Netflow 网络流量分析手册

作　者：聂晓亮（毛蛋哥）

目录

一、 作者简介

本书作者聂晓亮，网名毛蛋哥。2004 年毕业于北京联合大学信息工程学院，热爱网络相关知识及摄影，机缘巧合参加了 Cisco 认证培训，并获得了一些成绩。本书写于 2008 年 10 月，作者目前状态工作较为舒适，故有空闲时间完成此书。

聂晓亮（毛蛋哥）拥有自己的 Blog 及 Wiki 空间，其中记录了作者的工作、生活、学习。作者希望通过此书以及 Blog、Wiki 同全世界的网络爱好者分享其知识与快乐。

聂晓亮（毛蛋哥）的 Blog：http://nio.name

聂晓亮（毛蛋哥）的 Wiki：http://wiki.nio.name

欢迎交流：pharaohnie@gmail.com

二、 为什么会有这本书

在工作的几年当中，经常有朋友和一些网友问我一些关于流量分析的问题，诸如：

- 我们局域网怎么这么慢，是不是有人在下 BT？
- 192.168.0.1 也没人用，怎么网卡疯狂闪烁，它在做什么？
- 老板让我查查服务器为什么总是那么大流量，可我不知道从何下手。
- 公司出口带宽不够了，但一时有没那么多带宽预算，我在考虑是不是要关掉一些和公司业务无关的协议，但不知道应该关哪些协议。

诸如这类问题还有很多，对于专业的网络人员当然不是什么难题，但对于一般非 IT 类公司的网管人员也许就是一个不可能完成的任务。因此，才有了这本书，也就是说，这本书的目的就是帮助一般的网管人员了解他们所管理的网络运行情况，排查网速慢、个别主机大量占用带宽的问题。

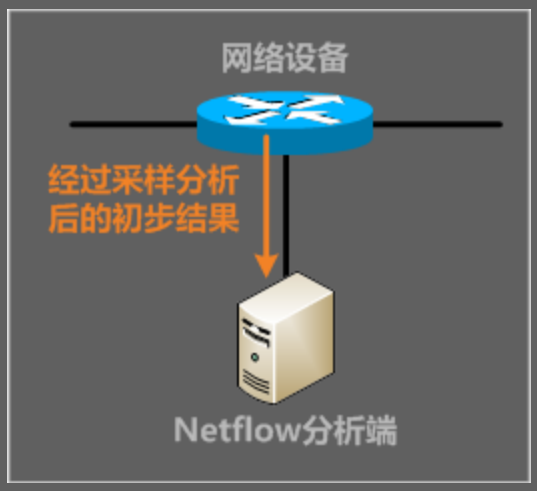
三、 流量分析原理

(一) 原始流量分析方式

原始流量分析是通过复制网络流量至分析端，然后对其进行分析。采用此类分析方式的好处是完全获取了网络的所有流量，可以对其进行深度分析，甚至是用户使用的搜索关键字；弊端是由于需要分析的流量较大，分析端负载会非常高，并不适合进行需要保存历史数据的长期分析。

(二) Netflow 分析方式

Netflow 是 Cisco 公司开发出的一套协议，用于专门解决原始流量方式所产生的问题。当在网络设备或其接口上开启 Netflow 功能后，网络设备会对需要进行分析的流量进行采样分析，并把采样分析的结果发送至分析段进行流量分析，当然这些采样分析的结果要比原始数据小的多的多。其中网络设备采样分析的结果数据会包含源地址、目的地址、源端口、目的端口、数据流的大小、数据流经过的接口、数据流的到达时间、数据流的送出时间等参数。



使用 Netflow 分析方式的好处显而易见，分析端得到的已经不再是原始数据，而是一个初步分析结果，只要对这些初步结果进行二次分析即可获得更多的数据。由于网络设备发送过来的初步分析结果远小于原始数据，因此分析端可以充分利用 CPU 做更多的历史分析，也就解决了原始数据分析方式所导致的无法分析较长时间数据的问题。当然，在 Netflow 分析方式中，由于分析端得到的不是原始数据，自然也无法获得像用户搜索关键字这样的详细信息。

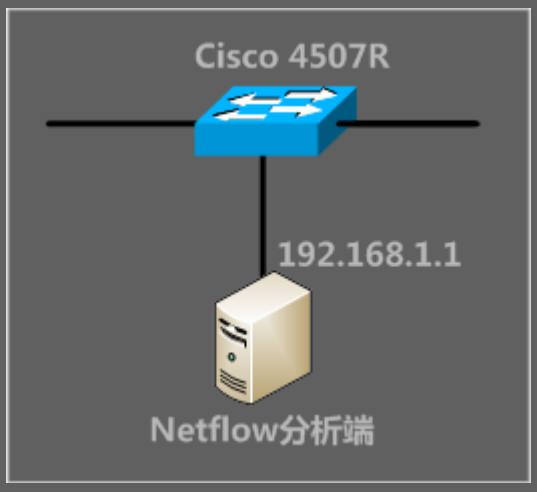
两种分析方式各有特点，建议在进行长期数据分析或进行流量统计分析时使用 Netlfow 分析方式。在需要对网络协议进行分析时，如 Skype 无法连接服务器这样的情况时使用原始数据分析方式。

四、 流量采样

(一) 在网络设备上开启 Netflow 功能

这里只列出如何在 Cisco 设备上开启 Netflow 功能。一般情况下，Cisco 路由器都能支持 Netflow 功能，而交换机只有一些高端系列能够支持 Netflow 功能。

下面以一台 Cisco 4507R 交换机配置为例说明如何在 Cisco 交换机上开启 Netflow 功能。



其中需要 Cisco 4507R 和 Netflow 分析端可以相互 Ping 通，才能确保 Netflow 数据可以顺利的送达分析端进行分析。4507R 上的 Netflow 基本配置如下：

```
ip flow ingress infer-fields
ip flow ingress layer2-switched
ip flow-export destination 192.168.1.1 9995
```

192.168.1.1 为 Netflow 分析端的 IP 地址，9995 表示 Cisco 4507R 向 Netflow 分析端的 UDP 9995 发送 Netflow 数据。

验证一下 Cisco 4507R 是否已经开始发送 Netflow 数据：

```
Cisco-4507R#show ip flow export
Flow export v5 is enabled for main cache
  Exporting flows to 192.168.1.1 (9995)
  Exporting using source interface Loopback0
  Version 5 flow records
  40 flows exported in 3 udp datagrams
  0 flows failed due to lack of export packet
  0 export packets were sent up to process level
```

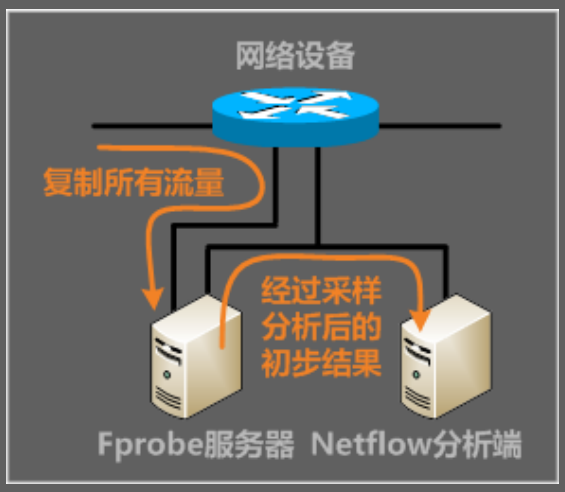
多进行几次验证，如果看到橘黄色字体的部分保持持续增长，则表明已经开始发送 Netflow 数据了。

