Packet-Level Traffic Measurements from the Sprint IP Backbone

Chuck Fraleigh, NetVMG
Sue Moon, KAIST
Bryan Lyles, Independent Consultant
Chase Cotton, Mujahid Khan, Deb Moll, Rob Rockell, and Ted Seely, Sprint
Christophe Diot, Intel Research

Abstract

Network traffic measurements provide essential data for networking research and network management. In this article we describe a passive monitoring system designed to capture GPS synchronized packet-level traffic measurements on OC-3, OC-12, and OC-48 links. Our system is deployed in four POPs in the Sprint IP backbone. Measurement data is stored on a 10 Tbyte storage area network and analyzed on a computing cluster. We present a set of results to both demonstrate the strength of the system and identify recent changes in Internet traffic characteristics. The results include traffic workload, analyses of TCP flow round-trip times, out-of-sequence packet rates, and packet delay. We also show that some links no longer carry Web traffic as their dominant component to the benefit of file sharing and media streaming. On most links we monitored, TCP flows exhibit low out-of-sequence packet rates, and backbone delays are dominated by the speed of light.

Overprovisioning is widely used by packet network engineering teams to protect networks against network element failure and support the rapid growth of traffic volume. So far, this approach has been successful in maintaining simple, scalable, highly available, and robust networks. It is important to realize that in packet networks which do not perform call admission control, there is often no way to control the amount or types of traffic entering the network. The provisioning problem therefore lies in figuring out how much excess capacity is required to provide robustness (e.g., resilience to multiple simultaneous link failures) and scalability. The current tools for network management, such as Simple Network Management Protocol (SNMP), are limited in their capabilities, since they only provide highly aggregated statistics about the traffic (e.g., average traffic load over five-minute intervals) and do not give insight into traffic dynamics on timescales appropriate for events such as packet drops. Another example is the demand traffic matrix, which is a crucial input to many network planning, provisioning, and engineering problems, but is difficult to obtain with available tools [1, 2].

Detailed traffic measurements are necessary to assess the capacity requirements and efficiently engineer the network. Research topics that can benefit from packet-level monitoring are:

- Developing traffic models that allow network operators to determine the amount of overprovisioning required in their network [3]
- Assessing the trade-offs between different levels of granularity in routing, and studying the traffic dynamics between POPs [2, 4]
- Developing algorithms to detect network anomalies such as denial-of-service attacks and routing loops [5]
- Studying the performance of TCP, and identifying where congestion is occurring in the network [6]
- Evaluating the network's capability to support new value-added services such as telephony and quality of service (QoS) [7]

In order to gain better insight into network traffic, we have developed the IP Monitoring (IPMON) system and have deployed it in the Sprint IP backbone network. The IPMON system is capable of:

- Collecting packet-level traces at multiple points on the Sprint IP backbone for link speeds of up to OC-48 (2.5 Gb/s)
- Marking each of the packets with a submicrosecond timestamp
- Synchronizing these traces to within 5 μs

Offline processing of the packet traces then enables detailed studies of the various aspects of traffic characteristics, such as delay and loss.

In this article we first describe the architecture and capabilities of the IPMON system. Then we point out the challenges we faced in collecting terabytes of data, and include our solutions to data sanitization. In the remainder of the article we

This work was done while Chuck Fraleigh, Sue Moon, Bryan Lyles, and Christophe Diot were at Sprint Advanced Technology Laboratory (ATL), Burlingame, California.

[1] *www.sprint.net*

0890-8044/03/$17.00 © 2003 IEEE

present our observations of traffic on OC-12 (622 Mb/s) and OC-48 links in the Sprint IP backbone network.[1]

Results presented in this article provide a high-level view of a major backbone network's traffic in 2001 and 2002, and highlight the changes that have occurred in traffic characteristics with respect to previous studies. First, we illustrate that SNMP statistics are not appropriate to detect short-term congestion. Then we identify the impact of new applications such as distributed file sharing and streaming media: on some links over 60 percent of the traffic is generated by these new applications, while only 30 percent is Web traffic. Our results on end-to-end loss and round-trip-time (RTT) performance of TCP connections are significantly different from previous observations. Lastly, we present results on the network delays experienced through a single router in the backbone, as well as the U.S. transcontinental delay measurement. Our findings are that packets experience very little queuing delay and insignificant jitter in the backbone.

The article is organized as follows. We discuss related work. Then we describe the monitoring system architecture. We present and analyze traffic measurements from the Sprint IP backbone network. This starts with a brief description of the 32 traces used in the article, and analyzes the traffic load broken into bytes, applications, and numbers of flows. The performance of TCP flows is evaluated in terms of RTTs and out-of-sequence packet rates. Lastly, delay measurements are presented. We then conclude the article and discuss future work.

Related Work

The challenges in designing a monitoring system for a comprehensive view of network performance are:

- The collection of detailed traffic statistics, including application mixes and traffic matrices, from heterogeneous network links
- Limiting the side-effects of the monitoring system on the monitored network
- Obtaining a global view of the monitored network from a limited number of monitoring sites

Existing monitoring systems partially address these three issues.

Network researchers have adopted two distinct approaches to data collection. The first approach uses an *active* measurement system to inject probe traffic into the network and then extrapolate the performance of the network from the performance of the injected traffic. The second approach is that of passively observing and recording network traffic. These *passive* measurement systems use the recorded traffic to characterize both the applications and the network performance. They record and archive full traces, which in turn can be later used for further analysis. One drawback is that they generate a large amount of measurement data. Due to the quantity of data produced, recording traces from very-high-bandwidth links is a serious challenge [8]. As a result, global observations have often been addressed by inference techniques, not by exhaustive passive monitoring of every link in a network.

OC3MON is a well-known passive monitoring system for OC-3 links (155 Mb/s) described in [9]. It collects packet-level traces or flow-level statistics. Packet-level traces can be collected only for a limited amount of time (a few minutes at a time), while flow-level statistics can be collected on a continuous basis. It has been deployed at two locations in the MCI backbone network to investigate daily and weekly variations in traffic volume, packet size distribution, and traffic composition in terms of protocols and applications [10]. OC3MON has now been extended to support OC-12 and OC-48 links[2] [11]. Passive monitoring systems require specific hardware to collect data on the network. In the case of OC3MON, data capture relies on tapping the fiber through a dedicated network interface card.

There are several projects that combine both active and passive measurement. The NetScope project [12] collects measurements from the AT&T network in order to study the effects of changing network routes and router configuration. Using NetFlow measurements from routers, the traffic demand for the entire network is derived [13]. The traffic demand is used in simulation to determine the effects of changing the network configuration. As part of an ongoing effort to develop better network measurement tools, a passive monitoring system called PacketScope has been developed and used to collect and filter packet-level information.

The Network Analysis Infrastructure (NAI) project measures the performance of the vBNS and Abilene networks. This system collects packet traces, active measurements of round-trip delay and loss, and Border Gateway Protocol (BGP) routing information. All of the 90 s packet traces from this project are available on their Web site.[3]

Some routers have built-in monitoring capabilities. Cisco routers have NetFlow [14]. It collects information about every TCP and UDP flow on a link. Juniper routers have a set of accounting tools to collect similar statistics as NetFlow [15]. There are other standalone commercial products for passive monitoring, such as Niksun's NetDetector and NetScout's ATM Probes. These systems, however, are limited to OC-3 or lower link speeds, and are thus inadequate for Internet backbone links.

Our monitoring infrastructure, called IPMON, is similar to the OC3MON system, but with extended capabilities that allow it to collect packet traces at up to OC-48 link speeds (2.48 Gb/s) for a period of at least several hours. The range of observable metrics is wider than with the above systems thanks to timestamps synchronized to within 5 μs of a global clock signal. In the next section we describe and discuss the IPMON components in greater detail.

