

# 對等式(P2P)資源分享網路

吳明蔚 林盈達

國立交通大學資訊科學系  
新竹市大學路 1001 號

E-MAIL : [benson@cis.nctu.edu.tw](mailto:benson@cis.nctu.edu.tw) , [ydlin@cis.nctu.edu.tw](mailto:ydlin@cis.nctu.edu.tw)

TEL : (03) 5712121 EXT. 56667

## 摘要

網路網路的變革日新月異，Peer-to-Peer (P2P) 技術能讓使用者「直接互相分享」。如同第二代網路技術 Mosaic 之後所代表的「超連結、多媒體」，顛覆第一代文字為主的網際網路。繼 Napster 之後代表的「P2P」網路架構，其衍生的思維、技術和商業模式，可是來勢洶洶！本文從 P2P 的演進歷史，介紹及比較 Napster 和 Gnutella 兩者典型的 P2P 架構，並對目前業界的 P2P 應用及最新消息作全面的整理，緊接著以封包格式、問題與瓶頸兩者來切入探討。文末針對 P2P 未來的研究議題作了完整的歸納。

關鍵詞：P2P、Napster、Gnutella、Freenet、gPulp

## I. P2P 歷史回顧：

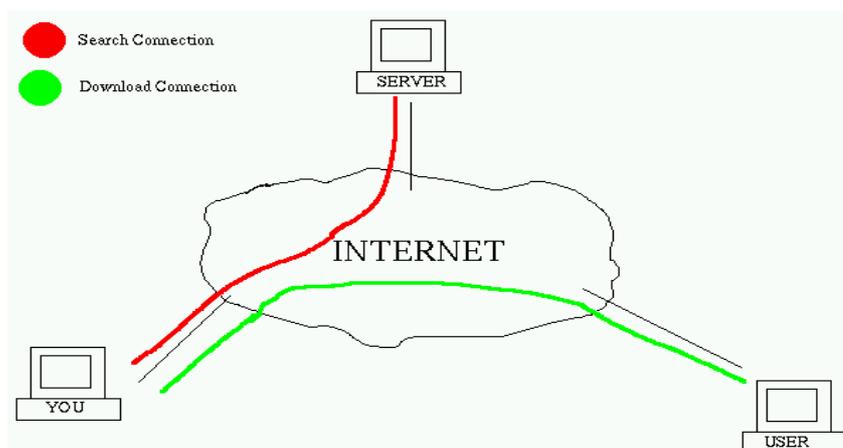
在 1999 年 5 月，Shawn Fanning 的 Napster，利用 P2P 技術，達到 MP3 音樂的檔案分享。同年因音樂版權問題，與 RIAA 官司纏身，當時被迫關站，但在網際網路上已開 P2P 之門。在 2000 年 3 月後，有許多 Napster 翻版(Napster-clone) 在網路上流通，諸如 Gnutella、CuteMX、Scour 等等。2000 年中期之後，P2P 技術從單純的檔案交換，更衍生出其他商業模式和應用，許多公司皆投入 P2P 開發，諸如 CenterSpan、Entropia、Groove 等等。P2P 的正式組織和會議也相繼召集舉辦，如 Intel 的 P2PWG(Peer-to-Peer Working Group)第一次會議在 2000/10/12 以及第二次會議是在 2001/2/7-8、O'Reilly 在 2001/2/14-16 的 P2P Conference 等等。

## II. P2P 的架構

什麼是 Peer？主機本身既是 Client 也同時是 Server，跳脫主從(client-server)

架構的固定模式，做到彼此可以對等地溝溝分享。所有的 Peer-to-Peer 是架構在 TCP/IP 協定，使用 TCP 封包，而非 UDP。專注在 MP3 檔案分享的 Napster，是 P2P 的應用之一，其集中式架構圖請參考圖一；反之，Gnutella 的分散式架構，請參考圖二。

