The Case for Resilient Overlay Networks

David G. Andersen, Hari Balakrishnan, M. Frans Kaashoek, and Robert Morris
MIT Laboratory for Computer Science
Cambridge, MA 02139
{dga, hari, kaashoek, rtm}@lcs.mit.edu
http://nms.lcs.mit.edu/ron/

Abstract

This paper makes the case for Resilient Overlay Networks (RONs), an application-level routing and packet forwarding service that gives end-hosts and applications the ability to take advantage of network paths that traditional Internet routing *cannot* make use of, thereby improving their end-to-end reliability and performance. Using RON, nodes participating in a distributed Internet application configure themselves into an overlay network and cooperatively forward packets for each other. Each RON node monitors the quality of the links in the underlying Internet and propagates this information to the other nodes; this enables a RON to detect and react to path failures within several seconds rather than several minutes, and allows it to select application-specific paths based on performance. We argue that RON has the potential to substantially improve the resilience of distributed Internet applications to path outages and sustained overload.

1 Introduction

Today's wide-area Internet routing architecture organizes the Internet into autonomous systems (ASes) that peer with each other and exchange information using the Border Gateway Protocol (BGP), Version 4 [10]. This approach scales well to a large number of connected networks, but this scalability comes at the cost of the increased vulnerability to link or router failures. Various recent studies have found that path failures are common and that the convergence time after a problem is detected is usually on the order of several minutes [5], and that path outages, routing anomalies, and active denial-of-service attacks cause significant disruptions in end-to-end communication [1, 8]. This reduces the reliability of end-to-end communication over Internet paths and therefore adversely affects the reliability of distributed Internet applications and services.

We propose *Resilient Overlay Networks* (RONs) as an architecture to improve the reliability of distributed applications on the Internet. Each application creates a RON from its participating nodes. These nodes are typically spread across multiple ASes, and see different routes through the Internet. RON nodes cooperate with each other to forward data on behalf of any pair of communicating nodes in the RON, thus forming an *overlay network*. Because ASes are independently administered and configured, underlying path failures between communicating nodes usually occur independently. Thus, if the underlying topology has physical path redundancy, it is often possible for a RON to find paths between RON nodes even if Internet routing protocols such as BGP (that are optimized for scalability) cannot find them.

Nodes in a RON self-configure into the overlay by exchanging information across the underlying Internet paths. Each RON node has "virtual links" to all other RON nodes, which it uses to maintain connectivity and exploit the underlying IP network's redundancy. When a RON node receives a packet destined for another, it looks for the destination in an application-specific forwarding table, encapsulates the packet in a RON packet, and ships it to the next RON node. In this way, the packet moves across the overlay via a sequence of RON nodes until it reaches the intended destination.

To find and use alternate paths, a RON monitors the health of the underlying Internet paths between its nodes, dynamically selecting paths that avoid faulty or overloaded areas. The goal is to ensure continued communication between RON nodes despite failures due to outages, operational errors, or attacks in the underlying network. RON nodes infer the quality of virtual links using active probing and passive observation of traffic, and exchange this information using a routing protocol. Each node can use a variety of performance metrics, such as packet loss rate, path latency, or available bandwidth to select an appropriate application-specific path. This approach has potential because each RON is small in size (less than fifty nodes), which allows aggressive path monitoring and maintenance.

A RON ensures that as long as there is *an* available path in the underlying Internet between two RON nodes, the RON application can communicate robustly even in the face of problems with the "direct" (BGP-chosen) path between them. The limited size of each independent RON is not a serious limitation for many applications and services. A video conferencing program may link against a RON library, forming a routing overlay between the participants in the con-

This research was sponsored by the Defense Advanced Research Projects Agency (DARPA) and the Space and Naval Warfare Systems Center, San Diego under contract N66001-00-1-8933.

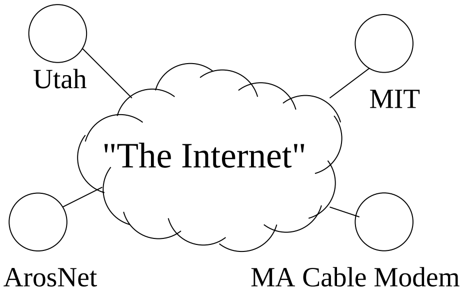

Figure 1: A common (mis)conception of Internet interconnections.

ference. Alternatively, a RON-based application-aware IP packet forwarder may be located at points-of-presence in different ASes, forming an "Overlay ISP" that improves the reliability of Internet connectivity for its customers.