(二) 网络设备不支持 Netflow

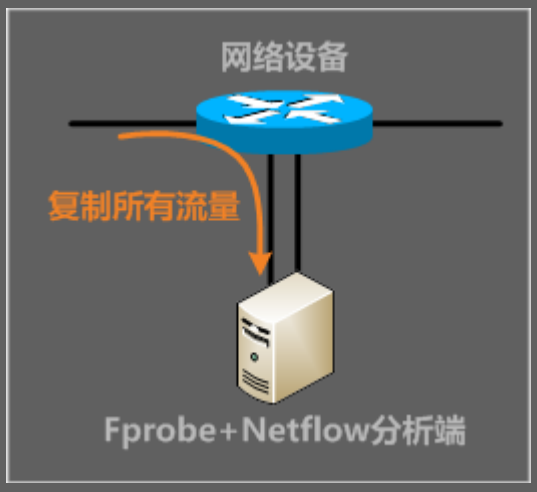
在一些非 IT 行业的中小企业以及一些中小学校中，网络环境相对简单，网络设备不一定支持 Netflow 功能。对于这样的环境也有相应的解决方法-Fprobe。

1. 部署方式

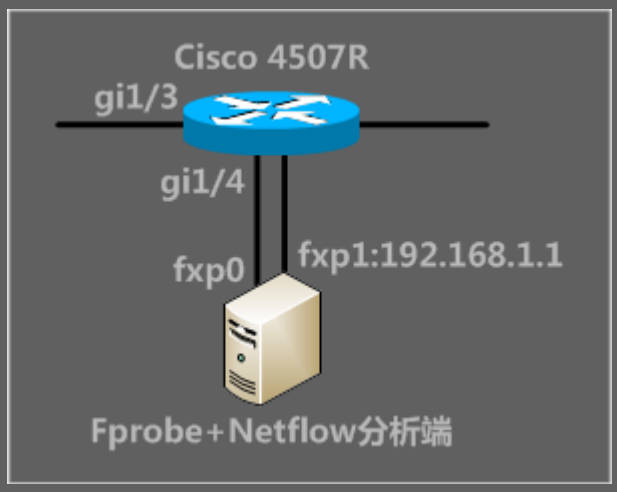
Fprobe 是一款在 FreeBSD（FreeBSD 的安装方法见五-(二)）下运行的软件，它可以将其接口收到的数据转化为 Netflow 数据，并发送至 Netflow 分析端。我们可以通过部署这样一台服务器，并将网络流量镜像至此服务器来实现对网络流量进行 Netflow 分析。其部署方式如下：



如果经费不足，可以将 Fprobe 和 Netflow 部署在同一台服务器中，如下图：



Fprobe 和 Netflow 部署在同一台服务器的方式较为普遍，因此我们将按下图举例 Fprobe 的安装及配置网络设备镜像流量，此处所使用的网络设备为 Cisco 4507R。



Cisco 4507R 与服务器使用 2 跟线缆连接，左边的线缆用于将进入 4507R 的 gi1/3 的流量镜像至服务器的 fxp0，由于 fxp0 接口用于接收镜像流量，因此 fxp0 可以不分配 IP 地址。右边的线缆用于保持服务器的网络连通性，这样可以使网络管理员从任何位置都可以用 192.168.1.1 访问服务器并进行流量分析。

2. 安装 Fprobe

```
fb# cd /usr/ports/net-mgmt/fprobe/
fb# make install clean
fb# rehash
```

3. 启动 Fprobe

查看服务器的接口名称。

```
fb# ifconfig
fxp0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric
0 mtu 1500
        options=8<VLAN_MTU>
        ether 00:90:27:a5:58:16
        media: Ethernet autoselect (100baseTX <full-duplex>)
```

```
        status: active
```

确认接口名称为 fxp0 后，启动 fprobe。

```
fb# fprobe -i fxp0 127.0.0.1:9555
```

上面的命令表示把接口 fxp0 收到的数据转化为 Netflow 数据并发送至本机的 UDP 9555。如果采用 Fprobe 和 Netflow 分开部署的方式，请将 127.0.0.1 改为 Netflow 分析端的 IP 地址，并保证 Fprobe 服务器和 Netflow 服务器可以相互 ping 通。

4. 镜像流量至 Fprobe 服务器

```
Cisco-4507R(config)#monitor session 1 source interface gi1/3 rx
Cisco-4507R(config)#monitor session 1 destination interface gi1/4
```

上面的命令表示将 gi1/3 收到的流量复制到 gi1/4。

5. 检测是否收到 Netflow 数据

```
fb# tcpdump -n -i lo0 dst port 9555
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on lo0, link-type NULL (BSD loopback), capture size 96 bytes
14:00:44.016020 IP 127.0.0.1.65504 > 127.0.0.1.9555: UDP, length 1464
14:00:49.018368 IP 127.0.0.1.65504 > 127.0.0.1.9555: UDP, length 1464
14:00:54.018791 IP 127.0.0.1.65504 > 127.0.0.1.9555: UDP, length 1464
14:00:59.018328 IP 127.0.0.1.65504 > 127.0.0.1.9555: UDP, length 1464
```

如果能收到如上所示信息（桔黄色部分），则表示已经成功的将 4507R 发送的数据转化为 Netflow 数据并发送至本机的 UDP 9555。

五、 部署服务器

Netflow 数据的收集及分析是通过 Nfsen 来实现的 ,Nfsen 同样是 FreeBSD 下的免费软件。它完全基于浏览器进行分析和控制。

(一) 硬件需求

Nfsen 对于 CPU 的要求较内存的更高一些，我们部署了一台服务器，CPU 为 Xeon 3.0，内存为 2G，硬盘 60G。这台服务器处理双向共为 1.8Gbps 的原始数据，Netflow 数据为 1Mbps。从这里也可以看出原始数据与 Netflow 数据的大小对比。

对于一个原始流量在 100Mbps 左右的单位 ,随便在中关村淘一台家用 PC，应该是完全够用的。

(二) 安装 FreeBSD

Nfsen 是一套基于 FreeBSD 搭建的 Netflow 分析平台，请在安装 Nfsen 前先安装 FreeBSD。FreeBSD 是一套免费的操作系统，具有良好的架构和安全性。目前最新版本为 7.0，下载链接为：

ftp://ftp.freebsd.org/pub/FreeBSD/releases/i386/ISO-IMAGES/7.0/7.0-RELEASE-i386-disc1.iso