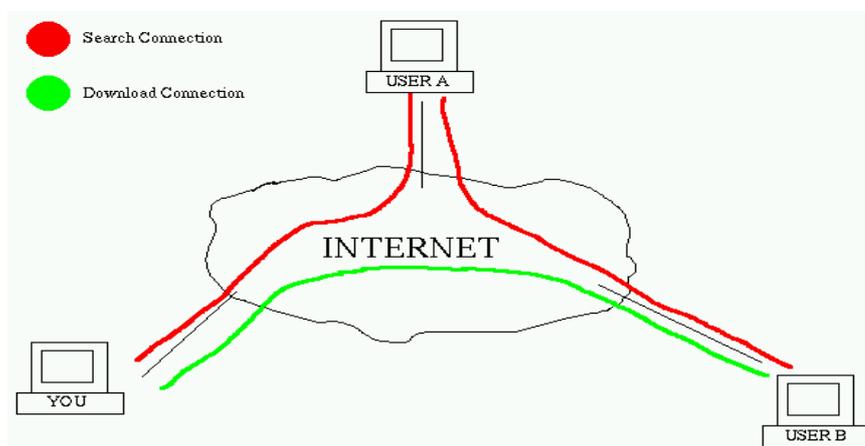
### A. Napster 的架構：



圖一 Napster 的架構及運作方式

1. **連線**：安裝 Napster 軟體，做為 Napster P2P 網路的 Peer(Client 端)，然後連上「事先設定」的伺服器。Napster Client 端會將使用者端的相關資料以及分享檔案列表傳送給 Server 端。
2. **搜尋**：當你想搜尋檔案時，會將查詢要求傳送給 Server，然後 Server 會尋找本身的資料庫的檔案列表，並將結果回傳回去。
3. **下載**：收到 Server 的結果後，你可以選擇想下載的檔案，並直接連線至對方(Peer)，進行 Peer-to-Peer 的傳送，而無須透過 Server 的介入。

### B. Gnutella 的架構：



圖二 Gnutella 的架構及運作方式

1. **連線**：安裝 Gnutella 軟體，做為 Gnutella P2P 網路的 Peer 端(是 Server 也是 Client)，沒有任何事先設定的伺服器來連結，因此對 Gnutella 網路可以說一無所知。因此，需要知道至少一台 Peer 的 IP 位址和 Port，做為第一個連線(起始點)。接著有三件事會發生：
  - i. 你宣布你的存在
  - ii. 他會告知其他 Peers
  - iii. 每個 Peer 回訊息給你藉由和其他交換資訊以瞭解整個 Gnutella P2P 網路。  
如果需要知道有目前有哪些起始點可以使用，可連線至這個網址：  
<http://www.ifi.ntnu.no/~havardot/betaz/servers.htm>
2. **搜尋**：當你想搜尋檔案時，會將查詢要求傳送給你有直接連線的 Peer(s)，這些 Peer(s)會繼續將你的要求傳送給他們有直接連線的其他 Peer(s)，然後繼續反覆下去。在這散佈過程的每一台 Peer 也都會尋找本身的檔案列表，並將結果回傳回去，如果找不到符合結果，無須回報。
3. **下載**：陸續收到 Peer(s)的結果後，你可以選擇想下載的檔案，並直接連線至對方(Peer)，進行 P2P 的傳送，而無須透過網路上其他 Peer(s)的介入。如果連線失敗，則很有可能是對方在防火牆之後，Gnutella 會重新發送一個下載要求(Push 下載)，並以 Step 2 的方式散佈此要求，對方(擁有你想下載的檔案)會嘗試從他那邊直接連線回來，進行 P2P 傳送。

### III. 檔案分享(File-sharing)的演進與比較：

目前在 P2P 檔案分享的領域，可以分類出三種明顯差異的應用軟體。首先是 Napster，其目標很單純，就是純 MP3 音樂檔案的分享，有一個集中式伺服器做目錄的建立，因此整個網路的資源消耗較低。使用者並無法匿名，分享出來的檔案在傳輸過程也沒有加解密，明顯的優點就是簡單使用和延展性較高。

