Internet Performance Monitoring

THOMAS M. CHEN, SENIOR MEMBER, IEEE, AND LUCIA HU

Invited Paper

The growing diversity of services in the Internet is motivating research to improve measurability of traffic and Internet performance. This paper surveys current projects and tools for Internet performance monitoring, ranging from passive router-based traffic flow measurement methods to large-scale active performance monitoring projects. The tools and methods are discussed according to their protocol layer, starting from the network layer (ATM, MPLS) to IP/ICMP and transport/application layers. At each protocol layer, the strengths and limitations of the methods are highlighted. Finally, issues and challenges for future research are reviewed.

Keywords*—*Internet performance, network monitoring, traffic measurement.

I. INTRODUCTION

A public demonstration of the ARPANET at the 1972 International Conference on Computers and Communication (ICCC) featured a new electronic mail application which would become the predominant network application for the next decade. Other applications such as FTP and telnet were developed soon after while the network was gradually commercialized. In 1985, NSFNET was established by the U.S. National Science Foundation (NSF) to serve the higher education community. The NSFNET backbone was initially restricted to academic and research uses, but commercial traffic was encouraged within the regional networks. NSF encouraged the development of commercial Internet service providers, and UUNET became the first in 1987. Commercial traffic was allowed on NSFNET starting in 1991 to supplement funding for research and education uses. Around the same time, gopher, the first point-and-click network application, was being developed at the University of Minnesota. In 1993, the National Center for Supercomputing Applications (NCSA) released the first graphical Web browser, Mosaic X (although the World Wide Web was proposed earlier by Tim Berners-Lee at CERN in 1989). In 1995, NSFNET was decommissioned in favor of a number of commercially administered backbone service providers.

Today, the Internet serves a spectrum of social and commercial purposes for an enormous worldwide user population, and continues to evolve as a global infrastructure for new services such as multimedia streaming. Consequently, service providers have been increasingly motivated to gain a deeper understanding of Internet behavior through measurements of traffic and network performance. Measurement data is useful for a variety of purposes such as verification of service level agreements (SLAs), accounting and billing, resource management, traffic engineering, and network planning. However, the practical problem of measuring end-to-end Internet performance has received surprisingly little attention. Although service providers undoubtedly monitor their own networks, the competitive nature of the Internet service market has discouraged industry-wide cooperation to enable large-scale Internet performance measurements. Cooperation is necessary to ensure that any large-scale instrumentation and methods used for monitoring the Internet (which has clearly not been designed for observability) will be consistent, accurate, scalable, and safe.

The performance metric of most interest is the user throughput, which is mainly affected by the packet loss metric ratio. For real-time applications, end-to-end packet delay and to a lesser extent packet delay variation are also important performance metrics [1]. The IP Performance Metrics (IPPM) working group in the Internet Engineering Task Force (IETF) has produced a framework establishing a common terminology and addressing issues of clock accuracy, timestamping, and effects of sampling on measurements [2]. Specific metrics include one-way and two-way connectivity, one-way packet delay, round-trip delay, and one-way packet loss [3]–[6]. Under study are metrics for packet delay variation and packet loss pattern statistics and a proposed one-way active measurement protocol (similar to a one-way ping). Performance metrics are closely related to SLAs which are becoming more common among service providers. SLAs are contracts defining performance

Manuscript received April 10, 2002; revised May 21, 2002.

T. M. Chen is with the Department of Electrical Engineering, Southern Methodist University, Dallas, TX 75275 USA (e-mail: tchen@engr.smu.edu).

L. Hu is with the University of Southern California, Los Angeles, CA 90089 USA (e-mail: lucia_hu@hotmail.com).

Digital Object Identifier 10.1109/JPROC.2002.802006.

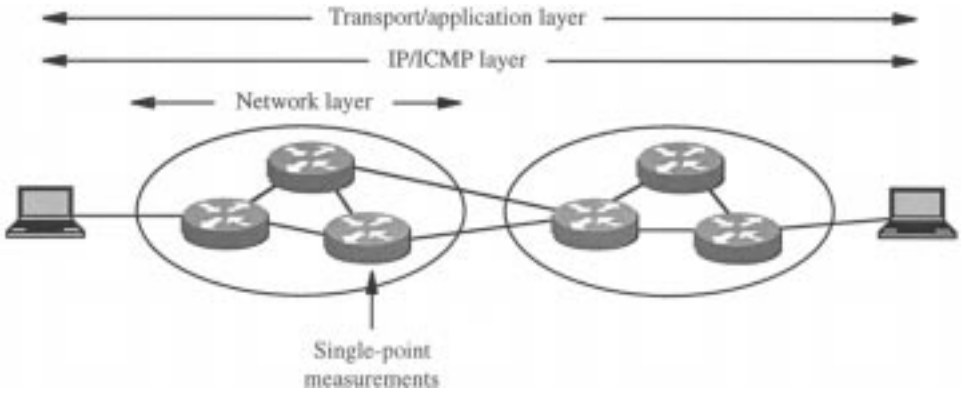

Fig. 1. Scope of measurements.

metrics (such as service availablity, latency, and throughput) and acceptable performance levels (e.g., the T3 NSFNET had a goal of 0.01% packet loss).

Performance might be interpreted by some in a broader sense to refer to all aspects of Internet behavior. For example, reliability in terms of mean time between service outages and mean duration of service outages is often associated with performance. Another complication in any discussion of performance is the close relation between network performance and traffic conditions. For example, traffic load and burstiness are obviously related to packet delay and loss. The discussion here is restricted to measurement of performance and traffic and does not address reliability, LANs, routing, topology discovery, or bandwidth estimation. These topics are closely related to Internet performance but merit detailed treatment as separate topics. Shared-medium LANs obviously present different challenges than the Internet. In LANs, packet sniffing and remote network monitoring (RMON) are well-known techniques used by network administrators to monitor LAN behavior and diagnose troubles. Packet sniffing is done by a host operating in promiscuous mode on the LAN (capturing every packet broadcast on the LAN). RMON is an extension of simple network management protocol (SNMP) to manage remote LANs by means of RMON probes which are sophisticated SNMP agents with an RMON MIB and local intelligence to perform packet filtering, packet decoding, statistics computation, problem detection, and alarm notification. However, the use of RMON by service providers has been limited due to the complexity and cost of the RMON probes. Routing is closely related to performance monitoring because dynamic routing protocols such as OSPF are designed to adapt to changing network conditions and hence directly effect traffic patterns. Routing dynamics is an important topic by itself and cannot be covered adequately within the scope here. Topology discovery is typically done by traceroute or a variation such as pathchar [7] or clink [8]. Bottleneck bandwidth estimation is commonly done by packet pair methods [9] or packet tailgating [10]. Although topology and bottleneck bandwidth have implications for network performance, this paper focuses on the direct measurement of performance.

