Packet-Level Traffic Measurements from a Tier-1 IP Backbone

Chuck Fraleigh, Sue Moon, Bryan Lyles, Chase Cotton, Mujahid Khan, Deb Moll, Rob Rockell, Ted Seely, Christophe Diot

Abstract— **Network traffic measurements provide essential data for networking research and network management. In this paper 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 terabyte SAN (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 show first that SNMP-based monitoring is not accurate enough. 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 packets rates. We finally establish that delay inside the backbone is dominated by speed of light.**

I. Introduction

OVER-PROVISIONING is widely used by packet network engineering teams to achieve failure resilience and support the rapid growth of traffic volume. So far this approach has been successful in maintaining scalable, simple, highly available, and robust networks. It is important to realize that in packet networks that do not perform call admission control, there is often no way to control the amount and 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 SNMP (Simple Network Management Protocol), are limited in their capabilities, since they only provide aggregate level statistics about the traffic (e.g. average link utilization over five minute intervals) and do not give insight into traffic dynamics on times scales appropriate for events such as packet dropping. Another example is the demand traffic matrix which is a crucial input to many network planning, provisioning, and engineering problems, but which is difficult to obtain with available tools [1], [2].

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

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

In order to gain a 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 (i) collecting packet-level traces at multiple points of the Sprint IP backbone for link speeds of up to OC-48 (2.5 Gbps), (ii) marking each of the packets with a sub-microsecond time-stamp, and (iii) synchronizing these traces to within 5 μs. Off-line processing of the packet traces then enables detailed studies of the various aspects of traffic characteristics such as delay and loss.

This paper first describes the architecture and capabilities of the IPMON system. We then present observations of traffic on OC-12 (622 Mbps) and OC-48 links in the Sprint IP backbone network[1].

Results presented in this paper provide a high-level view of a major backbone network traffic in 2001 and 2002, and highlight the changes that have occurred in traffic characteristics with respect to previous studies. We illustrate the limits of SNMP. We identify the impact of new applications such as distributed file sharing and streaming media. We find that on some links, over 60% of the traffic is generated by these new applications while only 30% is web traffic. We also present results on end-to-end loss and round-trip-time (rtt) performance of TCP connections that are significantly different from previous observations. We also present results on the network delays that are experienced through a single router in the backbone as well as the U.S. transcontinental delay measurement. We find that packets experience very little queuing delay and insignificant jitter in the backbone.

The paper is organized as follows. Section II discusses related work. Section III describes the monitoring system

C. Fraleigh (cjf@stanford.edu) was with Stanford Univ., Palo Alto, CA, and with Sprint ATL, Burlingame, CA. He is currently at NetVMG, Palo Alto, CA. S. Moon (sbmoon@sprintlabs.com), B. Lyles (lyles@sprintlabs.com), and C. Diot (cdiot@sprintlabs.com) are with Sprint ATL. C. Cotton (chase.j.cotton@mail.sprint.com), M. Khan (mkhan@sprint.net), D. Moll (deborah.a.moll@mail.sprint.com), R. Rockell (rrockell@sprint.net), and T. Seely (tseely@sprint.net) are with Sprint, Reston, VA.

[1] www.sprint.net

architecture. Section IV presents and analyzes traffic measurements from the Sprint IP backbone network. It starts with a brief description of the 32 traces used in the paper, and analyses the traffic load broken into bytes, applications, and numbers of flows. The performance of TCP flows is evaluated in terms of round trip times and out-of-sequence packet rates. Lastly, delay measurements are presented. Section V concludes and discusses future work.

II. Related Work

The challenges in designing a monitoring system for a comprehensive view of the network performance are (i) the collection of detailed traffic statistics, including application mixes and traffic matrixes, from heterogenous network links, (ii) limiting the side-effects of the monitoring system on the monitored network and (iii) 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 which then extrapolates the performance of the network from the performance of the injected traffic. Both the probe methodology and the probe volume are defined by the metrics to be observed and by the level of accuracy needed in the analysis. A major drawback of active monitoring systems is that probing for one metric, for example bottleneck bandwidth, will often bias the results for other metrics, for example packet loss. Furthermore, metric specific probes may result in it not being possible to use an archived data set to answer new questions. Another important limitation is that it is difficult to control the path taken by the probe packets.

The second approach is that of passively observing and recording network traffic. These passive measurement systems use the recorded traffic to characterize both the user applications and the network's performance. The drawback of passive monitoring systems is that they depend on the existence of appropriate network traffic. The capabilities of passive monitoring systems tend to be heavily defined by the characteristics of the hardware infrastructure used. Due to the quantity of data produced, recording traces from very high bandwidth links is a serious challenge [8]. If the data is summarized in order to reduce its volume, then issue (i) may be compromised. For the same reason, issue (iii) requires the deployment of monitoring hardware in various locations of the network. As a result, global observations have often been addressed by inference techniques, and not by exhaustive passive monitoring of every link in a network. Finally, passive systems which record and archive full traces allow for re-analysis of the archived data.

Ping, Traceroute and Pathchar are good examples of simple active monitoring tools. Paxson used these tools, and others, to perform pioneering work [9],[10] in network measurement. Paxson's measurement system has been further developed into the NIMI (National Internet Measurement Infrastructure) project [11]. NIMI relies on servers deployed at different locations of the Internet and which generate and exchange active monitoring traffic. Traffic metrics derived from these active probes include available bandwidth, delay, and packet loss. NIMI only partially addresses issue (i) since it only provides delay and loss data, and issue (iii) given that the number of deployed NIMI servers is "large enough". Issue (ii) is addressed in active measurements by limiting the probe traffic in order to minimize the impact on the network at the possible expense of measurement accuracy.

