rrdtool 教學

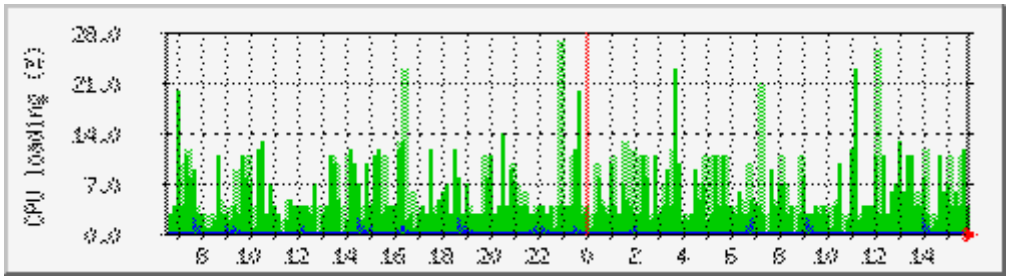
作者: abelyang <abelyang{at}twnic{dot}net{dot}tw>
version: 0.3
最後修正時間: 2003/08/27 00:10

前言

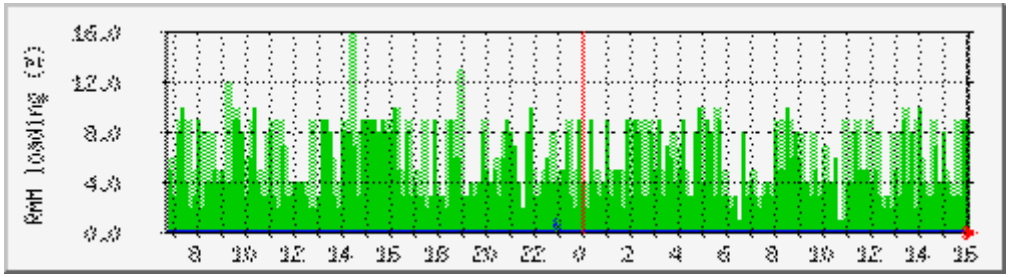
(http://www.rrdtool.org)
什麼是 rrdtool 呢 ? 其實他和 mrtg 是同一家族, 主要都是在產生 time-series 的圖檔(如流量,負載,溫度,人數.....),不過因為 mrtg 當初的考量是畫兩種資料在圖上(或四個值),後來原作者覺得不足,所以另外又開發了 rrdtool, rrdtool 本身可和 mrtg 結合,但其結合基本上僅在於將 mrtg 的文字檔的log 轉成 rrd 儲存格式,通常 user 尚需要 mrtg-rrd/rrdcgi 去轉換,不過總覺得美中不足,因為最終其實你用到的還是 rrdtool,雖然還有像 my14all (http://my14all.sourceforge.net/) 這類的 tools 可以轉換並畫圖,但其追根究底 還是以 rrdtool 為 base, 所以 rrdtool 變成了最終也是最好的選擇。

平心而論 rrdtool 的學習遠比 mrtg 來得困難,且相關文件資料也沒有 mrtg多,其中中文的參考又幾乎沒有,如果沒有較深厚的 Linux 基礎(尤其是 Shell Script) 或了解 SNMP,懂得英文及好學的心,否則是不建議學習 rrdtool 的.因為你可能很難去控制或獲得你所要的資料,亦可能難於表現圖檔。

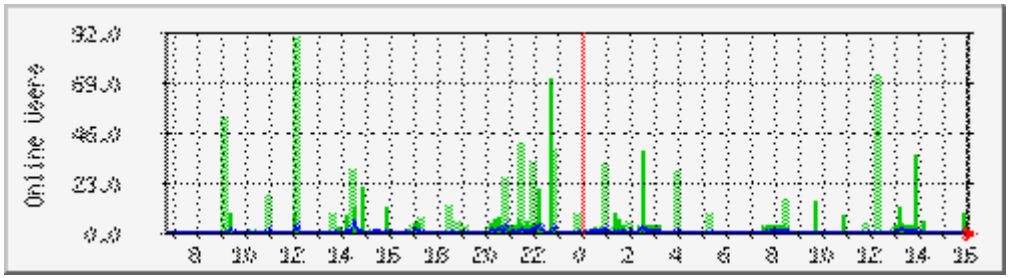
當然,每個人看法不見得相同,完全看個人需要而定,就像用 mrtg 畫 CPU Load, Memory Usage,HTTP Client, Process...., 純使用 mrtg 是較簡單且好用的, 但是你要做很多圖,每張圖之間的關聯生基本上可能需要你用眼睛去判斷.但如果使用rrdtool 可以讓你四張疊成一張,如此也較好比較出其中的因果關係,不過此時你得懂得控制圖的表現方式來達到顯示上最好的結果.基本上學 rrdtool 完全不需要有 mrtg 的經驗. 但最好對系統資訊獲得的方法(cmd/ snmp/Shell Script)熟一點會較好處理.



最大 CPU 純系統負載; 27.0 % 平均 CPU 純系統負載;5.0 % 目前 CPU 純系統負載; 0.0 %
最大 CPU 使用者負載; 2.0 % 平均 CPU 使用者負載;0.0 % 目前 CPU 使用者負載; 0.0 %

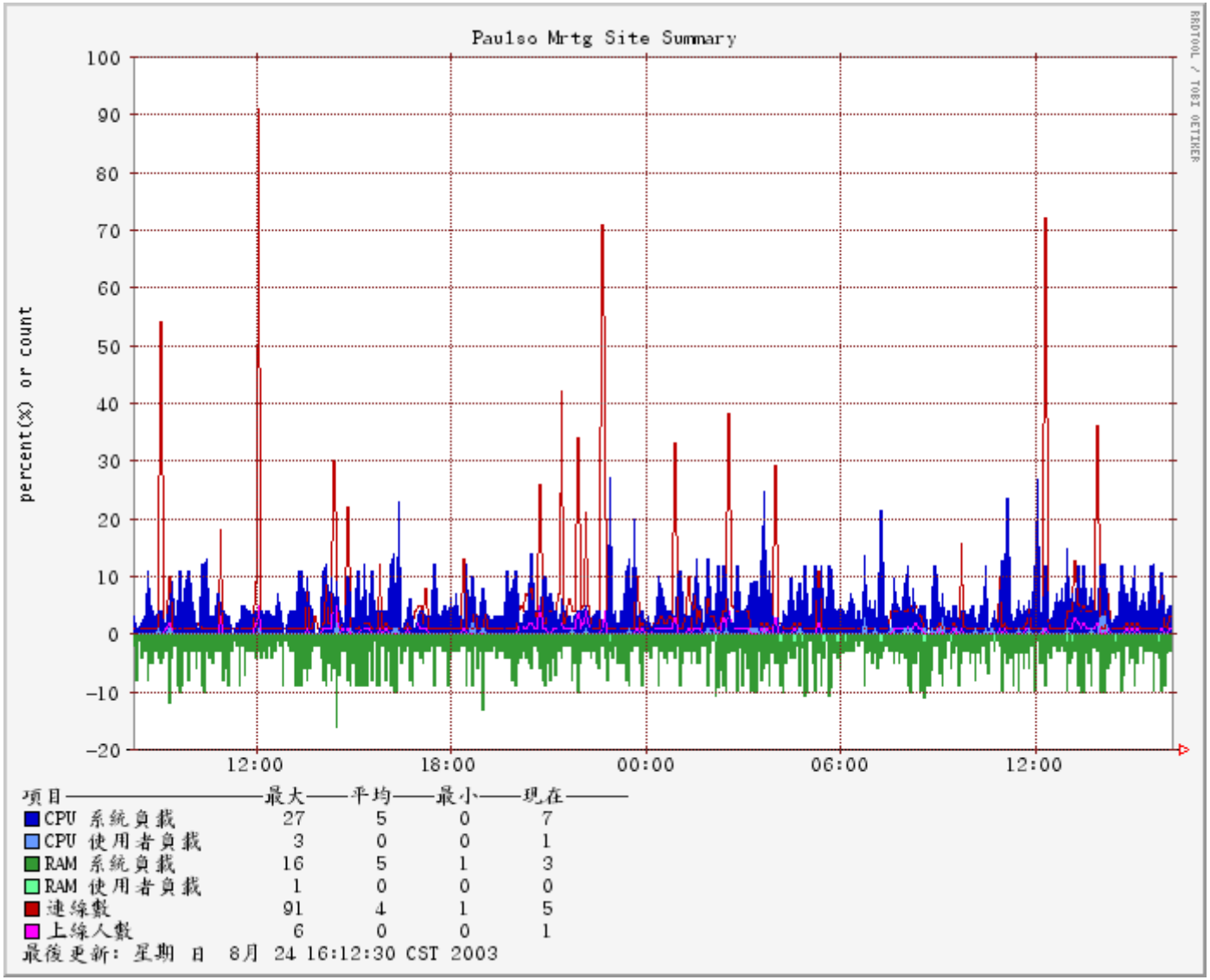


最大 RAM 系統負載; 16.0 % 平均 RAM 系統負載; 4.0 % 目前 RAM 系統負載; 3.0 %
最大 RAM 使用者負載; 1.0 % 平均 RAM 使用者負載; 0.0 % 目前 RAM 使用者負載; 0.0 %