A survey of performance monitoring methods can be approached in several different ways. For example, methods are usually classified as active (involving the addition of test traffic) or passive (no interference with normal traffic). Methods may also be classified according to whether they are carried out in the user plane (test traffic), control plane (signaling), or management plane; the number of measurement points involved; the performance metric to be measured; or the protocol layer where the method operates. We consider methods of increasing scope as shown in Fig. 1 which roughly corresponds to a bottom-up approach to the protocol layers. The scope of the method is also closely related to whether it is active or passive. Thus, the paper follows a direction from passive methods and then active methods. First, we review router-based methods which are generally passive single-point measurements of traffic flows. Next we discuss performance measurements in the network layer (below IP) and focus on asynchronous transfer mode (ATM) and multiprotocol label switching (MPLS) protocols. These label-switching protocols allow the possibility of active injection of operations and maintenance (OAM) data into the connection for in-service measurements. However, network-layer methods are limited to the scope of single subnetworks that are entirely ATM or MPLS. Moreover, it is uncertain at this point whether OAM will be implemented in MPLS. Above the network layer, the IP/ICMP layer allows measurements across the Internet regardless of the underlying network protocols. IP/ICMP methods are mostly active and derived from the venerable ping or traceroute. Above the IP/ICMP layer, measurements can be carried out at the transport or application layer which are advantageous when the performance of an application is the ultimate concern. Finally, we discuss a number of important issues for future research.

II. Challenges in Internet Monitoring

The challenges in monitoring the Internet are consequences of its size, heterogeneity, and decentralized nature [11]. It is difficult to even discover the topology (connectivity) because of its extensive geographic scope and complicated interconnections between subnetworks.

Network administrators must typically resort to ad hoc tools like traceroute or its many variations. For example, the skitter of the Cooperative Association for Internet Data Analysis (CAIDA) uses a traceroute-like technique sending out a sequence of ICMP messages with different time to live (TTL) field values which forces routers at varying distances to return ICMP error messages [12]. Because many routes may be discovered, special tools may be needed to present and visualize the topology in an understandable way.

The lack of a centralized administration makes it difficult (if not impossible) to impose a common measurement infrastructure or protocol. For example, deployment of active testing devices throughout the Internet would require a separate arrangement with each service provider. Although service providers must cooperate to enable end-to-end services, they are also competitive. To date, there has been no competitive advantage in cooperating on a common measurement infrastructure. Service providers have generally not been receptive to external studies of their network performance which are regarded as somewhat confidential.

SNMP does offer a standardized, widely used protocol for collecting local performance statistics from individual routers, e.g., packet counts per interface over 15-min intervals. It would be natural to consider extending SNMP to monitor network performance. For example, the multirouter traffic grapher (MRTG) tool is a Perl script that reads SNMP traffic count variables from multiple routers and updates graphs of traffic load on specified links in 5-min intervals [13]. Unfortunately, SNMP is not well suited for end-to-end measurements that are needed for performance metrics.

The connectionless nature of IP contributes to the difficulty of performance monitoring, because measurements for a packet following a particular route may not be relevant to another packet that could take a different route (between the same source and destination). Moreover, Internet routing protocols are designed to be dynamic and continuously adapting the selection of routes according to current network conditions. Thus, traffic patterns may be subject to regular changes (and, in the worst case, route flapping when a router might be misconfigured).

Another difficulty is the steadily increasing rate of transmission links (now OC-192 or 10 Gb/s) which can simply overwhelm routers or traffic analyzers trying to process packets. In comparison, current OC12mon monitors can handle OC-12 rates (600 Mb/s). An InMon sFlow probe attached to a switch can monitor up to 1 Gb/s traffic (more than 1.5 million packets/s). At very high traffic rates, routers or analyzers are forced to sample packets which introduces the possiblity of inaccuracies in the ultimate traffic statistics. Even at 1 Gb/s, the measurements of "raw traffic" can result in enormous amounts of data to process and store within a monitoring period.

The IP protocol allows routers to participate in performance measurements through the IP packet header options. For example, the IP timestamp option can be used to include a list of IP addresses for each router visited by the packet along with a timestamp from each router. However, this header option is not used today due to at least two serious limitations. First, the 40-byte timestamp option is limited to accommodate only up to four (router address, timestamp) pairs. Second, the timestamps would not be meaningful without precise time-of-day synchronization between routers which is not done today.

Ping and traceroute taking advantage of the ICMP protocol continue to be the most widely used tools today. Unfortunately, they are active methods. Active methods are more controlled than passive methods in the sense that test traffic can be sent to specific routes on demand, and do not depend on any special functions in the network because test traffic is forwarded like data packets. On the other hand, active methods raise three major concerns. First, the introduction of test traffic will increase the network load which can be viewed as an overhead cost for active methods. Second, test traffic can effect the measurements that they are trying to make, so the methodology must be designed carefully to minimize its impact on the measurement accuracy. Third, test traffic entering an ISP's network might be regarded as invasive by that service provider; for example, ICMP messages might be blocked, rate limited, or assigned lower priority than data packets. Hence, active methods depending on ICMP might measure performance that is signficantly worse than the actual network performance.

III. Router-Based Passive Measurements

Routers or traffic analyzers provide passive single-point measurements of traffic, often called workload measurements. Single-point measurements do not measure performance directly, but traffic characteristics are strongly correlated with performance. For example, traffic can be profiled according to its protocol composition (mixture of TCP/UDP, HTTP, SMTP, DNS, FTP, and other protocols) and statistical characteristics (average utilization, burstiness, flow durations, packet lengths). Protocol composition is important because different applications and protocols are known to exhibit different behaviors with subsequent implications on network resource utilization. As an example, TCP has congestion avoidance whereas UDP does not, meaning that TCP will back off in the event of congestion. Most bulk transfer protocols (HTTP, FTP, SMTP) run over TCP, so carriers usually observe a predominance of TCP traffic (e.g., around 95% of the traffic mix). When congestion occurs, TCP sources will respond by reducing their offered load whereas UDP sources will not, resulting in a higher ratio of UDP to TCP traffic. If the UDP offered load continues to increase, the throughput of TCP connections will be decreased. TCP must maintain a congestion window of four or more packets in order to recover from a single packet drop using the fast retransmit algorithm [14]. If the proportion of UDP traffic becomes high or the bandwidth available to TCP becomes too low for TCP connections to maintain a a reasonable transmission window, packet loss will increase dramatically (and TCP flows will be dominated by retransmission timeouts) [14], [15].

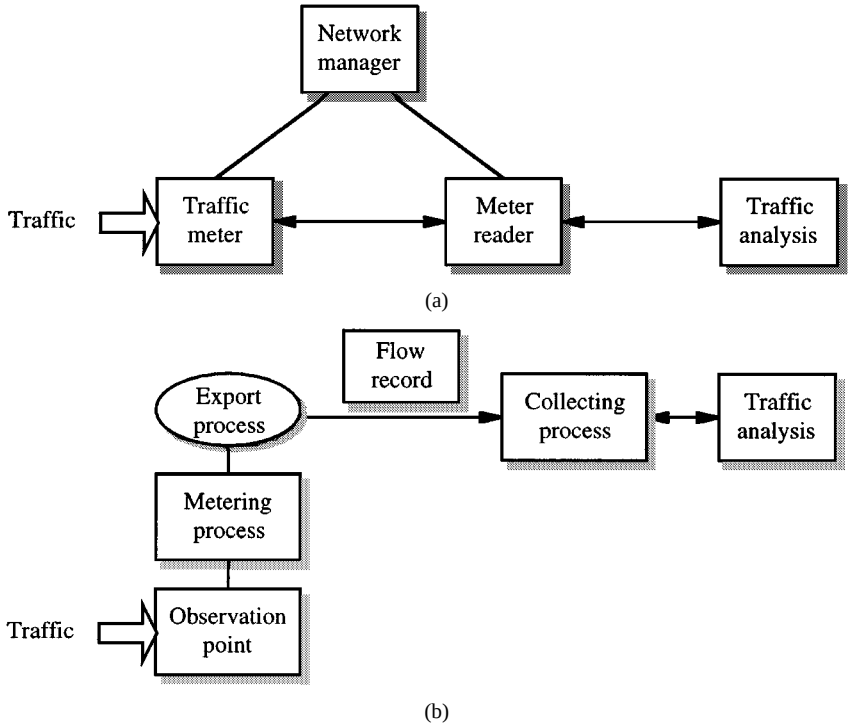

Fig. 2. (a) RTFM framework. (b) IPFIX framework.

