DOE/MICS Final Report

Project Title: *"Bandwidth Estimation: Measurement Methodologies and Applications"*

Project Website: http://www.caida.org/projects/bwest and http://www.pathrate.org/

Check one: **[X] SciDAC Project**
[ ] MICS/Base Project

PIs: **kc claffy** **Institution:** **CAIDA/SDSC/UCSD**
Constantinos Dovrolis **Institution:** **Georgia Tech**

Check one:

[ ]	1st Quarterly Report	Date:
[ ]	2nd Quarterly Report	Date:
[ ]	3rd Quarterly Report	Date:
[ ]	4th Quarterly Report	Date:
[X]	Final Report	Date: 31 Oct, 2004

PROJECT SUMMARY

The ability for an application to adapt its behavior to changing network conditions depends on the underlying bandwidth estimation mechanism that the application or transport protocol uses. As such, accurate bandwidth estimation algorithms and tools can benefit a large class of data-intensive and distributed scientific applications. However, tools and methodologies for measuring network bandwidth metrics (such as capacity, available bandwidth, and throughput) have been largely ineffective across real Internet infrastructures.

The first objective of this project was to develop measurement methodologies and tools for various bandwidth-related metrics, such as per-hop capacity, end-to-end capacity, and end-to-end available bandwidth. A second objective was to evaluate and compare existing and emerging bandwidth estimation tools. A third objective was to investigate the application of bandwidth estimation in transport protocols (especially in TCP), adaptive applications, and routing protocols. All objectives have been successfully met. This report summarizes the accomplishments of our project at UCSD. Note that a separate report summarizes the accomplishments of the Bandwidth Estimation project at Georgia Tech (PI: Professor Constantinos Dovrolis). The Bandwidth Estimation project had a significant impact on the SciDAC as well as the rest of the Internet research and user community. Based on our own as well as other tests, the publicly available bandwidth estimation tools developed at Georgia Tech, *pathrate* and *pathload*, are the most accurate and reliable tools developed by the end date of this project, even though several similar tools were developed later. *Pathrate* and *pathload* have been downloaded by more than 2000 users around the world, with about 200 downloads for both tools every month. In addition, several researchers (including DOE and others) used CAIDA's bandwidth estimation test laboratory for tool testing, evaluation and validation. CAIDA's ISMA

workshop on Bandwidth Estimation was the first in the field, bringing together most prominent researchers in this research area.

In terms of research impact, this project resulted in 9 publications in major journals and conferences (including four papers in IEEE premier networking journals). The Scientific Literature Digital Library (CiteSeer.IST) gives about 70 citations to the publications that resulted from this project (even though these papers were only published during the last three years).

Finally, the Bandwidth Estimation project attracted significant interest from both academia and industry, triggering an explosion of interest in this area. Indeed, this project in some respects marked the pervasive penetration of this topic in the field of network research; the near future should bring advances in both banwidth measurement methodologies and tools as well as useful applications of these measurements.

Accomplishments at CAIDA/SDSC/UCSD:

- **Survey on bandwidth estimation**

  Over the last few years, there has been significant progress in the area of bandwidth estimation. More than a dozen software tools have been written, claiming that they measure different bandwidth metrics using different methodologies. In 2003 we wrote a survey paper that described key developments in this area over the last few years, including a taxonomy of the currently available tools, emphasizing their main differences and similarities. The paper[1] was published in *IEEE Network* in November 2003.

- **Establishment of community-accessible bandwidth estimation test laboratory**

  We improved our lab environment and configuration and developed measurement methodologies. After long and arduous troubleshooting in which we discovered and resolved multiple hardware, software, router, and network configuration issues, we opened up our test laboratory for use by DOE collaborators and other researchers.

- **Cataloging community concerns regarding methodological soundness of testing approaches**

  Many studies in the last there years have pointed out a vast range of methodological problems in testing bandwidth estimation tools in either laboratory or real world environments. To provide motivation and context for our design and implementation decisions in our own testing, we catalog the most prominent methodological problems articulated in the literature.

- **Refinement of tool testing methodologies**

  We improved our methodology for testing tools, including the non-trivial challenge of generating realistic yet reproducible simulated cross-traffic. We also automated test data collection and improved our capabilities for independently measuring and graphing cross-traffic and probe traffic using a NeTraMet passive monitor [2].

- **Evaluation of E2E bandwidth estimation tools on high bandwidth paths**

  Results of our experiments confirmed factors that affect tool accuracy including: the presence of layer-2 store and forward devices; differences in the size of internal router queues; and high cross-traffic loads. All of these conditions, also described in Dovrolis' studies [3, 4, 5], are likely to occur in the Internet in the wild, complicating the task of end-to-end bandwidth estimation. In the last year of the project (2004) we did a final report evaluating the performance of all the major publicly available bandwidth estimation tools [6], which we will describe in detail in section 5.

- **Partial integration of bandwidth techniques into TCP kernel**

  We began, but did not have time to complete before the project ended[1], the integration of Georgia Tech's SOBAS algorithm [7] into a FreeBSD TCP stack.

[1] We had assumed we could get a no-cost extension to finish it but Thomas Ndousse denied it.

- **Packet Dispersion Techniques and Passive Capacity Estimation**

  We collaborated with Georgia Tech on early work (2001) analyzing packet pair and packet train dynamics, concluding that it is prohibitively challenging to measure the capacity of a path with just a few packet pairs. We did develop ways to estimate the capacity of a path with packet dispersion techniques, especially if the path is not heavily loaded. Integrating this knowledge base, we developed a capacity estimation methodology that Georgia Tech implemented in a tool called *pathrate*, considered the most state-of-the-art capacity estimation tool.

Over the course of the project, the community itself was in an unrelenting process of questioning underlying assumptions in both testing and measurement methodologies. In particular, Liu et al. *[8] argued that experimental bandwidth estimation work has reached its limit without more analytical depth in pursuit of a deeper understanding of the problem essence and of current proposals. We recognized the need for and undertook more fundamental studies in workload and performance characterization specifically focused on questions relevant to bandwidth estimation methodologies and techniques.*

- **Understanding Internet traffic streams: diversity and disparity**

  Note: This was cost-shared work with other CAIDA projects.

  We investigated the fundamental concept of network traffic streams, and the ways they aggregate into flows through Internet links. We developed a method of measuring the size and lifetime of Internet streams, and used this method to characterise traffic distributions at two different sites [9]. This work was published in IEEE Communications Magazine. Streams can be classified not only by lifetime ('dragonflies' and 'tortoises') but also by size ('mice' and 'elephants'), and we demonstrated that stream size and lifetime are independent dimensions. Internet Service Providers (ISPs) need to be aware of the distribution of Internet stream sizes, and the impact of the difference in behaviour between short and long streams. Indeed, the need to service populations of high diversity in the face of high disparity in resource consumption affects all aspects of network operation: planning, routing, engineering, security, and accounting. We also analyzed diversity/disparity from the perspective of selecting a boundary between mice and elephants in IP traffic aggregated by route, e.g., destination AS [10]. This work was published in PAM 2004.

- **Application of Internet spectroscopy techniques to link capacity characterization**

  In pursuit of richer insight into the fundamental dynamics that affect the bandwidth metrics we are trying to measure, we pioneered the field of *Internet spectroscopy*, developing a technique for revealing bandwidth characteristics of layer-2 technologies without requiring additional traffic probes. This spectroscopy technique is based on an algorithm where a radon transform of inter-packet delay distributions is coupled with entropy minimization [11].

  We demonstrated the feasibility of Internet spectroscopy techniques for analysis of rate limiting, packet interarrival delay and passive bitrate estimation of cell- or slot-based broadband connections. Working with highly diverse packet trace data, we find that delay quantization in micro- and millisecond range is ubiquitous in today's Internet and that different providers have strong preferences for specific delay quanta in their infrastructures.

- **Nonstationary Poisson view of Internet traffic**

Since the identification of long-range dependence in network traffic eleven years ago [12], its consistent appearance across numerous measurement studies has largely discredited Poisson-based traffic models. However, since that original data set was collected, both link speeds and the number of Internet-connected hosts have increased by more than three orders of magnitude. In pursuit of more fundamental understanding of traffic structure, we revisited the Poisson assumption, by studying a combination of historical traces and new measurements obtained from a major backbone link belonging to a Tier 1 ISP. We showed that unlike the older data sets, current network traffic can be well represented by the Poisson model for sub-second time scales. At multi-second scales, we find a distinctive piecewise-linear non-stationarity, together with evidence of long-range dependence. Combining our observations across both time scales leads to a time-dependent Poisson characterization of network traffic that, when viewed across long time scales, exhibits the observed long-range dependence. This traffic characterization reconciliates the seemingly contradicting observations of Poisson and long-memory traffic characteristics. It also seems to be in general agreement with recent theoretical models for large-scale traffic aggregation.

- **Visualization of bandwidth estimation data**

  An often overlooked, as well as persistently challenging, mode of exploratory data analysis is the use of visualization tools. CAIDA leveraged Georgia Tech's development of *ANEMOS*, an *Autonomous NEtwork MOnitoring System* for this project to visualize the output of bandwidth estimation tools [13]. The current ANEMOS prototype measures end-to-end available bandwidth with *Pathload*, and round-trip delays and losses with a UDP-based configurable variation of *Ping*. The measurements are archived using the *MySQL* database, and they can be visualized using *MRTG*. We tested this tool out on the high bandwidth (gigabit) TeraGrid path between SDSC and NCSA.

Description of accomplishments

1 Survey on bandwidth estimation

Application users on high-speed networks perceive the network as an end-to-end connection between resources of interest to them. In order to optimize the network utilization, users (or their applications) need the ability to discover the highest performing available end-to-end path to distributed resources. Therefore, they need tools and methodologies to monitor network conditions and to rationalize their performance expectations.

There are several network characteristics related to performance and measured in bits per second: capacity, available bandwidth, bulk transfer capacity, and achievable TCP throughput. Although these metrics appear similar they are not, and knowing one of them does not give a definitive indication of any of the others. In the first year of the project we surveyed the state-of-the-art in bandwidth estimation techniques [1]. We gave rigorous definitions of terms used in the field, described underlying techniques and methodologies, and provided a list of open source measurement tools for each of the metrics. Three clear challenges loomed:

- The accuracy of bandwidth estimation techniques was low, especially on high bandwidth paths (e.g., greater than 500Mbps), and difficult to ascertain without access to 'ground truth' for measured paths, i.e., controlled testing scenarios.

- All known bandwidth estimation tools and techniques assumed that routers serve packets in a First-Come First-Served (FCFS) manner. It is not clear how these techniques perform in routers with multiple queues, e.g., for different classes of service or in routers with virtual-output input queues.
- These fundamental methodological and functional issues served as disincentives for the community to invest effort in using bandwidth estimation tools to support applications, middleware, routing, and traffic engineering techniques, in order to improve end-to-end performance and enable new services.

The tasking in this project (for both UCSD and Georgia Tech) focused directly on meeting the above three challenges.

2 Establishment of community-accessible bandwidth estimation test laboratory

Collaborating (and cost-sharing) with the CalNGI's Network Performance Reference Lab [14] allowed CAIDA to develop a much richer testing environment than would have been possible with just our own budget. We built a high-speed testbed for use in testing available bandwidth estimation tools under identical and reproducible experimental conditions. This CAIDA/CalNGI Network Performance Reference Lab environment allowed us to conduct series of experiments using two different sources of realistic and reproducible cross-traffic, and to look deeply into internal details of tool operation.

