Design and Deployment of a Passive Monitoring Infrastructure

Chuck Fraleigh, Christophe Diot, Bryan Lyles, Sue Moon
Philippe Owezarski, Dina Papagiannaki, Fouad Tobagi

Abstract— **This paper presents the architecture of a passive monitoring system installed within the Sprint IP backbone network. This system differs from other packet monitoring systems in that it collects packet-level traces from multiple links within the network and provides the capability to correlate the data using highly accurate GPS timestamps. After a thorough description of the monitoring systems, we demonstrate the system's capabilities and the diversity of the results that can be obtained from the collected data. These results include workload characterization, packet size analysis, and packet delay incurred through a single backbone router. We conclude with lessons learned from the development of the monitoring infrastructure and present future research goals.**

I. Introduction

Network traffic measurements provide essential data for networking research and operation. Collecting and analyzing such data from a tier 1 ISP backbone, however, is a challenging task. The traffic volume ranges from tens of Mb/sec on OC-3 access links to 10 Gb/sec on OC-192 backbone links. The measurement equipment must be installed in commercial network facilities where physical space and power are constrained, and which are, in some cases, not staffed by any human operators. Data analysis involves processing terabytes of data, and must account for unusual phenomena such as routing loops and malicious network users.

This paper presents our experiences developing the Sprint IP Monitoring System, a traffic measurement system for the Sprint Internet IP network. The Sprint Internet network is a tier 1 IP backbone connecting 20 Points of Presence (POPs) in the United States. The monitoring system is designed to collect synchronized traces of traffic data from multiple links for use in in-depth research projects where aggregate statistics are insufficient.

The Sprint IP Monitoring System consists of three basic components:

- A set of data collection systems (IPMON systems) which collect TCP/IP headers of every packet transmitted on various links in the Sprint network, along with a measurement system that collects BGP and IS-IS routing information. Currently 11 IPMON systems are deployed at one POP in the Sprint network. Two additional POPs, each with 10 systems, are scheduled to be installed.
- A data repository which archives the traces collected by the IPMON systems.
- A 16 node computing cluster used for data analysis.

The design of the system focuses on four major issues: collecting packet traces from high speed links, synchronizing the traces, manipulating the large data sets, and administering the system. Collecting traces from high speed OC-3 (155 Mb/sec) and OC-12 (622 Mb/sec) links has been addressed by several prior measurement systems [1][2]. Our system follows the same general design, and we discuss some of the challenges when extending it to OC-48 (2.48 Gb/sec) speeds. Synchronizing the clocks that generate the packet timestamps is accomplished using a stratum-1 GPS reference clock distributed to the IPMON systems. The data set for a single 24 hour trace collected on all of the IPMON system is 1.1 TB. This data is compressed and transmitted from the IPMON systems to the data repository over a dedicated OC-3 link. The data repository and data analysis systems are interconnected using a gigabit Ethernet network. All system administration functions for the IPMON systems may be performed from the lab. In cases of extreme failures, the entire operating system may be reinstalled over the network.

The remainder of the paper describes the details of how we address these design issues in the IP Monitoring System and presents some sample results that demonstrate the system's capabilities. Section II discusses other work on IP monitoring systems and compares them with our monitoring infrastructure. Section III describes the architecture of the network in which our monitoring systems are installed. Section IV presents the design requirements and details of the monitoring system. Section V presents traffic measurements which demonstrate the capabilities of our system and which evaluate the system performance. Section VI concludes and discusses areas of future research.

II. Related Work

There has been much work on active and passive network measurement systems. A measurement system is called active if it injects measurement traffic, such as probe packets, in the network. Passive measurement systems, on the other hand, do not inject any measurement traffic but rather observe the actual traffic flowing in the network.

Active measurement systems include NIMI, MINC, Surveyor, AMP, and IEPM. The NIMI (National Internet Measurement Infrastructure) project developed an archi-

Chuck Fraleigh and Fouad Tobagi are with the Department of Electrical Engineering, Stanford University, Stanford, CA, USA. Email {cjf,tobagi}@stanford.edu

Christophe Diot, Bryan Lyles, Sue Moon, and Dina Papagiannaki are with the Sprint Advanced Technology Lab, Burlingame, CA, USA. Email {cdiot,lyles, sbmoon,dina}@sprintlabs.com

Philippe Owezarski is with LAAS-CNRS, Toulouse, France. Email owe@laas.fr

tecture for deploying and managing scalable active measurement systems [3]. The NIMI system uses tools such as *ping*, *traceroute*, *mtrace*, and *treno* to perform the actual measurements [4][5]. MINC (Multicast-based Inference of Network-internal Characteristics) measurement systems transmit multicast probe messages to many destinations, and infer link loss rates, delay distributions, and topology based on the observed correlations of the received packets [6]. Surveyor uses a set of approximately 50 GPS synchronized hosts to measure one-way and round-trip delay over various Internet paths [7]. The AMP (NLANR Active Measurement Project) system consists of a set of monitoring stations which measure the performance of the vBNS backbone [8]. IEPM (Internet End-to-end Performance Monitoring) monitors network performance between high energy nuclear and particle physics research institutions [9]. In addition, companies such as Keynote and Matrix conduct commercial network performance measurements [10][11].

Passive measurement systems include Simple Network Management Protocol (SNMP)-based network traffic measurement tools, *tcpdump*, NetFlow, and CoralReef. SNMP is the most widely used network management protocol in today's Internet [12]. Agents and remote monitors update a management information base (MIB) within network routers, and management stations retrieve MIB information from the routers using UDP. Most routers support SNMP and implement public MIBs as well as vendor-specific private MIBs. Using SNMP, for example, network operators can keep track of the number of packets and bytes that have arrived on an interface, the number of packets and bytes that have been dropped on an interface, and the number of transmission errors that have occurred on a link. Another common network monitoring tool is *tcpdump*. It collects packets transmitted and received by systems running the Unix operating system [13]. NetFlow is a monitoring system available on Cisco routers which collects flow statistics observed by an interface [14]. NetFlow provides more detailed information than is available through SNMP, but it requires an external system to record the NetFlow data. The CoralReef suite, developed by CAIDA and originally based on the OC3MON developed at MCI, collects timestamped packet traces from various ATM and SONET links [15][1]. This system is very similar to our monitoring system, but it does not have GPS synchronization.