由于 FreeBSD 官方已经发行了完整的中文安装手册，因此本书不再重述，请参考：

http://cnsnap.cn.freebsd.org/doc/zh_CN.GB2312/books/handbook/install.html

如安装中碰到请发送邮件至 pharaohnie@gmail.com

安装后请进行如下操作

```
freebsd-update fetch install
reboot
portsnap fetch extract
ntpdate time.windows.com
```

freebsd-update fetch install：为 FreeBSD 打补丁

reboot：重启以应用新的补丁

portsnap fetch extract：更新 FreeBSD 的软件至最新版

ntpdat time.windows.com :校准时间，这部很重要，否则分析出的时间不对。

(三) 安装 Nfsen

1. 安装 Apache22

```
cd /usr/ports/www/apache22/
make install clean          //取消勾选 IPV6
echo 'apache22_enable="YES"' >> /etc/rc.conf
echo 'apache22_http_accept_enable="YES"' >> /etc/rc.conf
/usr/local/etc/rc.d/apache22 start
```

2. 安装 Php5

```
cd /usr/ports/lang/php5          //勾选 APACHE，MULTIBYTE，取消勾选 IPV6
echo 'AddType application/x-httpd-php .php' >>
/usr/local/etc/apache22/httpd.conf
echo 'AddType application/x-httpd-php-source .phps' >>
/usr/local/etc/apache22/httpd.conf
cd /usr/local/etc
cp php.ini-recommended php.ini
sed -e 's/short_open_tag = Off/short_open_tag = On/g' -i .bak php.ini
/usr/local/etc/rc.d/apache22 restart
sed -e 's/\/usr\/local\/www\/apache22\/data/\/usr\/local\/www\/nfsen/g' -i .bak
/usr/local/etc/apache22/httpd.conf
sed -e 's/index.html/nfsen.php/g' -i .bak /usr/local/etc/apache22/httpd.conf
/usr/local/etc/rc.d/apache22 restart
```

3. 安装 Nfsen

```
cd /usr/ports/net-mgmt/nfsen          //默认选项即可
make install clean
sed -e '/peer1/d' -i .bak /usr/local/etc/nfsen.conf
nfsen reconfig
/usr/local/etc/rc.d/nfsen start
sed -e 's/\/usr\/local\/www\/apache22\/data/\/usr\/local\/www\/nfsen/g' -i .bak
/usr/local/etc/apache22/httpd.conf
sed -e 's/index.html/nfsen.php/g' -i .bak /usr/local/etc/apache22/httpd.conf
/usr/local/etc/rc.d/apache22 restart
```

(四) 安装 PortTracker

```
cd /usr/ports/net-mgmt/nfdump-devel
```

```
make
cd /usr/ports/net-mgmt/nfsen
make patch
cd work/nfsen-1.3/contrib/PortTracker/
sed -e
's/\.\.\/\.\.\/\.\.\/nfdump-current/\/usr\/ports\/net-mgmt\/nfdump-devel\/work\/nfdu
mp-snapshot-20070808/g' -i .bak do_compile
sed -e 's/\/usr\/local\/rrdtool-1.2.11\/include/\/usr\/local\/include/g' -i .bak
do_compile
sed -e 's/\/usr\/local\/rrdtool-1.2.11\/lib/\/usr\/local\/lib/g' -i .bak do_compile
./do_compile
cp nftrack /usr/local/bin/
rehash
mkdir /usr/local/var/nfsen/portsdb
sed -e 's/\/data\/ports-db/\/usr\/local\/var\/nfsen\/portsdb/g' -i.bak
PortTracker.pm
cp PortTracker.pm /usr/local/libexec/nfsen/plugins/
cp PortTracker.php /usr/local/www/nfsen/plugins/
cd /usr/local/etc
sed -e "s/# profile       # module/[\'live\',\'PortTracker\'],/g" -i .bak nfsen.conf
nftrack -I -d /usr/local/var/nfsen/portsdb/
/usr/local/var/nfsen/portsdb
touch portstat.txt portstat24.txt
chown -R www:www *
/usr/local/etc/rc.d/nfsen restart
```

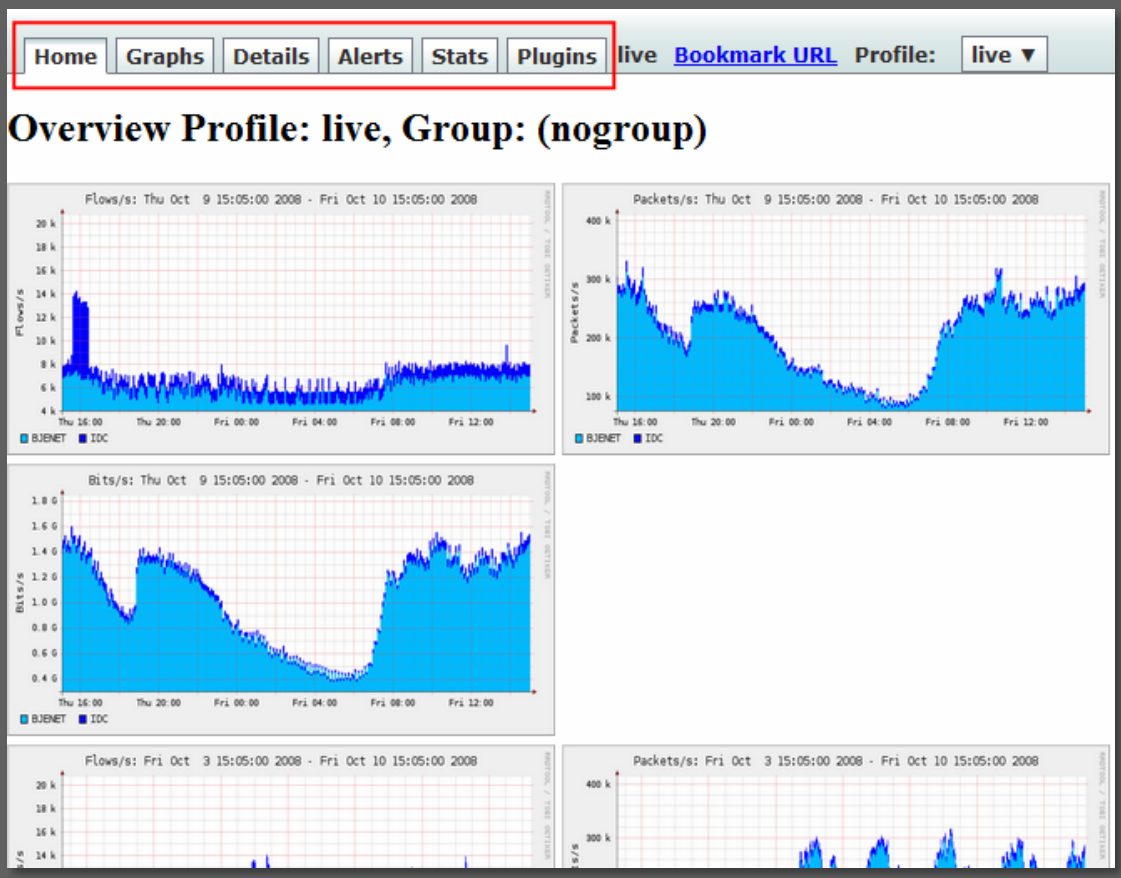
(五) 访问 Nfsen

在做好如上配置后，在浏览器直接输入服务器的 IP 地址，即可直接访问

Nfsen。

六、 抓贼攻略

如果前面的步骤都很顺利的话，到此 Nfsen 应该可以正常工作了，在浏览器中输入 Nfsen 服务器的 IP 地址进入 Nfsen 的 Web 界面，如下图所示：