In our current testbed configuration (Figure 2), all end hosts are configured on separate networks and connected to switches capable of handling jumbo MTUs (9000 B). Three routers in the tested end-to-end path are each from a different manufacturer. Routers were configured with two separate domains (both within private RFC1918 space) that route all packets across a single testbed 'backbone'. An OC48 link connects a Juniper M20 router with a Cisco GSR 12008 router, and a GigE link connects the Cisco with a Foundry BigIron 10 router. We use jumbo MTUs (9000 bytes) throughout our OC48/GigE configuration in order to support traffic flow at full line speed [15].

Bandwidth estimation tools run on two designated end hosts each equipped with a 1.8 GHz Xeon processor, 512 MB memory, and an Intel PRO/1000 GigE NIC card installed on a 64b PCI-X 133 MHz bus. The operating system is the CAIDA reference FreeBSD version 4.8.

Our laboratory setup also includes dedicated hosts that run *CoralReef* [16] and *NeTraMet* [17] passive monitor software for independent verification of tool and cross-traffic levels and characteristics. Endace DAG 4.3 network monitoring interface cards on these hosts tap the OC-48 and GigE links under load. *CoralReef* can either analyze flow characteristics and packet interarrival times (IATs) in real time or capture header data for subsequent analysis. The *NeTraMet* passive RTFM meter collects packet size and IAT distributions in real time, separately for tool and cross- traffic.

Several bandwidth estimation tool developers (including those in DOE) have taken advantage of our support for remote access to the testbed to conduct their own tests. Lessons from this testing experience will support future high speed network monitoring efforts.

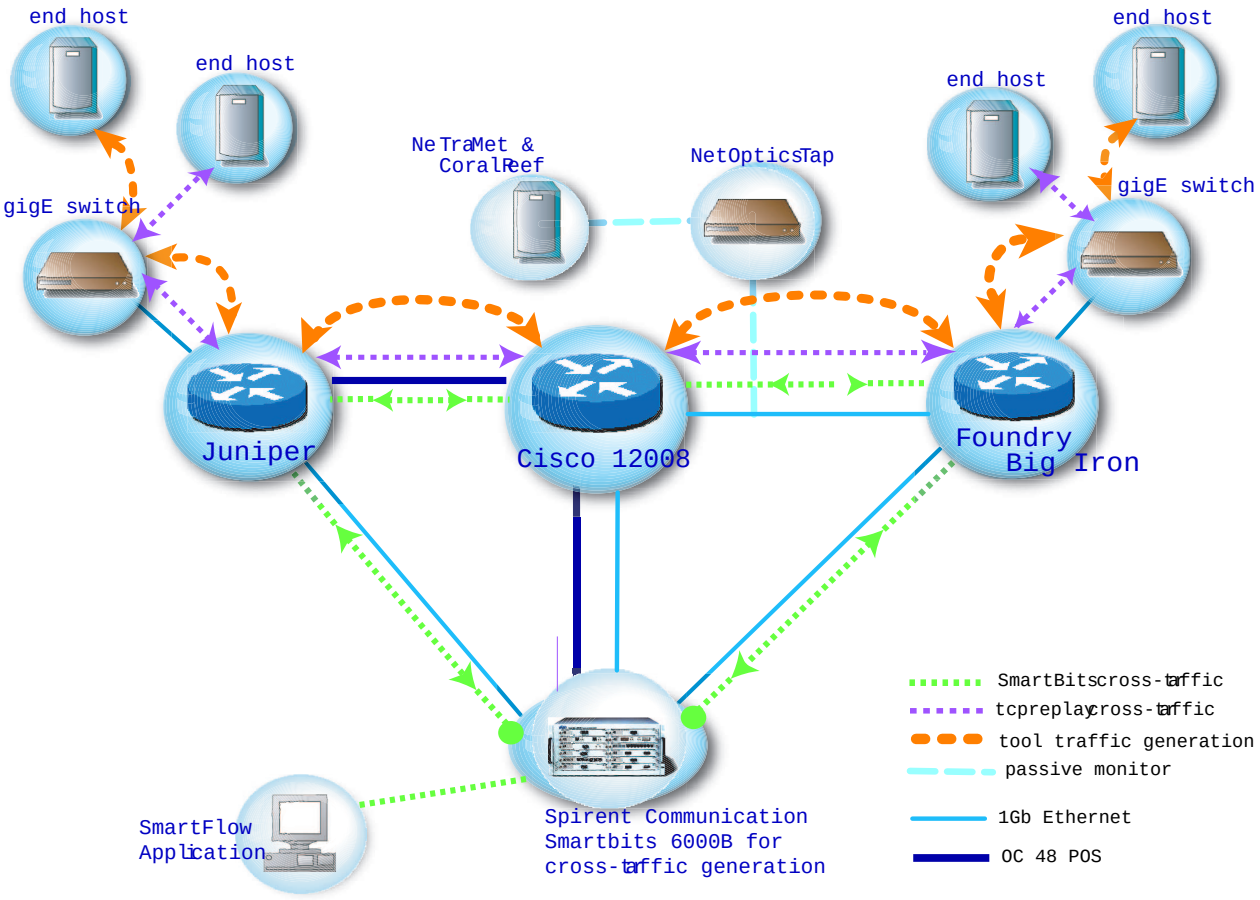

Figure 1: Bandwidth Estimation Testbed. The end-to-end path being tested runs through three routers and includes OC48 and GigE links. Tool traffic occurs between designated end hosts in the upper part of this figure. Cross-traffic is injected by either additional end hosts behind the jumbo-MTU capable GigE switches or by the Spirent SmartBits 6000 box (lower part of figure). Passive monitors tap the path links as shown for independent measurement verification.

3 Cataloging community concerns regarding methodological soundness of testing approaches

> "The throughput achieved by an application running between two hosts is a hopelessly complicated function of the states of the application and of the network." Coccetti, *et al.* [18]

Several studies have pointed out a vast range of problems in testing bandwidth estimation tools. In the interest of completeness, we highlight the most prominent problems in the literature and in the next section discuss our approaches to dealing with them.

As early as 2001, Matoba *et al.* [19] found that *pathchar* could yield dramatically inaccurate estimates in the presence of traffic dynamics or route alternation. They described the importance of two factors in obtaining an accurate and reliable estimation: the *confidence interval* and the

measurement period, and they proposed an adaptive method to control the number of measurement data sets. Unfortunately they were only able to validate their approach against lower capacity (T1 and T3) paths without any layer 2 hardware. The inability to test tools on high bandwidth, controlled paths is a pervasive methodological weakness in the literature, as we will see in the remainder of this section.

Dovrolis and Jain [4] made one of the first (and few existing) studies that attempted to validate experiment against real world (using SNMP data at a 5-min granaularity), although they had to make assumptions about the tight link in path being at the edge, which was 8.2Mbps and not so relevant for the high bandwidth ranges of interest to the DOE community. Furthermore, Akella *et al.* [20] demonstrate that it is not safe to assume that a path's bottleneck is at the edge.

In 2002, Coccetti *et al.* [18] investigated the wide range of risks in interpretation of bandwidth measurements. Most importantly, they found through carefully controlled laboratory experiments that bandwidth measurement tool results depend strongly on the configuration of queues in the routers, implying that considerable care must be exercised in the interpretation of output, in order to avoid pitfalls due to presence of QOS or other traffic engineering features in the network. Indeed, in the presence of many types of load balancing, no available tool can detect the true capacity of the links. As with the previous study, the authors were also not able to test on link bandwidths above a few Mbps but the general principles seem likely to apply to higher bandwidth paths.

In 2003, Hu and Steenkiste [21] contributed to the community's knowledge regarding testing methodology by characterizing the interaction between probing packets and the competing network traffic. Using a simple single-hop network model, they show that the initial probing gap is a critical parameter when using packet pairs (a common method) to estimate available bandwidth. They also find that the measurement accuracy of active probing is affected by factors such as the probing packet size, the length of probing packet train, and the competing traffic on links other than the tight link. They conclude that, in general, average-sized probing packets of about 500 to 700 Byte are likely to yield the most representative available bandwidth estimate. Smaller packet sizes may underestimate the available rate and may be more sensitive to measurement errors, while larger probing packet sizes can overpredict the available bandwidth. They also admit that another huge barrier to greater accuracy is the generally poor quality of timestamps in affordable measurement hardware. This timestamp precision issue recurred continually in the course of our work, and clearly needs dedicated research resources since the lack of high quality timestamps forestalls Internet research in a number of areas, not limited to bandwidth estimation.

In 2003, Strauss *et al.* [22] introduced their *spruce* tool and attempt to compare its accuracy and performance to other available bandwidth estimation tools, in particular *IGI* and *pathload*. While they found that *spruce*'s overall performance was superior to that of *IGI* or *pathload*, they admit the difficulty of finding paths for which they can find SNMP link utilization data against which to validate their tools, and they offer the explicit caveat that they do not claim their tested paths (on the MIT campus and the Abilene network) are representative. When they test their tools on PlanetLab, the hope for fine-grained utilization data between sites on the global Internet is even less likely, so they use a differential test (d-test) that measures changes in the available bandwidth rather than absolute values. The also make the explicit admission that this approach bears the assumption that traffic does not change signficantly bewteen phases of tests. As if this relatively weak scientific framework were not constraining enough, their measurements also require careful scheduling of probe traffic (i.e., input gap between a pair of probes must be accurate and sometimes as small as a few hundred microseconds) that blocks other programs during train transmission. Thus on anything but the lowest bandwidth paths, a dedicated box on both ends will be required for accurately measuring

available bandwidth.

Several researchers argued at the CAIDA's December 2003 ISMA workshop on bandwidth estimation [23] that all techniques for available bandwidth estimation required control of both endpoints, and that the fundamental reason behind this requirement is the need to measure the gap values or packet rain rate on the destination node in order to eliminate the effects of queueing in the reverse path and the asymmetry of Internet paths. One-endpoint solutions will require a much deeper understanding of how and to what extent reverse-path queueing affects the measurement. Indeed, this may be a question that transcends possible IP measurement technology and bears implications for future network architectures. In the meantime the best available probing techniques depend on the precise application requirement and we need tools that not only adapt to network paths but also can be tuned to application needs.

Two papers that discuss measurement methodology are particularly relevant to the DOE community. DOE researchers Cottrell and Log [24] give an overview of the IEPM-BW project, which supports bandwidth testing in support of DOE's bulk data transfer requirements. The purpose of IEPM-BW is to understand what throughputs are achievable, to identify and optimize constraints, and to make data and predictions available. In building and maintaining the infrastructure, they struggled considerability with software stability and OS dependencies, rendering validation the largest obstacle to progress.

Ubik *et al.* [25] studied performance monitoring of high speed networks from the NREN perspective. They found that *ABwE* (one of DOE's tools for bandwidth estimation) accuracy and reliability was still insufficent to assist congestion control to set the optimal sending rate and that results from different tools generally did not match. Thus, it was not easy to assess the available bandwidth with any confidence. In their view, the future of performance monitoring must include two primary tasks: (1) develop an extensible inter-domain platform for end-to-end performance monitoring. They have started development of such a platform within TF-NGN and will continue as part of the European GEANT2 project. (2) to develop a programmable monitoring adapter needed for fine-grained (i.e., precision timestamps) monitoring at speeds higher than 1 Gb/s. An adapter of this kind is being developed as part of the European SCAMPI projectc.