Other active measurement projects include Surveyor, which uses a set of 41 GPS synchronized systems to measure one-way network delay and loss [12]; PingER (Ping End-to-end Reporting), which measures packet loss and available bandwidth between high energy nuclear and particle physics research facilities [13]; and the RIPE Test Traffic Measurement project, which measures bandwidth and delay performance between 60 measurement systems in Europe, North America, and Israel [14].

OC3MON is a passive monitoring system for OC-3 links (155 Mbps) described in [15]. OC3MON collects packet-level traces or flow-level statistics. Packet-level traces can be collected only for a limited amount of time (only 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 investigates daily and weekly variations in traffic volume, packet size distribution, and traffic composition in terms of protocols and applications [16]. OC3MON has been extended in [17] to support OC-12 and OC-48 links[2]. 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.

NetFlow [18] is a Cisco proprietary passive measurement tool. NetFlow collects information about every TCP and UDP flow on a particular input or output link of a router. A flow record includes the source and destination addresses and port numbers, numbers of bytes and packets transmitted, and duration in time. The flow information is collected by the router and transmitted to an external system for storage. While NetFlow is a powerful measurement tool, unreliable transmission of its output to the collection site can be a concern when the path between the routers and the collection site is heavily loaded [1]. NetFlow can also impact the performance of a router in case of DoS (Denial-of-Service) attacks, for example.

Juniper Networks has a set of accounting tools to collect similar statistics as NetFlow [19]. Using the Internet Processor II ASIC, the Juniper routers support filter-based, MPLS-based, and destination class usage accounting. They also allow accounting profiles to be stored on a local disk. This feature adds flexibility in transferring the collected accounting data when managing high-capacity routers.

[2]The analysis results from two 1-hour-long OC-48 traces are available at http://www.caida.org.

There are several projects which combine both active and passive measurement. The NetScope project [20] collects measurements from the AT&T network in order to study the effects of changing network routes and router configuration. Using NetFlow measurements from router, the traffic demand for the entire network is derived [21]. 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 passsive monitoring system called PacketScope is developed and used to collect and filter packet-level information.

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

There are commercial products for passive monitoring, such as Niksun's NetDetector [4] and NetScout's ATM Probes [5]. These systems, however, are limited to OC-3 or lower link speed, and are thus not adequate for Internet backbone links.

Our monitoring infrastructure, called IPMON, solves issue (i) as our hardware capabilities allow us to record packet traces for any link capacity up to OC-48. The range of observable metrics is wider than with any of the above systems thanks to the timestamps synchronized to a global clock signal to the 5 μs accuracy. Issue (ii) is addressed because the system is passive; data collection does not disturb the traffic being observed. We partially address issue (iii) by deploying the monitoring entities in geographically distributed locations and by monitoring a diverse set of links. The IPMON components will be described and discussed in greater details in the next section.

III. 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.

A. The Sprint IP backbone network

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 Gbps) and OC-192 (10 Gbps) links. From each POP links radiate outwards to customers (e.g. large corporate networks, regional tier-2 ISPs or DSL-aggregation devices, and large server farms), which typically require higher bandwidth network connections[6]. Each POP may have links, known as private peering points, to other backbone networks as well as links to public network access points (NAPs). Because of traffic volume, major backbone networks often have peering links in multiple, geographicly distinct, POPs.

The Sprint IP backbone consists of approximately 40 POPs worldwide, of which about 20 are located in the U.S.A. Figure 1 shows an abstract view of the Sprint U.S. backbone topology. Within a POP the network is a two-level, hierarchical structure: access (edge or gateway) and backbone (or core). 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-Sonet (POS) framing which in turn runs over Sprint's DWDM (Dense Wavelength Division Multiplexing) optical network.

B. The IPMON monitoring infrastructure

A detailed description of the monitoring infrastructure is provided in [22]. We give here a short description of the IPMON architecture. IPMON consists of three elements (see Figure 1): a set of passive monitoring entities which collect the packet traces; a data repository that stores the traces once they have been collected; and an analysis platform which performs off-line analysis. Analysis is performed off-line 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 lab where many systems can access the data simultaneously. The second reason is we archive the traces for use in future projects.

B.1 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, the IP and UDP/TCP headers, as well as a sub-microsecond timestamp which indicates the time at which the packet was observed. The source and destination IP addressess are not anonymized, since they are needed in routing-related analysis.

Each monitoring entity is a dual-processor Linux PC (Dell PowerEdge 6000 series) with 1 GB main memory, a large disk array (100 to 330 GB), and a POS network interface card, known as the DAG card [23]. 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 box where software tranfers 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 PC. 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 the DAG card cannot

[3] http://moat.nlanr.net/PMA/
[4] http://www.niksun.com/
[5] http://www.netscout.com/products/probes.html
[6] Lower bandwidth customers, such as dial-up home users, connect to tier-2 ISPs which in turn connect to the backbone network.

