Measurement and Analysis of Single-Hop Delay on an IP Backbone Network

Konstantina Papagiannaki†, Sue Moon†, Chuck Fraleigh⋆, Patrick Thiran‡, Christophe Diot†

†: Sprint ATL,
1 Adrian Court,
Burlingame, CA 94010, USA
{dina, sbmoon, cdiot}@sprintlabs.com

⋆: NetVmg, Inc.
47529 Fremont Blvd.
Fremont, CA 94538
cjf@netvmg.com

‡: LCA - I&C - EPFL,
CH-1015 Lausanne,
Switzerland
Patrick.Thiran@epfl.ch

***Abstract*— We measure and analyze the single-hop packet delay through operational routers in the Sprint IP backbone network. After presenting our delay measurements through a single router for OC-3 and OC-12 link speeds, we propose a methodology to identify the factors contributing to single-hop delay. In addition to packet processing, transmission, and queuing delay at the output link, we observe the presence of very large delays that cannot be explained within the context of a FIFO output queue model. We isolate and analyze these outliers.**

Results indicate that there is very little queuing taking place in Sprint's backbone. As link speeds increase, transmission delay decreases and the dominant part of single-hop delay is packet processing time. We show that if a packet is received and transmitted on the same linecard, it experiences less than 20 μs of delay. If the packet is transmitted across the switch fabric, its delay doubles in magnitude. We observe that processing due to IP options results in single-hop delays in the order of milliseconds. Milliseconds of delay may also be experienced by packets that do not carry IP options. We attribute those delays to router idiosyncratic behavior that affects less than 1% of the packets. Lastly, we show that the queuing delay distribution is long-tailed and can be approximated with a Weibull distribution with the scale parameter, $a = 0.5$, and the shape parameter, $b = 0.6$ to 0.82.

***Index Terms*— single-hop delay measurement, queuing delay, link utilization**

I. Introduction

Delay is a key metric in data network performance and a parameter in Internet Service Providers' (ISPs') Service Level Agreements. In the Internet, packets experience delay due to transmission and propagation through the medium, as well as queuing due to cross traffic at routers. The characteristics of the traffic have significant impact on the queuing delay. Willinger et al. first reported that network traffic is self-similar rather than Poisson [1], and much research has been done since to explore the consequences of non-Poisson traffic on queuing delay. The Fractional Brownian Motion (FBM) model has been proposed to capture the coarse time scale behavior of network traffic, and results in queuing behavior that diverges significantly from that of the Poisson traffic model [2], [3]. Follow-up work shows that the wide-area network traffic is multi-fractal and exhibits varying scaling behavior depending on the time scale [4]. Recent work reveals that the queuing behavior can be approximated differently depending on the link utilization [5].

The above analyses, however, have been based on packet traces collected from a single link and fed into an output buffer, whose size and service rate vary. We are not aware of any measurement of the queuing delay on operational routers. The difficulty in measuring single-hop delay in a real network is threefold:

- Packet timestamps must be accurate enough to allow the calculation of the transit time through a router. This requires in particular that the measurement systems (i) offer sufficient resolution to distinguish the arrival times of two consecutive packets, and (ii) are synchronized to an accurate global clock signal, such as Global Positioning System (GPS). These two conditions need to be met so that the maximum clock skew between any two measurement cards is limited enough to allow accurate calculation of the transit time of a packet from one interface to another interface of the same router.
- The amount of data easily reaches hundreds of gigabytes. Data from input and output links need to be matched to compute the time spent in the router.
- Routers have many interfaces; tapping all the input and output links to have a complete picture of the queuing behavior of any single output link is unrealistic in an operational network.

We have designed a measurement system that addresses the first two of the above difficulties, and deployed it in the Sprint tier-1 IP backbone network to collect packet traces with accurate timestamps [6]. We use optical splitters to capture and timestamp every packet traversing a link (see details in Section II). We obtain the single-hop delay of packets by computing the difference between the timestamps at the input and output monitored links. The third difficulty is not easy to overcome due to deployment cost and space issues. Although this prevents us from characterizing the queuing experienced by *all* packets, it does not affect the evaluation of the single-hop delays reported in this paper. In Section II we present delay measurements of one hundred million packets matched among more than four billion packets and 400 gigabytes of data collected from the Sprint IP backbone network. In Section III we provide a methodology for the quantification of the various elements in single-hop delay. We identify the impact that (i) transmission across the switch fabric, (ii) the

presence of IP options, and (iii) increased output link speed have on the delay experienced by packets through a single node. Surprisingly, in addition to the expected elements, such as transmission, queuing, and processing delays, we observe very long delays that cannot be attributed to queuing at the output link. We use a single output queue model to isolate these delays, and discuss their potential origins. Once the queuing delay component has been quantified, we analyze its tail behavior in Section IV. We summarize our findings in Section V.

II. Delay Measurement

We have designed passive monitoring systems that are capable of collecting and timestamping the first 44 bytes of all IP packets at link speeds up to OC-48 (2.5 Gbps), using the DAG card [7]. These monitoring systems have been deployed on various links in four Points of Presence (PoPs) of the Sprint IP backbone. We have collected day-long packet traces, and analyzed them off-line. Details about the measurement infrastructure can be found in [6].

The monitoring systems are GPS synchronized and offer a maximum clock skew of 6 μs. Details on the clock synchronization and possible errors in the accuracy of our delay measurements can be found in [8]. Consistency in the results obtained on more than 30 links, connected to different routers in all four Points of Presence, across multiple days, gives us confidence in the accuracy of the single hop delay measurements presented in this paper.

A. Collected Data

We tap into the optical fiber and capture packets just before they enter and right after they leave a router. We denote the packet arrival time at an input link as T_{in} and the packet departure at an output link, as T_{out}. For any given packet n, the single-hop delay through the router is the difference between its arrival and departure timestamps: $d(n) = T_{out}(n) - T_{in}(n)$. This single-hop delay value corresponds to the total time a packet spends in a router.

Packet traces from more than 30 links, both OC-3 and OC-12, have been analyzed. In this paper, we use packet traces from four OC-3 links, collected on August 9th, 2000, and four OC-12 links, collected on September 5th, 2001. Those link pairs have been selected because they exhibit the highest delays observed among all our measurements. All packet traces were collected on routers of the same manufacturer and of the same architecture, running the same operating system version. We label a router's inbound link as `in`, and a router's outbound link as `out`, and refer to them as a *data set* for the remainder of the paper. Table I provides further details about the eight traces analyzed throughout the paper.

Monitoring systems are attached to selected links inside a PoP. Seven out of the eight selected traces were collected on links attached to quad-OC-3 and quad-OC-12 linecards. Quad linecards accommodate four equal speed interfaces, as shown in Figure 1. Packets may transit the router from one linecard to another (`set1`, `set2`), or from one interface of a linecard to another interface of the same linecard (`set3`).

