以 Java Applet 為基礎的網路掃描代理人

賴守全
銘傳大學電腦與通訊工程學系
sclai@mcu.edu.tw

郭文曲
銘傳大學資訊網路處
wckuo@mcu.edu.tw

摘要

網路通訊埠掃描是網路管理的一項重要工具，特別是當使用者因為商務或旅遊而移動到一些較不熟悉的網路環境時，透過通訊埠掃描，可了解從所在的位置，到所欲到達的網路伺服器端，是否有任何中間經過的網路，因封鎖了特定的通訊埠而造成使用上的障礙。在本文中，我們設計並實作了一個以 Java Applet 為基礎的網路掃描代理人，透過這個代理人，使用者不需另行安裝掃描軟體，網路管理者也不需移駕至使用者端，即可釐清從使用者端，到網路管理者所在的網路端，是否有任何通訊上的障礙，以加速網路障礙的排除，亦或作為確認網路安全防護是否符合預期的測試工具。經實務證明，這個運用動態伺服程序管理及 Java Applet 跨平台特性的網路掃描代理人，是個網路管理工作上的好幫手。

關鍵詞：網際網路、網路管理、網路安全、網路掃描、網路代理人。

1. 前言

網路技術的快速發展，讓網際網路普及到世界的各個角落，各種新興的網際網路應用，更是推波助瀾地，讓網際網路逐漸成為生活中的一部分。當各式網路應用蓬勃發展的同時，隨之而來的網路犯罪事件，也讓網路安全成為當前網際網路的重要議題。

因應網路安全的威脅，各式的網路安全產品也逐漸安裝於網路供應商，公司行號，甚至個人電腦上面。因此，當一個網際網路應用在使用者端啟動時，除了要經過各式各樣的網路中繼設備(如：網路存取點，網路交換器，網路路由器等)，以建立一條從使用者端到服務伺服器端的網路通道外，更要穿透通道中間各式網路安全設備的層層關卡，方能順利連線到伺服器端，取得所需要的服務(如圖 1)。

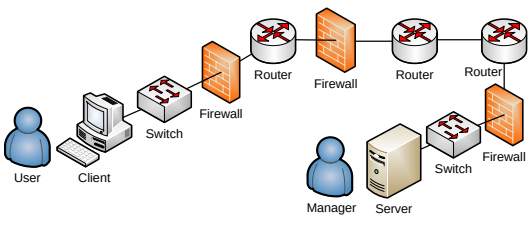

圖 1 網路服務基本連線架構

然而，在網路安全的考量下，通道中所經過網路(以下簡稱中繼網路)的網路管理者通常並不會公布其網路安全設備的建置及各項安全設定。或者，某些管理者使用網路安全設備的預設值來進行建置與設定，可能也並不清楚其所建置網路安全設備實際產生的效應。再者，就算某個中繼網路的管理者，公布其網路安全設定的內容，對一般普羅大眾的使用者來說，並不清楚其連線所經過的中繼網路有哪些，更遑論去查詢其所經過中繼網路的網路安全設定。因此，對大多數的使用者來說，除了 Web 瀏覽通常可以保持暢通之外，中繼網路的網路架構，宛如一個超大型黑盒子(如圖 2)，甚難掌握其實際運作概況。

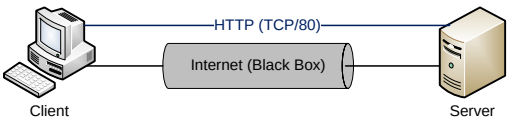

圖 2 網路連線黑盒子架構

當中繼的網際網路變成一個黑盒子的時候，對終端的使用者，以及提供服務端的網路管理者來說，當網路服務不如預期順暢的時候，網路連線障礙的排除，就會成為分別於黑盒子兩端的使用者和管理者的共同惡夢。確認連線的兩端可以順利的建立網路連線，是排除網路障礙的首要工作，因此，網路管理者必須引導使用者，從使用者所在的網路端，對伺服器端進行網路連通性的掃描測試(即通訊埠的連通掃描)，以釐清網路服務所必須使用的通訊埠是否遭到阻隔。

