DipZoom: The Internet Measurements Marketplace
(Position Paper)

Michael Rabinovich, Sipat Triukose, Zhihua Wen, and Limin Wang

EECS Department
Case Western Reserve University

Abstract. We describe DipZoom (for “Deep Internet Performance Zoom”), an approach to provide focused, on-demand Internet measurements. Unlike existing approaches that face a difficult challenge of building a measurement platform with sufficiently diverse measurements and measuring hosts, DipZoom offers a matchmaking service instead, which uses P2P concepts to bring together experimenters in need of measurements with external measurement providers. It then harnesses market forces to orchestrate the supply and demand sides in the resulting open eco-system. This paper outlines the overall design of DipZoom, and discusses payment, trust and security issues in the resulting open system.

I. Introduction

Internet measurements drive improvements in Internet infrastructure and provide the foundation for the Internet performance research. In particular, a need often arizes for a *focused, on-demand* measurement of a certain feature, where “focused” may refer to measurement targets, measuring hosts, or the measurement regime. For example, a company may need to know the performance of its Web site from client’s perspective, or be interested in the performance of a video stream in a particular format being downloaded by a client on a particular platform from a particular region, or the effective bandwidth, or loss rate, or a round-trip packet delay on a path from a particular region to the company site. While some research platforms (such as Scriptroute [29]) and, commercially, Keynote Systems [19] offer on-demand measurements, they can provide only a limited number of measuring hosts, and hence a limited perspective on the Internet performance. Indeed, Scriptroute can provide measurements mostly from the PlanetLab nodes, which are not representative of clients’ typical connectivity [2]. Keynote attempts to select carefully the location and connectivity of their measuring hosts to reflect typical connectivity of clients. Yet even it only has presence in 50 cities worldwide, with only one location in Africa, Central and South America, and China, and none in Russia or Middle East.

It is clearly difficult for a single company to deploy measurement infrastructure to satisfy any conceivable measurement needs that may arise. Worse, having a limited number of well-known measuring hosts allows companies being measured to “game the system”, by optimizing specifically for those measuring hosts. In fact, it has been widely rumored that content delivery networks (CDNs) deploy their infrastructure in the vicinity to Keynote’s measuring hosts and recognize and provide special treatment to requests from Keynote hosts. Finally, the closed proprietary system for Internet measurements limits user choice of the types of measurements. For example, in HTTP, a measurement of a page download done with and without pipelining and persistent connections, and with different caching settings at the client, may yield very different results. One cannot rely on the prowess of a single company to provide the entire range of various modes of client operations. It should be the prerogative of the experimenter to decide on the type, the capacities, the location, and the mode of operation of clients whose perspective is of particular interest to the experimenter.

Given the scale of the Internet and the unpredictability of the needs for particular measurements, it is obvious that these limitations can only be overcome by harnessing the capacity of the Internet users themselves. Recent efforts, notably the DIMES and Traceroute@home projects (see [7], [32] and papers listed therein) recruit end-users to conduct measurements for a specific experiment administered by a central location. The system we have designed and are implementing, named DipZoom (for “Deep Internet Performance Zoom”), aims to harness the end-users for unpredictable on-demand measurements administered by an arbitrary participant.

DipZoom is based on two key ideas. First, rather than trying to build a global-scale measuring platform, DipZoom implements a *matchmaking service* that merely provides “plumbing” to connect measurement providers and requesters. This approach is analogous to file-sharing peer-to-peer networks, and we utilize some P2P techniques in our project. Second, DipZoom uses a market approach with real money as a regulator of system behaviors without imposing centrally designed rules that might jeopardize the open spirit of DipZoom and turn off potential participants. In addition to providing incentive to participants and encouraging innovation (at some future point, we can even

This work is supported by the NSF grant CNS-0520105.

envision software vendors marketing DipZoom-compliant implementations of new measuring software assuming an appropriate attestation process can be developed), the market approach effectively addresses the frivolous usage issue and, as described later, helps prevent denial of service attacks from the measuring hosts.