Traffic burstiness and average utilization are also strongly correlated with the likelihood of congestion. Flow durations are important because long-lasting flows tend to have more impact on network performance. Packet sizes provide insight into the type of packet, e.g., short packets on the order of 40–44 bytes are usually TCP acknowledgment or TCP control segments (SYN, FIN, or RST); packets around 51–60 bytes are often DNS query/response packets or telnet packets containing a single character; and 552–576 byte packets correspond to default maximum segment sizes and signify IP fragmentation. In addition, large packet sizes are believed to be a factor contributing to self-similarity of traffic [16]. Self-similarity has been shown to result in longer than expected queues, implying that traditionally designed routers and switches may be more susceptible to congestion when traffic exhibits self similarity, although the tendency toward self-similarity can be greatly reduced by random early detection (RED), a widely accepted active queue management technique [17], [18]. Several studies of Internet traffic flows have been done, giving snapshots of traffic characteristics [19]–[22].

Today, routers collect limited traffic statistics which are reported to network managers through a network management protocol such as SNMP. Typical traffic statistics might be the number of received/forwarded packets, discarded packets, errored packets, port utilization, CPU utilization, and buffer utilization at each router interface accumulated over periodic intervals (e.g., 15 min). These traffic statistics might be inspected by a network manager for a quick "snapshot" to check that a router is working correctly.

Since 1997, Cisco Systems has offered a NetFlow capability in its large high-performance routers [23]. NetFlow is able to identify unidirectional traffic flows based on IP source/destination addresses, protocol field, type of service (ToS) field, source/destination port numbers, and router port. Currently, statistics can be collected about a traffic flow until the flow expires (detected by an inactivity timer). For now, flow statistics may include the flow start/stop times (first and last packets), number of bytes, number of packets, outbound interface (next hop address), and source and destination autonomous system numbers.

NetFlow essentially measures the volume and duration of each traffic flow to be analyzed off-line later for accounting, traffic engineering, application profiling, user profiling, and network planning but is not intended to be used for real-time network control. The flow data can be exported to a network management system or a FlowCollector workstation via UDP or other standard potocols. A FlowCollector workstation is specialized to receive NetFlow data from multiple routers and perform data filtering and aggregation. Multiple FlowCollector workstations can report to a NetFlow Server for data consolidation, summarization, and encryption for network transmission. NetFlow can be seen in operation in the Abilene backbone network [24]. NetFlow is close to a *de facto* industry standard and appears in routers from other vendors such as Juniper and Foundry Networks. A similar sFlow technology by InMon Corporation is found in Foundry Networks routers, but sFlow probes are also capable of conforming to the NetFlow data format [25]. Various software tools have been developed to process NetFlow data, e.g., flow-tools [26] and cflowd [27].

Recognizing the need for sophisticated traffic measurements, the IETF Real-time Traffic Flow Measurement (RTFM) working group developed a general framework for measuring statistical properties of traffic flows as shown in Fig. 2(a) [28]. The main component in the RTFM architec-

ture is a traffic meter that follows a "rule set" (set of program instructions to filter packets) to identify which packets to monitor. The current rule set is typically assigned by a network manager. The traffic meter also computes specific attributes of identified traffic flows defined in the current rule set and records the measured attributes into a flow record. Flow records are maintained in a database called a flow table. The records in the flow table can be retrieved by another RTFM component called a meter reader, possibly by FTP or a network management protocol such as SNMP. In the case of SNMP, the flow table may be viewed as an RTFM meter MIB and the traffic meter as an SNMP agent. An example of RTFM is NeTraMet (Network Traffic Meter) at the University of Auckland [29]. However, commercial implementations of RTFM have not materialized yet.

For a flow, the basic RTFM traffic meter attempts to collect limited usage data in terms of: 1) flow start/stop times (the first and last packets); 2) total bytes in forward and backward directions; and 3) total packets in forward and backward directions. These attributes are essentially counts so the flow record can be updated in a simple manner. However, this simple usage data has limited usefulness, mainly for network planning and accounting/billing. Some additional attributes have been proposed but not agreed upon yet, such as packet size distribution, data rate distribution, short-term data rate distribution, and short-term packet rate distribution.

A similar direction has been taken up recently by the IETF IP Flow Information Export (IPFIX) working group which is concentrating on the data collection and reporting aspects instead of the traffic measurements themselves [30]. The preliminary IPFIX architecture is shown in Fig. 2(b). The observation point refers to the location where the IP traffic is observed, whether it is a LAN, router, or transmission link. The metering process is the set of actions performed on the packets including filtering, classification, timestamping, sampling, and updating flow statistics. The export process prepares the flow data for transmission to flow collectors (e.g., network management systems). The objective is a standardized flow record format and transmission process (but the traffic metering process will not be specified).

Single-point measurements can be implemented near a router rather than as a native capability within a router. Routers usually feature a capability to mirror incoming traffic to a specific port, where a traffic meter can be attached. CoralReef is a software suite that has evolved from OC3mon/OC12mon [31], [32]. OC3mon was designed to passively tap OC-3 optical links (via a splitter) in MCI's vBNS backbone network and capture IP over ATM traffic. Implemented as software running on PCs with ATM interface cards, it is able to store the IP header information (the first ATM cell from an IP packet contains the IP/TCP headers) or all ATM cell headers for later off-line analysis. CoralReef runs on Unix workstations to process data collected from OC3mon/OC12mon hardware or data collected from the workstation's network interface by using the libpcap library. Libpcap is a standard way to access IP data and Berkeley packet filter (BPF) devices. The libpcap library is also used by NeTraMet, ntop [33], packet sniffer programs, and various commercial network analyzers such as Narus' traffic analyzers [34] and Niksun's NetVCR [35].

While single-point traffic measurements provide valuable data, performance measurements require the cooperation of at least two measurement points (perhaps two routers or two hosts). The measurements can be made actively by injecting test traffic from one measurement point to the other point, or made passively by marking data packets with a unique identification that can be recognized by the measurement points. In the active approach, test traffic is differentiated from data traffic, and only test packets are observed by the measurement points. The methodology should be designed properly such that the performance seen by the test traffic can be correlated with the actual performance seen by the data traffic. Examples of active measurements include pings and the ATM OAM procedure discussed later. Active approaches have the advantage that data packets do not need to be handled any differently than usual, but care must be taken to limit the test traffic overhead to a very small portion of the total traffic to reduce the consumption of additional bandwidth and minimize the effect of the test traffic on the network performance being measured. Hence, there is a basic tradeoff between more measurements for accuracy (with more test traffic) and fewer measurements for less overhead. Passive measurements have two important advantages over active measurements: no additional traffic is involved and the performance of data packets is observed directly. However, the practical problem of uniquely marking packets to enable measurements is constrained by existing protocols that have not allocated a packet header field for this purpose. Any packet marking procedure must be compatible with already defined packet header fields, which is problematic. Hence, passive performance measurements are not used in practice.

