# NETS-NR: A Shared Facility for Internet Reverse Engineering

We believe that the recent rapid progress in network measurement and analysis techniques and the increasingly widespread deployment of PlanetLab nodes provide a unique opportunity to dramatically improve our fundamental understanding of the Internet. To date, only IP path connectivity and BGP routing updates are measured at Internet scale on an ongoing basis; many techniques have been developed to measure and infer a broad set of other properties, but these tools have not yet been deployed at scale or run persistently. We propose to provide a key missing link – to lower the barrier of entry for researchers to prototype, evaluate, and deploy their tools continuously and at scale.

Specifically, we propose to design and build an open, shared facility for the continuous measurement, analysis, and archival of the Internet's underlying structure and operational controls. Our intent is to be as complete as possible, measuring not just the Internet's topology, but a rich set of link attributes (delay, delay variation, capacity, loss, utilization, MTTF), router attributes (geographic location, structural role, alias identity), and operational routing policy (intradomain link weights and interdomain peering and route selection policies). We believe there is tremendous synergy to be gained from being complete; the combination of metrics is greater than the sum of the parts, enabling, for example, new inferences to be drawn from the correlations between metrics.

A key part of our approach is an open architecture - where any network researcher or student can gain access to a live or archived feed of information about the Internet's structure and policies, and more importantly, can contribute new measurement and analysis tools to the community effort. Neither we, nor any other group, have a monopoly on all good ideas in Internet measurement or analysis, and thus the main part of our effort is to construct a framework for other researchers to plug their tools together and to run them continuously against the real Internet, in a safe and secure fashion.

**Intellectual Merit:**

The scientific case for better, ongoing measurements of the Internet is compelling - the Internet is a massively complex system, whose function is central to our society, and yet whose structure and operation is poorly understood due to a lack of data. Network modelers will have much richer sources of data to help put their research on a more solid foundation. Further, a shared facility will dramatically improve measurement validation, by doing continual runs, repeat analysis, synthesizing piecemeal data sources to allow for cross-checks, and enabling third party checks. Nevertheless, our approach carries risk. Perhaps foremost is the security challenge: we need to build a *distributed sandbox* for experimental measurement tools so that they do not cause unintentional denial of service or IDS alarms. Another significant challenge is cost-effectively scaling a diverse set of individual link and node metrics to be run at Internet scale.

**Broader Impact:**

We believe that our proposed effort will have significant broader impact, in terms of improving the quality and effectiveness of network research, and also in improving the security, efficiency, and flexibility of the Internet itself. Today, network researchers are hamstrung by a lack of quantitative data about the Internet's structure and operation, much as biologists prior to the Human Genome Project were hampered by a lack of data. The easy availability of a rich set of data, along with the ability to easily augment existing data sets, will be of particular help to students and faculty at underserved institutions, as it lowers the barrier to conducting relevant network research. Tool builders benefit by being able to use other people's work; network researchers benefit by using better models and by being able to demonstrate that their ideas would be effective on the wide diversity of behavior found in the real Internet; and society ultimately benefits from more reliable and efficient protocols.

## Contents

# NETS-NR: A Shared Facility for Internet Reverse Engineering

## 1 Introduction

Over the past few years, the network measurement research community has made stunning progress towards being able to efficiently estimate many of the properties of the Internet by measurement from unprivileged endpoints. Examples include individual link attributes, such as delay [39], delay variation [3], loss [29, 22], and capacity [20, 10, 31]; individual router attributes, such as geographic location [30], structural role [37], owner [24], IP aliases [37], and implementation features [28, 11]; as well as operational routing policy at both the intradomain [36] and interdomain [12] levels.

Complementing the progress on measurement tools, the increasingly widespread deployment of PlanetLab [33] provides a large and growing number of vantage points for hosting network measurements. Although PlanetLab was initially used primarily for measurements between PlanetLab nodes, we argue that its primary value as a measurement platform resides in measuring outwards from these vantage points across the public Internet. PlanetLab is now hosted at over 150 sites spread across twenty countries and five continents. Further, PlanetLab's size is doubling approximately every nine months, with an increasing focus on industry and international participation, improving its visibility into all corners of the public Internet.

We believe the combination of these two trends provides us a unique opportunity to dramatically improve our fundamental understanding of the Internet. Skitter [8] and RouteViews [26] have each demonstrated the enormous value that comes from collecting and archiving measurements on a global scale. Yet these efforts are limited to single metrics, IP path connectivity and BGP updates. We argue in this proposal that we can, and should, measure and infer many more Internet properties, regularly and at scale.

To address this, we propose to design and build an open, shared facility for the continuous measurement, analysis, and archival of the Internet's underlying structure and operational controls. A key part of our approach is an open architecture - where any network researcher or student can gain access to a live or archived feed of information about the Internet's structure and policies, and more importantly, can contribute new measurement and analysis tools to the community effort. Neither we, nor any other group, have a monopoly on all good ideas in Internet measurement or analysis, and thus the main part of our effort is to construct a framework for other researchers to plug their tools together and to run them continuously against the real Internet, in a safe and secure fashion. An open and easily modified system is particularly important for students and faculty at underserved institutions, as it lowers the barrier to conducting relevant network research.

