監聽式網際網路監測系統設計實務探討

張中一
國立中正大學電機所
嘉義縣民雄鄉三興村160號
chongyie@bigfoot.com

蔡尚育
國立中正大學電機所
嘉義縣民雄鄉三興村160號
m9023@cn.ee.ccu.edu.tw

侯廷昭
國立中正大學電機所
嘉義縣民雄鄉三興村160號
tch@ee.ccu.edu.tw

摘要[1]

萬維網(World Wide Web)已成為目前網際網路上最重要的應用，在目前的環境與技術下尚沒有人針對高速骨幹下萬維網等第七層應用的交通特性設計可長時間捕捉的工具。目前的研究由於受到技術與科技的限制，往往僅能做小規模區域性在較低速區域環境下的調查，或是專門針對某一特定的網站進行研究。

為了克服以上的問題我們利用 Coral Reef 這一套函式庫，開發了一套新的程式組可以針對高速網路（OC-3 或更高，取決於硬體設備與 Coral Reef 所能支援的範圍）進行第七層的分析，同時根據研究結果和經驗嘗試不同的系統架構以探討在使用非特製的機器時，目前技術下所能分析的第七層資訊的方法。我們也指出了在設計 OSI 第七層相關的軟硬體系統時應注意的事項，及未來發展上可能的趨勢。

一、不同的網路交通監測方法

在網際網路已經成為行銷與資訊流通的重要管道之際，監測以及分析網際網路的特性就成了研究以及產品設計人員在設計產品時相當重要的參考依據。傳統上要做到網際網路監測有幾種方法：可以分為在伺服器端讀取由伺服器產生的紀錄檔、使用訂製的瀏覽器，以及以監聽的方式抓取網路交通。每一種方法各有其優缺點：透過伺服器的紀錄檔可以完全地瞭解在伺服器端所有的行為，同時所需要的處理能量亦最少。缺點則為透過伺服器的紀錄檔無法得知尚未進入伺服器時網路交通的變化，以及使用者的行為，同時要廣泛地取得不同公司所擁有伺服器的記錄檔亦有其困難。透過訂製的瀏覽器可以完全瞭解到使用者在不同的網站以及網頁間的行為；缺點則是目前通用的瀏覽器例如 Microsoft™的 Internet Explorer 以及 Netscape 公司的 Communicator 等都屬於封閉式的系統，研究人員則無法仰賴這些軟體取得相關的資訊，Claffy [1, 2]在其提出的文獻中有詳細的說明。開放式（open source）的瀏覽器例如 Mozilla [3]等則缺乏使用者，所取得的數據往往流於使用者過少而缺乏代表性。以監聽的方式偵測網路交通則是介於兩者之間，可以充分地瞭解到在網路上某一節點所有經過的伺服器與使用者交通的資料，以最小的代價即可監視多點的交通。缺點則介於使用伺服器記錄檔與訂製化瀏覽器之間，無法精確瞭解到伺服器的動作以及使用者的行為。

為了能夠充分瞭解網際網路交通的特性，同時為了能替將來發展 OSI 第七層網路裝置時做為產品發展的參考依據。我們發展一套架構藉以偵測並瞭解 Web Traffic 的特性。在這裡我們不對觀察統計到的研究數據作分析報告，而是分享我們系統架構的設計經驗。

[1] 本計畫由工研院電通所 N300 贊助，合約號碼 T2-90057。

二、實驗環境和系統硬體設備

先前的研究，尤其是已發表的有關 web 的研究通常侷限於針對本身實驗室的交通或是在某一個特定網站上架設某些讀取 log 檔的軟體藉以監測從某一特定網站出來的某些特定應用的交通。Mah [4]在 1997 年針對 HTTP/1.0 交通作了一個調查式的研究，所描述的環境就僅是區域的實驗室環境。Mena 等[5]則在特定的網際網路收音網站上架設 sniffer 類的軟體，透過伺服器行為來瞭解某一類特定應用的特性。這些研究的結果往往會因為使用者所選擇的環境過於特殊化而缺乏普遍的代表性；同時受限於研究時的頻寬限制，所監測往往都只有低速的 10Mb/sec。在現今逐漸往高速網路邁進的時代，研究人員以及產品設計人員迫切需要針對最新較高速的網路以及負荷較重、較普遍性環境作設計的研究報告。

