以 P2P 為基礎的動態搜尋 Proxy Server 機制*

李敬江
元智大學資訊管理研究所
s937705@mail.yzu.edu.tw

林志麟
元智大學資訊管理研究所
jun@saturn.yzu.edu.tw

摘要

網際網路的快速發展帶動了資訊的分享及快速傳遞，然而某些組織因考量有些資訊可能被其造成傷害，進而對部分網路位址採取封鎖的機制，以管制並阻擋資訊的傳播及分享。為了解決網址被封鎖的問題，本研究以 P2P 的架構為基礎開發特殊的瀏覽器（JXTA Browser）及代理人程式（JXTA Proxy Agent）。在 JXTA Browser 方面提供動態搜尋 Proxy Server 的機制，透過 P2P 的傳遞方式，試圖搜尋可用的 Proxy Server，並與所尋得的 Proxy Server 連線呈現網頁內容。在 JXTA Proxy Agent 方面以附加功能的方式整合 Proxy Server，並設定其 IP、埠號(port)等資訊，使 Proxy Server 具備 P2P 的功能。當 JXTA Browser 發出請求時，將會接受來自 JXTA Proxy Agent 的 Proxy Server 資訊；若 Proxy Server IP 被封鎖無法連線時，JXTA Browser 將會自動設定下一個可用的 Proxy Server，以順利連線到被封鎖的網址。此外，本文針對 2 台 Proxy Server 模擬四種情境，分別測試不同的傳遞時間，以分析及探討實驗結果。

關鍵詞：P2P，Proxy，JXTA

1. 導論

隨著網際網路的蓬勃發展及全球網路使用人口急速成長，資訊傳播及呈現方式已轉移到網站，例如目前流行的 BLOG[2]日誌網站形式，可供資訊的分享及傳遞。然而，資訊傳遞方便迅速，機密訊息亦可輕易地藉由網際網路的傳遞而外洩。因此有一些組織，將一些非法資訊或是具有不願社會大眾所閱讀的資訊封鎖，使得使用者無法藉由一般的瀏覽器瀏覽及搜尋被封鎖的網站，即使網址已知也無法連結上，而導致網路上言論與著作的不自由。故如何將被封鎖的網站內容，藉由軟體的適當設計以突破此限制，使得資訊可以自由的取得及分享，便成了一個重要的課題。

提升網路運作效率的技術很多，其中藉由軟體提升網路傳輸效率的方法以使用代理伺服器（Proxy Server）居多。代理伺服器可以利用其內建的快取（cache）機制，有效降低不必要的網路傳送時間，達到提升傳輸效率的目的。瀏覽器設定代理伺服器的位址與連結埠後，若正在使用的代理伺服器位址被封鎖住，則無法提供連結欲連線的網站，而必須尋找另一個可用的代理伺服器，並再以手動方式完成其連線及取得網站資訊的工作。此外，所要連線的網路位址若被封鎖，也會造成無法連線的情形發生。

本文提出以動態方式搜尋代理伺服器的觀念，解決上述所衍生的問題。運用以 P2P 的技術及架構為基礎，設計自製的瀏覽器提供動態搜尋與設定代理伺服器的機制。並在未被封鎖的既有代理伺服器之下，運用內建的快取機制加速網頁執行的速度。代理伺服器透過附加功能的方式整合並執行所開發的代理人程式，以利使用者連上被遭封鎖的網站並擷取內容。

本文架構如下：第二節探討相關文獻，包括代理伺服器軟體的介紹、P2P 相關的重要技術、JXTA

＊本研究由國科會計劃贊助 計劃編號 NSC 93-2213-E-155-053

技術；第三節提出本研究的系統架構與方法，其系統架構分為客戶端及代理伺服器部分；第四節運用所開發的軟體模擬及分析實驗之數據。最後第五節根據本研究結果提出結論及未來可延伸之研究方向。

2. 文獻探討

本節介紹 Proxy 的概念及其軟體，並說明本研究應用的相關技術與文獻。Proxy 的概念為當某一瀏覽器透過 Proxy Server 瀏覽過某個網站上的網頁後，該網站資料就複製一份到 Proxy Server 的 Cache 中，當其他使用者需要瀏覽相同的網站時，可以直接從 Cache 中讀取網頁資料。故 Proxy Server 可以節省頻寬，及加快 Client 端讀取資料的速度。目前的 Proxy Server 種類繁多，以 HTTP Proxy Server 為例有 Brandgang[3]、MultiProxy[15]、Squid[19]等軟體。