Our vision is to build the measurement system as a composition of small tools, each tightly focused on measuring or inferring a single property. This is in contrast to previous approaches that attempt to integrate completely separate traces stored in incompatible formats. Our goal is to make it easy for network researchers to add tools to the system incrementally, as new measurement and inference techniques are discovered and validated. Tool builders will only need to focus on the added value their technique adds to the system, rather than on having to reproduce the rest of the infrastructure needed to run their technique at scale. This architecture of network researchers contributing extensions to a narrow core has proven successful in the past: our approach is modeled after the way the network simulator *ns* [27] facilitates continual improvement and refinement of simulation models by the network research community.

In our proposal, tools communicate with each other via a multi-terabyte persistent object store, or blackboard. The blackboard uses a publish-subscribe model to simplify connecting new experiments with the live or archived output of previously validated tools. Tools can thus be prototyped at a small scale, validated against the Internet and/or the results of related tools, and then promoted to run continuously to measure the entire Internet. While individual measurements are spread across PlanetLab, our model provides for central coordination, analysis, persistent storage, and most importantly, security.

The benefits of our approach, if successful, can be substantial. Today, network researchers are hamstrung by a lack of quantitative data about the Internet's structure and operation, much as biologists prior to the Human Genome Project were hampered by a lack of data. While any particular metric (or human gene) can be measured with significant effort, the combination of metrics, measured continuously, provides much more than the sum of the parts. Simultaneous measurement allows new inferences based on the relationships between properties and new studies of the correlations between properties. Network modelers will have much richer sources of data to help put their research on a more solid foundation. Network measurement researchers will be able to leverage each other's work, standing on each other's shoulders instead of each other's toes. It will also support a new methodology for validating proposed measurement techniques, addressing the fact that researchers generally do not have access to router-level network metrics for validation. Multiple methods can be run simultaneously against the same parts of the network; consistency can be checked by continual, repeat analysis; and third parties will have source code access for independent verification. Finally, we believe ongoing, wide scale measurements will enable new system and protocol designs based on the easy availability of detailed information about the underlying network, ultimately improving the reliability and efficiency of the Internet itself.

Despite the potential benefits, our approach of measuring the Internet as an open, community effort poses several significant research challenges. Our focus in this proposal is on the hard work of addressing the systems-level research challenges, to allow the network research community to focus on developing better measurement tools and using the resulting data sets to improve network research. Foremost among the systems-level challenges is security: we need to sandbox experimental measurement tools so that they do not cause unintentional denial of service or IDS alarms. Another challenge is cost-efficient scaling, to allow a broad spectrum of properties to be measured across the entire Internet on a regular basis, without consuming too much bandwidth. Yet another challenge is designing a flexible but simple framework for interconnecting measurement tools. The PI's are particularly suited to tackling these issues, due to their extensive experience in building measurement tools [37, 36, 22], as well as tools for measurement tool builders [33, 38].

The remainder of this proposal is organized as follows. We provide context for our proposal by discussing related work by the PI's and other research groups. In Section 3, we consider the feasibility of measuring a broad spectrum of network properties on an ongoing basis; we show that this effort, provided it is carefully managed, will not overwhelm PlanetLab's resources and can be archived efficiently. We outline our approach in Section 4. In Section 5, we discuss the research challenges to making this effort a success, and in Section 6 we outline our plan for constructing the system. Section 7 summarizes the proposal.

## 2 Results from Prior NSF Support and Other Prior Work

We first provide context for our proposed approach by briefly describing the PI's prior NSF support, the PI's previous efforts in building working measurement tools, and other related work.

### 2.1 Prior NSF Support

Anderson is well-known for large systems efforts, most recently as co-founder of the PlanetLab effort with Culler and Peterson. PlanetLab is his most closely related NSF-funded effort (PlanetLab: An Overlay Testbed for Disruptive Network Services, Grant #ANI-0335286, $625K over three years). PlanetLab aims to serve as a wide-area incubator for new distributed systems and Internet services. This collaborative effort is ongoing, but in the past year has reached a deployment of approximately 360 machines at 150 different sites spread throughout the Internet, including internationally. It has hosted 450 research projects from a wide set of institutions. PlanetLab is a key building block for this proposal, by providing a large number of diverse measurement vantage points.

Wetherall's most closely related NSF funding is a CAREER grant that explores the robustness of routing (CAREER: Dependable Network Communication, Grant #ANI-0133495, $355K over five years). This work is investigating consistency-checking and verification approaches to improve the handling of faults that are not fail-stop; while existing routing protocols can handle link cuts and router crashes well, they are much less tolerant of configuration mistakes, implementation bugs, and malicious insiders. This grant started two years ago and is ongoing. One major result was the first empirical study of globally-visible Internet backbone routing (BGP) misconfigurations [23]. In this study, we correlated BGP changes with end to end connectivity tests, to show that misconfigurations, while frequent, rarely affected connectivity.