Together, these two ideas would create a *marketplace* for Internet measurements: an open ecosystem, where anyone can offer measurements from their computers and other computing devices, and anyone can request measurements. Participants can set their prices, compete for requests, bid and solicit bids, etc., akin to a specialized eBay. Our hope is that the combination of an open system with market forces will lead to innovation and diversity in the offered measurements and measuring devices. Past experience, such as SETI@home (see www.seti.org), shows there are usually enough adopters of new Internet endeavors even without any reward. As the success of the UPromise service (see www.upromise.com) shows, a possibility of even a tiny reward is a strong motivator for a typical user. As the system grows, its usefulness as a peer-to-peer measuring infrastructure increases, which would encourage wider participation for a snowball effect.

This paper outlines our work-in-progress on DipZoom and discusses some challenges that arise in an open system of this type.

II. The System Overview

A conceptual architecture of the system is shown in Figure 1. The DipZoom ecosystem consists of measurement providers who install DipZoom measurement software on measuring hosts (referred below as measuring points, or MPs) and make these hosts available for measurements; the measurement requesters who request measurements form the measuring points; and the DipZoom Core, which matches measurement requesters with providers, processes payments, and enforces security and trust mechanisms.

The basic DipZoom protocol works as follows (we explain the rationale for the protocol design in Sections III and IV). An MP first downloads the desired measurement software. The downloaded software instance shares a uniquely generated secret key with the core, which will be used to encrypt future interactions between it and the core. The MP then registers with the core to specify its capabilities and characteristics (the kinds of supported measurements, the characteristics of the platform, the pricing, the maximum rate of measurements the MP is willing to perform). As part of the registration, the core assigns a unique ID to the MP. Subsequently, the MP announces to the core every time it comes on-line, to enable matching of requesters with currently operational MPs. A measurement requester sends its *service query* to the core, specifying the kind of measurement, and the number and characteristics (network connectivity, locale, etc.), of the MPs desired, and receives back a list of operational MPs and their pricing. After depositing the payment with the core, for each MP on the list, the requester receives a measurement *ticket* encrypted with the secret key of the MP's measurement software. The ticket includes the measurement request and the credential in the form of <*MP-ID, nonce*> (A "nonce" is a number guaranteed to be different every time it is generated by a host, and is a standard component in secure protocols.) The requester then directly communicates with the MPs using the obtained credentials. In response to a measurement request, each MP returns an encrypted message containing measurement results and the credential as received from the requester (to prevent the MP from replaying an old result for the current request). The requester then goes back to the core to decrypt the results and finalize the payment.

Although we are initially focusing on active measurements, the above protocol lends itself to passive measurements also, if the measuring software installed on the MP includes appropriate instrumentation. In this case, the requester would purchase a time interval for collecting passive measurements, and the provider would return the results once the collection is completed.

The following subsections provide further details on the DipZoom architecture and the issues involved.

A. Service Discovery and Matchmaking

DipZoom brings together loosely coupled players: a requester may obtain some measurements from a provider, and never contact it again. Providers come on-line and leave the platform frequently, and also change their offerings and prices.

To address this style of occasional interactions, we have designed DipZoom around the concept of Web services. We envision the measuring software implemented as a stripped down application server, running Web services that correspond to different types of measurements. DipZoom service discovery is based on UDDI [33] and WSDL [37] standards for Web services, and the interaction between measurement requesters and providers uses SOAP [28]. When a measuring point registers its services with DipZoom, it uses the UDDI format to describe its capabilities, characteristics, and pricing. It also describes the format of its SOAP methods using the WSDL language. When a requester asks DipZoom for a set of MPs, it specifies the service query as a UDDI query, and uses the returned WSDL signatures to construct proper SOAP requests for the MPs.

When there are more qualifying MPs for a given service query than the requester asked for, the DipZoom core can choose the MPs for the request among all the potential matches. DipZoom uses this discretion to drive its *calibration* and *ranking* of MPs (see Section II-C).