要從使用者端對伺服器端進行網路掃描，可以在使用者端的電腦上安裝具通訊埠掃描功能的軟體，即可達成目標。但是，要使用者在電腦上安裝這樣的軟體，並解釋該軟體掃描後的結果，在實務經驗上，考慮普羅大眾的使用者對電腦作業平台及網際網路的熟悉度，這並不是個輕鬆簡單的工作。另一個可能的方式，是網路管理者移駕到使用者的現場，進行軟體的安裝，並執行通訊埠的掃描工作。但在實務上，網路雖然無國界，但人的移動卻實際受限於地理屏障，有時候管理者並無法立刻移駕到使用者所在的位置，因此，我們需要一個不需要使用者安裝，又具備跨平台能力的網路掃描代理人，來代替網路管理者到達使用者的電腦現場，去執行網路掃描的工作。

在本文中，我們提出一個以 Java Applet [1]為基礎的網路掃描代理人系統，這個代理人可以透過使用者的瀏覽器下載到使用者的電腦，並自動對遠端伺服器的通訊埠進行掃描，並將掃描的結果同時呈現在使用者端及網路管理者端。因為 Java Applet 可以在各種支援 Java VM (Java Virtual Machine) [4]的作業平台上執行(如：Microsoft Windows、Mac OS、Linux、FreeBSD)，因此代理人可以移駕至各種使用者可能使用的作業平台上，代理網路管理者，順利完成通訊埠掃描的工作。

以下說明本論文的架構。在第二節中，我們將詳細說明網路掃描代理人的設計原理與架構，並在第三節，介紹我們如何實作這個掃描代理人系統。在第四節，我們將敘述這個系統的實驗成果及相關問題的討論。最後，第五節為本文的結論。

2. 網路掃描代理人的設計與架構

在本論文中，我們所考慮的網路環境，是使用者移動到他(她)所不熟悉的網路環境，欲使用某單位(如：學校或公司)所提供的網路應用服務，卻因故無法使用除了 Web 瀏覽以外的服務(例如：SMTP、POP3、Telnet、FTP、VPN 等)，因而向該單位的網路管理者尋求協助。當網路管理者確認提供服務的伺服器一切正常時，在不清楚使用者所在地之電腦網路環境的情況下，首要的工作就是確認使用者可以順利連接到伺服器提供服務的通訊埠。特別是在網路安全日益受到重視的今天，如果中繼網路的任何一個管理者封鎖了該服務所必須使用的通訊埠，就會造成網路服務無法正常使用。

網路管理者可以請使用者安裝網路掃描軟體來協助釐清狀況，但從實務經驗上，我們發現可能會面臨下列幾個問題：(1)使用者對軟體的安裝並不熟悉、(2)所提供的軟體並不支援使用者所使用的作業系統平台、(3)使用者不熟悉網路掃描軟體的操作、(4)使用者無法正確解讀掃描後的結果。然而，通常網路管理者無法立刻移駕到使用者所在的電腦端，因此我們考慮設計一個網路掃描代理人系統，讓網路管理者可經由網路派送網路掃描代理人到使用者端的電腦去進行網路掃描工作，如此，網路管理者不須移動到使用者端，亦不需麻煩使用者安裝網路掃描軟體，即可輕鬆獲得從使用者端進行網路掃描的結果。