由於取得商業網站的伺服器記錄檔有實質上的困難，而廣泛地散佈特製的瀏覽器並要求使用者取代目前被廣泛使用的 Internet Explorer 以及 Netscape Communicator 亦屬不可能的任務。因此透過監聽的方式取得所需的資料則成為代價最少，取得資料最具代表性的方法。

我們所使用的系統架在國立中正大學的計算機中心。國立中正大學計算機中心是雲嘉地區學術網路的區網中心，雲嘉地區所有教育單位對外的網路交通都必需要經過中正大學連接各 ISP 以及國家高速電腦中心的機房[2,3]。整個網路的拓樸如圖 1 所示。雲嘉區網的主要伺服器是一臺由教育部所擁有的 CISCO 7513 路由器。雲嘉區網對外聯絡的幹道有兩個方向，一個經由數個 T1 或是乙太網路連接民間的 ISP；另一則為以一個 OC3 連線連接 HINET 的機房後再連接至國家高速電腦中心。其頻寬的分配方式為 70Mb/sec 供連至國家高速電腦中心之用、50Mb/sec 供連接至 TANET2、20Mb/sec 專供連接 HINET 以及雲嘉區網，最後並提供 10Mb/sec 供爆發交通之用。在 CISCO 路由器之間以及 HINET 機房之間則有一個 CISCO LS1010 SWITCH。

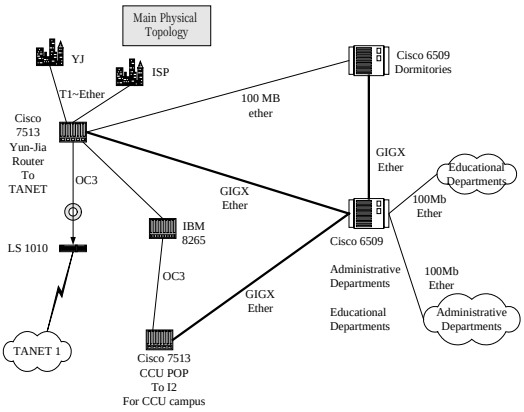

圖 1. 雲嘉區網中心拓樸圖.

我們的監測系統則為在 CISCO 7153 ROUTER 以及 LS 1010 SWITCH 之間。除少部分流向 Seednet、英普達以及遠傳等民間 ISP 的交通之外。所有自雲嘉區網流向台灣最重要的 ISP—HINET 以及流向國外的交通都會經過我們監測用的骨幹。為了能夠快速且自訂化地量測 ATM 光纖上的細胞（cell），我們使用了 Coral Reef 這一套軟體。這一套軟體提供了在 FreeBSD 上原生的 Fore ATM NIC 驅動程式以及許多的函式以讓研發人員針對自己的需求自訂程式取得網路的資訊。Coral Reef 函式庫並提供通透的 AAL 封包讓程式設計人員可以在光纖網路上通透地取得第三層乃至第四層的資料。

Coral Reef 運作的方式如圖 2 所示，在捕捉光纖網路時必須要透過兩個分光器連接在 ATM 網路的光纖上(一條 ATM 鍊結由兩條光

[2] 雲林縣及嘉義縣政府另有成立自己的學術網路中心，供縣級教育單位連線之用，此一部份交通不會出現在雲嘉區網對外連線的骨幹上。

[3] 雲嘉地區部分學校為取得較快的對國外連線頻寬部分會自行向 HINET 等 ISP 公司租用 ADSL 線路，此一部份交通不會出現在雲嘉區網對外連線的骨幹上。

纖所組成，一條負責去，另一條負責回。)，將光訊號分出之後,送到捕捉系統的兩張Fore ATM NIC 或是其它 Coral Reef 支援的 ATM 卡上，進行資料的後續處理。

