

對等式 (P2P) 網路技術

吳明蔚 林登達
Ming-Wei Wu, Ying-Dar Lin

摘要

隨著上網人數不斷的激增，寬頻網路的普及，早期廣泛採用的主從式架構，已逐漸不敷使用者在資料量及即時性的強烈需求，是以網際網路的連線模式正逐漸轉型，對等式網路的崛起扮演關鍵的推手。然而，此強烈需求目前正在臨許多問題，主要歸納有三點，其一最基本的連線問題，目前許多網路設備與對等式連線模式是相衝突的。緊接著，其二是能負荷多少連線使用者？對等式網路的運作方式及網路架構的延展性嗎？其三是能激發多少不同的應用層面？對等式網路又有哪些待解決的問題。可以預期的，只有對等式網路的相關問題獲得解決，網際網路才更能真正做到無遠弗屆。

關鍵字

P2P
對等式網路
Napster
Gnutella
延展性

1 · 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，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陸續召集舉辦，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、兩屆的於2002/3/7-8及2003/2/20-21舉行的國際對等式系統研討會(International Workshop on Peer-to-Peer Systems; IPTPS)。

2 · 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通分享。所有的Peer-to-Peer是架構在TCP/IP協定，使用TCP封包，而非UDP封包。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2-1；反之，Gnutella的分散式架構，請參考圖2-2。

2.1 Napster的架構

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端會將使用當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，並將結果回傳回來。

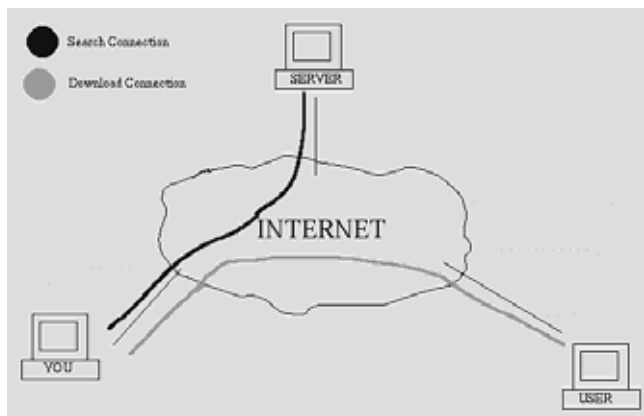


圖 2-1 Napster的架構及運作方式

3. **下載**：收到Server的結果後，你可以選擇想下載的檔案，並直接連線到對方(Peer)，進行Peer-to-Peer的傳送，而無須透過Server的介入。

2.2 Gnutella的架構

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。

如果需要知道目前有哪些起始點可以使用，可連線到這個網址：

<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，並直接連線到對方(Peer)，

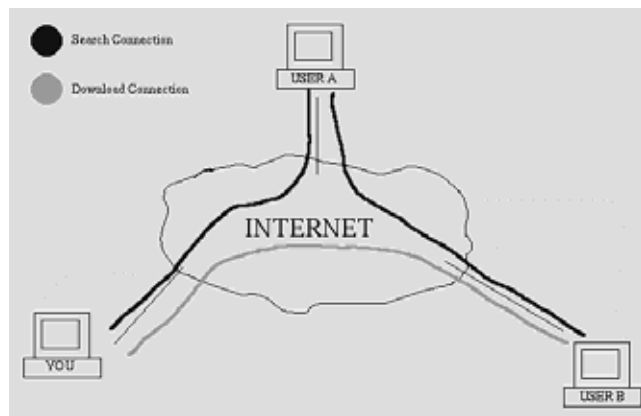


圖 2-2 Gnutella的架構及運作方式

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，則很有可能是對方有防火牆之後，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。

3 · 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，不限於檔案種類。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！儼然成爲一個去中心的檔案分享。檔案本

身會依據熱門程度，適當的移動或複製到需求最高的位置，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，可取消投票，因此不會有主觀的假投票發生。

三種檔案分享的分類表，請參考表 3-1。

當然，檔案分享絕不止 Napster 或 Gnutella 幾種，還有許多類似的應用軟體，茲分類整理幾種普遍的 Napster 分身於表 3-2。

4 · 目前問題與瓶頸

4.1 連線能力 (Connectivity)