Brandgang 軟體主要以 Java 語言開發，透過瀏覽器設定代理伺服器的方式連線，但是其效能不高且無提供代理伺服器認證安全上的機制。MultiProxy 軟體可以設定多個 Proxy Server 的位址及連接埠於設定檔中，載入 MultiProxy 後將會自動偵測目前所擁有的 Proxy Server 的連線狀態，但是當清單中的所有 Proxy Server 無法提供瀏覽器適當的網站連線時，需要以手動方式重新載入其他 Proxy Server 的清單。經過幾次測試後，才能讓使用者可以順利連線到所要求的網站。Squid 軟體為目前最多人使用的 Proxy Server，於 Unix、Linux 平台上執行，可處理 HTTP、FTP、Gopher 等通訊協定的要求；對於使用者的請求採取 non-blocking 的處理方式及支援 SSL 安全機制，但是若代理伺服器的網路位址被封鎖則無法提供網頁的擷取及下載。

以目前研究的相關文獻發現代理伺服器皆無提供動態搜尋 Proxy Server 的機制，且運行模式大多以主從式架構為主。然而根據 Frost& Sullivan 市調機構的研究報告[1]預測，目前 P2P 雖然在企業市場上還不常見，但到了 2007 年預計將有 620 萬企業用戶會使用具有 P2P 網路功能的企業級解決方案。因此可以預見未來許多的應用將以 P2P 架構為基礎，開發及提供多樣化的各項服務。P2P 架構[6][11][12][14][16][18]並非集中式結構，而是點對點的傳輸服務，使電腦同時擔任伺服器與使用者的角色。透過電腦間的直接連線，不需經過任何伺服器的中介處理，因此可以提昇網路效率，透過此種模式提供端點與端點的連線，達到更方便、快捷以及主動的與其他電腦端直接進行資源分享和訊息的交換。

目前有許多重要的 P2P 技術如 JINI[8][9]、JXTA[4][5][10][13][17]等。JINI 提供服務與裝置相互溝通的標準，以簡單的架構使服務可以很容易在網路內傳遞，由那些對服務有興趣的程式，自動建立跨平台的溝通管道。JXTA 取自 Juxtapose 的簡寫，是一套開放原始碼版本的 P2P 網路平台，可讓相同網路上的任何裝置，如手機、PDA、PC、伺服器等作資料的交換與聯繫。JINI 與 JXTA 最大的不同點在於 JINI 主要在區域網路內運作，雖然 JINI 可跨 LAN 通訊，但是需要透過網路上特殊的服務才行；而 JXTA 可以跨網域運作，主要為網際網路而設計，其程式和網路界限的關聯性非常低。二者差異請參表 2-1。

表 2-1、JXTA 與 JINI 的差異

	JXTA	JINI
目標	支援任何裝置的點對點開發	提供服務和裝置相互通訊的標準
技術	XML 技術	RMI 技術
運行範圍	區域網路、網際網路	區域網路
功用	非特定區域的軟體服務間之溝通	特定網域內的服務（如印表機）

本研究以 JXTA 技術實作 P2P 動態搜尋 Proxy Server 的機制，以下說明 JXTA 之基本概念：

- Peer：是一個虛擬的通訊點，相較其他 P2P 不同地方在於 JXTA 的 Peer 並不表示一個使用者，因為使用者可有多個點在一個裝置上運作，所以在 JXTA 中的 Peer 代表的是一個抽象化的使用者。
- Peer Group：整合多個 Peer 的方法，制定所要開放的特殊服務，公告於每一個群組成員，並可加入群組、退出群組及更新會員身分的功能。加入群組的好處在於端點傳遞及接收的訊息只侷限於群組中，使得非群組內的點不會接收到該訊息，因而可以有效減輕網路通訊的負擔，避免資源的浪費。此 Peer Group 類似目前 VPN(Virtual Private Network)[7]的構想。
- JXTA Message：以訊息的形式提供應用程式間傳遞所需要的資訊，其內容採用標準化的格式傳遞。
- Peer Pipes：是 JXTA 基本且重要的角色，提供點對點之間的虛擬通道，透過管道隱藏點對點之間實際連線的複雜性。
- Advertisement：一種描述 JXTA 訊息（Message）、Peer、Peer Group 或是 Service 的文件，可以協助交換 JXTA 網路上可用資源的相關資訊，其內容以 XML 標準的資料格式設計公告的內容。
- Gateway Peer：僅提供點與點之間轉傳訊息的服務，不負責處理 Peer 查詢的請求。
- Rendezvous Peer：提供 Gateway Peer 的功能與處理 Peer 間查詢的請求。Rendezvous Peer 也可以是查詢的中繼點，與其他 Rendezvous Peer 交換 Advertisement 以進行更近一步的查詢處理。

