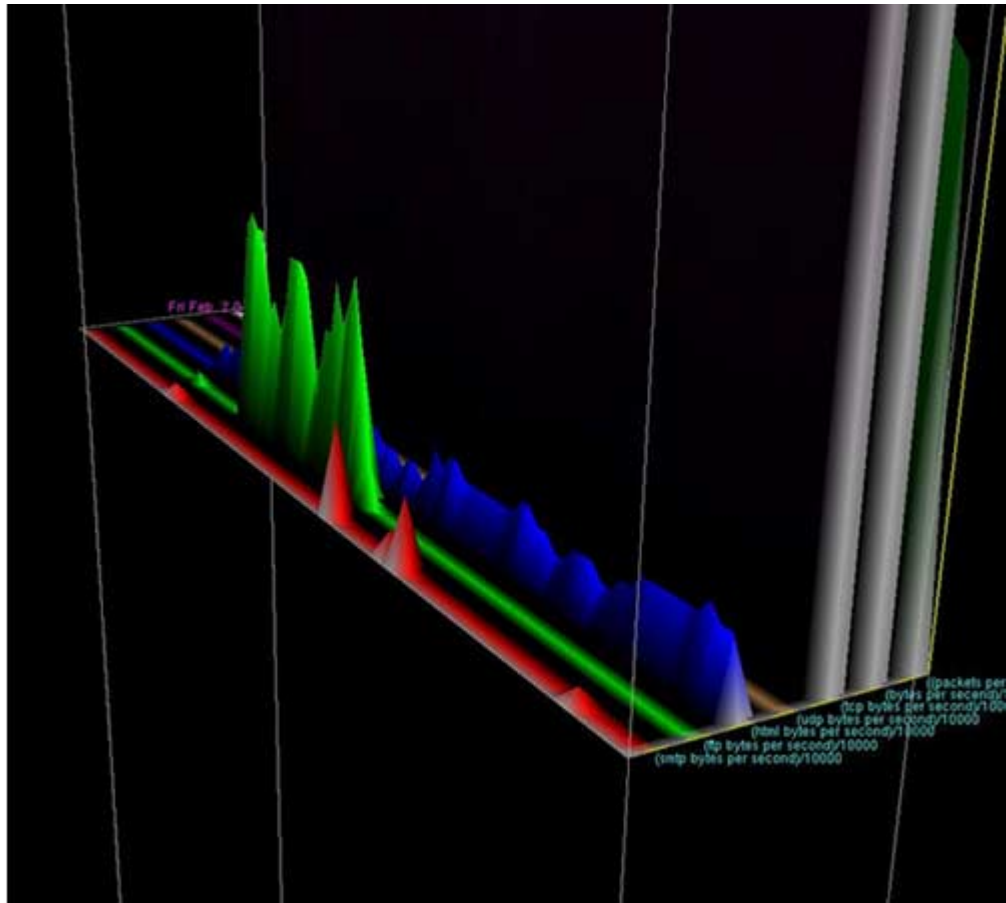


# Network Analysis Times

Vol. 2 (1)

April 2001

This image is a Cichlid 3-D visualization system generated graph showing the analysis of a trace file. Trace files are 1.5 minutes to 2 minutes in length (time, x-axis). The analysis shows the network statistics for the trace file; each metric is shown per second/10000. From left to right on the y-axis: smtp bytes [red], ftp bytes [green], html bytes [blue], udp bytes [brown], tcp bytes [1st gray]; the number of bytes and packets form the last two gray rows, respectively. (Image created by Justin Fields.)



## *In this issue:*

- **The Internet Trace User Community:**

**An invitation to network researchers and trace users** (p. 3)

- **Pings from around the world - reports on Network Analysis projects from some of our collaborators**

A Network Engineer's View of the Active Measurement Project (AMP) (p. 2)  
Bill Owens, New York State Education and Research Network (NYSERNET)

The next generation in Dag cards: Dag4 - an OC48c monitor (p. 4)  
Ian Graham, David Miller, and Jörg Micheel, WAND research group, University of Waikato

GenSyn - Generator of synthetic Internet traffic (p. 5)  
Poul E. Heegaard and Brynjar Å. Viken, Telenor R&D, Trondheim, Norway

Traffic on the Internet - a study of Internet games (p. 6)  
Sarah K. Joyce, Computing and Mathematical Sciences student, The University of Waikato

An Analysis of Napster and Other IP Flow Sizes (p. 8)  
Dave Plonka, University of Wisconsin - Madison



To subscribe, or if you have comments and/or questions, please contact us: [natimes@nlanr.net](mailto:natimes@nlanr.net)

© 2000 The NLANR Measurement and Network Analysis Group, located at the San Diego Supercomputer Center (SDSC), University of California, San Diego (UCSD). This work is supported by the National Science Foundation (NSF) (cooperative agreement no. ANI-9807479). Any opinions, findings and conclusions or recommendations expressed in this publication are those of the author(s), and do not necessarily reflect the views of the NSF.

## A Network Engineer's View of the Active Measurement Project (AMP)

**Researcher: Bill Owens, New York State Education and Research Network (NYSERNET)**

It is the call every Network Engineer dreads: the application owner who is trying to make something wonderful and new, work with his peer at a distant university, and cannot. Worse still, it worked last week but does not any more. The application is one you have never used, and cannot install on your own machine for testing. Nor does it give you any idea of what is broken.

Sound familiar? The usual response is to try some pings, but they probably will not tell you anything. The path is fine for pings, and probably HTTP, telnet, etc., but it does not support this particular application. Traceroutes would be great, but the researcher on the other end does not know how to do them, and you do not have any contacts with that school's technical staff. Also, *traceroute* and *ping* from different platforms can give widely varying results. In addition, neither can tell you what changed last week to break things. You may not know of anything you did on your own network, but maybe the other end changed something, or a change happened at one of the backbones, or your GigaPop. From there, the troubleshooting usually gets uglier and more difficult until you dig down to uncover the real problem, often hours and many phone calls later.

There is a better way: instrument the network so that you are constantly running traffic and path tests across all the possible connections you have to other schools. Impossible, you say? It would be, if someone had not already done it for you. The NLANR Active Measurement Project (AMP) has been working on just this problem since 1998, and has placed active measurement monitors at 118 sites, including universities and GigaPops all over the U.S. The AMP machines continuously *ping* and *traceroute* to each other (over 10,000 pairs of machines), and report the results back to the project servers. There, the data are digested into graphical and tabular reports, summarized, sliced, diced and presented for your use.

How can AMP help with the broken application problem from the first paragraph? First, of course, you have to have an AMP machine at your location. Head over to the AMP site (see resources to the left) to find out how to get one (or, best yet, discover that you already have one!). Odds are pretty good that the university on the other end of the connection has an AMP server, or, someone else on the same GigaPop does.

