Trajectory Sampling: White Paper
Draft

Nick Duffield Matthias Grossglauser

April 10, 2003

1 Executive Summary

Trajectory Sampling (TS) is a novel method to measure network traffic in potentially large network domains [7]. It is designed to provide detailed views of network traffic that can drive a wide variety of network engineering and management applications, such as traffic reporting and characterization, attack and intrusion detection and diagnosis, traffic engineering, and capacity planning.

TS implements consistent packet sampling. Conceptually, each packet traversing a measurement domain is sampled either on every link traversed, or on no link at all. Some highly compressed information on sampled packets are collected centrally in a collection system, which is then able to reconstruct the trajectory (path) of each sampled packet [5]. This set of sampled packet trajectories provides a complete and statistically representative view of the flow of traffic through the domain. A wide range of application-specific metrics and derived views (such as traffic matrices for traffic engineering or sink trees for a DDoS attack) can be inferred from this raw trajectory information.

The main advantages of trajectory sampling are:

- General and flexible: TS measurements are sufficiently fine-grained that a wide spectrum of application-dependent metrics of interest can be derived from them. These range from simple aggregate characterizations (e.g., the distribution of packet sizes or source addresses over all the packets traversing a peering link) to high-dimensional objects such as traffic matrices or path matrices [11], which measure the load on the network between every ingress-egress node pair, and the spatial flow of traffic through the network, respectively. Furthermore, TS is flexible in the sense that it is possible to customize various parameters (such as the set of links where sampling is enabled, the sampling rate, information extracted from packets, etc.) to suit particular applications and to optimize the tradeoff between measurement overhead and precision.
- Direct: TS measurements are direct - or complete - in the sense that it is not necessary for most applications to correlate trajectory samples with other auxiliary traffic and control data (such as routing tables) to compute the metrics and traffic views of interest. This is because trajectory samples are inherently very high-resolution measurements, providing detailed information about each sampled

packet including the full path followed by the packet. This eliminates many potential sources of error, reduces overhead and complexity, and simplifies the design of control and management applications built on top of a TS-enabled network.

- Scalable and implementable: Sampling techniques such as TS allow for an inherent tradeoff between measurement overhead (in terms of processing and bandwidth to collect the measurements) and precision of estimated metrics. As such, TS is inherently scalable, and can be incorporated cost-effectively into next-generation high speed interfaces. Furthermore, a TS measurement device on a router linecard is relatively simple, as it only needs to maintain minimal state. For example, it is not necessary for such a device to maintain per-flow state. The device performs a simple calculation in hardware for every packet traversing it, in order to make a sampling decision. More complex operations have to be performed only for the subset of sampled packets, which in typical applications constitute a small fraction of all the packets traversing the link. Also, the on-board memory requirements are minimal. For the sampled packets, the TS device extracts some fields of interest, generates reports, and exports report packets to a collection system.

- Future-safe: Trajectory sampling will be compatible with future internetworking technologies, and can therefore be expected to satisfy most needs for passive traffic measurement in the medium and long term. First, TS can be naturally integrated into an MPLS-enabled environment. In particular, TS makes it possible for packets to be "followed" through MPLS tunnels, which is an invaluable help in operating hybrid MPLS-IP networks. Third, TS is part of a broader IETF standardization effort for packet sampling measurement support: specifically, the PSAMP working group has recently been chartered to develop all the functional components of a packet sampling device, including support for trajectory sampling [2]. This ensures multi-vendor compatibility and industry-wide acceptance of this technology. Second, TS is inherently compatible with multicasting; the trajectory of a sampled multicast packet is simply a tree, rooted at the ingress node of that packet into the network, with all the egress points as its leaves. It is not necessary to extract any multicast group state from the network to follow the tree of a multicast packet. Third, TS can be adapted to IPv6. While the specific fields over which the hash function is computed to make sampling decisions is different, the fundamental principle of the method is the same and would require minor changes in both hardware and software to upgrade the measurement device and management systems to TS.