### 2.2 Scriptroute and Rocketfuel

The principal investigators have extensive experience in building working Internet measurement systems, and in building systems to help others make Internet measurements. One source of inspiration for this proposal has come from our experience building and operating Scriptroute [38]. Scriptroute, the first permanent service hosted on PlanetLab, provides ordinary users the ability to conduct network measurements from remote vantage points. Scriptroute combines the usage model of web-based public traceroute servers (anyone can submit a request to run a network measurement) with the flexibility previously found only in dedicated measurement testbeds (any legitimate measurement can be made, subject to resource limits and security controls to prevent misuse).

The Scriptroute system allows any user (after validating their email account) to upload lightweight measurement scripts to servers, where they are executed in a secure execution sandbox. Careful resource limits and strict security isolation are used to prevent the system from being used for launching denial-of-service and other attacks; for example, spoofed source addresses and other malformed packets are disallowed. Scriptroute is currently deployed across Planetlab, and has been used for graduate networking courses and research projects at many institutions, including MIT, Minnesota, Georgia Tech, UC San Diego, UC Santa Barbara, Hong Kong Polytechnic, and Intel Research. As an example of its utility, Tim Griffin of Intel Research Cambridge recently said that he had spent several weeks trying to figure out how to run a network experiment on PlanetLab; once he discovered Scriptroute, he was able to complete his project in a day.

While Scriptroute has been very successful, it has significant limitations; its model makes it very easy to run an experiment on a single node, but it takes significant additional effort to run an experiment across the Internet as a whole, to run experiments on an ongoing, persistent basis, and to integrate the output of multiple experiments. This proposal can be seen as our effort to apply the lessons we have learned from our experience with Scriptroute and our large-scale Internet mapping experiments [37, 36].

We should note that Neil Spring, the first author on the Scriptroute and Rocketfuel papers, and on the HotNets paper that served as the foundation for this proposal, has agreed to be a PI of this effort if it is funded. Since he has not yet graduated (he is on the academic job market, with his Ph.D. expected in the summer of 2004), we could not list him as senior personnel; instead, we have simply included in the budget funding for a month of summer support for Spring plus a student for the three years of the proposal. We plan to subcontract that funding once his next employer is determined.

### 2.3 Related Work

Various previous efforts have attempted to address part of what is proposed here; their success suggests to us that it is time to take a more ambitious and comprehensive approach. Routeviews [26] and Skitter [8] have shown the value of continuously archived wide-area measurements; each of those systems has spawned numerous research efforts based on the availability of their data. Routeviews and Skitter are limited to measuring AS and router-level topology, however. We argue below that many more network properties can and should be measured continuously at Internet scale. Several recent efforts have attempted to archive and index diverse information sources, such as the Internet Traffic Archive [18], the proposed SIMR system [1], and the Internet Measurement Data Catalog [17]. The principal difference between these efforts and ours is the level of integration and support for measurement tools; these other efforts envision primarily off-line access to previously collected traces. While we expect to use their work in designing shared schemas, we argue that there is tremendous leverage to be gained from tightly integrating a range of network measurement tools together. By providing a shared framework for network measurement tools, as well as live feeds of useful measurement data, we hope to encourage researchers to build on the results of each other's work, making the whole more than the sum of the parts. As a data point, both Skitter and Routeviews became much more widely used, and useful, when they started providing live data feeds to the research community.

## 3 Measurement Opportunities

In this section, we argue that a broad spectrum of interesting network properties can and should be measured continuously across the Internet today. The goal of our proposal is to enable these and other similar properties to be measured, stored, and the results used by network researchers; we describe the research challenges to achieving that goal in later sections. The metrics we discuss in this section are meant as examples, not as an exhaustive set of metrics we could support. The intent of this section is simply to outline the scope of the opportunity. We do plan to implement several tools as a proof of concept, but implementing a tool for every possible metric would be well beyond the resources we have requested in this proposal. Rather our goal is to provide the right core services to enable the community to continually develop and refine tools and run them against the Internet at scale.

Table 1 provides a roadmap for our discussion in this section. We first discuss network measurement techniques that been published and shown to work; note that these have generally been tested only on a few paths, in simulation, or on subsets of the Internet topology. Our proposal would enable these techniques to be scaled to the size of the Internet. We only list one or two techniques for each property, to illustrate feasibility. Often many different approaches have been proposed for each metric; our approach would allow these different techniques to be validated against each other on the same, live data.

We next suggest some additional network properties whose measurement will be enabled by this proposal; these properties have typically not been measured, even at a small scale, in part because they depend on the availability of measurements of several other properties. Our intent in this part of the discussion is to illustrate the leverage that we can get out of combining multiple data sources. Figure 1 provides a potential organization for data flow between various tools.