Other efforts have been made in routing and standardization of metrics. The Internet Performance Measurement and Analysis (IPMA) project investigates routing behavior and network failures [16]. The IETF IP Performance Metrics (IPPM) working group standardizes metrics for evaluating network performance based on observations realized within the projects described above [17]. To our knowledge, the only project to address network-wide traffic analysis in a comprehensive manner has been developed at AT&T [18]. This project relies on packet-level information collected by packet sniffers called PacketScopes, flow statistics collected using Cisco's NetFlow tools, and routing information. It also includes active components which collect loss, delay, and connectivity statistics. Results from active and passive components are combined to be used in network monitoring and management of the AT&T backbone.

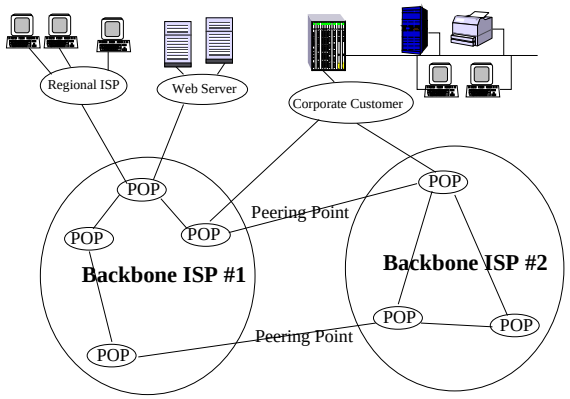

Fig. 1. Tier 1 IP backbone network

Our project is unique in that it allows trace collection at different points in a commercial backbone network, and it provides the capability to correlate these traces through highly accurate timestamps. Many other traffic measurement systems are installed in either research or access networks [19][8][20]. Some results from commercial IP backbones are available in [21][2], but they do not have the GPS synchronization capabilities of our system.

III. Backbone Network Description

A backbone IP network provides connectivity over a geographically wide area. The network topology consists of a set of nodes known as Points-of-Presence (POPs) connected by high bandwidth backbone links. These links are typically 2.5 Gb/sec OC-48 links or 10 Gb/sec OC-192 links. Each POP also contains links to customers (e.g. large corporate networks, regional ISPs providing dial-up access, large web servers), ranging from 1.5 Mb/sec T1 links to 622 Mb/sec OC-12 links. Figure 1 shows the architecture of a backbone network.

A backbone network connects to other backbone networks at private peering points or public network access points (NAPs). Two networks can peer at multiple points to accommodate the traffic volume between them as shown in Figure 1. The peering points are intended to carry traffic that originates from a customer connected to one backbone ISP and is destined to a customer of another backbone ISP. Most peering agreements prohibit transit traffic, or traffic whose source and destination are not customers of the backbone ISP. For example, Backbone ISP #2 would not accept traffic from Backbone ISP #1 if the destination was not one of Backbone ISP #2's customers.

In case of the Sprint Internet backbone, there are 20 POPs located in the continental United States. The POPs in the Sprint Internet backbone have a two-level hierarchical structure as shown in Figure 2. At the lower level, customer links are connected to access aggregation routers. The access routers are in turn connected to the higher level backbone routers. The backbone routers provide connectivity to other POPs, and they also connect to public

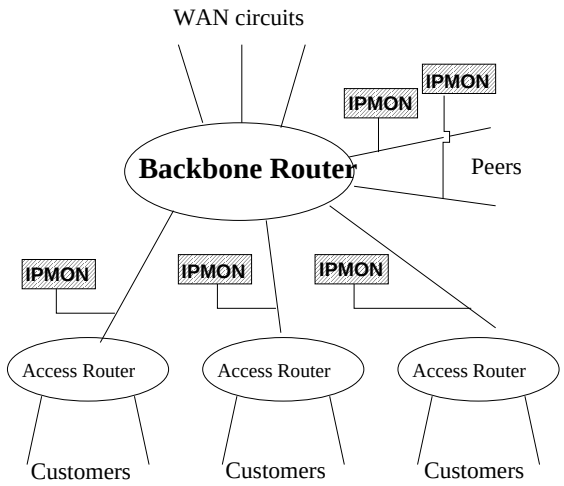

Fig. 2. POP architecture

and private peering points. The backbone links connecting the POPs are optical wavelengths with a bandwidth of 2.5 Gb/sec (OC-48) or 10 Gb/sec (OC-192). All of the links carry IP traffic using a proprietary version of the Packet-over-SONET (POS) protocol similar to that proposed in [22].

IV. System Description

The goal of the Sprint IP Monitoring System is to collect data from the Sprint Internet backbone that is needed to support a variety of research projects. The particular research projects include studying the behavior of TCP, evaluating network delay performance, investigating the nature of denial of service attacks, and developing network engineering tools. While each project could develop a customized measurement system, many of the projects require similar types of data and installing monitoring equipment in a commercial network is a complex task making a general purpose measurement system more preferable.

To meet this goal, the IP Monitoring system is designed to collect and analyze synchronized packet level traces from selected links in the Sprint Internet backbone. These packet traces consist of the first 44 bytes of every packet carried on the links along with a 64 bit timestamp. The clocks which generate the timestamps are synchronized to within 5 μs using a GPS reference clock. This provides the capability to measure one-way network delays and study correlations in traffic patterns.

Trace collection and analysis is accomplished using three separate systems. A set of data collection components (IPMON systems) collect the packet traces. The traces are transferred to a data repository which stores the traces until they are needed for analysis. Analysis is performed on a cluster of 16 Linux servers. The remainder of this section describes the requirements and architecture of each of these components.

bytes	field description
0	8 byte
4	timestamp
8	record size (fixed to 64 bytes)
12	POS frame length
16	HDLC header
20	First 44 bytes of IP packet
⋮	
64	

TABLE I
Packet Record Format

A. IPMON Systems

The monitoring systems, called IPMON systems, are responsible for collecting packet traces from the network. These system consists of a Linux PC with a large disk array and a SONET network interface, known as the DAG card [23][24]. To collect the traces, an optical splitter is installed on selected OC-3, OC-12, and OC-48 links in the network, and one output of the splitter is connected to the DAG card in the IPMON system.

The DAG card decodes the SONET payloads and extracts the IP packets. When the beginning of a packet is identified, the DAG card generates a timestamp for the packet, extracts the first 48 bytes of the POS frame which contains 4 bytes of POS header and 44 bytes of IP data, and transfers the packet record to main memory in the PC using DMA. The format of the packet record is shown in Table I. If the packet contains fewer than 44 bytes, the data is padded with all 0's. Once 1 MB of data has been copied to main memory, the DAG card generates an interrupt which triggers an application to copy the data from main memory to the hard disk. It would be possible to transfer the data from the DAG card directly to the hard disk, and bypass the main memory. Main memory, however, is necessary to buffer bursts of traffic as described later in the section.

