Reverse Engineering the Internet

Neil Spring

This is an unpublished thesis proposal. For the HotNets paper of the same name, go to http://www.cs.washington.edu/homes/nspring/papers/reverse.pdf

Abstract

Understanding the structure and design of the Internet is is increasingly important as we seek to improve its reliability and robustness. At the same time, as the network grows in scale and diversity, a complete and accurate view is increasingly hard to come by. In this paper, we present a framework for classifying network measurement tools by how they can contribute to network mapping and whole-Internet analysis. We describe techniques to accommodate the scale and heterogeneity of the network. While most network measurement tools focus on performance and pathologies, we focus on developing tools to understand structure, design, and routing policy.

1 Introduction

Measuring and understanding the Internet has become increasingly important as we rely on it more and its vulnerabilities and limitations are exposed. Understanding the design of the Internet is important for researchers interested in routing protocols [10, 53], multicast studies [78], selfish routing schemes [84, 6], and denial of service traceback and response [59, 74, 85, 87, 80]. Understanding the structure is also important from a consumer and regulatory standpoint, where an independent view of the reliability, performance, and provisioning of the Internet can help shape public policy [28, 67].

While administrative tools can be used to measure an individual network for management [34, 33], it is difficult to discover the structure of the Internet in general, or even simply a handful of specific ISP networks. While some ISPs publish high-level topologies, and a few research networks even publish some details, what few maps exist may be inaccurate and are of limited value in research. Because no single organization "owns" the Internet, research on the global Internet infrastructure suffers [8], while it demands a new set of tools that can cope with its heterogeneity and ever-increasing scale [27, 36].

Heterogeneity permeates every aspect of the Internet, making it difficult to extrapolate where any particular problem occurs, who is responsible, and what should be done to fix it. There are thousands of distinct administrative domains ("autonomous systems") cooperating to run the network, each with its own choice of hardware, choice of protocols, set of service-level agreements (SLAs), approach to traffic engineering, etc. However, each has only a limited view of the network and can rarely evaluate the far-reaching effects of a change [62]. Heterogeneity also means that various exceptional, buggy behaviors can complicate otherwise elegant analyses [69, 4].

Existing tools provide a partial view of the global picture. They often measure end-to-end (path) properties but these path properties can be hard to compose in a map [76, 100, 83, 68]. Some measure hop-by-hop properties, but send too many packets to be used at scale [58, 45, 30, 75]. While recent projects have composed many, many end-to-end measurements of loss and latency with "tomographic" analyses to infer hop-wise properties [70, 13], it is unclear whether properties that have traditionally required more traffic, such as bandwidth and congestion, can also be inferred in a lightweight manner.

In this thesis proposal, I assert that these problems of heterogeneity and scale can be addressed to "reverse engineer" the core of the Internet accurately and efficiently. The goal of this reverse engineering is to provide sufficient detail in an annotated map of the network to evaluate potential changes: new links, new protocols, new queue management, etc. With accurate, measured data, one might even look for correlations between properties – such as whether traffic engineering primarily avoids congested links or whether routing policy arrangements are symmetric. This thesis is about demonstrating that such a map can be constructed without new infrastructure, not about the resulting map. We use "core" above to avoid detailed external mapping of edge networks, such as corporate or campus networks, which are less useful for network simulation, are less interesting in measurement as more traffic traverses the core, and are more likely to perceive measurement traffic as a possible attack. The primary challenges in this

mapping are in recovering attributes of the network that are currently only observed by network administrators – network topology, routing policy and traffic loads.

The rest of this paper is organized as follows. In Section 2, we present an overview of the Internet architecture and introduce terms we will use in the rest of the paper. Section 3 presents a framework for network measurement projects based on their potential contribution to network mapping. Section 4 describes recent projects in this framework, along with brief discussions of potential short-term research directions. Section 5 briefly describes our recent work that begins to address the problems of discovering Internet structure, policy, and pathologies. Section 6 describes a component that remains to complete the picture, lightweight hop-wise available bandwidth estimation. Finally, we conclude in Section 7.

2 Internet Primer

The Internet is a collection of networks run by different Internet Service Providers (ISPs). Each network is made up of *routers* connected by *links*. In this paper, we treat all links as IP-level links, and ignore the physical topology underlying the network-level hop. Networks that connect to each other are said to *peer*, those locations where they connect are *peering points* and the connections are *peering links*, which are otherwise ordinary links except that they connect routers owned by two different ISPs. Each ISP is made up of one or more *autonomous systems* (ASes) representing an administrative domain. Large ISPs, such as AT&T, have different autonomous systems, one for each continent. Each AS runs two *routing protocols*, one interior gateway protocol for its own internal network and one exterior gateway protocol for communicating with its neighbors and the rest of the Internet. Finally, each ISP has several points of presence (POPs) that represent physical locations, roughly one per city, and the connections between POPs constitute the *backbone* or *core* of the network.

ISPs typically design their network to be robust to failures using redundant links and redundant routers. This arrangement of routers and links is *network design*. ISPs also attempt to balance traffic across the capacity they do have, keeping utilization (and therefore loss and queueing) low. The balancing of traffic and the general optimization of routes to improve performance are called *traffic engineering*.

The Internet Protocol (IP) is designed to carry data across different data link types like Ethernet, 802.11 wireless, ATM, etc. As such, it makes minimal assumptions about the reliability and speed of those underlying links, and provides minimal service beyond end-to-end connectivity. This means that the IP service model allows packet loss, delay (reordering), corruption, and even duplication. The Transmission Control Protocol (TCP) corrects for these faults to provide a reliable, error-free, byte-stream oriented transport protocol for applications to use. Because it detects and corrects faults, as well as adapts to the capacity of the network, TCP's adaptive behavior can be observed to measure these properties.

There are two primary performance metrics of the network, latency and bandwidth. *Latency* is the base transmission time of the smallest unit of data. When measured alone, it is the time a very small (40- to 64-byte) packet takes to transit the network. When measured with bandwidth, it is the time for the first bit, in other words the y-intercept if transmission time is a function of packet size. Latency is comprised of *queuing delay*, which varies with load, and *transmission delay*, which is fixed. *Bandwidth* is the additional time taken per bit, or the slope of that function relating packet size to time, not a physical property of the transmission medium. *Capacity* is the maximum possible bit rate offered by a link, while *available bandwidth* is the unused, idle capacity of the link.[1]

In each packet sent through the Internet, a header field called the time-to-live (TTL) prevents the packet from cycling about the network in an endless loop should routing be temporarily inconsistent. Routers decrement the TTL field when forwarding a packet along a path; if the field reaches zero, an error message is returned to the source. This is a basic primitive ultimately used by network measurement tools to address individual hops along a path. *Traceroute* is one such tool to discover the routers along a path; we occasionally use traceroute as a verb to represent executing the tool to discover the path to some other host.

3 Contextual Framework

In this paper, we are interested in tools that allow an outsider to infer the behavior of the network with sufficient precision to annotate a map. Researchers have developed many tools to discover network properties to varying degrees of precision. Figure 1 shows a conceptual hierarchy of the precision of existing tools. The width of the lower segments implies a density of tools that infer many properties of nodes and end-to-end paths, fewer that observe hops and trees, virtually none that measure maps, and none that synthesize measurements to annotate the map. We present the context for

[1] An alternate definition of available bandwidth, sometimes used, is the bandwidth a new flow could expect to achieve over a link, which may be greater than the idle bandwidth as existing traffic is displaced.