IPMON Architecture and Features

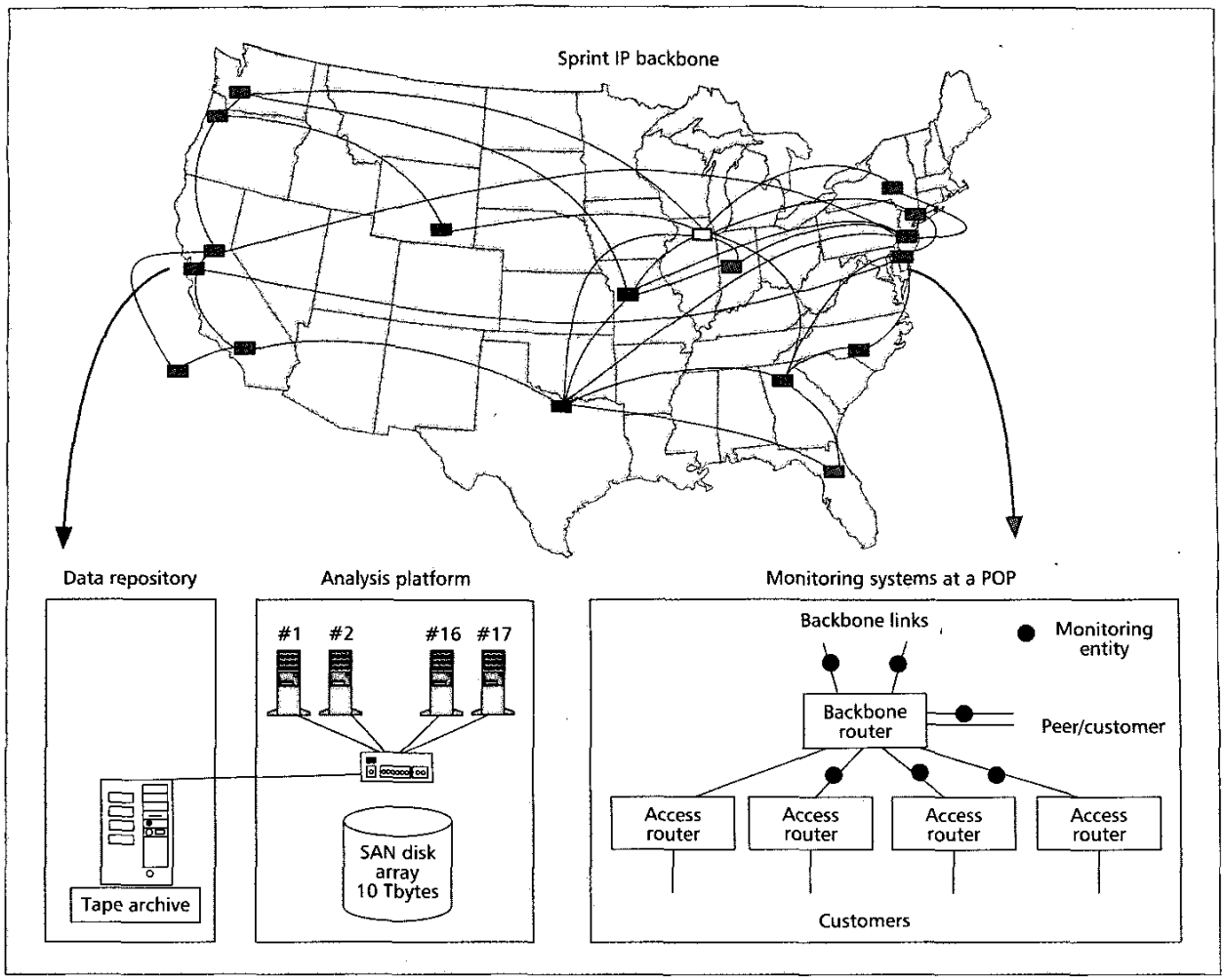
In this section we present the architecture of the Sprint IP backbone network and then give a high-level description of our passive monitoring system. We close the section with a brief summary of practical concerns in trace collection.

The topology of a tier 1 Internet backbone typically consists of a set of nodes known as points of presence (POPs) connected by high-bandwidth OC-48 (2.5 Gb/s) and OC-192 (10 Gb/s) links. From each POP, links radiate out toward customers, such as large corporate networks, regional tier 2 Internet service providers (ISPs), digital subscriber line (DSL) aggregation devices, and large server farms, that typically require higher-bandwidth network connections.[4] Each POP may have links, known as *private peering points*, to other backbone networks as well as links to public network access points

[2] *The analysis results from two one-hour-long OC-48 traces are available at http://www.caida.org.*

[3] *http://moat.nlanr.net/PMA/*

[4] *Lower-bandwidth customers, such as dialup home users, connect to tier 2 ISPs that in turn connect to the backbone network.*



■ Figure 1. *The IPMON system in the Sprint IP backbone.*

(NAPs). Because of traffic volume, major backbone networks often have peering links in multiple geographically distinct POPs.

The Sprint IP backbone consists of approximately 40 POPs worldwide, of which 18 are located in the United States. Figure 1 shows an abstract view of the Sprint U.S. backbone topology. Within a POP, the network has a two-level hierarchical structure: access (edge or gateway) and backbone (or core) routers. Customer links are connected to access aggregation routers. The access routers are in turn connected to the backbone routers. These backbone routers provide connectivity to other POPs, and connect to public and private peering points. The backbone links that interconnect the POPs have the speed of OC-48 or OC-192. Sprint uses packet over synchronous optical network (SONET) (POS) framing, which in turn runs over Sprint's dense wavelength-division multiplexing (DWDM) optical network.

The IPMON Monitoring Infrastructure

In this section we give a short description of the IPMON architecture.[5] IPMON consists of three elements (Fig. 1): a set of passive monitoring entities that collect packet traces; a data repository that stores the traces once they have been collected; and an analysis platform that performs offline analysis. Analysis is performed offline for two reasons. The primary reason is that the data is used in many different research projects, each of which has its own set of custom analysis tools. It is more efficient to perform the multiple types of analysis on a computing cluster in the laboratory where many systems can access the data simultaneously. The second reason is that we archive the traces for use in future projects.

Monitoring Entities — The monitoring entities are responsible for collecting the packet traces. Each trace is a sequence of packet records that contain the first 40 bytes of each packet, which are just the IP and UDP/TCP headers, as well as a submicrosecond timestamp that indicates the time at which the packet was observed. The source and destination IP addresses are not anonymized, since they are needed in routing-related analysis.

Each monitoring entity is a dual-processor Linux server (Dell PowerEdge 6000 series) with 1 GB main memory, a large disk array (100–330 Gbytes), and a POS network interface card, known as the DAG card [17]. Existing DAG cards are capable of monitoring links ranging in speed from OC-3 to OC-48. An OC-192 monitoring card is under development [8]. The DAG card captures, timestamps, and transfers the POS HDLC framing information and the IP packet headers to the main memory of the Linux server where a driver soft-

[5] *A detailed description of the monitoring infrastructure is provided in [16].*

ware then transfers the data to the disk array. An optical splitter is installed on the monitored link, and one output of the splitter is connected to the DAG card in the server. This is a receive-only connection; the DAG card does not have the capability of injecting data into the network. Since a receive-only passive optical splitter is used, failure or misbehavior of the monitoring entity or DAG card cannot compromise network integrity.

Each monitoring entity has a removable disk array of up to 330 Gbytes. This amount of disk space allows us to capture a minimum of several hours of trace data at full link utilization. We can either schedule trace collection for a predefined interval or allow it to run until space on the hard disks is exhausted. By Sprint engineering design, the network links are not fully loaded (except in extreme failure scenarios), and we are typically able to collect several days of measurement data.