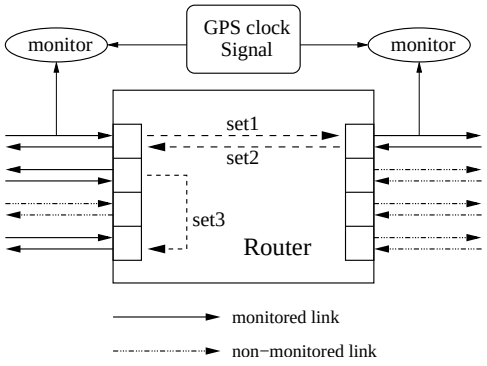


Fig. 1. Configuration of monitoring systems in a PoP.

In Table II, we present the architectural details of each data set collected. We denote each router participating in the measurements with its own index j. Data `set1` and `set2` were collected through the same router (Router1), and they correspond to the forward and reverse direction of the same router path (the incoming link of `set1` is the outgoing link of `set2`, and vice-versa). All data sets capture the behavior of the path between two quad linecards, with the exception of `set4`, that corresponds to the path between a quad-OC12 card and a single OC-12 card. Data `set1`, `set2`, and `set4` correspond to measurements involving two different linecards, whereas `set3` corresponds to measurements collected on the same linecard. In Section III-B we show how such architectural differences affect the delay values experienced by packets through a router.

B. Matching Packets

The first step in our methodology is to identify those packets that arrive on the input links and depart on the output links we monitor. We use hashing to match packets efficiently. The hash function is based on the CRC-32 algorithm [9]. Only 30 bytes out of the 44 bytes are hashed (including the source and destination IP addresses, the IP header identification number and the whole IP header data part). The other fields are not used since they may be modified by the router (e.g. TTL) or carry almost identical information in all IP packets (e.g. IP version field, TOS byte). Using the 24 least significant bits of the CRC-32 value, the hash function offers an average load factor of 5.7% when one million packets are hashed into a single table. We decided to use hash tables of one million packets, because one million average-sized packets transmitted at OC-3 speeds correspond to time periods larger than one second. We assume that one second is the maximum delay a packet can experience through a single node. The hash table size is increased to four million packets for the processing of the OC-12 traces for similar reasons.

To match packets, the traces are processed as follows: The first million packets from `out` are hashed into a table called H_1, and the timestamp of the last packet is recorded as $e(H_1)$. Then, in order of arrival, each packet from `in` is hashed and its key value is used as an index in H_1. If table H_1 contains a packet for that specific index, we compare *all* 44 bytes of the

Set	Link	Speed	Date	Start Time	End Time	# packets	Avg. Util.	# matches
1	in1	OC-3	Aug. 9, 2001	16:56:33 UTC	02:56:07 UTC	793,528,684	70 Mbps	2, 781,201
	out1	OC-3	Aug. 9, 2001	16:56:00 UTC	02:56:07 UTC	567,680,718	60 Mbps	
2	in2	OC-3	Aug. 9, 2001	16:56:03 UTC	17:41:04 UTC	28,213,976	30 Mbps	1, 175,674
	out2	OC-3	Aug. 9, 2001	16:56:04 UTC	17:41:04 UTC	48,886,948	50 Mbps	
3	in3	OC-12	Sep. 5, 2001	05:00:34 UTC	11:17:11 UTC	1,386,697,577	150 Mbps	17,613,103
	out3	OC-12	Sep. 5, 2001	05:00:34 UTC	11:17:11 UTC	1,116,885,094	250 Mbps	
4	in4	OC-12	Sep. 5, 2001	05:03:15 UTC	17:32:50 UTC	157,518,386	6 Mbps	70 ,423,140
	out4	OC-12	Sep. 5, 2001	05:03:15 UTC	17:32:50 UTC	169,006,605	6 Mbps	

TABLE I
DETAILS OF TRACES

Set	From	To	Router Name	Same Linecard	Different Linecard
1	quad-OC3	quad-OC3	Router1		√
2	quad-OC3	quad-OC3	Router1		√
3	quad-OC12	quad-OC12	Router2	√	
4	quad-OC12	OC12	Router3		√

TABLE II
ARCHITECTURAL DETAILS FOR THE ROUTERS WHERE THE TRACES WERE COLLECTED.

two packets. If they are the same, we have a match and we output a record of all its 44 bytes, along with the timestamps for its arrival on link `in` and departure on link `out`. This process continues until we reach a packet from `in` that has a timestamp one second or less than $e(H_1)$. Then we hash the next one million packets from `out` and create a second hash table H_2. Both H_1 and H_2 are used until the timestamp for a packet from `in` is greater than $e(H_1)$. When this happens, H_2 replaces H_1, and the processing continues.

Duplicate packets have been reported previously [10]. We occasionally observe them in our traces (they account for less than 0.001% of our packets), and have paid special attention to matching them. Duplicate packets have all 44 bytes identical, and therefore hash to the same value. In most cases we find that only after a packet left `out`, its duplicate arrived on `in`, making the classification unambiguous. We successfully match most duplicate packets with the correct arrival and departure timestamps. In other cases, we ignore the matches.

As a result of the above process, four traces of *matched* packets are produced. The numbers of matched packets are given in Table I. We use these traces in the next section to analyze the single-hop delay components.

III. DELAY ANALYSIS

We start with general observations on the delay measurements. We plot the empirical probability density function of the measured single-hop delay, and quantify step-by-step its contributing factors. The outcome of this step-by-step analysis is the derivation of the output queuing delay, which is analyzed in Section IV.

A. General Observations

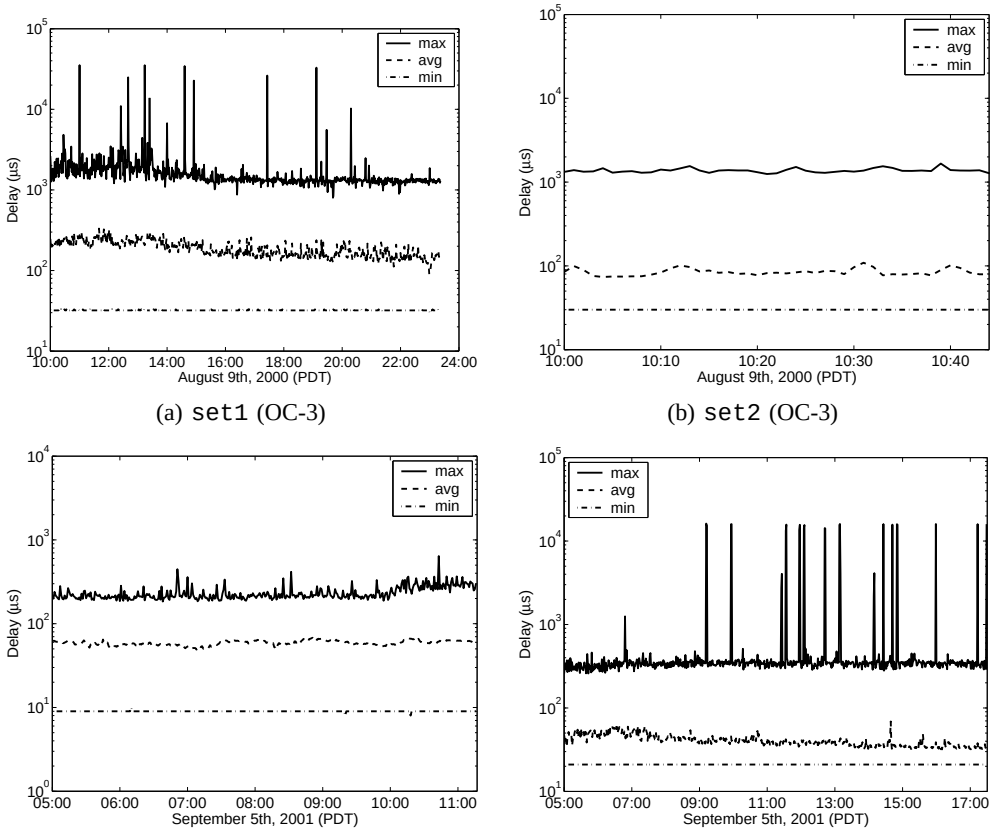
We denote the m-th matched packet as m, and the total number of matched packets for a given set by M. Figure 2 plots the minimum, average, and maximum values of the single-hop delay $\{d(m)\}$ across each one minute interval for all four data sets. We observe first that the minimum delay is stable throughout all the traces, while the average delay exhibits more oscillations and may drop as the link utilization decreases toward the evening. The minimum delay corresponds to the minimum amount of time a packet needs to go through a router. Therefore, given that the minimum delay is constant throughout the day, there is at least one packet that experiences no queuing in each one minute interval.

