P2P 協定行為與解析方法

彭偉豪 林義能 林盈達

摘要

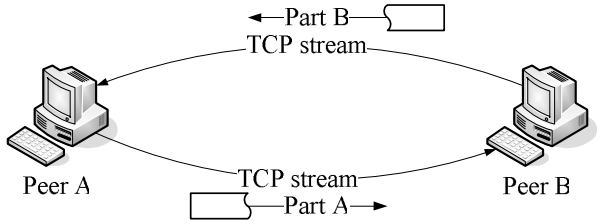
P2P 軟體打破傳統的 client/server 觀念，傳輸的兩方站在相等的立場分享彼此需要的資料。這樣的技術被廣泛運用在網路上各種領域的同時，也出現許多問題，例如惡意入侵、病毒攻擊與版權的侵權議題；管理 P2P 網路的機制亦因此受到重視與期待。而管理的前提便是需要知道該 P2P 協定的行為模式。

本文提出一個解析未知 P2P 協定的方法。我們使用 Ethereal 軟體作為擷取封包的工具，藉由取得的資訊來解析協定中有哪些 commands、options 與特殊功能，建立一個完整且乾淨的 P2P 網路環境、設計良好的測試用傳輸檔案、詳盡的分解連線步驟，幫助我們將協定的資訊藉由乾淨的封包內容明白的顯現出來。

1. P2P 的類別與應用

Peer-to-Peer 軟體之核心概念

自 1999 年 Napster 在網路上推出 MP3 音樂的分享機制後，P2P 技術便被廣泛應用在各種領域中，例如：Bittorrent 技術的檔案分享、分散式的計算平台等。不同於以往 client/server 傳輸的架構，在 P2P 的網路環境下，每台參與的電腦都被假設具備有同等的傳輸能力來分享資料，如圖一所示。節點 A 需要節點 B 的 B 部分檔案，相對的節點 B 需要 A 的 A 部分檔案，兩方互取所需，同時傳輸對方需要的部份。一般而言，我們會將 P2P 的應用分成如下分類：



圖一：P2P 網路概觀圖。

(1)資料分享：

檔案分享是非常普及的 P2P 應用模式，我們以是否存在伺服器對其分類。一種狀況是存在集中式的伺服器作為檔案來源的分派，這類型的代表軟體大都是只有在檔案傳輸時才用到 P2P 的技術，有最早期的 Napster、後來的 Edonkey/Emule 伺服器、BitTorrent 所採用的 tracker 技術等。另外一種作法是在 P2P 網路中，不存在任何主機擔任提供檔案來源分派的角色，每一個 peer 都可擔任類似伺服器的角色提供檔案來源功能；具代表性的有 Gnutella、Grouper、MSNFTP 與 BitTorrent(其中的 DHT 技術)等軟體。

(2)即時通訊：

一般來說，大多數的 IM 在 peer 與 peer 傳輸訊息的過程中，都會經過第三方的

伺服器才能送到另外一端的 peer，其中的某些功能使用了 P2P 的技術。例如 MSN 中兩個 client 在傳送檔案時使用的 MSNFTP 協定、Skype 連線後兩個 peers 間的通話，都是 P2P 技術的一種。

(3)分散式數據計算：

分散式數據計算的概念就如同一群人準備一起做一件事情，但是他們彼此手上也許還都有自己分內的事情要做，這些人會各自利用自己空閒下來的時間或是資源去完成這份工作的某些部分直到這一份工作完成。這類型的軟體有：Entropia、GPU(Global processing unit)、Intel philanthropic P2P program 與 KaZaA 的 P2P computing 計畫等。各類型的軟體大致可以整理如表一：

資料分享	即時通訊	分散式數據計算
Bittorrent family Edonkey/Emule Gnutella KaZaA Napster Clubbox MSNFTP	MSN Messenger Yahoo Messenger AOL Messenger ICQ Skype Jabber GoogleTalk	GPU(Global processing unit) Intel philanthropic P2P program Entropia