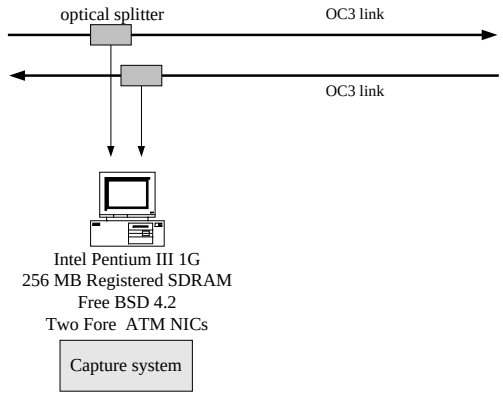

圖 2 Coral Reef 捕捉架構圖.

所使用的機器則為三部通用的 PC，一部為 IBM eServer 220，配有 Pentium III 1G、256MB Registered SDRAM、Iwill IDE RAID 卡、兩臺 IBM 7200 rpm IDE Bus 40GB HDD 組態成 RAID 0。另二部則為通用 PC，組態為 Pentium 733、128MB SDRAM、二臺 Quantum 7200 rpm IDE Bus 40GB HDD 組態成 RAID 0 和 Pentium 450、256MB SDRAM、一臺 Quantum 7200 rpm IDE Bus 40GB HDD。

三、系統架構設計探討

以監聽的方式量測網際網路且進行第七層的分析最大的問題在於所需要的儲存空間以及對電腦硬體性能的要求，諸如處理器時脈、記憶體、和 I/O 介面。為了要進行第七層的資料分析，勢必要解譯每一個封包的酬載。經過我們的實驗顯示在骨幹網路連續捕捉一小時並記錄一個方向的所有酬載,我們產生了約 32GB 的資料。對於任何不管是要線上或是離線處理的系統以及研發人員而言，這樣的資料量都太大。為了解決這樣的問題，我們嘗試計了數個架構,並比較其結果以瞭解其優劣所在。

架構一：

如圖 2 所示的架構，用一台 PC（為 IBM eServer 220，配有 Pentium III 1G、256MB Registered SDRAM、Iwill IDE RAID 卡、兩臺 IBM 7200 rpm IDE Bus 40GB HDD 組態成 RAID 0）處理細胞擷取和分析統計的工作，我們除了利用 CoralReef 延伸程式組擷取細胞重組並做封包處理外，還得搭配 Berkely DB 做一些輔助封包分析統計的資料庫工作。這個架構除了架構簡單、封包分析程式方便外，有太多的瓶頸無法突破。例如，一台電腦無法同時擷取封包並作分析處理和資料庫的工作，它的極限為：1、雙向捕捉到 OSI 第四層封包結構並儲存或 2、單向捕捉到 OSI 第七層封包結構並儲存。使用這兩個方式都只能離線作分析，無法即時分析統計封包，故得轉儲成封包記錄檔。如此瓶頸造成儘管雙向能捕捉到 OSI 第四層，但部分應用的資訊還是包含在其封包的酬載中，而這些酬載因架構瓶頸而無法轉儲起來，這是一台 PC 捕捉雙向交通的最大缺點。此外，雖然只抓單向可以整個 link 上全部 HTTP 封包完整捕捉之後再離線分析，但一些驗證工作和 HTTP session 的詳細互動情況無法完全掌握，因為我們最多只能看到單方向遞送的封包訊息。因此，架構一雖然架設簡單，但所處理的事亦不能太複雜，能力有限。

架構二：

如圖三所示，我們採用 Agent-Manger 架構，將原本的架構一多放一台 PC（為 Pentium 733、128MB SDRAM、二臺 Quantum 7200 rpm IDE Bus 40GB HDD 組態成 RAID 0）進去，原本的 IBM 1G Hz 的 PC 依然擔任擷取交通工作的 Agent 端，並將所重組好的封包透過 Socket programming 送往 Manager 端 PC。而另一台 Pentium 733Hz 的 PC 負責將從 Socket 收到的資料搭配資料庫即時地加以處理分析統計，為了克服將所有封包傳到遠端所佔用的龐大頻