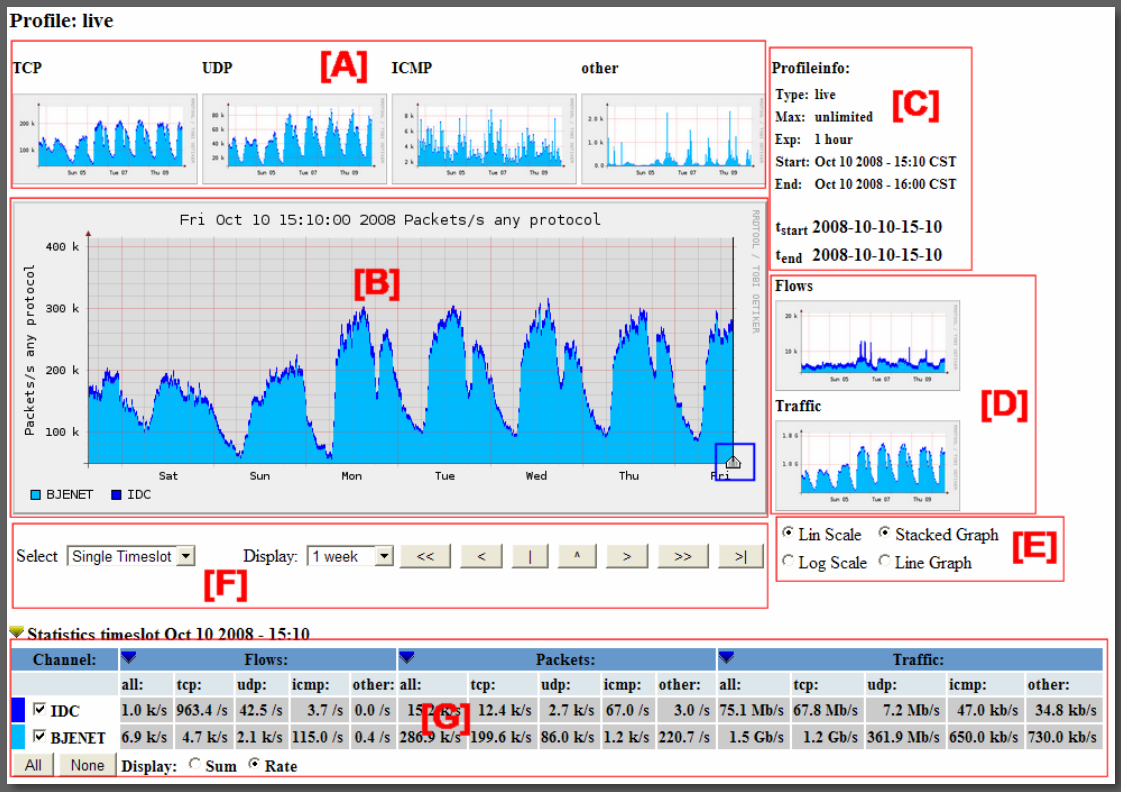
红框内为是 Nfsen 的主要选单。Home，Graphs，Alerts，Stats 是一些概览视图、报警以及配置，与流量的分析关系并不大，因此在这里直接跳过。下面来看看对于流量分析最有用的 2 个选单：Details 和 Plugins。

(一) 了解网络运行状况

我们所管理的网络运行情况是什么样子？比如每秒钟网络出口发送了多少数据，今天一天下来网络出口一共接收了多少数据等等，通过 Nfsen 的 Details 选单，我们可以对自己的网络做到一个大致了解。下面我们来一起探索 Details

选单吧。

点击并进入 Nfsen 上方的 Details 选单，如下图所示。在这个选单里，我们可以了解到网络的一些大致情况，如 TCP、UDP、ICMP 和其他协议的发包速率、流量速率以及流速率（一个流即使用相同源地址、目的地址、源端口、目的端口的一些列数据包。例如从 A 向 B 发送一张图片，期间发送的数据包会有多个，但数据流只有一个），并且可以根据历史时间进行查询等。



上图中用[A]-[G]标出了几个小的区域，下面我来依依解释：

[A] 在这个区域里可以选择查看 TCP、UDP、ICMP、其他协议和所有协议的相关参数。

[B] 在[A]区域选择相关协议后，会在[B]区域以大图显示。并且在此区域有一时间标（蓝色方框内），拖动此时间标可以选择需要查看的时间内的网络信息。

[C] 这个区域主要显示了当前时间标所在的时间，拖动时间标后，这个区域里的时间也会相应的改变。

[D] 在这个区域可以选择查看发包速率、流量速率或者流速率。

[E] 这个区域中可以选择让图表以线性方式、指数方式显示或者以堆积图或者线图显示。一般来说，以线性方式+堆积图的方式显示比较直观。

[F] 这里可以选择时间标的类型，包括某一时刻（Single Timeslot）或者一段连续时间内（Time Window），当选择一段连续时间内（Time Window）时，[B]区域内的时间标可以分别向左右两个方向拖动，以选择一段连续的时间。并且，在这个区域内还可以通过选择 Display 来选择半天、一天、两天、四天、一周、两周、一个月的时间来显示图表。结合时间标的选择（某一时刻或者一段连续时间内），可以做到某一段精确时间内的网络参数查询。此外，在[F]区域内还有一些向录音机按钮一样的按键，通过这些按键可以快速的控制时间标，这个非常简单，可以由大家自己去体会。

[G] 此区域内主要以表格形式显示了所有协议的发包速率、流量速率和流速率。拖动时间标，[G]区域内的数字也会随之改变。此外，选择总量（Sum）可以在表格中显示总发包数、总流量、总流数，选择速率（Rate）则可以显示发包速率、流量速率、流速率。最后，大家可以看到表格的最左侧有 IDC 和 BJENET 两行，这是因为在本示例中这台 Nfsen 服务器接收了 2 个 Netflow 数据源的缘故。因此，也可以看出一台 Nfsen 服务器可以同时分析多个 Netflow 数据的。