- No change to IP: A central design goal of TS was that no change to the IP protocol or to its packet format should be necessary. This ensures that TS can be deployed incrementally and completely transparently to any other system (routers, end-systems) outside the network of interest. It reduces friction points in the standardization and a rapid adoption of the technology, as it does not interfere in any way with protocols and systems other than those that perform the actual measurement tasks.

2 Network Engineering and Management Applications

In this section, we survey some typical uses of trajectory sampling in network management and engineering. We classify each one of these applications according to the following criteria:

- Time-scale: what are typical "reaction times" in the control loop that this measurement application is part of? Over what time intervals are metrics of interest in this application typically accumulated?

- Activation: an application is either *continuous*, i.e., measurements are collected proactively, or *on demand*, i.e., measurements are collected in response to a specific network condition.
- Granularity and scope: the "resolution" of the metric of interest in an application, from low (e.g., link utilization) to high (e.g., traffic matrix).
- Specialized measurements: what other sources of measurements other than TS could be employed to populate the metrics of interest?

The goals of this section are twofold. First, to catalogue the wide spectrum of control and management tasks and associated metrics that are required to operate a large state-of-the-art IP network. Second, to emphasize that TS has the potential to unify many of these measurement tasks using ubiquitous, consistent packet sampling.

2.1 Reporting and Characterization

Reporting and characterization entail computing highly aggregated statistics over long time-scales over the totality of the traffic carried by the network, or slightly smaller aggregates, such as the totality of traffic to or from a particular customer. The goal of these statistics is (a) to have a continuous verification of the overall operational health of the network; (b) to provide reports to customers about their traffic and to verify that service level agreements (SLAs) are adhered to; (c) to distinguish long-term trends in the evolution of the volume and composition of traffic, e.g., for capacity planning.

Profile:

- Time-scale: long (hours to months).
- Activation: continuous.
- Granularity and scope: low, typically highly aggregated statistics.
- Specialized measurements: SNMP MIB-II, NetFlow, sFlow.

2.2 Troubleshooting

Troubleshooting comprises the detection of anomalous network behavior (e.g., a link is congested), the diagnosis of this behavior (e.g., an OSPF link weight has been set too low), and correcting the problem. Troubleshooting typically proceeds through a sequence of increasingly refined measurements that are collected on-demand to "drill down" into a problem, and to confirm or refute hypotheses about what the underlying problem might be.

Profile:

- Time-scale: short (minutes to hours).
- Activation: on demand.

- Granularity and scope: incremental, “drill-down” to isolate problem.
- Specialized measurements: SNMP aggregate measurements [18] (MIB-II) for detection of overload conditions, etc.; NetFlow or sFlow to examine traffic on a specific link in more detail; RMON to obtain fine-grained measurements and packet traces from LANs.

2.3 Attack and Intrusion Detection and Diagnosis

Denial of service and other attacks and network intrusions are commonplace today, and having processes in place to deal with them is a must for a network operator. TS measurements can be used both in the detection phase, where the goal is essentially to look for suspicious changes in traffic patterns and for overload conditions, and in the diagnosis phase, where the attacker or set of attackers have to be isolated and their traffic blocked through appropriate filters.

Profile:

- Time-scale: short (minutes)
- Activation: continuous for detection, on demand for diagnosis
- Granularity and scope: low for detection (e.g., link utilizations of access links), high for diagnosis (sink tree of traffic destined to victim host)
- Specialized measurements: router support for attack tree inference [16, 17, 4]

2.4 Traffic and Network Engineering

Traffic engineering has the dual goal of satisfying customer performance demands and to optimize resource efficiency in the network [11]. The control actions that may be invoked to achieve these goals include changes in routing weights or the establishment of MPLS paths, traffic classification and packet marking, and resource allocation. These domain-wide control actions require domain-wide fine-grained traffic measurements, such as traffic matrices [9].

Network engineering tasks such as capacity and topology planning are conceptually similar in nature, but they occur over significantly longer time-scales (from weeks to months).