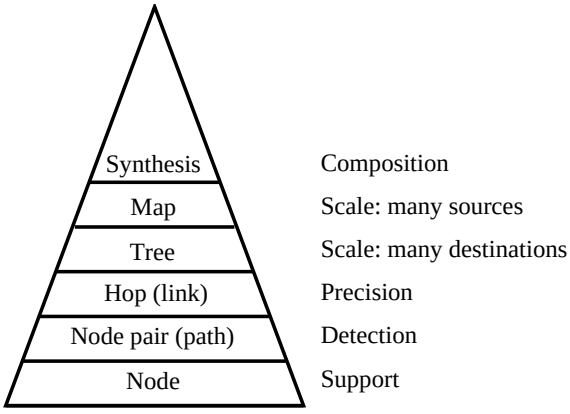

Figure 1: Hierarchy of network measurement precision and scale. The precision and scope of understanding is at left in the pyramid, while the challenges to overcome are at right.

measurement in this order, with increasing detail on the problem solving approaches as we near the top of the pyramid. Promotion from one level to the next often requires solving difficult problems of support, detection, precision, scale, and composition.

Figure 1 alludes to Maslow's hierarchy of needs from the field of psychology [64]. In 1943, Abraham Maslow studied sane adults (as opposed to asylum patients, children, or animals), and based on these studies formulated a hierarchy of human needs to explain human motivation. This hierarchy is comprised (from lowest to highest) of the physiological (food), security (avoid danger), love (acceptance), esteem (approval), and self-actualization (realizing potential). The hierarchy was later revised to include somewhat more detailed levels, but the insight is the same: humans satisfy needs in essentially this order, seeking food before safety before approval.

In understanding network design and performance, we face a similar hierarchy: understanding the global picture is an evolution from tools that first detect properties along some path, then focus on a link at a time, and ultimately permit properties to be measured at scale, forming a tree or combination of trees to form a map. To understand a new property, we need first to understand node (end-host or intermediate router) behavior, then to use that knowledge in building a tool to estimate a path property. After developing a methodology for measuring a path, extensions and engineering can measure hop-wise properties, perhaps recognizing the site of some anomaly or measuring the bandwidth of individual links. Such tools often require further engineering to run at scale – perhaps trading accuracy for efficiency. We now sketch each of these levels with selected examples.

Node measurement tools, such as tbit [69], nmap [38] and fsd [39], discover the properties of individual nodes (end hosts or routers) in the network. Such tools are fundamental, and measuring node properties is implicit in the design of more advanced tools. For example, traceroute implicitly measures the responsiveness of a node when it returns an ICMP error response. Also forming a foundation for more advanced tools, node location services like IDMaps [37] or GeoTrack [71] expose geographic properties of routers and end-nodes, which is useful for rendering on a map, estimating path latencies, or understanding geographic properties of links [56, 92].

Node-pair (or end-to-end path) measurement tools are similarly diverse. TCP-based tools, such as the active (probing-based) Sting [11, 83] and Treno [65], as well as the passive (trace-analysis based) tcpanaly[76] and TRAT [100], generally look at end-to-end paths, often characterizing several properties at a time. Passive measurements of conversations between pairs of machines are often used to estimate a property without precisely locating it in the network, for example, the measurements of reordering by Bellardo and Savage [12] or loss by Paxson [76]. Unfortunately, the results of these measurements are difficult to use in determining where problems occur. One approach compares many paths that do and do not traverse a peering point to determine whether peering points are bottlenecks [2], but this precision is insufficient for map annotation.

Tools that can pinpoint an attribute to a particular hop (IP level link) often build upon node-pair tools using limited time-to-live fields in the IP header to solicit ICMP time-exceeded errors. Such tools include traceroute [46] and its variants tcptraceroute [96], fft [66], etc., to measure connectivity and round-trip latency, and pathchar [45] and its variants pchar [58], clink [30], etc., that measure bandwidth. Alternatives to soliciting time-exceeded messages include finding paths directly to intermediate routers that are prefixes of an end-to-end path as used by cing [4], or using limited TTL in a one-way measurement in a "tailgating" technique [43, 54].

Tree-analysis tools can either compose a set of hop-based measurements to different destinations or compose a set of path measurements using inference or clustering techniques. Padmanabhan, *et al.*, compose inference using path measurements of TCP transactions with traceroute in their study of loss in the network [70]. They find that loss generally occurs on segments close to clients, though their vantage point of a well-provisioned server may bias the result. Other studies discover the properties of branches in the tree without looking at the underlying IP-level topology, such as

MINC [1], and Coates, *et al.* [26]. Such projects use shared behavior along segments of an (often multicast) tree to infer a particular property. The simple example that describes the intuition of multicast inference is loss – when loss is rare, one can assume that multiple stations that miss a packet share a (lossy) segment of the multicast tree.

Map-analysis tools compose tree-analyses from several vantage points to understand the network as a whole, rather than simply the network as observed from a single point. Such mapping approaches include early work by Pansiot and Grad [72], followed by Skitter [25], Mercator [40] and Lumeta [18, 24]. The usefulness of these connectivity maps has been limited to inferring relatively abstract graph-theoretic properties of degree distribution [32, 15], small-world-ness [16], resilience, expansion, distortion [93], etc. However, these metrics alone have changed the focus of topology generation from structural [99] to degree-distribution-based [48]. That is, rather than model the Internet as the hierarchical collection of ISPs and points-of-presence (POPs, roughly one per city per ISP), recent efforts have attempted to match the properties of the measured graph – a demonstration that real, measured data can shape research in surprising ways.

The synthesis of mapped attributes to annotate a map is essentially unexplored. While recent work has combined properties like geographic location with network paths [92] to understand how network paths traverse the globe, and other work renders Internet maps color-coded by ISP or address space [18, 25], work that brings together network attributes to annotate a map using external measurements is in its infancy. Demonstrating the ability to fill this vacuum using external measurement is the overall direction of this work.

Summarizing the hierarchy, to traverse from one level to the next requires solving important and often difficult engineering problems. Promoting from a path-wise tool, such as bprobe, to a hop-by-hop tool, such as pathchar, requires much more than simply setting a TTL field: it requires many packets and manipulation of aggregate statistics to estimate per-link bandwidths [30]. Promoting such an invasive tool further to one that can be used at scale is a particular (and open) challenge.

4 Previous Approaches and Opportunities

We summarize existing research projects and their tools in Table 1. While many properties have been studied to measure their prevalence along paths, few have been studied to discover where in the Internet they occur. We draw a distinction between tools that measure properties useful for the engineering and construction of a network, and those tools that measure the deliverable performance attributes. Our focus will be on the development of new tools to measure the former category, often using tools and techniques from the latter.

In Table 1, parentheses represent tools that have yet to be built. Hop-tools to precisely locate pathologies like reordering and duplication would be both useful and challenging. Equally challenging is to extend those and existing hop-measurement tools to run at scale and to help complete the annotated map of the Internet, filling the Tree and Map columns of the table.

4.1 Node Analysis Techniques

As we observe in Table 1, node measurement tools are incomplete in our ability to remotely measure failure and (processor) load. Our understanding of node failure is limited to internal studies like one from Sprint [44], that observe internal routing protocol traffic to detect link and router failure. They found that failures were generally short and occurred more often during scheduled maintenance intervals. To measure a failure rate externally, one would likely start by measuring a map of the network and repeatedly measuring paths to observe if they detour around a possible failure. This technique could be augmented with active probing of routers, perhaps looking for reset IP identifier counters, a sign that the in-memory state of a router had been cleared through reboot. In this way, the analysis of occasional failures to expose topology through BGP or other routing protocol dynamics [5] can be inverted to find the location of failures given a topology.

