CATI
Charging and Accounting Technology for the Internet
SNF SPP Projects 5003-054559/1 and 5003-054560/1

Fluxoscope
a System for Flow-based Accounting

Simon Leinen
SWITCH
Limmatquai 138, CH-8001 Zurich, Switzerland
Phone: +41 1 268 1530 FAX: +41 1 268 1568
E-Mail: simon@switch.ch

Workpackage and Task:	**WP4T2**
Deliverable Identifier:	**4.2.2**
Status:	**Public**
Version:	**0.4**
Revision:	**-**
Reviewed by:	**-**
Due Date:	**31.03.2000**
Delivery Date:	**15.03.2000**

Abstract

We present a traffic accounting system developed at SWITCH. Its applications include differentiated usage-based charging, long-term traffic analysis for capacity planning, and troubleshooting tasks such as the detection of routing anomalies or denial-of-service attacks. The system is based on *flow-based* accounting information generated by routers.

1. Introduction

In this paper, we present an accounting system which is used at SWITCH for purposes of charging, monitoring, and traffic planning. This system, called “Fluxoscope”, is based on Cisco NetFlow accounting, where routers apply fixed rules to group traffic into flows, generate accounting records for each flow, and send those accounting records to remote management stations.

The paper is organized as follows: In section 2. (Motivation), we present the reasons for developing the system. The router-based accounting mechanism we use is introduced in section 3. (Cisco NetFlow). The Fluxoscope system, its components and design, are described in section 4. (Fluxoscope). Finally, section 5. concludes the paper and gives an outlook on future plans for the system.

2. Motivation

SWITCH is a non-profit foundation which provides telecommunications services to the Swiss academic and research communities. Those services are mainly IP-based and include a backbone network service as well as high-speed connections to the rest of the Internet. As central funding only covers a small part of the operating costs, the larger part of those costs have to be recovered directly from connected organizations. In the past, an undifferentiated volume-based charging model has been used, in which the charges for a given organization depended mainly on the amount of traffic received[1] by that organization, irrespective of the origins of that traffic.

Advantages of this charging scheme include that it was easy to implement for the service provider, and easy to understand for users. However, it doesn’t reflect costs of operation well at all; those costs are very different e.g. for “on-net” traffic between two connected organizations and for traffic entering or leaving the network over expensive international links. In some cases, the incentive to reduce total traffic has caused connected organizations to cancel high-traffic services which are actually very cheap to provide, such as full USENET newsfeeds, or prevented the wider adoption of new ones, such as multicast connectivity.

On the other hand, moving to a “flat” charging scheme was considered undesirable by the user community, which found it more equitable if charges are distributed according to the respective costs generated by each organization’s users.

It was recognized that traffic over international links, and in particular the transatlantic connection, was the most important cost factor, and that the usage of such connections should be charged by usage, while “on-net” traffic over the new backbone network should be charged at a flat rate. Thus in 1997, SWITCH started investigating possibilities of more differentiated charging schemes. One major difficulty was to devise a new accounting system that could generate the more detailed base data for such a scheme.

[1] Connected organizations are only charged for inbound traffic. Reasons for this are (1.) that we want to encourage rather than penalize provision of access to interesting information by our members, and (2.) that traffic is largely asymmetric, and congestion never occurs in the outbound direction, so additional outbound traffic wouldn’t incur any additional cost.

3. Cisco NetFlow

In 1996, Cisco Systems started shipping a new router-based accounting mechanism called NetFlow[1]. When appropriately configured, a router sends a stream of accounting packets to a given UDP receiver (typically a workstation), where those data can be further aggregated and analyzed. Each accounting packet contains data about several ‘‘flows’’, where each flow is defined by source and destination IP address, protocol (e.g.: UDP/TCP/ICMP...), source and destination port number (for UDP/TCP; type/code for ICMP), and TOS byte. Accounting data sent for each flow include: number of packets seen number of bytes seen, time of first packet seen, time of last packet seen, incoming interface, as well as some information based on the router’s routing tables, such as outgoing interface, source/destination network mask, source/destination Autonomous System (AS) number.

