node 都有 k 個 neighbors，那麼一個新的

效能分析

PNU 機制的理念是在盡量讓 AVDS 越高而 AVIS 越低，我們想要來探討 PNU 機制下，AVIS 的估計值。

node 的加入將會增加 O(log $_k$ N)個新的認識(Recognition)。



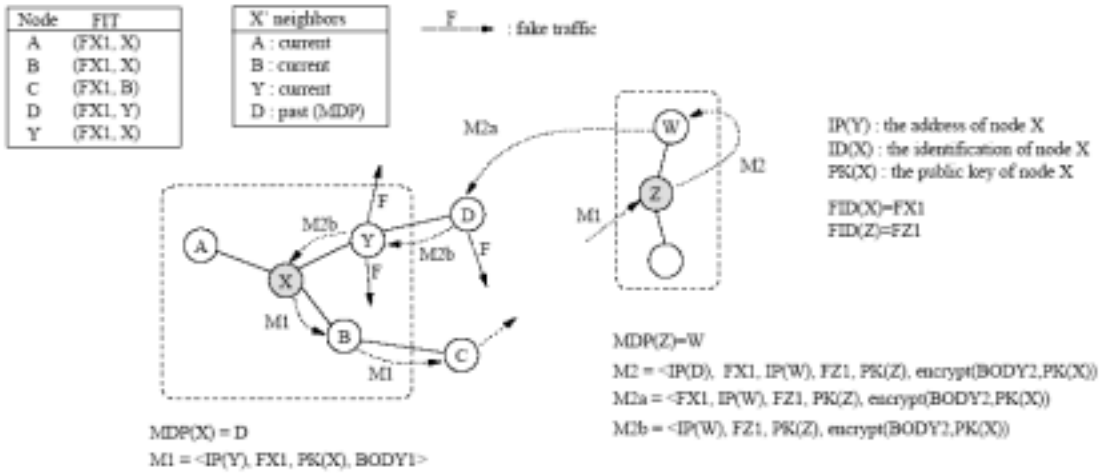
匿名通訊

在 EAS 中，我們實現匿名通訊的方式是每一個節點都配置一個節點作為轉接訊息的 Proxy，該節點稱之為 Message Delivery Proxy (MDP)，每個節點都配置一個表格，稱之為 Forwarding ID Table (FIT) 來儲存過往該節點的訊息的來源。此外，訊息的傳遞過程必須加密，我們採用的是 Private/Public Key 的方式。

以下面的圖例來說，假設節點 X 的 MDP 為節點 D，當 X 發起一個 Request M 時，X 會動態產生一個獨特的指標，稱之為 Forwarding Pointer (標示為 FID(X))，每一個廣播給其 Neighbor 的訊息當中包含了三種資訊: (1) FID(X)，(2) X 本身的 public key，(3) D 的 IP Address。假設 X 目前的 FID(X)為 FX1，當某個節點 H 收到訊息 M 時，會將(FX1,S)加入自己的 FIT 中，S 為 forward 訊息 M 給 H 的節點。

假設節點 Z 收到了一則訊息 M1，該訊息是從節點 X 所啟始，當 Z 想要送出回覆訊息 M2 給 X 時，Z 首先會以 X 的 public key 來加密訊息內容，然後 D 的 IP Address 以及 FX1 會附加於加密的訊息之後。完成上述動作之後，節點 Z 將 M2 傳遞給他的 MDP W，一但 X 收到加密後的 M2，會將附加的 D 的 IP Address 給移取出來，然後將修正後的訊息 M2a 傳遞給 D，當 D 收到 M2a 時，會尋找自己的 FIT，當發現有吻合的 forwarding 記錄時，會將該訊息的 Forwarding Pointer 移取出來，然後將修正後的訊息 M2b 傳遞給該筆記錄的 Forwarding 節點，也就是 X；最後，X 收到訊息 M2b 並且以本身的 Private Key 解密取得訊息內容。上述的訊息傳遞路徑稱之為 Routing Path。

注意到在訊息的 Routing Path 中，還可以加上干擾訊息，來增加被破解(猜出 ID 與 IP 的對應)的困難度，以上述例子來說，當節點 D 與 Y 收到訊息 M2a 與 M2b 之後，可以送出所謂的干擾訊息給他們的 Neighbor，這些干擾訊息都是亂數產生，與 M2a 和 M2b 是無關的。此外，干擾訊息的產生量可以與訊息的 HOPS 數成正比，當訊息越接近正確的目的地時，干擾訊息產生得越多。