IV. Active Network Layer Measurements

Performance can be monitored within an administrative domain (but not across the entire Internet) using the network layer protocol below IP. In particular, ATM and MPLS are discussed because in-service OAM performance monitoring has been designed as part of ATM, and MPLS is also a label-switching protocol. The network layer has a potential advantage over the IP layer because the switches can participate. For example, ATM switches have an active role in OAM procedures for fault notification and recovery. ATM has found extensive use in private broadband networks and high-speed wide area backbone networks. Performance management and fault management in the ATM layer make use of the same OAM protocol [36]. OAM cells have the regular ATM cell header but carry control data in the information field. Their flexibility enables a variety of in-service or out-of-service uses, such as alarms, monitoring, notifications, and testing.

The OAM performance management cell allows in-service performance monitoring as shown in Fig. 3. The basic idea is to embed OAM cells into the data connection at regular

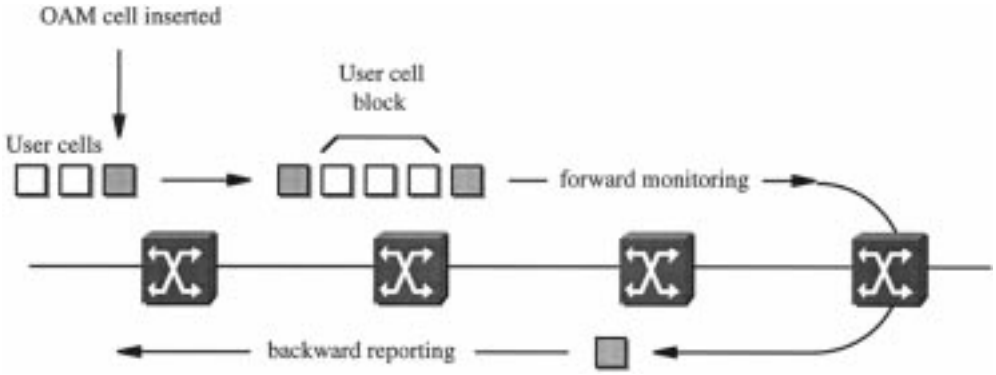

Fig. 3. ATM OAM performance management procedure.

points (OAM performance management cells are inserted between blocks of 2^n user data cells where n can range from 7 to 10). The OAM cells carry management information from the sender through an ATM virtual connection to the receiver. At the end of the virtual connection, the received block of user cells is compared with the OAM information carried in the following OAM cell. The results of the forward monitoring are reported in OAM cells in the backward direction.

The OAM performance management cell contains these fields: monitoring cell sequence number (MCSN), TUC_{0+1} (total user cells with CLP $= 0 + 1$), BEDC_{0+1} (block error detection code for $CLP = 0 + 1$ cells), TUC_0 (total user cells with CLP $= 0$), optional timestamp, TRCC_0 (total received cell count for CLP $= 0$ cells), BLER_{0+1} (block error result for CLP $= 0 + 1$ cells), and TRCC_{0+1} (total received cell count for CLP $= 0+1$ cells). The MCSN field identifies the OAM cell uniquely which is useful for detecting OAM cell losses. The TUC_{0+1} field is a cumulative count of the CLP $= 0 + 1$ user cells transmitted prior to this OAM cell. The TUC_0 field is similar except it applies only to CLP $= 0$ user cells. The TUC fields are needed to detect the loss of user cells at the end of the VP/VC connection. The difference between two consecutive TUC values is the size of the user cell block that was transmitted between two consecutive OAM cells. The cumulative cell count is used instead of simply the user cell block size in order to make the field robust against possible OAM cell losses. For example, if an OAM cell is lost, the TUC field in the next OAM cell can still be used to detect cell loss.

The BEDC_{0+1} field is an even parity BIP-16 error detection code computed over the information fields of the preceding block of CLP $= 0 + 1$ user cells. At the end of the VP/VC connection, the same error check is calculated over the received block of user cells and compared with the BEDC field in the OAM cell. The number of errored BIP-16 parity bits detected is returned in the BLER_{0+1} field in the OAM cell returned in the backward direction.

The optional timestamp field records the time when the OAM cell is inserted into the VP/VC connection. If the timestamp is returned in the backward reporting cell, it enables a measurement of roundtrip cell delay which may be useful for some higher-layer protocols but does not directly reflect the one-way cell transfer delay that is most relevant to ATM-layer quality of service (QoS). If the endpoints of the VP/VC connection are synchronized to the same time-of-day (e.g., by GPS), the timestamp field would enable a direct measurement of one-way cell transfer delay.

In the backward reporting cell, the TRCC_{0+1} field is a cumulative count of the CLP $= 0 + 1$ user cells received prior to the backward reporting cell. Like the TUC field, a cumulative count is used instead of the received user cell block size to make the procedure robust against the possible loss of OAM cells. The TRCC_0 field is similar except it applies only to CLP $= 0$ user cells. The combination of TUC and TRCC fields enables detection of cell loss. First, the size of the transmitted user cell block is calculated as N cells from the difference of consecutive TUC fields; the size of the received user cell block is calculated as K cells from the difference of TRCC fields. Next, N and K are compared and classified as one of three cases: 1) if $N = K$, then no cell loss is inferred; 2) if $N > K$, then $N - K$ cells are inferred to be lost; and 3) if $N < K$, then $K - N$ cells are inferred to be misinserted.

The OAM performance monitoring method is essentially an end-to-end approach where the end hosts inject OAM information into an ATM connection and observe the output, treating the ATM network as a black box. The ATM switches simply relay the OAM cells without modification (although they can observe them to collect performance data). Cell loss or misinsertion occurring somewhere in the VP/VC connection must be inferred with a possibility of mis-inference. For example, if $N = K$, then zero cells are inferred to be lost or misinserted, but the same result might have been caused by an equal number of lost and misinserted cells. If intermediate ATM switches are allowed to process and modify OAM cells, the OAM performance monitoring procedure can yield more accurate and informative data compared to the current black box approach. To distinguish the new procedure from the standard OAM procedure, it has been proposed to refer to the new cells as "management cells" which can be used for a variety of management and control purposes, in addition to performance monitoring [37].

MPLS is a label-switching protocol designed to combine certain features of ATM and IP [38]. Label switched paths must be established through the MPLS network. When IP packets enter the MPLS network, the ingress label switched

router (LSR) will attach a label to the packet to associate it with an LSP. Within the MPLS network, the packet will be forwarded based only on its label. When a packet departs from the MPLS network, the egress LSR will remove the label. The need for an OAM in-service performance monitoring procedure has been considered briefly by the IETF [39], [40] and ITU-T, but has not found industry support at this time. An OAM procedure has not been defined and would be constrained by backward compatibility issues. Still, an OAM mechanism in the MPLS user plane might have potential uses similar to the ATM OAM mechanism. A presumed MPLS OAM procedure could: 1) continuously check the integrity of an LSP; 2) allow LSRs to exchange fault notification information in the event of a detected fault; 3) allow testing of LSP segments on demand to identify the fault location; and 4) measure performance along the LSP in terms of packet loss, transfer delay, and bit errors. Presumably, OAM packets identified by a reserved label value could be injected and carried within an LSP between data packets. An important constraint is that the OAM procedure should be backward compatible with existing LSRs so that LSRs without OAM capabilities will simply forward or discard OAM packets without any detrimental effect on data packets. Certain OAM procedures such as loopback testing or continuity checking might not work properly if OAM packets are handled inconsistently by existing LSRs without OAM capabilities. Another possible problem for MPLS OAM might be the absence of a backward path. LSPs are currently defined as unidirectional (a bidirectional LSP is viewed as a combination of two unidirectional LSPs). The absence of a backward path would cause a problem for backward reporting in OAM performance monitoring, for example.