Because we were already using that vendor’s routers, we investigated its applicability in the context of the desired new approach to charging. For our accounting needs, this system turned out to strike a nice balance between accounting information level of detail and the amount of accounting data that has to be processed both by the routers and by external accounting software. An important requirement was that the impact on routing performance shouldn’t be so high as to create new bottlenecks at the levels of traffic that can be expected over the next few years.

4. Fluxoscope

When we started out the accounting project, we had rather vague ideas on what we wanted to measure and how NetFlow accounting could be used for that. We first looked at ANS’ *CFLOWD*[2] system but found that it was lacking appropriate aggregation mechanisms for use in our context. Because the further requirements were rather unclear at that time, we decided to develop our own aggregation and analysis software, rather than trying to add this functionality into CFLOWD,.

In November 1997, the first version of the “NetFlow listener” became operational. Since then it has been modified to add new NetFlow accounting functionality, eliminate bugs and enhance robustness. The system has later been renamed to “Fluxoscope” and used as a basis for experimental functionality unrelated to charging, such as the detection of anomalous traffic from DOS (Denial-of-Service) attacks.

4.1 Overall Architecture

Fluxoscope is a distributed system consisting of three larger components:

- The “listener” collects accounting data sent by one or several routers, performs some aggregation on them, splits them into time slices, and periodically writes new data out to files. Listeners should be placed near the NetFlow-generating routers in order to reduce load on the network and to avoid loss of accounting data.
- Data collection and maintenance tools periodically access the files that are generated by the distributed listeners and copy them to central data store. Files are compressed and archived once they pass a certain age. In addition, the data from those periodic files are summed up over longer periods to make computation of long-term statistics efficient.
- Data analysis tools access the central data store to generate various forms of reports, such as tabular data as input to the billing process, and graphical representations for network monitoring and long-term traffic analysis.

As we are mostly interested in “off-net” traffic, we run NetFlow accounting on each external BGP-4 border router[1]. In order to see both the traffic that enters our network and the traffic that leaves it, NetFlow accounting must be enabled on every interface of such a router.

4.2 NetFlow Listener

The listener receives one or multiple NetFlow accounting streams from external border routers. It first filters out all accounting records pertaining to “internal” (on-net) traffic (see section 4.2.3 “Recognizing External Traffic” below on how this is done).

The remaining traffic is "off-net" and should be accounted for. The aggregation method we use generates aggregates with the following coordinates:

- external AS number (provider or peer)
- customer ID
- “application protocol” (see section 4.2.4 “Recognizing Application Protocols” below)

For each aggregate, we store input and output bytes for time slices at regular intervals. Currently, we use five-minute intervals for compatibility with the widely-used MRTG tool[4].

The output files of the listener look something like this (the lines can be very long and are broken here for readability):

```
ETHZ            0       201345041 in   27181873 out WWW:142170251/19693962 TCP-other:35551867/1006788 FTP:17057922/
2702803   SMTP:1109871/2864752   UDP-other:3768462/66150   NNTP:652178/24579   DNS:402003/250591   NTP:240882/247417
ICMP:98182/127481 POP:124036/22314 IDENT:58468/68166 AFS:21011/81197 TELNET:76998/21313 IRC:11695/4090 Gopher:1023/
270 SQUID:192/0
UNIGE           0       199329295 in   23573096 out WWW:126172701/18274961 TCP-other:58591947/3200071 FTP:8745291/
591395  UDP-other:4416349/69110  SMTP:958975/1151480  DNS:400923/237254  ICMP:28237/35011  POP:6983/6447  TELNET:5944/
4607 IDENT:1869/2684 NTP:76/76
UNIZH           0       127180139 in    7989067 out WWW:114876625/6642215 FTP:4474557/92485 TCP-other:3990325/107856
UDP-other:2191436/47435  SMTP:640793/833236  NNTP:776938/17803  DNS:144439/89389  ICMP:30712/72954  TELNET:34254/52619
IP-in-IP:8731/10651 AX.25:1828/12838 IDENT:4629/5710 POP:4568/3496 NTP:304/380
...
SNL           6730             112 in          0 out DNS:112/0
VSNET         8243              40 in         40 out WWW:40/40
EHL           5378               0 in         72 out DNS:0/72
SNL           5378               0 in         62 out DNS:0/62
CSCS          5378               0 in         54 out DNS:0/54
```