接著是 Gnutella，其改成純分散式的架構，沒有集中式的伺服器，相對地網路的延展性變低，容易造成壅塞。Gnutella 支援的檔案格式比較多，幾乎所有的多媒體格式皆可，使用者在下載過程也無法匿名或資料加密。

最後是 Freenet，如其名，給予使用者完全自由的網路空間，提供不同資源的分享，不侷限在檔案種類。在 Freenet 中，每一位使用者皆是 Freenet 的一份子，不僅匿名或資料加密，一旦將檔案放在 Freenet，你自己也無法將其拿掉！儼然成爲一個大一統的檔案分享。檔案本身會依據熱門程度，適當的移動或複製到需求量高的位置，相對地，一旦一個檔案在 Freenet 上乏人問津，時間一久即會自然消失。至於檔案的熱門排行是由使用者下載後自行投票，如果不滿意，才取消

投票，因此不會有主觀的反投票發生。

三種檔案分享的分類表，請參考表一。

	Napster	Gnutella	Freenet
誕生於	1999/5 月	2000/3 月	2000/3 月
創始人	Shawn Fanning	Justin Frankel and Tom Pepper	Ian Clarke
目標	MP3 分享	檔案分享	大一統檔案分享
伺服器	集中式	分散式	分散式
網路資源消耗	低	高	中
支援檔案格式	MP3	多媒體	任何資料
檔案位置	靜態 (PCs)	靜態 (PCs)	動態 (Freenet)
熱門檔案？	推薦	下載次數	投票/取消投票
熱門檔案可能位置	-	-	移動或複製至附近
檔案生命週期	決定在你	決定在你	自然消失 (Freenet)
匿名	否	否	是
資料加解密	否	否	是
延展性	中	低	中
簡易使用	是	是	否
回饋歌手？	否	否	可能(*)
觀察別人下載	否	是	否

\* Freenet 正在與 Uprizer 合作發展一套回饋歌手的商業模式。

表一 Napster vs. Gnutella vs. Freenet

當然，檔案分享絕對不止 Napster 或 Gnutella 幾種，還有許多類似的應用軟體，茲分類整理五種最佳的 Napster 分身於表二。

排名	軟體名稱	優點	缺點
1	CuteMX	容易使用、線上試聽	N/A
2	Scour Exchange*	華麗介面、搜尋快速	結果太少
3	Gnutella	功能強大的介面	搜尋太慢
4	SpinFrenzy Xchange	清楚目錄分類	下載太慢
5	Audiogalaxy	準確搜尋結果	結果太少、FTP 協定

表二 Napster 分身 (Napster-clone) 的評比 (\* SE 已經被告到倒閉了...)