The IPMON system has 5 basic design requirements:

- Support data rates ranging from 155 Mb/sec (OC-3) to 2.5 Gb/sec (OC-48)
- Provide synchronized timestamps
- Occupy a minimal amount of physical space
- Prevent unauthorized access to trace data
- Be capable of remote administration

Next we describe how each of these requirements are met in the system design.

A.1 Data Rate Requirements

The data rate requirements for OC-3, OC-12, and OC-48 links are summarized in Table II. The first line of the table shows the data rate at which the DAG card must be able to process incoming packets. After the DAG card has received a packet and extracted the first 44 bytes, the timestamp and additional header information is added to the packet record and copied to main memory. If there is

	OC-3	OC-12	OC-48
link rate (Mb/sec)	155	622	2480
peak capture rate (Mb/sec)	248	992	3968
1 hour trace size (GB)	11	42	176

TABLE II
DATA RATE REQUIREMENTS

a sequence of consecutive packets whose size is less than 64 bytes, then the amount of data that is stored to main memory is actually greater than the line rate of the monitored link. The amount of internal bandwidth required to copy a sequence of records corresponding to minimum size TCP packets (40 byte packets) from the DAG card to main memory is shown on the second line of Table II. To support this data rate, the OC-3 and OC-12 IPMON systems use a standard 32 bit, 33 MHz PCI bus which has a capacity of 1056 Mb/sec (132 MB/sec). The OC-48 system, however, requires a 64 bit, 66 MHz PCI bus with a capacity of 4224 Mb/sec (528 MB/sec). It is possible to have non-TCP packets which are smaller than 40 bytes resulting in even higher bandwidth requirements, but the system is not designed to handle extended bursts of these packets as they do not occur very frequently. It is assumed that the small buffers located on the DAG card can handle short bursts of packets less than 40 bytes in size. The impact of this design decision is evaluated in Section V.

Once the data has been stored in main memory, the system must be able to copy the data from memory to disk. The bandwidth required for this operation, however, is significantly lower than the amount of bandwidth needed to copy the data from the DAG card to main memory as the main memory buffers bursts of small packets before storing them to disk. Only 64 bytes of information are recorded for each packet that is observed on the link. As reported in prior studies, the average packet size observed on backbone links ranges from about 300-400 bytes during the busy periods of the day [21]. For our design, we assume an average packet size of 400 bytes. The disk I/O bandwidth requirements are therefore only 16% of the actual link rate. For OC-3 this is 24.8 Mb/sec; for OC-12, 99.5 Mb/sec; and for OC-48, 396.8 Mb/sec. To support these data rates, we use a three-disk RAID array for the OC-3 and OC-12 systems which has an I/O capacity of 240 Mb/sec (30 MB/sec). The RAID array uses a software RAID controller available with Linux. To support OC-48 we use a five-disk RAID array with higher performance disks that can support 400 Mb/sec (50 MB/sec) transfers. To minimize interference with the data being transferred from the DAG card to memory, the disk controllers use a separate 32 bit 33 MHz PCI bus.

A.2 Timestamp Requirements

In order to correlate the traces, the packet timestamps generated by each IPMON system need to be synchronized to a global clock signal. This is accomplished using a dedicated clock on board the DAG card. The clock runs at a rate of 16MHz which provides a granularity of 59.6 ns between clock ticks.

Unfortunately, the oscillators used to drive the clock run just a little bit faster or a little bit slower than 16 MHz based on system temperature and the quality of the oscillator. Therefore, it is necessary to fine tune, or discipline, the clocks using an external stratum 1 GPS receiver located at the POP sites. The GPS receiver outputs a 1 pulse-per-second (PPS) signal which is distributed to all of the DAG cards located at the POP.

The clocks synchronization on board the DAG card operate in the following manner [25]. At the beginning of trace collection the clock is loaded with the absolute time from the PC's system clock (e.g. 7:00 am Aug 9, 2000 PST). The clock then begins to increment at a rate of 16 MHz. When the DAG card receives the first 1 PPS signal after initialization, it resets the lower 24 bits of the clock counter (note: 24 bits will count from 0 to 16 million. If the lower 24 bits of the clock are all 0 it represents the beginning of a second). Thereafter, each time the DAG card receives the 1 PPS signal, it compares the lower 24 bits of the clock to 0. If the value is greater than 0, the oscillator is running a little bit fast and the DAG card decreases the frequency slightly. If the value is less than 0, the oscillator is running a little bit slow and the DAG card increases the frequency slightly.

In addition to synchronizing the DAG clocks, the IPMON systems must also synchronize their own internal clocks so that the DAG clock is correctly initialized. This is accomplished using NTP. A broadcast NTP server is installed on the LAN which is connected to the monitoring systems and is capable of synchronizing the system clocks in the PC to within 200 ms. This is sufficient to synchronize the beginning of the traces, and the 1 PPS signal is used to further synchronize the DAG clock. There is an initial period where the 1 PPS is attempting to correct the initial clock skew, so we ignore the first 30 seconds of each trace to account for this.

There are several sources of error that may occur in the synchronization system. The first is clock skew between the 1 PPS signals generated by different GPS receivers located at different POPs. This error is minimal as we are using stratum 1 GPS receivers which are guaranteed to have a maximum clock skew of 500 ns. Another source of error is the difference in propagation time for the 1 PPS signal. The 1 PPS signal is distributed to the DAG cards using a daisy chain topology. The difference in cable length between the first and last systems is 8 meters, which corresponds to a propagation delay of 28 ns. Finally, the clock synchronization mechanism cannot immediately adjust to changes in the oscillator frequency, it needs to wait for the next 1 PPS signal. To test this aspect of performance, we measure the maximum clock error that is observed when the DAG card receives a 1 PPS interrupt. The maximum value we have seen in lab tests is 30 clock ticks which represents an error of 1.79 μs. The median error observed during these tests was 1 clock tick, or 59.6 ns. Adding all of these factors, the worst case skew between any two DAG clocks is less than 2 μs.