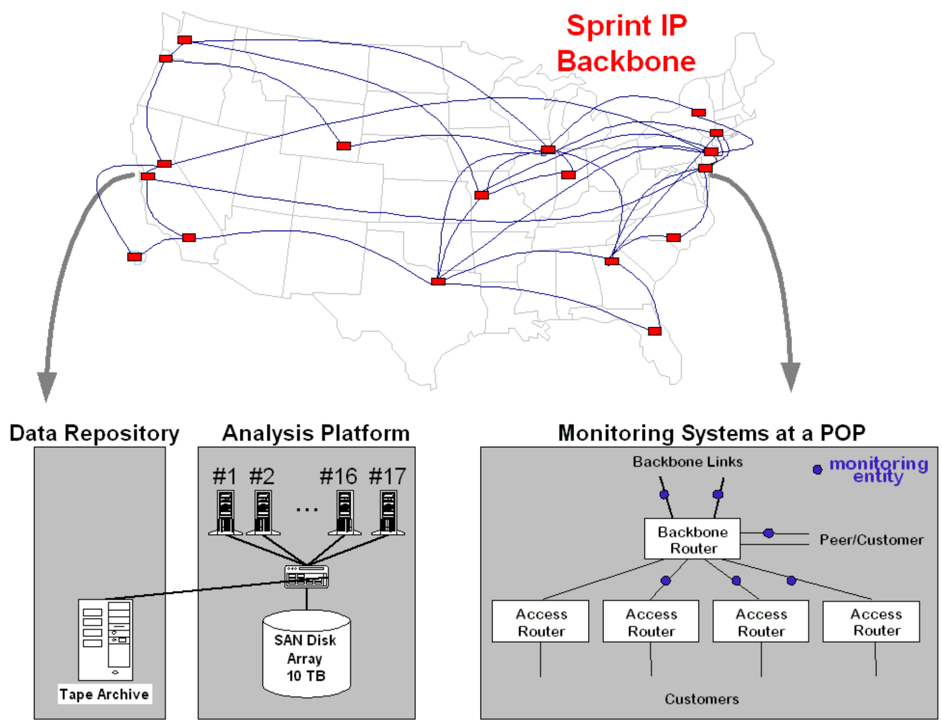

Fig. 1. The IPMON system in the Sprint IP backbone

compromise network integrity.

Each monitoring entity has a removable disk array of up to 330 GB. 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 pre-defined interval or allow it to run until space on the hard disks is exhausted. By 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 GPS signal. GPS is a satellite based system that provides a global time information with an accuracy of 20 nanoseconds. Hardware errors as well as other system related issues bring the maximum error on timestamps to 5 μs [22][23]. This synchronization ability allows us to measure the one-way network delay between two monitored links.

A total of 60 monitoring entities are installed at 4 different POPs, chosen on the basis of geographic diversity and connectivity. They monitor the traffic on OC-3, OC-12, and OC-48 links which connect the access routers, the backbone routers and several of the public peering links.

B.2 Data Repository

The data repository is a two level store consisting of a 12 TB removable tape library and a 10 TB 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 10TB 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.

To improve transfer time and decrease the storage capacity requirements, the trace data is compressed before it is transferred over the network or placed on the data repository. Using the compression program *bzip*, we are able to achieve compression ratios ranging from 2:1 to 3:1 depending on the trace characteristics. We are developing more sophisticated compression techniques in order to be able to support data capture on OC-192 links [8].

B.3 Data Analysis Platform

Data analysis is performed on a cluster of 17 high-end personal computers (PCs) connected via a Storage Area Network (SAN) to the 10 TB disk array. 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.
- Multi-trace analysis involves correlating traffic measurements from different links. This includes calculating delay and identifying packet losses. The key to performing multi-trace analysis is to identify an individual packet as it travels across multiple links in the network. To identify a packet

we use the 30 bytes 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 [24].

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

- The first set of tools is a set of custom tools which extract 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 which we use to identify the amount of traffic generated by different protocols (e.g. TCP, UDP) and applications (e.g. web, email, media streaming) [25].
- The third set of tools is used for multi-trace correlation. These tools use the hash-based algorithm for finding packets that have been recorded on multiple links and returns a list of these packets and the time at which they were observed on each link.

C. Trace Sanitization

The 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 the trace consistency. Hard disk failures are the most common in our experience.
- GPS synchronization can be lost. The monitoring entities are currently daisy chained to the GPS source and the failure of one entity affects all systems down the chain (GPS clock distribution is currently being upgraded to fix this specific problem). Also problems internal to a DAG card can cause that DAG card to become out of synchronization with others.
- Because the DAG hardware has been co-developed with the monitoring platform, hardware or software bugs have impacted the traces. For example, we have observed traces where packets were missing, or traces had sequences of zeroes. Misalignment or byte swapping has also been a problem.
- While they are tranferred from the collection site to the analysis platform, traces can be corrupted or truncated due to intermediate system failures: local disk failure, defective tapes, etc.

Therefore, before being used, traces need to be sanitized. We realized the need for sanitization from the first trace collection, and, as we discovered and fixed sources of corruption, have steadily improved upon the process. Today it is established as a systematic process that is run on every trace before the trace 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 analyse 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 the information, such as packet type, contained in the HDLC header. We then check that the structure of the packet is correct for the packet type.
- We check that the timestamps are monotonically increasing, and that the inter-packet time is both greater than the time required to transmitt the previous packet and that any gaps in the trace are reasonable[7].
- We detect traces out of GPS synchronization by calculating the delay between traces. If the minimum delay per minute between two traces does not remain constant, and fluctuates more than a few milliseconds, those two traces are considered out of synchronization.

Anytime a problem is detected, the corresponding trace is ignored and only those traces that are "sanitized" per process described above are used in analysis.