攻擊分析

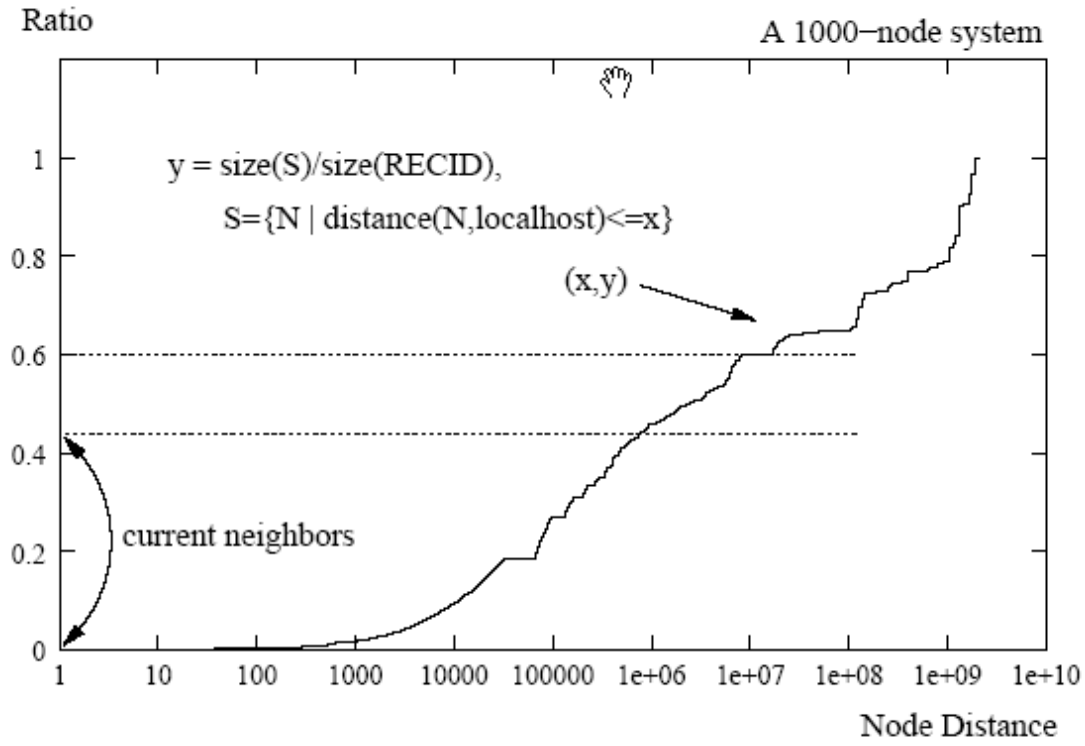
我們採用 Reiter and Rubin 的標準來分析我們的系統的安全性，根據這個標準，匿名通訊的特性可被分類為三個座標軸，第一個軸為匿名的類型:sender 或是 receiver，第二個軸為壞人的類型:本地的窺探者(local eavesdropper)、團隊合作型的一群壞人(collaborator of malicious nodes)、以及訊息的 sender 與 receiver 本身即為壞人。第三個軸為匿名的程度:absoulte privacy、beyond suspicion、probable innocence、possible innocence、exposed、以及 provably exposed。

我們的 EAS 系統經過分析，匿名的特性列表如下:

	one-to-all (broadcast)		one-to-one (reply)	
attacker	sender	receiver	sender	receiver
local eavesdropper	exposed	beyond suspicion	exposed	beyond suspicion
collaborating nodes	beyond suspicion	beyond suspicion	absolute privacy	absolute privacy
sender	N/A	N/A	N/A	absolute privacy
receiver	absolute privacy	N/A	absolute privacy	N/A
sender.MDP	beyond suspicion	N/A	beyond suspicion	absolute privacy
receiver.MDP	N/A	N/A	absolute privacy	beyond suspicion

MDP 的選擇

由於 MDP 是從所有認識(曾經或是目前)的節點，也就是 RECID，當中來選取，同時，若 RECID 當中的節點與本身的距離(Distance)範圍越廣，則表示 MDP 的選取空間也越大，也就是表示匿名的程度可以越高。我們分析了 1000 個節點的 EAS 系統，以下列出 RECID 的分布狀況:



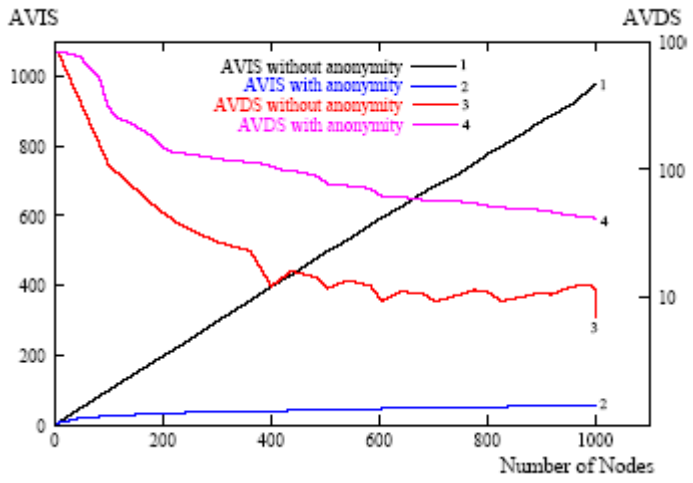
從圖中可看出，RECID 中 60%的節點其距離低於 10^7，而屬於其中的 42%的節點則為目前的 Neighbor；至於另外的 40%節點其距離則介於 10^7 到 10^9 之間。這項結果顯示 RECID 提供了相當寬廣的範圍可以來選取適當的 MDP，利用這點，我們可以根據通訊成本與匿名程度之間的平衡來選擇較近或較遠的 MDP。

效能分析

在這個部份，我們進行了一系列的模擬，藉以分析 EAS 的效能。

PNU 效能分析

我們測試了兩個系統，系統大小都是 1000 個節點，第一個系統將 PNU 功能關閉(即無 Anonymity)，第二個系統開啟 PNU 功能。我們觀察兩個指標:AVIS 與 AVDS，其結果如下:



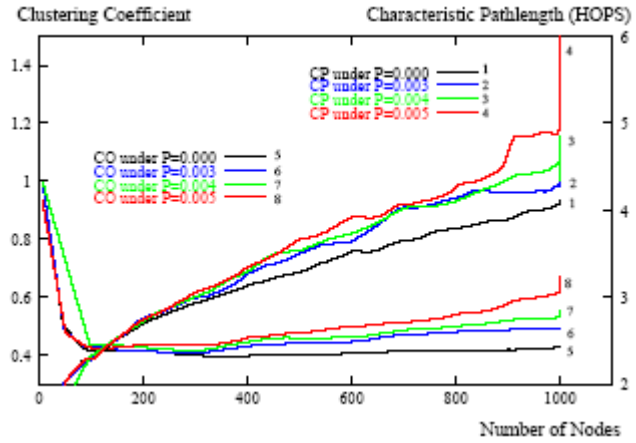
圖中我們可看出在 PNU 機制啟動之下，

AVIS 大幅降低，換句話說，也就是節點彼此之間的認識(Recognition)大幅減少。但是 AVDS 的部份卻比關閉 PNU 的系統的 AVDS 高，這意味著 PNU 將會犧牲一些系統通訊方面的效率 (因為 AVDS 越低表示所有節點越能夠與距離近的節點相連)

這樣的方式會讓原本的 NCC 效能降低，因此，我們改進 PNU 機制，使得除了原本的方式之外，再以亂數方式找一個額外的節點來讓 Neighbor 認識。給定一個節點 A，以及它的 Neighbor 的集合 S，假設 A 正在為它的 Neighbor H 找出 S 中最接近的節點，來使其兩者認識。我們設置一個可調整的機率 P，讓機率在小於等於 P 時，A 會在 S 中亂數選出一個節點來介紹給 H 認識。

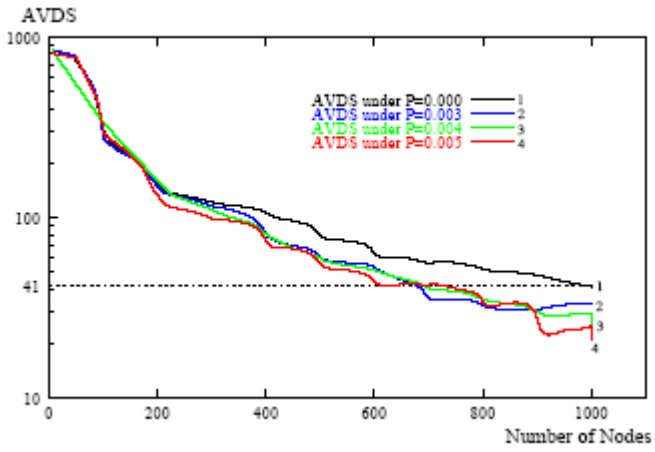
PNU + Random Recognition 效能分析 (I)