To detect load (a node's equivalent of network link congestion) an unpublished report by Cardwell and Savage suggests a method. They used the rate of change of IP identifiers as a relative estimate of Web server load – faster climb indicates that more packets have been generated, suggesting additional load on a Web server. Such an approach may not directly estimate a router's host processor load, as many of its activities (recomputing routing tables, collecting statistics) may not involve packet generation. We also know that routers take occasional "coffee breaks" where they are unresponsive while handling periodic maintenance tasks [73], but whether this can be applied to understand the processing load of a router is unclear.

4.2 Path Analysis Techniques

Detecting properties along paths has been well-studied. Vern Paxson's `tcpanaly` [76] is notable by its coverage of many different network behaviors by using "pas-

	Node	Path	Hop	Tree	Map
Network Management, Engineering, and Structure					
Location	GeoTrack [71], NetGeo [19]	–	–	–	–
Owner	Whois, BGP (us)	–	–	–	–
Connectivity	–	ping	traceroute	BGP inference [5], clustering [52]	Mercator [40], Lumeta [18] (x)
Routing Policy	–	–	–	( )	(x)
Demand Matrix	–	–	–	–	internal only: netflow-based [33], tomogravity [101], ( )
Media type	–	–	( )		( )
Network Performance, Behavior, and Pathologies					
Implementation	nmap [38], tbit [69]	–	–	–	–
Loss	–	tcpanaly [76], sting [83]	pchar [58] (x)	MINC [1], Venkat [70]	internal only: SNMP
Latency/Delay:	fsd [39]				
RTT	–	ping, king [41]	traceroute, pchar	( )	( )
One-way (OTT)	–	tcpanaly	cing [4]	( )	(( ))
Congestion (latency var.)	( )		((x))	MINC [1],	( )
Bandwidth:					
Capacity	–	bprobe [22], pathrate [29]	nettimer [54], pathchar	Katabi [50]	( )
Available	–	tcpanaly, pathload [47], chirps [82]	(( ))	( )	( )
Reordering	–	tcpanaly, sting-ii [11]	((x))	( )	( )
Duplication	–	tcpanaly	( )	( )	( )
Corruption	–	Stone[91]	((x))		( )
“Performance”	–	iperf [95], treno [65], TRAT [100], Olympics [9], SPAND [86], Ng [68]	b-find [2]	–	–
Failure	( )	Dahlin [23]	( )	( )	internal only: Sprint [44] ( )
Application mix	–	MCI [94], CAIDA [20]			

Table 1: Classification of tools by property and precision – some tools may interpret the same information to provide many metrics (such as pchar.) Their categorization above should be interpreted as a covering of space, not an enumeration of the features of measurement tools and projects. A ‘–’ represents a category that doesn’t apply, such as the bandwidth of a node. A “( )” represents a category that may prove useful in understanding, but has been un- (or under-) explored. A “(x)” represents a category in which we have already made significant progress. A “(( ))” suggests an ambitious but achievable direction. Blank cells may exist to the left, where path or hop measurements are redundant with projects that understand the map or a tree.

sive analysis" of packets that are already traversing the network combined with an understanding of TCP's reaction to various network behaviors. Other tools generate their own traffic and are considered "active." This distinction between active and passive techniques is not particularly important in the context of path analysis; an active technique can be made passive through patience (a technique used by early versions of nettimer [54]) and a passive technique can be made active by generating traffic. However, active techniques have greater potential for promotion to analyze per-hop behaviors because actively-generated traffic can be given different TTL values or be sent to intermediate routers. Active techniques also have the potential to measure the network as a whole rather than just the traffic passing a particular site.

Whether passive or active, path-analysis techniques can contribute to our goal of mapping the network in three important ways. First, the techniques provide an end-to-end check on the validity of hop-by-hop tools. Second, they provide detection of rare properties, like loss, reordering, and duplication, that can allow measurements of, say, hop-by-hop loss to be skipped if a path has no end-to-end loss. Third, they provide insight into how a property can be measured, allowing hop-wise tools to be built as extensions.

4.3 Hop Analysis Techniques

We now start to describe techniques of particular utility for reverse engineering an annotated map of the network. Hop-analysis techniques discover properties of individual IP-level links in the network; composing many of these hop measurements brings us closer to a network map. Network measurements that address properties of IP-level network links are typically based on TTL-limited probing, the same technique as at the heart of traceroute [46].

Bandwidth measurement tools such as `pathchar` [45], `clink` [30], and `pchar` [58] measure the serialization delay of variably-sized packets. Serialization delay is the time it takes a router interface to write each bit on a wire. These techniques are vulnerable to cross traffic, so require many packets to find samples that experience no queueing (only service time) in the network. Longer paths require more packets, not just to measure the extra links, but because the likelihood of experiencing no queueing is reduced. This makes a comprehensive study of bandwidth at scale difficult. Further, these techniques return erroneous results when intermediate, data-link layer switches are present [79, 75]. The ACCSIG tool [75] preserves the same probe traffic of variable packet sizes, but observes the delay variation pattern to avoid the need for minima filtering and the overhead it entails.

Recent approaches are more robust and deal with an absence of router cooperation using a "tailgating" technique. As with the previous tools, "tailgating" approaches also measure the serialization delay of a large packet, but observe bit delay by the effect it has on a "tailgating" packet that proceeds unobstructed through the rest of the network after the large packet's TTL expires. Tailgating has the advantage that it is possible to collect one-way timings, without conflating forward and reverse path properties, which is a significant advantage in that only the properties of the (known, single) forward path are involved. Further, tailgating does not require routers to generate ICMP messages. This enables the study of shared route segments including switched network paths (shared network segments below the IP layer and not otherwise visible) using methods from MINC, below. Using tailgating probes for topology inference has not been explored. Packet quartets [75] and cartouche probing [43, 42] improve upon the approach using more probe packets in a train. These approaches are less vulnerable to cross traffic on links leading up to the one being measured.

Another innovative tool analyzing hop properties is cing [4], which uses ICMP timestamps to measure per-link delays. The challenges for cing primarily involve clock synchronization, but a second requirement is that the ICMP timestamp requests traverse a prefix of the path to the destination being studied. This means that cing must ensure that the path to an intermediate router is a prefix of the path being studied using additional traceroutes. However, the advantage is that the properties of the return path back to the source and the forward path along to the ultimate destination are factored out.

4.3.1 Opportunities

Using NTP with tailgating approaches. NTP has the potential to provide more precise timestamps than the ICMP timestamp message or IP timestamp option. NTP may synchronize end-system clocks to microsecond resolution, and provide access to such a value, annotated with its precision and parent in the time synchronization tree, while ICMP timestamp messages are millisecond fields that may be updated less frequently. NTP has the advantage that remote sites do not need to run a special server, like that used by netperf [49] or iperf [95]. Interestingly, if NTP proves useful, it may be an interesting case of an application-layer protocol having more utility than one designed into the network-layer control plane.

Measuring and validating hop latency. Latency estimation is inevitably conflated with clock synchronization and understanding the relative skew, relative offset, and patterns of jumps in clocks. Paxson's work in [77] lays the groundwork for coping with individual problems on the clocks of hosts. Anagnostakis, *et al.* [4] dealt with the more numerous and less well-behaved clocks of routers with techniques to discard clock measurements that violated too many assumptions of stable behavior.

The techniques used in cing [4] assume symmetric paths, which makes clock normalization possible. However, the localization techniques of GeoPing and GeoTrack [92] suggest a different method, using geographic latency. It is possible that combining the two techniques would yield a better approach, or at the very least, help to validate cing's estimates of link delay.

Per-hop available bandwidth Merging the techniques that measure end-to-end available bandwidth with those of hop-wise capacity tools may permit development of a hop-wise available bandwidth estimator. We describe this in more detail in Section 6.