The maximum delay is more variable than the average delay. It shows occasional spikes of a few milliseconds reaching up to 35 ms for `set1` and 172 ms for `set4`. We also note that the maximum delay remains consistently above 1 ms for the OC-3 data sets, and 0.2 ms for the OC-12 data sets, even though the average delay decreases. We provide possible explanations in Section III-D.

B. Step-by-Step Analysis of the Single-Hop Delay

Figure 3 presents the empirical probability density function of $\{d(m)\}, 1 \leq m \leq M$, along with various statistics on the upper right corner of each plot. Average delay values are around 100 μs for the OC-3 data sets and decrease by a factor of four when the link speed increases to OC-12. We see that 99% of the packets experience less than 1 ms of delay on OC-3 links. For the OC-12 traces the 99th percentile of the single-hop delay distribution is below 100 μs. However, the observed maximum delay is data set specific, reaching up to 35 ms in `set1` and 172 ms in `set4`.

There are three distinct peaks at the beginning of each density function. Previous work by Thompson et al. reports that packets in the backbone do not have a uniform size distribution, but instead have three unique peaks at 40 to 44, at 552 to 576, and at 1500 bytes [11]. The sizes of 40 and 44 bytes correspond to minimum-sized TCP acknowledgment packets and *telnet* packets of a single key stroke; 552 and 576



(a) `set1` (OC-3) (b) `set2` (OC-3)

(c) `set3` (OC-12) (d) `set4` (OC-12)

Fig. 2. Minimum, average, and maximum delay per minute for the matched packets of all four data sets. The x-axis is adjusted to the duration of the data set. The y-axis spans between 10 μs and 100 ms for `set1`, `set2`, and `set4`. For `set3` the y-axis spans between 1 μs and 10 ms, given that delays are much lower than in the other three data sets.

byte packets correspond to default MTU sizes when path MTU discovery is not used by a sending host; and 1500 byte packets correspond to the Ethernet MTU size. In all data sets, more than 70% of the packets are 40, 576, and 1500 bytes long. We thus conjecture the three peaks at the beginning of the delay distribution to be related to the packet size. To verify this conjecture, we group the packets of those three sizes, and separately plot the empirical probability density function of the delay experienced by packets of the given size. Each distribution has a unique peak that matches one of the three peaks in Figure 3. We now identify and quantify the factors that contribute the same amount of delay for packets of the same size.

1) Transmission Delay on the Output Link: A first cause is the *transmission delay* on the output link. Transmission delay is proportional to the packet size and to the speed of the output link: l_m/C_{out}, where l_m is the length of the m-th matched packet, and C_{out} is the output link capacity[1]. We refer to the difference between the total delay of packet m and its transmission time on the output link as the *router transit time*, denoted by $d_{tx}^-(m)$: $d_{tx}^-(m) = d(m) - l_m/C_{out}$. The empirical probability density function of $d_{tx}^-(m)$ is plotted in Figure 4.

[1]Throughout this paper, for the OC-3 traces we set $C_{out} = 150.336\ Mbps$, which is the effective payload of POS OC-3. For the OC-12 traces $C_{out} = 601.344\ Mbps$.

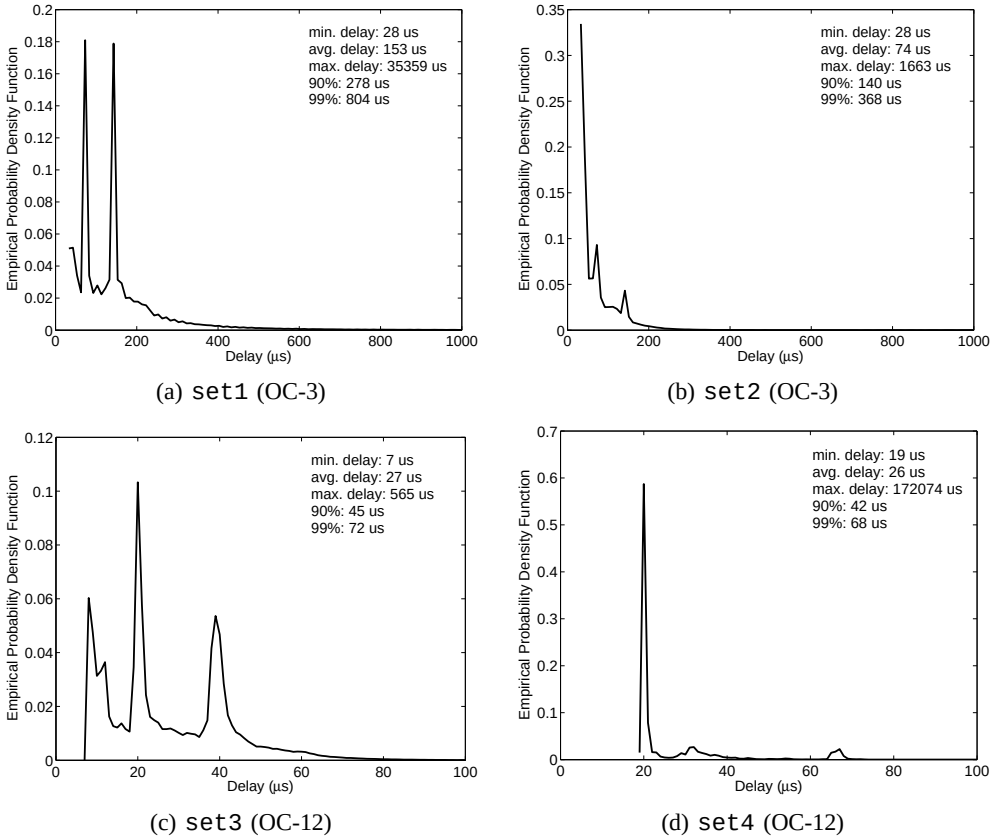
There still are three distinct peaks in the distribution, even though they are less pronounced than in Figure 3. This may indicate that there is still a part of the router transit time that depends on the packet size.