4 Refinement of tool testing methodology based on community concerns

The overwhelming and pervasive difficulties in the integrity of testing of bandwidth estimation tools led CAIDA to undertake an essential role in building a solid environment for more rigorous testing, and making that environment available to other researchers interested in doing more objective evaluation of tools.

4.1 Methods of generating cross-traffic

The algorithms used by bandwidth estimating tools make assumptions about characteristics of the underlying cross-traffic. In a situation when these assumptions are not applicable, tools cannot perform correctly. Therefore, it is essential to use test traffic that closely simulates traffic on real networks and reproduces its most critical characteristics, such as packet IAT and size distributions. In our study we conducted two series of laboratory tool tests using two different methods of cross-traffic generation, and carefully examined the characteristics of our test traffic in order to justify the validity of our choices and their impact on tools performance. We describe these methods below.

4.1.1 Synthetic cross-traffic

Spirent Communications SmartBits 6000B [26] is a hardware system for testing, simulating and troubleshooting network infrastructure and performance. It uses the Spirent *SmartFlow* [27] application that enables controlled traffic generation for L2/L3 and QoS laboratory testing.

Using SmartBits and *SmartFlow* we can generate pseudo-random yet reproducible traffic with accurately controlled load levels and packet size distributions. This traffic generator models pseudo-random traffic flows where the user sets the number of flows to produce and the number of bytes to send to a given port/flow before moving on to the next one (burst size). The software also allows the user to define the L2 frame size for each component flow. The resulting synthetic traffic emulates realistic protocol headers. However, it does not imitate TCP congestion control and is not congestion-aware.

In our experiments we varied traffic load level from 100 to 900 Mb/s which corresponds to 10-90% of the GigE link capacity. At each load level *SmartFlow* generated nineteen different flows. Each flow had a burst size of 1 and consisted of either 64, 576, 1510 or 8192 byte L2 frames. The first three sizes correspond to the most common L2 frame sizes observed in real network traffic [28]. We added the jumbo packet component because high-speed links must employ jumbo MTUs in order to push traffic levels to line saturation. While NLANR's PMA [28] data suggest a tri-modal distribution of small/medium/large frames in approximately 60/20/20% proportions, there is no equivalent published packet size data for links where jumbo MTUs are enabled. We mixed the frames of four sizes in equal proportions.

Packet IATs (Figure 2) ranged from 4 to more than 400 μs. We used passive monitors *CoralReef* and *NeTraMet* to verify the actual load level of generated traffic and found that it matched the requirements within 1-2%.

4.1.2 Playing back traces of the real traffic

We replayed previously captured traffic traces on our laboratory end-to-end path using a tool *tcpreplay* [29]. This method of cross-traffic generation reproduces realistic IAT and packet size distributions but is not congestion-aware. The playback tool ran on two additional end hosts (separate from the end hosts running bandwidth estimation tools) and injected the cross-traffic into the main end-to-end path via GigE switches.

We tested bandwidth estimation tools using two different traces:

- a 6-minute trace collected from a 1 Gb/s backbone link of a large university with approximately 300-345 Mb/s of cross-traffic load
- a 6-minute trace collected from a 2.5 Gb/s backbone link of a major ISP showing approximately 100-200 Mb/s of cross-traffic load.

Neither of the traces used in our testing contained any jumbo frames. Packet sizes exhibited a tri-modal distribution. Packet IATs (Figure 3) ranged from 1 to 60 μs.

We used CoralReef to continuously measure *tcpreplay* cross-traffic on the laboratory end-to-end path and recorded timestamps of packet arrivals and packet sizes. We converted this information into timestamped bandwidth readings and compared them to concurrent tools estimates. Both traces exhibited burstiness on microsecond time scales, but loads were fairly stable when aggregated over one-second time intervals.

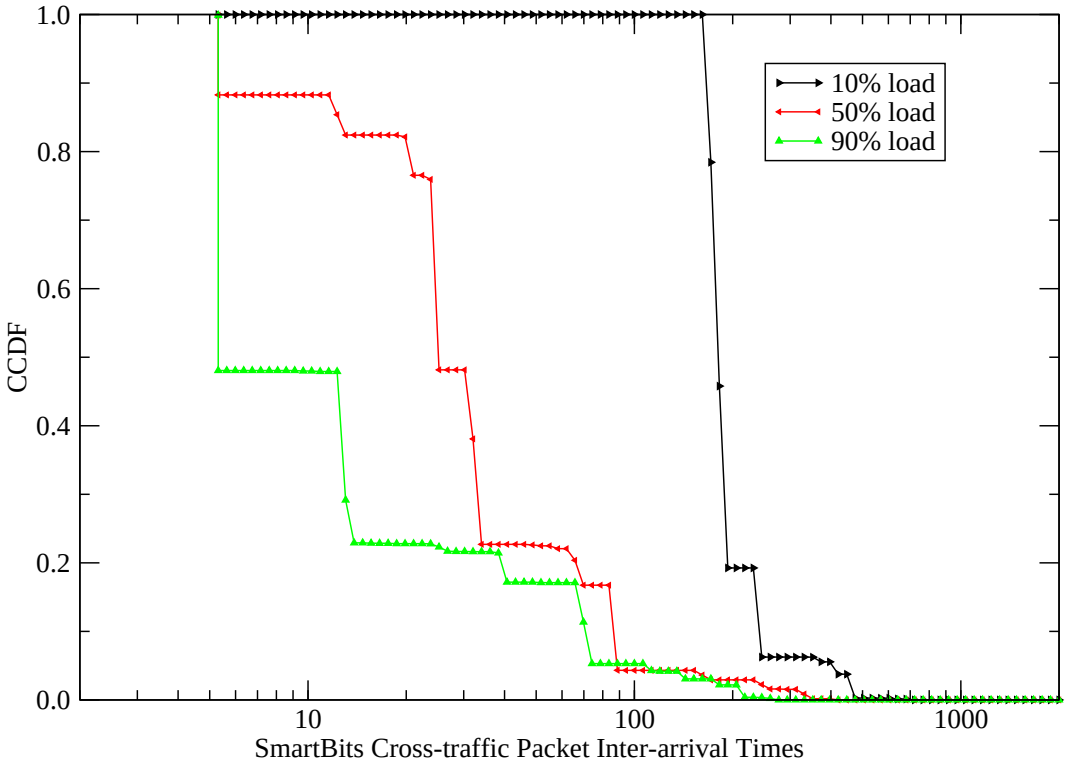

Figure 2: CCDF of packet IAT distribution for synthetically generated SmartBits cross-traffic at 10, 50, and 90% loads.

5 Evaluation of E2E bandwidth estimation tools on high bandwidth paths

In [6] and in the extended version of this study [30], we did our meticulous comparison of publicly available end-to-end bandwidth estimation tools on high-speed links. As the first comprehensive evaluation of publicly available tools for available bandwidth estimation on high bandwidth links, this study is an essential accomplishment of this project, and offers several unique contributions to the field. First, we considered and evaluated a larger number of tools than any previous authors. Second, we conducted two series of reproducible laboratory tests in a fully controlled environment [14] using two different sources of realistic, reproducible cross-traffic. Third, we experimented on high-speed (OC-48 and GigE) paths where we had a complete knowledge of link capacities and had access to SNMP counters for independent cross-traffic verification. We compared the accuracy and other operational characteristics of the tools, and analyzed factors impacting their performance.

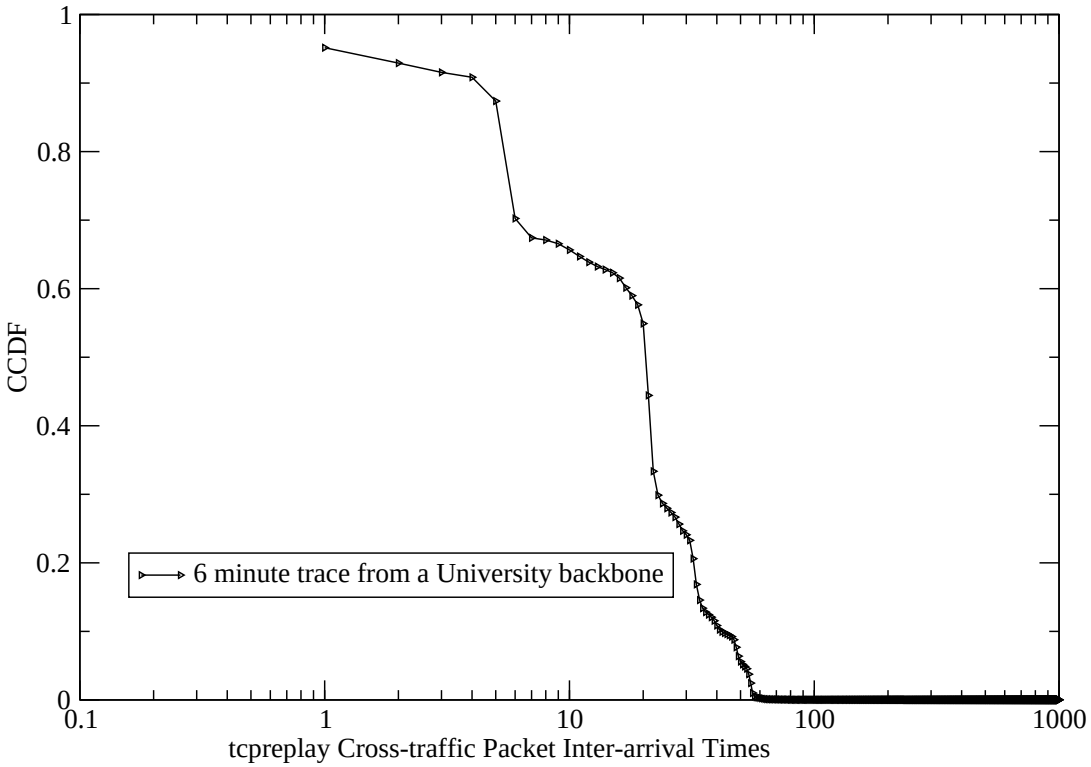

Figure 3: CCDF of packet IAT distribution on a replayed traffic trace.

5.1 Tools tested

In this study our goal was to test and compare tools that claim to measure the available end-to-end bandwidth (Table 1). We did not test tools that measure end-to-end capacity. By definition, end-to-end capacity of a path is determined by the link with the minimum capacity (narrow link). End-to-end available bandwidth of a path is determined by the link with the minimum unused capacity (tight link) [1].

From Table 1 we selected the following tools for evaluation: *abing*, *pathchirp*, *pathload*, and *spruce*. For comparison, we also included *iperf* [37] which attempts to measure achievable TCP throughput. This tool is widely used for end-to-end performance measurements and has become an unofficial standard [38] in the academic networking community. We were unable to test *cprobe* [32] because it only runs on an SGI Irix platform and we do not have one in our testbed.

5.2 Testing methodology

We used the methodology described in section 4.

Tool	Author	Methodology
abing	Navratil [31]	Pkt pair
cprobe	Carter [32]	Pkt Trains
IGI	Hu [21]	SLoPS
netest	Jin [33]	Unpublished
pathchirp	Ribeiro [34]	chirp train
pipechar	Jin [35]	Unpublished
pathload	Jain [36]	SLoPS
spruce	Strauss [22]	SLoPS

Table 1: Available Bandwidth Estimation Tools.