表一：目前各類受歡迎的 P2P 軟體。

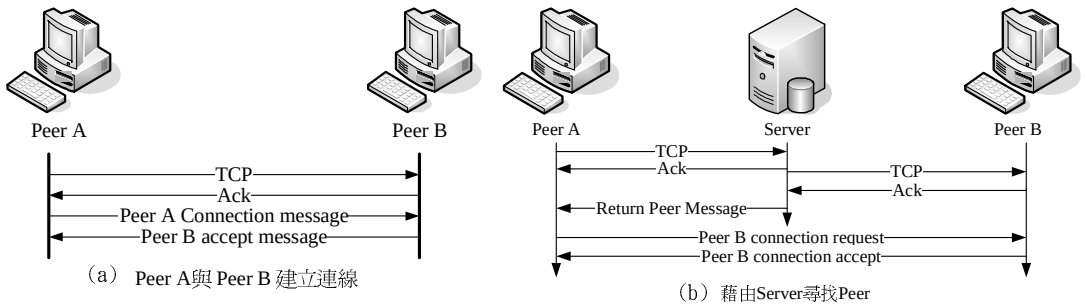
2. 各類別的“基本模式”為何？“特殊模式”為何？

針對各類行不同的 P2P 的通訊協定，我們可以將其歸納成兩種分類來討論，一種是各種 P2P 協定皆需要具備的”基本模式”，即協定的在建立連線、傳輸資料、結束連線時所以需要的功能；另外一種針對協定本身具備的特殊功能做討論，稱之為該協定所具備的”特殊模式”。例如 DHT 技術、分散式計算與資源分配等。

P2P 通訊協定應具備的基本功能

1. 尋找適當的 peer 以建立連線

我們用需不需要仲介的伺服器角色來做基本分類。圖二(a)表示兩者透過某種機制下已知對方的位址，在確認回應之後便可建立連線；圖二(b)則是 peer 要找到另外一個 peer 時，需要經過第三方仲裁來得知另外一個 peer 的位址，這種作法頻繁的使用在檔案傳輸與即時通訊類型的 P2P 軟體。



圖二：P2P 搜尋 peer 的方式：(a)透過機制直接尋找，(b)透過第三方。

就連線的部份，一般而言資料分享類型的 P2P 軟體不需要額外的設計，所以上述的兩種基本建立連線方式便涵蓋了各種類型的 P2P 軟體，例如：Grouper 採用同一群組內的聯絡人可以互相傳遞彼此需要的檔案，Emule 與 Bittorent 則是藉由伺服器去尋找可連線的 peer。

2. 找到後如何傳輸資料？

傳輸資料建築在 TCP/IP 的架構上。一般而言在建立連線之後，兩 peer 間得到了足夠的訊息後便開始著手傳輸資料的工作，在解析這個步驟通訊協定做了哪些事情可以針對資料傳輸的方式(*循序式*或者*區段式*)、及資料是否加密等。

3. peer 與 peer 間如何得知傳輸完成？

在兩個 peer 間傳輸完成時，協定會設計下達一個由接收方送到傳送方的命令，表示目前需要的資料區段已經完成接收了，兩方便結束目前的連線。

各類 P2P 通訊協定的特殊功能

(1)資料分享的 P2P 軟體所需的額外特殊功能

對傳輸資料時所需要的特殊功能而言，我們要思考 peer 間傳輸檔案時，是採用區段式(segmentation)或是循序式(sequential)的傳輸，或者是在區段式的傳輸時，檔案的區段如何被分佈在各 peer 之間。一般而言，不同版本的 P2P 軟體在功能上也會有許多差異，而這些差異通常便需要透過解析其協定訊息才能從中得到線索。

(2)即時通訊與數據計算的 P2P 軟體所需的額外特殊功能

在即時通訊的通訊協定中最需要強調的功能就是保障訊息可以低失誤與連線的穩定性，Skype 使用 supernode 便是一個例子。對於數據計算的 P2P 軟體而言，最重要的則是分享彼此閒置的資源，因此如何分配工作給一個可以空出資源的 peer，便是這類型 P2P 協定需要去決定的事情。此類型的 P2P 軟體由於協定大多經過加密保護其獨有的技術，要去解析這一類型的 P2P 協定並不容易。表二列出常用 P2P 軟體之特殊功能：

應用程式	Edonkey/Emule	Bittorrent	MSN	MSNFTP	Skype
傳輸方式	區段式	區段式	循序式	區段式	循序式
協定訊息加密	未加密	未加密	未加密	未加密	加密
資料傳輸	未加密	未加密	未加密	未加密	加密
動態通訊埠	使用	使用	固定式	使用	使用
檔案名稱	不可見	可見	可見	不可見	X
內定通訊埠	4661~4665	6881~6889	1863	X	443/80

