以開放原始碼軟體做簡易網路連線流量統計

謝志昌，蔡哲民

高醫電算中心

cch@kmu.edu.tw, tjm@kmu.edu.tw

論文摘要

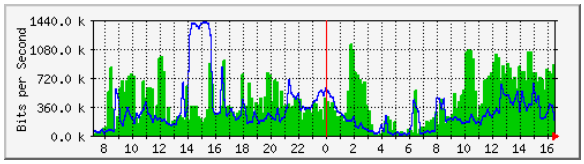
TANet 骨幹的壅塞是一個整體現象，若只仰仗區網中心去做頻寬管制是不足的,各校至區網中心的連結若能先做過濾，限制非必要的連線，那麼應能減輕 TANet 骨幹部份的壅塞狀況，也能對自校的數據專線做最佳的利用. 若各個學校欲在連上區網中心前做連線過濾，則需先得知校內各網路節點消耗專線頻寬的狀況,在這個需求下，MRTG 軟體已無法滿足，因爲它只能顯示出專線頻寬的整體利用狀況，卻無法提供校內每一個網路節點消耗專線頻寬的資訊，在本文中將提出一個可行的解決方案: 在低廉的花費 (一台具網路卡的 PC) 下，運用開放原始碼軟體 bpft，去紀錄進出校園網路對區網中心連結的每一次連線特徵，如:來源位址，目的位址，服務埠號，及資料傳輸量等，進而統計出消耗該專線頻寬的主要用戶，以協助網管人員做連線或頻寬的管制，根據高醫電算中心試行一學期的結果顯示，這項辦法，能有效地阻擋校外地下網站，並使得 T1 專線的頻寬在上班時段仍留有餘裕. 未來如果再加上自動頻寬限制功能，配置於對外路由器的前端，可成爲專線流量統計 (針對網點) 與頻寬管理的工具.

關鍵字: 網路管理，流量統計, Open Source

1. 動機與簡介

近年來由於網際網路的蓬勃發展，TANet 骨幹時常發生壅塞狀況，若 TANet 的使用者只期待增加骨幹頻寬，則由於寬頻專線費用仍高，使得 "開源" 有限.或者仰賴區網中心去管制網路流量，那成效仍然非常有限，因爲 TANet 骨幹壅塞是整體的現象，尖峰時刻，骨幹上流通的究竟是些什麼資料? 區網中心難以得知,所以也就無法有效地過濾掉不當的連線，由於網際網路的便利，和部份使用者不瞭解網際網路成文與不成文的規範，使得交換無合法版權軟體，MP3 音樂，或電影 VCD 的資料傳輸行爲常常發生，很輕易地就能將寶貴的頻寬消耗殆盡，因此在骨幹末端做"節流" 的工作反而更顯得有必要，如果各個學術單位在連至區網中心的連結上能先做過濾，限制非必要的連線，如勸戒作非正當用途的使用者，或直接阻擋地下網站，那麼應能減輕 TANet 骨幹部份的壅塞狀況，也能讓自校的數據專線發揮最大的效用.

找出過度消耗頻寬的用戶是 "節流" 的首要工作，這個時候我們發現 MRTG [1] 軟體 (請參考圖一，以及教育部所做的 TANet 骨幹流量統計 http://www.edu.tw/tanet/backone/) 的功能已顯不足:



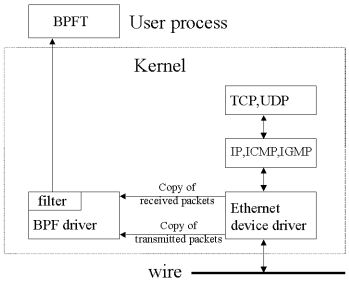
圖一: MRTG 的輸出

當網管人員看到 MRTG 的流量圖是 "一片慘綠" (流入滿載),或 "藍線高掛" (流出滿載) 時，必定想問: 到底是誰在消耗頻寬? 這問題卻是 MRTG 無法回答的，因爲它只能顯示出專線頻寬的整體利用狀況，卻無法提供校內每一個網路節點消耗專線頻寬的資訊. 這個問題常見的解決方案是購買路由器的 flow accounting 功能及其相關的配合硬體 (如: 高級工作站) 和軟體 (如: 工作站作業系統及 SQL 資料庫)，然而對日益結据的學校經費，這種解決方案的代價頗高，但以我們在高雄醫學大學電算中心的工作經驗: 只要用一台有網路卡的 PC，再配合開放原始碼軟體 bpft, 我們一樣可以在低廉的花費下，達到同等的目的.

以下本文就校園網路對外專線上連線之紀錄和統計的工作原理，實作與結果，討論與比較，及結論與未來工作做敘述.

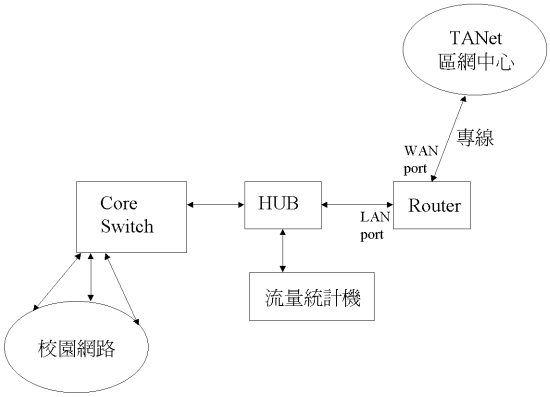
2. 工作原理

本文中所選用的開放原始碼軟體 bpft (The BPF Traffic Collector) [2],爲俄羅斯西伯利亞電信學院 CAD 實驗室所發展, 其工作的原理爲: 利用 BPF (BSD Packet Filter) [3], 收集同一實體線路上所有往來的封包(packet), 紀錄該網段上每一次連線的來源位址 (source IP address), 目的位址 (destination IP address), 使用的服務埠號 (service port number), 及傳輸的資料量.在此附帶說明 [4] BPF 是 BSD 系 UNIX kernel 提供的一個 driver, 它將乙太網路介面切入混亂模式 (promiscuous mode: 在此模式下該介面會把實體線路上所有的封包都收進來), 然後 BPF driver 會獲得乙太網路介面 driver 收送的每一個封包的複本, 再流入使用者指定的過濾器 (filter), 通過該過濾器的封包就會送達使用者行程, 如圖二:

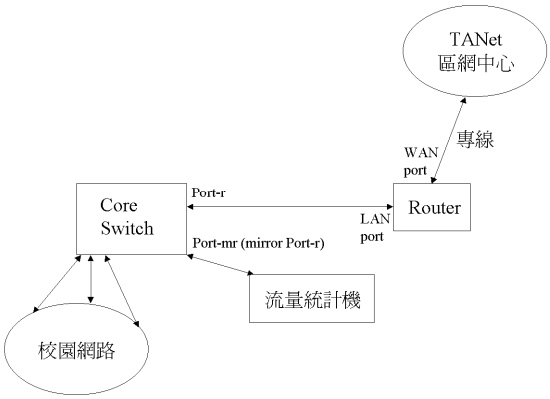


圖二: BSD Packet Filter

如果我們想要統計進出學校對外數據專線的網路連線流量, 那麼就必須去統計進出該專線的每一個封包, 而聽取這些封包最適合的地方就在專線路由器的 LAN port,所以只要讓流量統計機器的網路介面能聽到對外路由器 LAN port 的封包, 就可以利用 bpft 去做網路連線的流量統計, 一個簡單的方法是利用一台集線器 (HUB,在此不能用 Switch, 否則流量統計機器會聽不到封包) 讓流量統計機器和對外路由器的 LAN port 位於同一實體線路上, 如圖三所示, 或者如果校園網路的 core switch 支援 port mirroring 功能, 則只要設定 core switch 上的一個 port 去 mirror 對外專線的 traffic, 再把流量統計機器接上這個 mirror port, 如圖四所示, 按以上的配置完成後, 流量統計機器可藉由 bpft, 收集所有進出專線的封包, 把每一次的連線特徵都紀錄在流量統計機器的儲存媒體上, 等待管理者做進一步分析.