最大 連線數目 : 91.0 % 平均 連線數目 : 3.0 % 目前 連線數目 : 1.0 %
最大 上線人數 : 6.0 % 平均 上線人數 : 0.0 % 目前 上線人數 : 0.0 %

使用 rrdtool 匯整:



綠色為 RAM 之使用率,藍色為 CPU 負載,而紅色系則為連線數,如此,以三合一的方式,更能顯示連線數與系統的關係(這張圖可以看出其沒有太大相關)

下載與安裝

去官網下載 tarball 或 Google 找 RPM 皆可,個人都習慣用 tarball 裝,安裝方法同一般的程式

```
$>./configure --prefix=/usr/local
```

$>make
$>make install

Complier 過程中會有幾個 Warning,但是對整個環境並沒有影響.基本上安裝部份都不會有什麼問題, rrdtool 的 tarball 內即可附了 libgd,zlib 等自用的 lib,不會像 mrtg FAQ 一樣裝好了試一下打 rrdtool ,看會不會出現類似訊息

[root@pc071 study]# rrdtool
RRDtool 1.0.42 Copyright 1997-2001 by Tobias Oetiker <tobi@oetiker.ch>
Usage: rrdtool [options] command command_options
Valid commands: create, update, graph, dump, restore,
last, info, fetch, tune, resize, xport
RRDtool is distributed under the Terms of the GNU General
Public License Version 2. (www.gnu.org/copyleft/gpl.html)
For more information read the RRD manpages

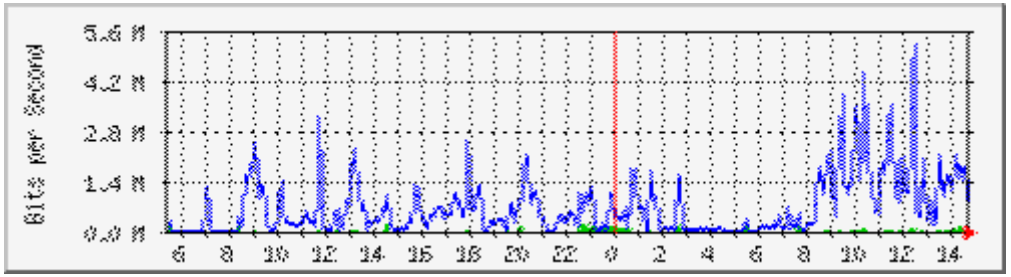
可以看出來 rrdtool 有 11 個 options, 此處介紹 create/update/graph 其餘的部份較屬於 RRD File 的備份/回存及資訊顯示等,與我們主題較無關.另外像 rrdcgi 或 rrd 的 perl module 皆不在我們的介紹範圍內,有興趣之人自可至官方網站查看.

建立 RRD 檔
建立 RRD file 的指令及意義你一定要弄懂,如此圖才能畫的好,不過唯有實作你才能體會的深, 只有看過是不夠的.相信有不少人看過 rrdtool 網站上的說明,個人覺得上面有幾個部份 英文蠻難的(你覺不覺得我就不知了)...
再說明之前我們先了解 mrtg log 的儲存格式...以一般而言 mrtg 大家習慣都是 5m 做 一次,那一天要做 288 次, 你的 mrtg 跑一年不就有 10 萬行的資料了,但實際上mrtg log 是會做一些處理的 實際的狀況是
每五鐘值存 603 筆,再來是
30 分鐘存 603 筆
2 小時值存 603 筆
一天值存 800 筆
註:詳細內容請參考您自身的 mrtg log 檔

mrtg 的優點
個人感覺即是簡單而好用,能符合多數人的需求.
mrtg 的缺點
依此,資料處理的方法會較固定,且log 檔才不會太大,畫 d w m y 圖時才會快,不過你可以想像,你的每五分鐘資料過了 50 個小時後就會變成 30 分鐘平均值....,當您需要一個月前每五分鐘值,以無法再從 mrtg 中找到.
mrtg 另外的限制是無法產生說我要最近3天,最後三個月,近十年來等資料,資料的運算處理上也較少,其僅能產生日週月等較固定的資料.

沒有(邏輯)運算式 (+,-,*,/,%,IF,大於,小於....,) 等....,如果用 rrdtool 這許多問題都可以解決了

運算式
如果你有兩個資料,一個是 Web 連線數,一個是 Web 資料傳輸量,此時你要將這兩個值畫在 mrtg 上,你會發現,連線數的線圖將小得幾乎看不見,因為傳輸量的值太大了,以致於不能於圖上充份表現出來.除非你的 script 先做了適當的運算,如傳輸量以 K 算,再輸出. (註:mrtg 雖可讓你用 K 值當 Y 軸,但是此時連線數的線值看起來就和0一樣了)



eth0 流量:藍線為 tx,綠線為 tr,但此時 tr 看起來幾乎感覺不到他的存在
(不知什麼是 TR/TX 建議您不要看下去了)

邏輯運算式
如果您的圖表上突然有一個很大的值,此時,mrtg 的圖檔原資料因 Y 軸的資料都將因此而壓縮,使用 rrdtool 可以用 GE (大於),LE(小於)…等運算式,讓您將這個值改成其他的值(有沒有意義需視您自身的需要而定)

rrdtool 建檔語法
rrdtool create filename
[--startl-b start time]
[--stepl-s step]
[DS:ds-name:DST:heartbeat:min:max]
[RRA:CF:xff:steps:rows]

看起來語法好像不多(因為不多所以讓很多人看不懂),但其實有點精深,這個建檔的動作其實就像建立 mrtg 的 log 檔,但是 rrd 讓你可以自訂五分鐘資料筆數,平均值為多少時間單位,最大值為多少時間單位,要存幾筆資料,即使用事後發現不足或有問題,依然可使用 tune 來調整.

create 顧名思義,即建檔
filename 隨你取,習慣上會以 .rrd 結尾

--start 這個 filename 的資料記錄起始日期 ,以 1970 年至今的秒數 (預設是現在)
--step 資料的間格時間,習慣上我們會設 300 (秒),您可視自己的需求而定

下面的部份難一點了哦~~要仔細體會了,直接以例子做說明,比如說我們要測 eth0 上的 某些 udp/tcp port 的流量及 總流量(IP 層以上):

DS:telnet:COUNTER:600:0:100000000 \
DS:smtp:COUNTER:600:0:100000000 \
DS:domain:COUNTER:600:0:100000000 \
DS:http:COUNTER:600:0:100000000 \
DS:total:COUNTER:600:0:100000000 \

DS Data Source DS "宣告" 的意思
telnet 是 DSN (name) , 欄位名稱,意即"變數名稱"
COUNTER 是DST(type),習慣上常用 GAUGE(個別值,像CPU loading) 及COUNTER (累計值,像流量資料) 在產生圖檔時, GAUGE 是 100 就畫在 Y 軸 100 上,但如果是 COUNTER , 前一值是 98,則會畫 2
600 是有效期(heartbeat),如果連續如果原來在 12:00 要產生資料而沒有產生,前後 300 秒 (共 600 秒)的平均值會算成 12:00 的值,如果都沒有值,則會成為 "UNKNOWN" (UN,就像 mrtg 沒有資料時,會畫一平線的狀況一樣),UN 在 rrdtool 裏面還是有作用
0:100000000 是說 DSN 的數值有效範圍,如果超出這個值,皆視為 UN,這裏也可以寫成 U:U 代表不限範圍

DS 的部份剛開始一定記不熟,不過用久了就不會有太大問題了,一個好記的方式即 "三文字,三數字".