IV. Measurement Results

In this section we present measurement results to demonstrate the capabilities of the IPMON system and to provide information on the characteristics of 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 [8]. 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 are TCP performance statistics. These results demonstrate the advantages of collecting packet-level measurements. The third set of results are packet delay measurements through a single backbone router and over a U. S. transcontinental path.

A. Trace Description

The IPMON system collects measurements from about 30 bidirectional links out of about 5000 links at 4 POPs in the Sprint IP backbone. Three POPs are located on the east coast of the U.S.A., and one POP on the west

[7] 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 clock synchronization problem.

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

coast. OC-48 links we monitor are all long-haul trans-continental connections. The other links either connect backbone routers to access routers within the POP, or connect peers and customers to the backbone as in Figure 1. Due to space limitation, we do not present results from all of the traces, but choose to use a subset of 32 traces for this paper. The goal of this paper 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 the 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.sprintlabs.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 paper are given in Table IV-A. The starting time of traces on Tuesday, July 24th, 2001 and Wednesday, September 5th, 2001 was 8am EDT; that on Friday, April 19th, 2002 was at 1pm EDT. Different days of the week were chosen in order to take into account time-of-day and day-of-week variations. Traces of 2001 are from OC-12 links, and those of 2002 are from OC-48. 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 is, the more packets are captured, and the shorter the trace is. 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 fom the same link, but in opposite directions. We do not have week-long traces for all monitored links, but only from a subset of links as shown in Table IV-A. Therefore, to study the week-long trends, we resort to SNMP statistics collected separately.

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

TABLE I
TABLE OF TRACES

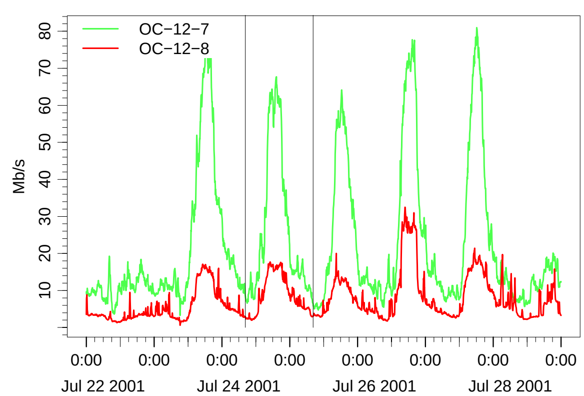

Fig. 2. Week-long time-series plot from SNMP

B. Workload Characteristics

B.1 Traffic Load in Bytes

Figure 2 shows the traffic load in 5 minute intervals collected using SNMP over one week. The SNMP statistics are collected from the same links that we collected OC-12-7 and OC-12-8 traces from. Daily peaks are visible between 9 am to 5 pm. On the weekend, the traffic decreases significantly. The same behavior is observed on all links with variations on peak height, duration, and hours, depending on the geographic location and the customer type of the link [4]. Figure 3 shows the traffic load measured in 1 second intervals collected on our monitoring platforms. The region marked by two vertical lines in Figure 2 corresponds to the 24-hour-long period shown in Figure 3. Figure 4 plots the average link utilizations in 1 second intervals of all the traces.

The following observations are of interest:

- Traffic load reported by SNMP is lower than that from the IPMON measurements. In OC-12-7 the maximum on July 24th, 2001, is about 68 Mbps in SNMP, while it reaches above 125 Mbps from the IPMON measurements. This is because the SNMP statistic is an average over 5 minutes, while the traffic load is calculated in 1 second intervals from the IPMON measurements. It shows that the traffic is more bursty in finer time granularity, and our monitoring system is capable of capturing it. In other words, SNMP statistics are not appropriate to detect short term congestions.
- We observe distinct weekly and diurnal patterns in Figures 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 the weekdays, and does not exhibit clear pattern. The low traffic load on the weekend is possibly due to business closing. Bhattacharrya et al demonstrates the impact of customer behavior on the overall traffic to the matching

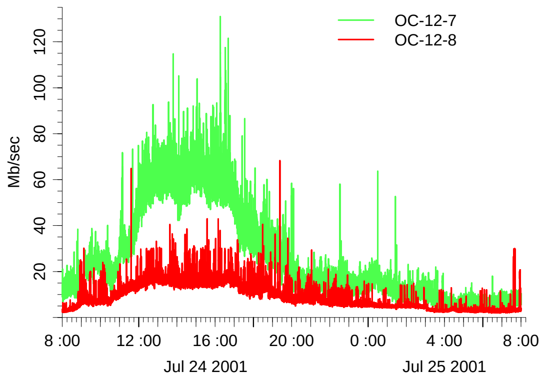

Fig. 3. Day-long time-series plot from IPMON

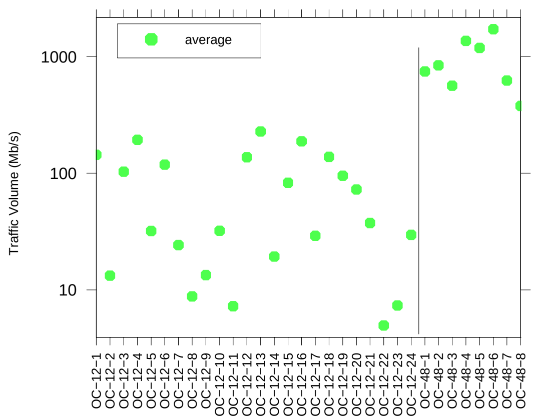

Fig. 4. Average link utilization vs. traces

effect [4].