2) Minimum Router Transit Time: When a packet arrives at a router, its destination address is looked up in the forwarding table, the appropriate output port is determined, and the packet is then transferred to the output port. Routers in our network do store-and-forward, as opposed to cut through [12]. This operation imposes a minimum amount of delay on *every* packet, proportional to its size, which is likely to explain those remaining peaks in Figure 4. Below we quantify the minimum router transit time experienced by packets in our data sets.

We plot the minimum value of the router transit time for each packet size L, $d_{min}(L)$, in Figure 5.

$$d_{min}(L) = \min_{1 \le m \le M} \{d_{tx}^-(m) | l_m = L\}$$

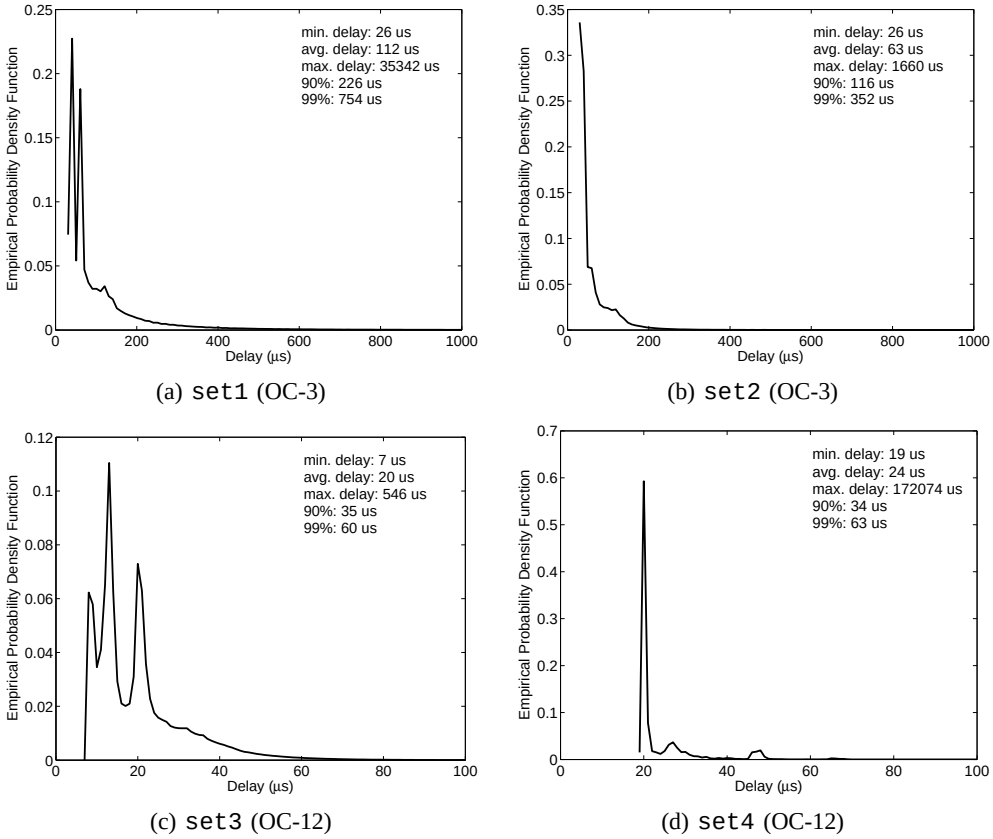
Figure 5 indicates that there exists a linear relationship between the two metrics. This relationship is made explicit through a linear regression. Given that all data sets feature an order of magnitude more packets for the size of 40, 576, and 1500 bytes, those three packet sizes are more likely to provide us with accurate minimum values for the router transit time. For this reason, we rely on the measurements for these three packet sizes in linear regression, and obtain Equation (1).



(a) `set1` (OC-3) (b) `set2` (OC-3)

(c) `set3` (OC-12) (d) `set4` (OC-12)

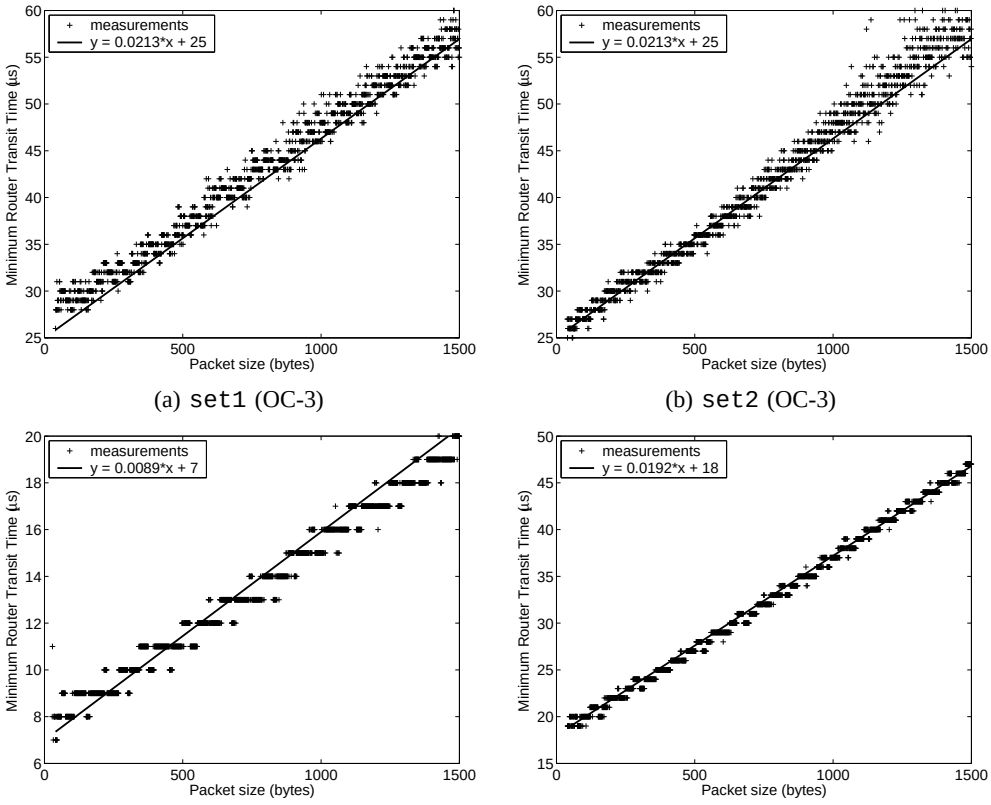
Fig. 3. Empirical probability density function of delay of matched packets $\{d(m)\}$.