此外，在 JXTA 網路中，當 Peer 開始執行時，將會建立一個 cm（cache management）目錄，此目錄將會儲存由本機建立與網路上取得的所有公告。當 Peer 需要使用公告時，該目錄即扮演快取的角色，Peer 便不需至網路上搜尋公告。快取後的結果可以在 session 之間保存，若快取中無所需的公告，則必須向其他 Peer 要求下載。

3. 系統架構

本文針對網址及 Proxy Server IP 被封鎖的問題，提出以 P2P 的方法突破其被封鎖的網路位址。經由公告探索的服務機制，搜尋提供可用 Proxy Server 的 Peer；傳遞請求的公告以取得 Proxy Server 的相關資訊，呈現被封鎖網址的網頁內容。本研究提出兩種系統流程，分述如下：

第一種流程在客戶端方面透過 JXTA Browser 自動尋找可用的 Proxy Servers 清單資訊，而代理伺服器方面則啟動 JXTA Proxy Agent Daemon 設定 Proxy 的 IP、埠號等資訊，以等待接收 JXTA Browser 的請求。當 JXTA Browser 搜尋到網路上的 JXTA Proxy Agent 後進行點對點的通訊，回傳及自動設定可用 Proxy Server 的資訊與呈現適當的網頁。然而若目前使用的 Proxy Server 無法提供網頁的正常連線及瀏覽時，將會從 Proxy Servers 清單資訊中，挑選下一個可用的 Proxy 作為設定使用，其 JXTA Browser 之系統流程如圖 3-1。

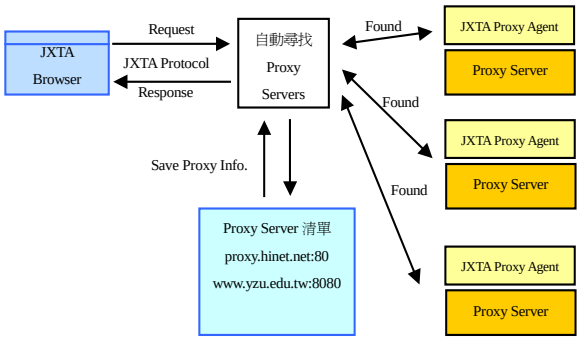

圖 3-1 JXTA Browser 之系統流程圖

而第二種流程在客戶端方面採用一般的 Browser，手動設定 JXTA Proxy Server 所在的網址及連結埠。JXTA Proxy Server 提供 Proxy Caching 的機制，當接收到 Browser 的網頁需求時，查看此

表 3-1、兩種架構的優缺點

	JXTA Browser 方式	JXTA Proxy Server 方式
優點	執行 JXTA Browser 後，即可直接連線，不需自行設定 Proxy Server 的 IP。	使用一般的 Browser 並設定 Proxy IP、連結埠即可運作。
	分散式架構。	集中式及分散式架構的結合。
	當無法連線時，會自動啟動搜尋可用 Proxy Servers 的機制。	當無法連線時，會自動搜尋可用的 JXTA Proxy Server。
	具有 Proxy Servers 的清單庫，作為備用 Proxy Server。	兼具快取機制及 P2P 功能的 Proxy Server。
缺點	需要額外執行 JXTA Browser 軟體。	網頁內容的擷取及傳輸速度較慢。
	若無法在所提供的 Proxy Servers 清單中協助連線到某網頁時，需等待搜尋以取得新的 Proxy Servers 清單。	JXTA Proxy Server 的網路位址及埠號有可能被封鎖，而無法提供服務。
	需設定代理伺服器端的 Proxy Server 組態設定檔。	需要實際開發 Proxy Server，實作上較不易。