B. Payment Mechanism

With its goal to create a playground where anyone can come and propose their own games, DipZoom needs to provide flexible pricing mechanisms, including free service,

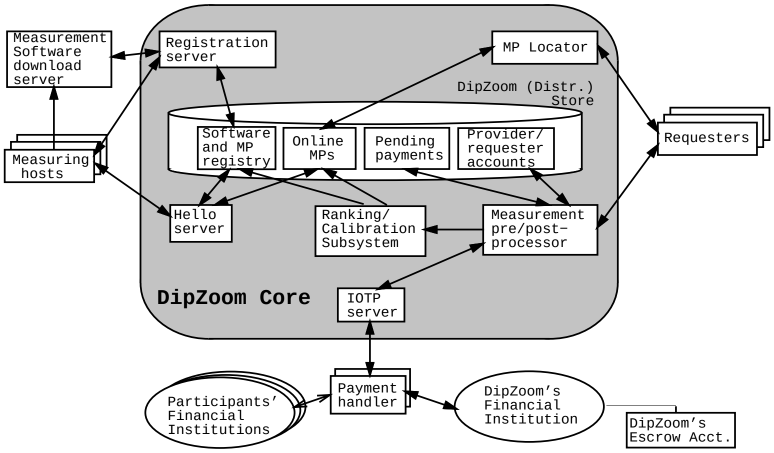

Fig. 1. A conceptual architecture of DipZoom

fixed prices, requester-centric auctions (with providers bidding for *requests for purchases*) posted by requesters, and provider-centric auctions (with more traditional requester biddings). Anticipating low per-measurement costs, DipZoom supports micropayments (e.g., [12]) by batching them in its internal provider and requester accounts. For actual transactions, DipZoom will interface with a real external payment system such as the one by Verisign [34], using the IETF's Internet Open Trading Protocol [14] to isolate payment handler specifics from the rest of DipZoom.

C. Calibration of MPs

As an open system, DipZoom raises many security and trust issues. We will later discuss how we plan to raise the bar for an attacker and cheater. However, without special hardware and operating system support, one cannot guarantee protection against a devoted outlaw. While Microsoft and Intel are working to provide this support [9], we are using selective MP *calibration* to deal with this challenge.

First, we will identify outlier MPs based on the deviation of the MP's measurement results from similar measurements reported by other MPs with similar characteristics. Note that despite its P2P nature, DipZoom has all the input data outlier detection since it decrypts all the measurement results for the requesters (see Section IV-B). Second, we plan to use *calibration*, to verify the legitimacy of outlier MPs as follows. We will deploy (or ask our colleagues to deploy) passive packet monitors (such as a *tcpdump* utility on low-volume sites, or a monitor appliance such as Gigascope [6] on high-volume sites) on some undisclosed sites; a script in the DipZoom core would then purchase measurements of these sites from outlier MPs, and compare their reported measurements with the measurement results obtained from the packet monitors. Note that unlike Keynote nodes, the calibration nodes are not disclosed as part of the system. This calibration process would allow DipZoom to distinguish and blacklist dishonest MPs. Obviously, the scale of the calibration is determined by the capacity of DipZoom core, and not all cheats will be identified. However, similar to tax evasion, the possibility of being caught should acts as a deterrent. In the future, DipZoom can augment MP calibration with explicit feedback from the requesters, implementing a full-blown reputation system.

III. Security

There are significant security issues an open ecosystem like DipZoom must address, some of which are outlined below.

Induced denial of service attack. In principle, one can use DipZoom to launch a reflection DoS attack against a site, by requesting many measuring points to perform a large number of measurements of the target site. Although the payment system would limit these attacks, DipZoom protects against an attacker who is willing to incur the expense by rate-limiting possible measurements. Prior work has shown high effectiveness of rate-limiting in protecting an open system [36]. Because the attacker may request measurements under different disguises, DipZoom applies rate limiting with respect to both targets and requesters.