為順利達成網路掃描之代理工作，網路掃描代理人應具備下列能力：

1. 具備網路通訊埠掃描功能。網路掃描代理人的主要目標在於依照指定的需求，完成通訊埠的掃描工作，因此網路代理人必須能夠連線至伺服器端指定的通訊埠，並確認連線是暢通的。
2. 可自動安裝於使用者端之電腦。對不熟悉電腦軟體安裝的使用者來說，軟體的下載及安裝可能是個挑戰，又如果所在的區域網路頻寬有限，或是在某些公用電腦上，使用者可能不具安裝軟體的權限，則軟體的安裝會是個大問題。因此，這個網路掃描代理人必須能夠以使用者的權限自動安裝於使用者電腦上。
3. 具備跨作業平台的特性。使用者會經由各種可能的作業平台來連接所提供的網路服務，因此這個網路掃描代理人必須能夠在該使用者所使用的作業平台上，安裝並進行掃描工作，如此方能獲得從該電腦進行網路連線的真實情況。
4. 網路管理者可於伺服器端讀取掃描結果。網路掃描結果的判讀，對一般使用者來，可能比起安裝軟體更為困難。因此，這個網路掃描代理人，應該代替網路管理者去執行網路掃描工作，並將其掃描結果送至伺服器端，讓管理者可以得知掃描結果。此外，管理者也可自伺服器端直接讀取伺服器端的結果，並加以判讀。

為達成上述目標，我們設計了一個網路掃描代理人系統。這個網路掃描代理人系統包含了通訊埠掃描伺服器端(Port Scan Server)和通訊埠掃描代理人(Port Scan Agent)這兩部分，透過兩者的互動，即可完成網路掃描的工作。為讓網路掃描代理人得以順利派送至使用者端，我們假設客戶端可以順利透過瀏覽器連線至伺服端的 Web 網站，並取得網路掃描代理人。這個假設是符合目前多數網路安全設定的現況，因為 Web 瀏覽可說是當前網際網路必備的應用，因此我們假設 HTTP (TCP/80)是暢通的。如果使用者連 Web 網站都無法瀏覽，我們認為他應該求助當地的網路管理者，而不是遠端其他網路服務的管理者。以下分別說明掃描伺服器端及掃描代理人的設計原理與架構。

2.1 網路掃描伺服器端之設計

網路掃描伺服器端的任務是指派代理人到使用者的電腦，提供通訊埠之連線伺服服務以供代理人進行掃描，並將網路掃描結果加以記錄，以供管理者查閱。以下分別說明其主要工作內容：

1. 在指派代理人部分，我們可以將事先編譯完成的代理人程式碼置放於伺服器端，當使用者端的電腦提出掃描請求時，即可將代理人程式碼送至使用者端，並提供代理人執行工作時所需的資訊，即可完成代理人指派的工作。
2. 在提供通訊埠連線伺服服務方面，其主要的任務就是提供掃描代理人在進行網路通訊埠掃描時，有對應的通訊埠服務程序來回應掃描代理人。最簡單的作法，就是把所有掃描範圍的每個通訊埠都建立一個通訊埠服務程序來負責聆聽並回應該通訊埠的掃描。當掃描範圍不大時，這個方法是可行的，然而，當掃描範圍相當大時(例如，1,000 個以上的通訊埠)，這個作法將會耗費相當的系統資源，伺服器的作業平台也不見得適應這樣的做法，且同時開啟大量的通訊埠也容易產生

系統安全性的顧慮，因此我們採用移動視窗(sliding window)的架構，動態建立通訊埠回應程序來服務掃描代理人。

假設這個移動視窗可記錄子程序資訊的個數為 P_w，也就是伺服器端同時最多會有 P_w 個子程序負責其所對應的通訊埠，當一個子程序完成其任務後，系統會將該子程序自移動視窗移除，並產生掃描範圍內負責下一個通訊埠的子程序，並置入移動視窗之中。P_w 的大小和掃描伺服端與掃描代理人之間的同步誤差有關，我們只要確保掃描伺服子程序在掃描代理人進行掃描時，已經領先到位準備服務代理人，即可順利完成全部通訊埠的掃描工作。