一網址是否可以連線，若可以則回傳網頁內容，並將其內容複製一份到 Cache 中；當無法處理請求時，則自動搜尋其他可用的 JXTA Proxy Server 尋求協助，以進行點對點的通訊及要求傳遞網頁內容於原本的 JXTA Proxy Server，其 JXTA Proxy Server 之系統流程如圖 3-2。

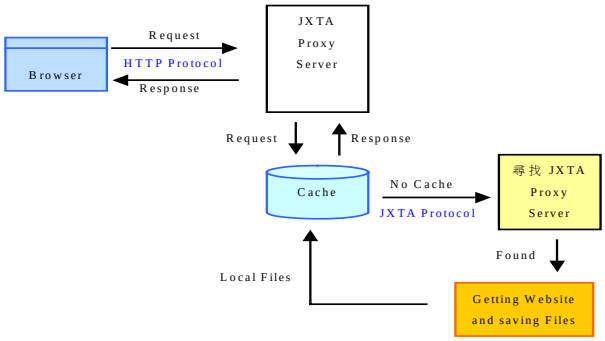

圖 3-2 JXTA Proxy Server 之系統流程圖

根據上述所提的兩種流程方式，各有其優缺點如表 3-1 所列。採用 JXTA Proxy Server 流程必須深入研究一般 Proxy Server 所提供的功能及內部實際建置的運行步驟，且需要配合相關的協定以利正常的運作，故實作上的困難較高。因此本研究只對 JXTA Browser 系統流程做進一步的研發。在此架構下，不需額外開發 Proxy Server，只需將 JXTA Proxy Agent 以附加的方式結合既有的 Proxy Server，並設定 Proxy Server 的 IP、埠號等資訊於組態設定檔案中即可。此外，藉由 Proxy Server 本身所提供的快取機制可以加快 client 端讀取網頁的速度，在系統開發難易度及效率方面皆能有所兼顧。

以下詳細介紹以 JXTA Browser 流程架構為基礎的實作方法，如圖 3-3 所示，本系統架構分為二部分：客戶端部分與代理伺服器部分。在客戶端方面建立 Search Proxy Peer 傳遞與接收訊息公告，而代理伺服器方面建立 Proxy Peer 提供 Proxy Server 資訊的服務，茲分述如下。

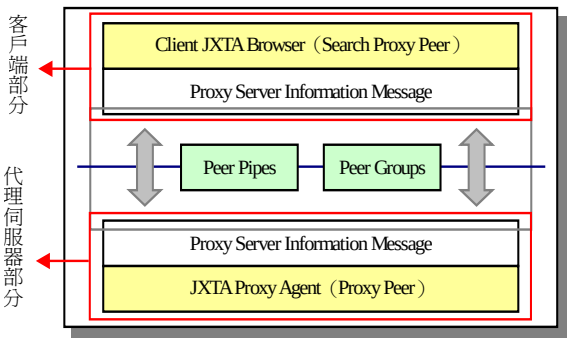

圖 3-3 系統架構圖

3.1. 客戶端部分（Client JXTA Browser）

客戶端部分提供動態搜尋可用代理伺服器的機制，將取得的代理伺服器之相關資訊儲存，並自動為 Browser 設定可用的代理伺服器，以利使用者可以連上被封鎖的網站，其流程如下：

1. 當瀏覽器接受到網址時，將會嘗試是否經由 HTTP 協定可以正常連線，若無法連線則透過 JXTA 提供服務。
2. 尋找網路上的 ProxyNet 群組，依據公告的探索機制查看 Search Proxy Peer 是否已經尋得該群組並加入，若尚未尋得則發佈公告及加入該群組。
3. 建立 Search Proxy Peer 的管道公告，設定管道的名稱、識別碼及類型為 unicase，表示此點的唯一性，並透過此公告通知同一群組中有新的 Peer 加入。
4. 透過管道公告尋找群組中提供 JXTA Proxy Agent 服務的 Peer，和這些 Peer 聯繫並建立輸出的管道，請求 Proxy Server 的資訊。
5. 儲存從 Proxy Peer 取得的 Proxy Server 資訊，嘗試網頁是否可以連線。

客戶端部分由「Proxy Server Information Message」、「JXTA Peer Pipes Communication」及「JXTA Dynamic Search Proxy Agent」構成如圖 3-4，其功能分述如下：