#### IV. 業界應用：

##### 1. 電玩 / 訊息網路：

- 1 CenterSpan ([www.centerspan.com](http://www.centerspan.com)) 專注在 peer-to-peer 聲音文字訊息傳送，在多人電玩環境的應用。
  - 1 Jabber ([www.jabber.com](http://www.jabber.com)) 是使用 XML 提供 peer-to-peer 即時訊息的系統，類似 ICQ、AOL 及 MSN 等即時訊息軟體。
2. 分散計算:
- 1 Applied MetaComputing ([www.appliedmeta.com](http://www.appliedmeta.com)) 的產品 Legion(羅馬軍團)提供企業做跨平台的 peer-to-peer 資源管理工具及 peer-to-peer 知識管理應用軟體。讓網路內的資源，緊密的整合，且在網路安全、功能及彈性上都有兼顧。
  - 1 Datasynapse ([www.datasynapse.com](http://www.datasynapse.com)) 提供財務公司 peer-to-peer 的網路，最終目標是全球性的財務網路。
  - 1 Entropia ([www.entropia.com](http://www.entropia.com)) 致力在計算能力分享的軟體設計，結合上萬台電腦的空閒時刻，進行分散式計算，免費協助解決一些人類的共同問題，例如搶救愛滋、藥物搜尋與運算、醫學影像處理等等。當然，某些百分比的計算能力該公司會發展營利用途。
3. 管理 / 合作：
- 1 惠普(Hewlett-Packard)計畫將推行已一年多的 e-speak 產品提供 peer-to-peer 上的應用。
  - 1 由 Lotus Note 創始人 Ray Ozzie 所創辦的 Groove Networks ([www.groove.net](http://www.groove.net)) 投入開發採購、批發、貨品管理及客服等應用軟體。
  - 1 英代爾(Intel)也著眼於企業級的 peer-to-peer 應用軟體，並成立了一個 Peer-to-Peer Working Group。
  - 1 Quiq ([www.quiq.com](http://www.quiq.com)) 鎖定在的前一千大企業(Fortune 1000)並提供虛擬 peer-to-peer 的社群環境。
4. 檔案分享
- 1 NextPage ([www.nextpage.com](http://www.nextpage.com)) 讓企業內部的檔案伺服器能以 peer-to-peer 的模式工作。
  - 1 在網路上流傳著微軟(Microsoft)有個代碼為「Farsite」的 peer-to-peer 計畫。其目標是僅以集中式伺服器負責目錄與協調，做到 peer-to-peer 檔案儲存，造就無伺服器的境界，也就是「serverless」。
5. Peers 代理人/ 分散式搜尋引擎：
- 1 從西門子出來的 WebV2 ([www.webv2.com](http://www.webv2.com)) 著眼於提供企業的 Supply chain，提供 peer-to-peer 的應用軟體讓不同廠商之間溝通、搜尋資料。

## 6. 超級分散式:

- 1 vTrails ([www.vtrails.com](http://www.vtrails.com)) 發展出 Full Duplex Packet Cascading (FDPC)，可以有效地減少網站面臨壅塞流量。FDPC 將陸續連線進來的使用者，先取一部份做第一個層(tier)，剩下的繼續將其分成許多層(tiers)。上一層使用者所下載的資料，會直接變成下一層使用者下載資料的來源，端看最小 hop 數目以及是否在同一地理區域，以此類推。也就是說 FDPC 將網站壅塞點，分散至一連串的使用者層(tiers)之間作 peer-to-peer 的下載。在每一台 vTrails 伺服器，會記錄地理位置圖，瞭解其使用者的位置及 IP 位址。
- 1 2AM 也有類似的:StreamingPeer

## V. 最新消息

**Scoundrel** 使用 IE 作為內建瀏覽器，提供前端點選下載介面。目前支援 Amazon.com。後端直接連線至 Napster 伺服器，進行下載！所有下載過程幾乎是全自動，使用者不需煩惱任何下載例外狀況。



**Espra** 實做 Freenet 的圖形介面，使用前需架好 Freenet gateway。



## VI. 實作 – 封包格式

針對進階使用者，在此以 Gnutella v.4 為例做介紹，其封包格式可以分為檔頭及資料。在 Gnutella 的網路裡，Pings 訊息是用來尋找其他 Peers 的存在，而 Pongs 訊息即是回應 Pings 訊息；Push 訊息是由使用者直接向在防火牆之後的目的地要求下載，由對方主動建立連線；Query 訊息是用來下達查詢，而 Query-hits 則是命中的查詢結果。

Header: 檔頭														
位元	欄位名稱	註解												
0-15	Message identifier	這是 GUID(Globally Unique ID).用來辨別是否收過此訊息。												
16	Payload descriptor (function identifier)	<table border="1"> <thead> <tr> <th>數值</th> <th>功能</th> </tr> </thead> <tbody> <tr> <td>0x00</td> <td>Ping</td> </tr> <tr> <td>0x01</td> <td>Pong (Ping reply)</td> </tr> <tr> <td>0x40</td> <td>Push request</td> </tr> <tr> <td>0x80</td> <td>Query</td> </tr> <tr> <td>0x81</td> <td>Query hits (Query reply)</td> </tr> </tbody> </table>	數值	功能	0x00	Ping	0x01	Pong (Ping reply)	0x40	Push request	0x80	Query	0x81	Query hits (Query reply)
數值	功能													
0x00	Ping													
0x01	Pong (Ping reply)													
0x40	Push request													
0x80	Query													
0x81	Query hits (Query reply)													