Finally, we summarize with a feasibility study - what if all of the metrics listed were collected for all Internet links, nodes, and paths, at a reasonable frequency? Since the resulting bandwidth cost would be significant, even for PlanetLab, we discuss one of the main contributions of our proposed effort: how to efficiently measure a diverse set of network properties at Internet scale.

| 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 [11] | TCP port scan | $1658\times\mathcal{N}$ |
| Node failure | §3.4.1 | Probe IP-ID every minute | 24 hours $\times$ 60 min. $\times\,\mathcal{N}$ |
| **Link Properties** | | | |
| Loss | tomography [29] | Observe TCP retransmissions | *tomographic:* 0 |
| | Tulip [22] | Observe IP ID sequences | ( 200 (small) + 100 (big) ) $\times\,\mathcal{L}$ |
| Reordering | Tulip [22] | IP ID ordering in responses | $200\times\mathcal{L}$ |
| Delay | geography [39] | Per hop geographic distance | Geography |
| Delay variation | cing [3] | Router ICMP timestamp requests | (estimated) 1000 $\times\,\mathcal{L}$ |
| Capacity | quartets [31] | Tailgating-based packet train | 40 (big packets) $\times\,\mathcal{L}$ |
| Link failure | §3.4.1 | Correlate route changes with policy to find infinite cost links | Routing, Topology |
| Idle capacity | §3.4.2 | Extend pathload [19], chirps [35] | 5 retries $\times$ 100 (big) $\times\,\mathcal{L}$ |
| | §3.4.2 | Model queuing | Delay variation, Capacity |
| Utilization | §3.4.2 | Capacity – Idle capacity | Capacity, Idle capacity |
| Layer 2 Connection | §3.4.3 | Point-to-point vs. multiple access by address allocation | Topology |
| Layer 2 Switches | §3.4.3 | Difference [34] between capacity measurements | Capacity |
| Duplication | §3.4.4 | Observe repeated IP identifiers | *tomographic:* 0 |
| Corruption | §3.4.4 | Observe bad TCP checksums | *tomographic:* 0 |
| **Topology (Graph) Properties** | | | |
| Topology (AS) | RouteViews [26] | Archive BGP tables | *database:* 0 |
| Topology (IP) | Skitter [8] | Traceroute to all destinations | $2\times\mathcal{S}\times\mathcal{A}$ |
| Topology (Router) | Mercator [13] | IP aliases and IP topology | Topology (IP), IP aliases |
| Routing (IP) | Rocketfuel [36] | Constraint solver infers metrics | Topology (Router) |
| Routing (AS) | Gao *et al.* [12] | AS relationships in BGP tables | *database:* 0 |
| Client location | §3.4.5 | Prefix density [25] with BGP dynamics [4] or topology | *tomographic:* 0 |
| Workload | tomogravity [40] | Traffic matrix estimation using gravity and tomography models | Client location, Utilization, Capacity, Routing |
| Overall | $3224\mathcal{N} + 1400\mathcal{L} + 640(\text{big packets})\mathcal{L} + 0.016\mathcal{A}^2 + 2\mathcal{S}\mathcal{A} = 23$ billion packets, 2.9 TB | | |
| Per source | 76 million packets, 9.8 GB, 909 Kbits/s = $91 per month per source | | |

Table 1: Measurable network properties and unoptimized cost. $\mathcal{L}$ is the number of router to router or router to host links to measure; there are 2.4M IP-IP links in Skitter, and we estimate two thirds (1.6M) are router to router links. $\mathcal{A}$ represents the addresses, 1M 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, approximately 300 by the middle of 2005. *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 new metrics that can be inferred from other metrics, enabled by the work in this proposal. The cost estimate at the bottom is based on 1Mbps $\approx$ $100 per month [14]. We believe the cost per site can be dramatically reduced by applying the optimizations described in Section 3.5.

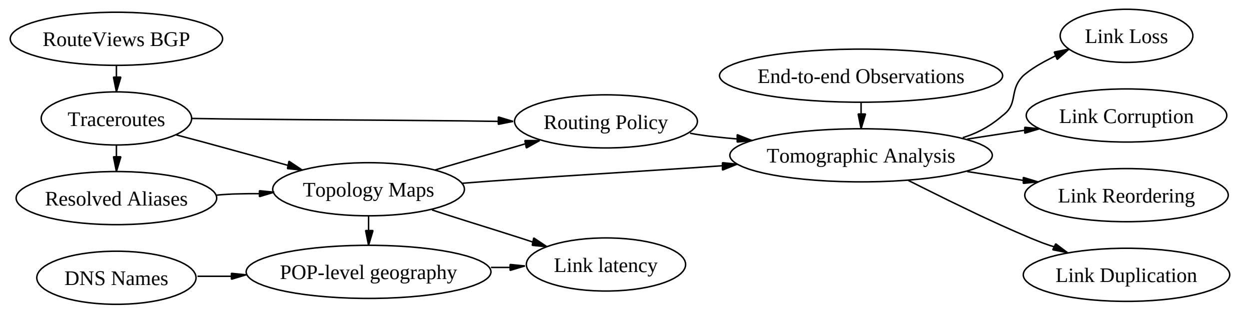


Figure 1: Possible data flow through the system; each new tool builds on the results of prior analyses.

### 3.1 Measurable 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 [13] 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).

### 3.2 Measurable 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 [22] 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 [22] builds on the techniques of Sting [5] 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 [3] tool uses ICMP timestamps to estimate the delay variation of path segments.

**Capacity** Existing methods cannot measure link capacity well for high speed links in the network core, but multiple vantage points may provide sufficient visibility to see most of the link capacities in the Internet. Link capacity can be measured by varying the packet size and tailgating packet pair. Variable packet size methods (pchar [21], clink [10]) are traditional but require thousands of packets. Packet quartets [31] and cartouche probes [15] refine nettimer [20] and measure capacity more cheaply. We see advances in capacity measurement as evidence that network measurement techniques are constantly improving.