表二：常用 P2P 軟體之特殊功能列表。

3. 如何自行解析一個 P2P 應用之協定

為什麼需要解析未知的 P2P 協定

解析未知 P2P 軟體使用的通訊協定的好處可以從防護與應用這兩個方面來探

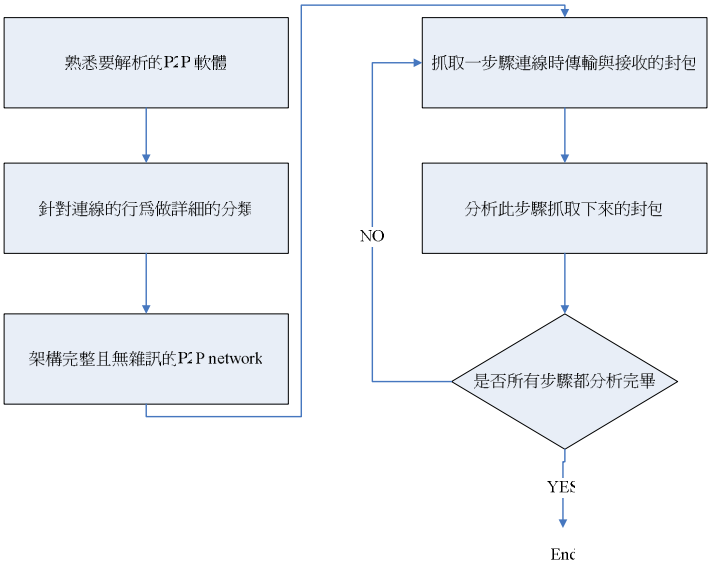
討。P2P 軟體存在頻寬問題與潛在威脅如病毒散布、惡意的網路攻擊，找出解析的方法可以幫助我們在面對這些問題時，更容易找出解決的相應之道。

由於 P2P 協定會發佈的資訊是架構在應用層之上，所以我們利用擷取網路中封包 data 欄位所得的資料，進而分析其協定可能的行為模式。這樣的方法是建立在完整且低雜訊(亦即減少不必要的封包)的 p2p 網路測試環境，藉由監控此網路環境中封包所傳遞的資料來分析協定中可能擁有哪些命令、選項與協定所具備的特殊行為。

測試工具與測試方法

本文使用 Ethereal 來監聽網路中傳輸的封包(見圖三)，它可以詳細的提供抓取下來的封包資訊。我們在建構好的測試環境中，針對連線的特性設定要抓取封包的過濾器，用來過濾出我們所需要的封包。例如：MSN messenger 使用固定的 port 1863，在 Ethereal 中設定過濾器語法”Tcp.port == 1863”，便可以擷取 MSN messenger 透過這個 port 所傳輸的封包。

我們將焦點放在封包中的 Data 欄位部份，分析其中有哪些命令及選項。接下來需要更進一步的思考如何切割分析每一個步驟。因為我們必須讓擷取下來的封包不會太過蘊亂，所以最好的作法便是針對此 P2P 軟體的每一個步驟做切割並且一步一步地擷取每一個步驟所傳輸的封包，能將行為分類的越詳細，就越能確保抓下來的封包資訊是有用的。分類過於粗糙將有可能會因為所接收到的指令數過多，無法輕易判別該指令的用途為何。例如單就端點間傳輸檔案時，可以將其行為拆解建立連線、計算檔案資訊、傳輸檔案、結束連線等步驟等。設計此解析 P2P 協定方法之主要精神建構在夠詳細的連線動作分類與消除可能的雜訊，在這些動作中當大部分的因素都在我們自己的掌控下(網路環境、軟體的連線動作、傳輸的資料等)，整個協定的行為模式自然就會浮現出來。圖四是我們解析一個未知的 P2P 協定時所使用的基本流程圖：



圖四：解析 P2P 協定之流程圖。

每一個步驟都有一些相對的問題需要考慮。在第一步熟悉 P2P 軟體時，我們必須考慮到同一種類的 P2P 軟體，可能有許多的衍生版本，那麼在分析協定時，便必須考慮要採用哪一種版本的軟體來做我們的測試主角(例如 Bittorrent 就有許多的衍生版本，這些版本的協定都是建構在 Bittorrent 上，但是不盡相同)，表二所要表達的，就是圖五在解析協定的每一步驟時，我們需要去考慮哪些問題，而這些問題便是用來幫助我們更加釐清協定之用。

測試步驟	需要考慮的問題
熟悉要測試的 P2P 軟體	選擇哪一個版本的軟體當作測試目標
	了解最少需多少台機器參與其中
針對連線行為做詳細的分類	從基本的三個步驟中，做出各細部的分類
架構完整且無雜訊的 P2P 網路	是否有伺服器的角色存在
	是否需要對外連線
截取某步驟傳輸時傳送與接收的封包	傳輸是否使用固定 port
	採用什麼方式傳輸資料
	連線在什麼條件下被終止
分析此步驟抓取下來的封包	協定如何分配 data 欄位
	資料是否有經過加密
	資料是否有被切割與其切割的大小
	連線動作可能用到的命令為何?