17	TTL	如同 TCP/IP 封包，Gnutella 的封包皆有 TTL(Time-To-Live)，避免封包在網路無止盡廣播，預設初始值都很小，約 5 左右，每經過一個 Peer 就會減 1，至 0 時就會丟棄。如果故意設成很大的 TTL，譬如 100，也會在廣播的過程被辨別出不合理，而被其他 Peer 減至一個合理值。
18	Hops	此訊息已經被 Forward 幾次。
19-22	Payload length	訊息資料內容長度。

#### **Payload: ping (function 0x00) 廣播要求**

沒有 Payload

#### **Payload: pong (query reply) (function 0x01) 廣播回應**

位元	欄位名稱	註解
0-1	Port	IPv4 port 號碼。
2-5	IP address	IPv4 位址。是 x86 的 byte order，Little endian!
6-9	Number of files	分享的檔案數
10-13	Number of kilobytes	分享的檔案量(大小)

#### **Payload: query (function 0x80) //搜尋要求**

位元	欄位名稱	註解
0-1	Minimum speed	需要回應此要求的主機最低速度，單位是 kilobytes/sec
2+	Search criteria	搜尋字串，NULL 做結尾。

#### **Payload: query hits (query reply) (function 0x81) 搜尋回應**

位元	欄位名稱	註解
0	Number of hits (N)	符合搜尋的結果數
1-2	Port	IPv4 port 號碼
3-6	IP address	IPv4 位址。是 x86 的 byte order，Little endian!

7-10	Speed	回應主機的連線速度，kilobits/sec												
11+	Result set	結果(有 N 份)												
		<table border="1"> <thead> <tr> <th>位元</th> <th>欄位名稱</th> <th>註解</th> </tr> </thead> <tbody> <tr> <td>0-3</td> <td>Index</td> <td>檔案 Index 號碼</td> </tr> <tr> <td>4-7</td> <td>Size</td> <td>檔案大小</td> </tr> <tr> <td>8+</td> <td>File name</td> <td>檔案名稱，雙 NULL 結束</td> </tr> </tbody> </table>	位元	欄位名稱	註解	0-3	Index	檔案 Index 號碼	4-7	Size	檔案大小	8+	File name	檔案名稱，雙 NULL 結束
位元	欄位名稱	註解												
0-3	Index	檔案 Index 號碼												
4-7	Size	檔案大小												
8+	File name	檔案名稱，雙 NULL 結束												
~16 bytes	Client identifier	回應端的 GUID，當 PUSH 時要用。												

### Payload: push request (function 0x40) Push 下載

位元	欄位名稱	註解
0-15	Client identifier	要執行此 Push 的主機 GUID。
16-19	Index	檔案的 Index 號碼。
20-23	IP 位址	要 Push 對象的 IPv4 位址。
24-25	Port	要 Push 對象的 IPv4 port 號碼。

## VII. 目前問題與瓶頸：

### A. 延展性(Scalability)

在 P2P 的環境下，尤其 Gnutella 的架構，當人數成長到萬人左右，壅塞狀況變得非常嚴重糟糕。Gnutella 之父 Frankel 甚至表示網路會無法負荷到 5,000 位使用者以上。

那麼，目前壅塞的瓶頸在哪？撥接數據機。

其實，有報告指出，平均的查詢頻率約為每秒十次，一旦超過此數值，回應 Pings 的 Pongs 明顯減少。另外也發現，每台 peer 最普遍是連線到另外三 peer。基本上，Gnutella 流量可以分為三種狀況(查詢、找人、回應)，共包含著五種型態的訊息：