Profile:

- Time-scale: long (from hours, e.g., routing, to months, e.g., in capacity planning).
- Activation: continuous.
- Granularity and scope: high, various matrices (traffic matrix, demand matrix, path matrix, etc.).
- Specialized measurements: NetFlow in conjunction with auxiliary network state information, such as routing tables to infer ingress/egress points, paths through network.

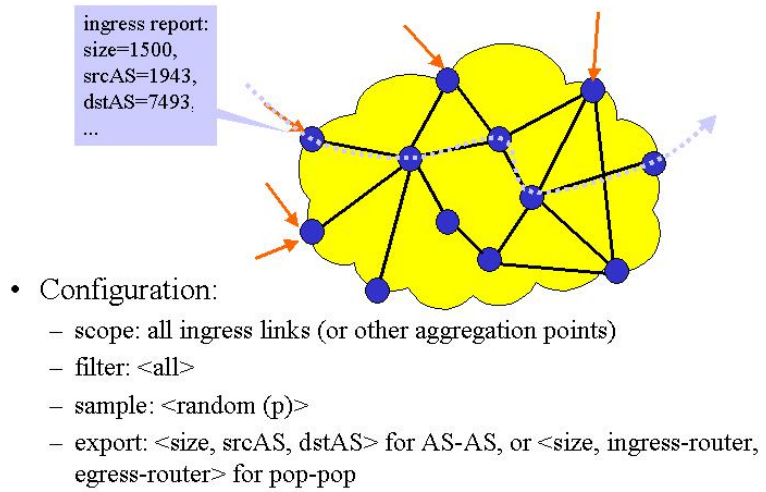

Figure 1: An example of how a TS-enabled network can be configured to measure a traffic matrix.

In summary, trajectory sampling can drive a wide spectrum of crucial measurement-based control and management applications. These applications and their associated measurements cover a range of time scales from minutes to months, may be activated on-demand or collected continuously, and are or very different levels of granularity.

Finally, we would like to emphasize that the above generic applications have many variants in practice. For example, a common occurrence is for a set of measurements to be performed only for a well-defined subset of the traffic in a domain. Examples: packets of a certain class, with a source or destination address with a given prefix, etc. Such "slicing and dicing" can be incorporated into a TS device by adding a general-purpose filtering function. Such filtering support is being standardized within the PSAMP effort.

3 Benefits to Operators and Vendors

3.1 Benefits for Network Operators

- Trajectory sampling can enable a wide spectrum of crucial measurement tasks such as traffic reporting, SLA verification, and characterization; troubleshooting and attack detection; traffic and network engineering. In a competitive, large scale environment, these tasks are crucial to provide reliable, predictable service efficiently.

- Dual use for *continuous tracking* of operational health of network and accumulation of measurements for potential post-mortem analysis, and *on-demand* activation of more fine-grained measurements ("drill-down") for troubleshooting, intrusion detection, etc.

- Direct approach, requiring no correlation and synchronization with auxiliary network state information. This simplifies the design of the measurement and management infrastructure, reduces

overhead, and eliminates many sources of error.

- TS is being standardized within the IETF working group PSAMP.

3.2 Benefits for Network Equipment Vendors

- Generality and flexibility of TS represents a unique opportunity for operators to deploy a *single, simple technology* that responds to increasing demand by operators for better visibility into traffic and network behavior, support for intrusion and attack detection, etc.
- It relieves the pressure for vendors to develop wide array of *application-specific measurement technologies* and support them across all platforms (e.g., IP traceback for DDoS attacks, NetFlow for traffic characterization and traffic engineering, sFlow or RMON for fine-grained packet capture).
- It relieves the pressure to support efficient export of auxiliary network state information, such as routing tables and interface state, e.g., to determine how traffic flows through the network.

