Reverse Engineering the Internet*

Neil Spring, David Wetherall, and Thomas Anderson
Department of Computer Science and Engineering, University of Washington

Abstract— **To provide insight into Internet operation and performance, recent efforts have measured various aspects of the Internet, developing and improving measurement tools in the process. In this paper, we argue that these independent advances present the community with a startling opportunity: the collaborative reverse-engineering of the Internet. By this, we mean annotating a map of the Internet with properties such as: client populations, features and workloads; network ownership, capacity, connectivity, geography and routing policies; patterns of loss, congestion, failure and growth; and so forth. This combination of properties is greater than the sum of its parts, and exposes the attributes of network design easily overlooked by simpler, uncorrelated models. We argue that reverse engineering the Internet is feasible based on continuing improvements in measurement techniques, the potential to infer new properties from external measurements, and an accounting of the resources required to complete the process.**

1 Introduction

Reverse engineering is the process of learning the design of an object by studying its implementation. For the Internet, we take "design" to mean how its components (links, routers, clients, and networks) are assembled and configured. An individual ISP can use private information to study its own network, sometimes publishing the result, but the heterogeneity and number of different ISPs prevents us from applying the same techniques to the whole Internet or assuming that the results from a single ISP would generalize.

The importance of quantitative data on the composition and operation of the Internet has been broadly recognized in the research community [11], yet the challenges of obtaining it by reverse-engineering are daunting. The Internet comprises millions of hosts and routers and is constantly adapting to an ever-changing workload of applications and traffic, so the process of reverse-engineering must be fast enough to observe a consistent picture. Because ISPs provide little support for observing the structure and state of their networks directly, new techniques must be devised to infer internal details from limited externally available information.

We argue that, with a community effort, it is possible to reverse-engineer the Internet in a single day of concerted measurement. These measurements can be repeated on a regular basis to provide an ongoing view of the structure and operation of the Internet. Recent advances in efficient network measurement techniques make this possible. For example, tomography can be used to measure link loss passively [29], correlating BGP routing updates can expose topology [5], tailgating approaches measure link bandwidth using few packets [21, 31], decoding DNS names can expose geography [30], and avoiding redundant measurements can make topology studies practical [37].

The challenge is to use these techniques to produce new measurement tools that are efficient enough to run quickly at the scale of the Internet. Skitter [9] shows that an individual property, IP connectivity, can be measured at Internet-scale with reasonable accuracy and on a regular basis. We seek to extend this map with a richer set of properties. We describe promising approaches to measuring and inferring missing properties of individual routers or links including failure, utilization, packet duplication and corruption, layer 2 topologies, and the location of clients. Measurement tools often probe the links along a single network path; we describe how to optimize these tools to measure regions of the network efficiently.

The benefits of a collaborative effort over existing piecemeal projects are significant. First, coordinated measurements expose relationships between properties that cannot otherwise be analyzed. For example, studies of the IP-level topology have shown heavy-tailed degree distributions with the implication that the IP graph may have a low attack-tolerance [1]. However, observing router role together with node degree shows this implication is misleading because most of the high outdegree nodes are relatively unimportant access routers rather than part of an ISP backbone. Second, relationships between properties can be used to derive or check further properties. For example, a model of network routing results from a set of chosen paths combined with a topology [36]. Third, coordinated measurement can save substantial redundant or unnecessary work. For example, measurements that show that router IP addresses are in different geographic locations also show that those IP addresses are not aliases, which would make the alias resolution process more efficient. Lastly, a community effort will facilitate the validation that is needed before results can be used with confidence. Those with internal information, such as ISP operators, can detect errors and provide feedback about where

*Appears in 2nd Workshop on Hot Topics in Networks (HotNets-II), November, 2003.

tools work and where they fail. And with a "live" repository that supports the ongoing collection and analysis of measurements, different researchers can compare different methods and repeat experiments at different times and over different ISP networks.