• Figure 4 shows that all of the links are utilized under 60%, and most of them under 30%. The results are consistent with our previous observations on the overall network performance [26]: most of the links are utilized under 50%, and less than 10% of the links in the backbone experience utilization higher than 50% at any given 5 min interval. This is a consequence of bandwidth over-provisioning. Over-provisioning is not a waste of resources, but is a design choice that allows Sprint to protect the network against multiple failures and to handle traffic variability incurred by the absence of access control. This is analogous to the use of working and protect circuits in traditional telecommunication networks.

• In Figure 3 we see occasional peaks in utilization. There can be many causes behind such peaks: Denial-of-Service (DoS) attacks, routing loops, and simply bursty traffic. In some traces we found an order of magnitude more TCP SYN packets than usual that are destined to the same addresses. We suspect those peaks are due to DoS attacks, but admit that it is not easy to verify if the destinations suffered Denial-of-Service attacks, since most organizations are reluctant to release such information. We also observed that transient loops caused spikes in utilization. In other cases peaks were simply due to very bursty arrival of packets. We leave the detailed study of these phenomena for future work.

• Traffic volumes are asymmetric on both link directions. This traffic asymmetry results from two factors. The first factor is the nature of an application. Many applications, such as 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 are example of an extreme case. OC-12-1 has an average traffic volume 200 Mbps, OC-12-2 has less than 20 Mbps. 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 of 2 and 5 to 1 in Figure 4. As OC-48 POP-to-POP links carry more diverse and aggregated traffic, the loads are less asymmetric than OC-12 links. It is hard to accurately extrapolate the popularity of traffic asymmetry in the network from our data. However, the data shows that it is not uncommon.

B.2 Traffic Load by Applications

Next we break down the traffic volume by application. We use the 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 applications is from CoralReef public suite [25]. We group similar applications into the following categories: web, mail, file transfer, peer-to-peer, streaming, and others. The web category include those packets from HTTP (Hyper Text Transfer Protocol) and HTTPS (Secure Hyper Text Transfer Protocol). Mail traffic is from POP3 (Post Office Protocol 3) and SMTP (Simple Mail Transfer Protocol). The file transfer traffic includes FTP (File Transfer Protocol) and SCP (secure copy). A new kind of application, we call peer-to-peer, has emerged recently, pioneered by Napter and Gnutella. It offers a way to share files among users, and became 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 DNS (Domain Name Service) 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 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. The access to the file sharing systems was then 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 sigficantly in comparison to the previous work [16]. For our observations and from proprietary observations on DSL customers, we conjecture that the unknown traffic is mostly made of peer-to-peer traffic.

Table II shows the minimum and maximum amounts of traffic each category contributes among the 32 traces used in this paper.

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 II
TRAFFIC VOLUME BREAKDOWN BY APPLICATION

The application mix is quite different from link to link. Figure 5 plots the average web traffic per link, and Figure 6 plots the average traffic of peer-to-peer and unknown traffic combined. In most traces web traffic represents more than 40% of the total traffic volume. This result is consistent with most prior traffic analysis studies [16], [17], [27]. 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%, and we see the emergence of peer-to-peer traffic which contributes almost 80% 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 on OC-12 traces. The monitored OC-48 links are inter-POP backbone links, and carry heavily aggregated traffic. This could explain the small variability amongst them. Our observations show that peer-to-peer traffic has become one of the two most dominant applications in the network along with the web traffic, and its emergence is not limited to a certain type of links.

Another important observation is that streaming applications are a stable component of the traffic, if not much in volume yet as the peer-to-peer applications. We observe 1 to 6% 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% of the traffic is due to TCP, even on the links with a significant percentage of streaming media. This is due to the fact that firewalls have encouraged the use of TCP rather than UDP for streaming media.

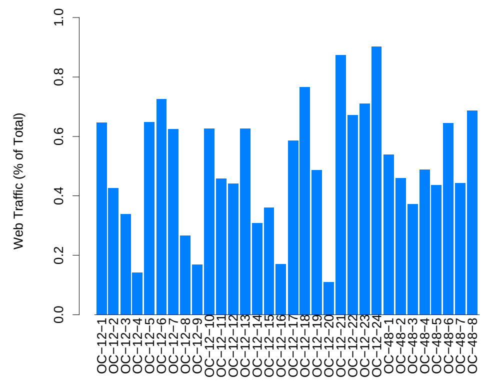

Fig. 5. Average volume of web traffic vs. traces

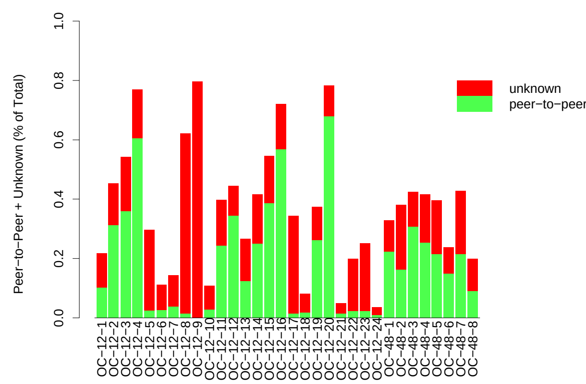

Fig. 6. Average volume of peer-to-peer traffic vs. traces

B.3 Traffic load in flows

Now we consider the traffic volume in flows per minute. A flow is defined by the 5-tuple {protocol type, source IP address, source port, destination IP address, destination port}. 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 packet with the same 5-tuple for 60 seconds. The 60 second timeout has been chosen based on previous work by Claffy [28] and on our own observations [8]. A day-long analysis of the same traces used in Figure 3 is presented in Figure 7. For all the traces, the average number of flows per minute is plotted in Figure 8.