### 3.3 Measurable Topology Properties

**Topology** Four levels of Internet connectivity have been studied. RouteViews [26] data includes the graph of connections between autonomous systems. Skitter [8] periodically collects the topology between IP addresses. Mercator [13] 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 [12], 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. Clearly the traffic matrix of a network is not immediately measurable from unprivileged endpoints, but a reasonable approximation may be within reach. Zhang *et al.* [40] present a technique termed tomogravity that combines topology, routing policy, client location, link capacities, and link utilization together to estimate the amount of traffic leaving one node in the network for another. At its core is a gravity model of network traffic, basically that the traffic from one site to another is proportional to the number of nodes or amount of traffic sourced at each node. To obtain this estimate, we must measure link utilization and the location of clients, described in the next section.

### 3.4 Inferring New Properties

Next, we discuss several network properties that have not received as much attention in the research literature: failure, utilization, layer 2 topology, duplication, corruption, and client location. We argue that measurements of these properties would also be enabled by our proposal; in many cases, the property can be inferred provided the metrics described above were available. A shared measurement framework is key to enabling these kinds of broader studies to be conducted. Again, we do not intend to implement each of these tools; our intent in this part of the discussion is to illustrate the leverage that the research community can get out of combining multiple data sources.

#### 3.4.1 Failure and Evolution

Failure is perhaps the most opaque property of the network. Failures generally last only a few minutes [16], 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 [16], 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.* [7]. 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 [6], 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.

### 3.4.2 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 [21]) and the bottleneck available capacity of a path (e.g. pathload [19]), 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 [3] and tulip [22] 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.* [2] 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.

### 3.4.3 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.* [9]. Similarly, differences between geographic latency and measured link latency can imply detour routing at the link layer.

#### 3.4.4 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. From measured topology and routing policy, the path that inbound duplicated or corrupted traffic traversed can be inferred. A researcher examining this property could deploy a modest number of passive packet monitors at selected sites, combine that data with our live measurements to infer inbound paths, to narrow the pathology to an individual link. This illustrates some of the synergies enabled by our work.

#### 3.4.5 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.* [4] 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 Research Challenge: Efficient Measurement at Internet Scale

We have argued above that we can approximate each of the properties in Table 1. The question is whether we can scale those measurements to the entire Internet. The limiting factor is the total amount of traffic generated by each vantage point because it controls how quickly or how often the measurements may be run. Clearly, if we measure all properties from all vantage points to all Internet destinations, the overhead would be enormous [37]. Thus, as a starting point, we assume that the distributed measurements are coordinated, so that each link and node is measured only once for each property. Table 1 shows the result; with simple extensions to present techniques, the amount of traffic needed is roughly 76 million packets per day, or less than 1 Mbps, per vantage point. Although this is within an order of magnitude of the overhead Skitter places on each of its nodes, it is large for some of the less well connected sites on PlanetLab.

One of the fundamental research contributions of our work will be to design, implement and evaluate ways to reduce and manage Internet-scale measurement overhead. We believe that we can reduce the overhead of our system to near that of Skitter, despite the fact that we are measuring twenty-five properties to Skitter's single metric of IP path topology. Most properties in Table 1 can be measured inexpensively, or can be inferred from the results of other tools. Those to be concerned about are topology, IP aliases, and link properties; fortunately, many opportunities exist for reducing the overhead of these metrics.

For example, topology can be measured using many fewer packets than the 6.5 million per site used by Skitter. 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 per site(the number of traceroute destinations plus the number of addresses a source discovers).

Similarly, $3224\mathcal{N}$, or 14% 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 handle ECN properly or for nmap, that the Web server is still available. Node properties can also be measured from anywhere; only well-connected sites with cheap connectivity could be used for these measurements, avoiding overloading nodes for which the measurements would be burdensome.

Alias resolution is the most expensive analysis in Table 1, accounting for 69% of the packets. The problem is that any two IP addresses in a trace can potentially be aliases of each other. 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 there is the potential to eliminate more when using more vantage points. An even more effective technique may be to use the geographic location of IP addresses inferred from their DNS names, as a way of quickly eliminating possible aliases (e.g., those in different cities). This illustrates the power of being able to combine multiple metrics together in the same system. Finally, we note that aliases are a node property, and so the optimizations described in the previous paragraph apply here as well - conducting measurements from well-connected sites and performing them only when there is evidence that it has changed. For example, topology and aliases change together, so changes in topology may be able to be used to trigger alias resolution.

Measuring link properties uses 14% of the packets but 57% of the bytes. A key contribution is that we will provide a framework to make it easy for tool designers to measure individual links or trees leaving a source without repetition; otherwise it would not be possible to scale link measurements to the entire Internet. Other optimizations include determining the best location to conduct each link measurement; many links are visible from many vantage points, and our experience with Rocketfuel suggested that the bulk of the measurements can be conducted from the most well-connected sites. Having a large number of vantage points helps the system measure as much of the Internet as possible, but that does not imply that all measurements must be spread equally across all nodes. Our Rocketfuel data, for example, showed that once we have a large number of vantage points, each incremental measurement node may only add visibility to a small number of new links or nodes. Another optimization is to leverage past measurement data. For example, 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 believe that we can keep the measurement overhead per vantage point to a reasonable level.