Table 1 is a roadmap for this paper. It lists the properties that expose network design – what we think constitutes a reverse-engineering of the Internet. In the next section, we outline the state-of-the-art in Internet measurement: those properties that can be measured today. In Section 3, we sketch approaches to estimating the remaining properties. We argue in Section 4 that these measurements can be made efficient enough that they can be run every single day. Finally, we summarize and discuss collaborative Internet measurement in Section 5.

2 State of the Art

In this section, we briefly describe what can be measured and analyzed today. While the methodologies we list have been published and shown to work, they have generally been tested only on a few paths, in simulation, or on subsets of the topology. A general challenge for reverse-engineering is to validate these techniques and generalize them to the scale and heterogeneity of the Internet.

2.1 Measured Node Properties

IP aliases Traceroute lists interface addresses, and different interface addresses for the same router must be resolved. The Rocketfuel tool, Ally [37], builds on the Mercator [15] alias work, using topology and DNS to guide the search for alias pairs.

Geography The placement of routers in cities exposes POP structure and backbone interconnections. GeoCluster [30] uses databases to infer node location. DNS names work for routers, but the database is tedious to maintain.

Owner Isolating networks by their administration allows separately engineered networks to be analyzed independently. Mao *et al.* [24] provide techniques to determine the autonomous system responsible for a router and host.

Router role Routers serve different purposes: some are backbone routers that connect to other POPs, others are access routers that aggregate customers. Rocketfuel used DNS and topological ordering to identify roles.

Implementation features TCP features, which are tracked by tbit [28], and services supported, which can be measured with nmap, affect the composition of traffic and show how quickly features are deployed. We do not expect it to be practical to run nmap to every host in today's security conscious Internet. It is an open question whether such techniques can be extended to infer router configuration parameters (e.g. for RED).

2.2 Measured Link Properties

Loss Packet loss has been inferred in a large scale study by Padmanabhan, *et al.* [29] using tomography. Tomography assigns loss rates to links using measured routing and end-to-end observed loss at a busy Web server. This technique holds promise for locating other rare events like duplication and corruption. Although efficient, tomography requires passive observation points that may be difficult to find, so active probing using Tulip [23] may be required instead, at least to measure some links.

Reordering Although not a performance problem in itself, reordering indicates fine-grained multi-path routing or a topology that includes parallel links. Tulip [23] builds on the techniques of Sting [6] to measure link reordering.

Delay End-to-end one-way trip times (OTT) can be measured with synchronized clocks, but for the delay between routers, geographic distance is an inexpensive approximation [36, 39]. Measurements may refine these estimates, but must use routing knowledge to account for asymmetric return paths.

Delay variation The variation in delay caused by queuing in contention with other flows is both a problem for time-critical traffic and an indication of congestion in the network. The cing [4] tool uses ICMP timestamps to estimate the delay variation of path segments.

Capacity Two methods can measure the capacity of a link: variable packet size and tailgating packet pair. Variable packet size methods (pchar [22], clink [12]) are traditional but require thousands of packets. Packet quartets [31] and cartouche probes [17] refine nettimer [21] and measure capacity more cheaply. We see advances in capacity measurement as evidence that network measurement techniques are constantly improving.

2.3 Measured Topology Properties

Topology Four levels of Internet connectivity have been studied. RouteViews [26] data includes the graph of connections between autonomous systems. Skitter [9] periodically collects the topology between IP addresses. Mercator [15] resolves IP aliases to construct a router-level topology. Finally, Rocketfuel [37] groups routers by geography to provide a POP-level (backbone) topology.

Routing Routing policy affects how packets are directed, and policies that differ from the default indicate traffic engineering. Policy is applied both at the IP level [36], which relates directly to performance and traffic engineering, and at the autonomous system level [14], which expresses the business relationships in the network. Both policies can be inferred by observing alternate paths that are not chosen, either because of intra-domain link metrics or inter-domain policy.