4.2.1 Flow Aggregation

SWITCH connects about 15 universities and research centers (jointly referred to as ‘‘sites’’ throughout the paper) to the Internet, each of which must be billed independently. In addition, we have external connections to around fifteen other ISPs; two are paid transit connections, the rest are settlement-free peerings.

It was defined early on that the new charging scheme should be free of volume-dependent charges for ‘‘on-net’’ (inter-site) traffic. Therefore we chose to only measure traffic with an external source or destination, and aggregate the flow-based accounting data according to ‘‘customer’’ organization (site) and external (peer/provider) network.

[1] An exception is our border router in New York; we don’t have a workstation at our New York PoP where the listener could be run, and sending the accounting stream over the transatlantic link would be very expensive, so we don’t run NetFlow accounting on that router. Instead, the interfaces where the transatlantic lines terminate (in Switzerland) are treated as if they were external connections, and the corresponding routers must run NetFlow accounting.

4.2.1.1 Peer Network

NetFlow accounting data already includes the AS numbers of the source and destination addresses, so this can be used to derive the external network. NetFlow routers can be configured to send either the ‘‘neighbor’’ or the ‘‘origin’’ AS numbers, but not both, nor any in between. As we are mostly interested in data per direct external connection, we use the ‘‘neighbor-AS’’ configuration on routers.

Like many other ISPs outside North America, SWITCH doesn’t carry a full BGP[5] routing table on its backbone. Instead, only routes from our European providers and peers are learned explicitly, while a default route points to our US provider for the rest of the Internet. For our situation in the global topology, this results in optimal routing without the overhead (in terms of router memory, processing cost, and convergence time) of carrying full routing on the entire backbone.

For source/destination addresses that are covered by the default route, NetFlow accounting packets include a respective AS number of zero. Unfortunately the same number is used for addresses that are routed by an internal routing protocol rather than by BGP. So for each of the (many) addresses where NetFlow gives us an AS of zero, we have to decide whether the address is local (belongs to a connected site or the backbone infrastructure) or external (belongs to an external network reached through the default route).

4.2.1.2 Connected Site

None of the sites connected to SWITCH use an AS number of their own; their addresses are all announced under SWITCH’s AS, and routing information between sites and the SWITCH backbone is exchanged using internal routing protocols such as RIP, OSPF, or static routes.

In order to find the site corresponding to an internal source/destination address, we use a longest-prefix match in a small table that maps IP address ranges to site identifications. This table is an extension of the extract of the routing registry database[6] corresponding to our own Autonomous System (AS559).

4.2.2 Time Slices

While the majority of flows is quite short-lived (less than a minute between first and last packets seen), a few long-lived flows can account for a large volume of traffic. Such long-lived flows are typically associated with large file downloads (FTP), audio/video streams, USENET newsfeeds (interserver traffic), or tunnels. If one would simply put all the traffic of each flow record into current time-slice, this would lead to spikes in the traffic graphs, as the maximum lifetime of a flow before being accounted for (thirty minutes by default, although this has become configurable in later NetFlow versions) can be much larger than the five minute time slices we use.

Therefore we look at the times when the first and last packets of the flow were seen, and distribute the byte count evenly over the time slices in that range. Although an even distribution of traffic within long-term flows is a simplification, this results in much more accurate traffic graphs.

4.2.3 Recognizing External Traffic

All router interfaces must be categorized into three classes: “backbone” interfaces connect to other backbone routers or other infrastructure, “customer” interfaces connect to member organization’s networks, and “external” interfaces connect to external networks to which we have a peering or customer relationship. The type of each interface should be documented in a configuration file (the ‘‘interface categorization file’’) by range of IP addresses.