This paper presents the case for developing distributed applications using RON (Section 2), outlines an approach by which RONs may be architected (Section 3), relates RON to previous work (Section 4), and concludes with a research agenda for future work (Section 5).

2 The Case for RONs

A common, but incorrect, view of the topology of the Internet is that institutions and companies connect to "The Great Internet Cloud." Figure 1 illustrates an example of four sites, **MIT**, **Utah**, **ArosNet**, and **MediaOne**, connected to the Internet cloud. In this view, the Internet is very robust, rapidly routing packets around failures and traffic overload, and providing near-perfect service.

Unfortunately, this ideal view of the Internet cloud is far from reality. The Internet Service Providers (ISPs) constituting the Internet exchange routing information using BGP, which is designed to scale well at the expense of maintaining detailed information about alternate paths between networks. To avoid frequent route changes that may propagate through many other ASes, frequent route announcements and withdrawals are damped; furthermore, convergence times on route changes take many minutes [5] with currently deployed BGP implementations. Last but not least, there are numerous financial, political, and policy considerations that influence the routes announced via BGP.

ISPs typically provide two types of connectivity: "transit" and "peering." If the ISP provides *transit* for a customer A, it tells other ISPs that they may reach A through the ISP. If an ISP has a *peering* relationship with A, it keeps this knowledge to itself; the ISP and its customers can reach A via this link, but the rest of the Internet may not. Peering relationships are often free, because they enable the more efficient exchange of packets without placing the burden of hauling packets on either partner, but globally announced transit relationships almost always involve some form of settlement.

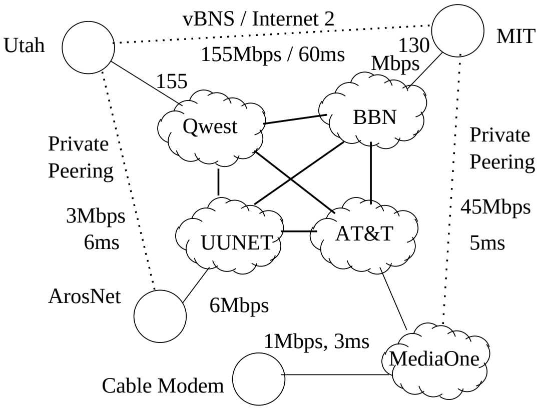

Figure 2: The details of Internet interconnections. Dotted links are *private* and are not announced globally.

Figure 2 redraws Figure 1 to reflect reality. MIT is connected to the Internet via BBN, and to Internet2. It has a private peering link to MediaOne in Cambridge (MA), so students can quickly connect to their MIT machines from home. Utah is connected to the Internet via Qwest, to Internet2, and to a local ISP, ArosNet, via a private peering link. ArosNet is connected to the Internet via UUNET, and MediaOne is connected to the Internet via AT&T. In this example, several desirable paths are unavailable globally: the private peering links for financial reasons (the parties have no apparent incentive to provide transit for each other) and the Internet2 connections because it is a research network.

These interconnections show two reasons BGP is unable to ensure "best"—or sometimes even "good"—routes, and route around problems even when different physical paths are available. The first reason, explained above, is a consequence of the economics and administration of peering relationships. The second relates to scalability.

For communication costs to scale well, BGP must simplify routing data enormously; for computational scalability, its decision metrics must be both simple and stable. BGP primarily uses its own hop-counting mechanism to determine routes and it exports a single "best" route for forwarding packets. This causes three important problems: first, as noted in the Detour study [11], BGP may make suboptimal routing decisions. Second, BGP does not consider path performance when making routing decisions, and so cannot route around a path outage caused by traffic overload. The result is that path outages can lead to significant disruptions in communication [1]. Third, BGP may take several minutes to stabilize in the event of a route change or link failure [5]. The result is that today's Internet is easily vulnerable to router faults, link failures, configuration or operational errors, and malice—hardly a week goes by without some serious prob-

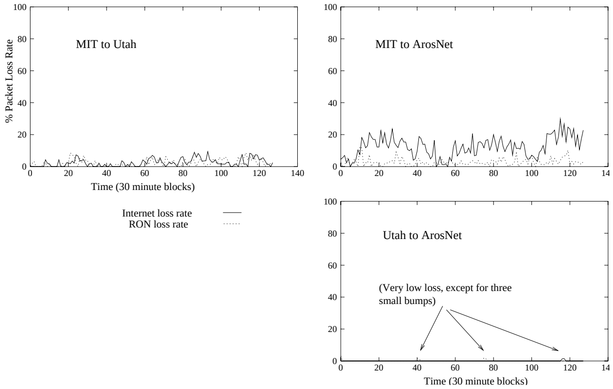

Figure 3: The upper right figure shows the loss rate with and without RON between `MIT` and `ArosNet`. RON was able to improve the loss rate considerably by routing through Utah. The upper left figure shows the `MIT` to `Utah` loss rate, and the lower right shows the `Utah` to `ArosNet` loss rate.

lem affecting one or more backbone service providers [6].