(a) `set1` (OC-3) (b) `set2` (OC-3)

(c) `set3` (OC-12) (d) `set4` (OC-12)

Fig. 4. Empirical probability density function of router transit time, $d_{tx}^{-}(m) = d(m) - l_m/C_{out}$.



(a) `set1` (OC-3) (b) `set2` (OC-3)

(c) `set3` (OC-12) (d) `set4` (OC-12)

Fig. 5. Minimum router transit time versus packet size L: $\min_m d_{tx}^-(m)$ for $m \in \{i | l_i = L\}$. Each figure contains one value, for the minimum router transit time, for each packet size observed in our data. Delay granularity is 1 μs.

$$d_{min}(L) = \begin{cases} 0.0213 \cdot L + 25, & \text{for } \texttt{set1/set2} \\ 0.0089 \cdot L + 7, & \text{for } \texttt{set3} \\ 0.0192 \cdot L + 18, & \text{for } \texttt{set4} \end{cases} \quad (1)$$

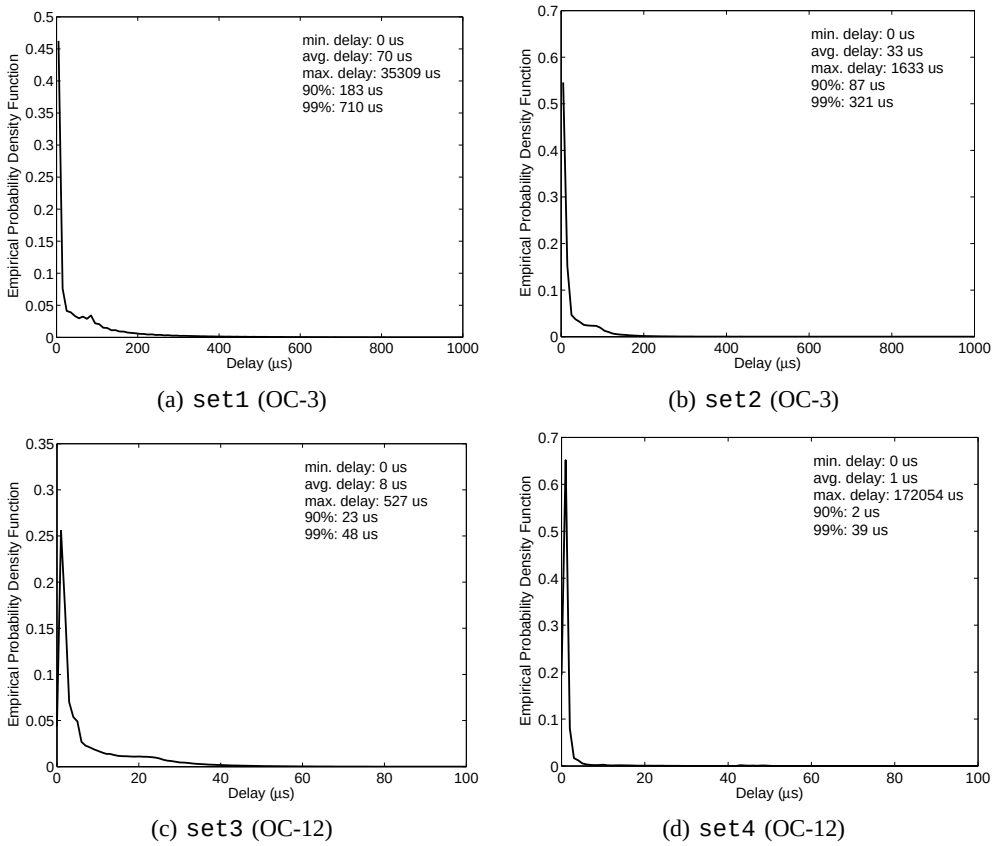
The linear relationship between minimum router transit time and packet size is consistent for the OC-3 data sets, and differs for the OC-12 data sets. We notice that for `set3` and `set4` we need two different equations to express the relationship between the minimum router transit time and the packet size. The reason for that is that packets that are received and transmitted on the same linecard exhibit a different behavior compared to the packets that need to transit the switch fabric. From Equation (1) we can identify the effect that such features of the router architecture may have on the packet delay.

According to Equation (1), transmission of packets across different linecards for OC-3 and OC-12 rates (`set1`, `set2`, and `set4`) leads to similar values for the slope capturing the linear relationship between packet size and minimum router transit time. However, the constant term is different and larger for the OC-3 case. This could be attributed to the fact that `set4` does not involve two quad linecards (Table II), or that OC-12 linecards utilize more recent technology, and thus may be offering faster packet processing than OC-3 linecards.

One important result derived from Equation (1) and Figure 5 is that packets which remain in the same OC-12 linecard are served much faster, i.e. in 7 μs to 20 μs. Packets that have to be transmitted across the switch fabric are served in 19 μs to 39 μs. Similar analysis performed on other data sets containing packets received and transmitted in the same quad linecard or across different linecards led to results consistent with this finding.

Subtracting $d_{min}(l_m)$ from the router transit time, $d_{tx}^-(m)$, we obtain the actual amount of time packets have to wait inside the router. The new empirical probability density function is presented in Figure 6.

Packet size related peaks have now disappeared and the delay distributions look similar for all data sets. The distribution is characterized by very low delays: 45% of the packets in `set1` and more than 50% of the packets in `set2` experience zero queuing delay. For the OC-12 data sets almost 30% of the packets in `set3` and 70% of the packets in `set4` go through the router without any queuing at the output link. Differences in the average delay can be explained by the packet size distribution of the data sets: `set1` and `set3` are dominated by packets larger than 500 bytes, while `set2` and `set4` contain mostly 40 byte packets. In addition, `set1` and `set3` consist of highly utilized links, thus featuring higher queuing delay values than `set2` and `set4`. Small peaks around 100 μs for the OC-3 data sets and 20 μs for the OC-12 data sets correspond to the transmission of a maximum sized packet at the respective line rate; thus accounting for the fact that a packet may arrive at the output queue and find it occupied by another, possibly maximum-sized, packet. The 99th percentile



(a) `set1` (OC-3) (b) `set2` (OC-3)

(c) `set3` (OC-12) (d) `set4` (OC-12)

Fig. 6. Empirical probability density function of $(d_{tx}^{-}(m) - d_{min}(l_m))$.

delays are very small; below 750 μs for the OC-3 data sets, and below 50 μs for the OC-12 data sets. Nevertheless, the maximum delay still reaches 172 ms for `set4` and 35 ms for `set1`.