Start at the Web Interface page, linked from the main AMP page. Your server has a separate table of all of the other sites with which it runs tests. Go to that page, then select the other institution from the list, you will be rewarded with a summary graph of round-trip times (RTTs) and jitter, along with access to detailed information for each week that the servers have been collecting data.

Want to know how things looked last week, when the application was working? You can probably see any changes that took place by examining the RTT graphs; drill down into the daily graphs if you want more detail, and check out packet loss data. Traceroute pages are available for each day, or you can examine other combinations of hosts and time periods. Each traceroute page links to the reverse path, so you can make a quick check of how things look from the other end.

If you have a path that has never worked, do not give up hope; AMP can still tell you a great deal about the network. Do you see daily increases in delay during peak traffic periods? If so, there is probably an overloaded network component in the path. How does that path compare to other sites on the same GigaPop? Or, with others in the same area of the country? Perhaps there is something funny about your path to Abilene or vBNS, or you might be using one of the 'fednets' without even knowing it. Or, you could be taking the commodity Internet in one or both directions.

Once the troublesome application is working, take time to explore the other ways of viewing AMP data, such as the Cichlid 3-D visualization system and the Otter traceroute visualization tool. And get in the habit of reviewing your own listing of sites, so you can develop a feel for which ones tend towards longer or shorter delay, higher packet loss, etc. You can find many problems by just keeping an eye on those results, and are ahead of the game if, and when, a complaint comes in.

The idea of using AMP to troubleshoot the network is far from new, and although it is not the project's primary purpose, the folks at NLANR have written a very thorough and useful paper on diagnosing all sorts of network problems using AMP data. The AMP Case Studies paper is available on the NLANR Web site (see resources to the left). Hopefully, you will not run into most of the problems they document, but I think that you will soon find that AMP is an indispensable part of your troubleshooting toolkit.



## Internet Trace User Community

**Researchers: Hans-Werner Braun, Principal Investigator, and Jörg Micheel, PMA team leader, NLANR Measurement and Network Analysis Group**

The Passive Measurement and Analysis (PMA) team of the National Laboratory for Applied Network Research's (NLANR's) Measurement and Network Analysis Group is creating an *Internet Trace User Community*, and invites Internet researchers to discuss the next generation of passive network measurements to be carried out on the High Performance (HPC) backbone networks.

Since 1995, the Measurement and Network Analysis Group has been collecting IP packet header traces to support research into understanding the systemic nature of the Internet. The 12 sites chosen for capturing traffic traces are located at high-bandwidth interconnection points that typically access links from GigaPops to the vBNS core. The measurement strategy was to capture samples eight times a day for a defined length of time. The initial sampling interval was 2 minutes. This was changed in order to limit the amount of data captured each day, the data is now captured for a length of 90 seconds (eight times a day).

Today, the system collects between 1.5 and 3.2 Gigabytes (GB) of compressed data per day. The Network Analysis Infrastructure (NAI) maintained by the group is designed as a service to the research community at large, and allows for WWW and ftp access to all data collected, (after anonymization procedures). The *Data cube* is an interface to search and browse trace data for specific metrics without having to download the large trace data files for detailed analysis. Please see resources box for more information.

Initially, the data collection systems (OC3mon) were based on inexpensive commodity hardware (PC's with FORE ATM NICs). Experience with those systems has highlighted some of the shortcomings in deploying standard network interface cards. The NLANR Measurement and Analysis Group has been vital in supporting the development of dedicated passive network measurement gear. As a result, recent monitors in the PMA infrastructure support high-precision time stamping, synchronization of both cards for bidirectional capturing, OC3c and OC12c links with ATM and PoS encapsulation, and the capability to synchronize to an external clock source, such as a GPS or CDMA time receiver.

The PMA team is currently placing new passive monitors - approximately 12 remain out of a set of 25 - at important points on the Internet2/Abilene network. In order to develop better monitor placement strategies and trace schedules, we believe that it has become crucial to understand the types of analyses that are being done with the captured data. With the previous set of monitors, the placement strategy was designed to provide good coverage of the overall network; therefore, each monitor captures a unique portion of the overall network data. With the new set monitors available, a more dense instrumentation of the network has become feasible. This means that the same traffic flow can be observed from multiple measurement points and that the distortion of the traffic pattern can be studied. Determining correlations between data captured at one point in the network with data captured at a different point should become possible. We are looking into providing different kinds of studies, such as long traces (hours, even days, or weeks). We are also planning to provide more detailed postprocessing (different sets of graphs) along with the published traces. At the same time, the group is seeking to reduce the amount of management overhead for maintaining the monitors and the data collection postprocessing. This implies changes to the trace schedules in order to (somehow) balance the amount of data collected.

### We are seeking your constructive discussion on the following topics:

- focus of your research in the area of passive measurement analysis
- and consequently:
  - monitor placement strategies
  - trace durations and schedules
  - trace postprocessing and WWW publishing
  - trace scenarios (router instrumentation, cross-U.S., transatlantic, etc. ...)
  - trace variety (LAN views, WAN access view, backbone view)
  - any other passive measurement topics that you find appropriate, and of interest

### For more information:

Please send your contributions (ideas, questions, discussion topics) to the Internet Trace User Community mailing list at: [traces@nlanr.net](mailto:traces@nlanr.net).

Please send subscription requests for this mailing list to [traces-request@nlanr.net](mailto:traces-request@nlanr.net).

Internet Trace User Community archive:  
<http://moat.nlanr.net/PMA/Traces/archive>

WWW access:  
<http://moat.nlanr.net/Traces/Traces/>

Data cube:  
<http://moat.nlanr.net/PMA/Datacube.htm>

---

Your contributions should be sent to the Internet Trace User Community mailing list (see resource box for address).

We are looking forward to hearing from you. On behalf of the NLANR PMA team, we appreciate your input.

## Pings from around the world - reports on network analysis projects from some of our collaborators

---

### The next generation in Dag cards: Dag4 - an OC48c monitor

**Researchers: Prof. Ian Graham, David Miller, and Jörg Micheel, WAND research group, Computer Science Department, University of Waikato (Hamilton, New Zealand)**

The Dag4 is a PCI network interface card designed for passive network monitoring at OC48c network links (2.5 Gb/sec), such as those deployed in today's high bandwidth Internet backbones.

In general, the architecture of the Dag4 card follows design principles utilized in the Dag3 architecture, a predecessor designed for lower speed network links (OC3c and OC12c). However, capturing network traffic at four times the bandwidth of an OC12c link requires significantly greater resources in order to reliably deliver network measurement data.