如果說對等式網路增強了網際網路的透明度，讓使用者能互相「看到」分享的資源，那麼傳統的網路安全設備如防火牆 (Firewall) 與網路名

稱轉換器 (Network Address Translation ; NAT) 即是降低了網際網路的透明度，讓外面的使用者不能隨意看到內部的使用者。由於防火牆的設計是基於安全考量，一旦系統管理者決定阻擋對等式網路的連線，也是無可厚非。因此我們在這裡僅就 NAT 所造成的問題作討論。

NAT 的功用最主要的是將內部網路使用之虛擬 IP 位址轉譯為 Internet 上正式可用之合法 IP 位址。使用 NAT 的另一個好處是，由於隱藏 LAN 使用之 IP 位址，增加了網路使用上的安全性，並且藉由轉址功能，利用少數可用的 official IP address 來配置轉換給內部大量的虛擬 IP 位址，而不需一一配置固定的 official IP address，如此可解決 IP 位址不足的問題。

然而，NAT 雖然將內部網路使用之虛擬 IP 位址轉譯為 Internet 上正式可用之合法 IP 位址，卻會造成 NAT 之外的機器無法直接與內部

	Napster	Gnutella	Freenet
誕日期	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	統一統檔案分享
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	決定古你	決定古你	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
觀察別人下載	否	是	否

表 3-1 Napster vs. Gnutella vs. Freenet

排名	軟體名稱	優點	缺點
1	CuteMX	容易使用、線上試聽	N/A
2	Gnutella	功能強大的介面	搜尋太慢
3	SpinFrenzy Xchange	清楚目錄分類	下載太慢
4	Audiogalaxy	準確搜尋結果	結果太少、FTP協定
5	E-Donkey	多執行緒下載	檔案切片可能無法重組
6	KaZaa	資料量極多	

表 3-2 Napster 分身 (Napster-clone) 的評比

的系統通訊，這將會是P2P網路環境的障礙。

目前，這類的NAT設備所造成的P2P穿透(Penetration)問題，是P2P領域的熱門問題，由P2P Working Group也一直有在積極討論。主要可以分為下列幾種狀況：

單一(single)NAT情況(圖4.1-1)，即僅某一連線peer是位於NAT後，而另一連線peer則沒有躲在NAT之後。這類的問題十分好解決，目前是由後者發送PUSH request給位於NAT之後的連線peer，由此peer主動建立連線即可。

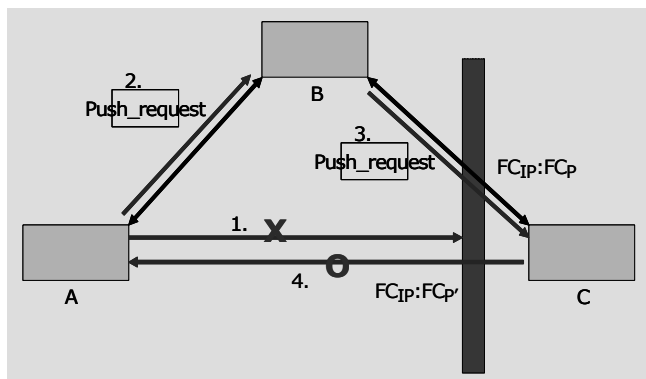


圖 4.1-1 單一 NAT

對立(opposing)NAT情況，即兩方連線peer皆位於NAT之後，沒有任何一方能夠將PUSH request直接送至對方，更遑論互相建立連線。為此，可延伸出兩種狀況，皆透過第三者peer端作為兩方訊息間接溝通的中間人：

此連線為UDP連線，如圖4.1-2，雙方先經過第三者交換NAT之IP位址及Port號碼，接著某一方需為另一方送一個封包讓NAT產生對映的資料，或俗稱打洞(hole punching)，如此對方