The main observation is that peaks in the number of flows in Figure 7 do not necessarily translate in volume peaks of Figure 3. Between 9 am and 11 am on July 24th, 2001, the number of flows is as large as that during the peak hours between noon and 5 pm. During the same time period, the traffic volume is often just half of that during the peak hours between noon and 5pm. OC-12-7 and OC-12-8 traces are from a link to a CDN (Content Distribution Network[9]) customer. The discrepancy in volume and

[9] A CDN is a mechanism to improve web content delivery to end users.

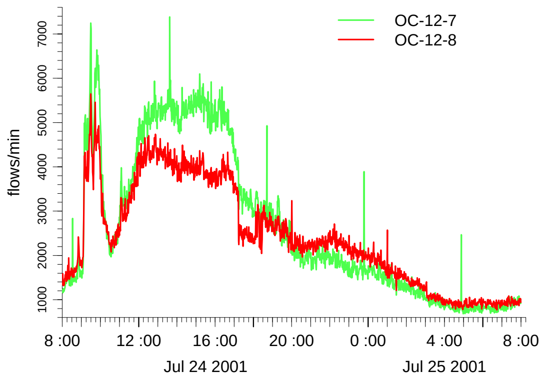

Fig. 7. Time-series plot of number of flows per minute

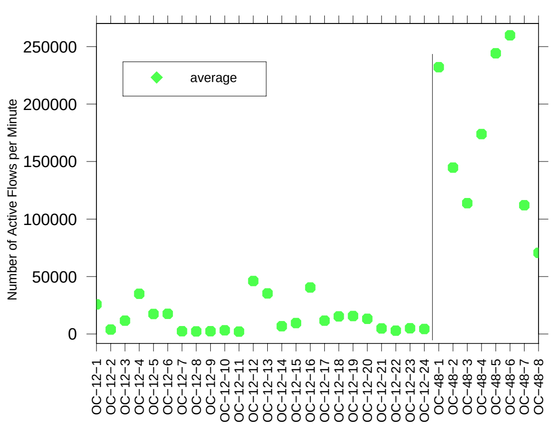

Fig. 8. Average number of flows per minute vs. traces

flow numbers is another example of the asymmetry in Section IV-B.1. We also observe a small number of occasional conspicuous peaks in flow numbers. Those peaks correspond to activities between servers of the CDN customer. However, they do not cause sudden increase in utilization in Figure 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 than 300,000 for all OC-48 links in Figure 8. In one 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 minute interval with the maximum number of flows of that specific trace revealed that it was likely due to a DoS attack with randomly spoofed source addresses. In the rest of the traces the maximum numbers of active flows are 1.1 to 4 times larger than the average numbers.

The result in Figure 8 is important as it demonstrates per-flow scheduling is feasible in hardware on access links. This observation means that new venues in traffic control can be explored, and that routers can go beyond TCP fairness and Active Queue Management[10].

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

B.4 Packet size distributions

Prior work has shown that the packet size distribution is tri-modal [16]. This was a result of a combination of TCP acknowledgements and the existence two distinct default message tranmission unit (MTU) sizes. Figure 9 demonstrates this tri-modal packet size distribution for two traces, OC-12-1 and OC-12-2. 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 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 in Figure 9, OC-12-10, exhibit more than three steps: 211 and around 820. The 211 byte packets correspond to a proprietary UDP application which carries a single 211 byte packet. Most 845 byte packets are from DNS (domain name service). 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 the periodic re-examination of assumptions about the distribution of packet sizes on an IP backbone network.

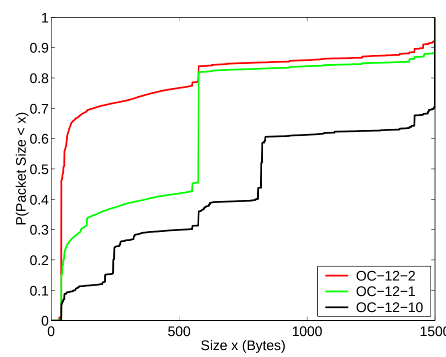

Fig. 9. Packet size cdf

C. *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 round-trip-time (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 [29]. Note that the rtt is measured end-to-end, i.e. 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 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 in 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 it would generally be possible to gather with active end-to-end measurements.

Figure 10 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 explainable 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 50ms. The traffic on these links is primarily from content distribution networks (CDNs). This is consistent with the results of Krishnamurthy et al that CDNs improve the overall response time of customer requests [30].