## 5 Research Challenge: A Shared Framework for Wide Area Measurement Tools

In the previous sections, we argued that there is a huge opportunity for running a large, diverse set of measurements on the Internet. The scope of the work required to make that happen is equally huge, at least an order of magnitude or more larger than what we've budgeted in this proposal. Implementing a known technique by itself is not hard; we implemented cartouche probes in Scriptroute in a few days. Rather, the real effort is in evaluating proposed techniques against each other to determine the best approach, and in making the resulting tool robust to the diversity of behavior encountered on the live Internet. Even if we did have the resources to implement and validate all of these techniques ourselves, we believe that would be counterproductive - instead, it is essential that we enable the community to continually develop and refine new tools and to run them at scale.

Our approach is to build a shared framework that makes it as easy as possible for network measurement researchers to implement a new measurement technique, run it experimentally against part of the Internet, validate and compare its results against other tools, and once its value is proven, scale it up and run it continuously. Of course, we plan to build initial implementations of a number of tools ourselves, to validate our support software, to provide examples to show how the system works, and to seed the system with interesting data that other tools can build upon. The use of example code was particularly successful in helping new users learn Scriptroute.

What are the barriers to tool builders today, especially in scaling their measurements across the Internet and/or in running their tools continuously? To be successful, it must be easier for tool builders to use our framework than to reimplement this support for themselves. Our list of barriers comes from experience – the difficulties we encountered in building scalable measurement experiments and in attempting to explain to graduate students how to safely conduct such experiments. Based on our experience, Scriptroute and PlanetLab by themselves are not the complete answer. These barriers include:

**Lack of data sets to use as building blocks** A researcher today interested in studying some new property often has to start from scratch, re-engineering most or all of the primitives needed to infer some high level property. For example, a recently conducted study of link failures required the student researcher to gather a fresh trace of the topology, intradomain policy inference, and hop-by-hop loss data. We need to make it easy for tools to produce data in a form that other researchers can use.

**Complex machine and data flow management** Conducting a large-scale measurement experiment is a significant programming challenge, requiring the coordination of hundreds of machines, any set of which will be down or disconnected at any given moment. The specific measurement assigned to each machine is topology dependent, and the output must be processed and fed to other tools. Every piece of the system must be designed to be tolerant of missing or noisy data. Yet at the core, the individual measurement is often very simple, for example, to collect a round trip measurement to every router.

**Security issues** While Scriptroute prevents many kinds of obvious misuse, it is still very easy to trigger IDS alarms, even when sending a small number of innocuous measurement packets. The scope for inadvertent error is even larger when running experiments across the entire Internet. A consequence is that experimenters must carefully plan each test, and even then, alarms are still possible. During one experiment we ran last year, we sent a few dozen packets that were similar but not exactly like traceroute probes to nodes on 10000 separate subnets. One recipient complained, with the result that a number of sites disconnected their PlanetLab hosts. While we were able to reconnect those sites eventually, we need to provide system support to minimize the frequency of such events.

A major part of our effort in this proposal is to design, implement and validate a set of support software for conducting wide area network measurements, to address these barriers. Our approach is straightforward, to abstract out the common elements shared between tools so that they can be re-used. Our goal is to enable researchers to construct new tools as the composition of number of smaller tools, each tightly focused on measuring or inferring a single property. We discuss various aspects of our proposed solution in the rest of this section.

### 5.1 Generalizing Link Measurement Techniques

The link properties discussed in Section 3.2 share a common pattern, an important special case for our toolkit. If the routers at each end of the link support direct measurement, the link property can be estimated as the difference in the values at the two endpoints. Otherwise, round trip measurements must be used, and topology and routing policy information is needed to drive multiple overlapping measurements from different vantage points, all just to infer the underlying value for the link. Depending on the topology and the capabilities of the routers, it may not be possible to discriminate the property to the individual link in all cases.

We plan to build a meta-tool that can take a simple Scriptroute program to make a single measurement of some property of interest (e.g., delay), and use stored topology and routing policy information to generate a set of commands for each vantage point, with the effect of measuring the property for every link visible to a PlanetLab node. The meta-tool will then store the results (see below), and optionally use historical information to guide more efficient measurement of the property in the future.

### 5.2 Publish/Subscribe Storage Model

A primary goal of our work is to facilitate the easy sharing of measurement results between related tools. Some tools may access only archived data, while others may need a live feed, e.g., to use measured data to trigger other measurements of interest. For example, an observed routing change may trigger alias resolution of a newly discovered node. We plan to support both on-line and off-line data feeds through a common publish/subscribe interface to a persistent object store. Since our focus is not on the mechanics of large scale storage systems, we plan to use existing tools for this purpose, such as the CMU self-* storage system. Note that we do not expect to define a standard schema for all tool builders to use, except by example (e.g., for topology and other simple metrics). Rather, tool builders will need to document their data formats if they want others to use them.