表三：解析協定的步驟與注意事項。

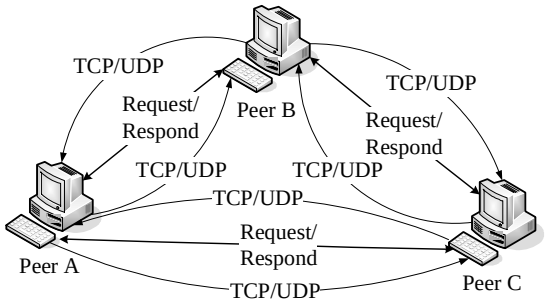
如何設計測試環境

(1)架構低雜訊影響的 P2P 網路

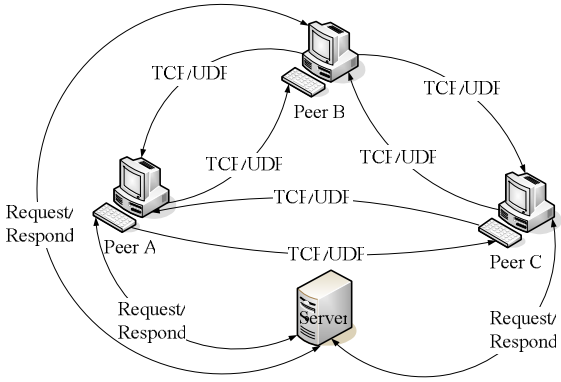
P2P 網路環境中，需要確保擷取封包時不會產生太多雜訊擾亂分析，這些雜訊的存在與否在於我們架構的網路環境是否存在其他不必要的連線，例如：Windows 下的 Live Update、MSN 在偵測到連線後自動發出登入的 request 與網路上其他的電腦彼此間傳遞封包；或者針對我們所要測試的 P2P 軟體而言同時有太多不同動作的連線產生，如：Bittorrent 在 tracker 上要求某一個檔案的 seed 時，同時已經連線到其他 seed 下載另外的檔案；或者當 Emule 正在從伺服器上查詢檔案資訊時，又被其他的 peer 發出連線的請求。這些因素都會造成我們在監看封包時，取得太多讓人混淆的資訊。所以就必須設計好我們在偵測此動作時，保證同一時間內只會擷取到動作連線產生的封包，如此便可以較容易的從我們擷取下來的封包看出在這個動作發生時，做了哪些事情、下了哪些命令。

架構一個無雜訊的 P2P 網路，理想的作法就是架構一個沒有多餘外界連線的私人區域網路以避免可能來自外界不需要的封包。而架構在這個區域網路下需要哪些成員，其考慮的因素包括：是否存在伺服器？最少需要多少成員參與傳輸？是

否有對外連線的需求(如 Skype 使用 supernode 時便需要)？ 傳輸資料的格式為何？圖六表示兩種不同類型的 P2P 網路架構圖：(1)表示 peer 間找到彼此並且可以傳輸資料前，需要透過 Server 做仲介的機制(2)表示 peer 間經過某種機制設計，可以直接連線到彼此並且傳輸資料。



(1)P2P network 架構圖 - 當環境中沒有Server時



(2)P2P network 架構圖 - 當環境中存在Server時

圖六：(1) peer 間彼此直接傳輸；(2) 透過伺服器擷取資訊的網路架構。

(2)設計傳輸用的低雜訊資料

當我們在分析一個 P2P 協定時，除了考慮到在網路環境中減少雜訊的產生外，還有另一個要注意的地方—傳輸資料的設計。若傳輸資料選擇不當，則我們從封包接收下來的訊息就會產生以下幾點混淆：

1. 無法確切得知傳輸資料的傳輸是否加密；
2. 傳輸資料的內容可能與命令、選項的訊息產生重複；
3. 無法得知傳輸資料傳輸是否為循序式或是區段式；
4. 較難猜測一個封包中傳輸資料的區段大小。

理想的傳輸資料設計應該是在數次傳輸步驟的封包擷取之後才會產生，原因是我們必須從前幾次的封包擷取中對於協定中存在的命令與選項做一些合理的猜測，才可以從自行設計的傳輸資料中避免掉這些命令關鍵字；另外對於協定制定傳輸資料的區段大小，我們也要用不同大小的傳輸資料去測試。好的傳輸資料應該要具備下列的條件：