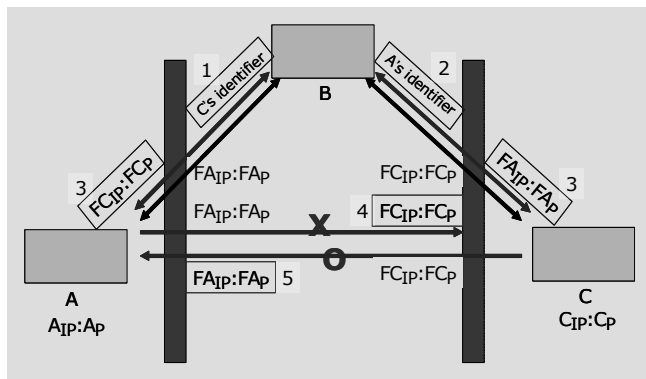


圖 4.1-2 對立 promiscuous-NAT，UDP連線

欲建立連線時才得以穿透(penetrate)，而P2P連線則得以建立。

此連線為TCP連線，如圖4.1-3，若為TCP連線，沿用上述方法時，會有問題，因為TCP連線需要3-way的handshaking，若對方還沒事先打洞好，則SYN封包過洞時，NAT會回RST封包，下次連線就變成port P加上某個常數，這會造成兩方的連線Port選擇不一致，而永遠無法互相打洞成功。為此，某一方需先送一個UDP封包作觸發(trigger)之動作，利用網路的latency，以UDP通知對方即刻傳送SYN封包，而對方則在UDP封包出洞後某一小段時間，才送SYN封包，如此可確保對方已經先行打洞。這種雙方互相送SYN封包的機制，又稱TCP simultaneous open，在RFC793中是有規範的。

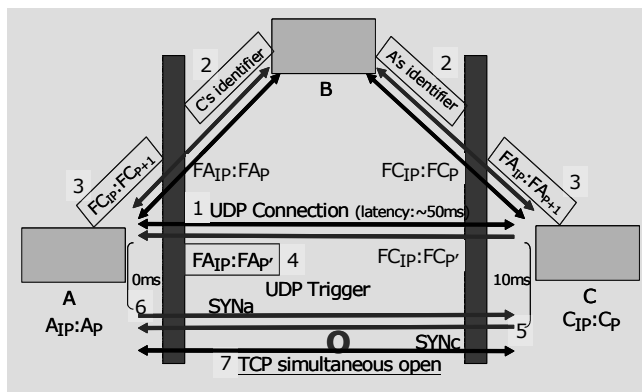


圖 4.1-3 對立 strict-NAT，TCP連線

除了上述適用於應用程式端的取巧方法之外，另一種普遍是直接透過第三者做為雙方的proxy，來建立三角關係的對等式連線。

4.2 延展性(Scalability)

在P2P的環境下，由於連線能力的解放，P2P網路內的使用者都可以互相建立連線，訊息量的傳遞就會相當可觀，尤其Gnutella的架構，當人數成長到萬人左右，擁塞狀況變得非常嚴重糟糕。Gnutella之父Frankel甚至表示Gnutella網路會無法負荷到5,000位以上的使用者。

在早期，Napster以集中式架構的主機來儲存使用者所分享的檔案資訊列表，是以資料的

搜尋並不需要大量的廣播詢問，而是使用者單純地詢問Napster主機，而主機回應正確的檔案位置，這類似以前的主從式架構，差異僅在於後續的檔案傳遞是兩個使用者互傳，即對等式連線。

到了Gnutella，其採取分散式架構，每個使用者自行維護一份檔案資訊列表。這所帶來的問題即是大量的訊息量，因為Gnutella所採用的方式是廣播，詢問至鄰近者的方式下，儘管搜尋的結果可能會收斂，卻造成整個網路的延展性很差。尤其當我們檢視網路上的某些使用者端，其連線頻寬有限，是否能負荷龐大的訊息量？

其實，有報告指出，Gnutella平均的查詢頻率約為每秒十次，每台peer最普遍是連線到另外三個peer。基本上，Gnutella流量可以分為三種狀況(查詢、找人、回應)，其包含著三種型態的訊息：