We plan to annotate the storage system to keep track of data pedigree. Our vision is that data sets produced by a tool might be part of a long chain leading to the inference of a new property, stored by that tool as a new data set. It is inevitable that eventually some technique used to produce some piece of data will be proven invalid, at least in some cases, and thus it will be important to track the various data sets derived from the (potentially) faulty data, so that they can be recomputed or at least flagged. This is a standard technique in the scientific community; one of the PI's (Anderson) has previously built such a system for tracking pedigrees in climate data.

A further step is to provide standard mechanisms for compressing and decompressing data sets in long term storage. An interesting avenue for our research is to explore the synergy between measurement planning and storage compression. In both cases, we want to design the measurement (or storage) to provide the greatest incremental information per bit. If measured data is very predictable, it can be easily compressed, but perhaps it should also be measured less frequently. Downstream tools may be able to dynamically decide whether to use old data or cause the measurement to be reissued, depending on the predictability of the result. Thus, building a toolkit of standard prediction models is likely to be useful in multiple contexts.

### 5.3 Secure Sandbox

An essential element of any shared, open Internet measurement facility is limiting the scope of tools to disrupt the real operation of the Internet. As in Scriptroute, we expect tools are written in a standard, sandboxed programming language, to allow easy manipulation by the core security system. Scriptroute not only prevents scripts from sending malformed packets (such as with spoofed source addresses), it also limits resource usage, for example, so that no node can send more than three unacknowledged TCP SYN packets to a given destination.

We intend to go beyond Scriptroute in two significant ways, motivated by the larger potential for problems as tools are scaled up. The first is to use stored topology data sets to construct a distributed resource sandbox. No set of nodes should be able to send more than a limited amount of traffic to any node or through any intermediate link, unless the target provides explicit permission (e.g., it's a PlanetLab node) or the measurement code has been carefully reviewed by a human.

Further, we plan to use our measured topologies to drive a simulation of what could happen for a new tool when it was run at scale. In essence, we should try to discover problems before we run a new tool on the live Internet. Such a simulation might determine, for example, that the proposed tool would unexpectedly overwhelm some corner of the network or trigger a standard IDS alarm. We plan to call this simulation, scriptlint. We currently publish a FAQ for Scriptroute tool builders as to how to avoid triggering IDS alarms; scriptlint could embody that set of heuristics. As with the distributed resource sandbox, scriptlint is enabled by having tools written in standard scripting language; the simulation would provide the measurement tool being tested typical responses that it would get back from the real Internet. The model is similar to the XBone or Emulab, but using our measured data to drive the simulation.

### 5.4 Tool Promotion

Finally we consider resource usage. Clearly, we do not want to allow experimental tools to write arbitrary amounts of data into the storage system, or to generate arbitrary amounts of measurement traffic, even if they do not overwhelm any individual part of the network. Section 4 argued that PlanetLab has the network resources to measure many properties across the entire Internet, but it does not have the network resources to run every possible tool at maximum scale. Nevertheless, we want to encourage anyone wanting to test out a new idea to use our system, particularly students who might be more likely to write runaway tools. Our approach is to make it easy to test new tools with restricted rights, e.g., with strict bandwidth and storage limits. Only after the code is tested and validated against other tools running on a subset of the Internet, would we raise resource limits to let it run across the entire network. And only after a tool runs well at scale, and demonstrates its value compared to competing tools, would we raise its resource limits again to run it continuously. For many properties, the tools will be cheap enough that we will be able to run multiple tools running simultaneously on the entire Internet; there are only a few properties, such as alias resolution, link loss rate and link utilization, where we may need to be especially careful.

## 6 Constructive Plan

In this section, we briefly discuss how we plan to stage our effort over the three years of the project.

In year 1, we intend to focus on the design and construction of a prototype of the framework; our approach will be to assemble as many off-the-shelf pieces as possible, to begin gaining experience interfacing to real network measurement tools. In parallel, we plan to develop tools for measuring several of the properties described in Section 3, for validating that our framework design will support a broad range of measurement tools. We already have simple versions of tools for measuring or inferring topology, routing policy, IP aliases, geography, router role, several link properties such as delay, loss, and capacity, as well as link and node failures. A substantial task will be to start running these on the scale of the Internet using our framework.

After demonstrating that our system works with our own tools, we plan to open the system to external tools towards the end of year 1 or the beginning of year 2. Our approach is guided by our experience with PlanetLab, where the key to success was as quickly as possible assembling a primitive usable system, opening it to external use, and then over time, improving the system based on external feedback. We expect to gain a huge amount of experience as the system is used, that will guide our effort for the remainder of the project.

Since we plan to support live users, we have budgeted for a full-time systems programmer to help ensure that the implementation is rugged, to complement the efforts of the PI's and the student researchers on the project. We also see an opportunity, once the framework is in place, to provide research opportunities for undergraduates to build initial implementations of some of the tools outlined in Section 3. This will help validate our goal of lowering the barrier of entry for tool builders to try their ideas at Internet scale.