當子程序順利收到掃描代理人的連線並確認後，該子程序便完成其任務，此時，系統紀錄該通訊埠為暢通的，並將其自視窗中移除，再配置新的子程序到移動視窗之中。當先前通訊埠均為暢通的情況下，這個順利完成任務的子程序，通常會是視窗中最頭端的子程序，也就是目前視窗中最早進入視窗的子程序，在這種情況之下，視窗內仍有 P_w 個子程序領先代理人的掃描，掃描將可以順利繼續推展。

當子程序未接收到掃描代理人的連線時，該子程序會一直等候在視窗中，此時我們必須利用逾時(timeout)的機制來管理移動視窗，以確保視窗內仍有足夠的子程序領先代理人的掃描。假設代理人至多 T_{scan} 時間就完成一個通訊埠的掃描，則當通訊埠被阻隔，且經過 T_{out} 時間後，移動視窗內至多會有 $\lfloor T_{out}/T_{scan} \rfloor$ 個子程序殘留，為避免掃描代理人領先掃描伺服器端，造成沒有對應通訊埠可以掃描的錯誤情況，我們必須確保視窗內仍有 1 個以上可用的子程序，即

$$1 \le P_w - \frac{T_{out}}{T_{scan}}, \qquad (1)$$

因此，當子程序殘留在視窗的時間超過 T_{out} 時，其中

$$T_{out} \le (P_w - 1) \times T_{scan}, \qquad (2)$$

必須將該子程序視為逾時(timeout)，並將其所負責的通訊埠視為被阻隔，結束該子程序的任務，以確保移動視窗的正常運作。

除此之外，我們也可以根據最新回應成功的子程序，來加速移動視窗的管理。即假定移動視窗內位置 P_i $(0 \le i < w)$ 的子程序已經順利回應代理人的掃描，但視窗頭端(即 P_0 位置)的子程序尚未收到連線，假設連線請求到達伺服端的延遲時差 (delay jitter) 至多不超過 T_{gap} 時間，則當 $P_i > P_{gap}$ 時，其中

$$P_{gap} = \frac{T_{gap}}{T_{scan}}, \qquad (3)$$

即可將 P_0 所對應的通訊埠視為被阻隔，如此亦可提高視窗內可用子程序的數量。

3. 在記錄網路掃描結果方面，可將通訊埠子程序本身的伺服情況，以及遠端代理人連線掃描結果，整合掃描代理人端的網路資訊(如：IP 位址)和掃描進行的時間，記載到資料庫中，可提供管理者進行即時和歷史資料的查詢。

依照上述工作內容及考慮因素，我們設計了一個網路掃描代理人系統的伺服端。當伺服端程式取得通訊埠的掃描服務範圍並完成初始設定後，即等候掃描代理人的啟動通知。當掃描代理人通知啟動後，伺服器端會產生 P_w 個子程序置入移動視窗中，每個子程各自依序負責一個掃描範圍內通訊埠的掃描伺服服務，並依照連線成功，或是連線逾時，回報母程序，完成其任務。每完成一個通訊埠，母程序就啟動下一個通訊埠的掃描伺服服務，直到全部通訊埠掃描完畢，其運作流程如圖 3 所示。

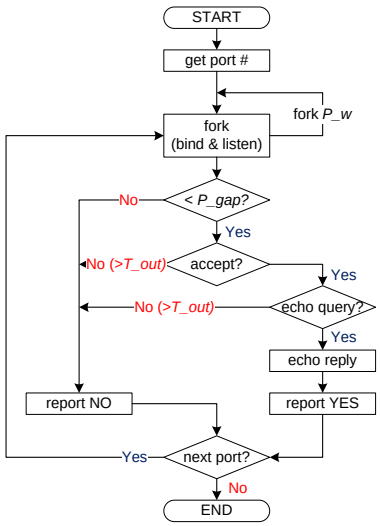

圖 3 網路掃描伺服器的運作流程

2.2 網路掃描代理人之設計

網路掃描代理人被指派到使用者電腦後，即依照掃描伺服端指定的工作內容，開始執行網路通訊埠掃描的工作，並將掃描結果呈現於使用者端。以下分別說明網路掃描代理人的主要工作內容：