Many of the restrictions of peering can be overcome. An organization that has Internet service in multiple ASes can run an application that is distributed across nodes located in the ASses, and use a RON to provide routing between these nodes. By explicitly constraining the size of any given RON to be small (under, say, 50 nodes), the aggressive exploration of alternate paths and performance-based path selection can be accomplished. Thus, RON's routing and path selection schemes emphasize failure detection and recovery over scalability, improving both reliability and performance of the RON application.

To obtain a preliminary understanding of the benefits of using RON, we evaluated the effects of indirect RON-based packet forwarding between the four sites mentioned in our examples: The University of Utah, MIT, ArosNet, and a MediaOne cable modem in Cambridge, MA. The interconnections between these nodes are as shown in Figure 2. In this topology, RON is able to provide both reliability and performance benefits for some of the communicating pairs.

2.1 Reliability

Figure 3 shows the 30-minute average packet loss rates between MIT and ArosNet. In these samples, the loss rate between MIT and ArosNet ranged up to 30%, but RON was able to correct this loss rate to well below 10% by routing data through Utah (and occasionally through the cable modem site). This shows that situations of non-transitive Internet routing do occur in practice, and can be leveraged by a RON to improve the reliability of end-to-end application communication.

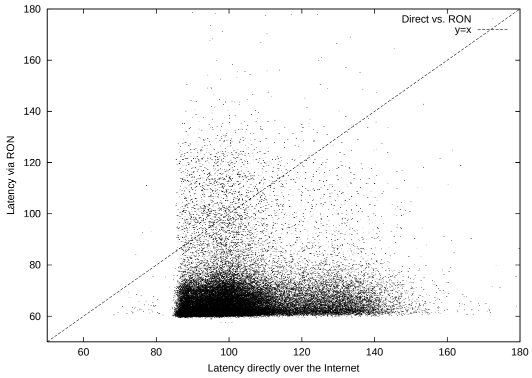

Figure 4: RON vs. direct samples. The samples are temporally correlated; the latency via RON is plotted on the Y axis, and the latency via the Internet is on the X axis. 0.5% of the outlying samples (215 / 51395) are not shown for readability. The dataset represents 62 hours of probes taken roughly 4 seconds apart.

2.2 Performance

We took measurements between the four sites using `tcping`, a TCP-based ping utility that we created. We sent one `tcping` flow over the direct Internet and another through the lowest-latency indirect path as estimated by the results of recent `tcping` probes. If the direct IP path had lower latency that the best indirect path, then the direct one was used since that is what RON would do as well.

Figure 4 shows the latency results between MIT and ArosNet, gathered over 62 hours between January 8 and January 11 2001. In 92% of the samples, the latency of the packets sent over a RON-like path was better than the Internet latency. The average latency over the measurement period decreased from 97ms to 68ms; indirect hops through both MediaOne and Utah were used, and some packets were sent directly. The benefit in this case arose partly from using the high-speed Internet2 connection, but more from avoiding the exchange between MediaOne and Qwest, which frequently went through Seattle!

2.3 Case Summary

These observations argue for a framework that allows small numbers of nodes to form an overlay that can take advantage of these improved paths. By pushing control towards the endpoints, or even directly to the application, the RON architecture achieves four significant advantages. (1) More efficient end-system detection and correction of faults in the underlying routes, even when the underlying network layer incorrectly thinks all is well. (2) Better reliability for applications, since each RON can have an independent, application-specific definition of what constitutes a fault. (3) Better performance, since a RON's limited size allows it to use more