The packet timestamps are generated by an embedded clock on the DAG card that is synchronized to an external Global Positioning System (GPS) signal. GPS is a satellite-based system that provides global time information with an accuracy of 20 ns. Hardware errors as well as other system related issues bring the maximum error on timestamps to 5 μs [16, 17]. This synchronization ability allows us to measure one-way network delay between two monitored links.

A total of 60 monitoring entities are installed at four different POPs, chosen on the basis of geographic diversity and connectivity. They monitor the traffic on OC-3, OC-12, and OC-48 links that connect access routers, backbone routers, and several private peering links.

Data Repository — The data repository involves two levels of storage, consisting of a 12 Tbyte removable tape library and a 10 Tbyte disk storage array. It is located at the Sprint Advanced Technology Laboratory (ATL). For short traces, a dedicated OC-3 link is available for transferring the data from the monitoring entities back to the ATL. Given that a full multi-POP trace set consists of approximately 10 Tbytes when trace collection is allowed to run until the disks fill up, the best method for transferring full traces back to the data repository is by physically shipping the removable hard disks. As a result of these constraints on transferring trace data, we do not schedule new traces until the previous trace data is either transferred or deleted.

Data Analysis Platform — Data analysis is performed on a cluster of 17 high-end servers connected to a storage area network (SAN) with a capacity of 10 Tbytes. Two categories of analysis are performed on the platform.

Single trace analysis involves processing data from a single link. This type of analysis includes, but is not limited to, determining packet size distributions, flow size distributions, and the amount of bandwidth consumed by different applications. In this work we define a flow by the 5-tuple {protocol type, source IP address, source port, destination IP address, destination port}.

Multiple trace analysis involves correlating traffic measurements from different links. This includes calculating delay and identifying packet losses. The key to performing multiple trace analysis is to identify an individual packet as it travels across multiple links in the network. To identify a packet we use 30 out of the 40 bytes of header information that provide unique identification of packets. These 30 bytes include the source and destination IP addresses, the IP header identification number, and possibly TCP and UDP header information (TCP and UDP information may not be available due to the use of IP options). Other fields, such as the IP version and checksum, are not used since they are identical in most IP packets or, in the case of the checksum, provide redundant information. To match packets on multiple links we use a hash-based search algorithm to determine if the same packet is observed in multiple traces [18].

The following three sets of analysis tools are most commonly used:

- The first set of tools is a set of custom tools that extracts information about individual flows from a single trace. These tools process an entire trace and return a list of flows, their start time, end time, and details about each packet in the flow.
- The second set of tools is the CoralReef public suite and custom tools that we use to identify the amount of traffic generated by different protocols (e.g., TCP, UDP) and applications (e.g., Web, email, media streaming) [19].
- The third set of tools is used for multiple trace correlation. These tools use a hash-based algorithm to find packets that have been recorded on multiple links and return a list of these packets and the time at which they were observed on each link.

Trace Sanitization

Trace collection is a complex process, and traces can be corrupted at any step of the process:

- The monitoring entities can fail. Problems range from operating systems to hardware failures. Any of these problems can potentially affect trace consistency. Hard disk failures are the most common in our experience.
- Hardware or software bugs of the DAG card have impacted traces. For example, we have observed traces where packets were missing or had sequences of zeroes. Misalignment or byte swapping has also been a problem.
- While they are being transferred from the collection site to the analysis platform, traces can get corrupted or truncated due to intermediate system failures (local disk failure, defective tapes, etc.).

We realized from the very first trace collection the need for trace sanitization. As we discover and fix sources of corruption, we have steadily improved the process. Sanitization has been established as a systematic process that is run on every trace before it is used in an analysis. The current steps in the sanitization process are described below. We understand that the list of sources of corruption is not exhaustive, and continues to grow, though slowly.

- We first check the hard disks on which the traces are stored for bad blocks and access problems.
- We analyze the DAG card log. While collecting a trace, the DAG card keeps track of GPS synchronization and increments a counter any time it misses a packet.
- We process the POS HDLC header and verify the consistency of each packet based on information, such as packet type. We then check that the structure of the packet is correct for the packet type.
- We check that the timestamps are monotonically increasing, that the interpacket time is greater than the time required to transmit the previous packet, and that any gaps in the trace are reasonable.[6]
- We detect traces out of GPS synchronization by calculating the delay between traces. If the minimum delay per minute

[6] *On OC-3 to OC-48 links it is extremely unlikely to have no packet in any interval of 100 ms. A long gap is often an indication of a clock synchronization problem.*

between two traces fluctuates more than a few milliseconds, those two traces are considered out of synchronization.

Any time a problem is detected, the corresponding trace is ignored. Only those traces that are sanitized per process described above are used in analysis.

Measurement Results

In this section we present measurement results to demonstrate the capabilities of the IPMON system and provide information on the characteristics of the backbone traffic in 2001 and 2002. The results are organized in three categories. First, we present traffic workload statistics (e.g., application mix, packet size distribution, flow size distribution). These results are not unique to our measurement system; they can be obtained using flow-level measurement systems such as NetFlow or CoralReef.[7] However, these results are the first published traffic statistics from a large number of OC-12 and OC-48 links in a production backbone network, and they show the impact of emerging applications such as distributed file sharing and streaming media. The second category of results is on TCP performance statistics. These results demonstrate the advantages of collecting packet-level measurements. The third set of results is packet delay measurements through a single backbone router and over a U.S. transcontinental path.

Trace Description

The IPMON system collects measurements from about 30 bidirectional links at four POPs out of about 5000 links in the Sprint IP backbone. Three POPs are located on the east coast of the United States, and one POP on the west coast. The OC-48 links we monitor are all long-haul transcontinental connections. The other links either connect backbone routers to access routers within the POP, or connect peers and customers to the backbone as in Fig. 1. Links we monitor are not selected randomly, but based on pragmatic constraints: the physical layer characteristics (only POS links, no Spatial Reuse Protocol or channelized links), link capacity (no OC-192 links yet), geographical locations (no space for our monitoring equipment at some POPs), types of customers (peer or customer), and research topics (traffic matrix, delay measurement, routing, etc.). Thus, we do not claim that our data is statistically representative of our backbone network.

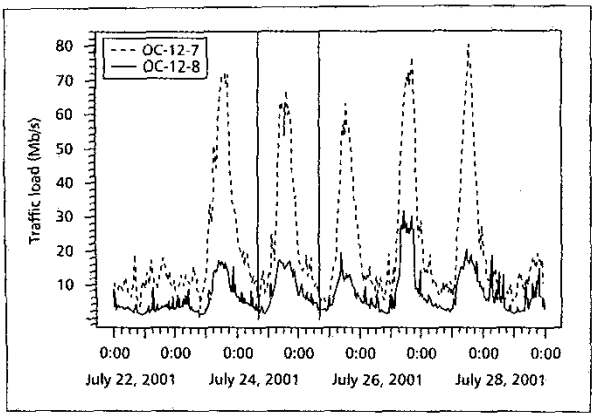
Due to space limitations, we do not present results from all of the traces, but choose to use a subset of the 32 most recent traces for this article. The goal of this article is to demonstrate the strengths and functionalities of the IPMON system, and present general observations made through them on the Sprint IP backbone network. For this purpose, we believe 32 traces are enough. For ease of presentation, we limit ourselves to only one or two traces in some of the figures. Readers are referred to the Data Management System at http://ipmon.sprint.com for the exhaustive list of available traces and analysis results.