Another source of error is that packets are not timestamped immediately when they arrive at the DAG card. They must first pass through a chip which implements the SONET framing. As this chip was initially designed to operate on 53 byte ATM cells, it is possible for two packets (one 40 byte packet and the first 13 bytes of the next packet) to be placed in a cell buffer at the same time. Since this buffer is read as an entire unit, both packets will have identical timestamps. However, their actual inter-arrival time is 2 μs if we are measuring an OC-3 link. This results in an additional 2 μs of timestamp error. This is only an issue for the OC-3 and OC-12 DAG cards. The OC-48 systems use a newer SONET framing chip which was designed to support POS directly and does not use 53 byte buffers.

The total effect of these errors is a maximum of 5 μs of clock skew between DAG cards. However, we are interested in measuring the delay experienced by packets as they traverse the network. This delay is typically measured on the order of milliseconds, so the 5 μs skew is acceptable. The only case where the clock skew affects the measurements are when we are interested in measuring the delay through a single router in the network. The minimum delay we have observed is 30 μs, so a 5 μs skew represents a 16% error in the measurement.

A.3 Physical Requirements

In addition to supporting the bandwidth requirements of OC-3, OC-12, and OC-48 links, the IPMON systems must also have a large amount of hard disk storage capacity to record the traces. As the systems are installed in a commercial network facility where physical space is a scarce resource, this disk space must be contained in small form factor. Using a rack-optimized system, the OC-3 and OC-12 IPMONs are able to handle 108 GB of storage in only 4U of rack space[1]. This allows the system to record data for 9.8 hours on a fully utilized OC-3 link or 2.6 hours on a fully utilized OC-12 link. The OC-48 systems have a storage capacity of 360 GB, but in a slightly larger 7U form factor. This is sufficient to collect a 2 hour trace on a fully utilized link. Fortunately, the average link utilization on most links is less than 50%, allowing for longer trace collection.

The physical size constraint is one of the major limitations of the IPMON system. Collecting packet level traces requires significant amounts of hardware. These traces are critical for conducting research activities, but trace collection is not a scalable solution for operational monitoring of an entire network. The ideal solution is to use traces collected by the IPMON system to study the traffic and develop more efficient monitoring systems targeted towards exhaustive monitoring of all links in the network for operational purposes.

A.4 Security Requirements

Preventing unauthorized access to trace data is an important design requirement. This includes preventing access to trace data stored on the systems as well as preventing access to the systems in order to collect new data. To accomplish this, the systems are configured to accept network traffic only from two applications: *ssh* and NTP. *ssh* is an authenticated and encrypted communication program similar to *telnet* that provides to access a command line interface to the system. This command line interface is the only way to access trace data that has been collected by the system and to schedule new trace collections. *ssh* only accepts connections from a restricted set of servers and it uses an RSA key based system to authenticate users. All data that is transmitted over the *ssh* connection is encrypted.

The second type of network traffic accepted by the IPMON systems is NTP traffic. The systems only accept NTP messages which are transmitted as broadcast messages on a local network used exclusively by the IPMON systems. All broadcast messages which do not originate on this network are filtered.

A.5 Remote Administration Requirements

In addition to being secure, the IPMON systems must also be robust against failures since they are installed, in some cases, where there is no human presence. To detect failures, a server in the lab periodically sends query messages over an ssh connection to the IPMON systems. The response indicates the status of the DAG cards and of the NTP synchronization. If the response indicates either of these components fails, the server attempts to restart the component through an *ssh* tunnel. If the server is not able to correct the problem it notifies the system administrator that manual intervention is required. In some cases, even the *ssh* connection will fail, and the systems cannot be accessed over the network. To handle this type of failure, the systems are configured with a remote administration card that provides the capability to reboot the machine remotely. The remote administration card also provides remote access to the system console during boot time. In cases of extreme failure, the system administrator can boot from a write-protected floppy installed in the systems and completely reinstall the operating system remotely.

The one event that cannot be handled remotely is hardware failure. The monitoring systems play no role in the operation of the backbone, and thus we decided not to provide hardware redundancy to handle failures.

B. Data Repository

The data repository is a large tape library responsible for archiving the trace data. Once a set of traces has been collected on the IPMON systems, the trace data is transferred over a dedicated OC-3 connection from the IPMONs to the data repository.

A single 24-hour-long trace from all of the monitoring systems currently installed consumes approximately 1.2 TB of disk space (this will increase to 3.3 TB when the additional 20 systems are installed). The tape library has 10 individual tape drives which are able to write data at an aggregate rate of over 100 MB/sec. The rate at which the data can be transferred from the remote systems, however,

[1] 1U is a standard measure of rack space and is equal to 1.75 inches or 4.45 cm.

is limited to 100 Mb/sec which is the capacity of the network interface cards on the IPMON systems. At this rate the raw data would take 26.4 hours to transfer from the IPMON systems to the tape library. To improve transfer time and decrease the storage capacity requirements, the trace data is compressed before being transferred back to the lab. Using standard compression tools such as *gzip*, we are able to achieve compression ratios ranging from 2:1 to 3:1 depending on the particular trace characteristics. This reduces the transfer time to about 12 hours.

This transfer time presents another difficulty when exhaustively monitoring a network for operational purposes. An alternative solution would be to avoid the data repository and perform the analysis on the monitoring systems themselves. This is a good solution if there is a single type of analysis that is being performed on the traces. However, the data is used for many research projects, and some of the analysis performed in several projects requires multiple iterations through the trace. In addition, we would like to keep an archive of the collected data so that it may be used for future projects.

C. Analysis Platform

All data analysis is performed off-line by a cluster of 16 Linux PCs. Some of the data analysis, such as measuring packet size distributions, could be performed on-line but others, such as measuring network delay, cannot. Measuring network delay requires identifying a packet on multiple traces. This involves exchanging significant amounts of data between two (or more) IPMON systems for every packet whose delay is measured. This could be accomplished on a local network if both systems are located in the same POP, but it would significantly increase the network load when processing data collected at multiple POPs. Since we do not want to perturb the network during trace collection, it is necessary to perform the analysis off-line.

There are two categories of analysis that are performed by the analysis platform:

- Single trace analysis involves processing data from a single link to measure traffic characteristics. This type of analysis includes, for example, determining packet size distributions, flow size distributions, and determining what types of applications are using different links. To efficiently perform this type of analysis, traces from different links are loaded onto separate PCs in the cluster and processed in parallel.
- Multi-trace analysis involves correlating traffic measurements among different links. This includes performing delay measurements and looking at round trip TCP behavior. This type of analysis is performed by dividing each trace into several time segments and loading the different time segments onto different machines. For example, PC #1 might contain the first 30 minutes from a set of 5 traces, and PC #2 might contain the next 30 minutes of those same 5 traces.