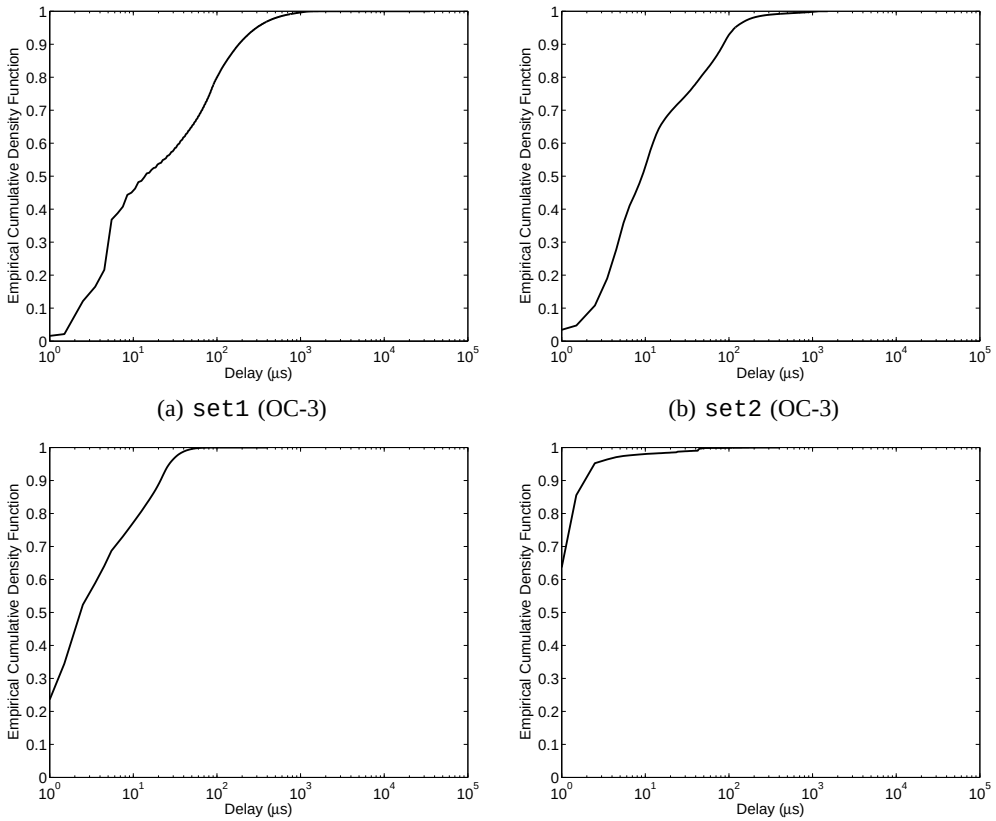
C. Possible Causes for Very Large Delay

In Figure 7, we present the cumulative distribution function (cdf) for the output queuing delay $((d_{tx}^{-}(m) - d_{min}(l_m)))$ observed for all four data sets. A key observation is that across all data sets the tail of the delay distribution is very long, accounting for the presence of very large delays. However, an examination of the output link data when the very large delays were observed shows that the link was not fully utilized while those packets were waiting. Therefore part of the long delays is not due to queuing at the output link. In the remainder of this section, we look into possible explanations for these large delay values.

One possible reason could be that the monitoring systems lose synchronization. We exclude measurement equipment fault as a cause for large delays for the following reasons. If the two measurement systems had gone out of time synchronization, the minimum and average delay in Figure 2 would exhibit a level shift over time, which is not visible. There is no way to tell if the system's software had a bug and produced the very large delays. However, it is extremely unlikely that a software bug affected only a handful of packets, still maintaining the strictly increasing nature of timestamps and keeping the minimum packet delay constant, both of which we checked in our traces.

A second reason, that can be easily verified, is that the packets experiencing long delays contain IP options. Most routers are designed to optimize the performance for the majority of packets. IP packets with options require additional processing in the protocol stack, and travel through a slower software path than packets without options. IP option packets are present in `set2`, `set3`, and `set4`. In Table III, we include the main statistics of the delay distribution derived from packets carrying IP options. Results indicate that packets with IP options spend at least 36 μs inside the router, and they usually account for the maximum delay in our single-hop delay distributions. The derived statistics should only serve as an indication for the magnitude of delay that packets with IP options may face while traversing a router, since the observed sample is too small to allow for generalization. Given that delay measurements for packets carrying IP options are not solely due to queuing, we do not include them in the remainder of our analysis.

Once IP option packets have been removed from our data sets, we find that the maximum delay for `set2` and `set4` drops significantly (Table IV). Due to the fact that there is a very small number of IP option packets present in our measurements, none of the other statistics of the distribution have significantly changed. Packets carrying IP options are capable of justifying the maximum delay in our data sets, but



(a) `set1` (OC-3) (b) `set2` (OC-3)

(c) `set3` (OC-12) (d) `set4` (OC-12)

Fig. 7. Empirical cumulative density function of $(d_{tx}^{-}(m) - d_{min}(l_m))$.

	`set2` 9 matches			
(μs)	minimum	average	median	maximum
$d_{tx}^{-}(m)$	242	453	307	1,659
	`set3` 21 matches			
(μs)	minimum	average	median	maximum
$d_{tx}^{-}(m)$	36	225	273	438
	`set4` 39 matches			
(μs)	minimum	average	median	maximum
$d_{tx}^{-}(m)$	270	11,219	320	172,074

TABLE III
DELAY STATISTICS FOR THE PACKETS THAT CARRY IP OPTIONS.

even after their removal the maximum delay experienced by packets in `set1` and `set4` remains in the order of tens of milliseconds. Other potential reasons behind the very large delay values are: (i) routers stopping forwarding packets for a short period of time when busy with some other CPU intensive task, (*e.g.* routing table updates, SNMP requests, and garbage collection in memory), an effect usually referred to as a "coffee break", (ii) router interface cards with multiple ports or backplane switch fabrics that could allow input or output blocking [13], (iii) memory locks or poor scheduling, etc.

D. Filtering Based on a Single Output Queue Model

When packets arrive at a router, they contend for resources to be forwarded to the destination output interface. The router can use various policies to resolve this contention. The FIFO (First-In First-Out) output queue model captures the essence of how a router should serve packets contending for the same resource in a best-effort fashion [12]. Thus, we model an output port of a router as a single output queue. While a single output queue is not an accurate model of all the operations performed in the router, it is sufficient to allow us to determine

	original set 2,781,201 matches					non IP-options set 2,781,201 matches				
in1 to *out1* (μs)	min.	avg.	90%	99%	max.	min.	avg.	90%	99%	max.
$d_{ix}^{-}(m)$	26	112	226	754	35,342	26	112	226	754	35,342
$d_{ix}^{-}(m) - d_{min}(l_m)$	0	70	183	710	35,309	0	70	183	710	35,309
	original set 1,175,665 matches					non IP-options set 1,175,656 matches				
in2 to *out2* (μs)	min.	avg.	90%	99%	max.	min.	avg.	90%	99%	max.
$d_{ix}^{-}(m)$	26	63	116	352	1,660	26	63	116	352	1,547
$d_{ix}^{-}(m) - d_{min}(l_m)$	0	33	87	321	1,633	0	33	87	321	1,520
	original set 17,613,183 matches					non IP-options set 17,613,162 matches				
in3 to *out3* (μs)	min.	avg.	90%	99%	max.	min.	avg.	90%	99%	max.
$d_{ix}^{-}(m)$	7	20	35	60	546	7	20	35	60	546
$d_{ix}^{-}(m) - d_{min}(l_m)$	0	8	23	48	527	0	8	23	48	527
	original set 70,423,140 matches					non IP-options set 70,423,101 matches				
in4 to *out4* (μs)	min.	avg.	90%	99%	max.	min.	avg.	90%	99%	max.
$d_{ix}^{-}(m)$	19	24	34	63	172,074	19	24	34	63	16,091
$d_{ix}^{-}(m) - d_{min}(l_m)$	0	2	2	39	172,054	0	2	2	39	16,045

TABLE IV

STATISTICS FOR THE OC-3 AND OC-12 DATA SETS AFTER THE REMOVAL OF PACKETS WITH IP OPTIONS.

if the delay of a packet is due to output queuing or not, using *only* the measurements we have at our disposal.