A NetFlow accounting record is considered to pertain to external (off-net) traffic if either the input or the output interface in the record represents an external interface.

The interface indices in NetFlow accounting records are the same used in the SNMP ifTable[3]. When the listener receives a NetFlow accounting packet from a hencetoforth unknown router, it reads the IP address configuration of each interface using SNMP. This interface configuration is then combined with the interface categorization file to derive a mapping from interface index to interface class.

The listener needs SNMP read access to some columns of the ipAddrTable as well as the ifDescr and ifOperStatus columns of the ifTable for this. The router’s sysName is used for readable messages. Currently, SNMPv1 or SNMPv2c can be used, so a community string with appropriate access rights must be known to the listener.

4.2.4 Recognizing Application Protocols

The listener tries to map each flow to a “well-known” application according to the protocol type and TCP/UDP port numbers. The set of "interesting" applications and the rules to recognize them must be configured a priori. It is expected that this set changes as new applications emerge as generators of significant traffic, and as existing applications become less important. The currently recognized application protocols are listed in Table 1.

TABLE 1. Recognized Application Protocols

Name	Detection Method
WWW	TCP ports 80, 81, 8000, 8080, 443
DNS	UDP/TCP port 53
NNTP	TCP port 119
FTP	TCP ports 20/21; passive-mode heuristics
TELNET	TCP ports 23 (telnet), 22 (SSH), 513 (login), 514 (shell)
SMTP	TCP port 25
ICMP	protocol 1
X	TCP ports 6000-6002
IRC	TCP port 6667
NTP	UDP port 123
SQUID	UDP/TCP ports 3128-3130
IP-in-IP	protocol 4
IPv6-in-IPv4	protocol 41
IDENT	TCP port 113
Gopher	TCP port 70
AFS	TCP ports 7000-7007
AX.25	protocol 93
GRE	protocol 47
POP	TCP ports 109/110
IGMP	protocol 2
TCP-other	protocol 6

Name	**Detection Method**
UDP-other	protocol 17
other	any other protocol

Although application protocol information isn't necessary for the charging application, it can be quite interesting for traffic planning and troubleshooting purposes.

A large part of the traffic can easily be classified as e.g. WWW, DNS, SMTP, NNTP traffic based on IANA-assigned or otherwise well-known UDP and TCP port numbers. However, there remains an important amount of TCP traffic with seemingly random port numbers. Further investigation shows that much of this traffic can be attributed to FTP data transfers in passive mode. In contrast to ''classic'' active-mode FTP data transfer, which uses TCP port 20 (ftp-data), both port numbers of passive-mode transfers are basically random. However, we can use an accounting record for an FTP control (TCP port 21) flow as a strong indication that TCP traffic between the same hosts with unknown TCP ports was actually FTP data traffic.

4.2.5 SNMP Agent

The NetFlow listener is configured to be started at boot time on the servers where it should run. Its status can be accessed using SNMPv1 or SNMPv2 requests to an SNMP agent which is integrated in the listener. The agent implements a custom MIB defined according to the SMIv2 rules. It includes a few global variables such as the total numbers of NetFlow packets and accounting records processed, as well as a table indexed by ''NetFlow engine''. Any entity from which an independent stream of NetFlow accounting packes has been received is considered as such an engine[1]. The table includes, for each engine, counts of accounting packets and accounting records processed, the next sequence number expected from the engine, and a counter of flows that have been lost.