Open ports on MPs. Opening a port on computers that run a measuring point for incoming requests can only increase the vulnerability of the computer, both by having a potential opening for breaking into the computer and by creating a target for a low-level DoS attack, such as a SYN flood [3]. Existing P2P networks have demonstrated that this vulnerability can be effectively controlled. Still, in the future, we would like to investigate a direction where MPs form an overlay network, and the requests are delivered to the target MP over the overlay. This would improve the overall resiliency of the system because each potential requester would have to supply a valid source IP address when joining the overlay, and each node in the overlay network only accepts packets from (a limited number of) neighbors.

High-Level DoS attack against a measuring point. An attacker can attempt to attack a DipZoom MP by excessive well-formed measurement requests sent from valid IP addresses. DipZoom provides two protective measures. First, it allows the measurement provider to specify the measurement rate that can be requested from its measuring point, and the core will enforce this limit when selecting MPs for any service query. An attempt to circumvent the limit by replaying the same request can be prevented by an MP by caching recently seen nonces and discarding most replayed requests. Second, the DipZoom protocol requires the requester to guarantee payment before sending the request to the MP, by depositing real money into an escrow account. Even if the attacker does not decrypt the results (and so the payment is not finalized), (s)he still must deposit the amount commensurate with the scale of the attack, and the attacker's identity is still known to the system, which will be a deterrent for malicious behavior.

Random nonce DoS attack. The inclusion of the MP-ID into a credential prevents another type of a DoS attack, where the attacker submits a request with random bits for an encrypted nonce. Without MP-ID, the measuring point would not recognize the decrypted random nonce as invalid and would perform the measurement, consuming its own and the target's resources. With MP-ID, the MP will discard a request unless its decrypted credential includes the valid MP-ID. Constructing such a credential is hard for the attacker without knowing the MP's security key, even though the MP-ID itself might not be a secret.

Measurement side effects. Some measurements may leave undersirable side effects and should not be invoked on some targets. For example, some devices used in home networks now offer management interface via the HTTP protocol, so that a configuration action is performed by submitting an HTTP request with an appropriate URL. If they do not run HTTP over secure sockets, and a measuring point in this network offers measurements of a page download, an attacker could reconfigure the device by requesting the MP to download the configuration URL. To prevent this, for certain measurement requests, the DipZoom core will attempt to perform a measurement itself once[1]. As long as DipZoom can perform the measurement from outside the MP's network, letting the MP do the same does not increase the vulnerability.

IV. Trust

Trust issues span several areas in Dipzoom, including system/software trust, payment trust and measurement trust.

A. Software Trust

To definitively address trust issues, DipZoom must be able to (1) assure that an MP executes genuine DipZoom measuring software, (2) trust certain system calls that the measuring software invokes, and (3) securely store MP-ID on the measuring point. Microsoft and Intel are working on an open trusted computer that would support the above, by offering a reliable *program attestation*, including a secure kernel, and a reliable and secure persistent *sealing* of a piece of information on a remote computer [9]. In the meantime, DipZoom only raises the bar for the cheater, an approach similar to what is currently used by the electronic media industry for digital rights management. Further, DipZoom then monitors and blacklists the cheaters as described in Section II-C.

Our basic assumption is that a malicious MP cannot extract the unique secret key from the measuring software, which DipZoom embeds into each downloaded instance of the software. Thus, as long as the MP can properly decrypt or encrypt information with this key, we assume that the MP executes the genuine unaltered software.

B. Payment Trust

A standard concern with payment systems is to provide assurance to the seller that the buyer will pay for the goods once the seller ships them, and to the buyer that the seller will ship the goods once the buyer pays for them. Consequently, the requester of the measurements must deposit the payment to DipZoom's escrow account before contacting the MPs that would perform the measurements, the MPs return the encrypted measurement results to the requesters, which are useless unless decrypted by the core, and the core transfers the payment from the escrow account to the MPs when decrypting the results.

However, the above solution is still vulnerable to replay attacks. A requester may attempt to replay an old request to avoid a payment. An MP may attempt to replay a response to be paid without performing a measurement. DipZoom uses the requester credentials to prevent these attacks. Indeed, if the requester replayed a request, any attempt to decrypt the response at the core will let the core recognize the offense (from the unexpected nonce in the response). A replayed response will be similarly recognized by the core at the decryption time. (Recall that we assume that the malicious MP cannot extract the secret key from the measuring software to decrypt the nonce.)