5.3 Tool evaluation results: laboratory

In this Section we present tool measurements in laboratory tests using synthetic, non-congestion-aware cross-traffic with controlled traffic load (*SmartFlow*) and captured traffic traces with even more realistic workload characteristics (*tcpreplay*). In Section 5.4 we show results of experiments on a real high-speed network.

5.3.1 Comparison of tool accuracy

Experiments with synthesized cross-traffic. We used the SmartBits 6000B device with the *SmartFlow* application to generate bi-directional traffic loads, varying from 10% to 90% of the 1 Gb/s end-to-end path capacity in 10% steps. We tested one tool at a time. In each experiment, the synthetic traffic load ran for six minutes. To avoid any edge effects, we delayed starting the tool for several seconds after initiating cross-traffic and ran the tool continuously for five minutes. Figure 4 shows the average and standard deviation of all available bandwidth values obtained during these 5-minute intervals for each tool at each given load.

Our end-to-end path includes three different routers with different settings. To check whether the sequence of routers in the path affects the tool measurements, we ran tests with synthesized cross-traffic in both directions. We observed only minor differences between directions. The variations are within the accuracy range of the tools and we suspect are due to different router buffer sizes.

We found that *abing* (Figure 4a) reports highly inaccurate results when available bandwidth drops below 600 Mb/s (60% on a GigE link). Note that this tool is currently deployed on the Internet End-to-End Performance Monitoring (IEPM) measurement infrastructure [39] where the MTU size is 1500 B, while our high-speed test lab uses a jumbo 9000 B MTU. We attempted to change *abing* settings to work with its maximum 8160 B probe packet size, but this change did not improve its accuracy on our testbed.

We investigated further details of *abing* operating on an empty GigE link. The tool continuously sends back-to-back pairs of 1478 byte UDP packets with 50 ms waiting interval between pairs. It derives estimates of available bandwidth from the amount of delay introduced by the network between the paired packets. Computing the packet IAT does not require clock sychronization. *abing* puts a timestamp into each packet, and the returned packet carries a receiver timestamp. Since these timestamps have a μs granularity, the IAT computed from them is also an integer number of μs. For back-to-back 1500 B packets on an empty 1 Gb/s link (12 Kb transmitted at 1 ns per bit) the IAT is either 11 or 12 μs, depending on rounding error. However, we observed that every 20-30 packets the IAT becomes 244 μs. This jump may be a consequence of interrupt coalescence or a delay in some intermediate device such as a switch. The average IAT then changes to more than 20

μs, yielding a bit rate of less than 600 Mb/s. This observation explains *abing* results: on an empty 1 Gb/s link it reports two discrete values of available bandwidth, the more frequent one of 890-960 Mb/s and occasional drops to 490-550 Mb/s. This oscillating behavior is clearly seen in time series of measurements (Figure 5) described below.

Another tool, *spruce* (Figure 4d), uses a similar technique, and unsurprisingly its results are impeded by the same phenomenon. It sends 14 back-to-back 1500 B UDP packet pairs with a waiting interval of 160-1400 ms between pair probes (depending on some internal algorithm). In *spruce* measurements, 244 μs gaps between packet pairs occur randomly among the prevailing 12 μs gaps. Since the waiting time between pairs varies without pattern, the reported available bandwidth also varies without pattern in the 300-990 Mb/s range.

Results of our experiments with *abing* and *spruce* on high-speed links caution that tools utilizing packet pair techniques must be aware of delay quantization possibly present in the studied network. Also, 1500 byte frames and microsecond timestamp resolution simply are not sensitive enough for probing high bandwidth paths.

In SmartBits tests, estimates of available bandwith by *pathchirp* are 10-20% higher than the actual value (Figure 4b). The consistent overestimation persists even when there is no cross-traffic. On an empty 1 Gb/s link this tool yields values up to 1100 Mb/s. We found no explanation for this behavior.

We found that results of *pathload* were the most accurate (Figure 4c). The discrepancy between its readings and actual available bandwidth was less than 10% in most cases.

The last tested tool, *iperf*, estimates not the available bandwidth, but the achievable TCP throughput. We ran it with the maximum buffer window size of 227 KB and found it to be rather accurate in tests with synthesized cross-traffic (Figure 4e). Note that any smaller buffer window size setting significantly reduces the *iperf* throughput. This observation appears to contradict the usual rule of thumb that the optimal buffer size is the product of bandwidth and delay, which in our case would be (10^9 b/s) x (10^{-4} s) $\sim$ 12.5 KB.

Experiments with trace playbacks. The second series of laboratory tests used previously recorded traces of real traffic. We extracted six-minute samples from longer traces to use as a *tcpreplay* source. As in SmartBits experiments, we delayed the tool start for a few seconds after starting *tcpreplay* and ran each tool continuously for five minutes.

Figure 5 plots a time series of the actual available bandwidth, obtained by computing the throughput of the trace at a one second aggregation interval and subtracting that from the link capacity of 1 Gb/s. Time is measured from the start of the trace. We then plot every value obtained by a given tool at the time it was returned.

As described in Section 4.1.2, we performed *tcpreplay* experiments with two different traces. We present tool measurements of the University backbone trace, which produced a load of about 300 Mb/s leaving about 700 Mb/s of available bandwidth. The tool behavior when using the ISP trace with a load of about 100 Mb/s was similar and is not shown here.

In tests with playback of real traces, *abing* and *spruce* exhibit the same problems that plagued their performance in experiments with synthetic cross-traffic. Figure 5a shows that *abing* returned one of two values, neither of which was close to the expected available bandwidth. *spruce* results (Figure 5d) continued to vary without pattern.

pathchirp measurements (Figure 5b) had a startup period of about 70 s when the tool returned only a constant value. The length of this period is related to the tool's measurement algorithm and depends on the number of chirps and chirp packet size selected for the given tool run. After the startup phase, *pathchirp*'s values alternate within 15-20% of the actual available bandwidth.

The range reported by *pathload* (Figure 5c) slightly underestimates the available bandwidth by <16%.

iperf reports surprisingly low results when run against *tcpreplay* traffic (Figure 5e). Two factors are causing this gross underestimation: packet drops requiring retransmission; and a too long retransmission timeout of 1.2 s (default value). In the experiment shown, the host running *iperf* and the host running *tcpreplay* were connected to the main end-to-end path via a switch. We checked the switch's MIB for discarded packets and discovered a packet loss of about 1% when two traffic streams merge. Although the loss appears small, it causes *iperf* to halve its congestion window and triggers a significant number of retransmissions. The default retransmission timeout is so large that it can consume up to 75% of the *iperf* running time. Decreasing the retransmission timeout to 20 ms and/or connecting the *tcpreplay* host directly to the path bypassing the switch considerably improves *iperf* performance. Note that we were able to reproduce the degraded *iperf* performance in experiments with synthetic SmartBits traffic when we flooded the path with a large number of small (64 B) packets. These experiments confirm that ultimately TCP performance in the face of packet loss strongly depends on the OS retransmission timer.

5.3.2 Comparison of tool operational characteristics

We considered several parameters that may potentially affect a user's decision regarding which tool to use: measurement time, intrusiveness, and overhead. We measured all these characteristics in experiments with SmartBits synthetic traffic where we can stabilize and control the load.

We define tool measurement time to be the average measurement time of all executions at a particular load level. On our 4-hop OC-48/GigE topology, the observed average measurement times were: 1.3 s for *abing*, 11 s for *spruce*, 5.5 s for *pathchirp*, and 10 s for *iperf* independent of load. The *pathload* measurement time generally increased when the available bandwidth decreased, and ranged between 7 and 22 s.

We define tool intrusiveness as the ratio of the average tool traffic rate to the available bandwidth, and tool overhead as the ratio of tool traffic rate to cross-traffic rate (Figure 6). *pathchirp*, *abing*, and *spruce* have a low overhead, each consuming less than 0.2% of the available bandwidth on the GigE link and introducing practically no additional traffic into the network as they measure. *pathload* intrusiveness is between 3 and 7%. Its overhead slightly increases with the available bandwidth (that is, when the cross-traffic actually decreases) and reaches 30% for the 10% load. As expected, *iperf* is the most expensive tool both in terms of its intrusiveness (74-79%) and overhead costs. Since it attempts to occupy all available bandwidth, its traffic can easily exceed the existing cross-traffic.

5.4 Tool evaluation results: real Internet paths

Previous comparisons of bandwidth estimation tools have been criticized for their lack of validation in the real world. Many factors impede (if not prohibit) comprehensive testing of tools on production IP networks. First, network conditions and traffic levels are variable and usually beyond the experimenters' control. This uncertainty prevents unambiguous interpretation of experimental results and renders measurements unreproducible. Second, a danger that tests may perturb or even disrupt the normal course of network operations makes network operators reluctant to participate in any experiments. Only close cooperation between experimenters and operators can overcome both obstacles.

We were able to complement our laboratory tests with the available bandwidth measurements on a 6 hop end-to-end path from Sunnyvale to Atlanta on the Abilene network. Both end machines had

1 Gb/s connection to the network and no traffic except from running our tools. The rest of links in the path had either 2.5 or 10 Gb/s capacities.

We chose not to run *spruce* on the Abilene backbone network since this tool failed in all our laboratory experiments on high-speed paths. As before, we ran *pathload*, *pathchirp*, *abing*, and *iperf* for 5 min each. The experiments occurred in that order, back-to-back. We concurrently polled the SNMP 64-bit InOctect counters for all routers along the path every 10 seconds and hence knew the per-link utilizations with 10s resolution. We calculated the per-link available bandwidth as the difference between link capacity and utilization. The end-to-end available bandwidth is the minimum of per-link available bandwidths. During our experiments, the Abilene network did not have enough traffic on the backbone links to bring their available bandwidth below 1 Gb/s. Therefore, the end machines' 1Gb/s connections were both the narrow and tight links in our topology.

Figure 7 shows our tool measurements and SNMP-derived available bandwidth. Measurements with *pathload*, *pathchirp*, and *iperf* are reasonably accurate, while *abing* readings wildly fluctuate in the entire range between 0 and 1000 Mb/s.

A seeming discrepancy between *iperf* measurements and SNMP-derived values is due to a large (>70%) overhead of the *iperf* tool. Consequent readings of SNMP counters tell us how many bytes passed through an interface of a router during that time interval. They report the total number of bytes without distinguishing tool traffic from cross-traffic. If a tool overhead is high, then available bandwidth derived from SNMP data during this tool execution will be low. At the same time, since the high-overhead tool attempts to measure the available bandwidth ignoring its own traffic, it will report more available bandwidth than SNMP reports. Therefore, *iperf* shows a correct value of the achievable TCP throughput of ∼950 Mb/s while simultaneous SNMP counters account for *iperf*'s own traffic and yield less than 200 Mb/s of available bandwidth. Note that a smaller discrepancy between *pathload* and SNMP results also reflects this tool's overhead (∼10% per our lab tests).

5.5 Tool evaluation: conclusions and future directions

We demonstrated how our testbed can be used to rigorously evaluate and compare end-to-end bandwidth estimation tools using generated cross-traffic that allows us to saturate high-speed paths with realistic and reproducible load. We found that *pathload* and *pathchirp* are the most accurate tools. *iperf* performs well on high-speed links if run with its maximum buffer window size. Even small (1%) but persistent packet loss seriously degrades its performance. Too conservative settings of the OS retransmission timer further exacerbate this problem. Results of our experiments with *abing* and *spruce* caution that tools utilizing packet pair techniques must be aware of delay quantization possibly present in the studied network. Using 1500B frames and microsecond timestamp resolution on gigE paths increases the introduced error; the combination is simply not sensitive enough for probing high-speed paths.