#### 1. 查詢檔案：Query 訊息

Query 訊息包含標頭約為 30 位元組，而 TCP/IP 這部分的標頭約為 40 位元組。三個 Peer 每秒十個查詢次數的流量約是每秒  $560 \times 3 \times 10 = 16,800$  位元。

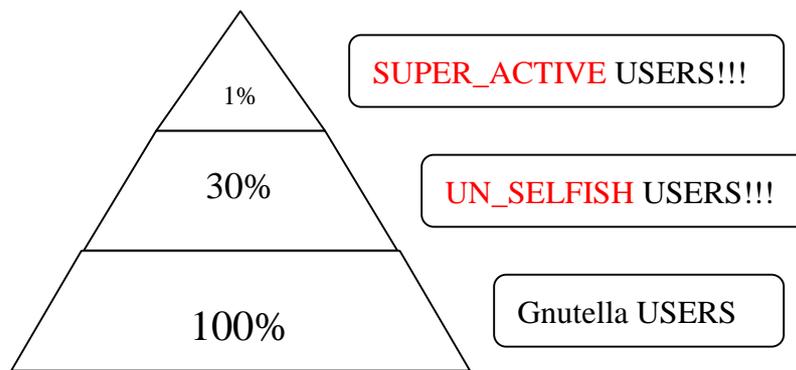
#### 2. 找其他 Peer：Pings 訊息

這部分，Pings 訊息的頻率約是 Query 訊息的兩倍。

#### 3. 回應其他 Peer：Pongs 訊息 / Query-hits 訊息 / Push 訊息

三種訊息加起來的量約相等於 Query 訊息的流量。

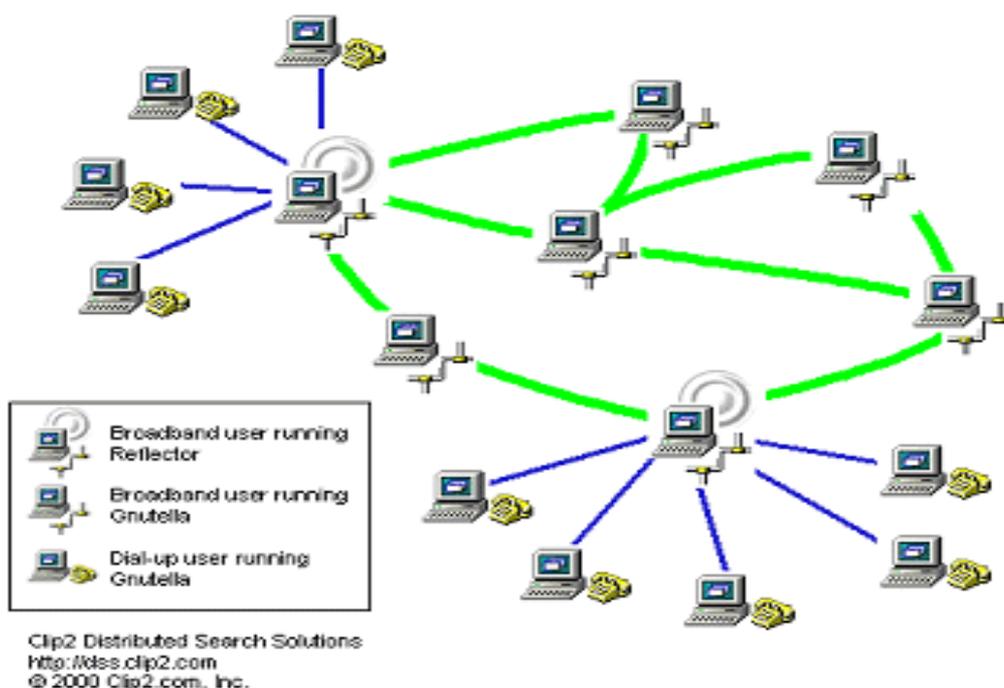
加總起來可以發現，這些流量為  $4 \times 16,800$  也就是每秒 67,200 位元，明顯超過撥接數據機的頻寬。附帶一題，有許多網站提供 Gnutella 使用者端的功能，讓使用者無須下載任何軟體即可線上搜尋下載。這些不僅增加網路的流量，也造成很多比例的使用者是「搭便車」(Free-rider)。請參考圖三。