C. Measurement Trust

How can a requester trust a measurement provided by the measuring point? Potential measurement trust issues include the following:

Fake MP registrations. An MP may want to lie to the DipZoom core about its service or capabilities, e.g., in an attempt to attract more requests. To counter this, all registration messages are encrypted by the secret key embedded in the MP's measurement software instance. The software obtains the MP characteristics being registered by executing appropriate system calls, and also adds a nonce into the registration message to guard against message replay. (Note that until Microsoft and Intel's vision for a trusted computer is realized, the malicious MP could

[1] Assuming it is within the target's rate limits. Otherwise the service query will be denied anyway.

in principle reimplement the system calls that report its capabilities, so that they would return false results.)

MP multiplexing and impersonation. An attacker may try to operate multiple MP instances on the same computer, in order to attract more requests or skew result averages. When Microsoft and Intel's vision for trusted computers is implemented, DipZoom can address this issue definitively by generating a unique MP-ID during the registration and *sealing* this ID on the MP's computer, which would allow DipZoom to reliably associate each MP with its computer. In the meantime, DipZoom enforces that only a single MP with a given MP-ID can be recognized as being on-line. Thus, a measurement provider can copy the downloaded MP software to another computer or create multiple MP instances on the same computer, but only one of them (the last to announce its on-line operability to the core) will be recognized by the core. We will require MPs to include its characteristics with every announcement to guard against MP impersonation.

Fake measurements. To sell more measurements than its capacity, an MP may try to return fake results without actually performing the measurements. DipZoom's credentials mechanism guard against this behavior, as well as against a third party attacker who may attempt to replay an intercepted old response as a substitute of the legitimate response.

V. Firewalls and NATs

How can DipZoom measurement requests reach a measuring point behind a firewall or a network address translation (NAT) device? There are two main types of firewall configurations: transparent and explicit. Transparent firewalls pass through connections with outside hosts that are initiated from behind the firewall. Similar to existing P2P networks, DipZoom will rely on outside proxies to serve as conduits for measurement requests. The firewalled MP initiates and maintains a connection to an outside proxy. Requesters send their requests to the MP through this proxy, which conveys these requests to the MP over the connection that the MP initiated. In particular, following the approach of Skype P2P network [27], proxy functionality can be made part of the measuring software, and non-firewalled MPs may double as proxies. In fact, because of the relatively small size of the measurement results, it is feasible for MPs to send responses via proxies as well, thus allowing both MPs and requesters to be firewalled.

Explicit firewalls, used by some large corporations, block any direct traffic between the internal hosts and the Internet; they require all outside communication to occur over application proxies deployed in a so-called DMZ area of the network. The only way for a host behind an explicit firewall to take part in DipZoom is for the corporation to deploy a DipZoom proxy in its DMZ area.

Proxies also allow the traversal of NAT devices. During registration, an MP establishes a persistent DipZoom ID. When reconnecting later, the MP connects to a proxy, which maintains the mapping from the MP's ID to its NAT-assigned external IP address and port, and the NAT box maintains the mapping between these and MP's private address and port numbers. The measurement requests from the requester will reach the MP via the proxy as in the case of the firewall.

VI. Related Work

Existing measurement platforms fall into two categories. *A-priori* platforms, such as NIMI [24], IDMaps [11], Surveyor [18], skitter [4], AMP [1], network weather service [38], and M-Coop [30], collect measurements irrespective of any particular requests, and then answer specific measurement requests by estimating the requested values from the collected generic data. *On-demand* platforms and tools, including Scriptroute [29] and Keynote [19], as well as the King tool [13], perform measurements for a given request. Unlike these systems, DipZoom leverages Internet users at large, facilitating diversity in measurements and measuring points available. As a simple illustration of limitations of existing means for on-demand measurements in today's heterogeneous Internet environments, Table I compares latency between a laptop using a Verizon's Broadband Wireless Internet service [35] and a host on Case's wired network, as measured directly by a ping command and by using King, a recent measurement tool that leverages hosts's DNS servers for more precise latency measurements [13]. The table shows that King measurements, while well-suited for wired networks, are inadequate for some emerging network environments.