FIGURE 1. Amount of accounting data processed by Zurich listener

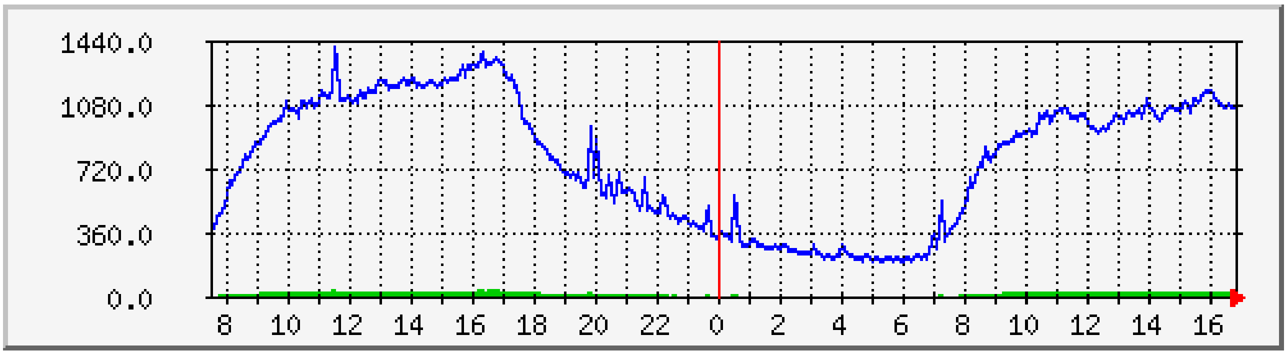


Figure 1 shows the number of accounting records processed from two border routers in Zurich, aggregating our US traffic as well as five peerings. During peak hours, the system has to cope with about 45 accounting packets per second, corresponding to more than 1,400 flows.

[1] a NetFlow engine can be a Cisco router running NetFlow in non-distributed mode, or a component such as an intelligent line card running NetFlow in distributed mode

4.3 Data Management

Fluxoscope's data management component consists of scripts which are periodically executed from the machine which hosts the master data store. These scripts perform the following tasks:

- Copy recently created/updated files from remote NetFlow listeners to the data store
- Remove old files from remote NetFlow listener machines
- Combine data files for larger intervals
- Generate graphical output and other reports.

4.4 Analysis and Reporting Tools

There are a few tools which operate on the data store to generate various types of graphical and textual reports, as well as an interactive Web-based tool which allows exploration of the detailed (aggregated) accounting data.

4.4.1 Graphical Representation

For an overview of traffic flowing over a given external link, we generate hourly plots where traffic of different application protocols is represented in different colors. Here are a few examples:

FIGURE 2. 24-hour Plot of US Traffic

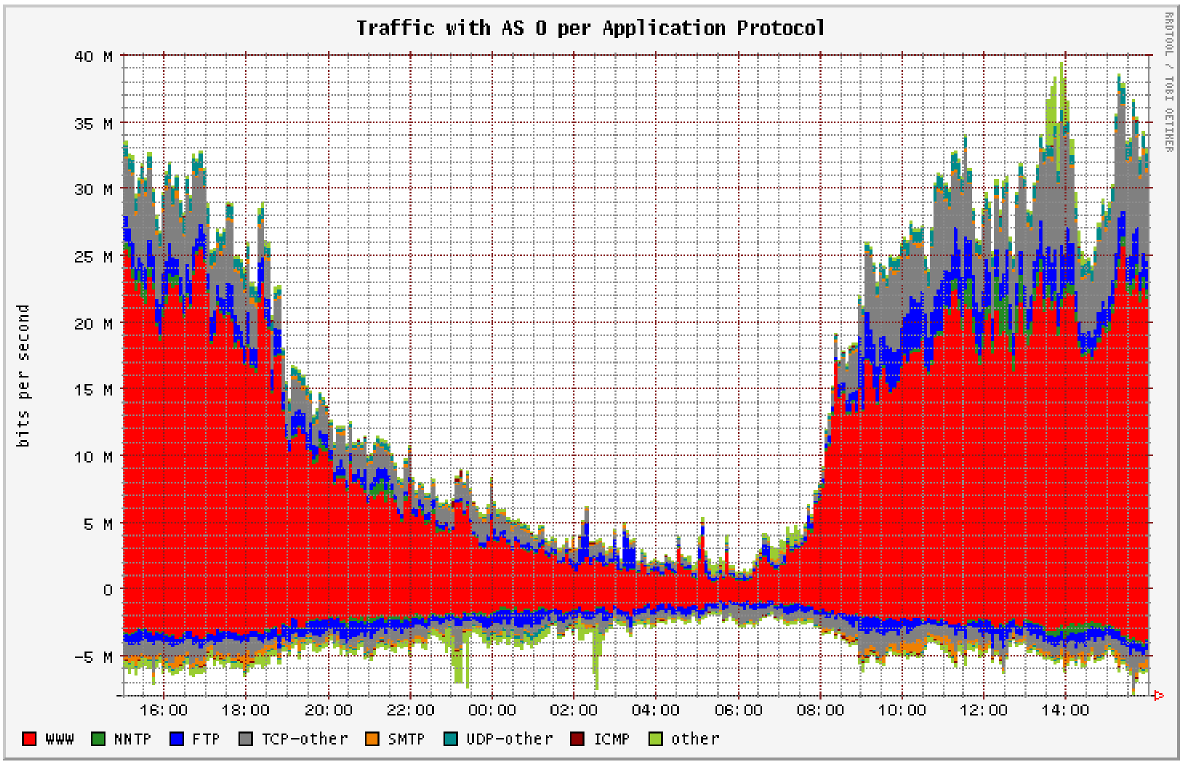
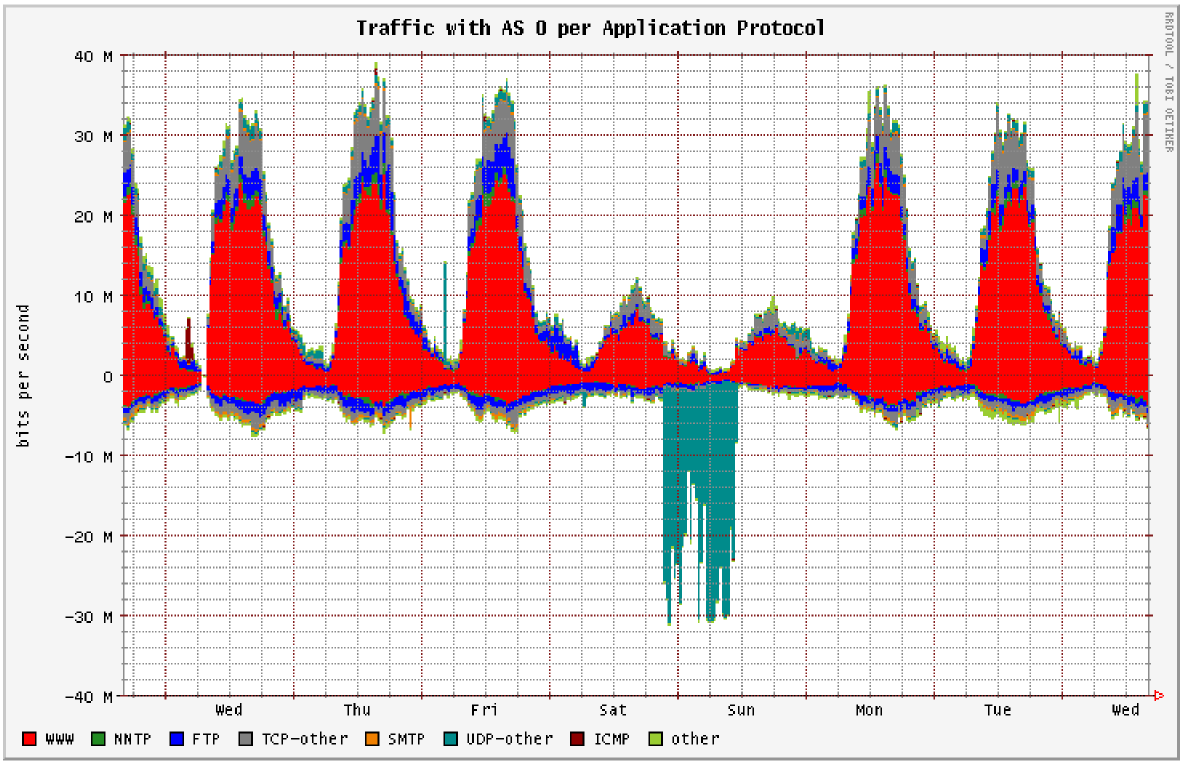