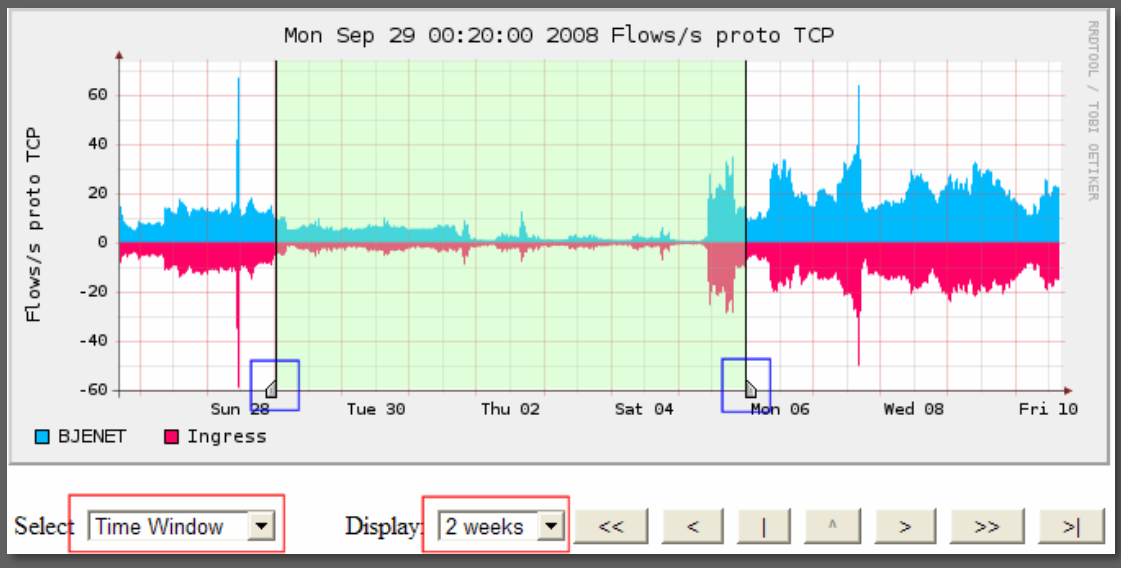
那么我们可以举一些例子来帮助大家更好的使用 Details 选单来了解网络运

行情况。

- 查询今天凌晨 4 点这一时刻的流速率。
  - 点击[D]区域的 Flows，拖动[B]区域内的时间标至凌晨四点。

- 查询今天凌晨 4 点这一时刻的总流数。
  - 点击[D]区域的 Flows，拖动[B]区域内的时间标至凌晨四点，再选择[G]区域内的 Sum。仔细观察与上面的例子间所得结果的区别。

- 查询上一周内 TCP 协议的发包速率。
  - 首先，在[A]区域中选择 TCP。其次，在[F]区域内将 Display 选择为 2 weeks，之后 Select 改为一段连续时间内（Time Window），现在可以试着拖动[B]区域中的时间标，来选择上周的数据。如下图所示：



通过对一段正常时期内网络运行数据的观察，大家应该已经了解网络的基本运行情况了，请认真记下这些数据，尤其是速率类数据（发包

速率、流量速率、流速率），一旦发现其中任意一个速率类数据大幅增长，那么你负责的网络已经出现了异常情况，赶快去排查吧。

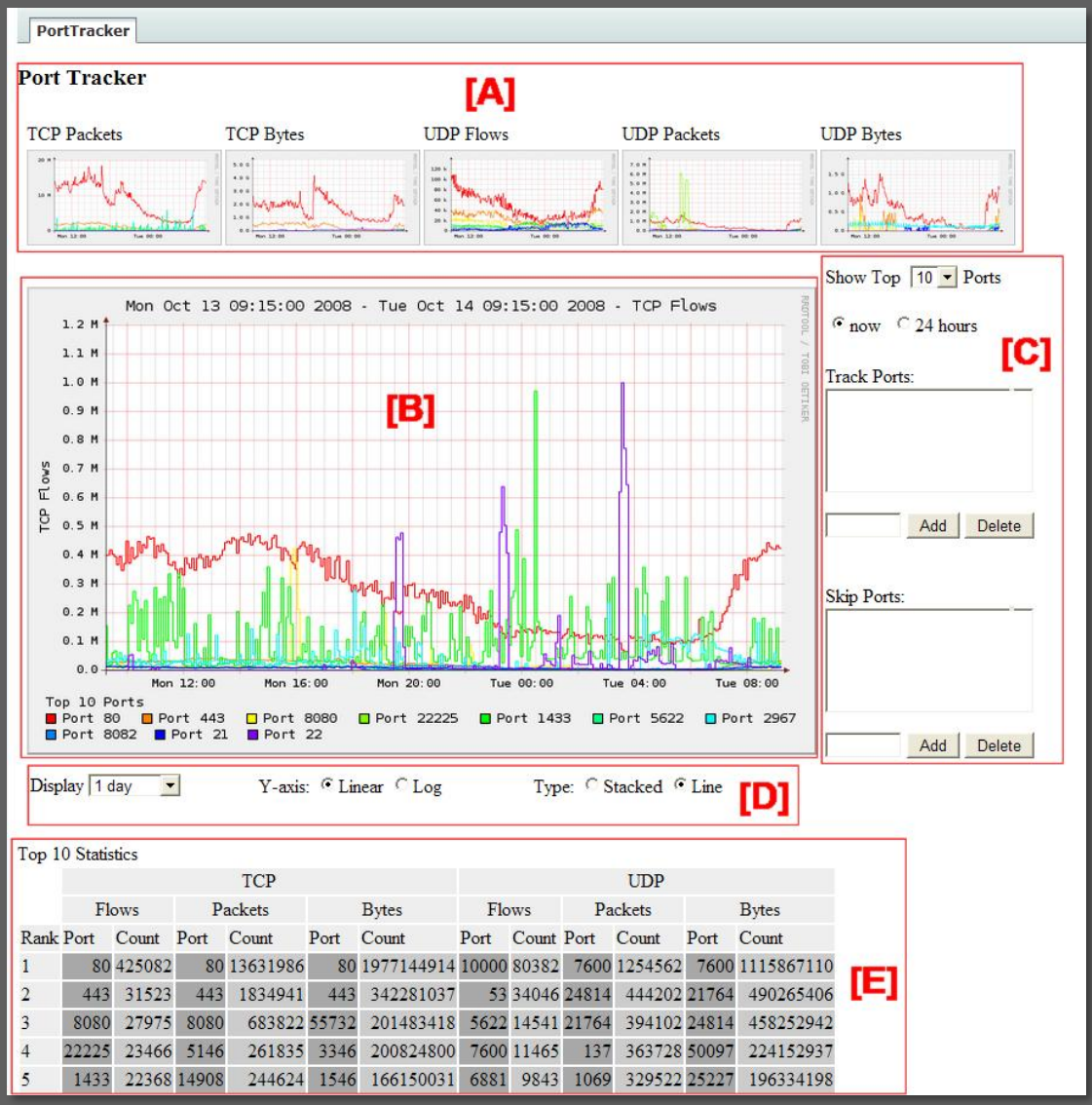
至此，我们已经知道如何了解自己所负责网络的一个基本概况。那么到底如何才能判断出是什么类型的流量占据了网络的大部分带宽而让公司的网这么慢呢？

(二) 什么协议吞了带宽

现在，我们知道自己网络的一个运行状况了。“一直持续在 10Mbps，真不小啊！平时公司里也就是上网、收发邮件之类的操作，怎么会这么大流量呢，到底是什么应用占了这么大带宽呢？“

下面我们就来看看如何用 Nfsen 的插件 Port Tracker 揪出占网络大量资源的协议。Port Tacker 是 Nfsen 官方的一个插件，他的主要功能是对网络协议（即端口）按流量速率、发包速率、流速率或者流量、发包数、流数进行排序，以帮助管理员发现非正常流量。

点击 Nfsen 的 Plugins 选单即可进入 Port Tracker 视图。如下图所示：



	TCP						UDP					
	Flows		Packets		Bytes		Flows		Packets		Bytes	
Rank	Port	Count	Port	Count	Port	Count	Port	Count	Port	Count	Port	Count
1	80	425082	80	13631986	80	1977144914	10000	80382	7600	1254562	7600	1115867110
2	443	31523	443	1834941	443	342281037	53	34046	24814	444202	21764	490265406
3	8080	27975	8080	683822	55732	201483418	5622	14541	21764	394102	24814	458252942
4	22225	23466	5146	261835	3346	200824800	7600	11465	137	363728	50097	224152937
5	1433	22368	14908	244624	1546	166150031	6881	9843	1069	329522	25227	196334198