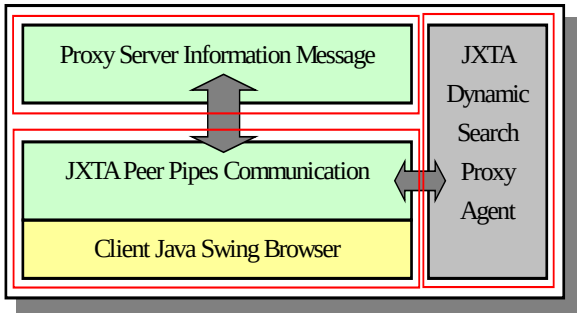

圖 3-4 客戶端部分架構圖

- Proxy Server Information Message：主要負責描述 Search Proxy Peer 與 Proxy Peer 所需的公告內容，是以 XML 標準的資料格式包裝公告中的訊息內容。由於公告被尋得時，將會被存放於本地端的 cm 資料夾中，故日後需要再一次搜尋公告時，將可以減少搜尋公告的時間。依公告處理的內容可以分為群組及管道公告，群組公告負責尋找名稱為 ProxyNet 的公告，透過此公告允許 Search Proxy Peer 端點加入到 ProxyNet 群組。管道公告負責描述與 Proxy Peer 間往返通訊的內容，透過識別 Request、PROXY_HOST、PROXY_IP、PROXY_PORT 等標籤名稱以取得對應的值。
- JXTA Peer Pipes Communication：建立及處理 Search Proxy Peer 與 Proxy Peer 間的輸出入管道。客戶端分別建立輸出、輸入二個單向管道，輸出管道負責建立 Search Proxy Peer 的管道公告，定義端點的名稱與傳遞的內容，以傳送請求之訊息於 Proxy Peer。輸入管道負責解譯含有 Proxy Server 資訊的公告內容，儲存 Proxy Server 的資訊以減少日後搜尋的時間。
- JXTA Dynamic Search Proxy Agent：Client Java Swing Browser 搜尋網路上名為 ProxyNET 群組的公告，若公告存在本機則表示已經加入該群組，反之則發送遠端探索的公告。在加入群組後，建立管道公告通知群組中的端點，尋找可用的 Proxy Peer。透過 JXTA Peer Pipes Communication 建立溝通的虛擬通道，將 Proxy Server 資訊傳回並儲存，以利 Client Java Swing Browser 設定 Proxy Server 資訊並嘗試連結網頁。

連結網頁時，有兩種情況需要考量；其一為請求的網頁是否真正存在，其二為所儲存的 Proxy Server 皆無法提供連線。為考量上述情況，在此提

供搜尋 Proxy Server 的次數限制；當在三次搜尋期間所尋得的 Proxy Server 皆無法提供網頁的正常連線及顯示時，瀏覽器將會呈現無法連線的資訊；這種方式可以減少等待網頁連線的時間及提供有限的探索服務機制。

3.2. 代理伺服器部分（JXTA With Proxy Server）

代理伺服器部分主要整合非封鎖區域內的既有代理伺服器，設定 Proxy Server 的組態設定檔案，包含 Proxy Host、Proxy IP 及 Proxy Port。當 Proxy Port 無指定時，以 3128 為預設值，設定完成後即啟動 JXTA Proxy Agent，提供接收與傳遞 Proxy Server 資訊的代理人服務，其流程如下：

1. 當 JXTA Proxy Agent 啟動時，根據公告的探索查看網路上是否存在 ProxyNet 群組，若存在則加入，反之則建立及加入新的 ProxyNet 群組。
2. 建立 Proxy Peer 管道公告的步驟與客戶端部分大致相同，不同者在於管道名稱以組態設定檔案中的 Proxy Host 命名。
3. 經由管道公告建立輸入管道，等待 Search Proxy Peer 的請求。
4. 當 Proxy Peer 接受到請求時，建立輸出管道並傳遞 Proxy Server 的資訊於 Search Proxy Peer。

代理伺服器部分由「Proxy Server Information Message」、「JXTA Peer Pipes Communication」及「JXTA Proxy Agent」構成，如圖 3-5。由於 Proxy Server Information Message 及 JXTA Peer Pipes Communication 的處理流程與客戶端大致相同，相異之處在於輸入與輸出管道接受資訊的不同，故不再詳加描述。而 JXTA Proxy Agent 的功能描述如下：