Workload Traffic matrices – how much traffic is exchanged between network edges – shape the evolution of the network. The tomogravity [40] approach provides a

Property	Tool	Technique	Packets and dependencies §4
Node (Host and Router) Properties			
IP aliases	ally [37]	Four-packet test for same IP ID counter, to TTL-nearby addresses	$5\times\mathcal{A} + 0.005 \times 4(0.9\mathcal{A})^2$
Geography	GeoCluster [30]	Groups prefixes by topology, BGP	*database:* 0
Router role	Rocketfuel [37]	DNS feature extraction, topology	*database:* 0
Owner	Mao *et al.* [24]	DNS, BGP, whois	*database:* 0
Implementation features	tbit [28]	TCP behavior inference	126 $\times\mathcal{N}$
	nmap [13]	TCP port scan	1658 $\times\mathcal{N}$
Node failure	§3.1	Probe IP-ID every minute	24 hours $\times$ 60 min. $\times$ $\mathcal{N}$
Link Properties			
Loss	tomography [29]	Observe TCP retransmissions	*tomographic:* 0
	Tulip [23]	Observe IP ID sequences	(200 (small) + 100 (big)) $\times$ $\mathcal{L}$
Reordering	Tulip [23]	Observe IP ID ordering in responses	200 $\times$ $\mathcal{L}$
Delay	geography [39]	Geographic distance $\div$ speed of light	Geography
Delay variation	cing [4]	ICMP timestamp requests sent to routers	(estimated) 1000 $\times$ $\mathcal{L}$
Capacity	quartets [31]	Tailgating-based packet train	40 (big packets) $\times$ $\mathcal{L}$
Link failure	§3.1	Correlate route changes with policy to find infinite cost links	Routing, Topology
Idle capacity	§3.2	Extend pathload [20], chirps [35]	5 retries $\times$ 100 (big) $\times$ $\mathcal{L}$
	§3.2	Model queuing	Delay variation, Capacity
Utilization	§3.2	Capacity – Idle capacity	Capacity, Idle capacity
Layer 2 Connection	§3.3	Point-to-point vs. multiple access by address allocation	Topology
Layer 2 Switches	§3.3	Difference [34] between capacity measurements	Capacity
Duplication	§3.4	Observe repeated IP identifiers	*tomographic:* 0
Corruption	§3.4	Observe bad TCP checksums	*tomographic:* 0
Topology (Graph) Properties			
Topology (AS)	RouteViews [26]	Archive BGP tables	*database:* 0
Topology (IP)	Skitter [9]	Traceroute to all destinations	$2 \times \mathcal{S} \times \mathcal{A}$
Topology (Router)	Mercator [15]	IP aliases and IP topology	Topology (IP), IP aliases
Routing (IP)	Rocketfuel [36]	Constraint solver infers metrics	Topology (Router)
Routing (AS)	Gao *et al.* [14]	Find AS relationships in BGP tables	*database:* 0
Client location	§3.5	Analyze prefix density [25] passively, with BGP dynamics [5] or topology	*tomographic:* 0
Workload	tomogravity [40]	Traffic matrix estimation using gravity and tomography models	Client location, Utilization, Capacity, Routing
Overall	$3224\mathcal{N} + 1400\mathcal{L} + 640(\text{big packets})\mathcal{L} + 0.016\mathcal{A}^2 + 2\mathcal{S}\mathcal{A}$ = 5 billion packets, 803 GB		
Per source	27 million packets, 4.0 GB, 372 Kbits/s = $37 per month per source		

Table 1: Properties that expose network design and approaches to their measurement. $\mathcal{L}$ is the number of router to router or router to host links to measure; there are 786,000 IP to IP links in Skitter, and we estimate there are two-thirds as many (500,000) router to router links. $\mathcal{A}$ represents the addresses, 419,752 in Skitter, and $\mathcal{N}$ the number left after alias resolution (we assume $\mathcal{N}$=$\mathcal{A}$). $\mathcal{S}$ is the number of sources or vantage points, 200 combining Skitter, Surveyor, PlanetLab, RON, and NIMI. *Tomographic* analyses are passive combined with routing policy and topology. *Database* measurements require at most $\mathcal{A}$ packets but usually zero. We list tools to show that properties can be measured, not to endorse certain tools over others. Section numbers represent research challenges discussed in this paper. The totals at the bottom are calculated without optimizations discussed in Section 4, and the cost estimate is based on 1Mbps $\approx$ $100 per month [16].

technique to estimate workload from measurable quantities, but first we must measure link utilization and the location of clients, described in the next section.