The link speeds, start times, and durations of the 32 traces used in the article are given in Table 1. The starting time of traces on Tuesday, July 24, 2001, and Wednesday, September 5, 2001, was 8 a.m. EDT; that on Friday, April 19, 2002, was 1 p.m. EDT. Different days of the week were chosen in order to take into account time-of-day and day-of-week variations. Traces from 2001 are from OC-12 links, and those from 2002 from OC-48 links. Since we use a fixed amount of hard disk space, the durations of the traces depend on the link utilization: the higher the link utilization, the more packets captured and the shorter the trace. We can also fix the trace collection time to a constant, as in the case of OC-48 traces. Even-numbered traces are from the opposite directions of odd-numbered traces; for example, OC-12-1 and OC-12-2 are from the same link, but in opposite directions. We do not have week-long traces for all monitored

Trace	Link Speed	Start Time	Duration
OC-12-1	OC-12	July 24, 2001	13h 30m
OC-12-2	OC-12	July 24, 2001	2d 2h 35m
OC-12-3	OC-12	July 24, 2001	15h 55m
OC-12-4	OC-12	July 24, 2001	7h 34m
OC-12-5	OC-12	July 24, 2001	1d 3h 17m
OC-12-6	OC-12	July 24, 2001	23h 7m
OC-12-7	OC-12	July 24, 2001	4d 18h 42m
OC-12-8	OC-12	July 24, 2001	4d 10h 1m
OC-12-9	OC-12	July 24, 2001	4d 57m
OC-12-10	OC-12	July 24, 2001	6d 48m
OC-12-11	OC-12	Sept. 5, 2001	11h 2m
OC-12-12	OC-12	Sept. 5, 2001	10h 6m
OC-12-13	OC-12	Sept. 5, 2001	6h 17m
OC-12-14	OC-12	Sept. 5, 2001	2d 9h 47m
OC-12-15	OC-12	Sept. 5, 2001	1d 2h 5m
OC-12-16	OC-12	Sept. 5, 2001	7h 24m
OC-12-17	OC-12	Sept. 5, 2001	1d
OC-12-18	OC-12	Sept. 5, 2001	17h 51m
OC-12-19	OC-12	Sept. 5, 2001	16h 7m
OC-12-20	OC-12	Sept. 5, 2001	14h 3m
OC-12-21	OC-12	Sept. 5, 2001	16h 2m
OC-12-22	OC-12	Sept. 5, 2001	4d 19h 3m
OC-12-23	OC-12	Sept. 5, 2001	14h 13m
OC-12-24	OC-12	Sept. 5, 2001	13h 7m
OC-48-1	OC-48	April 19, 2002	1h
OC-48-2	OC-48	April 19, 2002	1h
OC-48-3	OC-48	April 19, 2002	1h
OC-48-4	OC-48	April 19, 2002	1h
OC-48-5	OC-48	April 19, 2002	1h
OC-48-6	OC-48	April 19, 2002	1h
OC-48-7	OC-48	April 19, 2002	1h
OC-48-8	OC-48	April 19, 2002	1h

■ Table 1. *Table of traces.*

[7] *We actually use CoralReef public suite and SNMP data to validate the workload results.*



■ Figure 2. *A week-long time-series plot from SNMP.*

links, only from a subset of links as shown in Table 1. Therefore, to study week-long trends, we resort to SNMP statistics collected separately.

Workload Characteristics

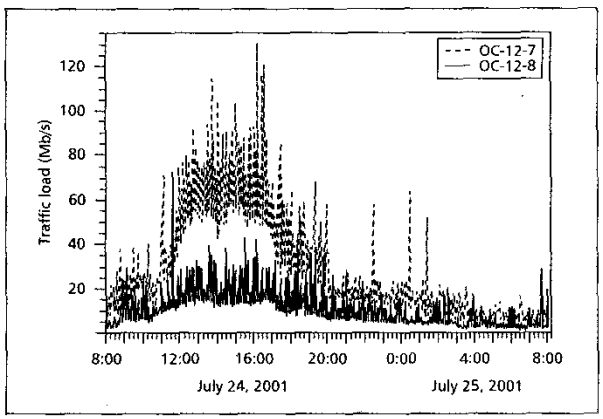
Traffic Load in Bytes—Figure 2 shows the traffic load collected over one week in 5 min intervals using SNMP. The SNMP statistics are collected from the same links from which we collected OC-12-7 and OC-12-8 traces. Daily peaks are visible from 9 a.m. to 5 p.m. On the weekend, the traffic decreases significantly. The same behavior is observed on all links with variations in peak height, duration, and hours, depending on geographic location and the customer type of the link [4]. Figure 3 shows the traffic load measured in 1 s intervals. The region marked by two vertical lines in Fig. 2 corresponds to the 24-h period shown in Fig. 3.

The following observations are of interest:

- Traffic load reported by SNMP is lower than that from IPMON measurements. On OC-12-7 the maximum from July 24, 2001, is about 68 Mb/s in SNMP, while it reaches above 125 Mb/s from the IPMON measurements. This is because the SNMP statistic is an average over 5 min, while the IPMON measured traffic load is calculated in 1 s intervals. This shows that the traffic is more bursty in a finer time granularity. In other words, SNMP statistics are not appropriate to detect short-term congestion.
- We observe distinct weekly and diurnal patterns in Figs. 2 and 3. From Monday to Friday, the traffic surges during the busy hours, and the load comes down significantly at night. The day-to-night traffic ratio is about 5:1 to 7:1. On the weekend the traffic load is significantly less than on weekdays, and does not exhibit clear patterns. The traffic load on the weekend is low possibly because it is outside of business hours.
- We observe that all OC-12 and OC-48 links have loads less than 90 Mb/s and 1.4 Gb/s, respectively. The results are consistent with our previous observations on overall network performance [20]: most of the links are utilized under 50 percent, and less than 10 percent of the links in the backbone experience utilization higher than 50 percent in any given 5 min interval. This is a consequence of bandwidth overprovisioning. Overprovisioning is not a waste of resources, but a design choice that allows Sprint to protect the network against multiple failures and handle traffic variability incurred by the absence of access control. This is analogous to the use of working and protect circuits in traditional telecommunications networks.
- In Fig. 3 we see occasional peaks in traffic load. There can be many causes behind such peaks: denial-of-service (DoS) attacks, routing loops, and bursty traffic. In some traces, we found an order of magnitude more TCP SYN packets than usual destined to the same addresses. We suspect those peaks are due to DoS attacks, for we observed many source addresses randomly spoofed toward the same destination address. But we admit that it is not easy to verify if the destinations suffered DoS attacks, since most organizations are reluctant to release such information. We also observed that transient routing loops caused spikes in traffic load. In other cases, peaks were simply due to very bursty arrivals of packets. We leave the detailed study of these phenomena for future work.
- Traffic on a bidirectional link is often asymmetric [21]. This traffic asymmetry results from two factors in the Sprint backbone. The first is the nature of an application. Many applications, such as the Web and ftp, are inherently asymmetric. One direction carries small request messages, and the other direction carries the actual Web data. For example, if a link connects to a Web server farm, the direction toward the server farm usually carries requests, and thus less traffic than the other direction. The second factor is routing. Most networks use the hot potato routing policy. Traffic destined to another network is passed to that network at the closest peering point. As a result, if a flow is observed on one direction of a link, it is possible that the reverse direction of the flow will follow a different route and will not be observed on the opposite direction of the link.

OC-12-1 and OC-12-2 contain examples of an extreme case. OC-12-1 has an average traffic volume of 200 Mb/s, and OC-12-2 has less than 20 Mb/s. OC-12-1 and OC-12-2 are to and from an international peer. Both the direction of Web requests and hot potato routing can explain the asymmetry on this link. Most links from 2001 exhibit traffic asymmetry between 2:1 and 5:1. As OC-48 POP-to-POP links carry more diverse and aggregated traffic, the loads are less asymmetric than on OC-12 links. It is hard to accurately extrapolate from our data how prevalent traffic asymmetry is in the network. However, the data shows that this is not uncommon, and traffic on links on the edge (in our case, OC-12 links) is more likely to be asymmetric.