寬，我們使用了自訂的固定長度格式來儲存所需要的資訊。圖 4 以及圖 5 說明了我們所使用的格式，長度為 24 位元組，所儲存的資料有時間標記（Timestamp，從 ATM 卡上的時鐘取得）、來源 IP 位址、目的 IP 位址、來源 TCP 埠號、目的 TCP 埠號、TCP 旗標，以及封包長度。

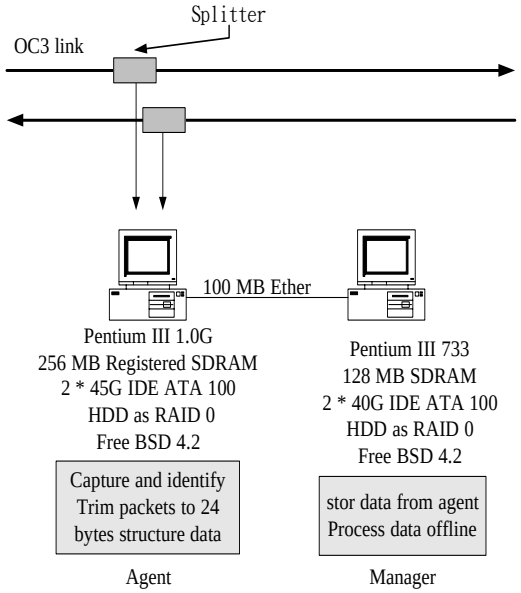

圖 3 Agent- Manager 架構式的遠端儲存方案

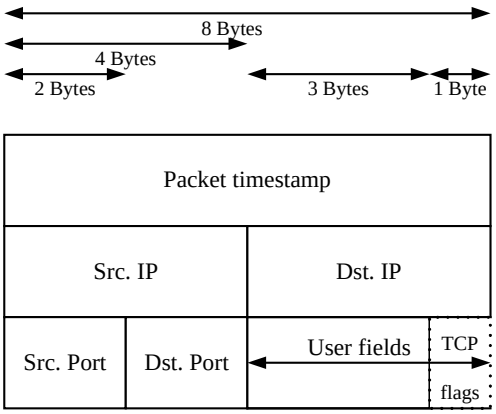

圖 4 所用的固定長度資料結構

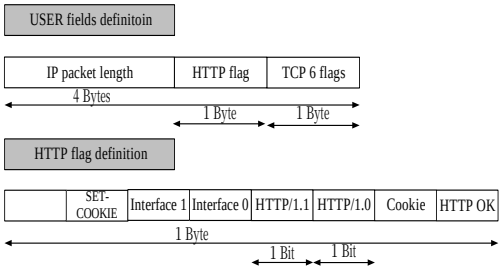

圖 5 使用者欄位(USER field)的定義

其中我們對一個 flow 的定義跟通用的定義不同，因為缺乏 SYN-FIN 完整的資訊（尚無法完整捕捉到雙向交通），故採用每一個 flow 所傳來的第一個帶有 SYN 的封包即視為一個 flow 的開始。而將收到的第一個帶有 FIN 或是 RESET 的封包視為一個 flow 的結束。我們除了使用 TCP/IP 的 SYN-FIN 機制當作 flow 的建立與中斷之外，並參考 Claffy [6]的作法使用了固定的逾時值作為某些 flow 閒置太久時將其自 flow table 移除的參考。每一個 flow 使用一個鍊結串列來表示，圖 6 說明了這個鍊結串列的結構。

The latest packet's timestamp of this flow	4 bytes
Total packet count of this flow	4 bytes
timestamp of packet n	8 bytes
timestamp of packet (n-1)	8 bytes
.....	8*(n-3) bytes
timestamp of the 1st packet	8 bytes

圖 6 系統二所使用之資料庫所儲存資料的結構

為了避免儲存每一個封包的 timestamp 所帶來的龐大儲存空間的消耗，我們提出了一個修正的資料庫儲存結構可以用較少的儲存空間消耗代表相同的資訊。圖 7 說明了所使用的資料結構格式。我們事先設定了我們需要的資料有：目前某 flow 最近到達封包的時間標記（供逾時檢查之用）、總共的封包個數（供