Other monitoring mechanisms have been proposed for MPLS and starting to see implementation in various degrees. MPLS ICMP refers to an extension to ICMP to include MPLS information in ICMP messages [41]. Like other routers, LSRs use ICMP messages for trouble reporting, e.g., sending a "destination unreachable" message back to the source if a packet cannot be forwarded. The destination unreachable message would contain a reason for the problem and part of the IP packet (including the header) but would not contain any MPLS-specific information such as the label stack when the LSR received the packet. The proposed MPLS ICMP would allow ICMP messages to carry the label stack information in addition to existing fields. Thus, if traceroute is used to discover the route through an MPLS network, the LSRs along the path will be discovered as well as the label stacks of the packet when it arrived at each LSR.

A proposed LSP-ping mechanism works in a similar manner to the traditional ping to test the integrity of an LSP on demand, except that it works through RSVP (resource reservation protocol) in the control plane [42], [43]. This would allow the control plane to discover the state of an LSP if regular pings (ICMP echo request messages) fail to be returned, suggesting that an LSP might have gone down. An ingress LSR sends an "LSP-ping" message containing a new RSVP "LSP_echo" object and a unique source identifier number to the egress LSR at the end of the tested LSP. The egress LSR copies the source identifier into an RSVP RESV message returned to the ingress LSR.

A generic tunnel tracing protocol has been proposed for discovering details of an IP route including any MPLS tunnels along the route (although it is not restricted to only MPLS tunnels) [44]. A host sends out successive UDP-encapsulated TraceProbe messages in to learn about each hop along the route. The TraceProbe message includes information about the route (application's address, ingress router address, destination address); a unique sequence number to match the returned TraceResponse messages; a top level hops (TLH) field used similarly to the TTL field in traceroute; and a tunnel hop identifier field (for the hop in a tunnel that is being discovered). The first TraceProbe goes one hop (TLH $= 1$) and prompts the next router to return a TraceResponse message. Among other information, the TraceProbe may contain a "tunnel identifier object" field about the type of tunnel that exists on that hop. Subsequent TraceProbes are sent further by incrementing their TLH values, and TraceResponses are returned from routers along the route. This round of TraceProbes is enough to learn the top level hops of a route and some initial information about any lower-level tunnels. A second round of TraceProbes/TraceResponses would uncover details about the next level of tunnels, and additional rounds of TraceProbes/TraceResponses would uncover increasingly lower levels of tunnels (if any).

V. Active IP/ICMP Layer Measurements

It is often desirable to measure performance at the IP layer because measurements across the entire Internet are possible and easy to carry out. In contrast, although network-layer mechanisms (such as ATM OAM) might be powerful, they are limited to the scope of a single homogeneous domain. However, the IP/ICMP protocol offers few options for performance monitoring at the IP layer. By far, most tools or methods are based on ping (ICMP echo and echo response messages) or traceroute (which exploits the TTL field in the IP packet header). Some variations of "classic" ping include Nikhef ping, skping (part of skitter), fping, pingplotter, gnuplotping, Imeter, pingroute, pathping, echoping, and traceping. Variations of traditional traceroute include Nikhef traceroute, pathchar, WhatRoute, neotrace, visualroute, Xtraceroute, GTrace, and network probe daemon (NPD). A list of these tools is maintained by CAIDA [45] and SLAC (Stanford Linear Accelerator Center) [46].

A few large-scale monitoring projects are using ping and traceroute to actively monitor the network performance between multiple selected points in the Internet. The basic idea is that performance measured on the routes of the virtual mesh defined by these monitoring points will reflect the performance of the general Internet if the monitoring points are numerous and geographically distributed around the network. Repeated pings are an easy way to obtain a sample distribution function of roundtrip time and an estimate of packet loss ratio (reflected by the fraction of unreturned pings) between two hosts. In some projects, dedicated hosts or special software are required at the participating sites for var-

ious reasons: the measurments involve complicated or programmed tests; the machines are isolated for security purposes; or the machines have special requirements. Although roundtrip times measured by ping are important, especially for adaptive protocols like TCP, ping is unable to measure one-way delay without additional means such as GPS to synchronize the clocks at the sender and destination hosts. Another difficulty is the low priority or blocking given to pings by some networks, because pings are invasive and might be involved in some types of denial of service attacks. Traceroute is good at discovering the routers between a pair of hosts. By observing the round-trip time (RTT) for each ICMP message returned from a router, it is possible to estimate the RTT to each router along the path. Traceroute will not encounter the possible ICMP blocking problem because UDP packets are used. However, traceroute has known limitations. For example, successive UDP packets sent by traceroute are not guaranteed to follow the same path. Also, a returned ICMP message may not follow the same path as the UDP packet that triggered it.

The Ping End-to-end Reporting (PingER) project at SLAC uses repeated pings around various Energy Sciences Network (ESnet) sites and other high-energy nuclear and particle physics (HENP) locations around the world [47]. A monitoring node sends 11 pings with a 100-byte payload at 1-s intervals, followed by 10 pings with a 1-kbyte payload at 1-s intervals, to each remote node listed in a configuration file. The first ping is discarded because it is assumed to be slow due to primary caches. Each combination of monitoring node and remote node is called a pair. PingER reportedly covers 1977 pairs with 511 remote nodes at 355 sites in 54 countries. The RTTs, packet loss, unreachability (inferred if all 10 pings are lost), quiescence (inferred if all 10 pings are returned), and unpredictability are measured. Unpredictability u is a metric calculated as

$$u = \sqrt{\frac{(1-r)^2 + (1-s)^2}{2}} \tag{1}$$

where r is the ratio of average to maximum ping rate (ping payload divided by average RTT) and s is the ratio of average to maximum ping success. Next, the performance of TCP is inferred from the ping statistics by the upper bound

$$\text{TCP rate} < \frac{\text{MSS}}{\text{RTT}\sqrt{p}} \tag{2}$$

where MSS is maximum segment size (typically 1460 bytes), RTT is the round-trip time estimated by TCP, and p is the packet loss rate [15].

The Active Measurement Program (AMP) project by the National Laboratory for Applied Network Research (NLANR) performs pings and traceroutes between various NSF-approved high-performance connection sites [48]. The objective is to carry out measurements of roundtrip time, packet loss, connectivity, and throughput between pairs of sites. To date, approximately 100 monitors have been deployed among these sites. Each monitor sends a single ICMP packet to each of the other sites every minute and records the RTT. Also, the routes between every monitor is recorded using traceroute every 10 min. The objective is to continuously take snapshots of the network status, providing a near-real-time view. Throughput tests can be run between any pair of monitors to measure bulk transfer capability but these tests are conducted only when needed due to traffic considerations.