3 Inferring New Properties

The next challenge we discuss is estimating the remaining properties in Table 1. We argue that it is possible to measure failure, utilization, layer 2 topology, duplication, corruption, and client location, although much research remains to address practical issues and validate techniques.

3.1 Failure and Evolution

Failures generally last only a few minutes [18], so externally observing failures, let alone determining where they occur, is daunting. However, as yet another instance of heterogeneity in the Internet, some links are more failure prone than others [18], which means that if we can identify the links and routers that are likely to fail, we can focus measurements on those.

Several primitives can detect node and link failures. A simple approach to detecting long term failure is to analyze changes in the measured network: removal and replacement of a link or router suggests a temporary failure. To measure short-term failures, however, new approaches are needed. First, when packets traverse a path that is inconsistent with routing policy, a failure may be the cause. Inferring the routing policy during the failure may show a link (or set of links around a failed node) as having "infinite" cost – exposing the failed component. This extends the approach of Chandra *et al.* [8]. To detect node failure explicitly, probe traffic could monitor routers to observe the IP identifier. The IP identifier (a unique value that helps fragment reassembly) is generally implemented as a counter [7], which may be cleared when a router is rebooted or powered off. In practice, the rate of change of the IP identifier varies from router to router, and large spikes (perhaps due to routing protocol exchanges – another indication of change) may conceal a fast reboot. We believe that future measurement studies can resolve these issues to detect failure.

3.2 Utilization and Workload

The utilization of a network link is the difference between its total capacity and the available (idle) capacity. As utilization alone has limited value, we focus on measuring it along with total link capacity. Tools exist to measure link capacity (e.g. pchar [22]) and the bottleneck available capacity of a path (e.g. pathload [20]), but not yet the available capacity of individual links.

As a dynamic property with troublesome statistical properties [32], the utilization of a link is likely to be difficult to measure. There are two possible approaches to estimating utilization. First, pathload may be extended using tailgating. Pathload uses an adaptive search to detect when it barely fills the idle capacity of a path. The available capacity of links before the bottleneck may be measured by modifying pathload's link-filling traffic to expire at various hops in the network while small tailgating packets continue on to a receiver. Links downstream of the bottleneck may be mapped by combining the results of other vantage points. Second, tools such as cing [4] and tulip [23] measure delay variation and loss to a router. It may be possible to analyze the distribution of these samples to estimate utilization. Alouf *et al.* [3] present an approach that models a link as an M/D/1/K queue, and uses queue length (which can be sampled by delay variation), loss rate, and capacity to solve for the capacity of the queue and link utilization. Results using these techniques can be validated against known workloads from cooperating ISPs.

3.3 Below IP

Internet mapping efforts that use traceroute are limited in their ability to discover the real, physical topology underlying the IP-level topology: MPLS, multi-access LANs, and other switched networks can appear as a mesh of virtual point-to-point links. It is important to identify these switched networks so that they do not hide the true properties of the media.

The immediate goal in understanding layer two topologies is to simplify measured network maps that include switched networks. Common address allocation policies may provide a solution. Point-to-point links in the core of the Internet (not dial-up PPP) are assigned addresses from /30 prefixes – address blocks with one address for either end of the link, a network address, and a broadcast address. As a result, in a network full of point to point IP links, one half of the addresses will be observed as unavailable (those that end in 0 and 3 modulo 4). Observing an interface address that is not part of a point to point link indicates that a larger address block (such as a /29 or /28) is being used by a shared or switched network. This method would not detect a "switch" in the middle of a point-to-point link, but the result should simplify measured maps.