One key requirement when performing multi-trace analysis is to be able to identify an individual packet as it travels across multiple links in the network. The only two pieces of information that should change as a packet travels through the network are the TTL and checksum fields in the IP header. By comparing the remaining 41 bytes of data we collect for each packet, we can identify a packet at multiple locations in the network. However, it is possible for two different packets to have the same 41 bytes. In theory this should happen infrequently since the ID field for each packet generated by a particular source should be unique. However, in the traces we collect we do observe duplicate packets due to systems generating incorrect IP id fields or due to link layer retransmissions, but these packets only represent .001% to 0.1% of the total traffic volume. In these cases, we typically ignore all packets which have duplicate values.

Fig. 3. Measurement Configuration

V. Measurement Results

In this section we present a sample of measurement results to demonstrate the types of data that can be collected using the IPMON measurement facilities. The presented data are not intended to make any generalizable statement on the nature of the traffic on an IP backbone. The intent is to validate our monitoring infrastructure and to demonstrate its capabilities using a few simple trace analyses.

For brevity, we present data from only two of the nine bi-directional monitored links. Both links are connected to the same core router in one POP within the Sprint IP backbone network. The trace *web-out* was collected on a link from the backbone router to an access router connected to a web hosting company, and *web-in* is the link from the access router to the backbone router. The *peer-out* trace was collected on a link from the backbone router to a peering point, and *peer-in* is the link from the peering point to the backbone router. Both the peering link and the web hosting link are OC-3 links. Figure 3 shows a diagram of the monitored links. It is important to note that there are other, unmonitored, links which are also connected to the backbone router. We do not exhaustively monitor all links in the POP. Table III provides the trace start times, trace end times, and trace sizes.

A. Workload Characterization

First we present the general characteristics of the traces. Figures 4 and 5 plot link utilization averaged over one minute periods. Figure 6 shows the application mix on the *web-out* trace. From these figures we make several observations:

- As reported in many other studies, the dominant source of traffic is http. We only present the results from the *web-out* link, but the results are similar on the other links.

Link	Start Time (PST)	End Time (PST)	Number of Packets (millions)
web-out	9:56, Wed, 8/9/2000	19:57, Wed, 8/9/2000	568
web-in	9:56, Wed, 8/9/2000	23:18, Wed, 8/9/2000	852
peer-out	9:56, Wed, 8/9/2000	9:55, Thurs, 8/10/2000	816
peer-in	9:56, Wed, 8/9/2000	9:55, Thurs, 8/10/2000	794

TABLE III
TRACE STATISTICS

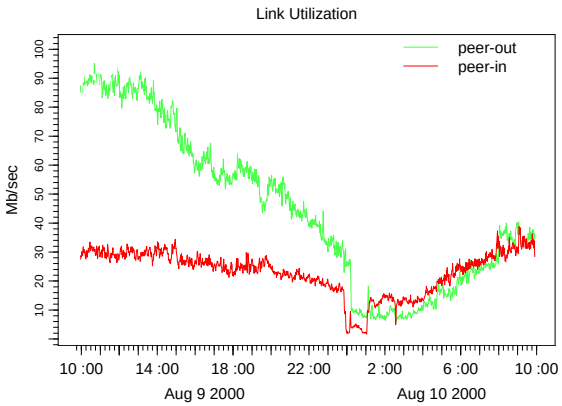

Fig. 4. Peering link utilization in Mb/sec

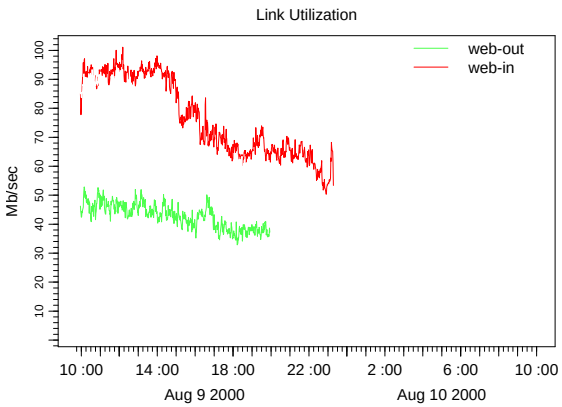

Fig. 5. Web link utilization in Mb/sec

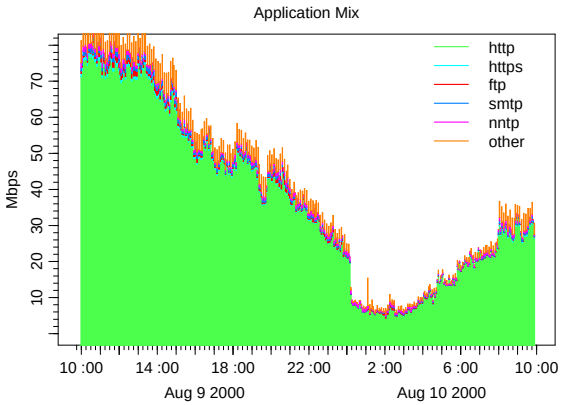

Fig. 6. *web-out* traffic breakdown by application

- The link utilization on the peering link changes dramatically (from nearly 100 Mb/sec to under 10Mb/sec) over a 24 hour period.
- Link utilization is not symmetric for either the peering link or the web hosting link. While this is expected for the web hosting traffic, as the web servers should generate much more data than they receive, the traffic was expected to be more symmetric on the peering point. All of the links we monitor exhibit such asymmetric characteristics. It may be possible to take advantage of this fact when allocating disk space for the traces if a single system is used to monitor both directions of a link.
- Link utilization is typically under 50% with peaks reaching just over 60%. The data we present represents two of the most heavily utilized monitored links. The drop in link utilization around midnight corresponds to a maintenance period.

Another point to note is that the web link trace reflects the limited scalability of our monitoring system. The reason why the trace in Figure 5 is truncated around midnight is that there was insufficient disk space. In an effort to increase the number of monitored links, we configured one IPMON system to monitor both directions of the web link, and tried to optimize the disk allocation to take advantage of the traffic asymmetries. However, this was not successful. The disk space on a single IPMON system is not sufficient to capture a full 24 hour trace for both of these links. Later traces collected on the web link used one IPMON system for each direction.