Over the life of the project, we plan to assemble a small cluster of memory-laden compute servers and approximately 50 TB of disk storage. Most of the budgeted equipment cost is for disk storage. Although the survey in Table 1 suggests that our system may generate a petabyte per year of measurement traffic, we expect that we will need to store much less (although still a substantial amount). Measured and inferred data sets will be compressed and stored at UW, as well as mirrored by the CMU petabyte self-* storage project. The local cluster is intended to support compute-and-memory intensive analysis tools that infer new properties from stored or streamed data sets. We will of course also support downloading or streamed remote access to the data. Researchers will be provided access to the analysis cluster via PlanetLab accounts; PlanetLab provides the abstraction of sandboxed linux machines to its users, so that we can host any legitimate network researcher wishing to run custom analysis on the stored data set. We plan to phase our equipment purchases by buying the cluster and a modest number of terabytes of disk space in year 1, and larger amounts of disk space in years 2 and 3, to benefit from the fact that disk cost per terabyte improves at roughly 100% per year.

## 7 Summary

The network research community has an unprecedented opportunity. The twin developments of widely deployed PlanetLab nodes plus rapid advances in network measurement research together make it possible to systematically measure dozens of link, node, topology, and operational properties, on a continual basis across the entire public Internet. We propose to design and build an open shared facility to continuously measure as many properties of as much of the Internet as possible. We argue in this proposal that a large number of interesting properties can be efficiently measured today, provided the right system support. A key to our approach is our open system design: we leverage, rather than fight, the research community's ability to devise ever better methods of measuring and inferring network properties of interest. Our goal is to make it easy for researchers to plug new tools into our system. The challenge, and the focus of this research, is to develop a framework for integrating many small measurement tools together, providing for distributed management, persistent storage, communication between related tools, debugging, and most importantly, security. The principal investigators are particularly suited to working on this kind of systems-oriented networking research, having built both network measurement tools and tools for network tool builders. If successful, this proposal will enable a new class of network research, informed by a deep understanding of the underlying structure and operation of the Internet.

## References


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

[2] S. Alouf, P. Nain, and D. Towsley. Inferring network characteristics via moment-based estimators. In *Proceedings of IEEE Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, Apr. 2001.

[3] K. G. Anagnostakis, M. B. Greenwald, and R. S. Ryger. cing: Measuring network-internal delays using only existing infrastructure. In *Proceedings of IEEE Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2003.

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

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

[6] C. Bormann, C. Burmeister, M. Degermark, H. Fukushima, H. Hannu, L.-E. Jonsson, R. Hakenberg, T. Koren, K. Le, Z. Liu, A. Martensson, A. Miyazaki, K. Svanbro, T. Wiebke, T. Yoshimura, and H. Zheng. Robust header compression. RFC 3095, July 2001.

[7] B. Chandra, M. Dahlin, L. Gao, and A. Nayate. End-to-end WAN service availability. In *USENIX Symposium on Internet Technologies and Systems (USITS)*, Mar. 2001.

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

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

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

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

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

[13] R. Govindan and H. Tangmunarunkit. Heuristics for Internet map discovery. In *Proceedings of IEEE Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2000.

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

[15] K. Harfoush, A. Bestavros, and J. Byers. Measuring bottleneck bandwidth of targeted path segments. In *Proceedings of IEEE Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, Apr. 2003.

[16] G. Iannaccone, C. Chuah, R. Mortier, S. Bhattacharyya, and C. Diot. Analysis of link failures in an IP backbone. In *ACM SIGCOMM Internet Measurement Workshop*, Nov. 2002.

[17] Internet measurement data catalog. `http://www.caida.org/projects/trends/imdc/`.

[18] Internet traffic archive. `http://ita.ee.lbl.gov/`.

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

[20] K. Lai and M. Baker. Nettimer: A tool for measuring bottleneck link bandwidth. In *USENIX Symposium on Internet Technologies and Systems (USITS)*, Mar. 2001.

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

[22] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson. User-level Internet path diagnosis. In *ACM Symposium on Operating Systems Principles (SOSP)*, pages 106–119, Oct. 2003.

[23] R. Mahajan, D. Wetherall, and T. Anderson. Understanding BGP misconfiguration. In *ACM SIGCOMM*, Aug. 2002.

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

[25] S. McCreary and k. claffy. IP v4 address space utilization. `http://www.caida.org/outreach/resources/learn/ipv4space/`, Aug. 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*, Aug. 2001.

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

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

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

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

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

[35] V. Ribeiro, M. Coates, R. Riedi, S. Sarvotham, B. Hendricks, and R. Baraniuk. Multifractal cross-traffic estimation. In *Proceedings of ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management*, Sept. 2000.

[36] N. Spring, R. Mahajan, and T. Anderson. Quantifying the causes of path inflation. In *ACM SIGCOMM*, pages 113–124, Aug. 2003.

[37] N. Spring, R. Mahajan, and D. Wetherall. Measuring ISP topologies with Rocketfuel. In *ACM SIGCOMM*, pages 133–146, Aug. 2002.

[38] N. Spring, D. Wetherall, and T. Anderson. Scriptroute: A public Internet measurement facility. In *USENIX Symposium on Internet Technologies and Systems (USITS)*, pages 225–238, Mar. 2003.

[39] L. Subramanian, V. N. Padmanabhan, and R. H. Katz. Geographic properties of Internet routing. In *Proceedings of the USENIX Annual Technical Conference*, June 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*, June 2003.