圖三: 以 HUB 配置流量統計機



圖四: 以 Core Switch mirror port 配置流量統計機

3. 實作與結果

3.1 工作平台

我們執行流量紀錄與統計的軟體平台採用一台低階的 PC, 其配備爲 Pentium 200 的 CPU, 記憶體 32 MB, 2 GB 的 IDE 硬碟一顆, 及一個 10Mbps 的乙太網路介面, 作業系統使用 FreeBSD 3.4-STABLE [5] (註: 若採用 Linux [6]一樣能夠支援). 由於高醫校園網路連至區網中心中山大學的是兩條 T1 專線 (TANet 和 TANet/I2), Router LAN port 的網路介面是 10 BaseT, 所以流量統計機採用 10Mbps 的乙太網路介面就足夠了.

3.2 bpft 的安裝與使用

由於 bpft 已在 FreeBSD 的 ports (已移植至 FreeBSD 的應用程式集合) 中, 所以有 bpft 的 package 可直接安裝.bpft 由一組工具程式所組成, 我們目前利用到的主要有三

支程式:
trafd -- 收集 TCP/UDP Traffic 的 daemon 程式, 藉由聽進來的封包, 去取得詳細的連線資訊, 包括個別連線兩端的 IP 位址, 使用到的埠號, 及傳輸資料量的大小, 其所收集到的資料存成二元紀錄檔 (此紀錄檔無法以人工判讀).
traflog -- 可將 trafd 的二元紀錄檔轉換成可供人工判讀的格式.
trafshow -- 可即時動態全螢幕顯示目前網路流量狀況, 讓網管人員能隨時得知有那些校外網點正在和校內網點傳輸大量資料.

目前藉由 bpft 這三支程式的功能, 在高醫校園內運作的專線頻寬管理方式有兩種:
1. 上班時段由電算中心工作人員不定時觀看 trafshow 的輸出, 若發現有大量資料傳輸 (目前標準定爲 100 MB), 經執行判斷程序 (domain name, port scan),如果資料提供者是不明站台 (通常是 Windows 上用 Server-U 架設的 FTP 站),即在 Core Siwtch 上設定 IP Filter, 以阻擋該站台.
2. 公布每日上, 下班時段及每月的流量排行榜 [7], 對連續上榜數日的網點, 與其所屬單位聯絡, 瞭解其用途.

流量排行榜的產生方法如下: 首先執行 trafd 把每天 24 小時網路流量狀態紀錄下來, 欲做統計時, 藉助 traflog (詳細用法參考附錄 A.1) 這支工具程式把 trafd 的紀錄檔轉成可供人工判讀的格式, 如列表一所示:

列表一: 經 traflog 轉換過的紀錄檔 kmu-traf-day.log 的部份內容

(cs0) netflow.kmu.edu.tw at Aug 9 08:00:01 - Aug 9 08:44:25
Summary: 504199684 data bytes, 538310188 all bytes, 2052 records

From	Port	To	Port	Proto	Data
140.116.7x.25	20	163.15.15x.2	client	tcp	137366204
206.132.18x.167	80	163.15.16x.75	client	tcp	59638524
140.117.1x.12	client	163.15.15x.32	119	tcp	55053059
163.28.13x.1	3128	163.15.18x.2	3128	tcp	40194295
163.28.13x.1	3128	163.15.15x.31	3128	tcp	10905238