The development of the Dag4 series was challenging due to the shortage of commercially available components operating at the required data rate. An initial prototype (4.0) supporting OC48c ATM cell capture was produced by March 2000. However, today's Internet backbones mainly deploy Packet-over-SONET (PoS) encapsulation, rather than ATM. In the new Dag4.1 design, the enhanced physical layer interface chip, Vitesse VSC9112, is being used, which makes the board capable of both ATM and PoS data capture.

The card is capable of delivering data at OC48c line rate by utilizing the PCI bus in 64-bit mode at 66 MHz; therefore, it requires a modern PC-based workstation, or server motherboard. The card has been designed with sufficient resources to allow for onboard processing of the captured data stream, offloading as much CPU processing as possible off the host. There are three field programmable gate arrays (FPGAs) on the board. The first FPGA handles packet arrival time-stamping, utilizing the Dag universal clock kit (DUCK) technology. An onboard FIFO smooths short bursts of short packets and allows for PCI bus contention. A DSP FPGA is designed to allow for arbitrary packet processing; an additional 2 MB of RAM will support the work of the DSP. The third FPGA deals with PCI bus mastering.



**A Dag 4.1 card -  
designed for passive network  
monitoring at 2.5 Gb/sec  
(OC48c network links).**

---

**For more information:**

**The Dag4 Series:**

<http://dag.cs.waikato.ac.nz/dag/dag4-arch.html>"

**The Dag Project:** <http://dag.cs.waikato.ac.nz/>

**Dag-news archives:**

<http://dag.cs.waikato.ac.nz/dag/dag-news/>

**Waikato Applied Network Dynamics (WAND) group:** <http://wand.cs.waikato.ac.nz/>

The Cooperative Association for Internet Data Analysis (CAIDA): <http://www.caida.org/>

**Ian Graham:** [ian\(at\)cs.waikato.ac.nz](mailto:ian(at)cs.waikato.ac.nz)

---

Project leader Ian Graham, Dag4 designer David Miller, and Dag software developer Jörg Micheel visited CAIDA in early November with the goal of testing the Dag4 in a real network environment. The test setup involved a Cisco GSR 12000 router and a Juniper M20 at the San Diego Supercomputer Center (SDSC). The team was able to demonstrate network capturing on the OC48c link carrying a regular Internet traffic mix. The tests showed the high-precision timestamping (DUCK) - an unique feature of the Dag network measurement cards - to work properly. The data was checked for content integrity via checksums and no errors were discovered. Distributions of packet sizes were found to be identical to previous measurements on the same link, confirming the proper operation of the card.

Testing and development of the card continued through the rest of the year 2000. The Waikato Applied Network Dynamics (WAND) group continues to work with interested partners operating OC48c IP backbones. A fully working Dag4.11 card is expected to be available early in the first quarter of 2001.

---

The Dag 4 series project is a subcontract through The Cooperative Association for Internet Data Analysis (CAIDA), located at the University of California, San Diego's San Diego Supercomputer Center (SDSC), to the University of Waikato, funded by the Defense Advanced Research Projects Agency (DARPA).

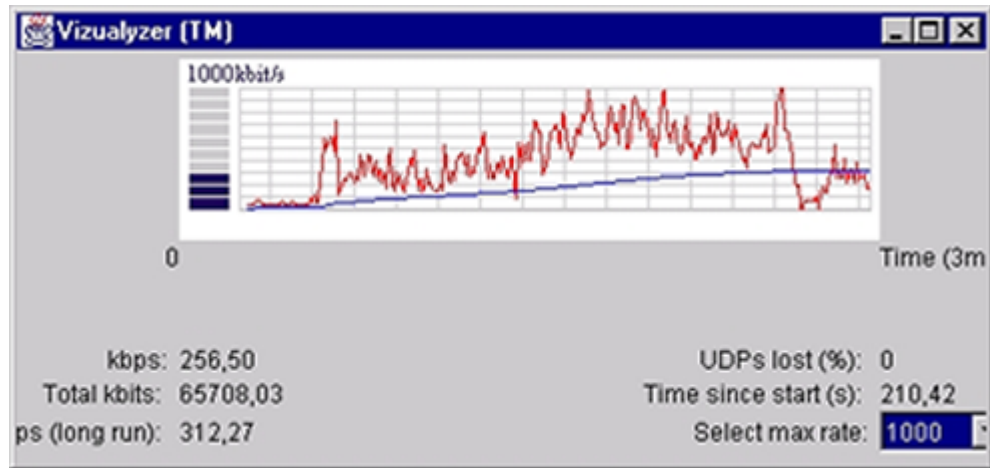
# GenSyn - Generator of synthetic Internet traffic

**Researchers:** Poul E. Heegaard and Brynjar Å. Viken, Telenor R&D, Trondheim, Norway

For the purpose of Quality of Service (QoS) testing of new applications and network mechanisms in the Internet, a generator of controllable, scalable, synthetic, but realistic, IP traffic is required. This has been the motivation for the development of GenSyn - a generator of synthetic Internet traffic implemented in Java.

**GenSyn in brief** - GenSyn provides a flexible and scalable modeling framework. The stochastic user behavior is described by state diagrams. The model is scalable because it allows a composition of users in each state, instead of creating a new instance of the process for every user. The stochastic user behavior model controls the creation of TCP connections and UDP streams through interface modules that link the GenSyn process to the underlying Internet protocol stack on the workstation. This means that on transitions between specific states in the stochastic model, an interface process will be initiated and IP packets are sent and received through the network. For more details, see references [1] and [2] and Figures 1 and 2 (see note regarding Figure 2).

**Using GenSyn in QoS testing** - GenSyn is currently being used for QoS performance testing in an experimental, IP based, communication platform that will provide differentiated services. To test network QoS mechanisms, a controllable, and reproducible, mixture of traffic streams with different characteristics is essential. This traffic mixture is generated by GenSyn using the source models that are currently available and described by the framework of GenSyn. This includes models of Web and ftp clients that generate TCP traffic by downloading pages and files from actual Web servers, and models that generate UDP traffic from a video server (using MPEG), from voice over IP (VoIP), and in a Constant Bit Rate (CBR) stream.



**Figure 1:** Sample of traffic trace generated by GenSyn. The trace monitor in GenSyn shows the number of bytes for the last second and accumulated over time.

**Figure 2** (not included here due to size constraints, please see the on-line version to view): Data samples collected while GenSyn is running - plots of throughput peer-to-peer for three different time accumulation granularities.

**GenSyn combined with passive measurements** - Very little measurement functionality is included in the GenSyn. Therefore, in the QoS performance experiments discussed in the previous section, the measurements are based on dedicated monitors with interface boards specialized to capture packet traces. The clocks of these interface boards are synchronized by GPS receivers and have a very high resolution. This instrumentation enables accurate measurements of performance metrics like throughput, unidirectional delay and loss.