Port Trakcer 视图中，共分为[A]~[E]，下面逐一进行介绍：

[A] 这里可选择对 TCP 或者 UDP 的协议（即端口）进行排序，并且排序方法有 3 个，按发包数（Packets）排序，按流量（Bytes）排序，按流数（Flows）排序。一般情况，按流量排序和按发包数排序更为常用。

[B] 在[A]区域选择排序方式后，会在[B]区域中以大图的方式显示，并在大图的下方以图示由小到大列出端口排名。

[C] Show Top 选单可以选择前几位排名，默认是前 10 位排名，一般情况下都是某一两个应用占据大量带宽，而且按前 10 位排名会消耗更多的 CPU 资源，

所以按前 5 位排名就足够了。下面可以选择当前时间排名（now）和按天进行的统计排名（24 hours）。当然，你就想知道某一个端口的相关网络参数也可以自行添加（Track Ports）。如果你认为 FTP 是你网络中的正常应用，没必要对其进行排名统计，你也可以忽略此端口的排名（Skip Ports），因为这个 2 个功能并不多用，这里就不再详细讲解，大家可以自己测试一下。

[D] 这个区域里共有 3 个选项。第一个选项（Display）和 Details 选单中的 Display 类似，可以选择 12 小时、1 天、2 天、4 天、1 周、2 周内的统计排名，选择的时间范围越长，CPU 消耗越多，计算时间也越长。第二个选项 Y-axis 可以选择按线性（Linear）和按指数（Log）显示图表，一般来数还是按线性（Linear）显示看着比较舒服。第三个选项 Type 可以设置图表显示类型 :堆积图(Stacked)和线形(Line),建议大家使用堆积图(Stacked) 方式查看图表，这样更容易看出排名端口所占网络资源的比例关系。

[E] 本区域中以表格的方式详细显示了 TCP、UDP 两种协议按流数（Flows）、发包数（Packets）、流量（Bytes）的排名。

下面，我还是给大家举一些例子吧。

- 我想看看今天 TCP 中哪些协议占用的网络带宽最大，看前 3 名就足够了。
  - 首先在[A]区域中选择对 TCP 按流量进行排序（TCP Bytes），[C]区域中 Show Top 选择 3，并选择按天进行的统计排名（24 hours），[D]区域中 Display 选择 1 天（1 day）。

- 现在网络特慢，查了查路由器，负载挺高的（一般情况下网络设备的负载高

都是由于很多非正常包必须由 CPU 处理所造成的，查出这些包并且用访问列表过滤掉就可以了），有可能是网络里的碎包太多了，查查目前的发包数的前 5 名排名吧。

- 因为无法判断到底是 TCP 协议的包数多还是 UDP 协议的包数多，因此需要在[A]区域中分别选择对 UDP 按发包数排序（UDP Packets）和对 TCP 按发包数排序（TCP Packets），并对比查看。之后[C]区域中 Show Top 选择 5，并选择当前时间排名（now），[D]区域中 Display 选择 1 天（1 day）就可以了。

现在，我们应该有能力查出到底哪些协议占据了大量网络资源。那么，我们该行动了么？先不要这么鲁莽，继续往下看。

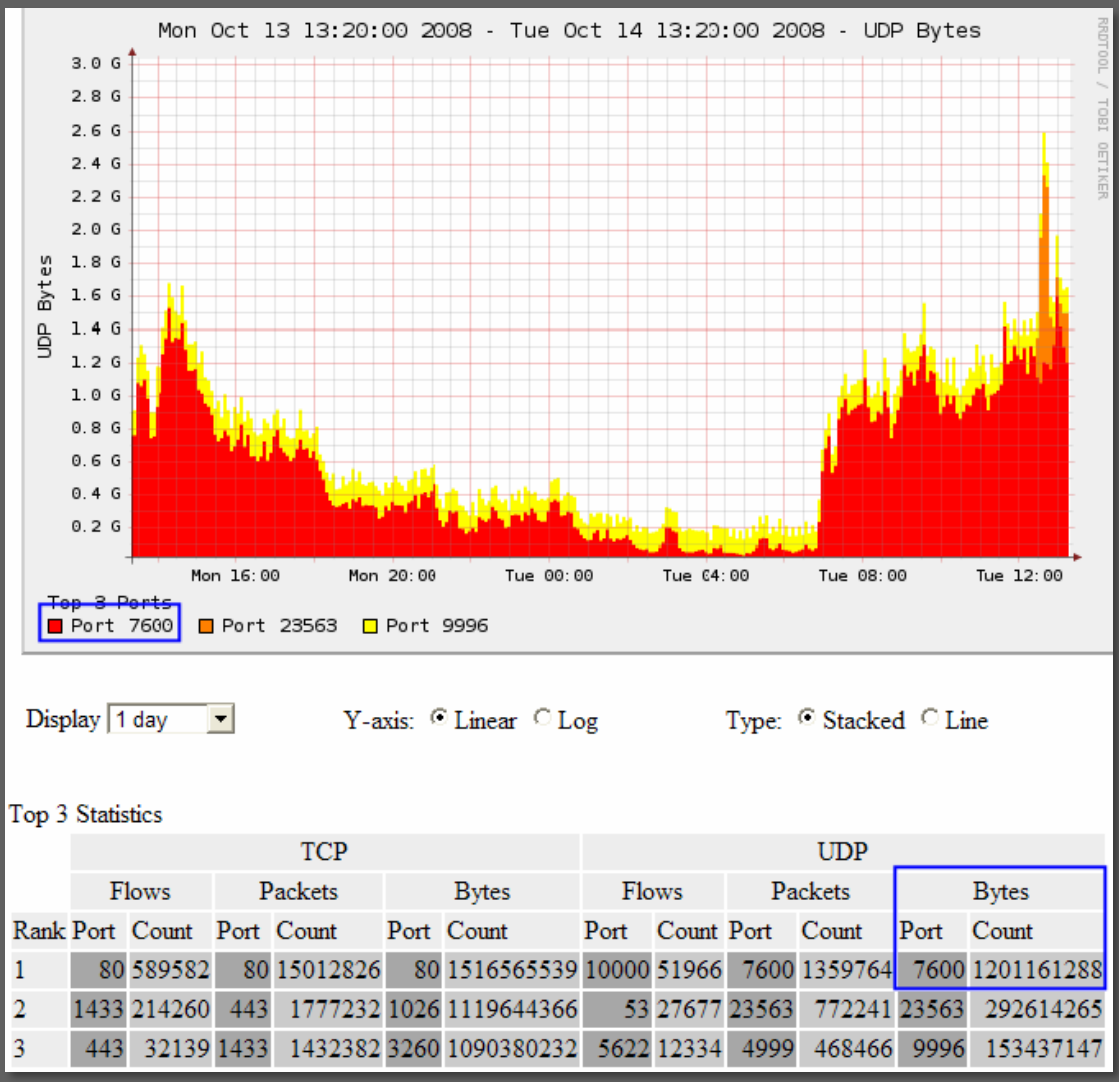
(三) 抓出罪魁祸首

也许有的网管员看了上面的内容后便开始行动了，了解网络运行情况，查看哪些协议占带宽最大。恩，不错，不出 3 分钟就查出来了，TCP 1433，封了吧。先等等！现在有 2 个问题，TCP 1433 是什么应用，如果只是某一台主机的 TCP 1433 不正常，把所有端口都封了，会不会影响到正常的 TCP 1433 应用。