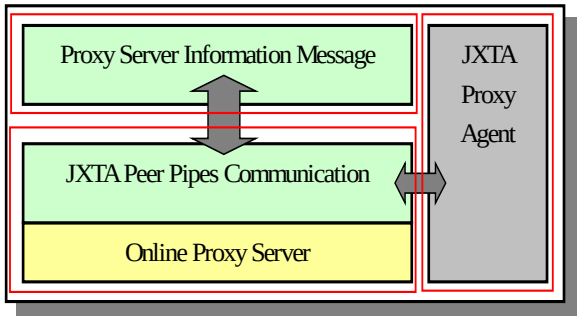

圖 3-5 代理伺服器部分架構圖

- JXTA Proxy Agent：根據 Online Proxy Server 的資訊設定完成 Proxy Server 組態設定檔案後，載入相關的 Proxy Server 參數。搜尋本機端及遠端的公告查看是否存在 ProxyNet 群組，若無則自動建立 ProxyNet。經由 JXTA Peer Pipes Communication 建立溝通的輸出入管道，以傾聽管道的方式等待請求。當接受到 Client Java Swing Browser 要求時，建立及傳遞具有 Proxy Host、Proxy IP 及 Proxy Port 標籤內容的管道公告。

由於在點對點網路上的延遲時間可能比傳統網路長很多，故在此採用三次探索公告的方式及設定訊息回傳的有限時間，等待公告取得後再進行下一步驟。當某個點在有限時間內無回應時，則繼續發佈新的公告探索新的點，若在有限次數內還是無回應則表示此點不存在。下一節將透過實驗進一步分析此一架構之效能。

4. 系統實驗

本實驗模擬四種情境，並分別對「尋找及加入 ProxyNet 群組」、「搜尋 JXTA Proxy Agent Peer」、「Proxy Server Information Message 的請求與接收」、「顯示網頁」四個階段，測試其時間與分析實驗結果。

4.1. 實驗設計

代理伺服器端部分提供二個具有 JXTA Proxy Agent 服務的端點，分為 Proxy Server1 位於 proxy.mis.yzu.edu.tw：3128；Proxy Server2 位於 proxy.yzu.edu.tw：8080，而客戶端執行 Client JXTA Browser。根據提供 JXTA Proxy Server 服務的點之存在與否分為四種模擬情形，並以 HTTP 直接連線的方式為比較基準點，其實驗設計如表 4-1。

表 4-1、實驗設計

	情境模擬
基準點	直接透過 HTTP 方式連線及呈現網頁
實驗一	Proxy Server1 Alive / Proxy Server2 Alive。
實驗二	Proxy Server1 Alive / Proxy Server2 Died。
實驗三	Proxy Server1 Died / Proxy Server2 Alive。
實驗四	Proxy Server1 Died / Proxy Server2 Died。

在 P2P 網路中主要透過集結點協助遠距離的探索，但是集結點通常都會存在於不同的網域或子網路內，故須在每個 Peer 的 JXTA PlatformConfig 設定檔視窗中，設定相同的 Rendezvous Seed Peer（如圖 4-1），以確保每個 Peer 都可以透過相同的 Rendezvous Peer，看到彼此存在的 Peer 或群組。