Traffic Load by Applications— Next, we break down the traffic volume by application. We use port numbers to identify the application. When either the source or destination port number of a packet corresponds to a well-known port number for a specific application, we deem the packet as belonging to the application. Detailed mapping between port numbers and the applications is from the CoralReef public



■ Figure 3. *A day-long time-series plot from IPMON.*

suite [19]. We group similar applications into the following categories: Web, mail, file transfer, peer-to-peer, streaming, and others. The Web category includes those packets from Hyper Text Transfer Protocol (HTTP) and Secure Hyper Text Transfer Protocol (HTTPS). Mail traffic is from Post Office Protocol 3 (POP3) and Simple Mail Transfer Protocol (SMTP). The file transfer traffic includes FTP and SCP. A new kind of application, which we call peer-to-peer, has emerged recently, pioneered by Napster and Gnutella. It offers a way to share files among users, and has become a popular medium to share audio and video clips. Popular peer-to-peer applications include Napster, Morpheus, Gnutella, and KaZaa. Streaming media traffic is from Realaudio, Windows Media Player, and iMesh. All other known traffic, such as Domain Name System (DNS) and news, is grouped into the others category. The unknown category is for those without identifiable port numbers. As the peer-to-peer file sharing systems have gained popularity, audio and video clips of large sizes have added a serious amount of traffic to most university networks and more specifically to the connections to their ISPs. Subsequently, on some university networks, access to the file sharing systems has been limited by preventing traffic to or from certain port numbers at the firewall. To circumvent this blockage, many file sharing applications adopted the use of dynamically allocated port numbers instead of using fixed-numbered (or well-known) ports. For this reason, the amount of unknown traffic in the backbone has increased significantly in comparison to previous work [10]. From our observations and proprietary observations of DSL customers, we conjecture that the unknown traffic is mostly made up of peer-to-peer traffic.

Traffic type	Min (%)	Max (%)
Web	11	90
Peer-to-peer + unknown	0.1	80
Streaming	0.2	26
Mail	0	6
File transfer	0	7
Others	5	21

■ Table 2. *Percentiles of traffic by application.*

Table 2 shows the minimum and maximum percentiles of traffic each category contributes among the 32 traces used in this article.

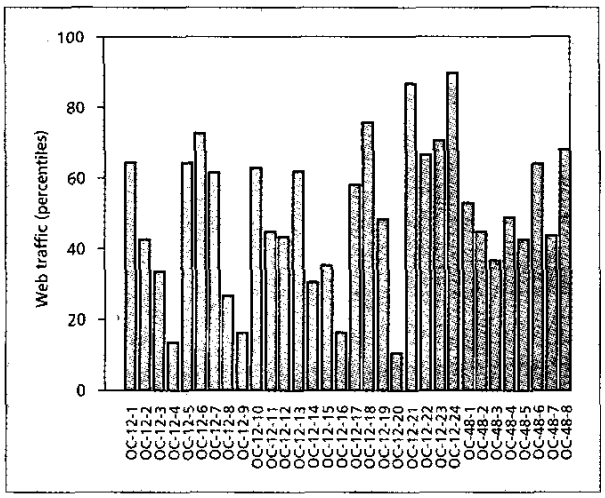
The application mix is quite different from link to link. Figure 4 plots the average Web traffic per link, and Fig. 5 plots the average traffic of peer-to-peer and unknown traffic combined. In most traces Web traffic represents more than 40 percent of the total traffic. This result is consistent with most prior traffic analysis studies [10, 11, 22]. However, on a handful of links (OC-12-4, OC-12-9, OC-12-16, and OC-12-20) the Web traffic contributes less than 20 percent, and we see the emergence of peer-to-peer traffic which contributes almost 80 percent of the total traffic on those links. Note that these links are customer and inter-router links. The OC-48 traces exhibit less variability between Web and peer-to-peer traffic than OC-12 traces. The OC-48 links we monitor are inter-POP backbone links, and carry heavily aggregated traffic. This could explain the small variability among them. Our observations indicate that peer-to-peer traffic may have become one of the two most dominant applications in the network along with Web traffic, and its emergence is not limited to certain types of links.

Another important observation is that streaming applications are a stable component of the traffic, if not as much in volume yet as the peer-to-peer applications. We observe 1–6 percent of streaming traffic even on OC-48 links.

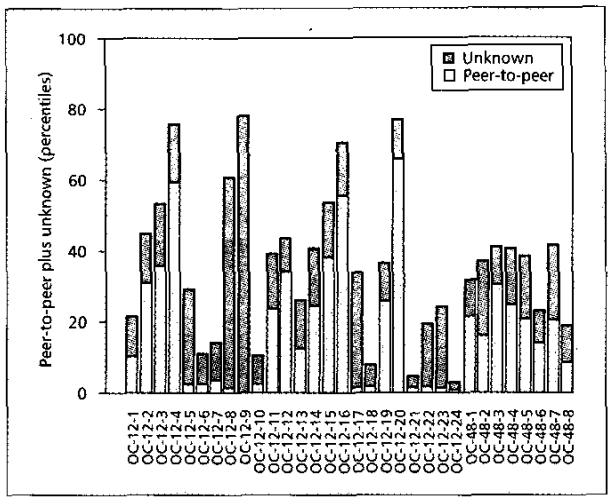
In addition to the application mix, we also consider the traffic breakdown by protocol (TCP/UDP/ICMP). We do not plot these results because in all cases above 90 percent of the traffic is TCP, even on the links with a significant percentage of streaming media.

Traffic Load in Flows — Now we consider the traffic in flows per minute. The start time of a flow is the time at which we observe for the first time a packet carrying a given 5-tuple. The flow ends when we do not see any packets with the same 5-tuple for 60 s. The 60 s timeout has been chosen based on previous work by Claffy *et al.* [23] and on our own observations [8]. A day-long analysis of the same traces used in Fig. 3 is presented in Fig. 6. For all the traces, the average number of flows per minute is plotted in Fig. 7.

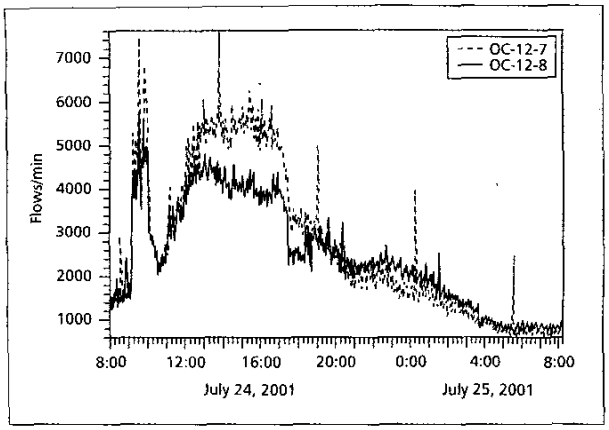
The main observation is that peaks in the number of flows in Fig. 6 do not necessarily translate to traffic load peaks of Fig. 3. Between 9 a.m. and 11 a.m. on July 24, 2001, the number of flows is as large as that during the peak hours between



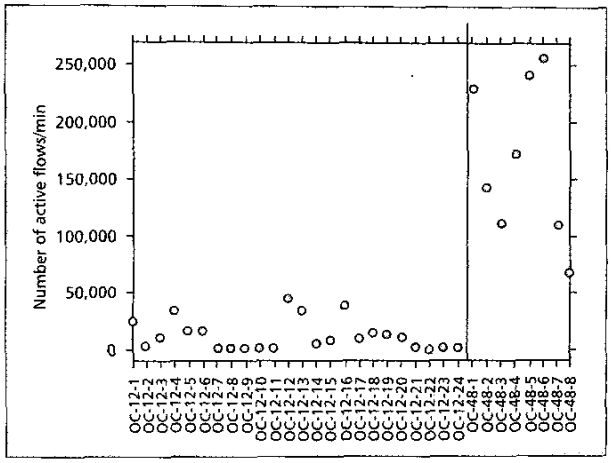
■ Figure 4. *Average percentiles of Web traffic vs. traces.*