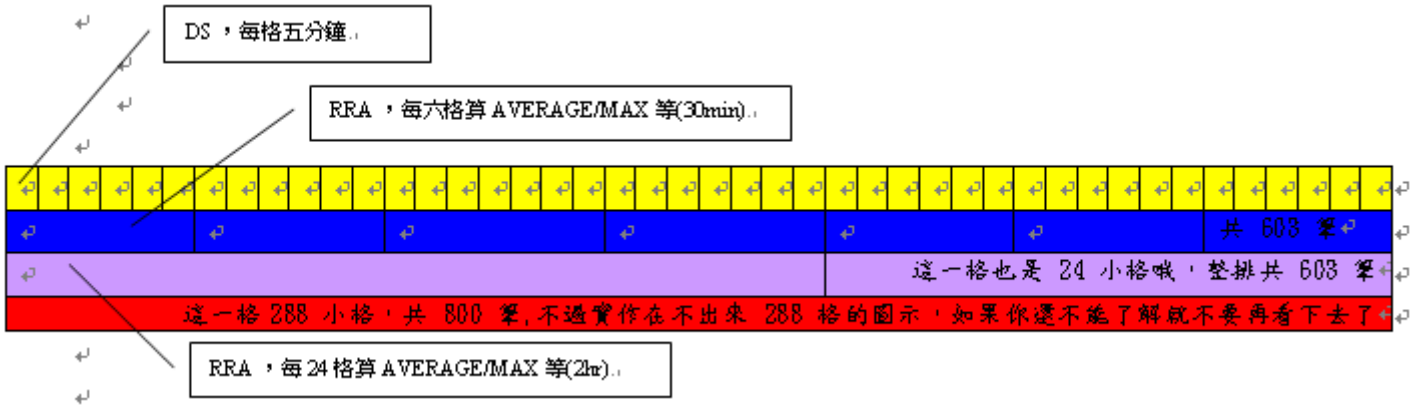
RRA 可能對才數人不容易理解,其實就是什麼資料要存幾筆
RRA:AVERAGE:0.5:1:603 \
RRA:AVERAGE:0.5:6:603 \
RRA:AVERAGE:0.5:24:603 \
RRA:AVERAGE:0.5:288:800 \
RRA:MAX:0.5:1:603 \
RRA:MAX:0.5:6:603 \
RRA:MAX:0.5:24:603 \
RRA:MAX:0.5:288:800

RRA 即 Round Robin Archive,你可以把它看成像 DS,但是這裏主要在處理資料筆數
AVERAGE 在 rrd 稱為 CF (consolidation function),此處我們使用平均數,共有四種類別 : AVERAGE, MIN, MAX, LAST 意即平均值，最大值，最小值，最後一筆
0.5:1:603 因為我們將 step 定為 300 秒是指若原計算時間點為 12:00 的話，記錄時要以 11:57:30~12:02:30 的平均值為主，這個值若在此時間點內只有一筆資料的話，其意即是平均值，所以此一值即表原 telnet/smtp…等共要記錄幾筆(若 mrtg 此值為 603)，603 是指要存 603 筆 (此處故意與 mrtg 同,

以利大家判斷),超過603筆，則最早之一筆將被移出.
0.5:6:603 僅就 6 解釋,取 6 筆資料(每筆為 step 值，在此意即 5 分鐘)為平均值(30 分鐘), 存 603 筆
0.5:24:603 24 即二小時
0.5:288:800 288 即一天

請注意,不是 0.5:1:603 中的1 就是五分鐘,其乃依據你的 --step 值而定,如果 --step 3600 , 那 0.5:6:603 這一行就是六小時合起來的平均值了.
若將 AVERAGE 換成MIN/MAX 的意義則是取該時間點中 (如上例之5min/30min/2hr..)之最大值或最小值，而通常在監測系統時最大值與平均值是較有實際意義的.

RRA 再解釋一下大家會較清楚:



並不會很難~~其實懂了意義就容易了，不過當出我在 K 資料時，到是有不少英文與理解上的問題呢!還得在 google 找許多資料來參考.至少你現在看的是中文,要理解,不能強記.大概的重點我想我都指出來了.

回想一下，下列的指令意義你還記得多少?

```
rrdtool create /root/study/tcpdump.rrd -s 300 -b `date -d "-1 month" +%s` \
DS:telnet:GAUGE:600:0:10000000 \
DS:smtp:GAUGE:600:0:10000000 \
DS:domain:GAUGE:600:0:10000000 \
DS:http:GAUGE:600:0:10000000 \
DS:pop3:GAUGE:600:0:10000000 \
DS:total:GAUGE:600:0:10000000 \
RRA:AVERAGE:0.5:1:603 \
```

RRA:AVERAGE:0.5:6:603 \
RRA:AVERAGE:0.5:24:603 \
RRA:AVERAGE:0.5:288:800 \
RRA:MAX:0.5:1:603 \
RRA:MAX:0.5:6:603 \
RRA:MAX:0.5:24:603 \
RRA:MAX:0.5:288:800

-b 處我讓他建立一個一月前為起始的資料,這裏我提供一個 rrdtool update 的範例檔
記注意,如果你已新增了 12:00 的資料, 12:00 之前的資料你就不能再更新了，rrdtool 會和你說 timestamp 小於最後一筆.

2.新增資料
rrdtool update filename [--template|-t ds-name[:ds-name]...] N|timestamp:value[:value...]
這個很好理解,基本上就是根據 DS來更新資料,如上述之 tcpdump.rrd,若有需要更新時及時
$>rrdtool update tcpdump.rrd 1061811856:114:0:50:1199:0:821073
上面的 1061811856 即時間值,如果就是要現在的時間值,則可以 N 代表,但要轉換成秒值,通常我們都會以
$>timestamp=`date +%s `
來轉現在秒數,如果是某些特定時間,則可以
$>timestamp=`date -d "2003/08/15 12:00" +%s`

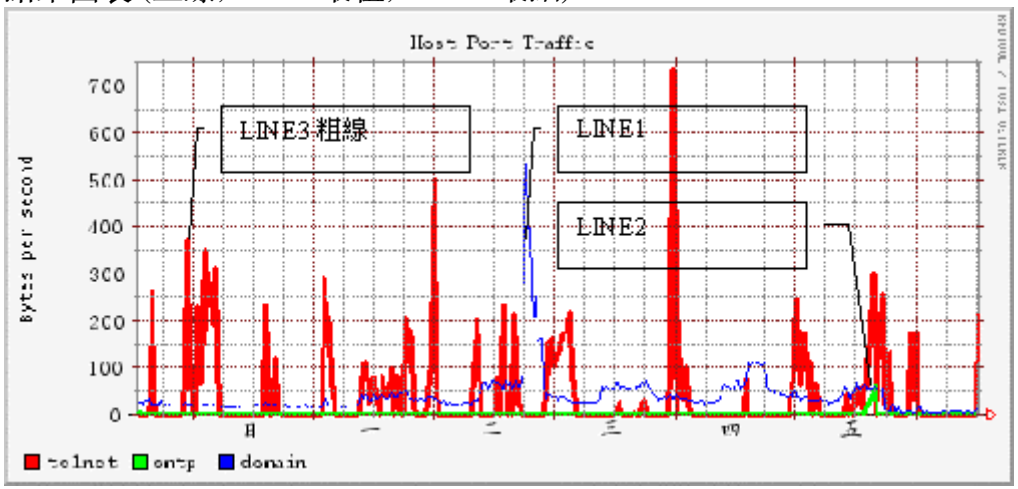
通常這裏你得寫個小程式取數據,或用 snmpget/snmpwalk 抓資料來做 rrdtool update，再用 crontab 根據你在 rrdtool create 時的 step 來決定執行排程的時間點
http://211.72.210.199/tcpdump.txt 這裏提供一個 update 範例檔給大家，其時間範圍為 2003/08/15~2003/08/25,step 為 300s,根據這個檔您自己可適時的建立自己的 rrd file,最好不要抄上面才好(放不進去 rrdfile,請再將 create 指令再看一次,一定是你漏了什麼了).