## For more information:

To obtain a free, noncommercial, license for the GenSyn traffic generator, visit the GenSyn Web site:

<http://www.item.ntnu.no/~poulh/GenSyn/gensyn.html>

Or, contact:  
Telenor AS, Telenor R&D  
GenSyn  
Otto Nielsensvei 12  
7004 Trondheim, Norway  
Phone: +47 7354 3845 (Poul E. Heegaard)  
Fax: +47 7354 3700  
email: [gensyn@edeber.nta.no](mailto:gensyn@edeber.nta.no)

Given the information contained in the IP header, it is necessary to export the packet traces collected at various measurement points to a central host in order to compute unidirectional performance metrics such as packet delay and packet loss. The delay of a packet from the ingress measurement point to the egress measurement point is computed by finding the same packet in the traces collected at both measurement points; this is illustrated in Figure 3.

**Scenario Designer** - There are a variety of additional ways that GenSyn could be used; for example, in a testbed where generation of traffic to several access points (routers) is required, and end-to-end properties will be examined. Typically in these cases, GenSyn will be running on dedicated machines distributed in the network being tested; in addition, specific measurement probes have to be activated. Setting up and managing such a distributed experiment is not a trivial task. Therefore, a system to assist and support analysts in setting up experiments, collecting and processing measurement data is essential. For this purpose, a scenario designer is currently in development. An example of a scenario outline as viewed in the planned GenSyn Designer can be seen in Figure 4, below.

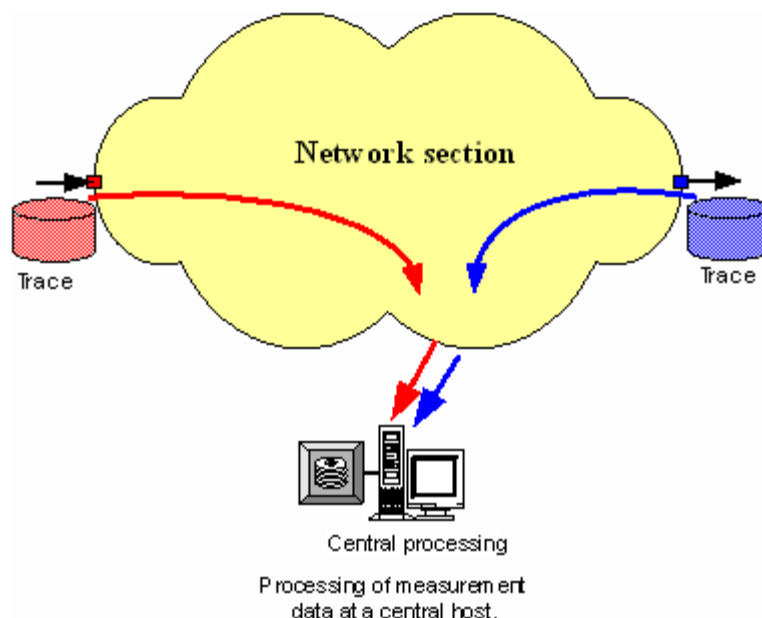


Figure 3: End-to-end measurements

## References

[1] Poul E. Heegaard. GenSyn - a Java based generator of synthetic Internet traffic linking user behavior models to real network protocols. Presented at ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management, September 18-20, 2000, Monterey, CA, USA.

[2] Poul E. Heegaard. GenSyn - a generator of synthetic Internet traffic used in QoS experiments. Presented at 15th Nordic Teletraffic Seminar, August 22-24, 2000, Lund, Sweden .

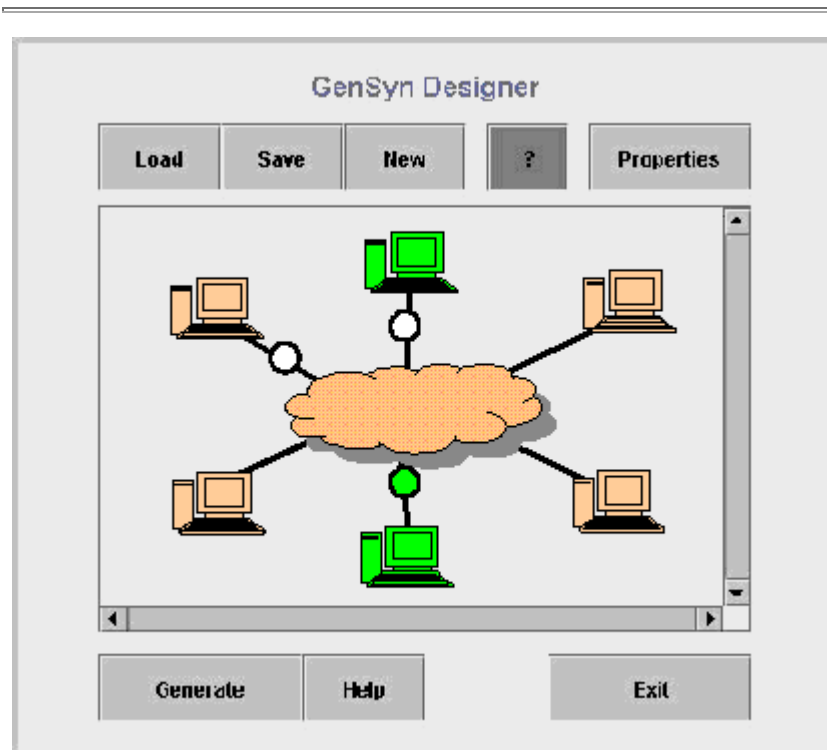


Figure 4: Schematic outline of a test scenario, as viewed in the GenSyn Designer (in development).

## Traffic on the Internet - a study of Internet games

**Researcher:** Sarah K. Joyce, Computing and Mathematical Sciences student, The University of Waikato (Hamilton, New Zealand)

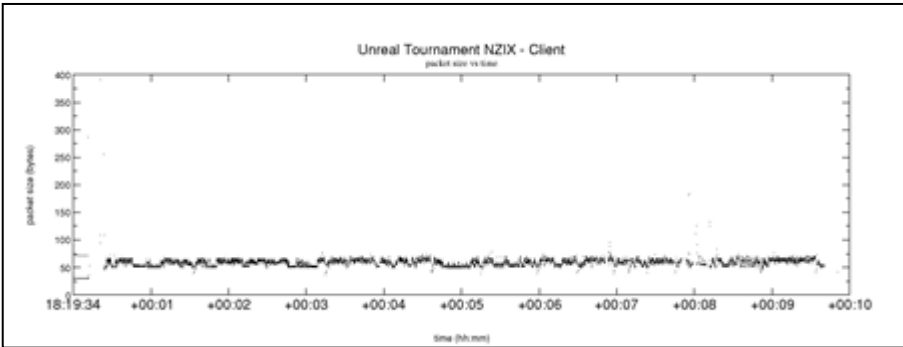
Game traffic on the Internet is of interest because games typically use UDP as the underlying transport protocol and this traffic is increasing in volume. UDP traffic differs from the majority of Internet traffic which is TCP based. However, there is concern that the

current increase in UDP traffic will have a negative effect on TCP throughput.