Other traffic characteristics are summarized in Table IV. This table shows the number of TCP/UDP/other packets; the minimum, average, and maximum packet sizes; the number of packets which were IP fragments; and the number of packets with IP and TCP options. The number of packets with IP and TCP options affect the amount of information that is provided by the trace data. If a packet contains either type of option, then the size of the TCP/IP header can exceed the 44 bytes of data we collect. If this is the case, we may lose information about the TCP port numbers, sequence numbers, or flags. As can be seen in the table, the number of packets which contain IP options is less than .0004% of the total traffic volume. The number of packets with TCP options, on the other hand, can be up to 12% of the total traffic. These options are used to reconfigure the TCP maximum segment size and perform MTU negotiation. In these packets, the only part of the header which we do not capture is the TCP options. We are able to record the source/destination ports, the TCP

trace	TCP packets	UDP packets	Other packets	min packet size	average packet size	max packet size	IP fragments	IP options	TCP options
web-out	530 million	33 million	4 million	20	339	1500	57,549	2068	67 million
web-in	806 million	34 million	13 million	20	540	1500	420,269	1548	66 million
peer-out	740 million	106 million	7 million	20	590	1500	292,450	828	79 million
peer-in	635 million	143 million	14 million	20	315	63,945	164,924	2915	67 million

TABLE IV
TRACE STATISTICS

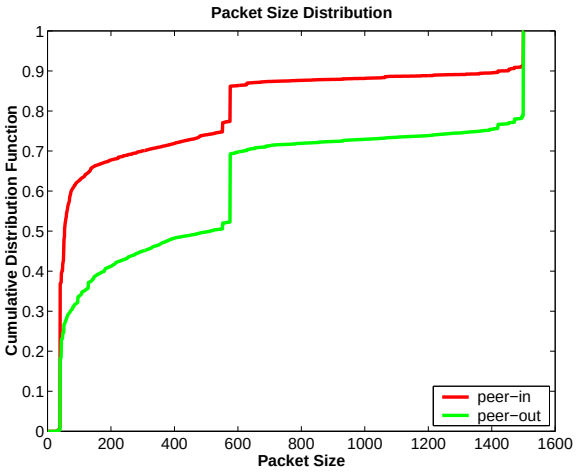

Fig. 7. packet size distribution on peering link

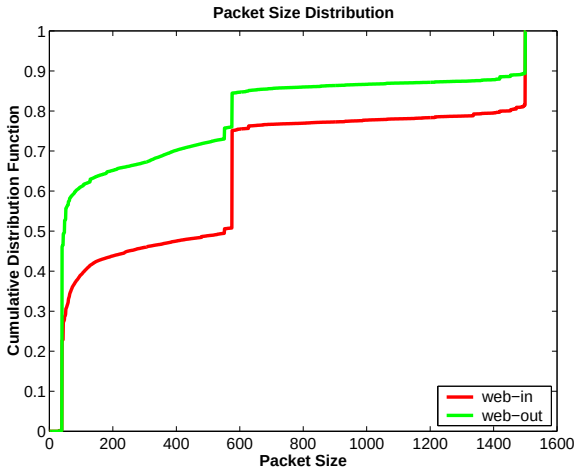

Fig. 8. packet size distribution on web link

sequence numbers, and flags.

B. *Packet size analysis*

The packet size characteristics of a link impacts two system design parameters:

- the duration of the trace that may be collected
- the rate at which the IPMON systems must process incoming packets

The IPMON systems record 64 bytes for each packet. Therefore, the duration of the trace is limited to a particular number of packets. If two links are running at the same link utilization, the system monitoring the link with the higher average packet size will be able to record a longer trace.

The packet size distribution we observe follows the same tri-modal distribution as observed in other studies [21][2][20]. The cumulative distribution function of the packet size distribution is shown in Figures 7 and 8. The packet size distribution has peaks at 40 bytes (minimum size TCP packets, 1500 bytes (maximum size Ethernet packets), and at 552 and 576 bytes (maximum size TCP packets from TCP implementations which do not perform MTU discovery). The minimum, average, and maximum packet sizes for each link are shown in table IV. The packet size distribution on the web links explains the reason for the discrepancy in trace durations. Each trace was configured to use an amount of disk space proportional to the link utilization (i.e. the *web-in* link was configured with approximately twice the capacity of the *web-out* link). The goal was to collect the same duration of trace from each link. However, as can be seen from the packet size distributions, the *web-out* link (which is the output link from the network to the web server customer) carries a large number of small packets containing web requests and acknowledgements, while the *web-in* link carries a large number of maximum size packets containing web data. Therefore, when determining the amount of disk space required to collect the traces, the traffic volume in terms of packets per second is a better measure than the traffic volume in terms of bits per second.

The traffic volume in terms of packets/sec are shown in Figures 9 and 10. These values are computed over one minute averages. The figures demonstrate that the data rate in packets per second on the *web-in* link is nearly the same as the data rate on the *web-out* link, rather than twice the data rate as predicted by the bits per second data rate.

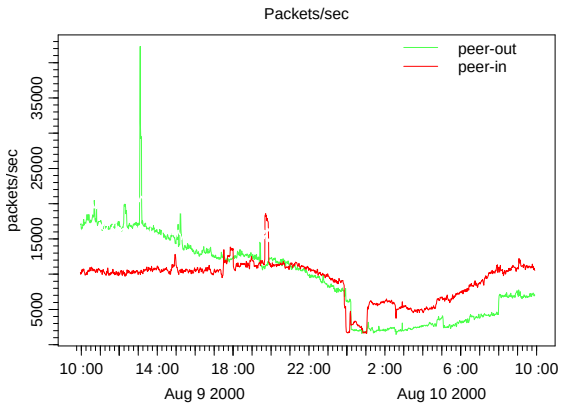

Fig. 9. Peer link traffic volume in packets/sec

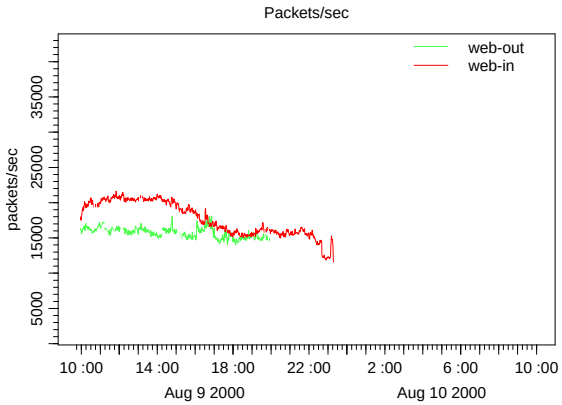

Fig. 10. Web link traffic volume in packets/sec

The figures also indicate how the packet size characteristics affect the data rate requirements of the IPMON systems. Figure 9 shows several peaks in the traffic volume in packets/sec on the peering point links. These peaks, however, do not correspond to equivalent peaks in overall traffic volume in bytes (i.e. there is no equivalent peak observed in Figure 4). The peaks correspond to bursts of small packets that occur in the network. In this case, the peaks actually represent a large number of SYN packets which are all transmitted to one particular destination, a common denial-of-service attack.