經 traflog 轉換過的格式, 每一行都紀錄著一個 session 的連線,包含來源位址, 目的位址, 服務埠號, 和資料傳輸量, 我們只要把校內節點每一個 session 的資料傳輸量累加起來, 再經過排序, 就可以知道消耗頻寬的主要用戶. 我們藉一個 Perl script (請參考附錄 A.2) 去統計 (包括進和出的方向) 前十大用戶, 結果如列表二:

列表二:

高醫對外網路流量監測報告 (上班時間)
監測時間: 00-08-09 08:00 - 00-08-09 17:30

KMU TOP 10 flow-in hosts: total flow-in: 4620531071 bytes
---

1)	163.15.15x.31	618061786	13%
2)	163.15.15x.2	549967494	11%
3)	163.15.15x.32	371509837	8%
4)	163.15.17x.203	285515805	6%
5)	163.15.16x.75	156776979	3%
6)	163.15.15x.34	152299841	3%
7)	163.15.18x.2	123767674	2%
8)	163.15.17x.121	119477354	2%
9)	163.15.17x.73	117865012	2%
10)	163.15.16x.222	94713779	2%

KMU TOP 10 flow-out hosts: total flow-out: 1519042610 bytes
---

1)	163.15.16x.23	229600614	15%
2)	163.15.15x.78	203016280	13%
3)	163.15.18x.1	167774647	11%
4)	163.15.15x.1	128371857	8%
5)	163.15.17x.73	122319885	8%
6)	163.15.15x.2	120024139	7%
7)	163.15.17x.23	77855991	5%
8)	163.15.15x.31	72438164	4%
9)	163.15.15x.1	37041963	2%
10)	163.15.15x.32	30985376	2%

4. 討論與比較

4.1 紀錄檔大小及紀錄分析速度

1. 紀錄檔大小

以高醫校園網路對外專線 (兩條 T1) 的 89 年 9 月份的流量爲例 (流入 236 GB, 流出 77 GB), trafd 的二元紀錄檔的大小爲 48.5 MBytes, 平均來說紀錄檔每日增加 1.6 MBytes, 經 traflog 轉換爲可供人工判讀的格式的紀錄檔大小爲 194 MBytes. 如果我們要保留一年份的網路連線紀錄,則只需要約 600 MB 的儲存空間 (以儲存二元紀錄檔計算, 供人工判讀的紀錄檔可由二元紀錄檔轉出, 故不需要儲存), 目前市面上常見的硬碟遠超過這個需求.

2. 紀錄分析速度

以高醫校園網路每日對外專線上班時段或下班時段的流量, traflog 轉換紀錄檔格式再加上 Perl script 做出前十大流量排行榜的時間, 在現行硬體平台上, 均於 2 分鐘內可以完成. 而每個月的統計結果, 以 89 年 9 月份爲例, traflog 轉換格式耗時 39 分 29 秒, 而 Perl script 加總, 排序耗時 27 分 26 秒. 在實作上, 可以藉由修改 traflog 的原始程式, 使其在解讀出二元紀錄檔的資料後, 直接加

總, 排序, 如此一來分析紀錄的效率應該會更好.

4.2 網路連線流量統計附帶的益處

1. 協助稽核網際網路安全

根據以往的經驗, cracker 經常做跨國性的攻擊或入侵, 而校園網路的主機常常成爲被覬覦的對象, 若我們已在校園網路骨幹的出入口做連線流量統計, 就可以事先從紀錄中去分析是否有來自國外的異常連線, 或者事後協助調查攻擊或入侵的上一層來源.

2. 分析網路流量特性, 藉以調整網路的組態或服務

Bpft 的紀錄包含了 TCP 及 UDP 連線的服務埠號, 我們可以藉此大致統計各種網際網路服務的使用狀況, 進而調整或新增區域性的網路組態或服務, 以求對暨有網路骨幹做更有效率的運用, 譬如: 當我們發現使用者常至區網中心的 FTP 伺服器抓取資料量大到某一個程度, 那麼就可以考慮 mirror 該 FTP 伺服器的部份內容至校內的 FTP 伺服器.