4.4 Tree Analysis Techniques

Tree analyses can be reached in two ways: the composition of hop-based measurement tools from a single site, or the use of path-measurement tools for inference of shared properties and shared segments. The latter approach is typically termed *tomography* by reference to the medical imaging procedures CT (computerized tomography) and PET (positron emission tomography), surprisingly not because both attempt to infer internal structure by external measurement, but because of similarities in the mathematical formulation of the problems.

Early work in tree analysis discovered shared properties using multicast trees. Multicast distribution of individual probe packets allow analyses like MINC [1] and MINT [13] to assign events such as loss and queueing to branches of the tree. When a probe packet is lost or delayed, all recipients of that packet along that branch see the loss or delay. Such work could be combined with multicast traceroute [35] to discover an IP-level multicast topology to annotate.

Padmanabhan, *et al.* [70] show that a similar analysis can be supported by unicast tools in a "passive" analysis. They study the TCP transactions that access the busy Microsoft Web site, use traceroute to determine which path is taken back to the client, then use Bayesian inference and Gibbs sampling to infer which link along the path was the site of persistent loss. Previous techniques, such as the multicast-based techniques above, actively generated traffic to measure loss rates; the authors' emphasis on passive analysis serves to distinguish the work from [31] and [13], but is not particularly useful. Each provides a combination of inference techniques needed to make unicast probing effective.

Akella, *et al.* [2] use a similar approach to determine where bottlenecks occur. They measure the performance of a series of adaptive UDP transactions, determine whether they see a bottleneck in the network (path-based detection) then analyze whether those paths traverse a peering point. Though currently a work in progress, with this analysis, they hope to determine whether peering points are often bottlenecks for network traffic.

The tree-measurement based "tomography" we describe here (where *path* measurements yield link properties as in [70, 1]) is the reverse of "tomography" as applied to traffic matrices (where *link* measurements yield path properties, as in [101, 97, 21]). Surprisingly, the dissimilarity ends there: both problems are addressed using the same methodology by Liang and Yu [57].

4.4.1 Opportunities

Efficient tree bandwidth measurement. Bandwidth (capacity) measurement takes many packets, so the goal is to keep it efficient. Multiple paths may traverse the same early links, and measuring these links only once can save traffic. Second, accuracy could be traded away for efficiency if estimates for faraway links do not need to be as accurate.

Tree reordering. While Padmanabhan [70] studies loss by observing retransmissions, the same analysis could discover sites of forward path reordering. Analysis of the stream of TCP acknowledgements exposes when packets were received out of order: when a TCP receiver sees an out of order packet, it immediately sends an acknowledgement of the last in-sequence packet received. This behavior of duplicate acknowledgements is just as observable as the retransmitted segments used to infer loss, and can be used to estimate where reordering occurs.

Web server-based measurement. Padmanabhan [70], Balakrishnan [9], and others have made use of the implicit conversations of a busy Web server to measure path properties. Use of a Web server or set of servers as measurement sources may simplify the problem of destinations that are otherwise invisible to the network because they are behind firewalls. One could imagine extending the single-server tomographic analyses to use several sites to the behavior of the

network as a whole, though this analysis may not be tractable.

4.5 External Mapping Techniques

In this subsection, we focus on Internet mapping techniques and the different approaches to measurement at the scale of the Internet. These techniques are relevant because they solve the challenges necessary to build a global picture of the network. In this section, we first contrast the work of Pansiot and Grad, Govindan and Tangmunarunkit (Mercator), Burch and Cheswick (Lumeta), and claffy, Monk and McRobb (Skitter). Images from three of these are in Figure 2. Then, we describe some open problems in network mapping.

Pansiot and Grad [72] measured a network topology using traceroute so that they could characterize multicast protocol proposals. In their study, the topology of the network influences the design of multicast trees in that many routers may not participate in forwarding, and many routers may simply forward a multicast packet as if it were a unicast packet, perhaps requiring less routing state. Pansiot and Grad collected two data sets. First, 12 sources were used to traceroute to 1,270 hosts, and 1 source (their own) to 5,000 hosts that had previously communicated with their department. To handle the problem of scale, Pansiot and Grad optimized their measurement tool, modifying traceroute in two ways. First, they did not probe three times per hop, but instead returned after the first successful response (retrying if no response was received). This has the potential to reduce the overhead of traceroute-based mapping by two-thirds. Second, they configured traceroute to start probing some number of hops into the trace, avoiding repeated, redundant probes to nearby routers. To provide an accurate map, they pioneered work on alias resolution[2]: the process of recognizing which IP interface addresses belong to the same router.

Burch, Cheswick, and Branigan [18, 24] use a different approach, increasing the number of destinations to ninety thousand, but using only a single source host. This work is primarily concerned with understanding small components, tracking change over time, and visualizing the overall network, and has resulted in Lumeta, a company specializing in mapping networks.

Govindan and Tanmunarunkit [40] in the Mercator project added "informed random address probing," in which traceroute destinations were chosen by consulting the global Internet routing table. Govindan, like Burch, uses a single source, but have many virtual sources by using source routing – a technique that "bounces" probes off remote routers. Like Pansiot, Govindan uses the source-address based alias resolution methodology, but enhances it using source-routed probes and repeated tests. Mercator's maps were validated by extracting components that belong to research ISPs and comparing those to the real maps.

CAIDA's Skitter [25, 15] project increases the number of destinations by picking Web servers (originally 29,000, though this has increased over time). They use six DNS servers as measurement sources to get a representative picture of the network. Like Burch, the Skitter project maintains a history of previous measurements to characterize the change in the network.

Different mapping projects approach efficiency in different ways, summarized in Table 2. Most mapping projects choose a small set of destinations, which is a form of sampling that may not be well justified. Different approaches solve the alias resolution problem for accuracy in Table 3. Previous approaches have used source-address-based alias resolution or ignored the problem. These differences may result from their different goals, summarized in Table 4. For example, those projects looking at evolution over long time scales have ignored the alias resolution problem,

These Internet mapping techniques focus specifically on the connectivity between routers. Structural views are limited to the opaque pictures like those in Figure 2. The whole Internet is a consistent goal, but sampling approaches are needed to make this feasible – from choosing limited destinations (Pansiot and Grad, Skitter) to choosing a single source (Lumeta) to choosing to run for weeks (Mercator).

The challenges for wide area Internet map construction are several; we list the three most important here. First, without many trees as vantage points, the observations of the network are biased in favor of high detail "bushyness" near measurement sources, and the appearance of long chains further away [55]. This need for many sources stresses the scalability of a measurement. As a further consequence, alias resolution becomes more important as different vantage points see different interfaces. However, unreachable routers that do not respond to alias resolution probes, the second challenge, thwart efforts to build an accurate map [89]. Finally, anonymous routers that do not even send time-exceeded messages to TTL-limited probes threaten to limit the ability to construct a map at all [98].

4.5.1 Opportunities

Focus on a smaller network. One approach to scaling tools to map networks is avoid the scaling problem as much as possible by focusing only on a sub-graph. This is the Rocketfuel approach, described in Section 5.

[2] Pansiot and Grad termed them *synonyms*; we prefer the Mercator terminology.