Regardless of the source of the traffic, a sequential arrival of small packets imposes a performance burden on the IPMON system. To evaluate this burden we count the number of packets of similar sizes which arrive sequentially (e.g. the number of 40 byte packets that arrive back-to-back) and plot the distribution in Figure 11. We categorize packets into three classes: small, medium, and large. Small packets are less than 500 bytes, medium packets are between 500 and 1000 bytes, and large packets are longer than 1000 bytes. Sequential arrivals of medium and large packets are similar on both *peer-in* and *peer-out* links. For small packets, the number of sequential arrivals can be quite large, and in the case of *peer-out*, even reach 599.

The number of sequential arrivals, however, does not tell how close in time the packets arrive. To capture the temporal aspects of packet bursts, we examine the peak arrival rate in packets per second on the monitored links shown in Figure 12. This figure shows the peak traffic rate (in packets/sec) at time scales ranging from 10 ms to 1 sec. The data is generated by computing the average arrival rate over 10 ms intervals for the entire trace. Using this data we compute the peak arrival rate observed over any of the 10 ms intervals. We then evaluate the peak rate at a range of time intervals ranging from 10 ms to 1 sec. Even at the smallest time scale, 10 msec, the peak arrival rate is only 231,000 packets/sec, well within 1.94 million packets/sec supported by the IPMON systems.

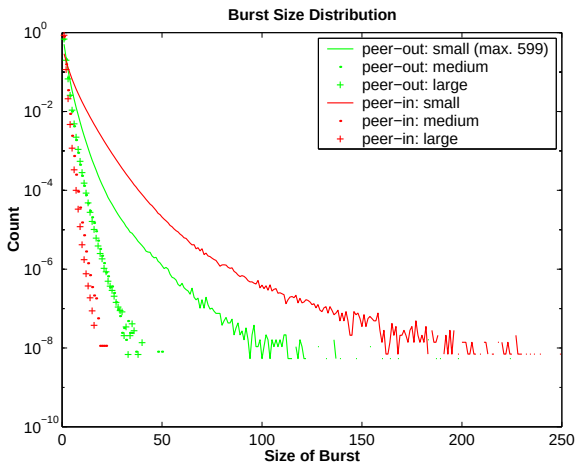

Fig. 11. Sequential packet arrivals on peer link

The peak rate determines the data rate required to copy traffic data from the DAG cards to main memory in the system. The packet sizes also affect the data rate required to store the data to disk. The systems were originally designed to support traffic with an average packet size of 400 bytes, which does correspond to the average packet size across all the links. However, from our measurements, the average packet size on a single link can be closer to 300 bytes. The disks on the OC-3 and OC-12 monitors have enough bandwidth to support the smaller average packet size, but the OC-48 monitors can only support up to 1.9 Gb/sec of traffic if the average packet size is 300 bytes. Fortunately, the OC-48 links we plan to monitor are not run at full link utilization. While there may be bursts of traffic which increase link utilization to 100%, Figure 12 indicate that these types of bursts only occur at small time scales on OC-3 links, and similar behavior is expected on the OC-48 links. The OC-48 systems are configured with a 512 MB memory buffer which can buffer eight seconds of data, and should be able to accommodate bursts which may occur.

C. Delay Measurements

One of the unique aspects of the IPMON system is its capability to measure network delays for actual network traffic. Most current delay measurements are performed using a set of probe packets which are transmitted at periodic or random time intervals. While these systems are able

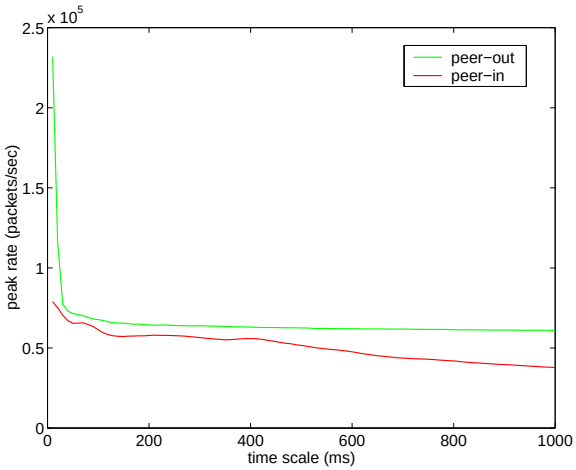

Fig. 12. Peak rate for peering link

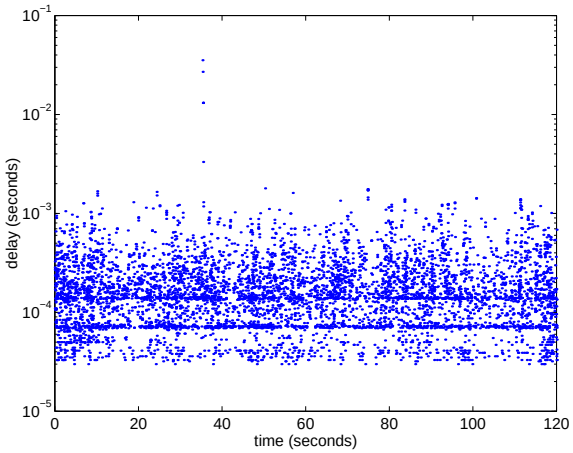

Fig. 13. Delay from *web-in* to *peer-out*

to provide a general idea about the delay performance of the probe packets, it is difficult to determine if the performance of the probe packets represent an accurate sampling of the delays experienced by actual network traffic.

The IPMON systems allow us to measure the delays seen by every packet that is transmitted between two points in the network. This is accomplished by identifying the same packet in two different traces and computing the difference in the timestamps. Figure 13 shows a two minute sample data set. The x-axis shows the packet arrival time, and the y-axis shows the delay experienced by the packet for traffic between the *web-in* and *peer-out* links. Since both links are located on the same core router, this data represents the single hop delay experienced by packets in the backbone. We currently have systems installed only in one location in the network, but additional systems are in the process of being installed. With data from these systems, we will be able to measure delays across many hops in the backbone.