Candidate bandwidth estimation tools face increasingly difficult measurement challenges as link speeds increase and router and switch functionality grows more complex. Of particular concern is the issue of timestamp precision and synchronization: as link speeds increase, intervals between packets decrease, making packet probe measurements more sensitive to timing errors. The standard 1 μs granularity of UNIX timestamps is acceptable when measuring 120 μs gaps between 1500 B packets on 100 Mb/s links but insufficient to quantify packet interarrival time variations on 12 μs gaps on GigE links. Available bandwidth measurements on high-speed links stress the limits of clock precision especially since additional timing errors may arise due to the NIC itself, the operating system, or the Network Time Protocol (designed to synchronize the clocks of computers over a network) [40].

As described in section 3, several other problems may be introduced by network devices and

configurations. Newer faster NICs often collect several packets before issuing an OS interrupt. Interrupt coalescence improves network packet processing efficiency but breaks end-to-end tools that assume uniform per packet processing and timing [5]. Hidden Layer 2 store-and-forward devices distort an end-to-end tool's path hop count, also resulting in estimation errors [3]. MTU mismatches add timing and end-to-end probing errors by artificially limiting path throughput. Modern routers that relegate probe traffic to a slower path or implement QoS mechanisms may also break crucial assumptions about packet handling made by end-to-end probing tools. Concerted cooperative efforts of network operators, researchers and tool developers can resolve those (and many other) network issues and advance the field of bandwidth measurements.

While accurate end-to-end measurement is difficult, it is also important that bandwidth estimation tools be fast and relatively unintrusive. Otherwise, answers are incorrect, arrive too late to be useful, or the end-to-end probe may itself interfere with the network resources that the user attempts to measure and exploit.

6 Partial integration of bandwidth techniques into TCP kernel

TCP researcher Egemen Kavak visited CAIDA for the summer of 2004 to work on integrating Georgia Tech's SOBAS algorithm [7] into a TCP stack. The SOBAS algorithm measures the throughput of a TCP connection and detects when the throughput has stabilized (the 'flat rate' condition in [7]). It then sets the socket buffer for the connection to the steady rate times RTT (a version of bandwidth-delay product.)

Egemen made substantial progress on implementing the SOBAS algorithm as part of the FreeBSD TCP stack. He implemented receiver-side RTT estimation, flat-rate condition checking and setting socket buffer. Normally, a TCP stacks does not do receiver side RTT estimation; RTT and RTT variance are only estimated by the sender using the Jacobson-Karels algorithm. However, due to our inability to get a no-cost extension for the remaining funds left in the grant, we did not have time to debug the code. This task remains incomplete.

7 Packet Dispersion Techniques and Passive Capacity Estimation

The packet pair technique aims to estimate the capacity of a path (bottleneck bandwidth) from the dispersion of two equal-sized probing packets sent back-to-back. It has been also argued that the dispersion of longer packet bursts (packet trains) can estimate the available bandwidth of a path. We examined such packet pair and packet train dispersion techniques in depth [41]. We first demonstrated that, in general, packet pair bandwidth measurements follow a multimodal distribution, and explained the causes of multiple local modes. The path capacity is a local mode, often different from the global mode of this distribution. We illustrated the effects of network load, cross-traffic packet size variability, and probing packet size on the bandwidth distribution of packet pairs. The bottom line is that it is prohibitively challenging to measure the capacity of a path with just a few packet pairs.

We then examined the dispersion of long packet trains. The mean of the packet train dispersion distribution corresponds to a bandwidth metric that we refer to as Average Dispersion Rate (ADR). We showed that the ADR is a lower bound of the capacity and an upper bound of the available bandwidth of a path. We showed that it is possible to estimate the capacity of a path with packet dispersion techniques, especially if the path is not heavily loaded. However, to do so, it is important to understand the dispersion techniques not only in the statistical sense, but in terms of the queueing

effects that shape the distribution of bandwidth measurements: network load, cross traffic packet size variability, probing packet size, train length, and cross-traffic routing. Integrating this knowledge base, we developed a capacity estimation methodology that we implemented in a tool called *pathrate*. We also reported on experiences with *pathrate* after having measured hundreds of Internet paths over the last three years.

8 Understanding Internet Traffic Streams: diversity and disparity

We recognize the need for and undertook more fundamental studies in workload and performance characterization specifically focused on questions relevant to bandwidth estimation methodologies and techniques. The next three sections will highlight such studies.

Note: This was cost-shared work with other CAIDA projects.

A dramatic recent increase in network and computer capability has allowed users to work with ever larger files. As a result we now observe that the average size of web objects has increased considerably over the last 5 years, with web objects up to 50 kB becoming common. Along with increasing file size, the last few years have seen the rapid growth in usage of an ever increasing set of peer-topeer file sharing systems. e.g. Napster, Gnutella, E-Donkey, etc. These peer-to-peer applications have significantly changed the traffic mix, so that a higher overall proportion of their streams have large numbers of bytes. In addition to streaming protocols carrying audio and video programs, VoIP or multimedia conferencing are increasingly common. Clearly these trends will continue. Our current observations [9] confirm that most streams are relatively short. However, few that are not short, which we call Long-Running streams (tortoises), have lifetimes of hours to days and can carry a high proportion (50% to 60%) of the total bytes on a link. We emphasize that streams can be classified not only by their size (mice and elephants), but also by their lifetime (dragonflies and tortoises). Furthermore, stream size and lifetime are independent dimensions; both are of interest in understanding the overall behaviour of streams in a torrent.

The rich hierarchy of categories used in IP traffic analysis yields many aggregated measures that can serve as foundations for differentiating typical from rare and extreme. Many of these measures are mutually exclusive, which can affect research conclusions unless the disjointness, in particular diversity/disparity and similar phenomena, are explicitly considered. We suggested [10] size disparity as a unifying paradigm shared by seemingly unrelated phenomena: burstiness, scans, floods, flow lifetimes and volume elephants.

We then analyzed concentration properties of byte and packet measures aggregated by IP address, prefix, policy atom and AS. We found that an attempt to faithfully quantify diversity/disparity in Tier 1 backbone data leads to an combinatorial explosion of the parametric space. To reduce the description complexity, we introduced a mice-elephant boundary called *crossover*: the fraction c of total volume contributed by a complementary fraction $1 - c$ of large objects. Studying sources and sinks at two Tier 1 backbones and one university, we found that many IP traffic aggregation categories have crossovers above the proverbial 80/20 split (80% of volume in 20% of sources), mostly around 95/5. Note that less than 20 ASes sent or received 50% of all traffic in both backbone samples, a disparity that could potentially simplify traffic engineering. The proposed concept of *crossovers* may serve as a bridge between the operational and research networking communities, translating a familiar concept into a mathematically precise value.

We also found that the Pareto models, previously used for file/connection/transfer sizes [42] and short-term prefix traffic volumes [43], require a significant bent ($\alpha \sim 0.5$) to account for the size disparity of aggregated and accumulated backbone traffic. On the other hand, a Weibull distribution

with shape parameter of 0.2-0.3 can serve as an alternative model for the tails of AS volume data.

More detailed results, including geotraffic volumes, diversity of objects that contribute over 1% of traffic, consumers of fixed (50, 90, 95, 99) traffic percentile volumes, crossover fractions and cutoffs, volume of mice and distribution plots (all for bytes and packets) are available at [44].

9 Application of Internet spectroscopy techniques to link capacity characterization

In [11] we built on the work in [45] to show that the Radon transform [46] of packet interarrival time distributions, coupled with entropy minimization, can be used for estimation of provisioned bandwidth and for identification of Layer 2 technologies such as ATM, rate-limited ATM, DSL, PPP, Ethernet and cable modems in IP traffic. More generally we have demonstrated the utility of the Internet spectroscopy approach for solving a variety of identification problems. We presented an algorithm that evaluates the interarrival delay quantum (cell time) for rate-limited cell-based links. The algorithm takes a joint 2D distribution of packet sizes and interarrival times as its input, converts it by a coarse-grained Radon transform to a family of 1D marginals. Each marginal has the semantics of an inter- and intra-packet delay (i.e. link idle time) histogram that corresponds to an assumed value of cell time. Our estimate of cell time is the value that minimizes entropy of such a marginal, i.e., makes it closest to a delta function.

As an application of the Radon transform technique, we determined the target cell time for the rate limiting performed by a university commodity ISP to provide a 20 Mbps connection over 155 Mbps link. This measurement allows us to verify any suspected under-fulfillment of the university's service contract. Knowledge of cell time enables us to compute the distribution of inter-packet delay. We find that this delay consists of two separate components overlapping in the time domain: a spike with a width of two cell times that corresponds to the rate limiters fluctuations around the target rate, and the true link idle time. The true idle time integral (ccdf) closely follows a Weibull curve, while individual values are subject to fine-grained delay quantization. We also find that the link's high load renders the packet arrival process quite different from Poisson. Combined with the rate limiter's long-term memory, this deviation makes the byte counting process strictly non-Gaussian over a wide range of aggregation intervals (up to 1 sec).

We analyzed bitrates and other properties of broadband mass-market connections and determined interarrival times for DSL and cable modem sources by a simplified one-dimensional version of the min-entropy Radon algorithm applied to packets of fixed size (40 or 1500 bytes). We found that delay quantization in broadband access infrastructure depends on providers, technologies and markets, but the number of observed spectra is limited. This result suggests that network spectroscopy has a potential for source identification, even down to the host level, if a library of interarrival quanta and interpacket delay distributions is available. To promote network spectroscopy to an operationally useful technology it will be necessary to enable automated analysis of measurement data by algorithms such as those presented thus far. One approach would require creating a library of delay spectra corresponding to known devices and link types. This library would allow recognition of variations in standards implementation specific to different markets and providers. Another obvious direction is the application of these techniques to other connection technologies, and to associate delay quanta with settings that reflect customer provisioning and rate-limiting policies of ISPs. Finally, still needed is an assessment of the accuracy of spectroscopy-based inference and potential for measurement artifacts that distort data and affect results, e.g., timing precision of available monitoring equipment.

10 Nonstationary Poisson View of Internet Traffic

Note: This was cost-shared work with other CAIDA projects.

Since the identification of long-range dependence in network traffic ten years ago, its consistent appearance across numerous measurement studies has largely discredited Poisson based models. However, since that original data set was collected, both link speeds and the number of Internet-connected hosts have increased by more than three orders of magnitude. In [47] we revisited the validity of the Poisson assumption by examining a number of current and historical traces of Internet traffic. We found that at sub-second time scales, backbone traffic appears to be well described by Poisson packet arrivals. Our study provides evidence for how the ongoing pattern of Internet evolution may potentially affect the future characteristics of its traffic. We conjecture that the particular way in which this increase in scale is unfolding seems to be pushing the Internet in the general direction of easier-to-understand and better-behaved traffic models (i.e., the Poisson assumption), or at least not in the direction of sophisticated traffic models. More specifically, based on traces from the MFN and WIDE backbones, we found that up to sub-second time scales, traffic is well characterized by a stationary Poisson model. This result is important because it covers the relevant time scales for the delivery of a single packet (i.e., the Round-Trip Time). Beyond that point, the traffic seems to take on a distinctive form of nonstationary behavior, which consists of short intervals of 'change-free regions' punctuated by sudden change events.