後處理時 manager 之用）、flow 內最小封包抵達間隔、最大封包抵達間隔、平均封包抵達間隔、以及連續兩個供計算封包抵達時間標準間隔用的八位元數字，最後另有一個此 flow 內第一個封包的時間標記（供計算 flow 的 duration 之用）。透過一個這樣 64 位元組長固定資料結構，即可取代原先長度不固定的 flow 表所能代表的資訊。

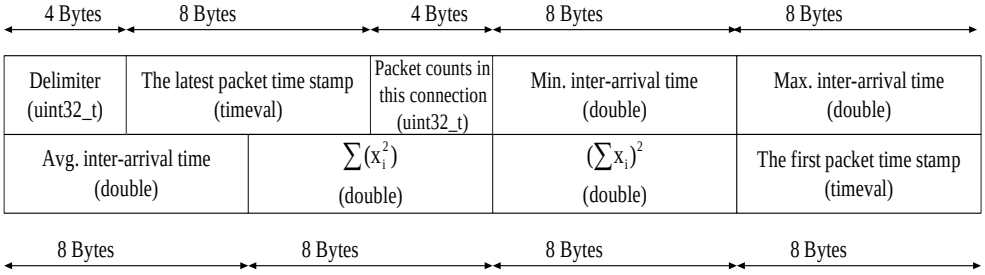

圖 7 縮減後的 flow 資料庫儲存結構.

架構二雖然可以即時的處理到封包 OSI 第七層的資料，但同時面對兩條每秒一百多萬 bits links 的流量交通，儘管是 IBM 1GHz 的 PC 還是無法來得及處理而遺失細胞，這說明了 I/O 是其一大瓶頸因為在細胞接收、細胞重組的負載非常大，儘管 CPU 並不忙碌但卻無法即時的處理完擷取在 Buffers 中的細胞而造成不得不丟棄細胞以容納新進來的細胞。因此，此架構亦無法完整掌握 HTTP session 的互動情況。

架構三：

我們根據經驗再多架設一台 PC（為 Pentium 450、256MB SDRAM、一臺 Quantum 7200 rpm IDE Bus 40GB HDD）到架構二中，整個架構和圖 3 差不多，只是多了一台電腦，並角色互調一下而已。我們將 IBM 1GHz 的 PC 當作 Manager，而另外兩台 PC 擔任 Agents 分別捕捉擷取單一方向的交通，一樣和架構二做相同的系統程序，將 Agents 擷取過濾後的封包資料透過 Socket Prodramming 將其送往 Manager 處理，但這樣的系統架構卻能分攤頻繁的 I/O 所帶來的負荷，並突破無法即時處理雙向交通的瓶頸。

傳統的網路監測器不論是屬於透過專屬硬體的產品，例如 NAI 公司的 OC-3 sniffer 或是惠普公司的 protocol analyzer 等通常必須要透過專屬的硬體且往往所能監控的時間有限。我們所設計的架構是為了長時間，能在骨幹上無封包漏失的量測為目標。透過通用的 PC 以及免費的軟體，我們的確部分地到達了這個目的。然而由於電腦硬體上的限制，往往所能進行的監測會受到限制。CPU 時脈的影響亦值得考慮。當我們使用的兩台 PC 除中央處理器和記憶體之外其餘次系統皆幾乎相同。離線處理相同的原始資料，所消耗的時間則會有四小時（Pentium III 1G）與六小時（Pentium III 733）的差距。經過分析，我們發現在執行程式期間系統所消耗的記憶體從未大於 64MB，虛擬記憶體所使用的數量為零[4]。因此 CPU 的性能在此則帶來重要的影響。另一點值得注意的是，I/O 的性能在讀取資料時亦會帶來顯著的影響。同樣的 Pentium 733 系統當我們將 RAID 裝置解除時，由於系統 I/O 的性能降至原來的一半（接近一半，不及一半）。當讀取數量極大的原始資料時，慢速的 I/O 次系統會明顯地拖慢系統的表現。系統無法來得及處理來往的封包交通，造成細胞遺失。於是我