4.3 其他網路連線流量統計軟體

能夠做網路連線流量統計的開放原始碼軟體, 並不止 bpft 而已, 據我們所知, 至少還有 ipfm [8], iplog [9], NeTraMet [10], ipmeter [11], 及 ntop [12] 等等, 功能上互有所長. 例如 ipfm 的使用較爲簡單, 不必另外撰寫統計總流量的程式, 但它目前 (0.10.4 版) 的功能只能累計一段時間內, 紀錄進出某一網路節點, TCP, 或 UDP 的傳輸資料量, 且不含使用埠號, 它的設定方法簡單, 只需要一個設定檔 ipfm.conf, 如列表三:

列表三: ipfm 的設定檔

```
#
# log traffic into KMU hosts
#
LOG TO 163.15.15x.0/255.255.255.0
LOG TO 163.15.15x.0/255.255.248.0
LOG TO 163.15.16x.0/255.255.240.0
LOG TO 163.15.17x.0/255.255.252.0
LOG TO 163.15.18x.0/255.255.255.0
FILENAME /var/log/ipfm/ipfm-ToKMU-%y.%m.%d
TIME 1 day
SORT IN
NORESOLVE
```

根據列表三這個設定檔, 我們只要在 0 點 0 分啓動 ipfm, 那麼在 /var/log/ipfm 目錄下, 就會產生每日進出學校專線的流量統計紀錄, 結果如列表四:

列表四: ipfm 的 log: ipfm-ToKMU-00.08.08 的部份內容

HOST	IN	OUT	TOTAL
163.15.15x.32	714113717	0	714113717
163.15.15x.31	496977752	0	496977752
163.15.18x.2	419175827	0	419175827
163.15.16x.75	383948348	0	383948348
163.15.15x.1	186610748	0	186610748
163.15.15x.34	137459786	0	137459786
163.15.16x.218	110149703	0	110149703
163.15.17x.73	84550256	0	84550256
163.15.18x.1	53428645	0	53428645
163.15.17x.12	51262188	0	51262188

如此就可以知道 2000.08.08 當日學校內的前十大網路頻寬 (流進) 消耗者.

Iplog 和 bpft 的 trafd 功能相近, 但其紀錄檔爲人工判讀格式, 紀錄時間一久, 會消耗掉可觀的硬碟空間, 且缺乏即時動態顯示網路連線狀況功能.

NeTraMet 的設計較爲複雜, 設定不易.

Ipmeter 需要較多其他的軟體配合: NeTraMet, Apache, PHP, 和 PostgreSQL, 是一個接近商業軟體品質的 IP Flow Accounting 系統, 因此需要等級較高的硬體平台支援.

Ntop 能提供文字介面或 Web 介面的即時流量統計資訊, Web 介面模式仍不夠穩定 (執行數小時後, 在使用者以瀏覽器存取流量統計資訊時, 常會發生記憶體分段錯誤而結束).

5. 結論與未來工作

使用上述的方法做專線網路連線流量統計已在高醫校園內試行了一學期, 事實證明在開放原始碼軟體的幫助下, 以低廉的花費, 我們一樣可以取得我們想要的資訊 -- 到底是誰在消耗頻寬, 在阻擋了一些地下站後, 使得 T1 專線的頻寬在上班時段能時常保有餘裕. 如果各校在對區網中心的連結上都能採用類似的管制方式, 那麼應該能減輕部份 TANet 骨幹壅塞的狀況.

由於本文上述的專線頻寬理管理方式, 只做到統計自動化, 在管理上尚需不少人力去執行網路交通狀況監看及地下站阻擋, 也尚未實現頻寬限制 (bandwidth limitation), 未來將朝頻寬管理自動化研究, 可預見的雛型是以現有流量統計機再加上頻寬限制自動化的功能.