For this study, models of games traffic on the Internet were constructed from data collected at two passive measurement sites within New Zealand: Auckland II (at the University of Auckland) and NZIX, the New Zealand Internet Exchange, which is a neutral peering point for ISPs within New Zealand, (hosted by the University of Waikato).

Two games were examined in detail: Quake World and Unreal Tournament. Default port numbers are used to identify particular Internet games. For instance, UDP port 7777 identifies a server for Unreal Tournament; Quake World servers use UDP port 27500. Therefore, the traffic sent by the server and to the server (by the client) can be distinguished from one another.

Passive traces taken at the two measurement sites were analyzed for the presence of games. The analysis was carried out using a number of heuristics, including port numbers and IP addresses. Game sessions were found to last for about 20 minutes typically; few lasted for more than 90 minutes. Traffic originating from the servers was found to vary greatly in packet size; by comparison, the client sends fairly small packets. (Please see graphs below showing client side and server side packet size vs. time; for larger graphs, please see the on-line version of this article.)



Packet size (in bytes) vs. time (hh:mm), Client side, Unreal Tournament NZIX.

When a game had been extracted, characteristics such as packet size and timestamp were filtered and graphed. Plots of different game sessions were produced for analysis. Packet size vs. time was of particular interest in this study, however, inter-arrival times were also investigated.

A number of factors caused difficulties in the study: significant differences in the two collection sites (academic vs. commercial, amount of bandwidth, and volume of games traffic); the two games examined in detail displayed different traffic patterns; and the large volume of data generated.

Despite these hurdles, two types of games models have been implemented using the data gathered. The first is based on the raw trace data, which was filtered into a suitable format to be processed by a simulator. The second analyzed a subset of the data from each game, producing empirical distributions (packet length and the time between each packet). Each distribution was then used to predict a set of game data.

The results of the study suggest that games do impact the throughput of TCP on a heavily loaded link. TCP implements a form of flow control to prevent flooding on the network, whereas UDP simply continues to send packets at a fixed rate.

The simulations show that any type of UDP traffic will limit the amount of TCP traffic that can move through the network. The effect UDP has on a link can be likened to putting a “cap” or

**For more information:**

Traffic on the Internet - A study of Internet games:  
<http://wand.cs.waikato.ac.nz/wand/publications/sarah-420.pdf>

The Auckland Data Set (paper):  
<http://wand.cs.waikato.ac.nz/wand/publications/barcelona-2001.pdf>

Auckland II Trace Data: <http://wand.cs.waikato.ac.nz/wand/wits/auck/2/>

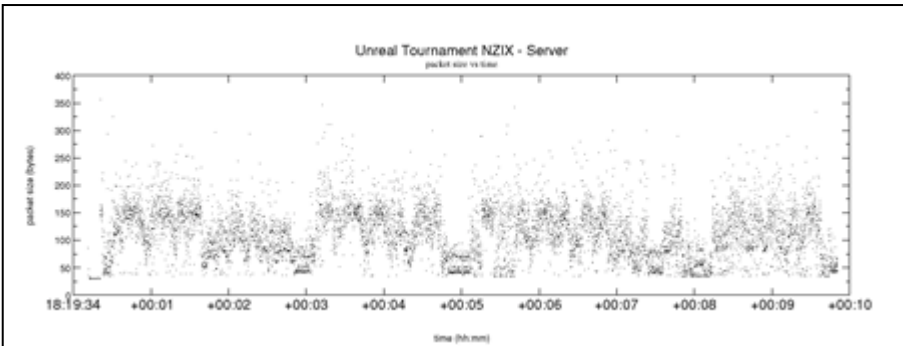
Auckland II Trace Archive: <http://moat.nlanr.net/Traces/Kiwitraces>

Waikato Applied Network Dynamics (WAND) Publications:  
<http://wand.cs.waikato.ac.nz/wand/publications/>

The New Zealand Internet Exchange (NZIX): <http://wand.cs.waikato.ac.nz/wand/wits/nzix/2/>

Analysis of Voice Over IP Traffic (paper):  
[http://wand.cs.waikato.ac.nz/wand/publications/jamie\\_420/final/](http://wand.cs.waikato.ac.nz/wand/publications/jamie_420/final/)

**John G. Cleary: [jcleary\(at\)cs.waikato.ac.nz](mailto:jcleary(at)cs.waikato.ac.nz)**



Packet size (in bytes) vs. time (hh:mm), Server side, Unreal Tournament NZIX.

limit on the network. Games affect TCP traffic in much the same way as Voice over IP does. This suggests that the underlying UDP protocol behaves in much the same manner, regardless of the application that is running it. UDP traffic is considered aggressive to network friendly applications deploying adaptive congestion control.

*The full report was submitted in partial fulfillment of the requirements for the degree Bachelor of Computing and Mathematical Sciences at The University of Waikato, October 11, 2000. Supervisor: Prof. John G. Cleary.*

---

This is an abridged version of the current issue of the *Network Analysis Times*. Full size images for all articles can be viewed on this issue's Web pages. The issue, in its entirety, can be found at: <http://moat.nlanr.net/NATimes/NAT.2.1/>.

Due to space limitations related to the publication of the printed version of this issue, a number of articles available on-line are not present here. These include: "Study of routing behavior through traffic analysis and traceroute measurements" by Pablo Molinero-Fernández and Nick McKeown of Stanford University and an overview article of the Measurement and Network Analysis Group's current passive measurement and analysis (PMA) data tools.

The *Network Analysis Times* is a newsletter which we use for sharing information, discussing issues, and creating new collaborations in the area of measurement and network analysis research. As such, we invite you to contact us regarding the submission of relevant articles for possible publication in future issues. Please contact us at: [natimes@nlanr.net](mailto:natimes@nlanr.net).

Maureen C. Curran, Editor

---

## **An Analysis of Napster and Other IP Flow Sizes**

**Researcher:** Dave Plonka, University of Wisconsin - Madison

**Introduction** - This is a report on an informal investigation of IP flow sizes that was initiated in response to questions posed in July, 2000, paraphrased as follows:

1. Is there a peak in the distribution of flow sizes which approximates the size of a typical MP3 file?
2. Do flows of Napster traffic exhibit a characteristic signature in terms of the sizes of its IP flows?
3. While the use of "sharing" applications such as Napster are increasing bandwidth used, is it also changing the typical size of IP flows?

We also wondered if flow size analysis might provide a useful indication of general trends in Internet workload.