Also of interest is the underlying network switch topology. Prasad *et al.* [34] showed that store-and-forward switches have an observable effect on pathchar-like tools. The extra store and forward stage represented by an "invisible" switch doubles the extra serialization delay observed when increasing the size of single packet probes. This makes the link appear only half as fast as it is. If any tool measures greater available bandwidth than link capacity, a switch is indicated. Similarly, one might use traffic designed to contend for only certain segments of a hypothesized switched link, as proposed by Coates *et al.* [10]. Similarly, differences between geographic latency and measured link latency can imply detour routing at the link layer.

3.4 Link Pathologies

Duplication and corruption are so rare in the network that their active measurement is likely to be too expensive. Passive measurement with tomography provides an alternative. Although the outbound path from a Web server can be measured with traceroute and loss observed through its recovery in TCP (as by Padmanabhan [29]), duplication and corruption are only visible on inbound traffic. With the collaborative reverse-engineering we propose, the path that inbound duplicated or corrupted traffic traversed can be inferred from other measurements. This means that a modest number of passive packet monitors at selected sites may be able to infer link pathologies, enhancing the completeness of the reverse engineering effort.

3.5 Client location

Clients create a demand for traffic. Although network mapping projects focus on collecting an accurate picture of the core of the network, clients ultimately shape the network – more clients create more demand, inducing ISPs to add capacity. Andersen *et al.* [5] showed that prefixes that share BGP behavior mirror the underlying topology. BGP updates are already collected and stored at several sites. The traceroutes collected as part of network mapping could also be used to determine where clients attach in the topology.

Knowing how much address space lies in a location alone is insufficient for suggesting their demand on the network. For example, smaller prefixes appear to be more densely populated with clients and servers [25]. As a result, knowing where prefixes connect should be augmented with a estimate of the number of active hosts attached to the network – potentially from an unobtrusive, but possibly biased, passive analysis at Web servers.

4 Efficient Measurement at Scale

Once we can approximate each of the properties in Table 1, the next challenge will be to scale to the entire Internet. The amount of traffic generated by each vantage point limits how quickly or how often reverse-engineering may be run. Table 1 provides a starting point for considering this workload. It shows that with present techniques, roughly 25 million packets are needed per vantage point. Surprisingly, this number is already within reach: it is only 4 times the traffic sent by Skitter today. Our goal is to make reverse-engineering clearly practical and acceptable by reducing this workload until it is comparable to that of Skitter. There is much scope for doing so because existing techniques were typically not designed to be efficient for Internet mapping.

Table 1 shows that it is inexpensive to estimate those properties that use passive analysis, database lookups, the results of other tools, or modest traffic to each address. The properties to be concerned about are topology, IP aliases, and link properties.

Topology can be measured using fewer packets than the 6.5 million used by Skitter, opening the door for a richer map to be measured with comparable traffic. Instead of running a basic traceroute at each vantage point to each of 500,000 destinations, if each vantage point ran its traceroute backwards until it merged with an already discovered part of the tree, it would take only 700,000 packets (the number of traceroute destinations plus the number of addresses a source discovers). Our goal is to estimate the remaining properties using the 5.8 million packets saved.

Alias resolution is the most expensive analysis in Table 1, consuming 53% of the packets even after simple improvements. In Rocketfuel [37], we used the TTL remaining in responses from routers as evidence that two addresses are not aliases. With additional vantage points, more alias pairs can be disproven cheaply – using five vantage points, we were able to eliminate 99.5% of alias pairs within an ISP, but looking at the entire Internet or using more vantage points may eliminate more. Both Mercator and Rocketfuel observed that 10% of router addresses were unresponsive to alias resolution – this further reduces the number of pairs that must be tested to $(0.9\mathcal{A})^2$. We expect that the number of addresses considered can be reduced further by, for example, removing all addresses with Web server hostnames. Finally, topology and aliases change together, so changes in topology may drive which addresses are tested as aliases.

The second largest term, $3224\mathcal{N}$, or 25% of the packets, are spent in learning the properties of nodes. Because these properties are slowly changing, it may be sufficient to verify that they are unchanged – for tbit, that a host still does not support ECN or for nmap, that the Web server is still available.