3. 畫圖
先用簡單範例,引起你的興趣... (看起來好複雜…)

#三線圖 (LINE1 是細線,尚有 LINE2,LINE3 (粗線條) 等)
RRD_FILE=/root/study/tcpdump.rrd
rrdtool graph html/example.png \
--title "Host Port Traffic " \

```
DEF:t1=$RRD_FILE:telnet:AVERAGE \
DEF:t2=$RRD_FILE:smtp:AVERAGE \
DEF:t3=$RRD_FILE:domain:AVERAGE \
LINE3:t1#ff0000:"telnet" \
LINE2:t2#00ff00:"smtp" \
LINE1:t3#0000ff:"domain" \
-h 200 -w 480 -s `date -d "-1 week" +%s` \
-v "Bytes per second"
```

結果圖表 (三線,LINE3 最粗, LINE1 最細)



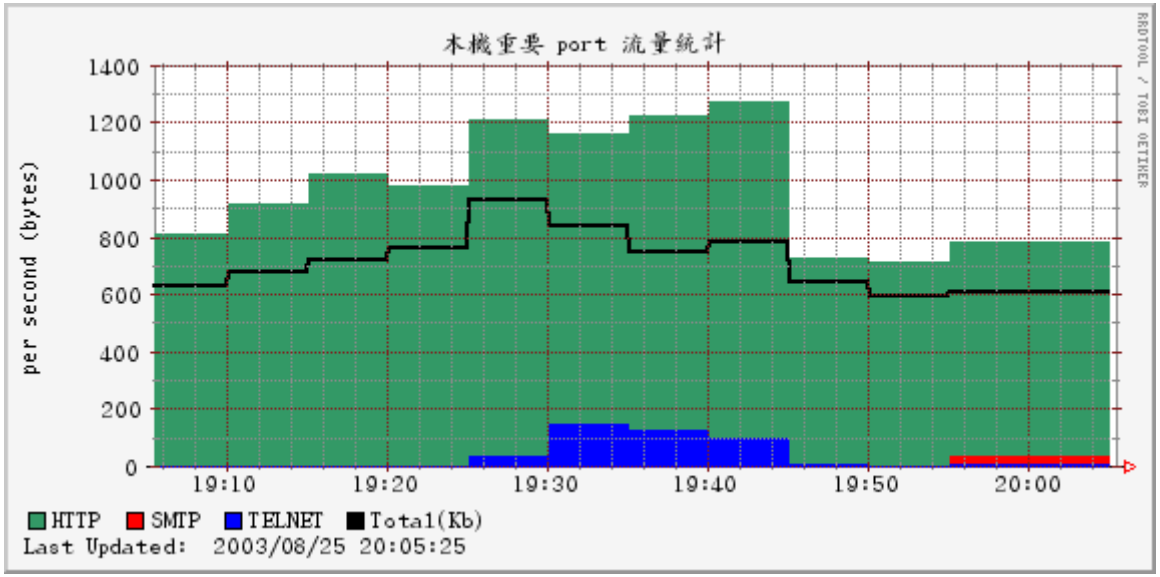
由這張圖可以看出來,以三條線來表示三個 port 時,線形有粗細之分（自己需定義）, 不過此時因每個時間點不同而有可能交叉,可能增加了我們閱覽時的困難.

因為以線來表示較難看出總合情況,所以我們要將每條線疊起來形成一個堆壘的圖,如此就看出來整個機器這幾個 port 的狀況

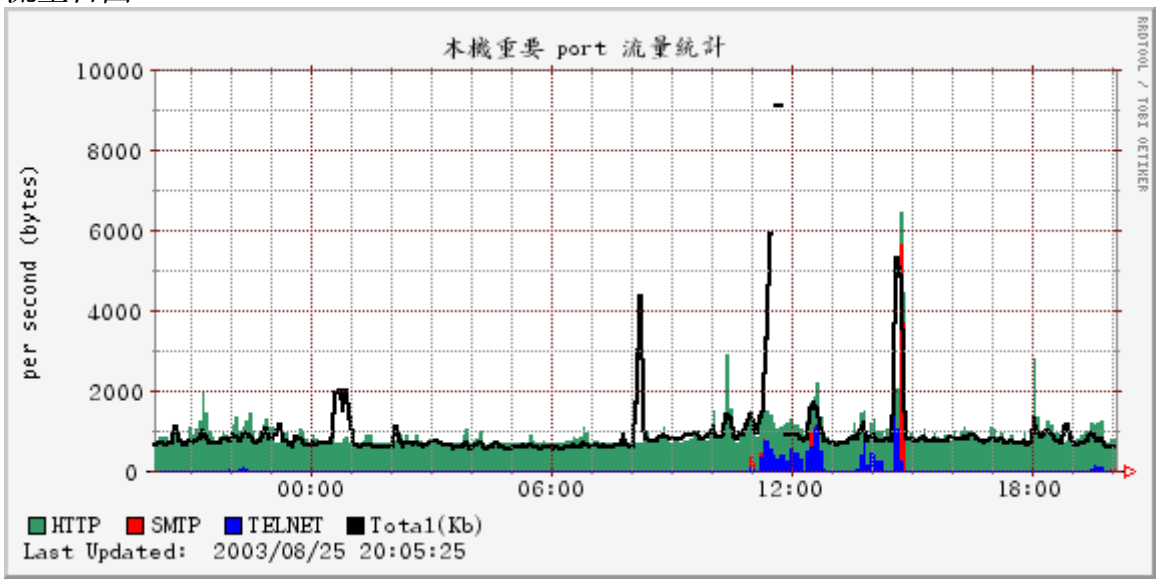
```
#或如下 (畫出時,天,週,的流量圖)
#filename: tcpdump-graph.sh
RRD_PATH="/root/study/tcpdump.rrd"
image_path="/root/study/html"
image_path=/home/httpd/html/enum/study
now=`date "+%Y/%m/%d %H:%M:%S"`
start_time=`date -d "2003/08/12 19:00" +%s`
time="hour day week "
for t in $time
```

```
do
/usr/local/bin/rrdtool graph $image_path/example-$t.png \
--title "本機重要 port 流量統計" \
DEF:t1=$RRD_PATH:telnet:AVERAGE \
DEF:t2=$RRD_PATH:smtp:AVERAGE \
DEF:t3=$RRD_PATH:domain:AVERAGE \
DEF:t4=$RRD_PATH:http:AVERAGE \
DEF:t5=$RRD_PATH:total:AVERAGE \
CDEF:v1=t1,t2,t3,t4,+,+,+ \
CDEF:v2=t1,t2,t3,+,+ \
CDEF:v3=t1,t2,+ \
CDEF:v4=t1 \
CDEF:v5=t5,1024,/ \
AREA:v1#339966:"HTTP" \
AREA:v3#FF0000:"SMTP" \
AREA:v4#0000ff:"TELNET" \
LINE2:v5#000000:"Total(Kb)" \
COMMENT:"\n" \
COMMENT:"Last Updated: $now" \
-v "per second (bytes)" -M -U 10 \
-Y -X b -h 200 -w 480 -s `date -d "-1 $t" +%s` -b 1024
done
```

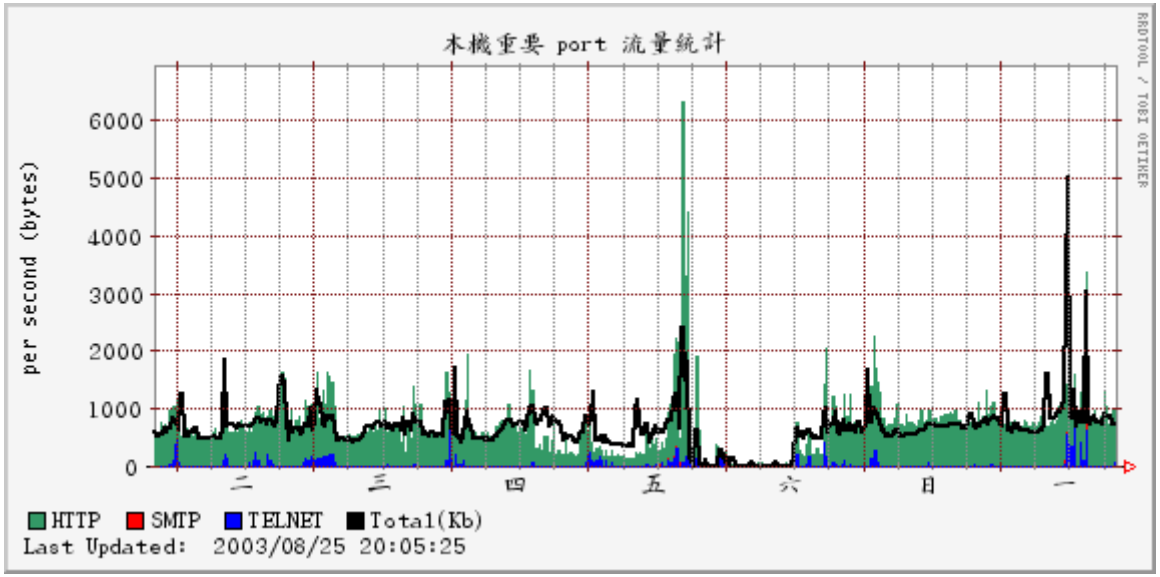
流量小時圖:


本機重要 port 流量統計
per second (bytes)
1400
1200
1000
800
600
400
200
0
19:10
19:20
19:30
19:40
19:50
20:00
HTTP
SMTP
TELNET
Total(Kb)
Last Updated: 2003/08/25 20:05:25

流量日圖


本機重要 port 流量統計
per second (bytes)
10000
8000
6000
4000
2000
0
00:00
06:00
12:00
18:00
HTTP
SMTP
TELNET
Total(Kb)
Last Updated: 2003/08/25 20:05:25

流量週圖



上述三圖我們是以區磈(AREA)來畫出,但是原資料如果我們以AREA來畫將出現互相覆蓋的情況,所以得做一些運作(CDEF 那一段),將適當的加總另外給予另外一變數,而 Total 之值因為過大,如果我們以原值畫於圖上將造成 HTTP/SMTP/TELNET 的圖形無法適當反應,所以將其除以 1024,而以 Kbytes 來表示. Rrdtool 是不是很活呢?不過也因為其較靈活所以你多少得花許多時間自己去體會.
個人覺得剛開始學時不容易掌握到要緊,只有靠練習才能生巧,rrd 畫出來的圖覺得比 mrtg 來得有變化,也更容易 "Customize",但 mrtg 是較好學習的. 若無特殊需要的確不需要另學 rrd.
(註:mrtg 3.x 應會改用 rrd , 之前有看到過這樣文章,我也是先 mrtg->mrtf-rrd->rrd)

rrdtool graph 簡單介紹
你若想知道得很清楚建議您到http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/manual/rrdgraph.html
參考其詳細語法,不必費心找其他的連結,因為我覺得其已經很詳細了.

我僅列出我要介紹的部份供大家參考,你慢慢去體會囉
rrdtool graph image-filename
-s 繪圖資料的起始時間,預設是一天前,可參考上面的 script ,-s `date …` 的應用
-e 繪圖資料的結束時間,預設是現在,亦可使用 date 方式來達到前三天至昨天圖檔
--no-minor 不要副格線
-t 圖檔標題
-v Y 軸說明
-w 資料區的寬度,資料區指的是數據顯示的部份,而非說明或圖例

-h 資料區的高度
-u Y 軸正值高度
-l Y 軸負值高度
DEF 重要的地方,其語法為 DEF:your_var:rrd_filename:DS_name:[AVERAGE|MAX..]
請參考上面的 tcpdump-graph.sh
CDEF 一個虛擬的變數,其值為 DEF 的某些運算,其運算式需寫成後序
EX: a=1+3 寫成 a=1,3 +
http=(smtp+http+telnet)/1024 寫成 http=1024,smtp,http,telnet,+,+,/
不懂？http://people.ee.ethz.ch/~oetiker/webtools/rrdtool/tutorial/ 這裏可以參考

LINE{1|2|3}:vname[#rrggbb[:legend]]
AREA:vname[#rrggbb[:legend]]
STACK:vname[#rrggbb[:legend]]
LINE1:your_var#rgb顏色值:說明,這個 your_var 需存在 DEF 或 CDEF 的宣告中,
AREA 則是畫出資料數值至 0 之間的區磈圖
STACK 則是畫出資料數值至其上的數值,也就是要有資料數值在 STACK 數值之上
請注意,如果使用 AREA/STACK 時需特別注意圖蓋圖的問題,一定要先畫大的值,再畫小的值,才會有層次的效果,不然,最大的數據若最後畫,是直接壓過去哦
COMMENT 說明欄字,如 COMMENT:"Last Updated" 將在圖上產生該文字,可以用 \n 等換行符號
GPRINT GPRINT:vname:CF:format vname 即DEF 中的 your_var,而 CF 看你要輸出的文字是AVERAGE/MAX/MIN/LAST 等數值,format 如同 printf 中的格式,
EX:
GPRINT:telnet:AVERAGE:"%10.0lf \n"
意即要輸出這段時間中 (-s ~ -e 中,telnet的平均值,%10.0lf 則是為了好算位置)
如果你不懂 printf, man 一下會比我解釋一大堆來得快.

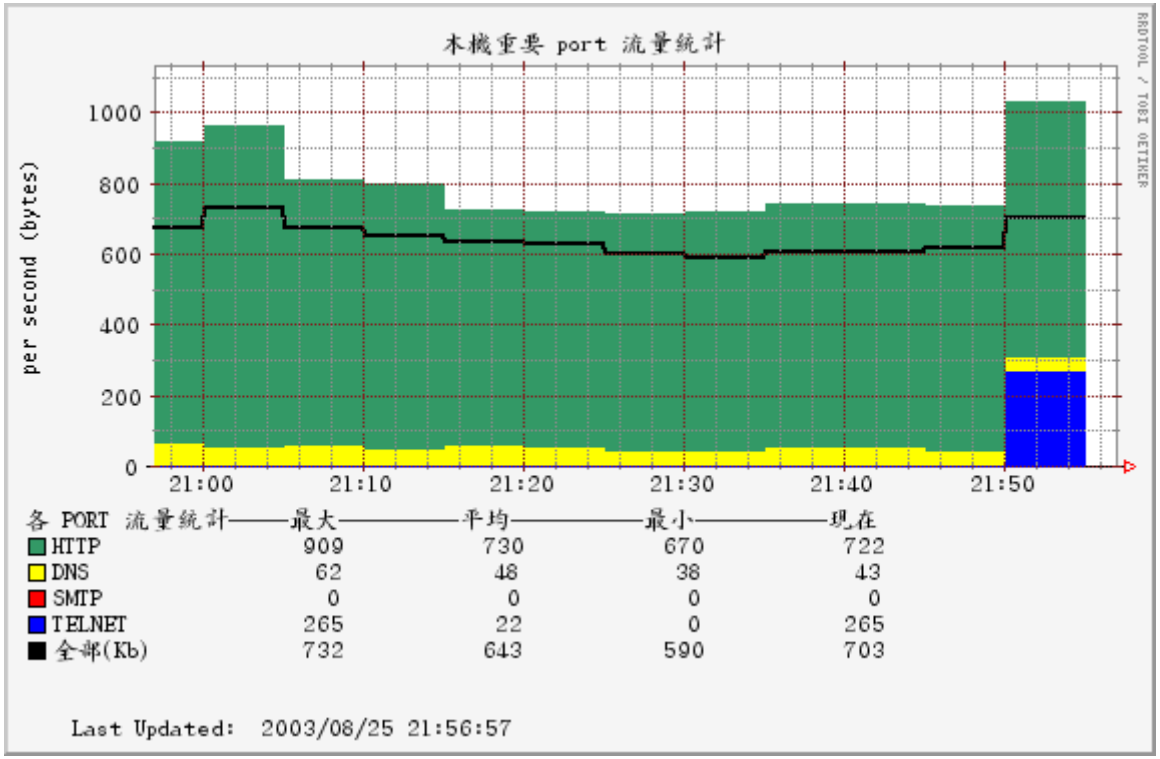
其他沒有介紹的參數就有待你自己去發掘了,還有很多沒有講到的,不過我都講完就沒有意思了.
依據這裏的介紹,我們再將流量圖美化些,讓它可以產生說明文字,更有助於我們的判讀,並將如何 update 資料也一併加入:

代碼:

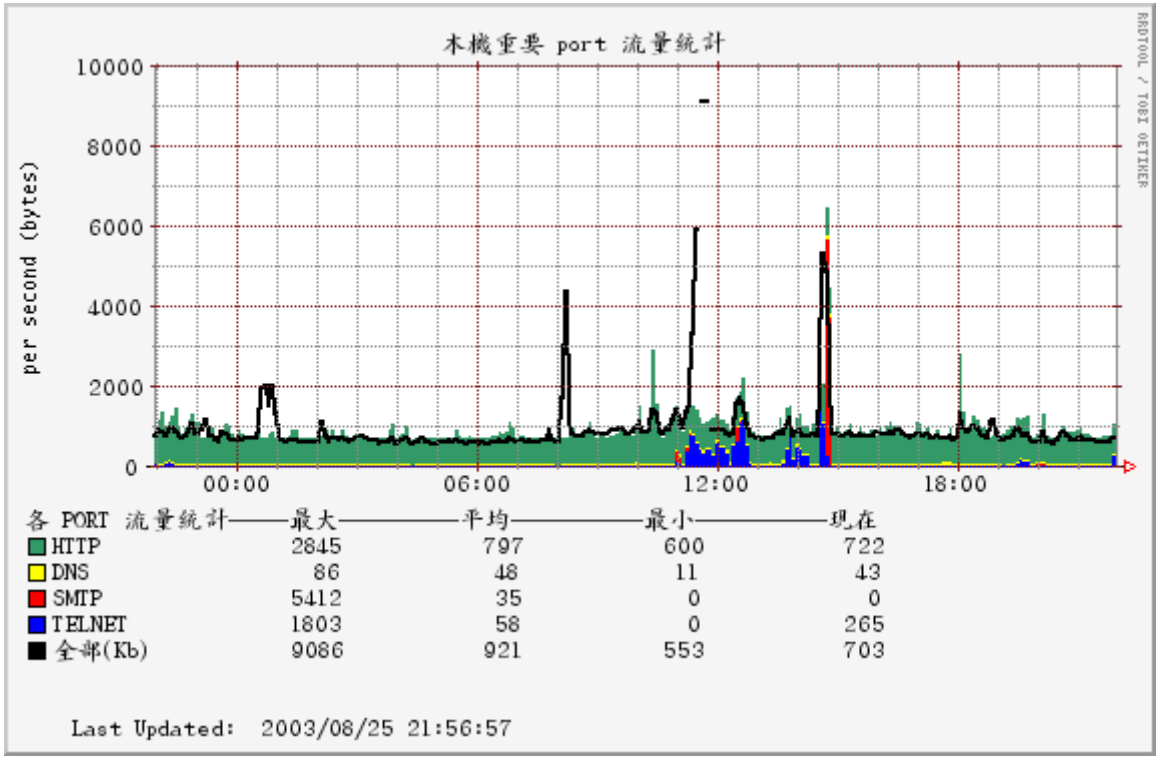
```
#tcpdump.sh
RRD_PATH="/root/study/tcpdump.rrd"
image_path="/root/study/html"
```

```
sec=300
killall tcpdump
mv ip.packet ip.packet.1
tcpdump -w ip.packet  tcp or udp or icmp &
scan_port="23 25 53 80 110"
rrd_data=""
for sport in $scan_port
do
        port=`tcpdump -r ip.packet.1 port $sport -v | sed -e 's/.*, len \(.*\))/
\1/g' |  tr '\n' '+'`
        port=`echo ${port}0| bc`
        port=`expr $port / $sec`
        rrd_data="$rrd_data$port:"
done
total=`tcpdump -r ip.packet.1 -v  |grep -v 'config'| sed -e 's/.*, len \(.*\))/\
1/g'  |  tr '\n' '+'`
total=` echo ${total}0 | bc`
now=`date +%s`
echo "rrdtool update tcpdump.rrd $now:$rrd_data$total" >>tcpdump.cmd
rrdtool update tcpdump.rrd $now:$rrd_data$total

image_path=/home/httpd/html/enum/study
now=`date "+%Y/%m/%d %H:%M:%S"`
start_time=`date -d "2003/08/12 19:00" +%s`
time="hour day week month year"
for t in $time
do
/usr/local/bin/rrdtool graph $image_path/example-$t.png \
--title "本機重要 port 流量"   \
DEF:t1=$RRD_PATH:telnet:AVERAGE \
DEF:t2=$RRD_PATH:smtp:AVERAGE \
DEF:t3=$RRD_PATH:domain:AVERAGE \
DEF:t4=$RRD_PATH:http:AVERAGE \
DEF:t5=$RRD_PATH:total:AVERAGE \
CDEF:v1=t1,t2,t3,t4,+,+,+ \
CDEF:v2=t1,t2,t3,+,+ \
CDEF:v3=t1,t2,+ \
CDEF:v4=t1 \
CDEF:v5=t5,1024,/ \
```

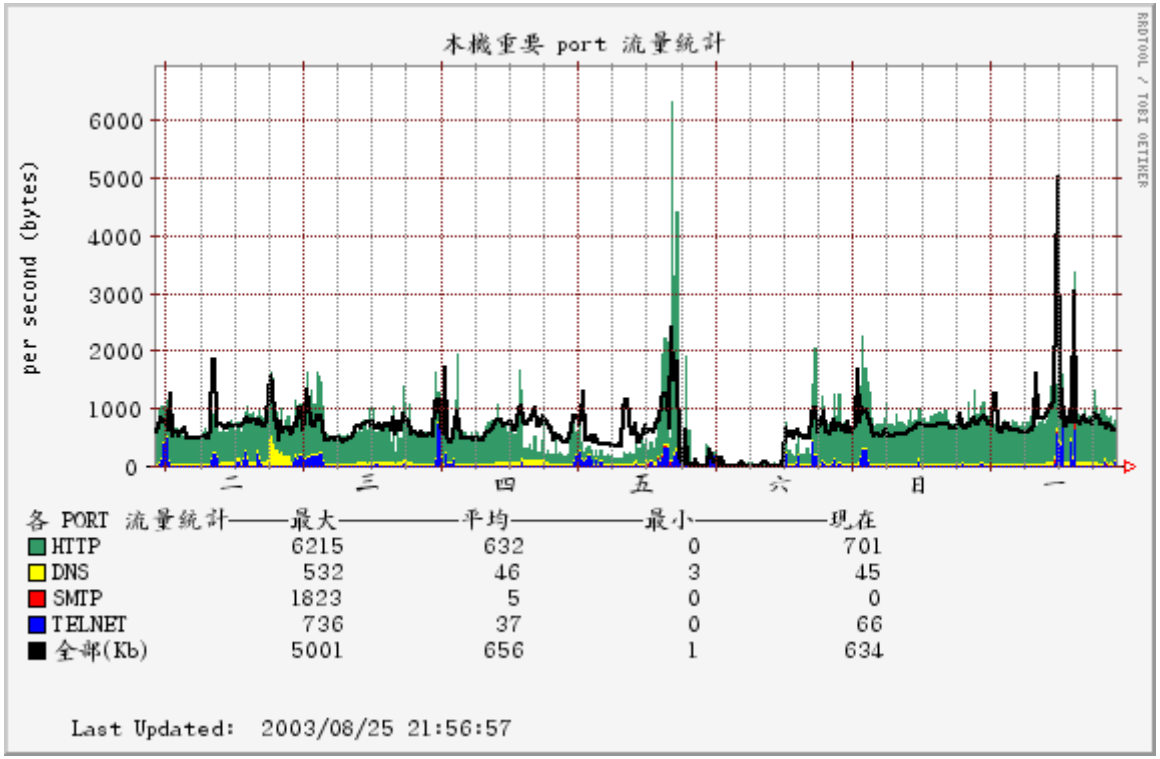
```
        COMMENT:"各 PORT 流量統計---最大------平均-------最小-------?#123;在\n"    \
        AREA:v1#339966:"HTTP" \
        GPRINT:t4:MAX:"      %12.0lf "  \
        GPRINT:t4:AVERAGE:"%12.0lf "          \
        GPRINT:t4:MIN:"%12.0lf "              \
        GPRINT:t4:LAST:"%12.0lf \n"           \
        AREA:v2#ffff00:"DNS"        \
        GPRINT:t3:MAX:"      %12.0lf "  \
        GPRINT:t3:AVERAGE:"%12.0lf "          \
        GPRINT:t3:MIN:"%12.0lf "              \
        GPRINT:t3:LAST:"%12.0lf \n"           \
        AREA:v3#FF0000:"SMTP"       \
        GPRINT:t2:MAX:"      %12.0lf "  \
        GPRINT:t2:AVERAGE:"%12.0lf "          \
        GPRINT:t2:MIN:"%12.0lf "              \
        GPRINT:t2:LAST:"%12.0lf \n"           \
        AREA:v4#0000ff:"TELNET"       \
        GPRINT:t1:MAX:"    %12.0lf "  \
        GPRINT:t1:AVERAGE:"%12.0lf "          \
        GPRINT:t1:MIN:"%12.0lf "              \
        GPRINT:t1:LAST:"%12.0lf \n"           \
        LINE2:v5#000000:"全部(Kb)"       \
        GPRINT:v5:MAX:"  %12.0lf "  \
        GPRINT:v5:AVERAGE:"%12.0lf "          \
        GPRINT:v5:MIN:"%12.0lf "              \
        GPRINT:v5:LAST:"%12.0lf \n"           \
        COMMENT:"\n" \
        COMMENT:"\n" \
        COMMENT:"    Last Updated:  $now" \
        -v "per second (bytes)" -M -U 10 \
        -Y -X b -h 200 -w 480 -s `date -d "-1 $t" +%s`
        done


hour:
```