1. 網路掃描代理人被傳送到使用者端後，可自我啟動，並讀取伺服器伴隨代理人傳送過來的其他資訊，來決定要掃描的主機及通訊埠範圍。
2. 網路掃描代理人將依照欲掃描通訊埠的範圍，自最小號碼的通訊埠，逐一進行掃描。當與伺服端成功建立連線後，代理人會送出詢問口令，並等候伺服端回應，如伺服端正確回應口令，即表示該通訊埠是暢通的。若連線遭拒絕，口令回應不正確，或是連線經過 T_{scan} 時間後，沒有得到回應，均視為通訊埠被阻隔。代理人將測試結果加以記錄後，即開始進行掃描範圍內下一個通訊埠的測試，直到全部的通訊埠測試完畢。

3. 待全部通訊埠掃描完畢後，代理人會將全部掃描結果告知使用者，並通知伺服端，之後網路管理者可自伺服器端得知掃描結果。此外，如果允許，在不影響代理人運作效能的情況下，代理人應將執行進度即時告知使用者，以利使用者了解掃描進行的概況。

依照上述工作內容及考慮因素，我們設計了一個網路掃描代理人，可依序進行網路通訊埠的掃描工作，其運作流程如圖 4 所示。

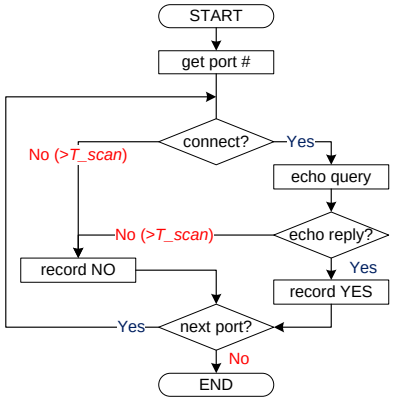

圖 4 網路掃描代理人的運作流程

2.3 網路通訊埠掃描流程

整合網路掃描伺服端及網路掃描代理人，即構成一個網路掃描代理人系統。當網路管理者接受到使用者的詢問時，如果懷疑是中繼網路(或使用者本身電腦上的防火牆)阻斷了從使用者端到伺服器端的通訊埠，管理者可以引導使用者，請其連線至網路掃描代理人系統的 Web 網站，填入適當資訊後，即可開始通訊埠的掃描工作。待網路掃描代理人執行完畢後，網路管理者即可得知從使用者端到伺服端的網路掃描結果，其運作流程如圖 5 所示。

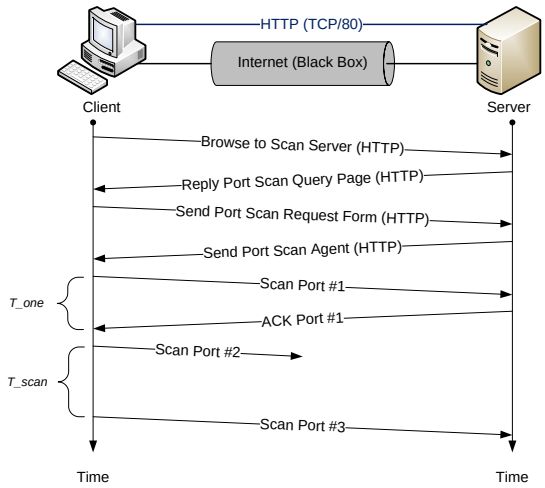

圖 5 網路通訊埠掃描流程

3. 網路掃描代理人之實作

為驗證網路掃描代理人之可行性，我們實作了一個網路掃描代理人系統。我們採用 Linux Fedora Core 6 作為我們的作業平台，並以 Apache v2.2.3 + PHP v5.16 [5]作為 Web 發展平台。