TABLE I

Direct ping latency (average of 1000 pings in each direction) vs. latency measured by King (100 queries in each direction), in msec.

Ave Ping RTT	King (VZW to Case) (min-max)	King (Case to VZW) (min-max)
280	16-17	17-18

Recent efforts (see [7], [32] and papers listed therein) recruit end-users for measurements but conduct a predetermined experiment from a central location and have no incentive mechanism; these systems are best-suited for a-priori measurements while DipZoom attempts to satisfy on-demand measurement needs. The commercial performance monitoring service offered by Gomez [26] comes perhaps the closest to DipZoom, in that Gomez also offers incentives to Internet users at large to become measuring hosts. The key difference with DipZoom is that Gomez is still a closed system, which itself determines its business model, sets prices and obtains measurements from the hosts selected on behalf of the measurement requesters. It also limits the measuring hosts admitted into its pool to those it considers useful for its business. DipZoom is merely a matchmaking service and as such only acts as a facilitator, a sort of "ebay" for Internet measurements. A number of companies support public trading in a variety of areas, including network capacity. DipZoom implements a marketplace for measurements themselves.

Many tools are available to measure a variety of metrics, including hop-by-hop bandwidth [15], the bottleneck bandwidth [16], [20], [5], TCP bandwidth [21], [17], latency [23], packet loss [31], [25], and aggregate performance of higher-level operations such as a web page download [10]. We plant to wrap some of these tools as measurement services that a DipZoom measuring point can use.

VII. Status

We have implemented the initial version of the system and deployed the DipZoom core. The code for both the measuring points and DipZoom clients is available for download from the project Web site [8]. We also deployed MPs on about 150 PlanetLab nodes. The built-in measurements currently include ping (for measuring round-trip delay to a measurement target), traceroute (to obtain a router path to a target), wget (to measure the download time of a specified Web page), and nslookup (to measure the delay of a DNS query, although MPs on PlanetLab nodes do not support this measurement). The current system currently supports only geographical querying for MPs based on a country, region (e.g., state in the US), and city, by utilizing the GeoIP database from MaxMind [22]. It currently does not support real money transactions. To provide NAT and firewall traversal ability at an early stage, the core currently acts as the relay point for all communication between MPs and the clients, and all parties send periodic messages to the core to maintain appropriate NAT mappings and firewall holes. We are currently working on providing a programmatic means to request measurements so that users could issue measurement requests from inside their applications rather than manually from a DipZoom client.

VIII. Conclusions

This paper describes our work-in-progress on DipZoom, a novel approach to provide focused on-demand network measurements. Recognizing that no single measurement platform can provide focused, on-demand measurements in all corners of the Internet, our approach proposes a matchmaking service instead, which brings together experimenters in need of measurements with external measurement providers. It then harnesses market forces to orchestrate the supply and demand sides in the resulting open platform. A simplified version of the system has been deployed, and software for measurement providers and requesters is available for download.

References

[1] Active measurement project. http://amp.nlanr.net/.

[2] Suman Banerjee, Timothy G. Griffin, and Marcelo Pias. The interdomain connectivity of PlanetLab nodes. In *Passive and Active Measurement Workshop*, April 2004.

[3] S. M. Bellovin. Security problems in the TCP/IP protocol suite. *Computer Communication Review*, 19(2):32–48, April 1989.

[4] CAIDA. Skitter. http://www.caida.org/tools/measurements/skitter/.

[5] Robert Carter and Mark Crovella. Dynamic server selection using bandwidth probing in wide-area networks. In *IEEE Infocom*, 1997.

[6] Chuck Cranor, Theodore Johnson, Spataschek Spataschek, and Vladislav Shkapenyuk. Gigascope: a stream database for network applications. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, San Diego, California, June 09–12, 2003*, pages 647–651, 2003.