本機重要 port 流量統計
per second (bytes)
1000
800
600
400
200
0
21:00
21:10
21:20
21:30
21:40
21:50
各 PORT 流量統計——最大——平均——最小——現在
HTTP 909 730 670 722
DNS 62 48 38 43
SMTP 0 0 0 0
TELNET 265 22 0 265
全部(Kb) 732 643 590 703
Last Updated: 2003/08/25 21:56:57

day:


本機重要 port 流量統計
10000
8000
6000
4000
2000
0
per second (bytes)
00:00
06:00
12:00
18:00
各 PORT 流量統計——最大——平均——最小——現在
HTTP 2845 797 600 722
DNS 86 48 11 43
SMTP 5412 35 0 0
TELNET 1803 58 0 265
全部(Kb) 9086 921 553 703
Last Updated: 2003/08/25 21:56:57

week:



加上數值的顯示是不是更清楚了呢 ?不過請注意 rrdtool 是不支援中文的,請您用英文來表示你的資料即可.
嗯~如果你覺得很複雜那是正常的,但是其實仔細觀察你可以發現,其實很多地方都很類似,我的感覺是在初學時才覺得複雜,等做過了一兩個成功的例子後就覺得很簡單了,學任何東西不也都是如此嗎 ?
嗯~tcpdump 抓封包那一段不在我的介紹範圍內,有興趣的人自可 man tcpdump 自己學習一番,或是有人要貢獻一下也是很好的.

提供大家我的幾個範例當參考,學習的過程中範例其實很重要的,但 rrdtool 的範例其實不多,不然就都是英文的

Question1: 我的簽名檔做法

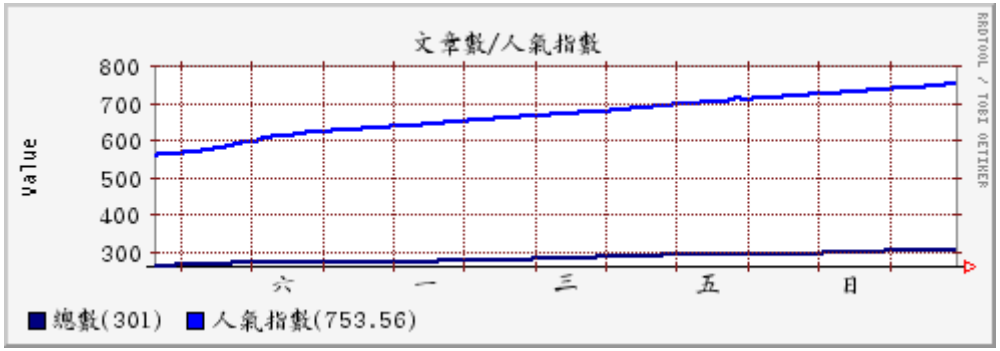
代碼:

```
#建 rrd file
rrdtool create /root/study/study-area.rrd -s 300 \
DS:post:GAUGE:600:0:10000 \
DS:welcome:GAUGE:600:0:10000 \
DS:post_c:COUNTER:600:0:10000 \
DS:welcome_c:COUNTER:600:0:10000 \
```

```
RRA:AVERAGE:0.5:1:9600  \
RRA:AVERAGE:0.5:6:4800          \
RRA:AVERAGE:0.5:24:1200         \
RRA:AVERAGE:0.5:288:600

#我的簽名檔
image_path=/root/study/html
rrd_file=/root/study/study-area.rrd
RRD_PATH=$rrd_file

#這裏我用無限迴圈做,非 crontab
sec=300
while [ 1 ]
do
# 取得自己所 post 的某一篇文章
wget http://phorum.study-area.org/viewtopic.php?t=18410 -O study-area.html
#抓文章中的文章總數及人氣指數
rrd_data=`cat study-area.html | grep 'abelyang' | tail -1 | sed -e 's/.*文章: \(
.*\)<br \/><br.*#FF6633">\(.*\)<\/font><\/b>.*/\1:\2/g`
post=`echo $rrd_data| cut -f 1 -d':'`
welcome=`echo $rrd_data| cut -f 2 -d':'`
now=`date +%s`
echo rrdtool update study-area.rrd $now:$rrd_data:$rrd_data >>study-area_rrd.cmd
rrdtool update $rrd_file $now:$rrd_data:$rrd_data

################################################

/usr/local/bin/rrdtool graph $image_path/study-area.png \
--title "文章數/人氣指數" \
DEF:v1=$RRD_PATH:post:AVERAGE \
DEF:v2=$RRD_PATH:welcome:AVERAGE \
DEF:v3=$RRD_PATH:post_c:AVERAGE \
DEF:v4=$RRD_PATH:welcome_c:AVERAGE \
LINE2:v1#000080:"總數($post)" \
LINE2:v2#0000FF:"人氣指數($welcome)" \
-v "Value" -M -U 10 -s `date -d "2003/08/14 15:00" +%s` \
-Y -X b --no-minor
```

```
#將圖用指令縮小成 250 pixel
convert -scale 250 $image_path/study-area.png $image_path/study-area1.png
sleep $sec
done
```

結果:



Question: 如何抓 CISCO Switch 每個 port 流量呢 ?

代碼:

```
#建 rrdfile
rrdtool create C2900.rrd -s 300 \
DS:ifInOctets1:COUNTER:600:U:U \
DS:ifInOctets2:COUNTER:600:U:U \
DS:ifInOctets3:COUNTER:600:U:U \
DS:ifInOctets4:COUNTER:600:U:U \
DS:ifInOctets5:COUNTER:600:U:U \
DS:ifInOctets6:COUNTER:600:U:U \
DS:ifInOctets7:COUNTER:600:U:U \
DS:ifInOctets8:COUNTER:600:U:U \
DS:ifInOctets9:COUNTER:600:U:U \
DS:ifInOctets10:COUNTER:600:U:U \
DS:ifInOctets11:COUNTER:600:U:U \
DS:ifInOctets12:COUNTER:600:U:U \
DS:ifInOctets13:COUNTER:600:U:U \
DS:ifInOctets14:COUNTER:600:U:U \
DS:ifInOctets15:COUNTER:600:U:U \
```

```
DS:ifInOctets16:COUNTER:600:U:U \
DS:ifInOctets17:COUNTER:600:U:U \
DS:ifInOctets18:COUNTER:600:U:U \
DS:ifInOctets19:COUNTER:600:U:U \
DS:ifInOctets20:COUNTER:600:U:U \
DS:ifInOctets21:COUNTER:600:U:U \
DS:ifInOctets22:COUNTER:600:U:U \
DS:ifInOctets23:COUNTER:600:U:U \
DS:ifInOctets24:COUNTER:600:U:U \
DS:ifInOctets25:COUNTER:600:U:U \
DS:ifInOctets26:COUNTER:600:U:U \
DS:ifOutOctets1:COUNTER:600:U:U \
DS:ifOutOctets2:COUNTER:600:U:U \
DS:ifOutOctets3:COUNTER:600:U:U \
DS:ifOutOctets4:COUNTER:600:U:U \
DS:ifOutOctets5:COUNTER:600:U:U \
DS:ifOutOctets6:COUNTER:600:U:U \
DS:ifOutOctets7:COUNTER:600:U:U \
DS:ifOutOctets8:COUNTER:600:U:U \
DS:ifOutOctets9:COUNTER:600:U:U \
DS:ifOutOctets10:COUNTER:600:U:U \
DS:ifOutOctets11:COUNTER:600:U:U \
DS:ifOutOctets12:COUNTER:600:U:U \
DS:ifOutOctets13:COUNTER:600:U:U \
DS:ifOutOctets14:COUNTER:600:U:U \
DS:ifOutOctets15:COUNTER:600:U:U \
DS:ifOutOctets16:COUNTER:600:U:U \
DS:ifOutOctets17:COUNTER:600:U:U \
DS:ifOutOctets18:COUNTER:600:U:U \
DS:ifOutOctets19:COUNTER:600:U:U \
DS:ifOutOctets20:COUNTER:600:U:U \
DS:ifOutOctets21:COUNTER:600:U:U \
DS:ifOutOctets22:COUNTER:600:U:U \
DS:ifOutOctets23:COUNTER:600:U:U \
DS:ifOutOctets24:COUNTER:600:U:U \
DS:ifOutOctets25:COUNTER:600:U:U \
DS:ifOutOctets26:COUNTER:600:U:U \
DS:gabege:COUNTER:600:U:U \
RRA:AVERAGE:0.5:1:4800  \
```

```
RRA:AVERAGE:0.5:6:2400 \
RRA:AVERAGE:0.5:24:1200 \
RRA:AVERAGE:0.5:288:600
```