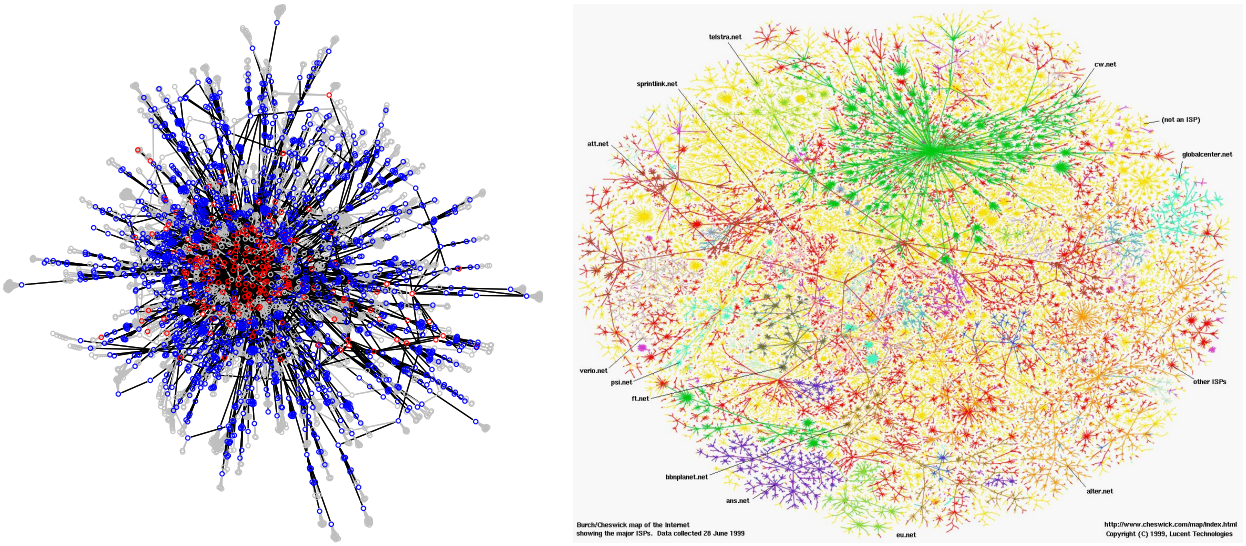

Figure 2: Internet map visualizations from different projects. At left is Mercator's view of an ISP named Cable and Wireless. At right is Burch and Cheswick's map of the Internet from 1999; Cable and Wireless is the green star in the upper right. For a larger picture, see [17]. These maps show two very different structures for the same network. The difficulty of interpreting this data, let alone verifying its accuracy, prevents its use.

Project	Approach to Efficiency
Pansiot and Grad [72]	Few destinations (1200 to 5000), tuned traceroute.
Lumeta [18]	Single source (more tree than map)
Mercator [40]	Run for three weeks.
Skitter [25]	Limit destinations to active Web servers.

Table 2: Different mapping projects achieve efficiency in mapping the Internet in different ways.

Map by DNS scanning. The DNS names associated with router interfaces include a lot of information about their location and connectivity. For example, Abilene routers are named like sttlng-dnvrng.abilene.ucaid.edu, implying a point-to-point link between Seattle and Denver. Adjacent IP addresses are likely to belong to the same network link, as IP network addresses are assigned first to network links, then IP addresses to their interfaces. The DNS approach has the opportunity to observe otherwise unseen interface addresses, useful for ensuring a complete map. The questions raised by this approach are in which ISPs can be accurately mapped using this approach – for example, will networks that use MPLS or other switched links extensively confound the analysis or be much easier to understand? One challenge will be in discovering and understanding peering relationships, because the DNS names associated with peering link interface addresses have less structure.

4.6 Map annotation and synthesis of properties

Connectivity alone can help to answer questions about the robustness of the network to partition and the state required by routing protocols and for multicast forwarding. However, it tells us little about performance because the capacity of links can vary over 6 orders of magnitude (33 kilobits per second modem to 40 gigabit per second OC-768 links), and a single network hop may as easily represent crossing a machine room as crossing an ocean.

Annotating a map with node properties does not appear to pose a significant challenge. For example, DNS names recover geography [71], and router role [89]. DNS with address space information from BGP also gives the AS that manages a router [63]. More challenging is adapting link-measurement tools to run at scale, as described in the previous section on tree measurement.

Project	Approach to Accuracy
Pansiot and Grad	Basic source-address alias resolution
Mercator	Source routed probing and retried alias resolution
Lumeta	No alias resolution required - single source
Skitter	Several sources, but no alias resolution described

Table 3: Different mapping projects achieve accuracy in mapping the Internet in different ways.

		Desired map	
		Single snapshot	Tracked across time
Sources	Few	Pansiot & Grad	Lumeta
	Many	Mercator	Skitter

Table 4: Different mapping projects along two axes, how many sources of traffic, and whether or not an evolutionary view is desired.

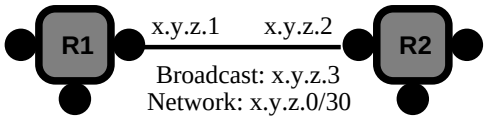

Figure 3: Address allocation of a point-to-point network. Shaded boxes represent routers, adjacent circles interfaces, and the line between them a point-to-point link. A /30 prefix (the first 30 bits of a 32-bit IP address are significant) leaves two addresses for use, two reserved as broadcast and network addresses.

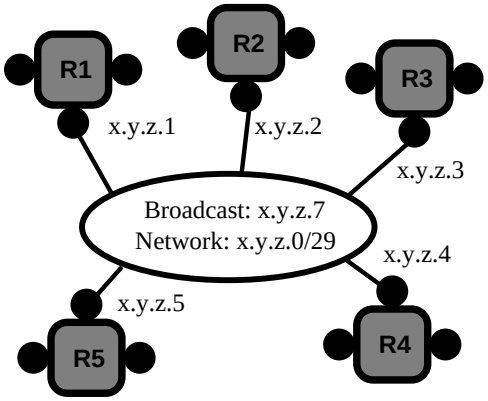

Figure 4: Address allocation of a multiple-access network. A /29 prefix (the first 29 bits of a 32-bit IP address are significant) leaves six addresses for use, two reserved as broadcast and network addresses. Those that end with 3 and 4 would have been unavailable if used in a point-to-point network.

4.6.1 Opportunities

Inferring network policy. A map of connectivity annotated with link weights is sufficient for modeling paths taken through an individual network. This allows us to recognize not just the connections between routers, but an aspect of their configuration to handle the workload. Additional characterization of the policy across pairs of ISPs proves more difficult, but it may be possible to classify policies into a handful of categories that represent the inter-ISP relationship. Our work on recovering policy is described in Section 5.

Media type inference. Underlying each IP-level hop is a data-link layer topology, such as an Ethernet, a wireless link, a ring, or a point-to-point link. If we can tell what data-link network is used, we can simplify the measured topology from the clique of IP-level connectivity measured by traceroute to the concentrated star or ring topology of a shared network when it exists. Each has particular behaviors that may expose just what type of network is used. For example, artifacts that influence measurements of bandwidth in [79, 75] suggest a method to identify switched networks. A switched network is one in which an intermediate node participates in forwarding a packet at a time, but does not participate in IP – this means that it delays, buffers, and even may drop traffic, but is not otherwise visible because it does not respond to traffic. Such a switch can be detected when when bandwidth measured by variable packet size approaches (such as pathchar) is half of a common value (eg. 50 Mbits instead of 100) or less than the delivered performance measured by a tool like pathload or iperf. Perhaps wireless links may be inferred by observation of the inter-frame spacing used in collision avoidance.