In the routers deployed in our network packet processing is heavily pipelined so that a packet experiencing the minimum router transit time should not introduce extra queuing for the next packet arriving at the input port. That is, the minimum router transit time simply delays a packet's arrival time at the output queue, without affecting other packets. We can thus assume that the m-th packet arrives at the output queue at $T_{in}^{'}(m) = T_{in}(m) + d_{min}(l_m)$, and set the service rate of the single output queue to the transmission rate of the output link, as illustrated in Figure 8.

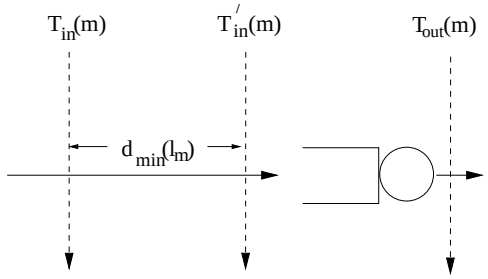


Fig. 8. Single output queue model of a router.

We expect a packet to wait at the output queue *if and only if* the output queue is busy serving other packets. The waiting time of a packet is $T_{out}(m) - l_m/C_{out} - T_{in}^{'}(m)$. In Figure 9 we plot the number of bytes transmitted at the output link `out1` during the time interval of $[T_{in}^{'}(m), T_{out}(m) - l_m/C_{out}]$ versus the size of the interval for `set1`[2]. The top line

[2]Similar behavior is observed for the other three data sets and is omitted due to space limitations.

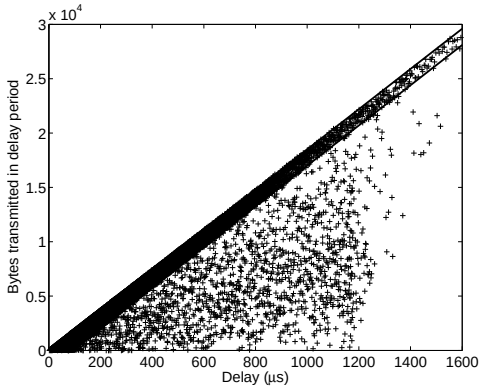


Fig. 9. Number of bytes transmitted between $T_{in}^{'}(m)$ and $T_{out}(m) - l_m/C_{out}$ on `out1`.

corresponds to the case when the number of bytes transmitted between the packet's arrival and departure time ($T_{in}^{'}(m)$ and $T_{out}(m) - l_m/C_{out}$) is *equal* to the number of bytes that would be transmitted if the link continuously sent packets at line rate. We observe that all data points lie below this top line. Moreover, most of the points that fall off this line are bounded by another line below, of the same slope, which allows for the transmission of one less maximum-sized packet. This latter line is described by $y = (C_{out} \cdot x)/8 - 1500$, where x is the size of the time interval in μs, and y is the number of bytes seen at the output link. We allow one maximum-sized packet as the error margin in our waiting time calculation, since the accuracy of the timestamps, the non-uniform distribution of SONET overhead in the signals, and the uncertainty about operations inside the router are likely to affect our computation. Those packets whose waiting times lie between the two lines are interpreted as follows: while a matched packet is waiting to

be transmitted between $T_{in}^{'}(m)$ and $T_{out}(m) - l_m/C_{out}$, the output link is fully utilized. We consider as the *filtered* data set those packets that lie between the two bounding lines in Figure 9. For `set1` the filtered set contains 94% of the total number of packets in the set. Other packets are considered to have experienced delay not due to output queuing[3] beyond the error margin and are filtered out. To evaluate the magnitude of the delay values that get filtered out by our simple output queue model, we proceed as follows.

We compute the amount of delay that each packet should have experienced in all four data sets according to the observed output link utilization. We then subtract the computed delay value from the actual delay value measured. The difference between those two values corresponds to the amount of additional delay that a packet experienced inside the router. In Figure 10(a) and Figure 10(b), we present the empirical probability and cumulative density function for the difference in delay experienced by the set of packets that got filtered out.

Figure 10(a) shows that the part of our delay measurements that cannot be explained according to our single output queue model may reach up to tens of milliseconds. An important observation is that for `set1` and `set2` the empirical probability density function shows a plateau between 10 μs and 1 ms. For `set4`, the plateau area spans between 10 μs and 200 μs. This behavior is consistent with a "coffee break" effect. When a router goes into a "coffee break", it stops forwarding packets for the duration of the break. Therefore, all packets that arrive at the router during this period have to wait until the router resumes the forwarding process, and therefore experience delays that may reach the duration of the break. In our case, the observed behavior resembles a "coffee break" of 1 ms in our OC-3 measurements, and 200 μs in our OC-12 measurements. What is interesting is that no such effect is evident for `set3`, where the maximum delay difference is limited to 17 μs. Recall that `set3` corresponds to delay measurements taken inside the same quad-OC-12 linecard. Therefore, such a finding could be an indication that the "coffee break" effect does not take place at the linecards themselves. Unfortunately, seeking explanation for such a phenomenon requires detailed knowledge of the router architecture, which constitutes proprietary information. Therefore we can only conjecture about possible reasons behind this behavior. Justification for the existence of delay discrepancies larger than 1 ms is even harder to provide. Queuing taking place at the input link and contention for access to the fabric switch could be possible explanations, but cannot be verified solely based on our measurements. In any case, as can be seen from Figure 10(b), such a phenomenon affects a very small number of packets, namely between 20% and 40% of the filtered out packets. This percentage corresponds to less than 1% of the total number of packets in each data set[4].

Given that delays experienced by packets beyond our error margins are not related to queuing at the output link, we continue our analysis only with the filtered data sets. We summarize the statistics for the router transit time and queuing delay for the original and filtered data sets in Table V. The average, 90th, and 99th percentile values of the delay distribution are all lower for the filtered data sets. Moreover, all of the delays larger than 5 ms in `set1` and `set4` have disappeared, and the maximum delay has dropped to 3.9 ms and 160 μs respectively. On the other hand, the maximum delay for `set2` and `set3` remains unchanged, indicating that the output link was fully utilized when the maximally delayed packet was being held back from transmission. We plot the minimum, average, and maximum values of the filtered delays for all four data sets in Figure 11. We compare to Figure 2 and notice that the maximum delay does not stay over 1 ms throughout the entire day. Consequently, the single queue model is effective in filtering the delays which are not due to queuing at the output link.