We found that the durations of the change-free intervals were exponentially distributed and uncorrelated, while the change events themselves appeared to be stationary with only a trivial one-step (negative) correlation in the increments. We note that these observations are also consistent with the theoretical results for large-scale aggregations of renewal processes which have been derived under the assumption of scaling the number of sources and network capacity together to keep the normalized offered load fixed. We also show that this type of traffic model (i.e., Poisson with nonstationarity at multi-second time scales) is consistent with the kind of long-range dependence commonly observed in network data over larger time scales. It would be interesting to analyze more data traces from: a) other backbone links, and b) links towards the periphery of the network. It could well turn out that different links exhibit different behavior especially at small time scales. Scaling phenomena especially at small time scales may be sensitive to the traffic mix in terms of applications and the idiosyncracies of low level protocols. This work has also left a number of interesting questions unanswered, which remain as subjects for further study. Most importantly, is the type of nonstationary behavior we see at multi-second time scales sufficient to explain everything, or are there even-larger scale effects remaining to be discovered? Another important open issue is finding the mechanism responsible for the distinctive piecewise-linear variation in the rate. Finally, we found that focusing on the proper time scale is a recurring theme. Although Whitt pointed out that the right time scale must be an increasing function of load placed on a network resource [48], Norros has observed that network traffic sources have the flexibility and intelligence to adapt their transmission policies to the resources currently available in the network [49]. Thus we conjecture that the traffic characteristics for the Internet backbone may continue to grow even better behaved in the future.

11 Visualization of bandwidth estimation data

In the spirit of technoogy transfer, and due to lack of general access to DOE infrastructure by outsiders, we deployed and evaulated the ANEMOS tool developed at Georgia Tech [13] for use on the TeraGrid infrastructure with a point-of-presence at SDSC. As described in [13], ANEMOS shares

characteristics with other network monitoring tools or architectures, such as Pinger [50], Surveyor [51], or the Network Weather Service [52]. One major difference is that ANEMOS provides rules and alarms. Specifically, the system evaluates user-specified rules on the collected data while the measurements are in progress, issuing alarms when rule conditions are satisfied. Another difference is that ANEMOS has been designed for modularity and extensibility, allowing the user to plug-in and use any text-based measurement tool with minimal modifications in the ANEMOS software. Also, the user can request the measurements to be performed either in real-time, or to be scheduled as a batch process. All the interactions with the system are through a Web-based GUI.

We also experimented with other techniques for visualizing interdomain bandwidth measurements, but did not have time to continue work in this area. Figure 11 shows an early example. Visualization of this type of data has not yet received warranted attention from the community.

Broader impact and outreach

12 Caveats and future directions in bandwidth estimation

We list a variety of challenges, both methodological, architectural, and logistic.

- **CAIDA's testbed hosts run FreeBSD**, so our test results reflect how these tools run against the FreeBSD TCP stack. Results against different TCP stacks may vary. It would be interesting to compare our results against similar tests using Linux with Web100 autotuning enabled.

- **There is a strong need for higher quality timestamps on affordable, preferably COTS, monitoring equipment.** Veitch and Pasztor [53] have done the most promising work in this area, but have lacked resources to effect pervasive quality software deployment. Funding agencies should prioritize this type of effort, since it promises dramatic improvements for the entire Internet measurement community.

- **Misconceptions regarding the interpretation and validation of bandwidth estimation measurement abound and persist.** Dovrolis articulated a concise but excellent list of ten such methodological landmines in a 'Ten Fallacies and Pitfalls in End-to-end Available Bandwidth Estimation', presented at Internet Measurement Conference 2004 [54]. We list them briefly, but the paper is essential reading for anyone interested in the future of this field:

  1. ignoring the variability of the available bandwidth process
  2. ignoring the relation between the probing stream duration and the averaging time scale
  3. assuming that 'faster estimation is better'
  4. assuming that packet pairs are as effective as packet trains in measurement
  5. estimating the tight link capacity with end-to-end capacity estimation tools
  6. ignoring the effects of cross-traffic burstiness
  7. ignoring the effects of multiple bottlenecks
  8. assuming that one-way delay statistics can be safely represented as a single value
  9. assuming that iterative probing will converge to a single available bandwidth estimate (as opposed to a range)

10. evaluating the accuracy of available bandwidth estimation through comparisons with bulk TCP throughput

- **Application of bandwidth estimation tools to DOE infrastructure will require concerted resources for systems integration**. Cottrell *et al.* recognize the need to augment DOE measurement infrastructure with new techniques as they are developed and refined. However, there are also substantial software requirements required for automating the identification of significant performance changes and gathering the associated relevant information, e.g. traceroutes before and after a performance change, time and magnitude of the change, topology map, and time series plots of the performance changes [55]. Such functionality tends to be underestimated in both utility as well as cost.

- **Access to real infrastructure for testing tools 'in the wild' is critical and remains a persistent challenge**. Even DOE researchers cannot get access to SNMP counters for their own backbone infrastructure. This is severely hindering research and development progress on measurement tools. Funding agencies need to recognize and navigate (or modulate) this constraint, or the progress will continue to be far below what it could be.

13 Research Community Involvement

On December 9-10, 2003, CAIDA hosted the first Bandwidth Estimation workshop (BEst) supported by CAIDA, IMRG, and the DOE Office of Science [56]. This workshp brought together the most active researchers in the field, as well as some operational networking experts (e.g., from CableLabs), to discuss a range of bandwidth estimation topics, from terminology and metrics definition issues to future research and development priorities in banwidth estimation. Participant surveys indicated that the workshop was a great success. The agenda, slides, and final report are published at: http://www.caida.org/outreach/isma/0312/report.xml.

14 Interactions and collaboration

Over the last three years, our group has collaborated with several other DOE researchers. The main DOE-funded researchers that we often interacted with include Les Cottrell (SLAC), Tom Dunigan and Nagi Rao (ORNL), Brian Tierney, Deb Agarwal, Jin Guojun (LBNL), and Matt Mathis (PSC). These collaborations include scientific discussions at conferences, workshops, and technical meetings, testing of bandwidth estimation tools, sharing of simulation code, sending/receiving comments on research papers, etc.

In addition, kc claffy co-authored a slideset with Les Cottrell and Brian Tierney that was presented at the Large Scale Network meeting on June 10, 2003 at the National Science Foundation. This talk, entitled "priorities and challenges in Internet measurement simulation and analysis" is available on the web at:
http://www.caida.org/outreach/presentations/2003/lsn20030610/.

Georgia Tech has continuously supported the community by making the bandwidth estimation tools developed under this grant (accessible at http://www.pathrate.org). There have been thousands of hits to that site and logs of their tool downloads show that users come from a wide variety of Internet domains (mostly .edu, .net, and .com) as well as from all over the world.

Together with our two bandwidth estimation tools, many users are also familiar with our work through research papers. Publishing papers at major conferences brings visibility to this project,

and to the entire SciDAC program, and convinces users that these measurement tools are based on solid estimation techniques, rather than on questionable heuristics.

Finally, we were present at the SciDAC booth of the SuperComputing 2002 conference, demonstrating the tools and their underlying measurement methodologies.

ACKNOWLEDGEMENTS

We gratefully acknowledge access to the Spirent 6000 network performance tester and Foundry BigIron router in the CalNGI Network Performance Reference Lab created by Kevin Walsh. Many thanks to Cisco Systems for the GSR12008 router, Juniper Networks for the M20 router, and Endace, Ltd. for access to their DAG4.3GE network measurement card. Nathaniel Mendoza, Grant Duvall and Brendan White provided testbed configuration and troubleshooting assistance. Feedback from remote testbed users Jiri Navratil, Ravi Prasad, and Vinay Ribeiro was helpful in refining test procedures. Aaron Turner provided us with a lot of support on installing and running *tcpreplay*. We are grateful to Matthew J Zekauskas for invaluable assistance with running experiments on the Abilene network.

References

[1] R. S. Prasad, M. Murray, C. Dovrolis, and K. Claffy, "Bandwidth Estimation: Metrics, Measurement Techniques, and Tools," *IEEE Network*, Nov. 2003.

[2] T. Lindh and N. Brownlee, "Integrating active methods and flow meters - an implementation using netramet," in *Proceedings Passive and Active Measurements (PAM) workshop*, Apr. 2003.

[3] R. S. Prasad, C. Dovrolis, and B. A. Mah, "The Effect of Layer-2 Store-and-Forward Devices on Per-Hop Capacity Estimation," in *Proceedings of IEEE INFOCOM*, 2003.

[4] C. Dovrolis and M. Jain, "End-to-End Available Bandwidth: Measurement Methodology, Dynamics, and Relation with TCP Throughput," *IEEE/ACM Transactions in Networking*, August 2003.

[5] R. Prasad, M. Jain, and C. Dovrolis, "Effects of Interrupt Coalescence on Network Measurements," in *PAM*, 2004.

[6] A. Shriram, M. Murray, Y. Hyun, N. Brownlee, A. Broido, M. Fomenkov, and k claffy, "Comparison of Public End-to-end Bandwidth Estimation Tools on High-Speed Links," in *PAM 2005, to appear*, Apr. 2005.

[7] R. Prasad, M. Jain, and C. Dovrolis, "Socket buffer auto-sizing for high-performance data transfers," *Journal of Grid Computing: Special Issue on High Performance Networking*, vol. 1, no. 4, 2004. citeseer.ist.psu.edu/634042.html.

[8] X. Liu, K. Ravindran, B. Liu, and D. Loguinov, "Single-hop probing asymptotics in available bandwidth estimation: sample-path analysis," in *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pp. 300–313, ACM Press, 2004.

[9] N. Brownlee and kc claffy, "Understanding internet traffic streams: Dragonflies and tortoises," *IEEE Communications*, vol. 40, Oct. 2002.

[10] A. Broido, Y. Hyun, R. Gao, and k claffy, "Their share: diversity and disparity in IP traffic," in *Proceedings of PAM*, Apr. 2004.

[11] A. Broido, R. King, E. Nemeth, and k claffy, "Radon spectroscopy of inter-packet delay," in *Proceedings of the High-Speed Networking (HSN) Workshop*, June 2003.

[12] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the Self-Similar Nature of Ethernet Traffic (Extended Version)," *IEEE/ACM Transactions on Networking*, vol. 2, pp. 1–15, Feb. 1994.

[13] A. Danalis and C. Dovrolis, "ANEMOS: An Autonomous Network Monitoring System," in *Proceedings of Passive and Active Measurements (PAM) Workshop*, 2003.

[14] San Diego Supercomputer Center , "CalNGI Network Performance Reference Lab (NPRL)." http://www.calngi.org/about/index.html.

[15] L. Jorgenson, "Size Matters: Network Performance on Jumbo Packets," July 2004. http://www.internet2.edu/presentations/jtcolumbus/200407190-MTU-Jorgenson.htm.

[16] K. Keys, D. Moore, R. Koga, E. Lagache, M. Tesch, and k Claffy, "The architecture of CoralReef: an Internet traffic monitoring software suite," in *Passive and Active Network Measurement (PAM) Workshop, Amsterdam, Netherlands*, April 2001.

[17] N. Brownlee, "NeTraMet." http://www.caida.org/tools/measurement/netramet/.

[18] F. Coccetti and R. Percacci, "Bandwidth measurements and router queues," Tech. Rep. INFN/Code-20 settembre 2002, Instituto Nazionale Di Fisica Nucleare, Trieste, Italy, 2002. http://ipm.mib.infn.it/bandwidth-measurements-and-router-queues.pdf.