3.3 Benefits for Network Management Vendors

- TS eliminates a major source of complexity in sophisticated traffic management applications, namely the need to correlate multiple sources of measurement data and network state information. This correlation step is difficult and error-prone, because data may be incompatible, subject to errors and timing inconsistencies, etc. Developing robust applications on top of a multi-vendor environment is therefore very challenging. The fact that TS is a *direct* technique simplifies the design of robust applications.
- TS is being standardized with the PSAMP IETF working group; this means that it will be straightforward for application vendors to support a *multi-vendor environment*.
- Prototype of a traffic engineering and visualization tool called Trajectory Engine demonstrates the viability of the approach [5].

4 Comparison with other Approaches

In this section, we survey existing traffic measurement techniques and compare them to trajectory sampling. We consider the following criteria for each technique:

- Generality and flexibility: How large is the set of applications supported by the measurement technique? Can the technique be easily adapted to specific constraints and tasks?
- Granularity and scope: What is the (temporal and/or spatial) resolution of the measurements extracted from the network by the technique?
- Timeliness: Are there delays inherent in the measurement technique?

- Overhead: How much computational, storage, and bandwidth overhead does the technique incur?
- Implementation: What are the major technical challenges in supporting the technique cost-effectively even at the highest link speeds?

4.1 SNMP (without RMON)

The SNMP standard defines the representation and protocols for the exchange of network management information between agents and management systems [19, 18]. In addition to this, the standard also includes a set of Management Information Bases (MIBs), i.e., a standardized set of variables that can be queried in routers and other network elements.

- Generality and flexibility: MIB-II (and related vendor-specific MIBs in the `enterprise` group) are the result of a multi-year standardization effort. It is not possible to easily extend or alter the set of variables that can be observed.
- Granularity and scope: Mostly highly aggregated, e.g., counters of various events (number of packets and bytes into and out of interfaces and ports, number of errors, CPU load, etc.) However, there is no way to drill down further if this predefined variables are not sufficient, e.g., to infer the cause of a link overload.
- Timeliness: To avoid agent overload, polling rates are not high in practice; a typical polling interval is 5 mins. This is too long for some real-time control applications.
- Overhead: Not problematic.
- Implementation: Not problematic, agent is typically implemented in software.

4.2 RMON

The Remote Monitoring (RMON) MIBs significantly extend the agent's functionality over the traffic measurement support embedded in MIB-II [19, 15]. An RMON agent has more local memory and more intelligence in order to be robust to connectivity outages between the agent and the management system, and to reduce communication overhead between the two. RMON achieves this in the following way: (a) by allowing an RMON probe (e.g., a router or switch or a dedicated probe) to examine all traffic on shared-media LANs to which it is attached, rather than just the traffic actually passing through it; (b) by defining several additional MIBs with more complex statistics than MIB-II, such as a MAC-level traffic matrix over a LAN segment, or a `history` group, which implements a local buffer to record past measurements; (c) by making more extensive use of asynchronous notifications to the management system when certain user-defined alarm conditions are met; and (d) by providing very flexible support to define filter conditions over packets and to capture matching packets for retrieval back to the management system.

- Generality and flexibility: RMON is a much more highly configurable measurement system than MIB-II. However, (a) there is no support for statistically sampling packets, which makes it impossible to exploit the inherent tradeoff in sampling between overhead and precision; (b) there is no support for "agent-initiated" export of captured packets (packets are stored locally and have to be explicitly queried for by the management system; (c) it is not possible to combine packet reports on the fly with local router state, such as source and destination AS number.

- Granularity and scope: Very fine-grained, per-host statistics on shared LANs, support to filter packets according to very flexible boolean operations over packet header and content, capturing of filtered packets, and export of desired fields.

- Timeliness: Similar to MIB-II above.

- Overhead: Very application-dependent. Filtering and capturing are typically very demanding operations in current RMON implementations, potentially slowing down a router considerably when they are enabled.