答案当然是只希望封掉有非正常主机的非正常端口，又不是蒋委员长，何必宁错杀一千，不放过一个。

下面咱们来看看如何抓出真正的罪魁祸首。假设使用 Port Tracker 已经查

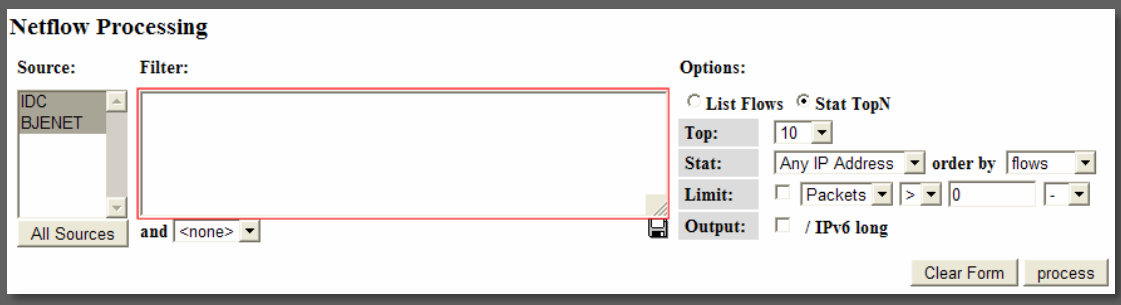
出 UDP 7600 端口的流量非常大，如下图篮框中所示：



Top 3 Statistics

	TCP						UDP					
	Flows		Packets		Bytes		Flows		Packets		Bytes	
Rank	Port	Count	Port	Count	Port	Count	Port	Count	Port	Count	Port	Count
1	80	589582	80	15012826	80	1516565539	10000	51966	7600	1359764	7600	1201161288
2	1433	214260	443	1777232	1026	1119644366	53	27677	23563	772241	23563	292614265
3	443	32139	1433	1432382	3260	1090380232	5622	12334	4999	468466	9996	153437147

下面的工作就相对简单了，只要找出问题主机即可。点击 Details 选单，将光标定位至 Netflow Processing 的 Filter 处。

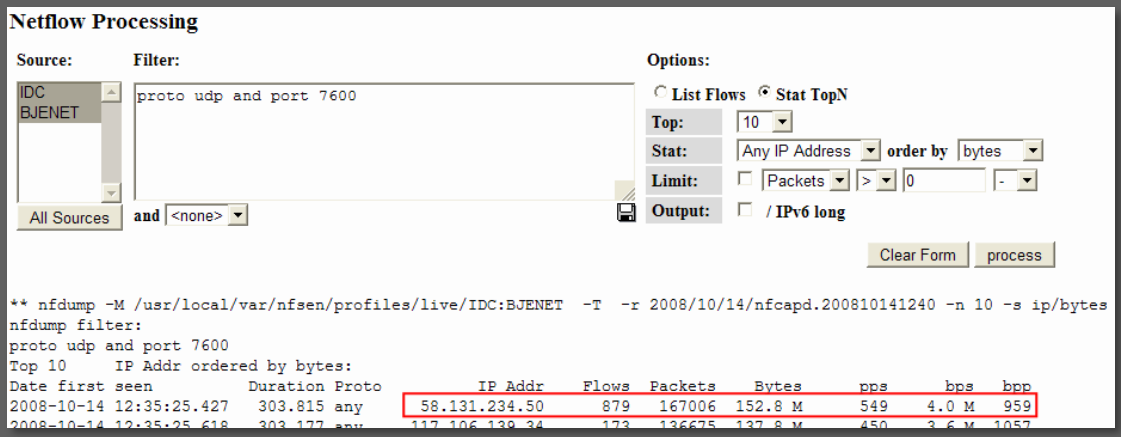


这里就要用到 tcpdump 的语法规则了（对这部分不熟悉的同志可以去 http://www.tcpdump.org/tcpdump_man.html 查询），在 Filter 处键入：

查看用7600的主机

```
proto udp and port 7600    //只查看 UDP 协议和 7600 端口的网络数据
```

Options 中选择状态排名（Stat TopN），状态（Stat）选择任意 IP 地址（Any IP Address），排序（order by）选择字节（bytes），以上选项意为对 UDP 7600 端口按流量大小排序，并列出 IP 地址。点击 process 开始进行排序，结果如下：

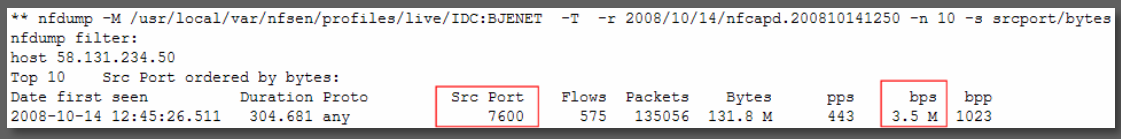


从这里可以看出 58.131.234.5 的 UDP 7600 流量非常大，高达 4Mbps 之多。那么我们可以试着进行更深一步的判断：到底是源端口产生的流量还是目的端口产生的流量。

继续在 Filter 中重新输入：

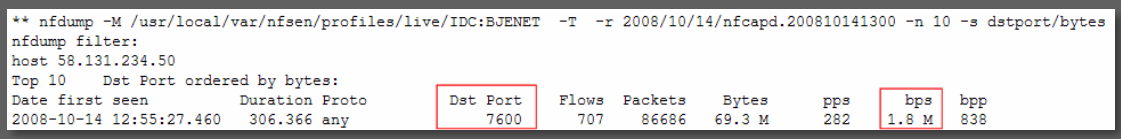
```
host 58.131.234.50    //只查看主机 58.131.234.50 的网络数据
```