[19] K. Matoba, S. Ata, and M. Murat, "Improving Bandwidth Estimation for Internet Links by Statistical Methods ," *IEEE Trans. on Communications*, June 2001.

[20] A. Akella, S. Seshan, and A. Shaikh, "An empirical evaluation of wide-area internet bottlenecks," in *Proceedings of the IMC '03*, Oct. 2003.

[21] N. Hu and P. Steenkiste, "Evaluation and Characterization of Available Bandwidth Probing Techniques," *IEEE Journal on Selected Areas in Communications, JSAC Special Issue on Internet and WWW Measurement, Mapping, and Modeling*, vol. 21(6), August 2003.

[22] J. Strauss, D. Katabi, and F. Kaashoek, "A measurement study of available bandwidth estimation tools," in *IMW*, 2003.

[23] N. Hu and P. Steenkiste, "Towards Tunable Measurement Techniques for Available Bandwidth," Dec. 2003. `http://www.caida.org/outreach/isma/0312/abstracts/hu.pdf`.

[24] R. L. Cottrell and C. Log, "Overview of IEPM-BW Bandwidth Testing of Bulk Data Transfer," in *Supercomputing 2002*, Nov. 2002.

[25] S. Ubik, V. Smotlacha, and N. Simar, "Performance monitoring of high-speed networks from the NREN perspective," in *TERENA Networking Conference, Rhodes, Greece*, June 2004. http://staff.cesnet.cz/ ubik/publications/2004/terena2004.pdf.

[26] Spirent Corporation, "Smartbits 6000B." http://spirentcom.com/analysis/view.cfm?P=141.

[27] Spirent Corporation, "Smartflow." Documentation at http://spirentcom.com/analysis/view.cfm?P=119.

[28] NLANR, "NLANR Passive Measurement Analysis (PMA) Datacube," August 2004. http://pma.nlanr.net/Datacube/.

[29] A. Turner, "tcpreplay 2.2.2 - a tool to replay saved tcpdump files at arbitrary speed," July 2004. http://tcpreplay.sourceforge.net/.

[30] A. Shriram, M. Murray, Y. Hyun, N. Brownlee, A. Broido, M. Fomenkov, and k claffy, "Comparison of Public End-to-end Bandwidth Estimation Tools on High-Speed Links," Oct. 2004. extended report.

[31] J. Navratil, "ABwE: A Practical Approach to Available Bandwidth," in *PAM*, 2003.

[32] R. Carter and M. Crovella, "Measuring Bottleneck Link Speed in Packet-Switched Networks," Tech. Rep. 96-006, Boston University, 1996.

[33] G. Jin, "netest-2." http://www-didc.lbl.gov/NCS/netest.html.

[34] V. Ribeiro, "pathChirp: Efficient Available Bandwidth Estimation for Network Path," in *PAM*, 2003.

[35] G. Jin, G. Yang, B. R. Crowley, and D. A. Agarwal, "Network Characterization Service (NCS)," tech. rep., LBNL, 2001.

[36] M. Jain and C. Dovrolis, "Pathload: an available bandwidth estimation tool," in *PAM*, 2002.

[37] "Iperf." http://dast.nlanr.net/Projects/Iperf.

[38] L. Cottrell, "Internet End-to-End Performance Monitoring: Bandwidth to the World (IEPM-BW) project," tech. rep., SLAC - IEPM, June 2002. http://www-iepm.slac.stanford.edu/bw/.

[39] SLAC, "Internet End-to-end Performance Monitoring - Bandwidth to the World (IEPM-BW) Project," August 2004. http://www-iepm.slac.stanford.edu/bw/.

[40] A. Pasztor and D. Veitch, "Active Probing using Packet Quartets," in *Proceedings Internet Measurement Workshop (IMW)*, 2002.

[41] C. Dovrolis, P. Ramanathan, and D. Moore, "Packet Dispersion Techniques and Capacity Estimation," *IEEE/ACM Transactions on Networking*, Jan. 2005.

[42] M. E. Crovella and A. Bestavros, "Self-similarity in World Wide Web traffic. Evidence and possible causes," in *IEEE/ACM Transactions on Networking*, 1997.

[43] K. Papagiannaki, N. Taft, and C. Diot, "Impact of flow dynamics of traffic engineering principles," in *INFOCOM*, 2004.

[44] A.Broido, Y.Hyun, R.Gao, and kc claffy, "Their share: diversity and disparity in IP traffic (Supplement to PAM 2004 submission)," 2004. http://www.caida.org/analysis/workload/diversity.

[45] A. Broido, E. Nemeth, and k claffy, "Spectroscopy of DNS Update Traffic," in *Proceedings of ACM SIGMETRICS*, June 2003.

[46] E. Weisstein, "Radon Transform," 1999. http://mathworld.wolfram.com/RadonTransform.html.

[47] T. Karagiannis, M. Molle, M. Faloutsos, and A. Broido, "A nonstationary poisson view of internet traffic," in *Proceedings of IEEE INFOCOM*, Apr. 2004.

[48] K. Sriram and W. Whitt, "Characterizing Superposition Arrival Processes in Packet Multiplexors for Voice and Data," *IEEE J. Select. Areas Communcations*, vol. 4, 1986.

[49] L. Norros, "On the Use of Fractional Brownian Motion in the Theory of Connectionless Networks," *IEEE J. Select. Areas Communcations*, vol. 13, 1995.

[50] W. Mathews and L. Cottrell, "The PingER project: Active internet performance monitoring for the HENP community," *IEEE Communications*, vol. 38, pp. 130–136, May 2000.

[51] S. Kalidindi and M. Zekauskas, "Surveyor: An infrastructure for internet performance measurements," in *Proc. INET'99*, 1999.

[52] R. Wolski, N. Spring, and C. Peterson, "Implementing a performance forecasting system for metacomputing: the network weather service," in *Proc. of Supercomputing*, 1997.

[53] A. Pasztor and D. Veitch, "PC Based Precision Timing Without GPS," in *Proceedings of ACM SIGMETRICS*, 2002.

[54] M. Jain and C. Dovrolis, "Ten fallacies and pitfalls on end-to-end available bandwidth estimation," in *IMW*, 2004.

[55] C. Logg, L. Cottrell, and J. Navratil, "Correlating Internet Performance Changes and Route Changes to Assist in Troubleshooting from an End User Perspective," in *Proceedings of PAM 2004*, 2004. http://www.pam2004.org/papers/285.pdf.

[56] CAIDA, "Internet Statistics and Metrics Analysis workshop on Bandwidth Estimation Methodologies," Dec. 2003. `http://www.caida.org/outreach/isma/0312/`.

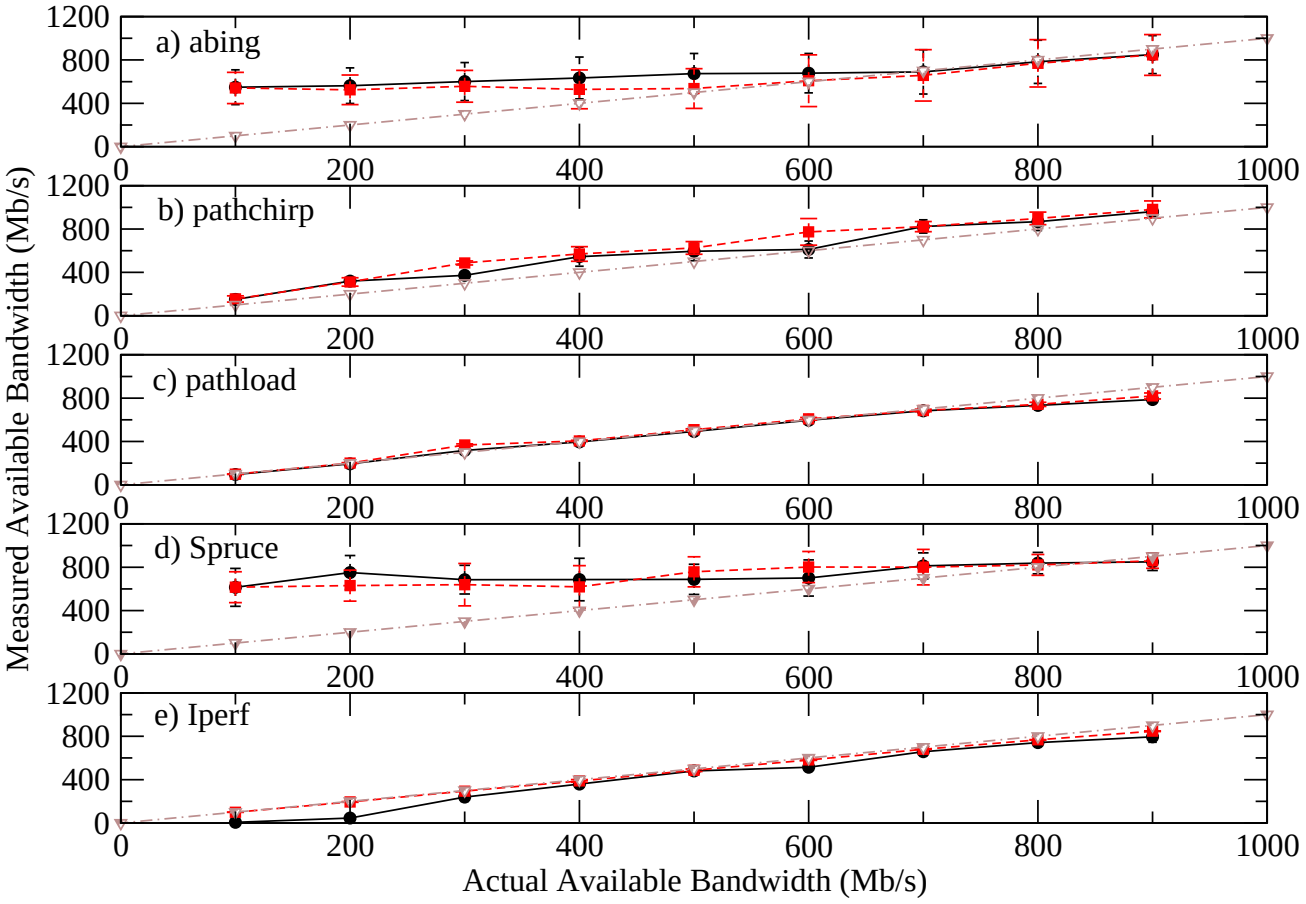

Figure 4: Comparison of available bandwidth measurements on a 4 hop OC48/GigE path loaded with synthesized cross-traffic. For each experimental point, the x-coordinate is the actual available bandwidth of the path (equal to the GigE link capacity of 1000 Mb/s minus the generated load). The y-coordinate is the tool reading. Measurements of the end-to-end path in both directions are shown. The dash-dotted line shows expected value.