Measuring link properties uses 19% of the packets but 65% of the bytes. We assume that network link measurement tools are easily modified to measure individual links or trees leaving a source without repetition. Most of the overhead of link measurement is the cost of running pathload for each link in the network. It may be possible to guide (and therefore shorten) pathload's adaptive search for a sustainable rate based on past link utilization and recently measured capacity.

With these techniques – verifying past information instead of measuring anew, avoiding redundant measurement, and guiding the analysis with measurements of other properties – we hope to improve the efficiency of reverse engineering to be comparable to that of Skitter.

5 Looking Ahead

In this paper, we have argued that reverse-engineering the Internet is within reach of the research community, and that a collective effort would achieve significantly more than the isolated efforts of individual researchers. We next describe what such a collective effort would look like the challenges it would face.

We envision a "measurement blackboard" that supervises the execution of measurements, archives and aggregates their results, and exports raw data and summary views including network topologies. Unlike Route Views and Skitter, which have shown the value of continuously archived wide-area measurements, the measurement blackboard includes such diverse information as the performance of links, the location of nodes, and routing policies, as well as raw measurements in the form of packet traces and traceroute output. Unlike the Internet Traffic

Archive [19] and the proposed SIMR system [2] for indexing and archiving diverse Internet measurements, new measurements will build on the fresh, intermediate results of others. To support such tightly coupled measurement, the measurement blackboard will execute measurement and inference programs that researchers write to a common API. Scriptroute [38] running on PlanetLab [33] provides a suitable large-scale measurement platform. This architecture of researchers contributing extensions to a narrow core is similar to the way the network simulator *ns* [27] has facilitated improved experimentation and comparison.

The main challenge we have explored in this paper is the design of techniques that can estimate network properties efficiently at the scale of the Internet. Another challenge is the construction of the measurement blackboard for exchanging and archiving measurements. Beyond this work by researchers, there are challenges that arise from others who are inadvertently involved. Site administrators using intrusion detection systems may mistake measurement traffic for the prelude to an attack, even when traffic is sent responsibly and poses no real threat. The challenge for researchers is to engineer a consensus as to what constitutes responsible measurement. Although ISPs stand to benefit from reverse-engineering efforts through improved diagnostic tools and operational knowledge, they continue to be wary about the competitive disadvantages of publishing what is considered proprietary information. The challenge for researchers is to provide sufficient operational benefit to discourage ISPs from blocking measurements and encourage them to publish more data on their own.

References

[1] R. Albert, H. Jeong, and A.-L. Barabási. Error and attack tolerance of complex networks. In *Nature*, vol. 406, pp. 378–382, 2000.

[2] M. Allman, E. Blanron, and W. M. Eddy. A scalable system for sharing Internet measurement. In *Passive & Active Measurement (PAM)*, 2002.

[3] S. Alouf, P. Nain, and D. Towsley. Inferring network characteristics via moment-based estimators. In *IEEE INFOCOM*, 2001.

[4] K. G. Anagnostakis, M. B. Greenwald, and R. S. Ryger. cing: Measuring network-internal delays using only existing infrastructure. In *IEEE INFOCOM*, 2003.

[5] D. G. Andersen, N. Feamster, S. Bauer, and H. Balakrishnan. Topology inference from BGP routing dynamics. In *ACM SIGCOMM Internet Measurement Workshop*, 2002.

[6] J. Bellardo and S. Savage. Measuring packet reordering. In *ACM SIGCOMM Internet Measurement Workshop*, 2002.

[7] C. Bormann, *et al.* Robust header compression. RFC 3095, 2001.

[8] B. Chandra, M. Dahlin, L. Gao, and A. Nayate. End-to-end WAN service availability. In *USITS*, 2001.

[9] k. claffy, T. E. Monk, and D. McRobb. Internet tomography. In *Nature*, 1999.

[10] M. Coates, R. Castro, and R. Nowak. Maximum likelihood network topology identification from edge-based unicast measurements. In *ACM SIGMETRICS*, 2002.

[11] Computer Science and Telecommunications Board, National Research Council. *Looking Over the Fence at Networks: A Neighbor's View of Networking Research*. The National Academies Press, 2001.