當使用者連上網路掃描代理人系統的 Web 網站時，我們透過 PHP 顯示使用者端的 IP 位址，讓使用者可根據管理者的指示填入掃描範圍(如圖 6)。使用者送出掃描範圍後，伺服端利用 PHP 產生動態網頁，將掃描代理人的派送單及掃描代理人所需的參數資料傳回使用者端(如圖 7)。使用者的瀏覽器在收到代理人派送資訊後，會自伺服端取回並啟動代理人 Java Applet (PortScan.class)。啟動後的代理人，會自派送單中，讀取掃描運作所需的參數，並開始網路通訊埠的掃描。

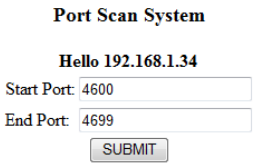

圖 6 網路掃描登入畫面

```
<html>
<title>Port Scanner</title>
<body>
<applet code="PortScan.class" width="300" height="300">
        <param name="port0" value="4600">
        <param name="port1" value="4699">
        <param name="code" value="20070731-040300">
</applet>
</body>
</html>
```

圖 7 網路代理人派送網頁

在網路掃描伺服端，我們使用 C 語言及 Unix socket [6]來開發伺服端程式。伺服程式會被 PHP 啟動，並讀取 PHP 所傳送的參數來決定提供服務的對象(代理人端的 IP 位址)及通訊埠伺服的範圍。程式利用陣列實作移動視窗來存放 P_w 子程序資訊(包含子程序編號，負責的通訊埠號碼，及啟用時間)。程式一開始會先產生 P_w 個子程序，每個子程序各負責一個通訊埠的掃描伺服。子程序和母程序之間，透過 Unix socket 傳遞資訊，如此每個子程序的運作結果都可以傳送給母程序來加以記錄。母程序可以根據移動視窗內的資訊來管理子程序，當子程序負責聆聽的通訊埠號和已經完成的通訊埠號碼相差超過 P_{gap}，或子程序超過 T_{out} 時間沒有回報母程序，母程序會結束該子程序，並啟動掃描範圍內的下一個子程序。經過多次的實驗，依據方程式(1)，(2)和(3)，我們發現，令 P_w=5，及 P_{gap}=3，T_{out}=8 sec (在 T_{scan}=2 sec 的情況下)即可滿足一般連線狀況的掃描測試，也就是說，在伺服器端我們只需要極少的系統資源，即可滿足大範圍的網路掃描工作。

在網路掃描代理人部分，我們使用 JDK 6 Update 2 產生相容於 JDK v1.4 的代理人(Java Applet ByteCode) [2]作為掃描代理人，如此即可滿足目前大多作業平台上的 Java VM。利用 Java VM 在各種作業平台的普及性，這個代理人具有高度的跨平台特性。當這個代理人啟動時，會先讀取伴隨伺服端網頁傳送過來的資訊(圖 7 的 param 參數)，以決定通訊埠的掃描範圍。代理人完成初始設定後，會等待使用者按下啟動按鈕(如圖 8 的 GO 按鈕)。當使用者按下啟動鈕後，代理人會透過 Web 連線通知伺服端，如此伺服端會開始通訊埠掃描的伺服服務，之後便開始通訊埠掃描的工作。

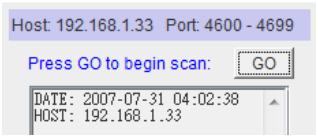

圖 8 網路掃描代理人起始畫面

網路掃描代理人會設定每個通訊埠的連線逾時為 T_{scan} 時間，並依照通訊埠的順序，逐一進行通訊埠的掃描測試。若連線成功，且完成口令詢答，則記錄通訊埠為暢通。若連線遭拒，或連線逾時，或應答不正確，則記錄通訊埠為遭阻隔。在過程中，即時的掃描資訊會顯示在畫面上(如圖 9)，若使用者因故欲中斷代理人的網路掃描工作，可隨時按下停止按鈕(如圖 9 的 STOP 按鈕)，代理人會通知伺服端停止伺服子程序，並停止掃描。