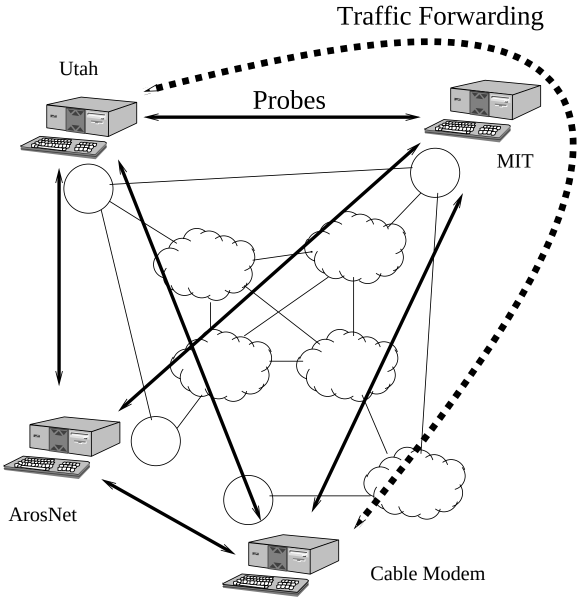

Figure 5: The general approach used in the RON system. Nodes send probes to determine the network characteristics between each other. Using their knowledge of the network, they potentially route traffic through other nodes. In this example, traffic from Utah to the Cable Modem site is sent indirectly via MIT.

aggressive path computation algorithms than the Internet. (4) Application-specific path selection, since RON applications can define their own routing metrics.

3 Approach

The RON approach is conceptually simple. Figure 5 outlines this approach: The RON software sends *probes* between RON nodes to determine the network characteristics between them. Application-layer *RON routers* share this information with the other RON nodes, and decide on next hops for packets. When appropriate, the traffic between two RON nodes is sent indirectly through other RON nodes, instead of going directly over the Internet.

We designed the RON software as libraries, usable by unprivileged user programs. The components of the RON software provide the mechanisms necessary for application-layer indirect routing. RON needs (1) methods to *measure* the properties of the paths between nodes and *aggregate* this information; (2) an algorithm to *route* based on this information; and (3) a mechanism to *send data* via the overlay. We describe each of these components below.

3.1 Monitoring Path Quality

RON nodes measure path quality using a combination of active probing by sending packets across virtual links, and passive measurement of the results achieved by data transfers over the virtual links. Because our goal is to provide better service than the default paths, we must measure links that may not be in use by data transmissions, necessitating the use of active probes. Passive measurements, however, can provide more information with less bandwidth cost by using traffic that must already flow over the network. This is why we use both forms of monitoring.

Measurements may either be *system-defined*, e.g., “the latency between two hosts,” or they may be *application-defined*, e.g., “the time to download this object from a mirror site,” similar to the approach taken in SPAND [12]. The designers of an overlay network cannot be omniscient about the desires and metrics that are important to future users; a well-designed system must provide both a rich set of system-defined metrics for ease of use, and the ability to import and route based on application-defined metrics to accommodate unforeseen applications.

It is impractical to send days of detailed performance history to all other participants in the network so that they can decide on the best path over which to transfer data. Furthermore, a reliable system must handle participating nodes that crash, reboot, or rejoin the RON. Measurement data, particularly network probe data, is often extremely noisy and must be smoothed before it is of use to clients. The RON system must therefore have a mechanism for *summarizing* the performance data it collects, before transmitting it across wide-area network paths to other RON nodes.

Hosts on the same LAN will frequently experience similar network conditions when communicating with other hosts. To reduce the impact of network probe traffic and increase the base of information available to the routing system, hosts on the same LAN should be able to share information about the performance of remote paths and sites. From these requirements, we conclude that the RON system should support a shared *performance database* that local hosts can use to share and aggregate performance data. To avoid introducing more points of failure into the system, both the performance database and its clients must treat the data stored in it as *soft state*. Clients must not fail if information they want is not in the database. The correct functioning of the database must not depend on the presence of information about particular clients or remote networks.

3.2 Routing and Forwarding

Indirect hops through the network require additional bandwidth, time, and computation. We believe that we can achieve the major benefits of an overlay using only a few indirect hops. Our design currently calls for computing paths only with single indirect hops. To send packets indirectly, the RON architecture should use UDP, not IP or some new protocol, to permit implementation as an unprivileged process. The small size of each RON allows us to exchange topology and performance information using a link-state routing protocol.

Intermediate forwarding nodes should not require application-specific knowledge about the packets they handle. We take the idea of *flow labels* from IPv6 [7] and MPLS [4]: The RON endpoints should tag their flows with an appropriate routing hint ("Minimize latency") and with a flow identifier, permitting downstream routers to handle the packets without needing to understand the protocols contained in the encapsulated packets. For instance, a video conferencing application may send its audio and video data as logically separate streams of data, but may want them to be routed along the same path to keep them synchronized. By pushing flow labeling as close to the application as possible, these decisions can be made at the right place. Early flow labeling also reduces the load on the intermediate nodes, by simplifying their routing lookups.