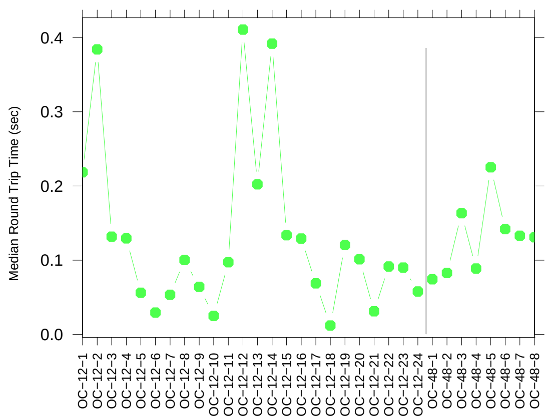

Fig. 10. Median round-trip time vs. traces

Figure 11 shows the rate of out-of-sequence packets for TCP flows defined by the 5-tuple as above. Possible causes of out-of-sequence packets are: retransmission after loss, unnecessary retransmission, duplicates, and reordering. Jaiswal et al. reports 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 end-to-end loss of a flow, it provides an upper bound on the number of losses we can detect from our measurements[11].

In Figure 11 we see that in all traces 90% of the flows experience no out-of-sequence packets; in only a handful of traces is the 99th percentile above 30% out-of-sequence. The maximum out-of-sequence packet rate often reaches above 90%, but this may be a result of short flows losing

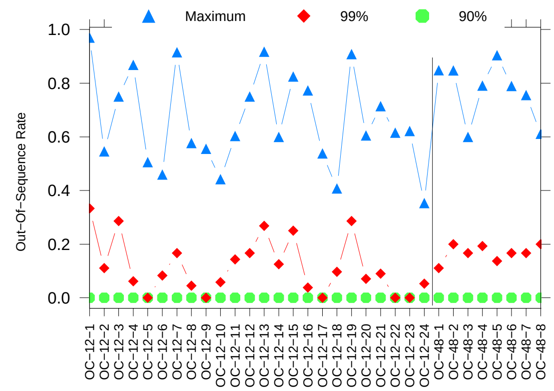

Fig. 11. Out-of-sequence rate vs. traces

most of their packets and reporting a high loss rate. The fact that the 90% of flows experience out-of-sequence rate of 0% on all the monitored links shows that most TCP flows experience no end-to-end loss.

D. Delay measurements

The accurate knowledge of packet delay characteristics is important, since delay is a major metric in the definition of Service Level Agreements (SLA). 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 (such as VoIP). Furthermore, many of the active probes use ICMP packets which 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 global synchronization 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 in our backbone. By monitoring links entering and exiting a single router we can measure the detailed queuing behavior of the router. By monitoring links in different geographic locations we can measure the queuing behavior of the backbone.

Our previous work on single-hop delay shows that the delay distribution through a single router is usually long tailed [24]. By understanding the traffic scaling behavior and its impact on the delay, we can approximate the delay distribution, and use the knowledge in provisioning the network [31], [32].

Obtaining delay distributions through multiple POPs is more challenging than single-hop delay distribution. Between two links on the opposite sides on the backbone network, there are often multiple paths that packets can

[11] 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 so that it does not affect our results significantly.

traverse. However, not all the paths carry traffic between two links, but only those with the minimum cost calculated by the IGP (Interior Gateway Protocol). Thus 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 Figure 12 are obtained between San Jose and New York, and reflect 200 million packet matches [12]. Packets identified in these delay distributions crossed 5 IP POPs and 8 core routers.

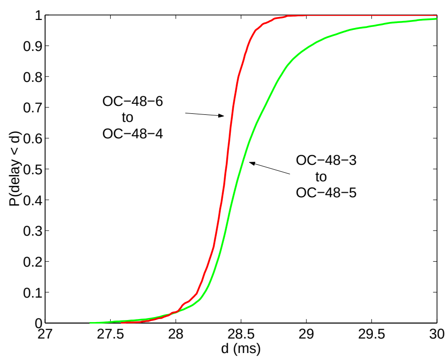

Fig. 12. Delay distributions

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% 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% of packets experienced less than 31 ms delay, we observe a very small number of packets that experienced delay above 100 ms. In [24] router idiosyncracies are identified as a cause behind large delays.

The analysis of the delay distributions reveals two major characteristics of the backbone. First, transmission delays are currently dominated by the speed of light. Second, the jitter is extremely low and there is no obstacle in deploying delay sensitive applications on the backbone.

V. Conclusions

We described 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, there is the capability of simultaneously collecting fine granularity information on multiple, geographicly 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 4 POPs on the Sprint IP backbone network, and collected days worth of traces. This paper presented a synthesis of the results from traces collected in July and September 2001 and April 2002. Interested readers are referred to `http://ipmon.sprintlabs.com` for additional results. Ongoing work is focussed on the deployment of the IPMON systems on OC-192 links and on upgrading the DAG card in order to add new filtering and sampling capabilities.

We observed that link load characteristics often vary from link to link and that these variations can often be correlated to the nature of the customers connected to the POP. As one might expect, as traffic becomes more highly aggregated, for example on OC-48 links backbone links, there is a higher degree of consistency. We also showed for the first time that some links no longer carry web traffic as their dominant component. File sharing and media streaming applications can represent up to 80% of the traffic volume. This new phenomenon can change significantly the nature of the Internet traffic and the way the Internet should be engineered. We also computed the number of active flows and showed that it is small enough to make per-flow queueing an appealing technology to control the traffic, and to 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 the light. Our result also show that the backbone is not an obstacle to the deployment of VoIP on the Internet.

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 which 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 embedded measurement facilities. 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.

VI. Acknowledgements

We would like to thank Konstantina Papagiannaki and Gianluca Iannaccone for helping us with constructing the figures; 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 in this article. Instead we choose to acknowledge their institutions: Sprint IE, Sprint IBO, Sprint ATL, the University of Waikato, and CAIDA.