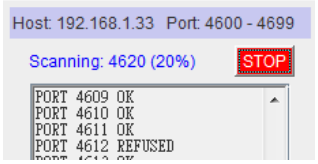

圖 9 網路掃描代理人反應掃描進度

全部通訊埠掃描完畢後，網路掃描代理人會顯示掃描完畢的訊息，並將全部掃描結果顯示在畫面上(如圖 10)，供使用者參考。

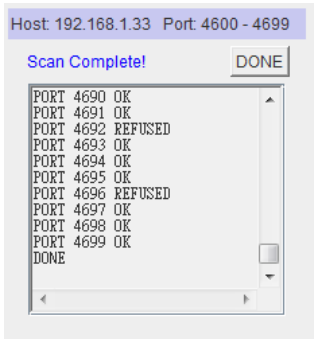

圖 10 網路代理人完成掃描

4. 問題與討論

在設計及實作這個網路掃描代理人的過程中，我們發現幾個值得探討的問題：

1. Java Applet 版本的問題。在設計網路掃描代理人的初期，我們希望能運用 Java 的優勢，讓我們開發完成的 Java Applet 能夠在不同的作業平台上順利執行。然而，當我們完成第一個版本的時候，就遭遇到在 Windows 平台上，因瀏覽器的 Java plug-in 版本較舊，而無法順利執行的情況。為讓完成的網路代理人能夠順利在更多的 Java VM 上執行，我們選擇產生 1.4 版 Java VM 的 ByteCode 來做為我們的網路代理人。另外，我們也希望網路代理人能夠支援行動裝置，以探索行動上網的世界，但是行動裝置上的 Java VM (如：J2ME [3])和在一般 PC 上的 Java VM (J2SE) 並不相同，我們需要另行開發程式碼，方能達成這個目標。

   在開發過程中，為顯示即時的動態，我們也發現 Java AWT 和 Swing 在即時呈現圖形的處理上，是有相當差異的。此外，我們運用 Java Thread 來分別處理網路掃描和圖形處理部分，以減少時間上的誤差，並提高對代理人的控制性。

2. 網路掃描伺服器端與網路掃描代理人同步的問題。為達成利用少量系統資源，即可完成大範圍掃描的任務，我們採用了移動視窗的架構來伺服網路掃描代理人。這樣的架構，在通訊埠暢通時，伺服端與代理人可以透過程成功的連線回應，來同步兩端的進度，讓伺服端可以保持微幅領先代理人端。然而當代理人端與伺服端的掃描連續遭遇阻隔時，兩端的進度就僅剩下各自的逾時架構來處理。這種狀況如果時間一旦拉長，就可能因為累積性的時間誤差，而產生兩端無法互相匹配的問題，因而無法成功完成掃描。特別是，當伺服端沒有收到連線，但代理人端因為收到中繼網路設備拒絕連線的回應而順利向前推展時，問題會更為嚴重。

   為降低伺服端與代理人發生同步問題的可能性，我們可以選用使用較大的 P_w，或減少 T_{scan} 與 T_{out} 的差值，來增加系統對長時間同步誤差的適應性。但是較大的 P_w 會佔用較多的系統資源，而較高的 T_{scan} 會造成掃描一旦遭阻隔時，掃描進度的推展會變得非常緩慢，使用者可能不耐久候。因此如何選擇適當的參數，讓系統能同時提高效率與可靠性，甚至讓程式能自動配合執行情況自我調整參數，讓系統能自動適應更多的網路環境，是值得我們繼續努力的目標。

3. 網路掃描伺服端如何同時服務一個以上網路代理人的問題。當完成以伺服器服務一個網路掃描代理人的工作後，我們就著手思考，如何讓這個伺服器端可以同時服務一個以上的網路掃描代理人，如此更能提高他的實用性。考慮一個通訊