3.3 Sending Data

The basic RON data transmission API is simple: The conduit that provides the input and output for the RON must provide a function to call when there is data to be delivered, and must either notify the RON forwarder explicitly *or* provide a `select`-like mechanism for notifying the forwarder when data is available for insertion into the RON. Both of these alternatives are appropriate for use in the libraries implementing the RON functionality; the needs of the application should determine which is used.

3.4 Applications and Extensions

The components of RON described thus far are necessary for a basic user-level packet forwarding system, but applications that integrate more tightly with the routing and forwarding decisions are capable of more complex behavior. We discuss a few usage scenarios below, considering how they interact with the base RON functionality.

RONs can be deployed on a per-application basis, but they may also be deployed at a border router. There, they can be used to link entire networks with Overlay Virtual Private Networks. An Overlay ISP might even buy bandwidth from a number of conventional ISPs, paying them according to a Service-Level Agreement, and selling "value-added" robust networking services to its own customers.

When used to encapsulate network-level traffic, RONs can be combined with Network Address Translation (NAT) to permit the tunneling of traffic from remote sites not enabled with overlay functionality. For example, consider the network from Figure 2. A RON node located in the EECS department at MIT could be used by the other sites to proxy HTTP requests to `www.mit.edu`, accelerating Web browsing for off-site collaborators. Traffic would flow through the overlay to the MIT RON node, from which an HTTP request would be sent to the Web server. The HTTP response would be sent to the MIT RON node, and from there, relayed to the requesting host over the overlay.

Another use of RONs is to implement multi-path forwarding of flows. TCP performs poorly when subject to the large jitter and packet reordering that is often imposed by splitting one flow between multiple paths, but sending *different* TCP flows between the same two hosts (or two networks) poses few problems. The flow labeling component of a RON provides the first handle necessary to achieve this goal, and a routing component that performs flow assignment would provide the other part.

When a cooperating RON system either controls the majority of the available bandwidth on its links, or is given quality of service (QoS) guarantees on individual links of the network within a single ISP, it may be possible to then use the overlay network to provide global QoS guarantees to individual flows that traverse the overlay[1].

3.5 Routing Policies and Deployment

As with any overlay or tunneling technique, RONs create the possibility of misuse, violation of Acceptable Use Policies (AUPs), or violation of BGP transit policies. At the same time, RONs also provide more flexible routing that can *enhance* the ability of organizations to implement sophisticated *policy routing*, which is the ability to make routing decisions based upon the *source* or *type* of traffic, not just its destination address. This is an old idea [2], but its use in backbone routers have been scarce because of the increased CPU load it frequently imposes.

RONs interact with network policies in two ways. Because RONs are deployed only between small groups of cooperating entities who have already purchased the Internet bandwidth they use, they cannot be used to find "back-doors" into networks without the permission of an authorized user of those networks. The upholding of an organization's AUP is primarily due to cooperation of its employees, and this remains unchanged with the deployment of RONs.

More importantly, the smaller nature of RONs running atop powerful desktop computers can be used to implement policy routing on a per-application basis. One of our goals is the creation of a policy-routing aware forwarder with which administrators can easily implement policies that dictate. For instance, one policy is that only RON traffic from a particular research group may be tunneled over Internet2; traffic from the commercial POPs must traverse the commercial Internet.

3.6 Status

We have implemented a basic RON system to demonstrate the feasibility of our end-host based approach and are continuing to refine our design and implementation. We are deploying our prototype at a few nodes across the Internet and

[1] This possibility was suggested by Ion Stoica.

are measuring outages, loss rates, latency, and throughput to quantify the benefits of RON. We have built one RON application, an IP forwarder that interconnects with other such clients to provide an Overlay ISP service.

4 Related Work

The Detour study made several observations of suboptimal Internet routing [11]. Their study of traceroute-based measurements and post-analysis of Paxson's [8, 9] data shows that alternate paths may have superior latency or loss rates. These studies used traceroutes scheduled from a central server, which may undercount network outages when the scheduler is disconnected. Our research builds on their analysis by elucidating an approach for an architecture to exploit these properties. The Detour framework [3] is an in-kernel packet encapsulation and routing architecture designed to support alternate-hop IP packet routing for improved performance. In contrast, RON advocates tighter integration of the application and the overlay, which permits "pure application" overlays and allows the use of application-defined quality metrics and routing decisions. Furthermore, the main objective of RON is reliability, not performance.