修改 Options 中的状态（Stat）为源端口（SRC Port），列出源端口，点击 process 开始排序，结果如下：



如上图所示，这台主机源端口 7600 的流量速率为 3.5Mbps。下面把 Options 中的状态（Stat）修改为目的端口（DST Port），列出目的端口，点击

process 开始排序，结果如下：



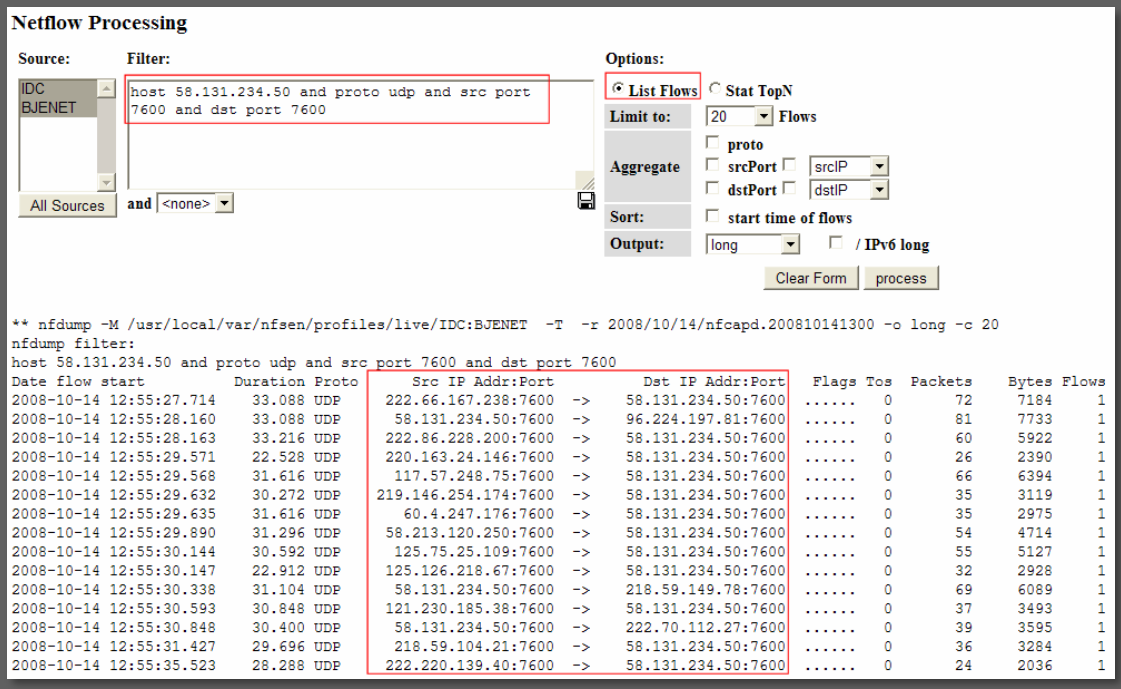
```
** nfdump -M /usr/local/var/nfsen/profiles/live/IDC:BJENET  -T  -r 2008/10/14/nfcapd.200810141300 -n 10 -s dstport/bytes
nfdump filter:
host 58.131.234.50
Top 10    Dst Port ordered by bytes:
Date first seen          Duration Proto        Dst Port   Flows  Packets    Bytes      pps     bps  bpp
2008-10-14 12:55:27.460   306.366 any             7600     707    86686   69.3 M      282   1.8 M  838
```

如上图所示，目的端口 7600 的流量速率为 1.8Mbps，也不小啊。一般来说使用相同源端口和目的端口的程序并不多见，很有可能是病毒作怪。如果有兴趣，咱们再看看，到底都是什么样的流量再作怪。

在 Filter 中重新输入：

host 58.131.234.50 and proto udp and src port 7600 and dst port 7600　　//只查看和主机 58.131.234.50 通讯，且源端口和目的端口都是 UDP 7600 的网络数据。

Options 中选择列出流（List Flows），这样即可列出所有和主机 58.131.234.50 通讯，且源端口和目的端口都是 UDP 7600 的网络数据记录，点击 process 进行统计，结果如下：



Netflow Processing

Source: IDC BJENET — All Sources

Filter: host 58.131.234.50 and proto udp and src port 7600 and dst port 7600

and <none>

Options: List Flows / Stat TopN; Limit to: 20 Flows; Aggregate: proto, srcPort srcIP, dstPort dstIP; Sort: start time of flows; Output: long / IPv6 long; Clear Form; process

```
** nfdump -M /usr/local/var/nfsen/profiles/live/IDC:BJENET  -T  -r 2008/10/14/nfcapd.200810141300 -o long -c 20
nfdump filter:
host 58.131.234.50 and proto udp and src port 7600 and dst port 7600
```

Date flow start	Duration	Proto	Src IP Addr:Port		Dst IP Addr:Port	Flags	Tos	Packets	Bytes	Flows
2008-10-14 12:55:27.714	33.088	UDP	222.66.167.238:7600	->	58.131.234.50:7600	.....	0	72	7184	1
2008-10-14 12:55:28.160	33.088	UDP	58.131.234.50:7600	->	96.224.197.81:7600	.....	0	81	7733	1
2008-10-14 12:55:28.163	33.216	UDP	222.86.228.200:7600	->	58.131.234.50:7600	.....	0	60	5922	1
2008-10-14 12:55:29.571	22.528	UDP	220.163.24.146:7600	->	58.131.234.50:7600	.....	0	26	2390	1
2008-10-14 12:55:29.568	31.616	UDP	117.57.248.75:7600	->	58.131.234.50:7600	.....	0	66	6394	1
2008-10-14 12:55:29.632	30.272	UDP	219.146.254.174:7600	->	58.131.234.50:7600	.....	0	35	3119	1
2008-10-14 12:55:29.635	31.616	UDP	60.4.247.176:7600	->	58.131.234.50:7600	.....	0	35	2975	1
2008-10-14 12:55:29.890	31.296	UDP	58.213.120.250:7600	->	58.131.234.50:7600	.....	0	54	4714	1
2008-10-14 12:55:30.144	30.592	UDP	125.75.25.109:7600	->	58.131.234.50:7600	.....	0	55	5127	1
2008-10-14 12:55:30.147	22.912	UDP	125.126.218.67:7600	->	58.131.234.50:7600	.....	0	32	2928	1
2008-10-14 12:55:30.338	31.104	UDP	58.131.234.50:7600	->	218.59.149.78:7600	.....	0	69	6089	1
2008-10-14 12:55:30.593	30.848	UDP	121.230.185.38:7600	->	58.131.234.50:7600	.....	0	37	3493	1
2008-10-14 12:55:30.848	30.400	UDP	58.131.234.50:7600	->	222.70.112.27:7600	.....	0	39	3595	1
2008-10-14 12:55:31.427	29.696	UDP	218.59.104.21:7600	->	58.131.234.50:7600	.....	0	36	3284	1
2008-10-14 12:55:35.523	28.288	UDP	222.220.139.40:7600	->	58.131.234.50:7600	.....	0	24	2036	1

从上图看出，58.131.234.50 使用 UDP 源端口 7600 和很多不一样的主机的 UDP 目的端口 7600 进行了通讯。直到这里，我们可以斩钉截铁的说，这个流量一定是异常流量，封掉它。开始行动吧。

```
access-list 101 deny udp host 58.131.234.50 eq 7600 any
access-list 101 deny udp host 58.131.234.50 any eq 7600
access-list 101 permit ip any any
!
interface fastethernet 0/1
 ip access-group 101 in
```

至此为止，一次完整的网络流量问题排查结束。

一次完整的排查过程：

Details 选单：了解网络运行情况。

Plugins 选单：对协议进行排序，查出什么协议占了大部分网络资源。

Details 选单：找出罪魁祸首。

七、 感谢

这本书的完成，要感谢很多人。

- 老婆大人-雪娇（http://xuejiao.name），厨艺超好，给我做很多饭菜，饱了肚子，营养了大脑。
- Cisco，带我走进了网络的大门。
- 小丘、小芊，可爱的女儿们，在我写书之余不至于那么无聊。