**Our Flow Definition** - The way in which IP flows are defined varies amongst investigators and vendors. This work utilized Cisco NetFlow export data collected by the Cooperative Association for Internet Data Analysis's (CAIDA's) *cflowd* and post-processed by *FlowScan* [cflowd, Plonka 2000a]. Therefore, our IP flow is defined by the NetFlow V5 flow export PDU [NetFlow]. In short, a "flow" is a unidirectional series of IP packets of a given protocol, between a source and destination port, within a certain duration. The tunable NetFlow "timeout active" value was set to one minute. This means that active flows were expired and exported in as little as one minute after being instantiated.

**A Look at Average Flow Sizes** - During the 1999 and 2000 academic years, the average IP flow size appears to have increased slightly. This is evidenced by time-series plots of average flow size such as Figure 1 (on back cover), shown below. Furthermore, the average size of Napster flows is significantly larger than that of IP flows in general. Note that because measurement of Napster traffic at the University of Wisconsin - Madison began in March of 2000, no earlier Napster flow data is shown in Figure 1 (see back cover).

Although Napster flows are larger than average IP flows, other popular well-known applications such as ftp-data transfers have average flow sizes that far exceed those of Napster. While not included here, similar plots show the average ftp-data flow usually contains more than 200 kilobytes.

**About the Flow Samples** - This analysis uses raw flow data collected at the border of the campus network at the University of Wisconsin - Madison. This data was collected in three 24 hour periods, each sample roughly representing the traffic during one day of the Fall 1999, Spring 2000, and Fall 2000 semesters of the academic calendar. Table 1 summarizes the average rate of inbound and outbound IP traffic for the campus as a whole during each 24 hour sample. The rate of Napster traffic is unknown for the Fall 1999 sample because it predates the implementation of Napster flow identification in FlowScan.

**Table 1.**

Semester	Sample "Day"	Total			Napster		
		Flows	Inbound	Outbound	Flows	Inbound	Outbound
Fall 1999	September 15-16	56,625,942	26 Mb/s	45 Mb/s	unknown	unknown	unknown
Spring 2000	May 12-13	75,315,768	45 Mb/s	73 Mb/s	7,509,938	7 Mb/s	21 Mb/s
Fall 2000	November 16	98,366,891	60 Mb/s	110 Mb/s	unknown	13 Mb/s	31 Mb/s

There was no rhyme or reason underlying our choice of these particular sample "days". The first two samples (Fall 1999 and Spring 2000) were simply the only contiguous 24 hour periods for which data was available, because we did not systematically retain detailed logs of campus traffic from so long ago. These samples were retrieved from the backup tapes of infrequent manual backups of our analysis machine. These backups were performed only for disaster recovery. As such, it was just by chance that they might coincide with interesting points in time regarding the use of "file sharing" applications. For this investigation we assume that traffic during these days is somewhat representative of traffic during each semester as a whole.

Figure 2 (back cover) shows time-series graphs of daily average outbound and inbound IP traffic rates throughout the entire range in which the samples were taken. This figure illustrates the overall growth trend of the campus traffic throughout the time in which the samples were collected. Note that the sample days are marked as red, green, and blue vertical rules in the graph. Throughout the figures herein, the colors red, green, and blue are used for each of the three samples, Fall 1999, Spring 2000, and Fall 2000, respectively.

Regarding Figure 2, at the time of the Fall 1999 sample, the amount of Napster traffic is not known, however it is generally believed to be negligible by comparison to the later samples. By the time of the Spring 2000 sample, Napster traffic represented a significant proportion of the campus traffic as a whole. (Specifically, Napster is thought to have represented 29% of our campus outbound traffic, and 15% of the inbound traffic.) Also in the Spring 2000 sample, Napster flows represented 13% of the outbound flows and 7% of the inbound flows. By Fall 2000, Napster usage was responsible for even more inbound and outbound traffic.

In table 1 we see that the Fall 1999, Spring 2000, and Fall 2000 24-hour samples contained 56, 75, and 96 million flows respectively. This increase in the number of flows is indicative of the increased Internet usage that our campus observed throughout the 1999/2000 school year. This continuous increase in data traffic has been recently investigated and reported by others [Coffman 2000].

**About the Flow Size Distribution Graphs** - In Figures 3 through 5 (back cover) we used two methods to visualize flows size distributions.

The first method, used for Figure 3, employs a line graph to plot the distribution of the cumulative percentage of total flows across 32 flow size intervals in units of packets and bytes. Subsequent size intervals are incremented by consecutive powers-of-two. That is, the first interval along the horizontal axis represents flows of sizes 1 and 2 ( $2^1$ ), the second represents 2 through 4 ( $2^2$ ), then 4 through 8 ( $2^3$ ), 8 through 16 ( $2^4$ ), and so on, up to the maximum size representable ( $2^{31}$  through  $2^{32}$ ): approximately those between 2 billion and 4 billion. This plot is similar to that used in past investigations which examined the distribution of packet sizes [MCI], except that our horizontal axis compresses 4 billion discrete sizes into a manageable set of only 32 intervals.

A second method is used for Figures 4 and 5. Like the previous, Figure 4 employs a line graph joining points plotted across the 32 size intervals. However, the percentage of total content delivered is plotted rather than cumulative percentage of total flows. Figure 5 contains histograms which plot the percentage of total bytes delivered for each size interval.

**Napster Flows** - Because this investigation looked for evidence of Napster's influence on IP flow sizes overall, it is useful to understand what sort of flows are produced by Napster. IP flows produced by the Napster application and work-alike clones are of, at least, these eight different types:

1. TCP initial connections from client user to "redirect" server
2. TCP responses from "redirect" server to client user (specifying address of an "index" server)
3. TCP commands/requests from client user to "index" server
4. TCP responses from "index" server to client user

5. ICMP ECHO from client user to candidate "server" user (28 byte packets)
6. ICMP ECHOREPLY from candidate "server" user to client user (28 byte packets)
7. TCP request from client user to "server" user (request and subsequent ACKs)
8. TCP responses from "server" user to client user (possibly containing MP3 content)

The term "NapUser" is used below to label traffic believed to be generated by an application using the Napster protocol. Unless otherwise specified in the following discussion, NapUser flows comprise all of those Napster flow types. These NapUser flows were identified by a method implemented in FlowScan [Plonka 2000b].

Throughout the figures, Napster values are plotted in purple and magenta. Figure 3 contains plots of the sizes of Napster application flows in terms of packets and bytes, both ICMP/TCP combined, and TCP alone. The "NapUser TCP bytes" and "NapUser TCP packets" plots represent just the Napster TCP flows, and therefore emphasize the flows representing the *content*-carrying flows, i.e., those representing the interaction with Napster index servers and representing the bidirectional TCP data streams which carries the MP3 data.