The National Internet Measurement Infrastructure (NIMI) project measures the performance between various sites using traceroute or TCP bulk transfer [49]. Measurements are conducted by special platforms that must be deployed at sites. The platforms run NPD, a program functioning as a measurement server that can accept requests to either measure the route between the NPD host and a remote host using traceroute, or to source or sink a TCP bulk transfer and record the packets using tcpdump. A new method is being developed to multicast from a NIMI platform to a large number of receivers and analyzing the pattern of received packets to infer delay and loss between nodes [50].

The Surveyor project is unusual in attempting to measure one-way packet delay and loss (rather than round-trip delay) following IPPM performance metrics [51]. One-way delay measurements are enabled by measurement probes that are equipped with GPS for time synchronization. Measurement probes are dedicated machines, and measurements are made only between these machines. Dedicated hardware ensures that each machine is consistent and runs with controlled load. Also, they must be equipped with GPS receivers and secured to maintain data integrity. Poisson streams of test traffic are sent at an average rate of two per second. Each UDP packet carries 12 bytes of data, essentially a sequence number and timestamp. Poisson streams (where test packets are separated by random interpacket times according to an exponential probability distribution) are considered to provide measurement samples that are more "random" than periodic streams of test packets. For example, periodic measurements would miss any periodic network events that happen to occur between measurements and would always catch periodic events that coincide with the periodic measurements. In contrast, Poisson measurements are random in the sense that they are uniformly distributed over any given interval of time, which results in truly random snapshots of the network condition.

CAIDA's skitter is a traceroute-like software tool for discovering topology and measuring RTTs on discovered paths [12]. Fifty-two-byte ICMP echo request messages are sent out with varying TTL values, and the RTTs are recorded. The probing frequency is limited to one packet every 2 min to each destination and 300 packets/s to all destinations. Unlike the other large-scale projects mentioned earlier, skitter attempts a much broader geographic coverage (actually across the entire Internet) instead of a number of pre-selected monitoring sites. Beginning in 1998, hundreds of thousands of hosts around the world, represented by many destinations throughout the IPv4 address space, are probed (as well as intermediate routers to each host). The result is a spanning tree diagram rooted at a polling host and extending outward

to the polled destination hosts. The results are used to visualize the macroscopic network topology, detect sources of abnormal delay, locate critical paths in the network, and detect low-frequency routing changes.

Some ISPs are choosing to publish performance data measured over their networks, usually as evidence to claim conformance to SLAs. For example, performance of AT&T's IP network can be viewed in terms of backbone delay, backbone loss, connection success rate, and availability [52]. An active testing method is reportedly used to measure roundtrip delay and packet loss between pairs of cities. Cable and Wireless offers SLA performance measurements obtained by pings and traceroute [53]. Real-time and monthly cumulative statistics of latency and loss are shown between regions. Qwest makes available IP network performance statistics including on-demand measurements of availability, FTP delay, HTTP delay, packet latency, and packet loss [54].

Performance data provided by ISPs might be suspect without corroborating data from independent sources. A number of commercial sources provide large-scale monitoring of various Internet routes using pings (other commercial services use the application/transport layer as described later). These are motivated partially by users who want to verify their ISP performance or compare ISPs. The Internet Traffic Report sends pings on a set of major routes from multiple servers located around the world and updates the map of five continents every 15 min [55]. The recent ping delay sample from a router is compared to all previous responses from the same router over the past week, and then a "traffic index" score between 0 and 100 is assigned depending on how the new ping sample compares with previous responses. Packet loss is also calculated from the percent of unreturned pings. Naturally, the accuracy of the results is subject to the problems of ping mentioned earlier.

Matrix Net Systems offers the Internet Weather Report with maps showing RTTs measured by pings from Austin, Texas, to thousands of Internet domains worldwide [56]. The roundtrip time is an average of five pings to each site. The results are projected onto various geographical maps and updated every four hours. Matrix Net Systems also offers Matrix.Net ISP Ratings consisting of measurements of median latency, packet loss, and reachability for various ISP networks [57]. Beacons are PCs that run custom data collection software to measure a set of routes (called a viewlist) selected for each ISP. Viewlists are claimed to be carefully contructed to accurately sample a network's performance from an external viewpoint (ISPs are encouraged to cooperate in their construction). ISPs are monitored from multiple beacons distributed around the world according to geography, Internet topology, and proximity to a large number of Internet nodes. A beacon conducts a scan every 15 min sending ICMP echo requests to each destination on its viewlists, and records the RTT. In addition to pings, beacons also use FTP, DNS, SMTP, and other protocol traffic for supplemental measurements. From the collected data, ISPs are ranked by performance for the previous day, week, and month. It should be noted however, that some ISPs have strenuously objected to the measurement methodology as inaccurate.

VI. Transport/Application Layer Measurements

Although end-to-end performance measurements can be carried out at the IP layer or the transport/application layer, the transport/application layer is capable of measurements closer to the application's perspective. In addition, the transport/application layer is appealing because it does not depend on ICMP as in ping or traceroute (with their inherent problems). At the transport/application layer, the basic idea is to run a program emulating a particular application or TCP that will send test traffic through the Internet; the performance (delay, loss, throughput) of the emulated application or TCP connection will be measured from the test traffic. A drawback is that usually custom software needs to be installed at the hosts to enable the measurements. Also, tests could involve considerably more traffic through the network than simple pings.

Traceroute Reno (TReno) is an emulation of TCP to measure throughput or bulk transfer capability [58]. It combines traceroute and an idealized version of the flow control algorithms in Reno TCP. TReno probes the network with either ICMP echo packets or UDP packets with low TTL values, which solicits ICMP errors (as in traceroute). The probe packets are subject to the same congestion effects as TCP. TReno uses the same sequence numbers to emulate TCP and performs an idealized version of the TCP congestion avoidance and slow start algorithms. Although TReno emulates TCP, it is different in that only the sending host needs to maintain state information, and TReno does not actually retransmit lost packets but records a virtual retransmission. Also, TReno emulates idealized TCP with selective acknowledgment (SACK) while some existing implementations of TCP do not use selective acknowledgment.

Throughput TCP (ttcp) and netperf have been widely used to measure transport-layer throughput; ttcp is a client/server benchmarking program to measure throughput and retransmissions, and netperf is a network performance benchmarking program developed by H-P [59]. It measures bulk data transfer and request/response performance using TCP or UDP. One host runs netperf and another host runs netserver. When netperf runs, the netserver program at the other host will be invoked by the establishment of a control connection to pass test configuration information and results. The control connection is always TCP, and a separate connection is made for the test. The most common test is a 10-s TCP stream performance test. A UDP bulk transfer or TCP request/response test can also be done.

For on-demand application testing, a common method is to install software agents at multiple hosts, as with NetIQ's Chariot or Qcheck [60]. These proprietary agents are configured with scripts to emulate various applications, and collect application-level performance (throughput, delay) measurements from tests.