Figure 2 shows the protocol distribution on SWITCH's transatlantic line over the period of (slightly more than) twenty-four hours. The traffic pattern is relatively smooth due to the high level of aggregation, i.e. it represents thousands of users using the connection, each at relatively low rate. Around 14.00, close to the rightmost edge of the graph, there is an unusual dip in traffic volume, which may be the result of a routing problem near our commercial Internet access in New York. Traffic is largely dominated by WWW traffic. Since the link has recently been updated, and this is during term break, there is no congestion on the link.

Note that the "positive" (upper) side of the graph represents incoming traffic, i.e. traffic from the US to SWITCH, and the "negative" side corresponds to outgoing traffic, i.e. traffic towards the US. On this transatlantic link, the traffic is highly imbalanced, with incoming traffic exceeding outgoing traffic by a large factor except for late night and early morning (European time).

FIGURE 3. 8-day Plot of US Traffic



In Figure 3, the traffic pattern for the same link is shown over a period of an entire week. The most notable anomaly is the large amount of outbound "other UDP" traffic between Saturday night and Sunday around noon. This has been part of a denial-of-service attack performed through a server at a university which was broken into. If this peak is ignored, the graph nicely shows the typical weekday/weekend pattern.

5. Conclusions and Outlook

Flow-based accounting mechanisms such as RTFM, LFAP or Cisco NetFlow provide valuable input for the better understanding of network traffic on a large scale. It can also be used to support differentiated usage-based charging schemes. Recently, commercial billing and customer care applications are becoming available that can process these types of accounting data.

The in-house development of Fluxoscope has given us much insight into the possibilities and problems associated with flow-based accounting, as well as a highly extensible platform for experimenting with new applications. The largest part of the system is written in Common Lisp. This had several advantages, including easy modification during development, native support for large integer arithmetic, and implicit storage management ("Garbage Collection"), which have contributed to making the system powerful and stable. On the other hand it creates a support problem because Lisp isn’t familiar to most people in the organization. To ensure support for the “mission critical” part of the system, namely the generation of those data which are necessary for accounting, a subset of Fluxoscope’s functionality will be reimplemented in a more “mainstream” programming language such as Java.

An open problem remains the distribution of detailed accounting data for organizations which want to break up costs internally. Out of the several possible solutions, we are investigating a system where NetFlow accounting records relevant to a particular connected organization are forwarded to a postprocessor at that site. One advantage of this would be that each organization can use whatever software they like to generate accounting and statistics reports (some already have their own custom developments).

6. References

[1] *NetFlow Services and Applications*, Cisco White Paper
http://www.cisco.com/warp/public/cc/cisco/mkt/ios/netflow/tech/napps_wp.htm

[2] Daniel W. McRobb, *Cflowd Design*, September 1998
http://www.caida.org/Tools/cflowd/design/design.ps
See also the CFLOWD site: http://www.caida.org/Tools/cflowd/

[3] K. McCloghrie, M. Rose, *Management Information Base for Network Management of TCP/IP-based internets: MIB-II*, March 1991
http://sunsite.cnlab-switch.ch/ftp/doc/standard/rfc/12xx/1213

[4] T. Oetiker, *MRTG (Multi Router Traffic Grapher) Web site*
http://ee-staff.ethz.ch/~oetiker/webtools/mrtg/mrtg.html

[5] Y. Rekhter, T. Li (eds.), *A Border Gateway Protocol 4 (BGP-4),* RFC 1771
http://sunsite.cnlab-switch.ch/ftp/doc/standard/rfc/17xx/1771

[6] T. Bates, E. Gerich, L. Joncheray, J.-M. Jouanigot, D. Karrenberg, M. Terpstra, J. Yu, *Representation of IP Routing Policies in a Routing Registry* (RIPE-181), October 1994
http://www.ripe.net/ripe/docs/ripe-181.html