Interpreting DNS names and IP address allocation

provides an alternative to hunting telltale performance attributes. Many router interfaces have names that imply they belong to SONET rings, gigabit Ethernet links, or T-3 links. Even those that don't are likely to have IP addresses allocated from ranges that make it obvious whether they belong to multiple-access networks. Each network is given a (small) prefix, and typically the two addresses at the top and bottom of the range are reserved as broadcast and network addresses. Point-to-point links are given /30 prefixes, which reserves those addresses that end in 00b or 11b, leaving those that end with 01b or 10b available for both ends of the link. This is shown in Figure 3. Multiple-access networks will be allocated larger prefixes, meaning that some addresses on multiple access networks should end in, for example 011b (3) or 100b (4). An example allocation for multiple-access networks is shown in Figure 4. An interface address that ends in either 00b or 11b is likely to belong to a multiple access network of some sort; adjacent addresses are likely to connect to the same network.

5 Methodology

In this section, we summarize our recent work that enables accurate and efficient annotated map construction. While my proposed thesis focuses on discovering design and configuration, we also describe supporting performance and debugging tools that more generally discover attributes required to evaluate network protocols in simulation.

The Rocketfuel [89] project focused hundreds of public traceroute servers on the task of mapping a handful of large ISP networks. The goal was to measure very accurate maps using the expectation that using many more vantage points would expose many more paths. Our insight was to combine the information available through DNS and BGP to guide the mapping process and interpret the result. We also innovated in the design of an alias resolution procedure that can recognize which IP addresses (assigned to interfaces) belong to the same router; a task of crucial importance for an accurate map. The experiences gained on this project shape our later work. First, alias resolution is difficult and error-prone, but necessary for an accurate result. Second, large scale network measurements inevitably trigger security alarms resulting in some complaints from remote system administrators, so good citizenship and ties with local system administrators are important. Third, filtering out measurement errors is problematic – it is not always obvious when traceroute measures a false link – so robust tools that can detect signs of inaccuracy and avoid giving false results are important.

In follow on work [60], we combined the measured maps with the original traceroutes collected to model the internal routing policy of the ISPs. The insight is that given a map, the observed, chosen path is shorter (has lower cost) than alternate paths. By assembling rules of this form, we can use linear programming to infer a cost for each link that is consistent with observed routing. Three insights make this possible. First, many constraints are redundant – there are very many alternate paths from one point to another, only a few of these are reasonable contenders. Second, the ingress-to-egress measurements are incomplete for each pair of POPs, however, each sub-path of a chosen path is also a chosen path, improving overall completeness. Third, some noise is present in the data as occasionally, perhaps due to router failure, an alternate path is actually taken, so constraint hierarchies [14] are used to deal with this error. The final solution that assigns weights to links is not unique, so the actual link-cost settings are likely to differ from those that are inferred. However, the weights provide a simple, compact description of observed routing.

We proceeded further to try to understand inter-domain policy, or how traffic is managed when it crosses from one ISP to the next [88]. When studying global Internet routing policy, we made simplifying assumptions about the topology and used the geography to understand the indirectness of chosen routes. That is, when paths are diverted or special cased, traffic engineering is indicated because the direct route is avoided – the ISP must have some reason to avoid the direct, simplest path. Using these analyses, we found that we could infer the relationship between ISPs, for example, whether they use "early" or "late" exit routing. The combination of Rocketfuel with internal and external policy completes a picture of network design and observable configuration; what remains undiscovered are the specific choices of configuration values, the goals achieved by these configuration choices, and unused backup connections that carry no traffic. However, topology and policy have been sufficient to parameterize recent studies of traffic engineering approaches [7, 102].

The need for robust, flexible tools for network measurement motivated the design of Scriptroute [90], which is an environment for distributed network measurement. The Scriptroute approach combines mobile, untrusted code with a security policy that restricts "unsafe" network behaviors, such as port-scanning and denial of service attacks. Using Scriptroute, we have modified network mapping tools for robustness to routing changes (rockettrace), integrated Rocketfuel's alias resolution techniques with the DNS to handle unresponsive routers (ally), and implemented efficient dis-

tributed mapping techniques (reverse path tree). Scriptroute allows us to be ambitious in the scale of our network mapping without stressing the public traceroute server infrastructure used in Rocketfuel.

With this architecture for flexible measurement, we have constructed tools to measure and locate the pathologies previously studied only on a whole path basis [61]. We term this approach "Internet diagnosis," and the techniques allow a hop-by-hop measurement of congestion by observing variation in queueing delay, and of forward per-hop loss and reordering using IP identifiers. While the goal of the study was to enable users to assign blame for poor perceived performance, these lightweight techniques, integrated with network mapping, help to expose the heterogeneity of the network for simulation and evaluation. Although these tools do not directly expose network design, they form a basis for understanding factors involved that guide the evolution of networks.

6 Near-future work

A long-term goal of this network measurement work is to provide a measured network for simulation studies. This measured network should be as real, demanding accuracy in measurement, and as complete (both in detail and annotations) as possible. In the short-term, our goal is to measure the properties needed to explain and understand the network's design, considering performance only as it helps explain the network's design. A complete picture of a network's design requires at least a glimpse into the workload it supports. This workload is expressed as a *traffic matrix*, the amount of traffic that traverses the network from each source to each destination. The challenge is recovering this information without using management tools like netflow or SNMP, which report link utilization to administrators. In this section, we outline possible approaches to construct a measurement tool that recovers link utilization, or its complement, available (idle) bandwidth.

There are two potential techniques to estimate link utilization, or conversely, unused or available bandwidth. A measurement of utilization coupled with capacities makes it possible to use the methods in [97, 101] to infer a traffic matrix over a routing inferred by [60]. The traffic matrix is relevant because the effectiveness of a network is evaluated in terms of its ability to handle its workload, and it is reasonable to expect that the network has been tuned with its workload in mind.

The first approach to measure available bandwidth is to load links to force them to queue traffic; the available bandwidth is that which could be consumed without lengthening the queue. This approach is that of pathload [47], which should support easy extension to be used in a per-hop basis. Pathload adaptively determines how much additional data must be sent through a bottleneck to cause queueing. Combining this with the tailgating technique of nettimer [54] would allow loading links before the bottleneck without loading a path's bottleneck link. However, the need to load links makes it impossible to measure the available capacity on links after a bottleneck. For purposes of ISP mapping, this may be a real problem if the access to the ISP is a bottleneck link, at least relative to the network backbone.

The second approach is a measurement of the distribution of queueing delay on the link. The Delphi [82] approach uses a series of "packet chirps" to observe cross traffic that modifies the spacing of chirped packets. One approach to estimating the available capacity on hops before the end-to-end bottleneck would be to send TTL-limited packet chirps. Alternately, one could measure the distribution of queuing delay directly, using approaches from cing [4]. Any measurements of forward path delay greater than the minimum suggest queueing along the path. Ideally, cross traffic even after the bottleneck should have an effect on the distribution of delay samples, which may allow a reasonable guess as to its utilization using Little's formula. Little's formula relates the length of the queue to the arrival rate and the expected wait time [51]. Alouf, *et al.* [3] use Little's formula to develop a methodology to estimate the cross traffic sharing a bottleneck link (as well as the queue capacity). This approach was evaluated only in simulation, but appears promising. The most accurate moment-based estimator presented in [3] relates the measured loss probability (P_L), measured average queue delay (R), and known link capacity (μ) with utilization (ρ) using the formula:

$$R = \frac{1}{\mu(1-\rho)} - \frac{P_L \log(P_L/(1-\rho(1-P_L)))}{\mu(1-\rho)(1-P_L)\log(\rho)} \quad (1)$$

An interesting engineering question is how (and whether) to combine capacity measurement with available bandwidth estimation. Such optimizations may be of particular importance in building scalable tools.

The development of two independent techniques based on different methodologies (loading the link until queueing and measuring queue delay distributions) has the potential to support cross-validation. A second validation strategy would be to compare with the Abilene network statistics available at http://www.abilene.iu.edu/noc.html. Unfortunately, these are somewhat digested for graphical presentation, so a different access path may be needed for real-time comparison between the tool's measurement and the SNMP-retrieved utiliza-

tion statistics.