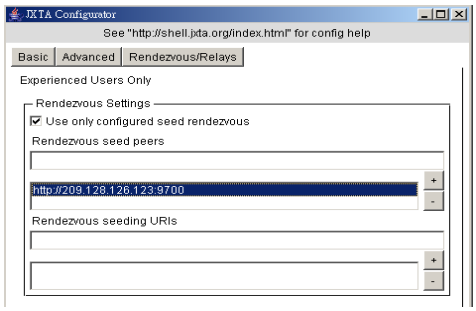

圖 4-1 設定 Rendezvous Seed Peer

4.2. 實驗結果

實驗一：在 Proxy Server1、Proxy Server2 二者皆存在的情況下進行，各階段測試的時間如表 4-2。

瀏覽器接收到網址時，透過 HTTP 協定即可直接連線並顯示網頁；其花費的時間為 0.187 秒，因無使用 JXTA 服務，故階段一至階段三的時間皆為 0 秒。在透過 JXTA 服務尋找可用的 Proxy Peer 方面，由於本機端的公告尚未存有 ProxyNet 群組的資訊，故需要遠端探索群組公告，因此在階段一「尋找及加入 ProxyNet 群組」花費 2.47 秒的時間。當加入群組之後便搜尋可用的 JXTA Proxy Agent Peer 及進行連線溝通，分別花費 0.298 與 0.485 秒，顯示網頁則與透過 HTTP 協定連線的時間並無顯著的差異，但是經由 JXTA 搜尋服務所耗費的總測試時間卻比透過 HTTP 協定超過 3.285 秒。

表 4-2、Proxy Server1、Proxy Server2 皆存在之各階段測試時間

	階段一	階段二	階段三	階段四	總時間
透過 HTTP	0	0	0	0.187	0.187
透過 JXTA(一)	2.47	0.298	0.485	0.219	3.472

實驗二：為 Proxy Server1 存在，而 Proxy Server2 不存在的情況下進行實驗，其實驗結果如表 4-3。

在本地端的公告發現有存在二個 Proxy Peer，故與找到的 Proxy Peer 進行連線，但是由於 Proxy Server2 實際上並不存在，因此在階段二搜尋 JXTA Proxy Agent Peer 方面耗費 30.313 秒的時間。當找到可用的 Proxy Server1 時，則在進行點對點的溝通連線及顯示網頁所耗費的時間並無顯著的差異，但因搜尋 JXTA Proxy Agent Peer 階段額外耗時，故總測試時間需要 33.063 秒。

表 4-3、Proxy Server1 存在，Proxy Server2 不存在之各階段測試時間

	階段一	階段二	階段三	階段四	總時間
透過 HTTP	0	0	0	0.187	0.187
透過 JXTA(一)	2	30.313	0.375	0.375	33.063

實驗三：為 Proxy Server1 不存在，而 Proxy Server2 存在，但是其無法提供正常服務的情況下進行實驗，其實驗結果如表 4-4。

因尋找及加入 ProxyNet 的群組在第一次搜尋時，已經將群組公告存於本機端，故第二次及第三次的搜尋時間皆比第一次少於 1.9 秒。此外亦同實驗二，由於 Proxy Server1 並不存在，故階段二搜尋 JXTA Proxy Agent Peer 需要耗費較多的時間。經過三次搜尋可用的 Proxy Peer 後，因 Proxy Server2 仍然無法提供正常的 Proxy Server 服務，故在階段四顯示網頁的測試時間皆為空值，三次的總測試平均時間為 31.203 秒。

表 4-4、Proxy Server1 不存在，Proxy Server2 存在之各階段測試時間

	階段一	階段二	階段三	階段四	總時間
透過 HTTP	0	0	0	0.187	0.187
透過 JXTA(一)	1.985	30.281	0.25		32.516
透過 JXTA(二)	0.046	30.5	0.172		30.718
透過 JXTA(三)	0.062	30.125	0.188		30.375

實驗四：為 Proxy Server1、Proxy Server2 皆不存在的情況下進行實驗，其實驗結果如表 4-5。

第一階段所發生的情形如同實驗三，於第二、三次耗費的時間皆比第一次減少約 7 秒。由於所提供的二個 Proxy Peer 皆不存在，因此在第二階段所耗費的時間約為實驗三之第二階段的 2 倍，即為 60 秒。因找不到提供 Proxy Server 服務的 Proxy Peer，故在第三階段及第四階段的測試時間皆為空值。在總測試時間方面，耗費約 61 秒。

表 4-5、Proxy Server1、Proxy Server2 皆不存在之各階段測試時間

	階段一	階段二	階段三	階段四	總時間
透過 HTTP	0	0	0	0.187	0.187
透過 JXTA(一)	0.704	60.843			61.547
透過 JXTA(二)	0.016	60.14			60.156
透過 JXTA(三)	0.031	60.36			60.391

4.3. 實驗結論

由實驗一可知當搜尋的 Proxy Peer 皆存在時，除了第一階段尋找、加入 ProxyNet 群組的時間花費較多外，其餘階段皆只花費不到 0.5 秒的時間即可完成。從實驗三、四的結果分析可得知，在階段一中當 ProxyNet 群組已經被找到並加入時，可以有效改善日後搜尋群組的時間。