- Implementation: Routers and switches typically only implement a subset of RMON groups, and exclusively for LAN interfaces. This is because a full implementation of RMON, especially including support for packet filter and capture, is very costly. As a result, full RMON implementations are only available in dedicated RMON probes; also, RMON support is currently not available for high-speed backbone interfaces. It is doubtful that RMON will every make inroads into the core of the network, and as such is not relevant for operators for large backbones.

4.3 Cisco NetFlow

Cisco NetFlow is an example of a flow measurement approach [1]. A *flow* is a set of packets traversing a link that share some common property, such as identical source and destination address and some timing proximity, such as an upper bound on the interarrival time between consecutive packets. Depending on its definition, a flow may (approximately) correspond to a TCP session between two hosts, the traffic between two customer sites, or other traffic aggregates of interest.

The central component of the measurement device is a *cache of active flows*. Every arriving packet has to be matched against the active flows in the cache in order to update per-flow metrics of interest, such as the total number of bytes or packets for this flow. Once a flow expires, e.g., because the maximal interarrival time has been exceeded, a flow report is exported and sent to the collection system.

- Generality and flexibility: Flow measurements can serve a rather wide set of applications, except those for which flows are not the right abstraction, i.e., where valuable information is lost in the aggregation process. For example, flow measurements would not allow to estimate packet size distributions.

- Granularity and scope: Many metrics of interest that do not depend on individual packets can be derived from flow measurements. For example, by measuring traffic over all ingress links into a network, the totality of the traffic carried by that network can be accounted for and characterized (e.g., distribution of source/destination address, port, or AS). However, it is very difficult or impossible to

measure the paths taken by flows through the network, because (a) the overhead of measuring flows at every link would be considerable, and (b) it is difficult to correlate flow records at different links. Therefore, it is necessary to correlate NetFlow records with routing information to infer the flow of traffic through the network [8].

- Timeliness: A reporting delay is inherent in the method, as a flow is reported only once it has terminated. Depending on how long flows are allowed to remain in the cache, this may limit the applicability to real-time control problems that require tight feedback loops, e.g., in attack detection or in restoration.
- Overhead: The number of flows that will be generated depends not only on the definition of what constitutes a flow (e.g., interarrival timers), but also on the traffic itself. The overhead is therefore unpredictable. For example, in a DDoS attack, many flows consisting of a single packet may be measured because source addresses are spoofed. This may overload the measurement system (flow cache overflow, export bandwidth).
- Implementation: The bottleneck and most costly component in a flow measurement device is the high-speed cache of active flows. This cache has to be accessed for every packet, and it has to be large enough to hold the possibly large number of flow records. TS compares favorably to NetFlow, as there is no need to keep any state in the measurement device, and no costly per-packet lookups into a high-speed cache are necessary.

4.4 sFlow

sFlow is an Internet standard for packet-sampling support in routers and other network equipment [13]. Packets can either be sampled $1/n$, meaning that every nth packet traversing an interface is sampled, or randomly based on a pseudo-random number generator. Attributes of interest are extracted from the sampled packets and exported through UDP to a collection system. Along with these packet reports, other auxiliary information (e.g., counters, source and destination AS) can be associated with the report as well.

As sFlow does not embody support for consistent sampling (which in trajectory sampling is achieved through the use of hash functions), it is not possible to ensure that a sampled packets gets reported from every link it traverses. In general, then, we only get a single snapshot for each packet, rather than a full trajectory. While this information is sufficient for a range of important applications, it implies that applications that rely on metrics that depend on the sampled packets' paths (such as a DDoS attack sink tree or a routing loop) must rely on auxiliary network state information to *infer* these paths. This results in additional overhead, and is a potential source of errors. TS in contrast provides a *direct* observation, which is more robust and efficient.

- Generality and flexibility: Packet sampling as supported in sFlow can drive a wide range of applications. As discussed above, the main difference between sFlow and trajectory sampling is that full packet trajectories cannot be extracted *directly* from sFlow packet samples, because sampling is not consistent.
- Granularity and scope: Packet sampling is not inherently limited in the granularity of the information extracted from measurements. There exists an obvious tradeoff between measurement overhead and precision.