四、架構效能探討

[4] 我們所使用的資料庫系統，會將檔案儲存在硬碟上。這一部份並不使用到虛擬記憶體。

們繼續將架構擴充設備，將 Agent 端放置兩台電腦，每台各分別捕捉一個方向的 links：incoming traffic 和 outgoing traffic。再透過 socket 傳送到 Manager 端的電腦執行分析或資料庫歸檔等後端程式處理。從中，我們發現原本卡在 I/O 問題（細胞接收、重組等）造成的細胞遺失，因交通負載的攤分而使 Agent 端電腦的 I/O 負載減輕許多而不再遺失細胞。而後端程式也因系統設計時未將全部 HTTP 封包的酬載送往 Manager 端電腦，只是一些為達到某目標測量目的而過濾後的封包欄位和部分第七層酬載，Manager 端尚可應付，若日後想觀察的更仔細，manager 端需以伺服器叢集的架構來負責大量的 CPU 性能要求與 I/O，促使處理封包的能力更強，作更進一步的分析處理。

五、結論

第七層網路資料的分析與監測是一項需要消耗大量運算資源與儲存空間的研究。在網際網路上大部分的交通都已經是萬維網（World Wide Web）應用下，分析監測第七層交通即會帶來大量儲存空間需求。在本地端儲存媒體的容量有限，而諸如磁帶等大量存媒體性能不符所需時，遠端儲存裝置或是管理者例如 SAN（Storage Area Network）等勢必為未來的趨勢。然而系統設計者宜注意當與遠端儲存媒體之間的連線是採用 TCP/IP 方式連線時，維護 TCP/IP 堆疊所帶來龐大 CPU 負擔。在 Alacritech [7]公司的產品中提出了一套將 TCP/IP 堆疊以硬體方式完成，以減輕中央處理器負擔的解決方案也許有助於解決這方面的問題。

我們的研究亦指出，系統設計人員在規劃儲存結構時應審慎地考量是否能用固定的格式資料取代保留所有相關記錄的作法。我們設計程式經驗中發現了如果已經確定所要量取的資料為最小封包抵達間隔、最大封包抵達間隔等資訊時，在離線建立 flow table 時每個 flow 僅以固定的欄位可以比保留每一個封包的時間標記，帶來長足的性能改進。

仰賴現存的科技，要能夠完全無漏失地線上分析網際網路的資料是一件困難的事情，尤其是當有線上針對所欲取的資料送進資料庫時更是困難。我們建議相關領域的分析人員不要堅持無漏失的作法。少量的漏失要比消耗大量的金錢與時間來解決漏失的問題更值得。

最後，當需要進行雙向的捕捉分析時，我們建議研究人員以 TCP 的 sequence number 作為同步起始的依據。因為網路卡驅動的時間並不一致。我們所建立出的 agent-manager 系統，可以成為將來應用層分析時儲存與分析的基本藍圖。

六、參考文獻

[1] K. Claffy, “Measuring the internet,” IEEE Communication Magazine, vol. 4, Jan 2000, http://www.caida.org/outreach/papers/ieee0001/

[2] T. Monk and K. Claffy, “Internet data acquisition & analysis: Status & next steps,” 1997, http://www.caida.org/outreach/papers/data-inet97.html

[3] “Mozilla,” http://www.mozilla.org

[4] B. Mah, “An empirical model of http network traffic,” in Proc. INFOCOM 2000, 2000, vol. 1, pp. 101-110

[5] Mena, A., and Heidemann, J., “an Empirical Study of Real Audio Traffic,” in Proc. of INFOCOM 2000, P. 101-110, vol. 1, pp. 101-110

[6] K. Claffy, et al., “A parameterizable method for Internet traffic flow profiling,” Selected Areas in Communication, IEEE Journal on, Vol. 13, Iss. 8, pp. 1481-1494, Oct. 1995

[7] “Alacritech,” http//www.alacritech.com