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(e.g. 56K)。附帶一提，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)。根據Gnutella的官方統計，1%的人提供絕大多數的檔案資源，而約30%的人仍願意提供資源，卻有69%的人是只下載，不分享。

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. 從瓶頸發源地改進，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。至於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4.2-1。

2. 從人數方面改進，維持一定比例的族群人

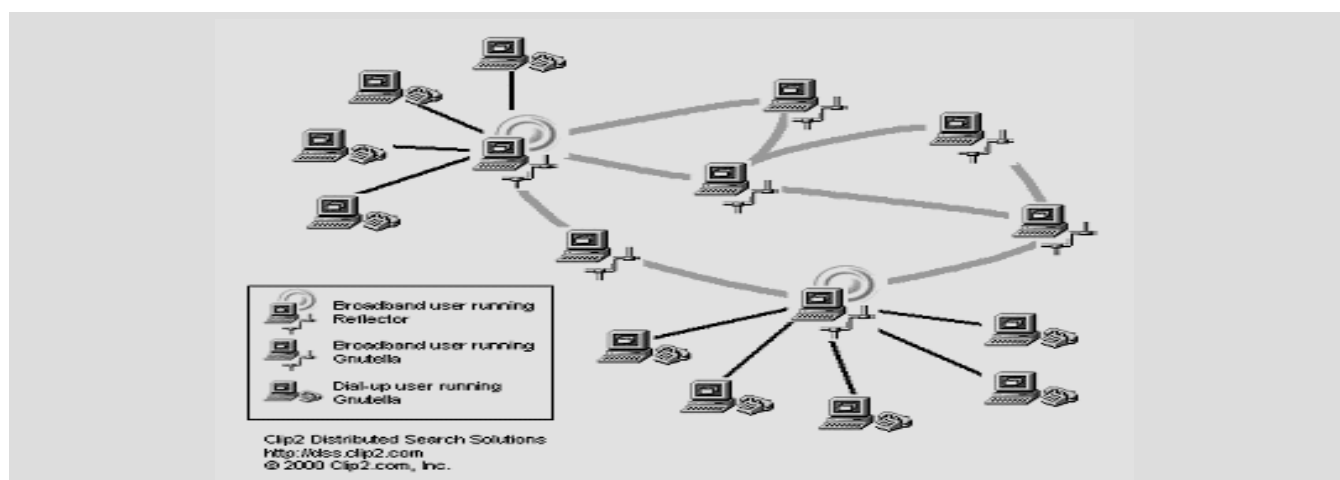


圖 4.2-1 Proxy應用在對等式網路

小，以Sun所提出的JXTA協定為例，其以範圍(Scope)的概念，界定社群的大小，避免對所有使用者任意發出廣播封包。

3. 從使用者端的程式方面改進，避免使用廣播，而是有條件的傳遞訊息，譬如設定訊息的TTL(time-to-live)來控制傳遞範圍，或使用本地端的快取機制來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。

4.3 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，除了內密的適度保密外(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，遵守固定的規條(Codes of Conduct)，慢慢建立自己的名聲，一旦違反規條，依據嚴重程度或三犯出局，將其名聲降到零。Peers之間也可以互相勸導違反紀律的危險人物。

從既有的網路安全技術來看，對等式網路在安全性方面可以有如下幾種作法：

1. 電子交易，以SSL做到應用層端點對端點的安全。對於XML資料，可輔以XML-Signature及XML-Encryption來做到更細部且更有彈性的安全保護。
2. 資料流，以IPSec做到傳輸層的安全。通常使用3DES加密以及SHA-1或MD5驗證演算法。
3. 檔案本身需以checksum或數位簽章等類似機制來驗證是否遭到竄改或假冒。

5 · 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，來探討使用者所注意的議題。

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，很難從法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 .Net

(2) 以昇陽 (Sun) 為主的 JXTA

(3) 以 IBM 為主的 WebSphere

最終，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。

6 · 業界應用

6.1 電玩/訊息網路

- CenterSpan (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。
- Jabber (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。

6.2 分散計算

- Applied MetaComputing (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。
- Datasynapse (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。
- Entropia (www.entropia.com) 致力於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。