1. 可以明確判讀的連續內容；
2. 避免命令訊息出現在傳輸內容中。

針對第一個條件來說，一般而言除非該 P2P 協定針對不同的檔案型態有不同的處理方式，不然我們都可以利用 ASCII code 的內容來組成一個檔案，一方面封包擷取下來時容易判定其中的內容是不是自己設計的檔案，一方面也較容易去修改其中的內容以符合我們的需要。第二點則是要避免命令的字串與檔案內容在封包中的 data 欄位中被放在一起而造成混淆，所以我們在設計檔案內容時有可能會需要不斷地修改設計直到不會產生混淆的內容。

(3)逐步解析出協定的命令與選項

P2P 協定的命令與傳輸的資料都是放在 TCP 封包中的 data 欄位下，由於這些欄

位的配置都是固定的，而檔案的內容又是我們預先設計好的形式，所以要猜測出哪一段字串是命令並不太難。

實作範例-以 MSNFTP 為例，實際分析其協定（以成功傳輸的行為為例）

針對 MSNFTP 檔案傳輸之測試環境建構

MSNFTP 測試所需的成員有：傳輸方、接收方與一台負責仲介資訊的 MSN 伺服器。其中 MSN 伺服器是由 Microsoft 提供，也就是傳送、接收兩方都是需要在 MSN 伺服器先登入之後才能傳輸檔案。所以在建立測試環境時，我們需要準備好兩個測試用的帳號與兩台測試用的電腦，在做測試時要避免額外的連線影響我們擷取資訊。

設計一個提供測試分析用的訊息結構

一開始可以用簡單的 a~z 重複 100 次的純文字內容檔案，內容大約佔 4KB 左右，接著由帳號 A 傳遞給帳號 B，從擷取下來的封包可以看到其中的命令字串是一些有意義的簡單縮寫，後面接著一串 ID；這代表內容設計上，應該要避開數字以避免跟 ID 有所重複，如此反覆修改內容便可以設計成一個完整的測試檔案。

針對其封包所需分析的資訊

MSNFTP 的成功連線動作可以細分如下：

(1)兩個 peers 間確定對方的存在(MSN 伺服器提供的訊息)；

(2)兩個 peers 間確定彼此使用的 MSNFTP 協定版本一致；

(3)若是答應接收的話，則接收端發送確認訊息；

(4)確認後發送方開始傳輸資料；

(5)傳送完畢後結束 MSNFTP 連線。

從封包的擷取中，我們可以解析出在 MSNFTP 中，每一個步驟中產生的命令，如同表四所列：

命令	作用	參數
INVITE	兩方搭起連線的命令	發送方送出，接收方確認
VER	確認彼此的 MSNFTP 協定版本	發送方送出，接收方確認
USR	接收端確認後傳送的命令	自己的 name 與確認 key 值
FIL	發送端收到確認後傳回檔案的賸餘	檔案的 byte 數
XFR	接收方請發送方開始傳輸	無
BYE	結束傳輸	ID:16777987 表示傳送成功

表四：MSNFTP 協定的命令與選項。

結論

本文提供一個破解未知 P2P 協定時所需要的技巧，藉由文章中介紹的流程：建構低雜訊的網路環境、設計良好的傳輸資料、詳細的分解每一步連線，在解析協定時將會花費更少的時間，得到的結果也能更精準。而因為各家協定設計有相當程度的差異，要找出一個能百分之百解析各種協定的通則是不容易的；但是藉由上述的流程來分析所抓取下來的資訊，將有助於我們在分析 P2P 通訊協定時結果的準確性。

參考資料

[1] 蔡孟甫、林盈達, “A novel gateway architecture for managing dynamicport peer-to-peer traffic,” thesis dissertation, National Chiao Tung University, 2005

[2] The eMule protocol specification, http://www .cs.huji.ac.il/labs/danss/presentations/emule.pdf.

[3] Ethereal documentation, http://www.ethereal.com/docs/.

[4] MSN messenger protocol forum , http://msnpiki.msnfanatic.com/index.php/Main_Page.

[5] MSN messenger protocol documentation, http://zoronax.bot2k3.net/.

[6] Bittorrent official protocol, http://www.bittorrent.com/protocol.html.

[8] P2P networks documentation, http://ntrg.cs.tcd.ie/undergrad/4ba2.02-03/p9.html.

[9] Edonkey, http://www.edonkey2000.com.