圖三 Gnutella 使用者參與度

假設就算今天撥接數據機已經過時，這種延展性的問題是否就消失呢？不會，依然會發生在速度次等的族群中，譬如低速的 ADSL 或專線。因此，此延展性問題必須治本而非治標。茲整理出四個可能改進高流量問題的方向。

1. 從瓶頸發生點改進，因為撥接數據機雖然拖累整個網路流量，相對地也受害於網路壅塞，因此撥接數據機過度參與訊息的廣播和轉向，並不是件好事，而是應該只讓頻寬足夠的 Peer 來處理訊息封包，也就是所謂的 Proxy。目前 Clip2(<http://dss.clip2.com>)公司已經有現成免費軟體 Reflector 讓使用者下載，並成為網路的 Proxy，而撥接數據機則允許一條連線至該 Proxy，而不是開放許多連線接受其他 Peer 的訊息流量，免掉訊息封包造成的壅塞。請參考圖四。



圖四 Proxy 應用在 Gnutella 網路

2. 從人數方面改進，維持一定比例的族群大小，譬如每個 Gnutella 網路的人數控制在 5000 人以下。
3. 從程式方面改進，減少訊息的互動頻率。
4. 從協定改進，檢討欄位的必要性並適當增加一些欄位，對於有些使用者故意低報本身的頻寬，可以限制一定比例的下載速度頻寬比；對於有些使用者願意擔任 Proxy 可以提升其訊息的散播距離，也就是加入功過褒貶的制度。

## B. 安全性(Security/Trust)

在 P2P 的分享環境中，安全可以分為兩個部分：(1) 金流，即電子交易；(2) 資料流，如檔案分享。關於金流，多以 IPsec 從 IP 層保護或 SSL 從應用層下手來保障使用者隱私資料，進而搭配付費機制如 SET。而資料流，除了內容的適度保密外(這是個取捨，全部加解密也意味著吃掉更多的頻寬)，也有賴雙方互信機制的建立，但目前卻十分缺乏。在 P2P 分享的環境，是可以加入信任(Trust)來維持彼此的互信。也就是給予每位使用者一個名聲值，遵守固定的規範條文(*Codes of Conduct*)，慢慢建立自己的名聲，一旦違反規範，依據嚴重程度或三犯出局，將其名聲降到零。Peers 之間也可以互相散佈違反紀律的危險人物。

這裡的規範條文可以分為三種方式來進行：

1. 固定條文：只要加入 P2P 這個大家庭，就得遵守這些條文，不得異議。
2. 協調條文：由連線雙方互相同意對方的規範條文。
3. 學習條文：以所有人的條文之交集，視為最底線的條文。

## VIII. 未來議題

P2P 式網路架構的觀念，其實就是將點對點(Point-to-Point)的層次，從主從式架構(Client-and-Server)連結(connection)進一步成為 Peer 之間的連結。然而，觀念雖然簡單，戲法也人人會變，但巧妙各有不同。筆者試圖從一般使用者、企業，以及軟體整合三個角度來探討 P2P 的發展以及研究議題。

首先，從一般使用者的角度而言，他們所關心的是 P2P 的"sharing"(分享)能力，整個設計理念應該是這個 P2P 軟體能分享什麼(what)以及如何分享(how)，也就是說設計的 P2P 軟體要夠「好用」，而不只是一個不好玩，60 分及格的玩具。接著筆者以功能導向的思考方式，來探討使用者所在意的議題。

1. 全面性 (All-in-one)：結合多種功能於一身。
2. 延展性 (Scalability)：包括網路架構的存取速度及能負荷的社群大小。
3. 商業模式 (Commercial business models)：如何能在免費的內容上獲得加