By measuring link utilization to recover a traffic matrix, a complete picture of network engineering can be recovered from the outside. Vardi introduces the problem of recovering traffic matrices from link loads [97], including methods for handling multi-path routing. However, Vardi assumes a Poisson arrival process, which motivates the work of Cao *et al.* [21], who use trace data to correct the Poisson assumption and validate using a small network. Recent work [101] has sought to make such methods practical and robust to error by integrating a gravity model. A gravity model relates the amount of traffic between pairs of nodes as the product of their (destination-independent) traffic volume, in much the same way as the gravitational force between bodies is proportional to the product of their masses. This tomographic gravity model provides a methodology to scale the analysis. Measured link utilizations combined with a little guesswork based on the distribution of address space (say, there are more machines in San Francisco than Sacramento) to parameterize the gravity model seem promising for recovery of a full traffic matrix.

The major limitation to the effectiveness of these measurements will likely be scaling to the very fast link speeds in the backbone of the Internet. The transmission time of a large (1500 byte) vs. small (40 byte) packet across a gigabit network is only 12 μsec, straining the precision of commodity clocks. This means that measurements requiring precise timing are likely to be dwarfed by noise without specialized hardware. The approaches for estimating link capacity assume that this 12 μsec difference can be measured accurately. Without an accurate link capacity measure, the available bandwidth measure may not in the end give link utilization, even though its use of loss rate and queueing delay estimation make it less susceptible to imprecise clocks.

While the limited detail and accuracy of external inference will prevent such tools from replacing internal network management tools, we hope that measured data will prove useful to the development of new protocols and the evaluation of the Internet's robustness.

7 Conclusions and (distant) future work

In this paper, we have described the techniques of network measurement with an eye toward inferring its internal structure and design. While the prevalence of many properties is well-studied, developing techniques to discover where they occur with any precision is challenging. Adapting these measurement tools to work at the scale of the Internet requires careful design and sometimes new approaches.

In this thesis proposal, my focus is the external understanding of the design and management of networks that make up the Internet. Future work in measurement can complete the construction of an annotated map that allows large parts of the Internet to parameterize a simulation. Periodic measurement can construct an evolving history of the design and performance of these networks, with the potential to characterize the long term effects of conspicuous failures like the Baltimore tunnel fire [81].

The measurements themselves enable new studies into how ISPs can better manage their networks, perhaps using new algorithms to set link policy. A motivating goal of Rocketfuel was to recover topologies upon which we could evaluate robust interior gateway routing protocols. Further afield, researchers and developers can test new protocols or designs on individual network topologies and workloads to demonstrate to the network operator that they solve a real problem, and quantify the expected benefit of deployment.

References

[1] A. Adams, T. Bu, R. Cáceres, N. G. Duffield, T. Friedman, J. Horowitz, F. LoPresti, S. B. Moon, V. Paxson, and D. Towsley. The use of end-to-end multicast measurements for characterizing internal network behavior. *IEEE Communications Magazine*, 2000.

[2] A. Akella, S. Seshan, and A. Shaikh. An empirical evaluation of wide-area Internet bottlenecks. In *ACM SIGMETRICS*, June 2003. (poster).

[3] S. Alouf, P. Nain, and D. Towsley. Inferring network characteristics via moment-based estimators. In *IEEE INFOCOM*, Apr. 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*, Nov. 2002.

[6] D. Anderson, H. Balakrishnan, M. F. Kaashoek, and R. Morris. Resilient overlay networks. In *SOSP*, Oct. 2002.

[7] D. Applegate and E. Cohen. Making intra-domain routing robust to changing and uncertain traffic demands: Understanding fundamen-

tal tradeoffs. In *ACM SIGCOMM*, Aug. 2003. (to appear).

[8] R. Atkinson and S. Floyd. IAB concerns and recommendations regarding Internet research and evolution. Internet draft: draft-iab-research-funding-00.txt, Feb. 2003.

[9] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, M. Stemm, and R. H. Katz. TCP behavior of a busy Internet server: Analysis and improvements. In *IEEE INFOCOM*, Mar. 1998.

[10] A. Basu and J. Riecke. Stability issues in OSPF routing. In *ACM SIGCOMM*, Aug. 2001.

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

[12] J. C. R. Bennett, C. Partridge, and N. Schectman. Packet reordering is not pathological network behavior. *IEEE/ACM Transactions on Networking*, 7(6), Dec. 1999.

[13] A. Bestavros, J. Byers, and K. Harfoush. Inference and labeling of metric-induced network topologies. In *IEEE INFOCOM*, June 2002.

[14] A. Borning, B. Freeman-Benson, and M. Wilson. Constraint hierarchies. *Lisp and Symbolic Computation*, 5(3), 1992.

[15] A. Broido and kc claffy. Internet topology: connectivity of IP graphs. In *SPIE Int'l symposium on Convergence of IT and Communication*, Aug. 2001.

[16] T. Bu and D. Towsley. On distinguishing between Internet power law topology generators. In *IEEE INFOCOM*, Apr. 2002.

[17] H. Burch and B. Cheswick. Burch/cheswick map of the Internet. http://research.lumeta.com/ches/map/gallery/isp-ss.gif, June 1999.

[18] H. Burch and B. Cheswick. Mapping the Internet. *IEEE Computer*, 32(4):97–98, 102, Apr. 1999.

[19] CAIDA. NetGeo - the Internet geographic database. http://www.caida.org/tools/utilities/netgeo/.

[20] CAIDA. Traffic workload overview. http://www.caida.org/Learn/Flow/tcpudp.html, June 1999.

[21] J. Cao, D. Davis, S. VanderWiel, and B. Yu. Time-varying network tomography: Router link data. *Journal of the American Statistical Association*, 95(452):1063–1075, Dec. 2000.

[22] R. L. Carter and M. E. Crovella. Dynamic server selection using bandwidth probing in wide-area networks. In *IEEE INFOCOM*, 1997.

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

[24] B. Cheswick, H. Burch, and S. Branigan. Mapping and visualizing the Internet. In *USENIX Annual Technical Conference*, 2000.

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

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

[27] 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.

[28] Computer Science and Telecommunications Board, National Research Council. *The Internet Under Crisis Conditions: Learning from September 11*. The National Academies Press, 2003.

[29] C. Dovrolis, P. Ramanathan, and D. Moore. What do packet dispersion techniques measure? In *IEEE INFOCOM*, Apr. 2001.

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

[31] N. G. Duffield, F. LoPresti, V. Paxson, and D. Towsley. Inferring link loss using striped unicast probes. In *IEEE INFOCOM*, 2001.

[32] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the Internet topology. In *ACM SIGCOMM*, 1999.

[33] A. Feldmann, A. Greenberg, C. Lund, and N. Reingold. Deriving traffic demands for operational IP networks: Methodology and experience. In *ACM SIGCOMM*, Sept. 2000.

[34] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, and J. Rexford. Netscope: Traffic engineering for IP networks. *IEEE Network Magazine*, Mar. 2000.

[35] W. Fenner and S. Casner. `mtrace`. Internet Draft: draft-idmr-traceroute-ipm-04.txt, Feb. 1999.

[36] S. Floyd and V. Paxson. Difficulties in sumulating the Internet. *IEEE/ACM Transactions on Networking*, Feb. 2001.

[37] P. Francis, S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, and L. Zhang. IDMaps: A global Internet host distance estimation service. *IEEE/ACM Transactions on Networking*, Oct. 2001.