Keynote is an example of a commercial venture to benchmark performance at the application layer. Keynote offers

Internet performance indexes that measure the average dial-up download time for the home pages of 40 major US-based websites in different sectors (business, consumer, etc.) [61]. These measurements are taken by automated agents attached to key points in the Internet backbone in the largest metropolitan areas in the US. The agents use local low-latency, uncongested connections to supposedly ensure that any problems seen by the agent are due to ISP problems and not the agent's own access link. Their statistics are interpreted to represent well-connected business users. Additional agents are connected via low bandwidth T-1 lines, DSL lines, cable modem, and dial-up lines, which are more typical of small business users and home users. Agents run standard Windows operating systems and TCP/IP software to be similar to end users experiences. They are programmed to emulate web browsing to major websites and measure download performance. As with Matrix Net Systems, Keynote has drawn sharp criticism from ISPs on a few points. First, it has been noted that the Keynote measurements are essentially web downloading tests affected by many variables, but no decomposition of the measured performance into the separate variables is done. Also, Keynote uses the web downloading tests to infer conclusions about backbone performance, which is not directly measured. Additional objections have been raised about the scientific rigor of the methodology and its underlying assumptions. Finally, it has been noted that Keynote seeks the cooperation of ISPs for the placement of probes, and those ISPs providing cooperation and funding typically improve their results in the Keynote measurements. The pressure on ISPs to provide cooperation and funding have been cause for objections to Keynote and Matrix Net.

VII. Research Issues

Ultimately, the purpose of performance monitoring is to assure Internet users and service providers that their services are meeting expectations or to identify the causes of problems if services are inadequate. In this survey, it is evident that a combination of *ad hoc* measurement methods are used in practice today. They are adequate for rudimentary measurements but are too limited for the next-generation Internet supporting a diversity of demanding services.

The definition of meaningful and comprehensive metrics must be the first issue to be addressed . The IETF IPPM work represents an important step toward common, standardized IP-layer performance metrics. However, performance statistics defined in a time-average sense will naturally fall short in completely characterizing the Internet which is inherently a large, distributed, and dynamic system. Statistics collected and averaged over a past period of time may not be relevant to current conditions. Performance metrics should be defined to also capture the dynamics of the Internet, but this is not well understood due to the complexity of the Internet system. There have been limited studies of routing dynamics and TCP congestion avoidance behavior, but the general dynamics of the Internet on a system-wide level remains a challenging research problem. The problem will become more complicated when the Internet evolves beyond the current best-effort service architecture, and performance will have to be characterized for different service classes.

The second important issue is how to performance measurements should be made. Passive measurements require high-speed instrumentation in routers to meter traffic at the rates of transmission and perform high-speed processing of enormous volumes of measurement data to make results available in near real time. Active measurements consist of a combination of ping variations, traceroute variations, and emulated applications or TCP. Large-scale active measurements are being carried by various organizations, but little effort is being spent to ensure uniformity or coordination.

Why not let ISPs monitor their own networks and publicize their performance data? First, ISPs are occupied with daily operations of their networks and looking for competitive advantages. Investment in better performance monitoring methods is likely to remain a low priority unless there is a compelling business case. Second, it is uncertain whether service providers would be willing to report internal problems, and doubtful that they would welcome external studies to monitor their networks. This might make it difficult for users to diagnose troubles.

As an alternative, a common measurement infrastructure might ensure that performance measurements will be end-to-end, consistent, statistically accurate, fair, secure (from unauthorized theft, eavesdropping, or tampering), and safe. Today, this need is partially addressed by PingER, NLANR, NIMI, and Surveyor, which will continue to be valuable research platforms. This need is also being addressed by commercial ventures attempting to benchmark ISP performance. A common measurement infrastructure might alleviate the pressure on ISPs to appease these commercial ventures. But more research is needed to determine the possible need for a common measurement infrastructure, and if needed, to converge on the most appropriate monitoring methodologies. All of the large-scale monitoring projects are based on active methods but active monitoring is inherently invasive and should be carefully designed and controlled.

Finally, a major issue is the substantial gap between existing capabilities for measurement and analysis. For example, we can easily collect an enormous amount of traffic data, but traffic analysis is still a time consuming job for specialists. Many traffic models have been developed from traffic studies over the years, but traffic control in practice is simplistic and based on assumptions that may or may not be accurate. We believe that the industry needs much better analytical tools to bind traffic and performance measurement more closely with network management.

VIII. Conclusion

In this paper, we have surveyed a large number of methods and tools for measuring traffic and network performance. For router-based traffic measurement, we believe the instrumentation of standardized traffic metering and reporting into

next-generation routers (in the sense of IPFIX) will be very helpful for traffic studies.

Detailed performance monitoring can be done at the network level, e.g., with ATM and perhaps eventually with MPLS. However, substantial challenges remain for performance measurements across the entire Internet. The industry has relied for many years on two primitive utilities, ping and traceroute, for IP-layer measurements. An abundance of tools built on variations of these utilities are widely available, but these tools are inadequate for detailed or precise performance monitoring.

When the next-generation Internet becomes capable of supporting QoS and more demanding applications become commonplace, there will be a more urgent need for sophisticated performance measurements. The development of better tools and methods, and a common measurement infrastructure to support them, will require industry-wide cooperation that is not evident today.

ACKNOWLEDGMENT

The authors are deeply indebted to the liaison editor, Prof. B. Jabbari, the guest editors, and anonymous reviewers for their many constructive comments which have been invaluable for improving an earlier version of this paper.