There are several points to note from this figure. First, the minimum delay across the entire interval remains almost constant, around 30 μs. This is the smallest delay interval that it is possible to measure in the network, so the 5 μs error that may be introduced by the clock synchronization mechanism is acceptable. Second, there is a rather large increase in delay at time 30 seconds. The delay experienced increases to nearly 30 ms, which is unusually large for a single hop delay. This type of delay is also observed at a small number of points elsewhere in the trace. These excessive delays are the types of data that are difficult to observe using probe traffic. The long delays are experienced only by a small number of packets, but the impact on these packets is extremely large. The source of these long delays is currently under investigation, but it is believed to be due to a pathologic behavior of the router rather than actual queuing delays.

VI. Conclusion

We describe the Sprint IP Monitoring system, a passive monitoring system that collects packet-level traces from the Sprint Internet IP backbone. The systems are capable of supporting OC-3, OC-12, OC-48 data rates, and are synchronized to within 5 μs using a stratum-1 GPS reference clock. We present the system design and demonstrate the performance of the system with several sample measurements.

The advantage of our system is it provides the capability to collect traces from multiple locations in the network and correlate the traces through highly accurate timestamps. This provides the capability to study both single link characteristics (e.g. workload and packet size distributions), as well as characteristics which require data from multiple links (e.g. delay, TCP behavior, network provisioning). It is also very flexible in that the data is not targeted towards a single use. The packet traces are useful in many diverse research projects.

The disadvantage of the system is that the amount of data collected is very large. Data from a single 24 hour period exceeds 3.3 TB. This requires both a large amount of resources to be installed in network facilities for data collection purposes and a large amount of resources to perform data analysis. While our system supports monitoring 31 different links and can be extended to monitor several dozen additional links, scaling the system to exhaustively monitor the entire network is impractical.

In future work we plan to analyze in depth the traffic observed on various links on the network. These results will be used to:

- Design provisioning and dimensioning tools to better anticipate customers needs and increase customer satisfaction, eventually making it possible to provide various classes of service.
- Gain a better undestanding of the traffic characteristics on an IP backbone, and design more accurate traffic models.
- Work with router designers to design embedded measurement facilities.

Acknowledgements

In the course of this project we received help from so many people that it is impossible to acknowledge them individually. Instead we choose to acknowledge their institu-

tions: SprintLink, Sprint ISC, Sprint ATL, the University of Waikato, and CAIDA.

References

[1] J. Apsidorf, "OC3MON: Flexible, affordable, high performance statistics collection," in *Proceedings of INET*, June 1997.

[2] KC Claffy, Greg Miller, and Kevin Thompson, "The nature of the beast: Recent traffic measurements from an internet backbone," in *Proc. INET'98*, Geneva, Switzerland, July 1998.

[3] A. K. Adams and M. Mathis, "A system for flexible network performance measurement," in *Proceedings of INET 2000*, June 2000.

[4] "Mtrace ftp site," ftp://ftp.parc.xerox.com/pub/net-research-/ipmulti.

[5] "Treno web page," http://www.psc.edu/networking/treno_info-.html.

[6] R. Cáceres, N. Duffield, D. Towsley, and J. Horowitz, "Multicast--based inference of network-internal loss characteristics," *IEEE Transactions on Information Theory*, vol. 45, no. 7, pp. 2462–2480, November 1999.

[7] S. Kalidindi and M. J. Zekauskas, "Surveyor: An infrastructure for internet performance measurements," in *Proceedings of INET '99*, June 1999.

[8] T. McGregor, H.-W. Braun, and J. Brown, "The NLANR network analysis infrastructure," *IEEE Communications*, vol. 38, no. 5, May 2000.

[9] W. Matthews and L. Cottrel, "The PingER project: Active internet performance monitoring for the HENP community," *IEEE Communications*, vol. 38, no. 5, pp. 130–136, May 2000.

[10] "MIQ ratings methodology," http://ratings.miq.net/method.html.

[11] "The interaction of web content and internet backbone performance," http://www.keynote.com/services/html/wp-_compdata.html, Keynote white paper.

[12] William Stallings, *SNMP, SNMPv2, SNMPv3, and RMON 1 and 2*, Addison Wesley, 3rd edition, 1999.

[13] "Tcpdump web page," http://ee.lbl.gov.

[14] "NetFlow services and applications," http://www.cisco.com/-warp/public/cc/pd/iosw/ioft/neflct/tech/napps_wp.htm, 2000, Cisco white paper.

[15] "CoralReef website," http://www.caida.org/tools/-measurement/coralreef.

[16] C. Labovitz, A. Ahuja, and F. Jahanian, "Experimental study of internet instability and wide-area backbone failures," in *Proceedings of the 29th International Symposium on Fault-Tolerant Computing (FTSC-29)*, Madison, Wisconsin, June 1999.

[17] V. Paxson, G. Almes, J. Mahdavi, and M. Mathis, "Framework for IP performance metrics," RFC 2330, IETF, May 1998.

[18] R. Cáceres, N. Duffield, A. Feldmann, J.D. Friedmann, A. Greenberg, R. Greer, T. Johnson, C.R. Kalmanek, B. Krishnamurthy, D. Lavelle, P.P. Mishra, J. Rexford, K.K. Ramakrishnan, F.D. True, and J.E. van der Merwe, "Measurement and analysis of IP network usage and behavior," *IEEE Communication*, vol. 38, no. 5, May 2000.

[19] V. Paxon, "Empirically-derived analytic models of wide-area tcp connections," *IEEE/ACM Trans. on Networking*, vol. 2, no. 4, August 1994.

[20] S. McCreary and K.C. Claffy, "Trends in wide area ip traffic patterns," in *ITC Specialist Seminar*, Monterey, California, May 2000.

[21] K. Thompson, G. Miller, and R. Wilder, "Wide area internet traffic patterns and characteristics," *IEEE Network*, Nov 1997.

[22] W. Simpson, "PPP in HDLC-like framing," rfc 1662, IETF, July 1994.

[23] J. Cleary, S. Donnelly, I. Graham, A. McGregor, and M. Pearson, "Design principles for accurate passive measurement," in *PAM 2000*, Hamilton, New Zealand, April 2000.

[24] "Dag 4 SONET network interface," http://dag.cs.waikato.ac.nz-/dag/dag4-arch.html.

[25] "Dag synchronization and timestamping," http://dag.cs-.waikato.ac.nz/dag/docs/dagduck_v2.1.pdf.