由於 PNU 的原始設計是在讓節點為它的每一個 Neighbor 找出最接近的節點來認識，



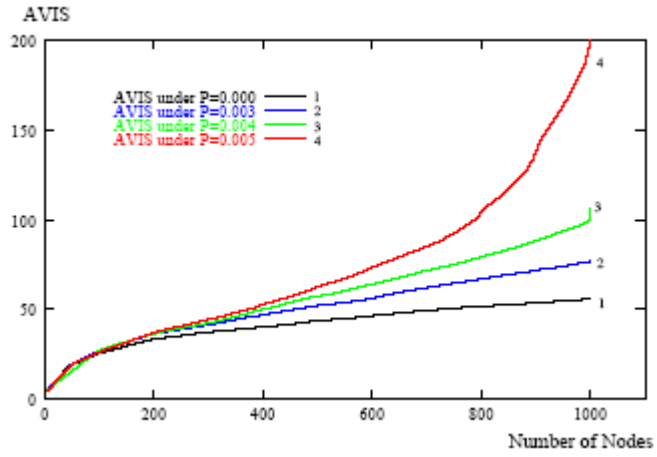
這表示 P 值不能設得太大，否則 AVIS 快速升高的結果將使得匿名程度大為降低。

本圖說明了 PNU + Random Recognition 的方式有助於 NCC 的效能提升，在 P 越大的情形下，CO (Clustering Coefficient)從 0.4 提高到了 0.6，而 Characteristic Pathlength 卻是從 4 提高到了 5。



PNU + Random Recognition 效能分析 (II)

Random Recognition 對於 AVDS 與 AVIS 也有相當的影響，當發生 Random Recognition 的機率越高時，AVDS 理論上將越來越低，而 AVIS 則將越來越高，下圖證實了我們的猜測。注意到當 P 設為 0.005 時，AVIS 升高的速度變得非常快，



四、成果自評

我們的 EAS 系統實現了有效率的匿名通訊，在通訊效能與匿名之間取得平衡，犧牲一些匿名的效能，但可換取通訊效能方面的大幅提升。在每個節點最多認識 O(LogN)的 Neighbor 的情形下，系統的 CO 與 CP 都能夠與原本沒有匿名機制的系統(即 DSE 系統)相比較。我們的匿名通訊採用的是每個節點都配置一個通訊用的 Proxy，稱之為 MDP，來讓匿名訊息都能在少數的 HOPS 之內就可以傳遞到目的地。另外，使用者還可以自行調整匿名與效能的程度，當使用者希望注重匿名效能而願意犧牲一些通訊效能時，可以從 RECID 中取得較遠者來作為 MDP，當使用者希望注重通訊效率而願意犧牲一些匿名效能時，則可以從 RECID 中取得較近者來作為 MDP。

五、參考文獻

[1] Karl Aberer, Magdalena Punceva, Manfred Hauswirth and Roman Schmidt. Improving Data Access in P2P. IEEE Internet Computing, 6(1), 2002.

[2] Andy Oram. Peer-to-Peer Harnessing the Power of Distributed Technologies. O'Reilly 2001.

[3] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of `small-world' networks. Nature, vol. 363, pp. 202-204.

[4] Sylvia Ratnasamy, Scott Shenker and Ion Stoica. Routing Algorithms for DHTs: Some Open Questions. First International Workshop on Peer-to-Peer Systems (IPTPS), 2002.

[5] Matei Ripeanu, Ian Foster and Adriana Iamnitchi. Mapping the Gnutella Network: Properties of Large-Scale Peer-to-Peer Systems and Implications for System Design. IEEE Internet Computing Journal, 6(1), 2002.

[6] Ian Clarke, Oskar Sandberg, Brandon Wiley and Theodore W. Hong. Freenet: A Distributed Anonymous Information Storage and Retrieval System. Lecture Notes in Computer Science, vol. 2009, pp. 46+ ,2001.

[7] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, Hari Balakrishnan. Chord: A Scalable Peertopeer Lookup Service for Internet Applications. Technical Report TR-819, MIT, March 2001.

[8] Kunwadee Sripanidkulchai. The popularity of Gnutella queries and its implications on scalability. The O'Reilly Peer-to-Peer and Web Services Conference, September 2001.

[9] Andy Oram. Peer-to-Peer Harnessing the Power of Distributed Technologies. O'Reilly 2001. pp. 94-122.

[10] Proceedings of Designing Privacy Enhancing Technologies: Workshop on

Design Issues in Anonymity and Unobservability, July 2000.
[11] The Free Network Project. http://freenetproject.org/
[12] The free haven project. http://freehaven.net/.
[13] References H. Zhang, A. Goel, R. Govindan, Using the Small-World Model to Improve Freenet Performance, Proceedings of IEEE Infocom, 2002. 14.