■ Figure 5. *Average percentiles of peer-to-peer traffic vs. traces.*



■ Figure 6. *A time-series plot of number of flows per minute.*



■ Figure 7. *Average number of flows per minute vs. traces.*

noon and 5 p.m. During the same time period, the traffic load is often just half of that during the peak hours between noon and 5 p.m. The OC-12-7 and OC-12-8 traces are from a link to a content distribution network (CDN)[8] customer. The discrepancy in load and flow numbers is another example of the asymmetry discussed earlier. We also observe a small number of occasional conspicuous peaks in flow numbers. Performing a DNS lookup on the source and destination IP addresses of these flows, we find that the peaks are attributable to a large number of flows between servers of the CDN customer. However, they do not cause sudden increases in traffic load in Fig. 3.

The second observation is that the average number of active flows per minute is less than 50,000 for all OC-12 links and less than 300,000 for all OC-48 links in Fig. 7. In one OC-12 trace, the maximum number of active flows per minute is 10 times larger than the average, but remains under 400,000. A look into the 1 min interval with the maximum number of flows of that specific trace revealed that it was likely due to a DoS attack as described earlier. In the rest of the traces, the maximum numbers of active flows are 1.1–4 times larger than the average numbers.

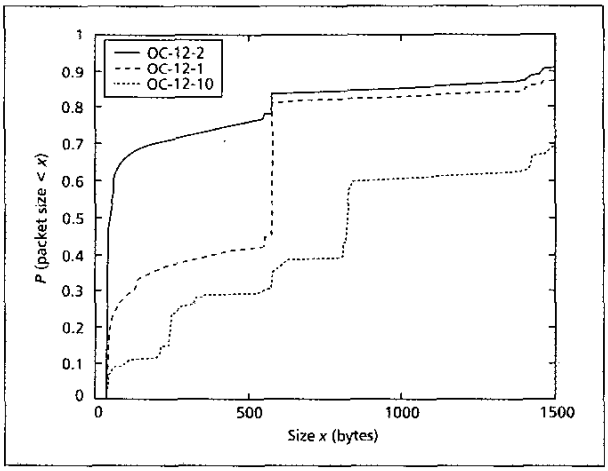
The result in Fig. 7 is important as it demonstrates that per-flow scheduling may be feasible in hardware on access links. This observation means that new avenues in traffic control should be explored, and that routers may go beyond TCP fairness and active queue management.[9]

Packet Size Distributions — Router designers find packet size distributions useful in optimizing the per-packet processing for the most common sizes. Prior work has shown that the packet size distribution is trimodal [10]. This was a result of a combination of TCP acknowledgments and the existence of two distinct default message transmission unit (MTU) sizes. Figure 8 demonstrates this trimodal packet size distribution for two traces, OC-12-1 and OC-12-2. These were selected as they show the typical distributions seen on most of the links we monitored. For these two traces, there are three steps at around 40, 572, and 1500, where 40 is for TCP ACKs, and 572 and 1500 are the most common default MTUs. When there is traffic asymmetry due to applications on the link, one step is more dominant than the others depending on the direction. The third trace, OC-12-10, exhibits a total of five steps with additional steps at 211 and around 820. The 211 byte packets correspond to a CDN proprietary UDP application that uses an unregistered port and carries a single 211 byte packet. Most 845 byte packets are from DNS. The 821 and 825 byte packets are generated by media streaming applications. Trace OC-12-10 clearly shows that the emergence of new applications requires that we revisit assumptions about the distribution of packet sizes on an IP backbone network.

TCP Performance

Except for the packet size distribution analysis, the results in the previous section do not require packet-level measurements. Such data can be collected using flow-level aggregate measurements. On the other hand, studying TCP performance requires knowledge about all packets transmitted in a TCP flow. In this section we demonstrate the types of TCP measurements possible with IPMON by presenting results on the RTT distribution and out-of-sequence packet statistics for the TCP flows.

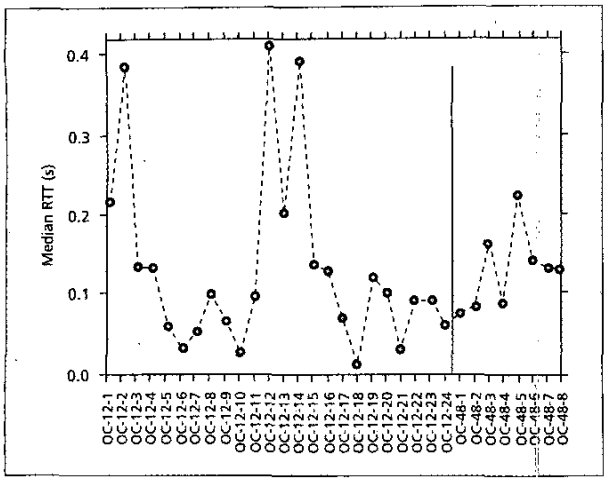
The RTT is measured as the time elapsed between a SYN packet and the first ACK packet that completes the three-way handshake, as proposed in [24]. Note that the RTT is measured end-to-end; it includes the time spent on the host computer, and the transmission time on the access link to the host computer (which can be as large as 150 ms



■ Figure 8. *Packet size cumulative distribution function.*

[9] *Recent developments in network processors allow per-flow states of more than a million concurrent flows to be processed by a router interface at line speed: http://www.agere.com*

[8] *A CDN is a mechanism to improve Web content delivery to end users.*



■ Figure 9. *Median round-trip time vs. traces.*

in the case of a dial-up modem). In addition, we can only compute the RTT for flows for which we observe the SYN/ACK pair: the RTT of a flow is accounted for in only one direction. Thus, to have a complete and accurate picture of RTT distribution for all flows on a link, RTT distributions from both directions should be combined. However, due to routing asymmetry, this is not always feasible. Also, the RTT of a flow is not a constant value as it may change over the duration of the flow due to changes in network congestion or routing: a single value of RTT taken at the beginning of a flow can only be a rough estimate of the RTT distribution for the flow. All these limitations in the methodology should be taken into consideration in interpreting the RTT results below. However, measuring RTT in the middle of the network allows us to collect many more data points than generally would be possible with active end-to-end measurements.

Figure 9 shows the median RTTs vs. traces. On all links, the median RTT lies below 450 ms. Three traces, OC-12-2, OC-12-12, and OC-12-14, have the median RTT above 300 ms. This result is easily explained because the links from which these traces were collected are primarily connected to European customers. Six traces (OC-12-6, OC-12-7, OC-12-10, OC-12-18, OC-12-20, OC-12-24) have the median RTT below 50 ms. The traffic on these links is primarily from CDNs. This is consistent with the results of Krishnamurthy *et al.* that show CDNs improve the overall response time of customer requests [25].

Figure 9 shows the rate of out-of-sequence packets for TCP flows defined by the 5-tuple as discussed earlier. Possible causes of out-of-sequence packets are retransmission after loss, unnecessary retransmission, duplicates, and reordering. Jaiswal *et al.* report that most of such out-of-sequence packets are due to retransmission after loss [6]. While this may seem to be a crude estimate for the end-to-end loss of a flow, it provides an upper bound on the number of losses we can detect from our measurements.[10]

In Fig. 10, we see that in all traces, 90 percent of the flows experience no out-of-sequence packets; in only a handful of traces is the 99th percentile above 30 percent out-of-sequence. The maximum out-of-sequence packet rate often reaches above 90 percent, but this may be a result of short flows losing most of their packets and reporting a high loss rate. The fact that 90 percent of flows experience an out-of-sequence rate of 0 percent on all the monitored links shows that most TCP flows experience no end-to-end loss.