- Timeliness: Comparable to TS.
- Overhead: Comparable to TS.
- Implementation: The sampling device for sFlow only requires a small, simple set of operations per packet to decide whether a packet should be sampled or not, and to extract the information of interest from sampled packets. TS is slightly more complex than sFlow, as the sampling decision depends on a hash function computed over the packet, rather than just a counter for $1/n$ sampling. However, this overhead is comparable to that of computing a CRC over the packet, and can easily be performed at line speed in hardware.

4.5 Active Measurements

Active measurements consist of (a) injecting probing traffic into the network, and (b) measuring performance metrics of interest over these probes (e.g., [14, 3, 12, 10]). The main goal of active measurements is to directly measure the availability and performance of *services* provided by the network, from simple end-to-end connectivity all the way to full web client-server exchanges or complex e-commerce transactions. In general, active measurements provide the most authoritative verification that the network is functioning properly and that it provides the services it is designed to support. As such, active measurements often provide the first indication of trouble when a failure occurs in the network.

- Generality and flexibility: The goal of active measurement is fundamentally different from the passive measurement methods described above. Active measurements complement passive measurement methods; they typically are better suited for detecting performance problems and outages, and passive measurements are more likely to be used in the diagnosis phase.
- Granularity and scope: Usually low, e.g., statistics on delay, loss, and throughput (at different layers) between pairs of end-systems.
- Timeliness: Good.
- Overhead: Active measurements inherently impose additional load on the network and possibly on end-systems. This may distort measurement results if care is not taken that the probing traffic has a negligible impact on the network performance.
- Implementation: In general, active measurements are generated and collected at the periphery of the network in general-purpose computers. One of the most difficult and potentially expensive engineering problems is time synchronization between different measurement points. Other than this, active measurements are usually implemented purely in software, and rely on an infrastructure that is considerably less costly than that required for passive measurements at line speeds in the core of the network.

Table 1 summarizes our overview of measurement technologies and the applications they enable. Note that this table focuses specifically on the vantage point of a large, high-speed IP backbone; thus, for example, RMON is not a viable option for several applications, as it has only been deployed at the network edge.

	SNMP	RMON	NetFlow	sFlow	TS
Reporting	●	◒	●	●	●
Troubleshooting	◒		●	●	●
Attack detection & diagnosis			◒		●
Traffic engineering			◒	◒	●
Generality and flexibility		◒	●	●	●
Granularity & scope		●	◒	◒	●
Timeliness				●	●
Low overhead	●			●	●
Implementable	●		◒	●	●

●	enabled
◒	limited

Table 1: Comparison of applications enabled by different measurement technologies.

5 Research

Trajectory Sampling was first proposed in [6, 7]. An important focus of this initial study was the statistical validity of pseudo-random sampling based on hash functions. Specifically, the question was whether a sample drawn from a larger population of packets would allow to draw inferences about the full population (e.g., the distribution of packet sizes, source and destination addresses, or the probability for a packet being dropped at a particular link), *as if the sampling process had been random*. This essentially depends on having enough entropy, or variation, in packets. In this respect, it is always beneficial to compute the hash function over a larger part of the packet (excluding, of course, fields that vary from hop to hop, such as TTL), in that the sampling process will "look more random". In [6, 7], extensive simulation results indicated that including the IP and TCP header in the hash function seemed to be sufficient to ensure that trajectory samples are statistically representative for the full population.

Another issue addressed in [6, 7] was the tradeoff between the overhead to transport trajectory samples to a central collection system for processing, and the precision of estimators (or metrics) computed from these samples. A method was proposed to minimize this overhead, whereby a compact *label* is computed for every sampled packet using a second hash function computed over the packet. Although a small fraction of samples have to be eliminated because of the possibility of label collision (i.e., two or more packets having the same label), the compression of the trajectory information collected from the network is significant, compared with collecting entire packets (or even packet headers). Labels would typically be between 20 to 30 bits per sampled packet.