Figure 4 shows that Napster flow sizes, when measured in packets, peak in the 512-1000 packets interval and again in the 4000-8000 packets interval. When measured in bytes, they peak in the 512KB to 1MB range, and again in the 4MB to 8MB range. These byte measurements are roughly the product of those peak packet counts and 1500 byte MTU size commonly used for ethernet. Therefore, it is likely that those peaks are caused by the type of Napster TCP flow which carries most of the MP3 content.

From examination of the raw Napster flows we know that most Napster-related flows are actually the small ICMP flows from Napster clients to candidate servers. As such, well over half the flows produced by the application carry a trivial amount of content as measured in bytes or packets. However, the average Napster-produced TCP flow is larger than the average flow amongst all types and therefore Napster does appear to have increased the size of the average Internet IP flow.

**IP Flow Size Distributions** - Examination of Figure 3 (back cover) and the size distribution amongst IP flows of all types, there is similarity among the percentages of flows of particular sizes amongst all samples (red, green, and blue). For instance, in each sample, about half of the flows are less than 512 bytes in size. The specific numeric results and the finding that the distributions are stable over a long period of time (about a year) are similar to those reported following recent investigations of flow "lifetimes" [Brownlee 2000].

One curiosity visible when considering the cumulative percentage of flows vs. flow size is that, in Spring 2000, 5.7% of the flows were less than 32 bytes in contrast with 0.1% from the previous fall. The Fall 2000 statistic remained similar to that of Spring 2000. This is evident in Figure 3 where the solid lines, representing flow size in bytes, leave the zero value on the vertical axis. Remembering that a full 27% of the Spring traffic during the Spring sample was Napster traffic, it is likely that those small flows represent the 28-byte "ping" packets generated by Napster.

Consider Figure 4 (back cover), the distribution of flow sizes based upon the percentage of total IP content delivered, both the packet and byte plots shift slightly toward larger flow sizes as we progress from Fall 1999 (red), to Spring 2000 (green), and to Fall 2000 (blue). Also, there was an increase in the percentage of byte content delivered in flows between 4MB and 16MB in size. Specifically, the Spring 2000 sample shows a spike in those two intervals, somewhat mimicking the pure NapUser flow size distribution.

Figure 5 (back cover) shows the distribution of flow sizes in bytes, according to the percentage of the total content in bytes that were delivered by flows of that size. The median intervals have been identified and labeled. This figure seems to show that flow sizes are increasing in that a number of larger intervals (such as those between 4 and 32 megabytes) are responsible for having delivered more of the content. Since earlier figures showed that the percentage of flows of those sizes has not noticeably increased, it appears that the flows within those intervals are increasing in size, and are sometimes promoted up to the larger size intervals. This increase has resulted in the median shifting from the 2MB to 4MB interval, up into the 4MB to 8MB interval.

**Potential Problems** - In part, Cisco NetFlow defines its flows based upon timeouts. This user configurable timeout has been configured in such a way that a TCP data stream will timeout after approximately one minute regardless of whether or not the TCP stream is still active. So, a single TCP stream may be, and often is, represented by more than one flow in each direction. For this reason it is somewhat difficult to correlate TCP streams with our flow-based measurements.

Additionally, if the offered load of IP traffic does not change, but the usable bandwidth available to the users increases, one would expect to see an increase in flow size simply because bulk data transfer applications should be able to transfer more content before the flow expires. We have not yet looked for trends in this *flow rate*, so we can't necessarily enumerate all the factors contributing to the increases in flow size.

**Summary** - Napster flows are clearly larger than average, even when its numerous but often overlooked "ping" flows participate in the calculation. Furthermore, Napster flows appear to follow a characteristic pattern - namely that most Napster content is delivered in flows of sizes between 4 megabytes and 16 megabytes. This is perhaps not surprising since it is a reasonable estimate of the range of the MP3 files typically exchanged.

In response to the questions posed at the beginning of this investigation, we can respond thusly:

1. For IP flows of approximately 1 minute and lesser duration, there is a peak in the distribution of general IP flow sizes that matches a peak in Napster flow sizes. This peak shows that over 30% of the Internet traffic (in bytes) is transferred by flows of sizes between 4MB and 16MB.
2. The cumulative Napster flow size distribution exhibits a signature quite distinct from the cumulative flow size distribution for general IP flows.
3. During the time that the bandwidth utilized by Napster and other "sharing" applications has dramatically increased, the size of flows that carry the largest percentage of Internet traffic has subtly increased as well.

Considering these results, it seems likely that use of the popular file-sharing applications such as Napster will not only continue to increase bandwidth usage solely by virtue of their popularity, but will also likely shift the distribution of flow sizes higher. Operational implications of such a shift in Internet workload and usage characteristics warrants further study.

**Future Directions** - It is possible that the increasing amount of Internet content being transferred by large flows is due to an increase in the effective throughput observed by applications. If Internet bulk data transfers are enjoying an increasing bit transfer rate, the flow "active timeout" may cause those flows to expire prior to termination of the underlying application session, thus causing a larger number of flows to accumulate within the size interval containing the product of that active timeout value and bit transfer rate. An investigation of flow durations and flow rates needs to be performed to determine if flows rates are increasing.

The time-series measurement and visualization of flow rates may be a useful feature to add to existing flow-based passive measurement systems. If the observed flow rates are found to correlate with the user's perceived application performance, we would have a new tool by which to passively measure quality of service in near real time.

**Acknowledgments** - I thank these folks for their answers *and* questions:

k claffy <kc@caida.org>  
Michael Hare <mhare@doit.wisc.edu>

## References

- J. Nevil Brownlee, "NeTraMet Flow Lifetimes and Implications for Routing Context," 2000.  
<http://www.caida.org/analysis/workload/netramet/lifetimes/CAIDA'scflowd/>  
<http://www.caida.org/tools/measurement/cflowd/>  
Cisco's IOS NetFlow feature:  
<http://www.cisco.com/warp/public/732/netflow/>  
K.G. Coffman, A. M. Odlyzko, "Internet growth: Is there a "Moore's Law" for data traffic?" 2000.  
<http://www.research.att.com/~amo/doc/networks.html>  
[MCI] Sample Packet Size Distribution:  
<http://www.caida.org/outreach/presentations/Soa9911/mgp00025.html>  
D. Plonka, "FlowScan: A Network Traffic Flow Reporting and Visualization Tool," 2000a.  
<http://net.doit.wisc.edu/~plonka/lisa/FlowScan/>  
D. Plonka, "UW-Madison Naptser Traffic Measurement," 2000b.  
<http://net.doit.wisc.edu/data/Napster/>