值的報酬？計費與仲介需要做個拿捏。

4. 互通面 (Interoperability)：異質 P2P 軟體之間是否有共通標準 (standards)？異質 P2P 社群又如何做到資源互通？
5. 友善性 (Interface I/O)：簡單容易的使用者介面，而非繁瑣的專有名詞及凌亂的操作。
6. 安全面 (Security/Trust)：金流交易的安全性，以及資訊分享的正確性。
7. 法律面：P2P 網路強化了使用者與使用者之間的互通能力，但同時也容易涉及法律問題。一般而言，很難去正面的控制使用者行為，以 Freenet 為例，所傳輸的資訊皆經過加密，則可迂迴地讓業者本身避過法律問題 (因為無法得知使用者傳輸的內容)，讓使用者自律。

再者，從企業的角度而言，P2P 能帶來的最大利益，是在於"collaborating" (合作) 能力，其設計理念是在於如何讓同仁之間互相合作，共同工作，要能為企業的生產力加分，而這個 P2P 軟體才會「有用」，而非 90 分的好產品，卻不能幫企業的生產力加分，那也圖落無用武之地。因此，會有以下幾點需要考量。

1. 共通的文法 (Common technical grammar) 使每個個體能夠透過共通介面來互相連結。底下會依不同需求，提出業界的解決方案。
  1. 資料本身：以 XML (eXtensible Markup Language) 描述資料，而非 HTML。
  2. 需要 RPC (Remote Procedure Call) 等功能執行 Peer 端的程式：可以再搭配 W3C 所提出的 SOAP (Simple Object Access Protocol)。
  3. 需要描述服務 (service description)：如同描述資料本身一般，服務本身也是需要描述才能讓使用者辨識使用。可使用 W3C 所提出的 WSDL (Web Services Description Language)。
  4. 需要尋找服務 (service discovery) 或甚至包含 1-3 的完整解決方案，可參考微軟提出的 UDDI (Universal Description Discovery and Integration)。
2. 共通的 APIs
  1. 以微軟 (Microsoft) 為主的 Hailstorm
  2. 以昇陽 (Sun) 為主的 Jxta

最末，從軟體應用的角度，P2P 軟體也得面臨到"integrating" (整合) 的問題，就設計理念而言，產品和玩具之間，是需要做的拿捏，也就是有趣和實用之間是需要平衡。且在可期的未來，集中式 (如網頁存取) 和分散式 (如檔案分享) 的服務模式，也會面臨整合。

## IX. 結論

從 Napster 到 Freenet，可以發現分享的東西不侷限在 MP3 檔案；從眾多廠商相繼投入不同領域的 P2P，更可以瞭解 P2P 的應用是廣泛無限的。目前最受期

待的目標是取代 WWW 的地位，將 P2P 的技術成爲第三代的網際網路。Freenet 就是以這樣的目標自許，另外可能的 GnutellaNG(Next Generation) – gPulp(General Purpose Location Protocol)也是以這樣目標在努力。或許，最大的變革，莫過於搜尋引擎就在自己的電腦中，網頁連結是 On-demand 且沒有 Dead-links。

無論如何，純粹的檔案分享（如 Napster）已是昨日黃花，整合使用者和企業之間需求，兼顧正確的商業模式，才是真正的 P2P 的資源分享網路。

## **X. 參考資料：**

1. Peer-to-Peer Working Group, <http://peer-to-peerwg.org/>
2. Gnutella unofficial website, <http://gnutella.wego.com/>
3. Reports, specifications, numerical data of the Gnutella network, <http://dss.clip2.com/>
4. Oreilly's P2P Development Center, <http://www.oreillynet.com/p2p/>
5. gPulp, <http://gnutellang.wego.com/>
6. Freenet Project, <http://www.freenetproject.org>
7. Scoundrel Project, <http://scoundrel.on.openave.net/>
8. espra, <http://espra.net>