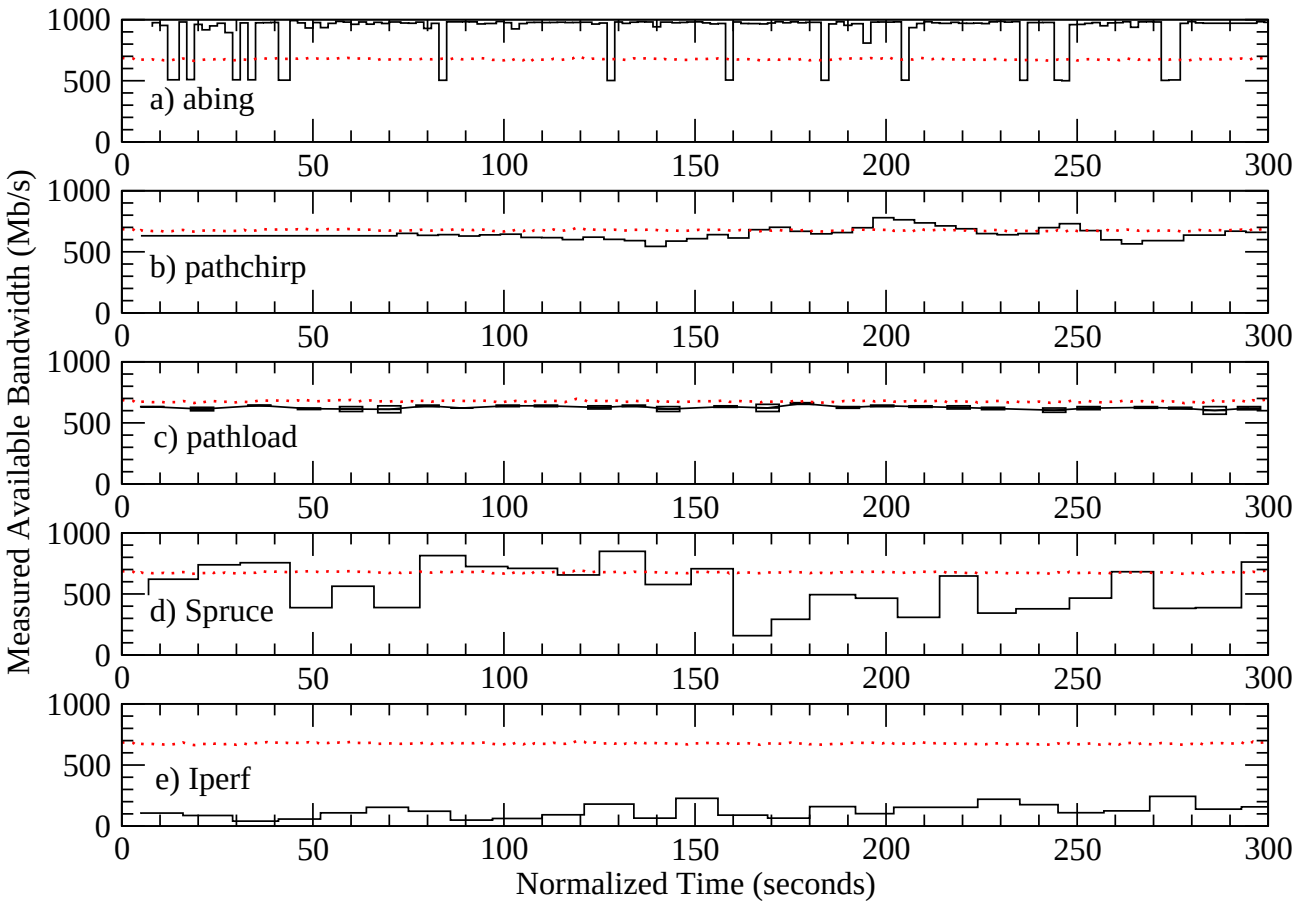

Figure 5: Comparison of available bandwidth tool measurements on a 4 hop OC48/GigE path loaded with played back real traffic. The X-axis shows time from the beginning of trace playback. The Y-axis is the measured available bandwidth reported by each tool. The dotted line shows the actual available bandwidth that was very stable on a one second aggregation scale.

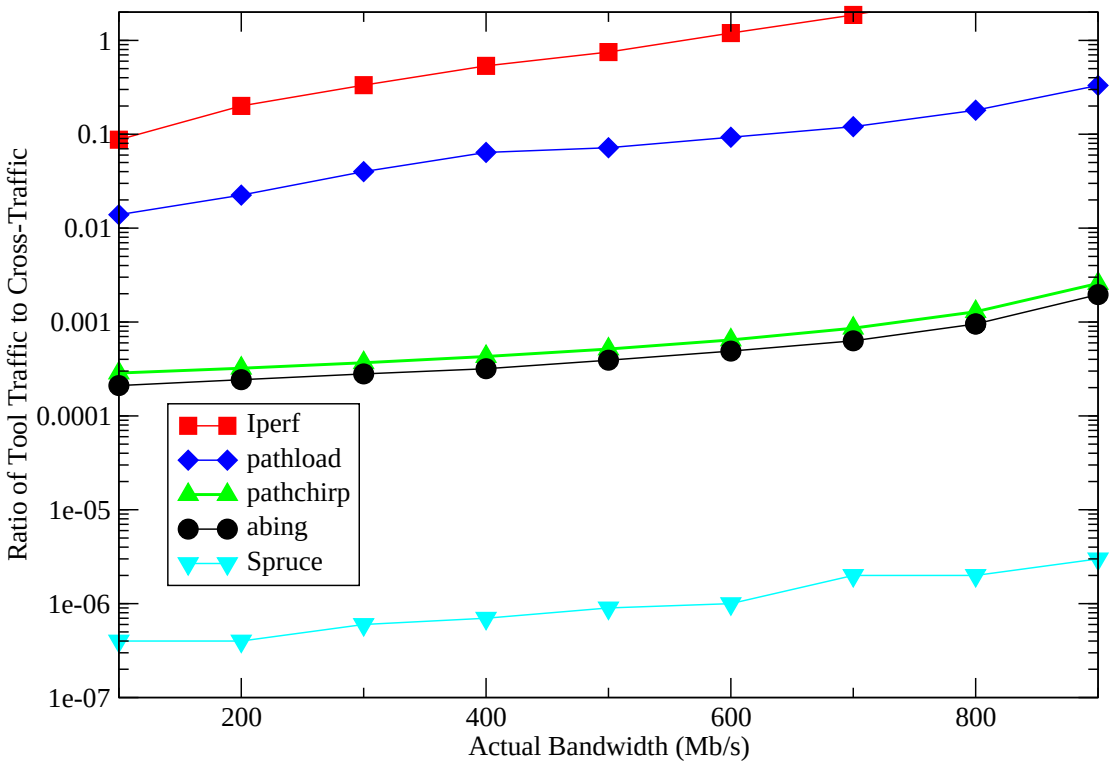

Figure 6: Tool overhead vs. available bandwidth. Note that *pathchirp*, *abing*, and *spruce* exhibit essentially zero overhead.

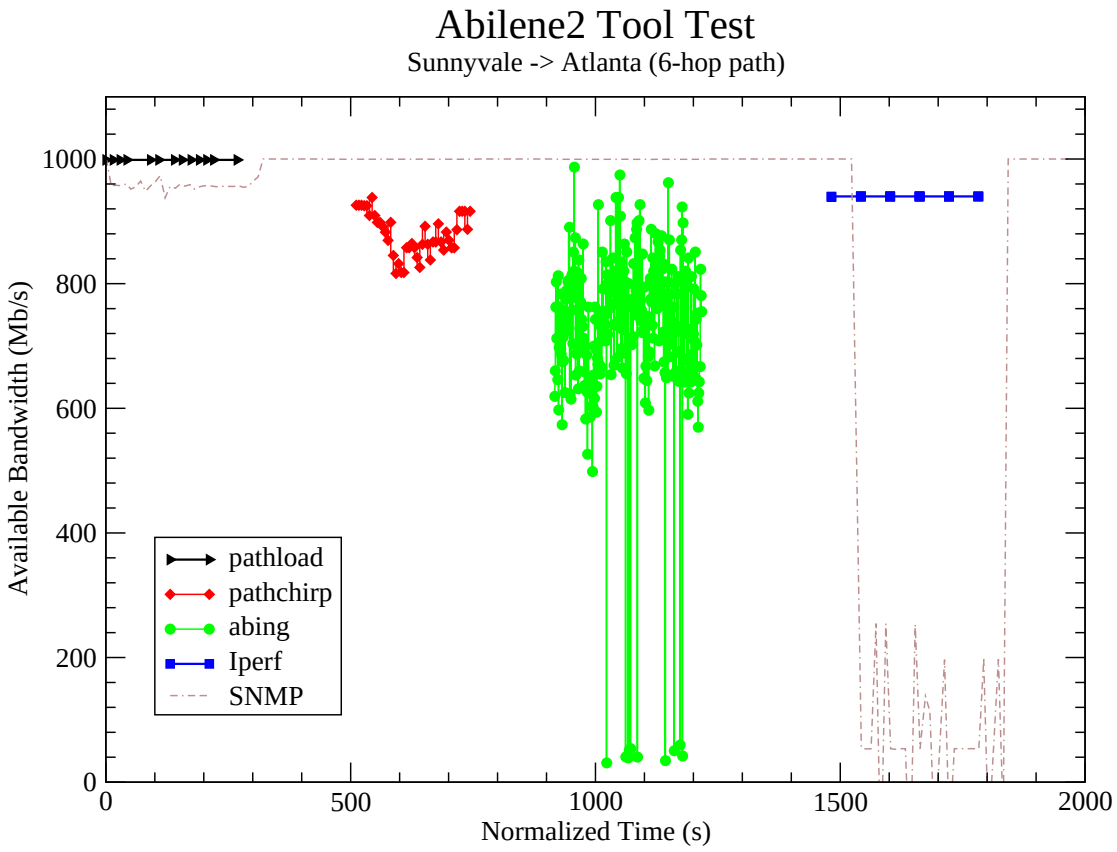

Figure 7: Real world experiment conducted on the Abilene network. The dashed line shows the available bandwidth derived from SNMP measurements. See explanations in the text.

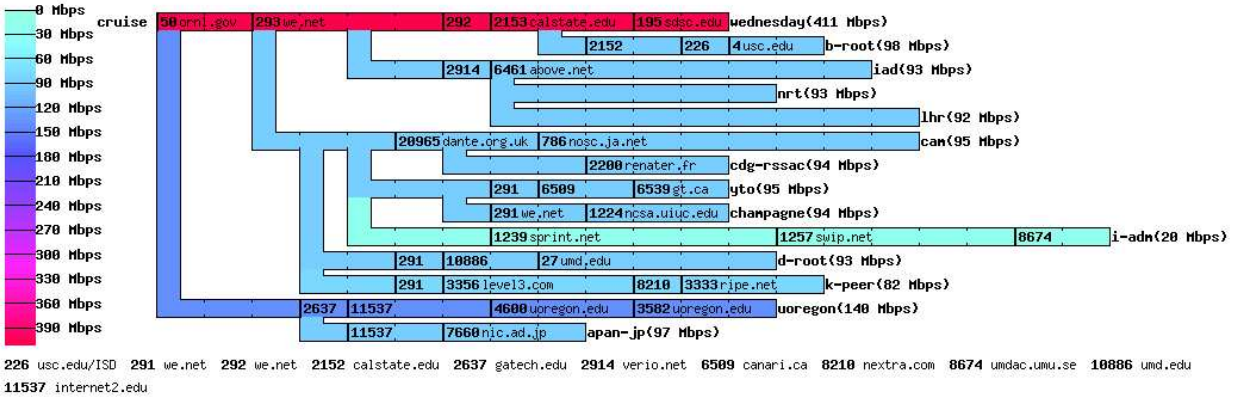

Figure 8: Visualization of available bandwidth estimates across a number of different automonomous systems, for a number of different paths.