6.3 管理/合作

- 惠普 (Hewlett-Packard) 計畫將推行目前多多的 e-speak 產品提供 peer-to-peer 上的應用。
- 由 Lotus Note 創始人 Ray Ozzie 所創辦的 Groove Networks (www.groove.net) 投入開發採購、批發、貨品管理及客服等應用軟體。
- 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。
- Quiq (www.quiq.com) 鎖定目前的前一千大企業 (Fortune 1000) 並提供虛擬 peer-to-peer 的社群環境。

6.4 檔案分享

- NextPage (www.nextpage.com) 讓企業內部的檔案伺服器能以 peer-to-peer 的模式工作。
- 在網路上流傳著微軟 (Microsoft) 有個代碼為 「Farsite」的 peer-to-peer 計畫。其目標是讓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」。

6.5 Peers 代理人/分散式搜尋引擎

- 從西門子出來的 WebV2 (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。

6.6 超級分散式

- vTrails(www.vtrails.com) 發展出 Full Duplex Packet Cascading(FDPC)，可以有效地減少網站面臨雜訊流量。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將網站雜訊點，分散到一連串的使用者層(tiers)之間作peer-to-peer的下載。由於同一台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。
- 2AM也有類似的：StreamingPeer

7 · 結論

從Napster到Freenet，可以發現分享的東西不再限於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。對等式網路已經擁有廣大的使用者數量，我們相信，在連線問題及延遲性問題獲得充分解決後，會有更多的使用者加入對等式網路的懷抱，而在安全性獲得普遍認同後，許多現有的由應用服務亦會逐漸轉型為對等式應用服務。

參考文獻

- [1] K. Sripanidkulchai, The popularity of gnutella queries and its implications on scalability, in O'Reilly's www.openp2p.com (Feb. 2001).
- [2] Bidirectional peer-to-peer communication with interposing firewalls and NATs, Peer-to-peer working group, Aug. 2001, Available: <http://www.p2pwwg.org/tech/nat/Docs/NATWhitePaper095.pdf>
- [3] Li Gong, "JXTA: a network programming

environment", IEEE Internet Computing, vol. 5-3, pp. 88-95, May 2001.

- [4] J.D.Touch, "Those pesky NATs", IEEE Internet Computing, vol. 6-4, pp. 96, July 2002.
- [5] Shih-Pyng Shieh, Fu-Shen Ho, et al., "Network address translators: effects on security protocols and applications in the TCP/IP stack", IEEE Internet Computing, vol. 4-6, pp. 42-49, Nov. 2000.
- [6] Gnutella : <http://gnutella.wego.com/>
- [7] Andy Oram, "PEER-TO-PEER Harnessing the Benefits of a Disruptive Technology," Published by O'Reilly & Associates, Inc.
- [8] Napster: <http://www.napster.com/>
- [9] Groove Networks: <http://www.groove.net/>
- [10] ICQ : <http://web.icq.com/>
- [11] Jabber : <http://www.jabber.org/>
- [12] Clip2: <http://www.clip2.com/>
- [13] O'Reilly P2P : <http://www.openp2p.com/>
- [14] Endeavors Technology's Magi P2P Collaboration Software : <http://www.endeavors.com/>
- [15] Folding@Home: <http://folding.stanford.edu/>
- [16] Peer-to-Peer Working Group : <http://www.peer-to-peerwg.org/>

作者簡介

吳坤壽



現為交通大學資訊科學所博士班二年級，是高速網路實驗室的成員之一，有豐富的安全開道器設備之測試經驗，是工研院交大網路測試中心的兼職工程師，2002年榮獲研華第四屆TIC100科技創新競賽之最佳資訊長暨決賽首獎。其個人研究領域包括網路安全及P2P對等式網路。

林昶達



現為交通大學資訊科學系教授及工研院交大網路測試中心主任，畢業於台灣大學資訊工程系，擁有加州大學洛杉磯分校資訊科學碩士學位，曾主持多項國科會及工研院計畫，2002年成立工研院交大網路測試中心。其個人研究領域包括網路協定暨演算法之設計、分析及實作，品質服務保證和內網網路。