在實驗二至四的分析結果中發現總測試時間主要被搜尋 JXTA Proxy Agent Peer 階段所影響，以致於間接影響到點對點的通訊及顯示網頁的測試時間。原因在於當某個 Proxy Peer 暫時離線時，由於公告本身內建的有效時間尚未到期，無法立即更新公告而導致舊有的公告依然存在，但是 Peer 卻不存在的情況發生。針對上述的情形本系統建立錯誤後重試的機制，透過週期性嘗試三次的方式探索公告與尋找公告中的 Proxy Peer，因此造成總測

試時間較長的情況發生。

5. 結論與未來展望

本研究主要針對網路位址在被封鎖的情況下，如何透過 JXTA Browser 動態選擇 Proxy Server 以達到解決網路位址被封鎖的問題。由實驗一的結果可以得知當取得的 Proxy Peer 存在且正常提供服務時，使用者將可得到最大的效益。當使用 JXTA Proxy Agent 服務的點愈多時，表示可以提供可用 Proxy 的機率將會愈大，故會減少網站連線的總時間。此外在具有 Proxy Server 環境下執行 JXTA Proxy Agent 時，需要於 JXTA Proxy Agent 中設定 Proxy Server 的資訊，使得 Proxy Peer 可以被其它點探索與溝通。

關於藉由 JXTA 技術開發 P2P 相關應用之議題，仍有許多的發展空間。目前也有不少學者朝此方向努力，如提出以 JXTA 技術擷取及傳遞遠端網頁內容。在此提出幾點後續研究之方向：

一. 在 JXTA 瀏覽器方面可以採用 Plug-in 的方式，將 JXTA Browser Agent 整合一般的瀏覽器，而不需要執行特製化的瀏覽器。透過整合功能完善的瀏覽器，可以讓使用者更方便操作及增加便利性，加上 Java 的跨平台特性可以輕易的執行於任何平台上。

二. 本實驗環境僅針對同一網域進行測試，未來將可考量以不同網域測試其效能及耗費的時間。

三. 由於本實驗對象僅包含 2 個 Proxy Peer、1 個 Search Proxy Peer，故無法完全表達網路上多個 Peer 的測試時間，未來可以考量模擬更多的 Peer 讓實驗結果更精準。

四. 後續研究可以發展真正的 P2P Proxy Server 軟體，以供實務上使用。

6. 參考文獻

[1] 通訊雜誌,"第 128 期 2005 年 7 月號",HTTP://www.cqinc.com.tw/grandsoft/cm/128/abi.htm

[2] Blogger HTTP://www.blogger.com/start，BlogMarks

[3] Brandgang HTTP://www1.tip.nl/~t515027/brandgang/

[4] Daniel Brookshier, Darren Govoni, Navaneeth Krishnan, Juan Carlos Soto, JXTA：Java™ P2P Programming,Person Education Inc.2002.

[5] Emir Halepovic, Ralph Deters, Building a P2P Forum System with JXTA, 2002 IEEE.

[6] Guntala HTTP://www.gnutella.com/

[7] Internet.com,HTTP://www.webopedia.com/TERM/V/VPN.html

[8] Jini.org,HTTP://www.jini.org/

[9] Jini Network Technology, HTTP://www.sun.com/software/jini/

[10] JXTA.org, HTTP://www.JXTA.org

[11] Li Gong , "JXTA:A Network Programming Environment" in Prco. IEEE Internet Computing , Volume: 5 Issue: 3 , Page(s): 88 -95, 2001.

[12] Limewire. HTTP://www.limwire.org

[13] Li Gong, JXTA: A Network Programming Environment, 2001.

[14] Matei, R.; Iamnitchi, A.; Foster, P., "Mapping the Gnutella Network" in Proc. Internet Computing, Volume: 6 Page(s): 50 -57, 2002.

[15] MultiProxy HTTP://www.multiproxy.org/

[16] Napster. Napster. HTTP://www.napster.com，Napster

[17] Nico Maibaum, Thomas Mundt, JXTA: A Technology Facilitating Mobile Peer-To-Peer Networks”, 2002.

[18] REILILY. Hello JXTA! HTTP://www.onjava.com，Raffi Krikorian，04/25/2001

[19] Squid HTTP://www.squid-cache.org/