[10] *If a packet is lost before it reaches the link we monitor, and is somehow retransmitted in order, there is no way we can determine that a loss has occurred. We believe this case is unusual enough that it does not affect our results significantly.*

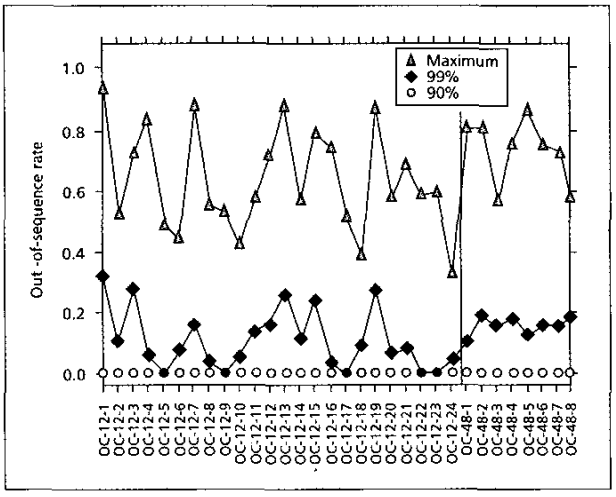
Delay Measurements

An accurate understanding of packet delay characteristics is important, since delay is a major metric in the definition of service level agreements (SLAs). Delay and delay variation (i.e., jitter) are critical to applications such as voice over IP (VoIP). Currently, delay measurements rely on active measurements. While these measurements provide good estimates of the average network delay, they require a large amount of probe traffic to be generated in order to be useful in the construction of models, in the evaluation of SLAs, or in the assessment of application feasibility (e.g., VoIP). Furthermore, many of the active probes use ICMP packets that are handled with a lower priority in routers, and whose delay may not be representative. Unlike active probes, our delay measurements are derived from all packets that traverse the network from one observation point to the other.

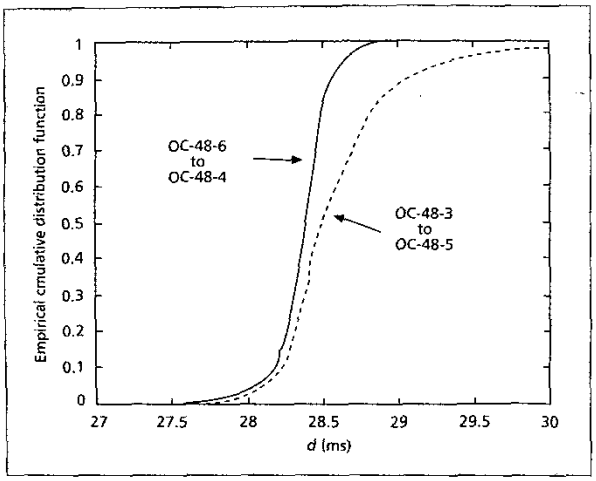
The GPS mechanism we have implemented in the monitoring systems gives us an accurate measurement of the delay a packet experiences in our backbone. A packet, observed at time t on one link and at time $t + q$ on another link, actually spent time q traveling between these links. By monitoring links entering and exiting a single router, we can measure the queuing behavior of the router. By monitoring links in different geographic locations, we can measure the queuing behavior of the backbone.

Obtaining delay distributions through multiple POPs is more challenging than single-hop delay distributions. We do not always find common packets in a pair of OC-48 backbone traces. However, when we do find matching packets in two OC-48 traces, the number of matched packets is very large. U.S. transcontinental delay distributions in Fig. 11 are obtained between San Jose and New York, and reflect 200 million packet matches in a 1 h period.[11] Packets identified in these delay distributions crossed five POPs and eight core routers.

[11] *For delay distributions from other traces, we again refer readers to http://ipmon.sprint.com*



■ Figure 10. *Out-of-sequence rate vs. traces.*



■ Figure 11. *Delay distributions.*

The minimum delays are 27.58 ms from OC-48-6 to OC-48-4 (from San Jose to New York), and 27.34 ms from OC-48-3 to OC-48-5 (from New York to San Jose); the average delays are 28.37 ms and 28.58, and the 99.9 percent delays are 28.99 ms and 31 ms, respectively. The jitter on these paths is consequently limited to less than 3 ms. This amount of jitter is not sufficient to impact the performance of delay-constrained applications such as media streaming or VoIP. While over 99.99 percent of packets experienced less than 31 ms delay, we observe a very small number of packets that experienced delay above 100 ms. Router idiosyncrasies are identified as a cause of large delays.[12]

The analysis of the delay distributions reveals two major characteristics of the backbone, as partially monitored in our work. First, the main contributing factor in network delay is the speed of light. Second, jitter is extremely low. As our measurements do not cover the entire backbone network and represent only a small portion of it, the two characteristics apply only to those paths we monitored.

Conclusions

We describe a passive monitoring system that is capable of capturing packet-level traces on high-speed backbone links. This monitoring infrastructure is innovative in two aspects. First, it has the capability of simultaneously collecting information with fine granularity on multiple geographically dispersed links. Second, all of the collected information is timestamped with a GPS-synchronized global clock, giving us the ability to do detailed analyses of packet queuing and transmission behaviors on an Internet backbone.

We have deployed our monitoring infrastructure on multiple OC-3, OC-12, and OC-48 bidirectional links in four POPs in the Sprint IP backbone network, and collected weeks of traces. This article presents a synthesis of the results from traces collected in July and September 2001 and April 2002. Interested readers are referred to http://ipmon.sprint.com for additional results. Ongoing work is focused on the deployment of IPMON systems on OC-192 links and on upgrading the DAG card in order to add new filtering and sampling capabilities.

We observe that link load characteristics often vary from link to link and that these variations are often correlated to the nature of the customers connected to the POP. As one might expect, as traffic becomes more highly aggregated (e.g., on OC-48 backbone links), there is a higher degree of consistency. We also show that some links no longer have Web traffic as their dominant component. In those traces, file sharing and media streaming applications represent up to 80 percent of the total traffic. We also compute the number of active flows and show that it is small enough to consider per-flow queuing as a feasible technology to control the traffic and provide new services. Finally, we showed that TCP flows on most links exhibit low out-of-sequence packet rates and that backbone delay is dominated by the speed of light.

Our approach would not scale to monitoring every link in a tier 1 backbone, but deployed on the current scale it provides crucial data for understanding the dynamics of network traffic; data that is not available from existing router-based monitoring tools. In the long term, the goal of this project is to identify which metrics need to be monitored in real time and to work with router vendors to design measurement functions embedded in routers. It is through precise understanding of traffic dynamics that we will be able to make the design and control of Internet backbones an engineering science.

[12] *We suspect that routing updates, IP option packets, and SNMP requests interfere with packet forwarding [18].*

Acknowledgments

We would like to thank Konstantina Papagiannaki and Gianluca Iannaccone for helping us construct the figures, and Vern Paxson for pointing out the difficulties in trace sanitization. In the course of this project we received help from so many people that it is impossible to acknowledge them individually here. Instead we choose to acknowledge their institutions: Sprint IE, Sprint IBO, Sprint ATL, the University of Waikato, and CAIDA.