In [5], a prototype collection and visualization system called the Trajectory Engine is described. The goal of this prototype is to demonstrate the broad set of applications supported by trajectory sampling without requiring other auxiliary network measurements. The Trajectory Engine consists of several modules for real-time reconstruction of trajectories from label streams received from the network, a database to store

these measurement, a querying interface to formulate common queries that arise in traffic engineering, and a visualization interface to graphically display the results of such queries.

References

[1] Cisco Netflow.
`http://www.cisco.com/warp/public/732/netflow/index.html`.

[2] PSAMP IETF working group.
`http://www.ietf.org/html.charters/psamp-charter.html`.

[3] Traceroute.org.
`http://www.traceroute.org`.

[4] S. Bellovin. ICMP Traceback Messages. *Internet Draft - work in progress*, March 2000.
available from `http://www.ietf.org`.

[5] N. G. Duffield, A. Gerber, and M. Grossglauser. Trajectory Engine: A Backend for Trajectory Sampling. In *Proc. Network Operations and Management Symposium (NOMS)*, Florence, Italy, April 2002.
`http://www.research.att.com/~mgross/Papers/trfmeas_sfibook.ps`.

[6] N. G. Duffield and M. Grossglauser. Trajectory Sampling for Direct Traffic Observation. In *Proc. ACM SIGCOMM 2000*, Stockholm, Sweden, August 2000.

[7] N. G. Duffield and M. Grossglauser. Trajectory Sampling for Direct Traffic Observation. *IEEE/ACM Transactions on Networking*, 9(3):280–292, June 2001.

[8] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford, and F. True. Deriving traffic demands for operational IP networks: Methodology and experience. *IEEE/ACM Transactions on Networking*, June 2001.
`http://www.research.att.com/~jrex/papers/ton01.ps`.

[9] B. Fortz and M. Thorup. Internet Traffic Engineering by Optimizing OSPF Weights. In *Proc. IEEE INFOCOM 2000*, April 2000.

[10] F. Georgatos, F. Gruber, D. Karrenberger, M. Santcroos, A. Susanj, H. Uijterwaal, and R. Wilhelm. Providing Active Measurements as a Regular Service for ISPs. In *Proc. PAM 2001 Workshop on Passive and Active Measurements*, Amsterdam, April 2001.

[11] M. Grossglauser and J. Rexford. Passive Traffic Measurement for IP Operations. In *The Internet as a Large-Scale Complex System (Kihong Park and Walter Willinger, eds.)*. Oxford University Press (to appear), 2002.

[12] Van Jacobson. Pathchar.
`ftp://ftp.ee.lbl.gov/pathchar`.

[13] N. McKee P. Phaal, S. Panchen. InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks. RFC 3176, IETF, September 2001. `http://www.rfc-editor.org/rfc/rfc3176.txt`.

[14] V. Paxson, G. Almes, J. Mahdavi, and M. Mathis. Framework for IP Performance Metrics. *RFC 2330, available from* `http://www.ietf.org/rfc`, May 1998.

[15] David T. Perkins. *RMON: Remote Monitoring of SNMP-Managed LANs*. Prentice Hall, September 1998.

[16] Stefan Savage, David Wetherall, Anna Karlin, and Tom Anderson. Practical Network Support for IP Traceback. *IEEE/ACM Transactions on Networking*, 9(3):226–237, June 2001.

[17] A. C. Snoeren, C. Partridge, L. A. Sanchez, Ch. E. Jones, F. Tchakountio, S. T. Kent, and W. T. Strayer. Hash-Based IP Traceback. In *ACM SIGCOMM 2001*, San Diego, CA, August 2001.

[18] William Stallings. SNMP and SNMPv2: The Infrastructure for Network Management. *IEEE Communications Magazine*, March 1998.

[19] William Stallings. *SNMP, SNMP v2, SNMP v3, and RMON 1 and 2 (Third Edition)*. Addison-Wesley, Reading, Mass., 1999.