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

[39] R. Govindan and V. Paxson. Estimating router ICMP generation delays. In *Passive & Active Measurement (PAM)*, Mar. 2002.

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

[41] K. P. Gummadi, S. Saroiu, and S. D. Gribble. King: Estimating latency between arbitrary Internet end hosts. In *ACM SIGCOMM Internet Measurement Workshop*, Nov. 2002.

[42] K. Harfoush, A. Bestavros, and J. Byers. Measuring bottleneck bandwidth of targeted path segments. Technical Report BUCS-2001-016, Boston University CS, July 2001.

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

[44] 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.

[45] V. Jacobson. Pathchar. ftp://ftp.ee.lbl.gov/pathchar.

[46] V. Jacobson. Traceroute. ftp://ftp.ee.lbl.gov/traceroute.tar.Z.

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

[48] C. Jin, Q. Chen, and S. Jamin. Inet: Internet topology generator. Technical Report CSE-TR-433-00, University of Michigan, EECS dept., 2000. http://topology.eecs.umich.edu/inet/inet-2.0.pdf.

[49] R. Jones. Netperf. http://www.netperf.org/.

[50] D. Katabi, I. Bazzi, and X. Yang. A passive approach for detecting shared bottlenecks. In *IEEE Int'l Converence on Computer Communications and Networks (ICCCN)*, 2001.

[51] L. Kleinrock. *Queueing Systems*, volume 1. John Wiley & Sons, 1975.

[52] B. Krishnamurthy and J. Wang. On network-aware clustering of Web clients. In *ACM SIGCOMM*, Sept. 2000.

[53] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian. Delayed Internet routing convergence. In *ACM SIGCOMM*, Sept. 2000.

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

[55] A. Lakhina, J. Byers, M. Crovella, and P. Xie. Sampling biases in IP topology measurements. In *IEEE INFOCOM*, Apr. 2003.

[56] A. Lakina, J. W. Byers, M. Crovella, and I. Matta. On the geographic location of Internet resources. In *ACM SIGCOMM Internet Measurement Workshop*, 2002.

[57] G. Liang and B. Yu. Pseudo likelihood estimation in network tomography. In *IEEE INFOCOM*, Apr. 2003.

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

[59] R. Mahajan, S. M. Bellovin, S. Floyd, J. Ioannidis, V. Paxson, and S. Shenker. Controlling high-bandwidth aggregates in the network (extended version). http://www.aciri.org/pushback/, July 2001.

[60] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson. Inferring link weights using end-to-end measurements. In *ACM SIGCOMM Internet Measurement Workshop*, Nov. 2002.

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

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

[63] Z. M. Mao, J. Rexford, J. Wang, and R. Katz. Towards an accurate as-level traceroute tool. In *ACM SIGCOMM*, Aug. 2003. (to appear).

[64] A. H. Maslow. A theory of human motivation. *Psychological Review*, (50), 1943.

[65] M. Mathis. Diagnosing Internet congestion with a transport layer performance tool. In *INET'96*, June 1996.

[66] N. McCarthy. fft. http://www.mainnerve.com/fft/.

[67] The national strategy to secure cyberspace, Feb. 2003. http://www.whitehouse.gov/pcipb/cyberspace_strategy.pdf.

[68] T. S. E. Ng, Y. Chu, S. G. Rao, K. Sripanidkulchai, and H. Zhang. Measurement-based optimization techniques for bandwidth-demanding peer-to-peer systems. In *IEEE INFOCOM*, Apr. 2003.

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

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

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

[72] J.-J. Pansiot and D. Grad. On routes and multicast trees in the Internet. *ACM Computer Communication Review*, 28(1):41–50, Jan. 1998.

[73] K. Papagiannaki, S. Moon, C. Fraleigh, P. Thiran, F. Tobagi, and C. Diot. Analysis of measured single-hop delay from an operational backbone network. In *IEEE INFOCOM*, June 2002.

[74] K. Park and H. Lee. On the effectiveness of route-based packet filtering for distributed DoS attack prevention in power-law internets. In *ACM SIGCOMM*, Aug. 2001.

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

[76] V. Paxson. End-to-end Internet packet dynamics. In *ACM SIGCOMM*, Sept. 1997.

[77] V. Paxson. On calibrating measurements of packet transit times. In *ACM SIGMETRICS*, 1998.

[78] G. Philips, S. Shenker, and H. Tangmunarunkit. Scaling of multicast trees: Comments on the Chuang-Sirbu scaling law. In *ACM SIGCOMM*, Aug. 1999.

[79] 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.

[80] P. Radoslavov, H. Tangmunarunkit, H. Yu, R. Govindan, S. Shenker, and D. Estrin. On characterizing network topologies and analyzing their impact on protocol design. Technical Report CS-00-731, USC, 2000.

[81] A. Ratner. Train derailment severs communications. *The Baltimore Sun*, July 2001. http://www.sunspot.net/news/local/bal-te.bz.fiber20jul20,0,1743066.story.

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

[83] S. Savage. Sting: a TCP-based network measurement tool. In *USITS*, Oct. 1999.

[84] S. Savage, A. Collins, E. Hoffman, J. Snell, and T. Anderson. The end-to-end effects of Internet path selection. In *ACM SIGCOMM*, Aug. 1999.

[85] S. Savage, D. Wetherall, A. Karlin, and T. Anderson. Practical network support for IP traceback. In *ACM SIGCOMM*, Aug. 2000.

[86] S. Seshan, M. Stemm, and R. H. Katz. SPAND: Shared passive network performance discovery. In *USITS*, Dec. 1997.

[87] A. C. Snoeren, C. Partridge, L. A. Sanchez, C. E. Jones, F. Tchakountio, S. T. Kent, and W. T. Strayer. Hash-based IP traceback. In *ACM SIGCOMM*, Aug. 2001.

[88] N. Spring, R. Mahajan, and T. Anderson. Quantifying the causes of path inflation. In *ACM SIGCOMM*, Aug. 2003. (to appear).

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

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

[91] J. Stone and C. Partridge. When the CRC and TCP checksum disagree. In *ACM SIGCOMM*, Aug. 2000.

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

[93] H. Tangmunarunkit, R. Govindan, S. Jamin, S. Shenker, and W. Willinger. Network topologies, power laws, and hierarchy. Technical Report CS-01-746, USC, 2001.

[94] K. Thompson, G. J. Miller, and R. Wilder. Wide-area Internet traffic patterns and characteristics. *IEEE Network*, 11(6):10–23, Nov. 1997.

[95] A. Tirumala, F. Qin, J. Dugan, and J. Ferguson. Iperf. http://dast.nlanr.net/Projects/Iperf/, May 2002.

[96] M. C. Toren. tcptraceroute. http://michael.toren.net/code/tcptraceroute/.

[97] Y. Vardi. Network tomography: Estimating source-destination traffic intensities from link data. *Journal of the American Statistical Association*, 91(433):365–377, Mar. 1996.

[98] B. Yao, R. Viswanathan, F. Chang, and D. Waddington. Topology inference in the presence of anonymous routers. In *IEEE INFOCOM*, Apr. 2003.

[99] E. W. Zegura, K. Calvert, and S. Bhattacharjee. How to model an internetwork. In *IEEE INFOCOM*, 1996.

[100] Y. Zhang, L. Breslau, V. Paxson, and S. Shenker. On the characteristics and origins of Internet flow rates. In *ACM SIGCOMM*, 2002.

[101] 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.

[102] Y. Zhang, M. Roughan, C. Lund, and D. Donoho. An information-theoretic approach to traffic matrix estimation. In *ACM SIGCOMM*, Aug. 2003. (to appear).