看不懂請回頭再看 create 的文字解釋,最後多一個 gabege 欄位是為了好處理用, 且這裏我沒有用到 MAX 的 RRA,你可視自己的狀況.這裏我是用 Cisco2950-24,先在 Switch 上開 snmp-server
並做好 Access Control

代碼:

```
Switch(config)#snmp-server community Your_community_string RO Access-List-ID
```

抓資料,建議先看這一篇,了解原理
http://phorum.study-area.org/viewtopic.php?t=18410

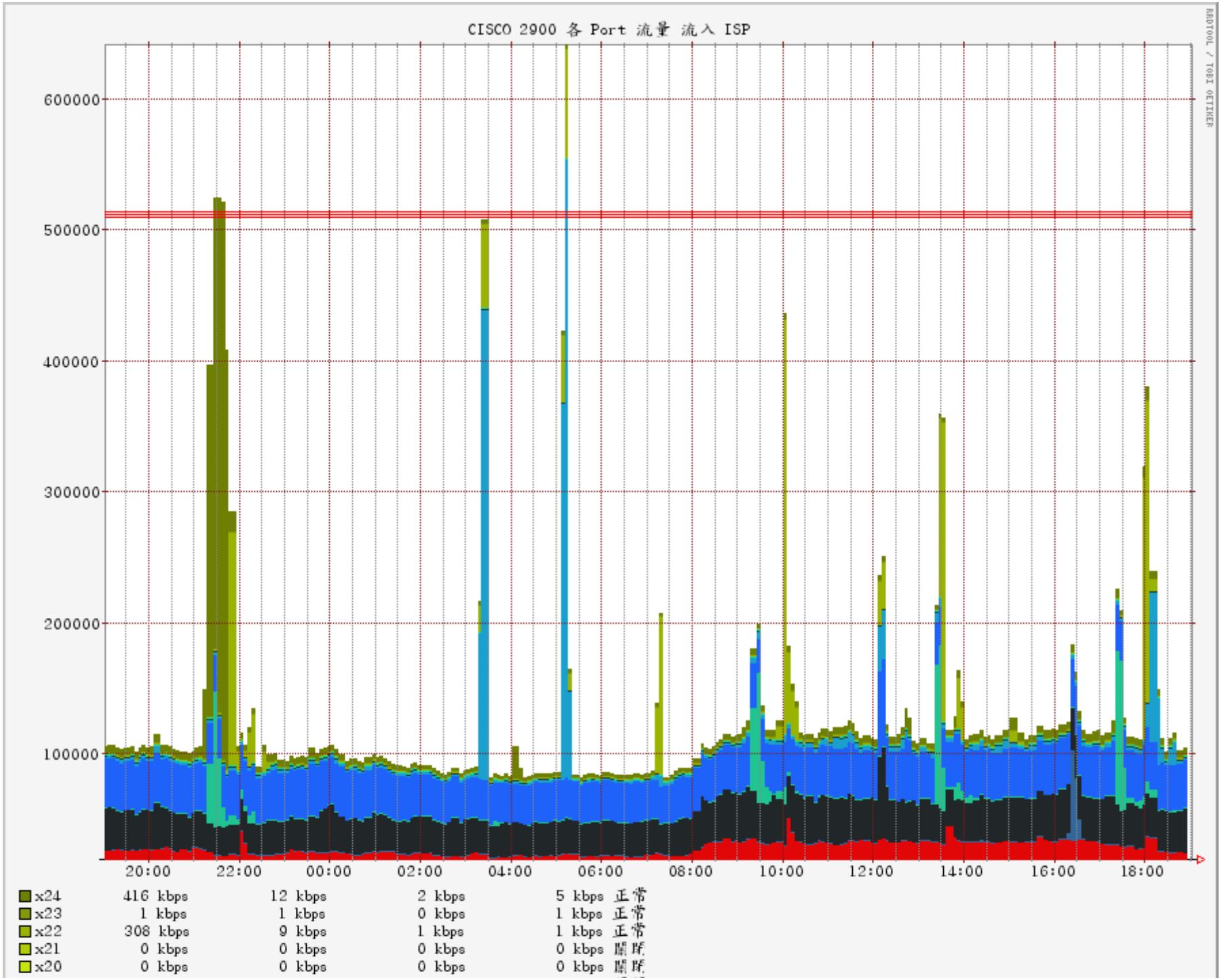
代碼:

```
#取得 24 Port (In/Out) 所有流量記錄
ITEM="ifInOctets ifOutOctets"
export MIBS=CISCO-C2900-MIB
rrd_data=""
for oid in $ITEM
do
#取得 Switch 每一個 Port 的 In/Out 流量
rrd_data=`snmpwalk -v 2c Your_Switch_IP Your_community_string $oid | sed -e 's/.*:\(.*\)$/\1/' | tr '\n' ':'`$rrd_data
done
now=`date +%s`
echo "rrdtool update seednet_C2900.rrd $sec:${rrd_data}0" >> C2900_rrd.cmd
rrdtool update C2900.rrd $now:${rrd_data}0
```

輸入資料都是很容易處理的.畫圖...嗯~~這是有點複雜...你可以先以上面 tcpdump 為例單獨先畫出每個 port 流量, 慢慢練習加一個 port,並了解問題所

在,如此才能體會的到哦!別想一步登天,如此只會徒浪費時間在學習上而以!

下圖為各 port 的堆疊圖 (正常關閉是繪圖前以 snmp 取得 port status)


CISCO 2900 各 Port 流量 流入 ISP
600000
500000
400000
300000
200000
100000
20:00
22:00
00:00
02:00
04:00
06:00
08:00
10:00
12:00
14:00
16:00
18:00
x24 416 kbps 12 kbps 2 kbps 5 kbps 正常
x23 1 kbps 1 kbps 0 kbps 1 kbps 正常
x22 308 kbps 9 kbps 1 kbps 1 kbps 正常
x21 0 kbps 0 kbps 0 kbps 0 kbps 關閉
x20 0 kbps 0 kbps 0 kbps 0 kbps 關閉

x19	0 kbps	0 kbps	0 kbps	0 kbps 關閉
x18	0 kbps	0 kbps	0 kbps	0 kbps 關閉
x17	0 kbps	0 kbps	0 kbps	0 kbps 關閉
x16	0 kbps	0 kbps	0 kbps	0 kbps 關閉
x15	0 kbps	0 kbps	0 kbps	0 kbps 關閉
x14	0 kbps	0 kbps	0 kbps	0 kbps 關閉
x13	0 kbps	0 kbps	0 kbps	0 kbps 關閉
x12	0 kbps	0 kbps	0 kbps	0 kbps 關閉
x11	1 kbps	1 kbps	0 kbps	1 kbps 正常
x10	1 kbps	1 kbps	0 kbps	1 kbps 正常
x09	461 kbps	9 kbps	1 kbps	1 kbps 正常
x08	0 kbps	0 kbps	0 kbps	0 kbps 關閉
x07	65 kbps	34 kbps	25 kbps	35 kbps 正常
x06	122 kbps	5 kbps	1 kbps	1 kbps 正常
x05	68 kbps	29 kbps	20 kbps	33 kbps 正常
x04	67 kbps	1 kbps	0 kbps	1 kbps 正常
x03	49 kbps	27 kbps	19 kbps	23 kbps 正常

Last Updated:五 8月 22 19:01:33 CST 2003
x01 x02 for IPv4/IPv6 Router

---

abelyang 在 星期三 八月 27, 2003 12:57 am 作了第 6 次修改

原文出處：http://phorum.study-area.org/viewtopic.php?t=18496