Content Delivery Networks (CDNs) use overlay techniques and caching to improve the performance of specific applications, such as HTTP and streaming video. The functionality provided by the RON libraries may ease the development of future CDNs by providing some basic routing components.

The X-Bone is designed to speed the deployment of IP-based overlay networks [13]. It provides a GUI for automated configuration of IP addresses and DNS names, simple overlay routing configurations, and remote maintenance of the overlays via secure HTTP. The X-Bone does not yet support fault-tolerant operation or metric-based route optimization. Its management functions are complementary to our work.

5 Summary and Research Agenda

This paper made the case for developing reliable distributed Internet services and applications using Resilient Overlay Networks (RONs), an application-level routing and packet forwarding system. A RON improves the end-to-end reliability of Internet communication by taking advantage of alternate paths and enabling application-controlled path selection in a way that traditional BGP-based Internet routing cannot.

While measurements collected by us and others suggest that RONs might work well in practice, several key research questions need to be addressed. Some of these are:

1. *How many intermediate hops?* We hypothesize that, in practice, it is sufficient to consider paths that include at most one intermediate RON node to obtain the benefits of improved reliability and performance. If this is true, it will simplify RON's path selection mechanisms and allow the implementation of a variety of application-controlled metrics.

2. *How do we choose routes?* Route selection involves summarizing link metrics, combining them into a path metric, and applying hysteresis to come up with an estimate of the route quality. How do we best perform these actions for different link metrics? How do we filter out bad measurements, and perform good predictions? How do we combine link metrics (such as loss and latency) to meet application needs?

3. *How frequently do we probe?* The frequency of probing trades off responsiveness and bandwidth consumption. The speed with which failed routes can be detected will determine how well RONs will improve end-to-end reliability.

4. *What routing policies can RON express?* RONs may allow more expressive routing policies than current approaches, in part because of their application-specific architecture.

5. *How do RONs interact?* What happens if RONs become wildly popular in the Internet? How do independent RONs sharing network links interact with one another and would the resulting network be stable? Understanding these interactions is a long-term goal of our future research.

References

[1] Chandra, B., Dahlin, M., Gao, L., and Nayate, A. End-to-end WAN Service Availability. In *Proc. 3rd USITS* (San Francisco, CA, 2001), pp. 97–108.

[2] Clark, D. *Policy Routing in Internet Protocols.* Internet Engineering Task Force, May 1989. RFC 1102.

[3] Collins, A. The Detour Framework for Packet Rerouting. Master's thesis, University of Washington, Oct. 1998.

[4] Davie, B., and Rekhter, Y. *MPLS: Technology and Applications.* Academic Press, San Diego, CA, 2000.

[5] Labovitz, C., Ahuja, A., Bose, A., and Jahanian, F. Delayed internet routing convergence. In *Proc. ACM SIGCOMM '00* (Stockholm, Sweden, 2000), pp. 175–187.

[6] The North American Network Operators' Group (NANOG) mailing list archive. `http://www.cctec.com/maillists/nanog/index.html`, Nov. 1999.

[7] Partridge, C. *Using the Flow Label Field in IPv6.* Internet Engineering Task Force, 1995. RFC 1809.

[8] Paxson, V. End-to-End Routing Behavior in the Internet. In *Proc. ACM SIGCOMM '96* (Stanford, CA, Aug. 1996).

[9] Paxson, V. End-to-End Internet Packet Dynamics. In *Proc. ACM SIGCOMM '97* (Cannes, France, Sept. 1997).

[10] Rekhter, Y., and Li, T. *A Border Gateway Protocol 4 (BGP-4).* Internet Engineering Task Force, 1995. RFC 1771.

[11] Savage, S., Collins, A., Hoffman, E., Snell, J., and Anderson, T. The end-to-end effects of Internet path selection. In *Proc. ACM SIGCOMM '99* (1999), pp. 289–299.

[12] Seshan, S., Stemm, M., and Katz, R. H. SPAND: Shared Passive Network Performance Discovery. In *Proc. 1st USITS* (Monterey, CA, December 1997).

[13] Touch, J., and Hotz, S. The X-Bone. In *Proc. Third Global Internet Mini-Conference in conjunction with Globecom '98* (Sydney, Australia, Nov. 1998).