[7] Dimes (distributed internet measurements & simulations). http://www.netdimes.org/.

[8] http://grouper.case.edu/dipzoom/.

[9] Paul England, Butler Lampson, John Manferdelli, Marcus Peinado, and Bryan Willman. A trusted open platform. *Computer*, 36(7):55–62, 2003.

[10] Free Software Foundation. GNU wget. http://www.gnu.org/software/wget/wget.html.

[11] Paul Francis, Sugih Jamin, Cheng Jin, Yixin Jin, Danny Raz, Yuval Shavitt, and Lixia Zhang. IDMaps: a global internet host distance estimation service. *IEEE/ACM Trans. Netw.*, 9(5):525–540, 2001.

[12] S. Glassman, M. Manasse, M. Abadi, and Gauthier Gauthier. The Millicent protocol for inexpensive electronic commerce. In *World Wide Web Journal: The Fourth International WWW Conference Proceedings*, pages 603–618, 1995.

[13] K.P. Gummadi, S. Saroiu, and S.D. Gribble. King: Estimating latency between arbitrary internet end hosts. In *Proceedings of the Second SIGCOMM Internet Measurement Workshop*, 2002.

[14] Internet Open Trading Protocol (IOTP), version 1.0. Request for Comments 2801.

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

[16] Manish Jain and Constantinos Dovrolis. End-to-end available bandwidth: Measurement methodology, dynamics, and relation with tcp throughput. In *Proceedings of SIGCOMM*, Pittsburgh, PA, August 2002.

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

[18] S. Kalidindi and M. J. Zekauskas. Surveyor: An infrastructure for internet performance measurements. In *INET'99*, 1999.

[19] Measurement and monitoring: Web site perspective. www.keynote.com/solutions/website_perspective.html.

[20] Kevin Lai and Mary Baker. Nettimer: A tool for measuring bottleneck link bandwidth. In *USITS*, 2001.

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

[22] Manmind geoip databases. http://www.maxmind.com/app/ip_locate.

[23] Mike Muuss. Ping. directory.fsf.org/network/misc/ping.html.

[24] Vern Paxson, Jamshid Mahdavi, Andrew Adams, and Matt Mathis. An architecture for large-scale internet measurements. *IEEE Communications*, 36(8):48–54, 1998.

[25] Stefan Savage. Sting: a TCP-based network measurement tool. In *USITS*, 1999.

[26] http://www.porivo.com/.

[27] Skype. http://www.skype.com.

[28] SOAP version 1.2 specification. http://www.w3.org/TR/soap12.

[29] Neil Spring, David Wetherall, and Tom Anderson. Scriptroute: A public internet measurement facility. In *Usenix Symp. on Internet Technologies and Systems*, 2003.

[30] Sridhar Srinivasan and Ellen W. Zegura. Network measurement as a cooperative enterprise. In *IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems*, pages 166–177. Springer-Verlag, 2002.

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

[32] Traceroute@home, Laboratoire lip6 – CNRS. http://tracerouteathome.net/.

[33] The Universal Description, Discovery and Integration (UDDI) Protocol. http://www.uddi.org/specification.html.

[34] Verisign online payment processing. www.verisign.com/products-services/payment-processing/online-payment/index.html.

[35] Verizon wireless Internet BroadbandAccess. http://www.verizonwireless.com/b2c/mobileoptions/broadband/.

[36] Limin Wang, Kyoungsoo Park, Ruoming Pang, Vivek Pai, and Larry Peterson. Reliability and Security in the CoDeeN Content Distribution Network. In *Proceedings of the USENIX 2004 Annual Technical Conference*, June 2004.

[37] Web Services Description Language (WSDL) 1.1. http://www.w3.org/TR/wsdl, 2001.

[38] Rich Wolski, Neil T. Spring, and Jim Hayes. The network weather service: a distributed resource performance forecasting service for metacomputing. *Future Generation Computer Systems*, 15(5–6):757–768, 1999.