附錄 A

A.1 traflog 的使用方法

```
#!/bin/sh

BEGIN=`date +'%m%d`0800
END=`date +'%m%d``1730
traflog -a -n -b ${BEGIN} -e ${END} > /var/log/kmu-traf-day.log
```

A.2 統計消耗頻寬前十大的 Perl script

```
#!/usr/bin/perl

$log = $ARGV[0];

%client = ();
%server = ();

$bcnt = 0;
$tbcnt = 0;
if (open(FP, $log)) {
   while (<FP>) {
      if(/^([\d\.]+)\s+[\w\-]+\s+([\d\.]+)\s+
          [\w\-]+\s+\w+\s+(\d+)/) {
         if (inKMU($1)) {
            if ($server{$1}) {
               $server{$1} += $3;
            } else {
               $server{$1} = $3;
            }
         } elsif (inKMU($2)) {
            if ($client{$2}) {
               $client{$2} += $3;
            } else {
               $client{$2} = $3;
            }
         }

      } elsif (/^\s*Summary:\s*(\d+)/) {
         $tbcnt += $1;
      }
   }
   close FP;
}

$total_in = 0;
foreach $c (keys %client) {
            $total_in += $client{$c};
}

$total_out = 0;
foreach $s (keys %server) {
            $total_out += $server{$s};
}

print "----------------------------------------\n";
print "KMU TOP 10 flow-in hosts: total flow-in: $total_in
bytes\n";
print "----------------------------------------\n";
$hcnt = 1;
foreach $c (sort { $client{$b} <=> $client{$a} } keys %client) {
   if ($hcnt > 10) {
      last;
   }
   printf("%d)\t%s\t%s\t%2d\%\n",
      $hcnt, $c, $client{$c}, $client{$c} * 100 / $total_in);
   $hcnt++;
}

print "\n";
print "----------------------------------------\n";
print "KMU TOP 10 flow-out hosts: total flow-out: $total_out
bytes\n";
print "----------------------------------------\n";
$hcnt = 1;
foreach $s (sort { $server{$b} <=> $server{$a} } keys %server) {
   if ($hcnt > 10) {
      last;
   }
   printf("%d)\t%s\t%s\t%2d\%\n",
      $hcnt, $s, $server{$s}, $server{$s} * 100 / $total_out);
   $hcnt++;
}

sub inKMU {
   if ($_[0] ### is the IP of KMU ###) {
      return 1;
   }
   return 0;
}
```

參考文獻

[1] Tobias Oetiker, and Dave Rand, http://www.ee.ethz.ch/~oetiker/webtools/mrtg/mrtg.html

[2] Vladimir Vorobyev,CAD lab., Siberian State Academy of TelecommunicationNovosibirsk, Russia. ftp://ftp.turbo.nsk.su/pub/unix/bpft-X.Y.tgz

[3] McCanne, S., and Jacobson, V. 1993. "The BSD Packet Filter: A New Architecture for User-Level Packet Capture" *Proceedings of the 1993 Winter USENIX Conference*, pp. 259-269,San Diego, Calif.

[4] W. Richard Stevens *TCP/IP Illustrated, Volume 1: The Protocols*

[5] FreeBSD development team http://www.freebsd.org

[6] Linux Online Inc. http://www.linux.org

[7] 高醫電算中心,高醫網況報導 http://bbs.kmu.edu.tw/cgi-bin/boards.cgi?KMU-NETSTAT

[8] Robert Cheramy, and Andres Krapf http://www.via.ecp.fr/~tibob/ipfm/

[9] Ojnk Software Design http://ojnk.sourceforge.net/

[10] Nevil Brownlee http://www.auckland.ac.nz/net/NeTraMet/

[11] IP23 Gesellschaft fur IP-basierte Dienstleistungen mbH http://www.ipmeter.com/

[12] Luca Deri http://www.ntop.org/ntop.html