REFERENCES

[1] "Internet protocol data communication service – IP packet transfer and availability performance parameters," ITU-T Draft Recommendation I.380, Geneva, Switzerland, 1999.
[2] V. Paxson *et al.*, "Framework for IP performance metrics," IETF RFC 2330, 1998.
[3] J. Mahdavi and V. Paxson, "IPPM metrics for measuring connectivity," IETF RFC 2678, 1999.
[4] G. Almes, S. Kalidindi, and M. Zekauskas, "A one-way delay metric for IPPM," IETF RFC 2679, 1999.
[5] ——, "A round-trip delay metric for IPPM," IETF RFC 2681, 1999.
[6] ——, "A one-way packet loss metric for IPPM," IETF RFC 2680, 1999.
[7] A. Downey, "Using pathchar to estimate internet link characteristics," *Comp. Commun. Rev.*, vol. 29, pp. 241–250, Oct. 1999.
[8] Clink. [Online]. Available: http://rocky.wellesley.edu/downey/clink/
[9] J.-C. Bolot, "End-to-end packet delay and loss behavior in the internet," in *Proc. ACM Sigcomm'93*, 1993, pp. 289–298.
[10] K. Lai and M. Baker, "Measuring bandwidth," in *Proc. IEEE Infocom 99*, Mar. 1999.
[11] K. Claffy and T. Monk, "What's next for internet data analysis? Status and challenges facing the community," *Proc. IEEE*, vol. 85, pp. 1563–1571, Oct. 1997.
[12] CAIDA Skitter. [Online]. Available: http://www.caida.org/tools/measurement/skitter/
[13] Multi Router Traffic Grapher (MRTG). [Online]. Available: http://ee-staff.ethz.ch/~oetiker/webtools/mrtg/
[14] S. Floyd and K. Fall, "Promoting the use of end-to-end congestiona control in the internet," *IEEE/ACM Trans. Networking*, vol. 7, pp. 458–472, Aug. 1999.
[15] M. Mathis *et al.*, "The macroscopic behavior of the TCP congestion avoidance mechanism," *Comp. Commun. Rev.*, vol. 27, pp. 67–82, July 1997.
[16] M. Crovella and A. Bestavros, "Self-similarity in world wide web traffic: Evidence and possible causes," *IEEE/ACM Trans. Networking*, vol. 5, pp. 835–846, Dec. 1997.
[17] B. Tsybakov and N. Georganas, "On self-similar traffic in ATM queues: Definitions, overflow probability bound, and cell delay distribution," *IEEE/ACM Trans. Networking*, vol. 5, pp. 397–409, June 1997.
[18] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. Networking*, vol. 1, pp. 397–413, Aug. 1993.
[19] K. Thompson, G. Miller, and R. Wilder, "Wide-area internet traffic patterns and characteristics," *IEEE Network*, vol. 11, pp. 10–23, Nov. 1997.
[20] J.-S. Park, J.-Y. Lee, and S.-B. Lee, "Internet traffic measurement and analysis in a high speed network environment: Workload and flow characteristics," *J. Commun. Networks*, vol. 2, pp. 287–296, Sept. 2000.
[21] P. Danzig *et al.*, "An empirical workload model for driving wide-area TCP/IP network simulations," *Internetworking Res. Exper.*, vol. 3, pp. 1–26, 1992.
[22] K. Claffy, H.-W. Braun, and G. Polyzos, "A parameterizable methodology for internet traffic flow profiling," *IEEE J. Select. Areas Commun.*, vol. 13, pp. 1481–1494, Oct. 1995.
[23] Cisco NetFlow. [Online]. Available: http://www.cisco.com/warp/public/732/Tech/netflow/
[24] Abilene NetFlow Nightly Reports. [Online]. Available: http://www.itec.oar.net/abilene-netflow/
[25] P. Phaal, S. Panchen, and N. McKee, "InMon Corporation's sFlow: A method for monitoring traffic in switched and routed networks," IETF RFC 3176, 2001.
[26] Flow-Tools Information. [Online]. Available: http://www.splintered.net/sw/flow-tools/
[27] CAIDA Cflowd. [Online]. Available: http://www.caida.org/tools/measurement/cflowd/
[28] N. Brownlee, C. Mills, and G. Ruth, "Traffic flow measurement: Architecture," IETF RFC 2722, 1999.
[29] N. Brownlee, "Traffic flow measurement: Experiences with NeTraMet," IETF RFC 2123, 1997.
[30] J. Quittek, T. Zseby, and B. Claise, Eds., "Requirements for IP flow information export," IETF, to be published.
[31] J. Apisdorf, K. Claffy, K. Thompson, and R. Wilder, "OC3MON: Flexible, affordable, high performance statistics collection," in *Proc. INET'97*, Kuala Lumpur, Malaysia, June, 24–27 1997.
[32] K. Keys *et al.*, "The architecture of CoralReef: An internet traffic monitoring software suite," in *Proc. PAM2001 Workshop on Passive and Active Measurements)*, Amsterdam, The Netherlands, Apr., 23–24 2001.
[33] L. Deri and S. Suin, "Effect traffic measurement using ntop," *IEEE Commun. Mag.*, vol. 38, pp. 138–143, May 2000.
[34] Narus IBI Platform [Online]. Available: http://www.narus.com/ibi/
[35] Niksun NetVCR. [Online]. Available: http://www.niksun.com/products/netvcr.html
[36] "B-ISDN operation and maintenance principles and functions," ITU-T Rec. I.610, Geneva, Switzerland, 1993.
[37] T. Chen *et al.*, "Monitoring and control of ATM networks using special cells," *IEEE Network*, vol. 10, pp. 28–38, Sept. 1996.
[38] R. Callon, A. Viswanathan, and R. Callon, "Multiprotocol label switching architecture," IETF RFC 3031, 2001.
[39] N. Harrison *et al.*, "Requirements for OAM in MPLS networks," IETF draft, work in progress.
[40] D. Allan *et al.*, "A framework for MPLS user plane OAM," IETF draft, work in progress.
[41] R. Bonica, D. Tappan, and D. Gan, "ICMP extensions for multiprotocol label switching," IETF draft, work in progress.
[42] P. Pan *et al.*, "Detecting data plane liveliness in RSVP-TE," IETF draft, work in progress.
[43] B. Braden *et al.*, "Resource reservation protocol (RSVP) – Version 1 functional specification," IETF RFC 2205, 1997.
[44] "unpublished,", to be published.
[45] CAIDA Performance Measurement Tools Taxonomy. [Online]. Available: http://www.caida.org/tools/taxonomy/performance.xml
[46] SLAC Network Monitoring Tools. [Online]. Available: http://www.slac.stanford.edu/xorg/nmtf/nmtf-tools.html
[47] W. Matthews and L. Cottrell, "The PingER project: Active internet performance monitoring for the HENP community," *IEEE Commun. Mag.*, vol. 38, pp. 130–136, May 2000.
[48] T. McGregor, H.-W. Braun, and J. Brown, "The NLANR network analysis infrastructure," *IEEE Commun. Mag.*, vol. 38, pp. 122–128, May 2000.
[49] V. Paxson *et al.*, "An architecture for large-scale internet measurement," *IEEE Commun. Mag.*, vol. 36, pp. 48–54, Aug. 1998.
[50] A. Adams *et al.*, "The use of end-to-end multicast measurements for characterizing internal network behavior," *IEEE Commun. Mag.*, vol. 38, pp. 152–158, May 2000.

[51] S. Kalidindi and M. Zekauskas, "Surveyor: An infrastructure for internet performance measurements," in INET'99, June 1999.
[52] AT&T IP Network Performance. [Online]. Available: http://ipnetwork.bgtmo.ip.att.net/
[53] Cable and Wireless Global Internet Backbone SLA Performance Statistics. [Online]. Available: http://sla.cw.net/
[54] Qwest IP Network Statistics. [Online]. Available: http://stat.qwest.net/index.html
[55] Internet Traffic Report. [Online]. Available: http://www.InternetTrafficReport.com/
[56] Internet Weather Report. [Online]. Available: http://www.mids.org/weather/
[57] Matrix.Net ISP Ratings. [Online]. Available: http://ratings.miq.net/
[58] M. Mathis and J. Mahdavi, "Diagnosing internet congestion with a transport layer performance tool," in *Proc. INET'96*, Montreal, ON, Canada, June, 24–28 1996.
[59] Netperf Public Page. [Online]. Available: http://www.netperf.org/netperf/NetperfPage.html
[60] NetIQ Corp. [Online]. Available: http://www.netiq.com/
[61] Keynote Performance Benchmarks [Online]. Available: http://www.keynote.com/company/html/services.html

Thomas M. Chen (Senior Member, IEEE) received the B.S. and M.S. degrees from the Massachusetts Institute of Technology, Cambridge, and the Ph.D. degree from the University of California, Berkeley, all in electrical engineering.

He is an Associate Professor in the Department of Electrical Engineering and a faculty affiliate of the Linda and Mitch Hart e-Center at Southern Methodist University (SMU), Dallas, TX. Prior to joining SMU, he worked on ATM research at GTE Laboratories (now Verizon), Waltham, MA. He is the coauthor of the monograph *ATM Switching Systems* (Norwell, MA: Artech House, 1995).

Dr. Chen is a Senior Technical Editor for IEEE NETWORK, a Senior Technical Editor for *IEEE Communications Magazine*, past Founding Editor of *IEEE Communications Surveys*, and an Associate Editor for *ACM Transactions on Internet Technology.* He was the recipient of the IEEE Communications Society's Fred W. Ellersick best paper award in 1996.

Lucia Hu graduated from the Texas Academy of Math and Sciences, Denton, TX, in May 2002.

Her work on this paper was contributed during a research internship at Southern Methodist University, Dallas, TX, in the summer of 2001.