References

[1] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford, and F. True, "Deriving traffic demands for operational IP networks: Methodology and experience," *IEEE/ACM Trans. on Networking*, June 2001.

[2] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot, "Traffic matrix estimation: existing techniques compared and new directions," in *Proc. of 2002 ACM SIGCOMM*, Pittsburgh, August 2002.

[3] C. Barakat, P. Thiran, G. Iannaccone, C. Diot, and P. Owezarski, "A flow-based model for Internet backbone traffic," in *Proc. of ACM SIGCOMM Internet Measurement Workshop*, Marseilles, France, November 2002.

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

[4] S. Bhattacharyya, C. Diot, J. Jetcheva, and N. Taft, "Pop-level and access-link-level traffic dynamics in a tier-1 pop," in *Proc. of ACM SIGCOMM Internet Measurement Workshop*, San Francisco, November 2001.

[5] U. Hengartner, S. Moon, R. Mortier, and C. Diot, "Detection and analysis of routing loops in packet traces," in *Proc. of ACM SIGCOMM Internet Measurement Workshop*, Marseille, France, November 2002.

[6] S. Jaiswal, G. Iannaccone, C. Diot, J. Kurose, and D. Towsley, "Measurement and classification of out-of-sequence packets in a tier-1 IP backbone," in *Proc. of ACM SIGCOMM Internet Measurement Workshop*, Marseilles, France, November 2002.

[7] C. Boutremans, G. Iannaccone, and C. Diot, "Impact of link failure on VoIP performance," in *Proc. of 12th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*, Miami, May 2002.

[8] G. Iannaccone, C. Diot, I. Graham, and N. McKeown, "Monitoring very high speed links," in *Proc. of ACM SIGCOMM Internet Measurement Workshop*, San Francisco, November 2001, pp. 267–272.

[9] V. Paxon, "End-to-end routing behavior in the Internet," *IEEE/ACM Trans. on Networking*, vol. 5, no. 5, October 1997.

[10] V. Paxon, "End-to-end Internet packet dynamics," *IEEE/ACM Trans. on Networking*, vol. 7, no. 3, June 1999.

[11] V. Paxon, J. Mahdavi, A. Adams, and M. Mathis, "An architecture for large-scale Internet measurement," *IEEE Communications*, vol. 36, no. 8, 1998.

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

[13] 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.

[14] M. Alves, L. Corsello, D. Karrenberg, C. Ögüt, M. Santcroos, R. Sojka, H. Uijterwall, and R. Wilhelm, "New measurements with the RIPE NCC test traffic measurements setup," in *Proc. of Passive and Active Measurement Workshop*, Fort Collins, Colorado, March 2002.

[15] J. Apsidorf, K. C. Claffy, K. Thompson, and R. Wilder, "OC3MON: Flexible, affordable, high performance statistics collection," in *Proc. of INET*, June 1997.

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

[17] K. C. Claffy, G. Miller, and K. Thompson, "The nature of the beast: Recent traffic measurements from an Internet backbone," in *Proc. INET'98*, Geneva, Switzerland, July 1998.

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

[19] "Juniper networks solutions for network accounting," http://-www.junipercom-techcenter-techpapers-200010.pdf, 2000, Juniper Networks white paper.

[20] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, and J. Rexford, "Netscope: Traffic enginnering for IP networks," *IEEE Network*, March/April 2000.

[21] 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.

[22] C. Fraleigh, C. Diot, B. Lyles, S. Moon, P. Owezarski, D. Papagiannaki, and F. Tobagi, "Design and deployment of a passive monitoring infrastructure," in *Proc. of Passive and Active Measurement Workshop*, Amsterdam, The Netherlands, April 2001.

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

[24] K. Papagiannaki, S. Moon, C. Fraleigh, P. Thiran, F. Tobagi, and C. Diot, "Analysis of measured single-hop delay from an operational backbone network," in *Proc. IEEE INFOCOM 2002*, New York, June 2002.

[25] K. Keys, D. Moore, R. Koga, E. Lagache, M. Tesch, and K. C. Claffy, "The architecture of coralreef: An Internet traffic monitoring software suite," in *Proc. of Passive and Active Measurement Workshop*, Amsterdam, April 2001.

[26] S. Iyer, S. Bhattacharyya, N. Taft, N. McKeown, and C. Diot, "A measurement based study of load balancing in an IP backbone," Tech. Rep. TR02-ATL-051027, Sprint ATL, Burlingame, California, May 2002.

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

[28] K. C. Claffy, H.-W. Braun, and G. C. Polyzos, "A parameterizable methodology for Internet traffic flow profiling," *IEEE Journal on selected areas in communications*, vol. 13, no. 8, pp. 1481–1494, October 1995.

[29] H. S. Martin, A. McGregor, and J. G. Cleary, "Analysis of Internet delay times," in *Proc. of Passive and Active Measurement Workshop*, Auckland, New Zealand, April 2000.

[30] B. Krishnamurthy, C. Wills, and Y. Zhang, "On the use and performance of content distribution networks," in *Proc. of ACM SIGCOMM Internet Measurement Workshop*, San Francisco, November 2001.

[31] C. Fraleigh, F. Tobagi, and C. Diot, "Provisioning IP backbone networks to support latency sensitive traffic," in *submitted for publication*, 2002.

[32] Z.-L. Zhang, V. Ribeiro, S. Moon, and C. Diot, "Small-time scaling behaviors of Internet backbone traffic: An empirical study," in *submitted for publication*, 2002.