埠僅能被一個子程序所伺服的情況下，我們無法以動態程序管理機制來服務掃描區域重疊的網路代理人。因此，對於少數的系統常用通訊埠(well-known ports)，或是常見的網路應用服務的通訊埠，我們可以利用靜態的伺服程序來因應。至於其他範圍的掃描，可以利用註冊網路代理人掃描範圍的方式，檢查目前進行中的掃描是否有重疊的部分，來同時服務更多的網路掃描需求。

4. 網路伺服器所遭遇的網路安全問題。在我們進行網路掃描實際測試的過程中，我們並不意外的發現，某些特別的通訊埠(例如：網路病毒攻擊的通訊埠)，當通訊埠的伺服程序啟動後，立刻就有外來的未知程序不斷地連線到該通訊埠。該通訊埠的伺服子程序，會疲於應付這些如排山倒海而來的連線，也因此造成該伺服子程序無法順利服務網路掃描代理人的連線。因此，我們利用 Linux 系統內建的防火牆(firewall)來隔絕這些非請自來的連線，僅對網路代理人所申請的掃描範圍，開放代理人端連線至伺服器端，如此即可提升對伺服端系統的保護，同時亦可順利達成服務網路掃描代理人的目標。
5. 網路掃描代理人可做為網路安全檢測工具。受限於 Java Applet 的安全特性，我們無法利用網路掃描代理人，來取得從使用者端到其他伺服器的連線概況，但是，這樣的網路代理人已經可以讓網路管理者在不需移駕到他方的情況下，得以窺見從使用者端到掃描伺服器端的網路黑盒子的概況。因此，對系統管理者來說，欲了解其所管理的網路及所使用的網路安全設備是否達成其防護目的，可請託遠端的友人，利用這個代理人系統代為檢測，或者是當管理者移動到他處時，亦可隨時機動進行檢測。此外，對於較熟稔網際網路的使用者來說，這個系統亦可做為他移動到不熟悉的網路環境時，主動進行通訊埠暢通檢測的工具。當使用者可自己連線到代理人系統網站，輸入欲探測的通訊埠號碼，根據代理人所呈現的掃描結果，即可對所處位置的網路安全管理政策略知一二。

5. 結論

網路通訊埠掃描是網路管理實務上的一項重要工具，利用網路掃描，可釐清使用者無法使用某些網路應用，是否係因為通訊埠被中繼網路阻隔所造成。在本文中，我們提出了一個以 Java Applet 為基礎的網路掃描代理人，利用這個代理人，使用者不需另外安裝網路掃描的軟體，網路管理者也不需移駕到使用者所在的地方，即可順利完成從使用者端到網路管理者端的網路通訊埠的掃描工作。

這個以 Java Applet 為基礎的網路掃描代理人，配合遠端的掃描伺服端，運用動態的伺服程序管理，僅使用少量的系統資源，即可達成讓網路管理者了解自遠處使用者端進行網路通訊埠掃描到伺服端的結果。我們實作了這個網路掃描代理人系統，經實驗證明，這個網路掃描代理人系統可以在各種目前常用的電腦作業平台上執行，並順利完成網路掃描的工作，是一個網路管理者協助使用者排除網路障礙，或是使用者移動到不同網路環境時，自我檢測網路通訊概況的實用工具。此外，這個網路掃描代理人，也可以做為網路管理者，機動檢測所管理網路的網路安全防護機制是否符合預期的行動式網路安全測試工具。

參考文獻

[1] Applets, http://java.sun.com/applets/.

[2] Elliotte Rusty Harold, Java Network Programming, 3rd Ed., O'Reilly, 2004.

[3] Java ME Platform, http://java.sun.com/javame/.

[4] Tim Lindholm and Frank Yellin, The Java Virtual Machine Specification, 2nd Ed., 1999.

[5] PHP Manual, http://www.php.net/manual/en/.

[6] W. Richard Steven, Bill Fenner, and Andrew M. Rudoff, Unix Network Programming, Addison Wesley, 2004.