**Analysis Tools** - The following tools were used during this analysis:

- **cflowd**: <http://www.caida.org/tools/measurement/cflowd/>
- **RRDtool**: <http://ee-staff.ethz.ch/~oetiker/webtools/rrdtool/>
- **Cflow** perl module and **flowdumper** perl script: <http://net.doit.wisc.edu/~plonka/Cflow/>
- **FlowScan**: <http://net.doit.wisc.edu/~plonka/FlowScan/>
- **flowsize** perl script: <http://net.doit.wisc.edu/~plonka/flowsize/>
- **gnuplot**: <ftp://ftp.gnuplot.vt.edu/pub/gnuplot/>
- **Grace**: <http://plasma-gate.weizmann.ac.il/Grace/>

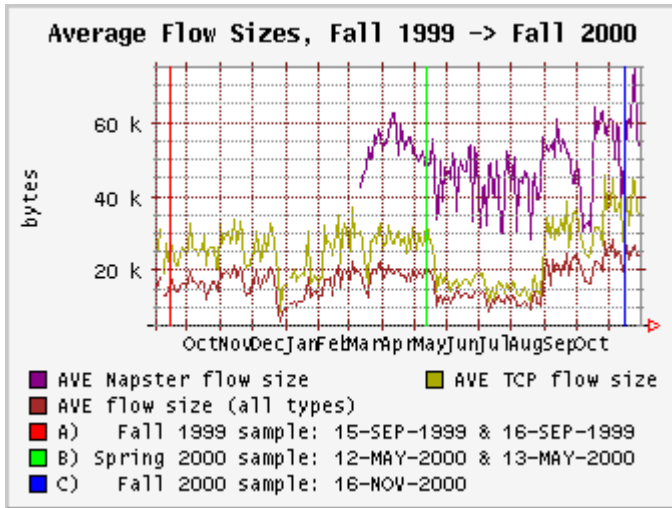


Figure 1.

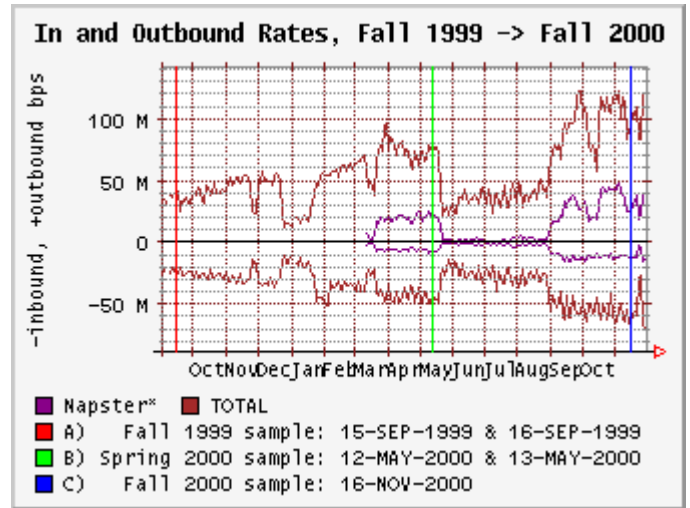


Figure 2.

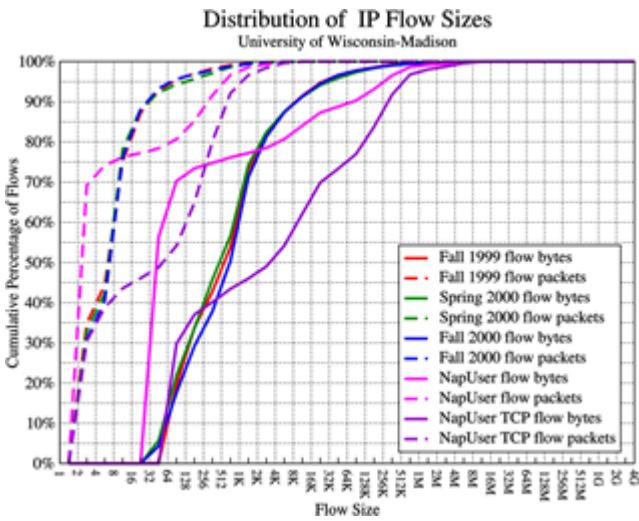


Figure 3.

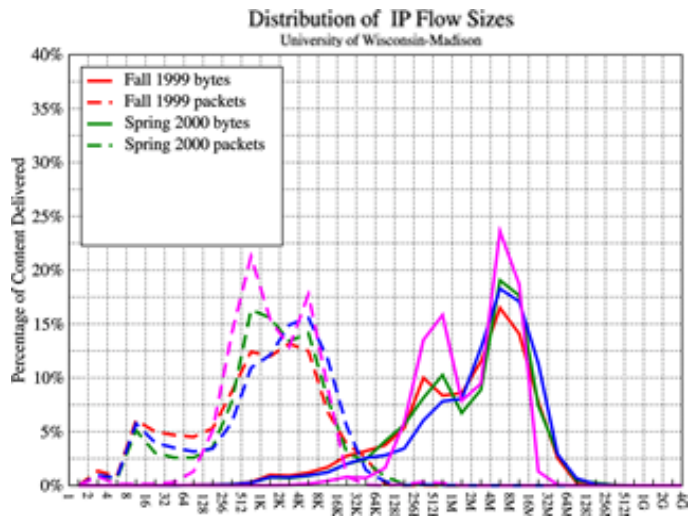


Figure 4.

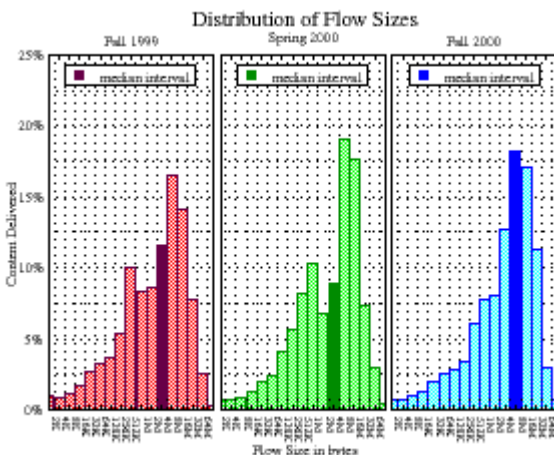


Figure 5.

The *Network Analysis Times* is a newsletter for sharing information, discussing issues, and creating new collaborations in the area of measurement and network analysis research. It is available at: <http://moat.nlanr.net/NATimes/>.

To subscribe, or if you have comments and/or questions, please contact us at: [natimes@nlanr.net](mailto:natimes@nlanr.net).

The *Network Analysis Times* is a publication of the National Laboratory for Applied Network Research (NLNAR), Measurement and Network Analysis Group at the San Diego Supercomputer Center, University of California, San Diego.

Applied Network Research (ANR) home page: <http://moat.nlanr.net/>