References

[1] A. Feldmann *et al.*, "Deriving Traffic Demands for Operational IP Networks: Methodology and Experience," *IEEE/ACM Trans. Net.*, June 2001.
[2] A. Medina *et al.*, "Traffic Matrix Estimation: Existing Techniques Compared and New Directions," *Proc. 2002 ACM SIGCOMM*, Pittsburgh, PA, Aug. 2002.
[3] C. Barakat *et al.*, "A Flow-Based Model for Internet Backbone Traffic," *Proc. ACM SIGCOMM Internet Measurement Wksp.*, Marseilles, France, Nov. 2002.
[4] S. Bhattacharyya *et al.*, "POP-Level and Access-Link-Level Traffic Dynamics in a Tier-1 POP," *Proc. ACM SIGCOMM Internet Measurement Wksp.*, San Francisco, CA, Nov. 2001.
[5] U. Hengartner *et al.*, "Detection and Analysis of Routing Loops in Packet Traces," *Proc. ACM SIGCOMM Internet Measurement Wksp.*, Marseille, France, Nov. 2002.
[6] S. Jaiswal *et al.*, "Measurement and Classification of Out-Of-Sequence Packets in a Tier-1 IP Backbone," *Proc. ACM SIGCOMM Internet Measurement Wksp.*, Marseilles, France, Nov. 2002.
[7] C. Boutremans, G. Iannaccone, and C. Diot, "Impact of Link Failure on VoIP Performance," *Proc. 12th Int'l. Wksp. Net. and Op. Sys. Support for Digital Audio and Video*, Miami, FL, May 2002.
[8] G. Iannaccone *et al.*, "Monitoring Very High Speed Links," *Proc. ACM SIGCOMM Internet Measurement Wksp.*, San Francisco, CA ,Nov. 2001, pp. 267–72.
[9] J. Apsidorf *et al.*, "OC3MON: Flexible, Affordable, High Performance Statistics Collection," *Proc. INET*, June 1997.
[10] K. Thompson, G. Miller, and R. Wilder, "Wide Area Internet Traffic Patterns and Characteristics," *IEEE Network*, Nov. 1997.
[11] K. C. Claffy, G. Miller, and K. Thompson, "The Nature of the Beast: Recent Traffic Measurements from an Internet Backbone," *Proc. INET '98*, Geneva, Switzerland, July 1998.
[12] A. Feldmann *et al.*, "Netscope: Traffic Engineering for IP Networks," *IEEE Network*, Mar./Apr. 2000.
[13] R. Caceres *et al.*, "Measurement and Analysis of IP Network Usage and Behavior," *IEEE Commun. Mag.*, vol. 38, no. 5, May 2000.
[14] "NetFlow Services and Applications," Cisco white paper, http://www.cisco.com/warp/public-/ cc/pd/iosw/ioft/neflct/tech/napps wp.htm, 2000.
[15] "Juniper Networks Solutions for Network Accounting," Juniper Networks white paper, http://www-.juniper-com-techcenter-techpapers-200010.pdf, 2000.
[16] C. Fraleigh *et al.*, "Design and Deployment of a Passive Monitoring Infrastructure," *Proc. Passive and Active Measurement Wksp.*, Amsterdam, The Netherlands, Apr. 2001.

[17] J. Cleary *et al.*, "Design Principles for Accurate Passive Measurement," *Proc. Passive and Active Measurement Wksp.*, Hamilton, New Zealand, Apr. 2000.
[18] K. Papagiannaki *et al.*, "Analysis of Measured Single-Hop Delay from an Operational Backbone Network," *Proc. IEEE INFOCOM 2002*, New York, NY, June 2002.
[19] K. Keys *et al.*, "The Architecture of Coralreef: An Internet Traffic Monitoring Software Suite," *Proc. of Passive and Active Measurement Wksp.*, Amsterdam, The Netherlands, Apr. 2001.
[20] S. Iyer *et al.*, "An Approach to Alleviate Link Overload as Observed on an IP Backbone," *Proc. IEEE INFOCOM 2003*, San Francisco, CA, Apr. 2003.
[21] V. Paxson, "End-to-End Routing Behavior in the Internet," *IEEE/ACM Trans. Net.*, vol. 5, no. 5, Oct. 1997.
[22] S. McCreary and K. C. Claffy, "Trends in Wide Area IP Traffic Patterns," *ITC Specialist Seminar*, Monterey, CA, May 2000.
[23] K. C. Claffy, H.-W. Braun, and G. C. Polyzos, "A Parameterizable Methodology for Internet Traffic Flow Profiling," *IEEE JSAC*, vol. 13, no. 8, Oct. 1995, pp. 1481–94.
[24] H. S. Martin, A. McGregor, and J. G. Cleary, "Analysis of Internet Delay Times," *Proc. Passive and Active Measurement Wksp.*, Auckland, New Zealand, Apr. 2000.
[25] B. Krishnamurthy, C. Wills, and Y. Zhang, "On the Use and Performance of Content Distribution Networks," *Proc. ACM SIGCOMM Internet Measurement Wksp.*, San Francisco, CA, Nov. 2001.

Biographies

CHUCK FRALEIGH (cjf@stanford.edu) received a B.S. degree in computer and electrical engineering from Purdue University in 1996. He received his M.S. and Ph.D. degrees in electrical engineering from Stanford University in 1998 and 2002, respectively. From 1998 to 2002 he was a student visitor at Sprint ATL and is now with NetVmg. His research interests include Internet traffic measurement and modeling, network provisioning, and the design of network measurement systems.

SUE MOON (sbmoon@cs.kaist.ac.kr) received her B.S. and M.S. from Seoul National University, Korea, in 1988 and 1990, respectively, all in computer engineering. She received a Ph.D. degree in computer science from the University of Massachusetts at Amherst in 2000. She was at Sprint ATL from 1999 to 2003, and now teaches at KAIST in Korea. Her research interests are in network performance measurement and monitoring: network delay, traffic scaling behavior analysis, and network anomalies.

BRYAN LYLES (jblyles@acm.org) is an independent consultant. From 1998 to 2003 he was a senior scientist at Sprint, where he was responsible for founding the IP research group. Prior to Sprint he was a member of research staff at Xerox PARC. He received his B.A. degree from the University of Virginia and his Ph.D. from the University of Rochester, New York.

CHASE COTTON (chase.j.cotton@mail.sprint.com) is director of IP Engineering and Technology Development at Sprint in Reston, Virginia, where he manages the worldwide deployment of Sprint's Tier 1 IP backbone, SprintLink, and other IP products. Prior to joining Sprint he managed IP, DSL, and VPN design, deployment, and testing activities at SBC, Pacific Bell, and Southern New England Telephone. He also directed computer communications research for 11 years at Bellcore (now Telcordia). He received a Ph.D. from the University of Delaware.

MUJAHID KHAN (mkhan@sprint.net) received his B.S. and M.S. in electrical engineering in 1990 and 1993, respectively. He has been working at Sprint IP Services since 1996. Before joining Sprint, he worked for UUNET.

DEB MOLL (deborah.a.moll@mail.sprint.com) received a B.S. in operations research and industrial engineering from Cornell University in 1996 and has been at Sprint since then.

ROB ROCKELL (rrockell@sprint.net) is a principal network design engineer at Sprintlink. He received a B.S. in physics in 1996 from Carnegie Mellon University and has been with Sprint since then.

TED SEELY (tseely@sprint.net) is a principal network design engineer for Sprintlink engineering. He has been working in the communications industry for 15 years. Prior to joining Sprintlink, he worked as a consultant for various carriers. He received his training in electronic engineering in the military, and deployed secure communication networks there.

CHRISTOPHE DIOT (christophe.diot@intel.com) received a Ph.D. degree in computer science from INP Grenoble in 1991. From 1993 to 1998 he was a research scientist at INRIA Sophia Antipolis, working on new Internet architecture and protocols. From 1998 to 2003 he was in charge of the IP research team at Sprint ATL. He recently moved to Intel Research, Cambridge, United Kingdom. His current interests are measurement techniques and Internet architecture.