[12] A. B. Downey. Using pathchar to estimate Internet link characteristics. In *ACM SIGCOMM*, 1999.

[13] Fyodor. NMAP: The network mapper. http://www.insecure.org/nmap/.

[14] L. Gao. On inferring autonomous system relationships in the Internet. In *IEEE Global Internet Symposium*, 2000.

[15] R. Govindan and H. Tangmunarunkit. Heuristics for Internet map discovery. In *IEEE INFOCOM*, 2000.

[16] J. Gray. Distributed computing economics. IEEE TFCC Newsletter, 2003. http://www.clustercomputing.org/content/tfcc-5-1-gray.html.

[17] K. Harfoush, A. Bestavros, and J. Byers. Measuring bottleneck bandwidth of targeted path segments. In *IEEE INFOCOM*, 2003.

[18] G. Iannaccone, *et al.* Analysis of link failures in an IP backbone. In *ACM SIGCOMM Internet Measurement Workshop*, 2002.

[19] The Internet Traffic Archive. http://ita.ee.lbl.gov/.

[20] M. Jain and C. Dovrolis. End-to-end available bandwidth: measurement methodology, dynamics, and relation with TCP throughput. In *ACM SIGCOMM*, 2002.

[21] K. Lai and M. Baker. Nettimer: A tool for measuring bottleneck link bandwidth. In *USITS*, 2001.

[22] B. Mah. Estimating bandwidth and other network properties. In *Internet Statistics and Metrics Analysis*, 2000.

[23] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson. User-level Internet path diagnosis. In *SOSP*, 2003.

[24] Z. M. Mao, J. Rexford, J. Wang, and R. Katz. Towards an accurate AS-level traceroute tool. In *ACM SIGCOMM*, 2003.

[25] S. McCreary and k. claffy. IP v4 address space utilization. http://www.caida.org/outreach/resources/learn/ipv4space/, 1998.

[26] D. Meyer. Routeviews. http://www.routeviews.org.

[27] UCB/LBNL/VINT network simulator - ns, 2000.

[28] J. Padhye and S. Floyd. Identifying the TCP behavior of Web servers. In *ACM SIGCOMM*, 2001.

[29] V. N. Padmanabhan, L. Qiu, and H. J. Wang. Passive network tomography using bayesian inference. In *ACM SIGCOMM Internet Measurement Workshop*, 2002.

[30] V. N. Padmanabhan and L. Subramanian. An investigation of geographic mapping techniques for Internet hosts. In *ACM SIGCOMM*, 2001.

[31] A. Pásztor and D. Veitch. Active probing using packet quartets. In *ACM SIGCOMM Internet Measurement Workshop*, 2002.

[32] V. Paxson and S. Floyd. Wide-area traffic: The failure of poisson modeling. *IEEE/ACM Transactions on Networking*, 3(3), 1995.

[33] L. Peterson, T. Anderson, D. Culler, and T. Roscoe. A blueprint for introducing disruptive technology into the Internet. In *HotNets-I*, 2002.

[34] R. S. Prasad, C. Dovrolis, and B. A. Mah. The effect of layer-2 switches on pathchar-like tools. In *ACM SIGCOMM Internet Measurement Workshop*, 2002.

[35] V. Ribeiro, *et al.* Multifractal cross-traffic estimation. In *ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management*, 2000.

[36] N. Spring, R. Mahajan, and T. Anderson. Quantifying the causes of path inflation. In *ACM SIGCOMM*, 2003.

[37] N. Spring, R. Mahajan, and D. Wetherall. Measuring ISP topologies with Rocketfuel. In *ACM SIGCOMM*, 2002.

[38] N. Spring, D. Wetherall, and T. Anderson. Scriptroute: A public Internet measurement facility. In *USITS*, 2003.

[39] L. Subramanian, V. N. Padmanabhan, and R. H. Katz. Geographic properties of Internet routing. In *USENIX Annual Technical Conference*, 2002.

[40] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg. Fast accurate computation of large-scale IP traffic matrices from link loads. In *ACM SIGMETRICS*, 2003.