IV. Queuing Delay Tail Behavior

In this section, we analyze the tail of the queuing delay distribution. This analysis will help us identify possible models for the queuing delay in the backbone that could be exploited in simulation environments. We show that our results agree with previous analytical findings described in [2].

Tail behavior can be categorized into three types: light tailed, long tailed, and heavy tailed. A *light tailed* distribution has a probability density function whose tail approaches zero at least as rapidly as an exponential distribution. A distribution is said to have a *heavy tail* if $P[X > x] \sim kx^{-a}$ as $x \to \infty$, $0 < a < 2$ [14]. This means that regardless of the distribution for small values of the random variable, if the asymptotic shape of the distribution is hyperbolic, the distribution is heavy tailed. The simplest heavy tailed distribution is the Pareto distribution which is hyperbolic over its entire range and has a probability mass function $p(x) = ak^a x^{-a-1},\ a, k > 0,\ x \geq k$, where k represents the smallest value the random variable can take. *Long tailed* distributions decay slower than an exponential, without being heavy tailed. Lognormal and Weibull[5] distributions with the shape parameter $b < 1$ belong to long tailed distributions.

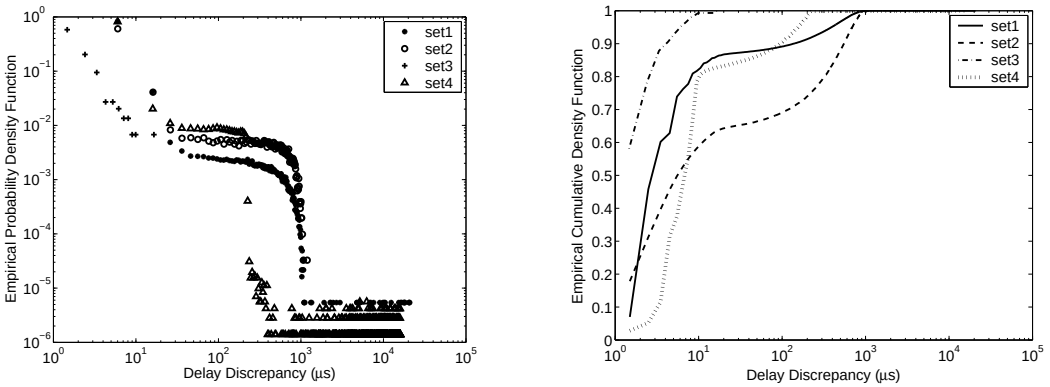
The network traffic is known to be long-range dependent, and such traffic can be modeled as Fractional Brownian Motion (FBM) [4]. Norros shows that the queuing delay distribution of the FBM traffic is approximated by a Weibull distribution [2].

To examine what tail category our delay distributions fall into, we first plot the complementary cumulative distribution function (CCDF) of $(d_{tx}^{-}(m) - d_{min}(l_m))$ in log-log scale for the first hour of the filtered sets, where link utilization remains approximately constant (Figure 12). If the tail of the CCDF forms a straight line, then the distribution may be heavy tailed. From Figure 12, it is not clear whether this is the case for our data sets. We use the `aest` tool to formally check if the queuing delay distribution is heavy tailed [15]. The obtained

[3]Strictly speaking, transmission and propagation delays are not due to queuing as well. However, we limit the use of non-queuing delay only to the delay experienced at the output queue.

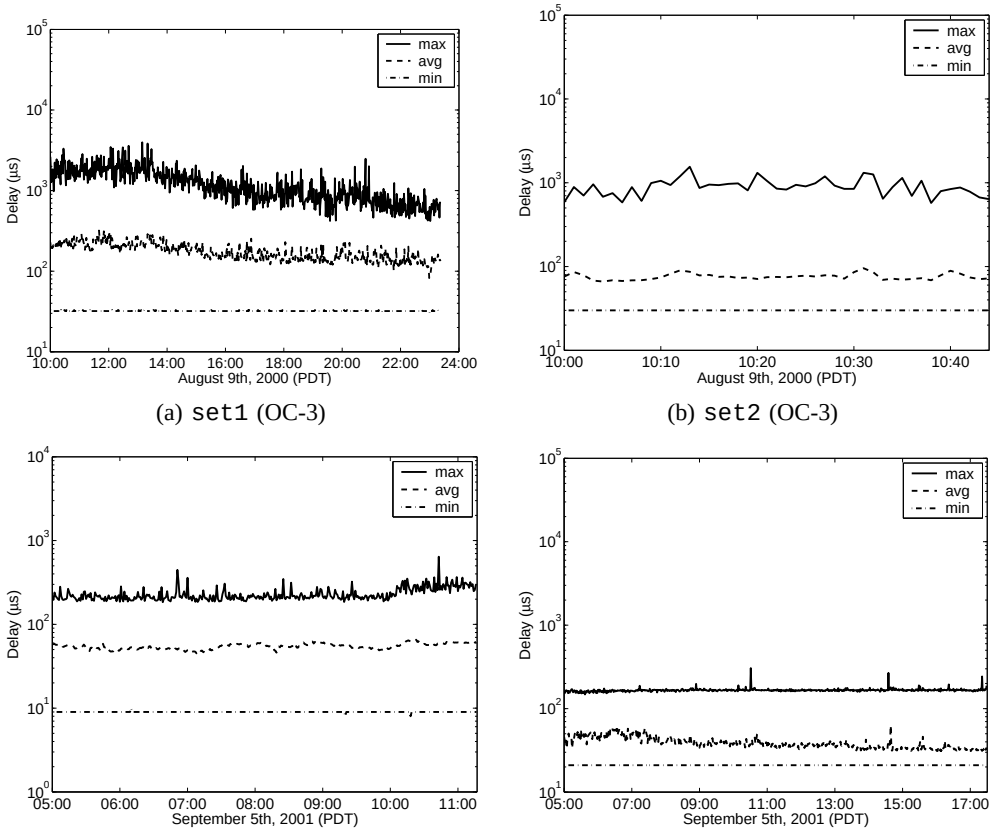
[4]The final percentage of packets that get filtered out is higher than 1% because of small delay discrepancies, below 10 μs.

[5]The probability density function of a Weibull distribution is given by $f(x) = \frac{bx^{b-1}}{a^b} e^{-(\frac{x}{a})^b}$, with $a > 0,\ b > 0$; α is called the scale parameter, while b is called the shape parameter.



(a) Empirical Probability Density Function (log-log) (b) Empirical Cumulative Density Function (log-normal)

Fig. 10. Unexplained part of the delay in the measurements that get filtered out.



(a) `set1` (OC-3) (b) `set2` (OC-3)

(c) `set3` (OC-12) (d) `set4` (OC-12)

Fig. 11. Minimum, average, and maximum single-hop delays per minute for the filtered packets of all four data sets. The x-axis is adjusted to the duration of the data set. The y-axis spans between 10 μs and 100 ms for `set1`, `set2`, and `set4`. For `set3` the y-axis spans between 1 μs and 10 ms, given that delays are much lower than in the other three data sets.

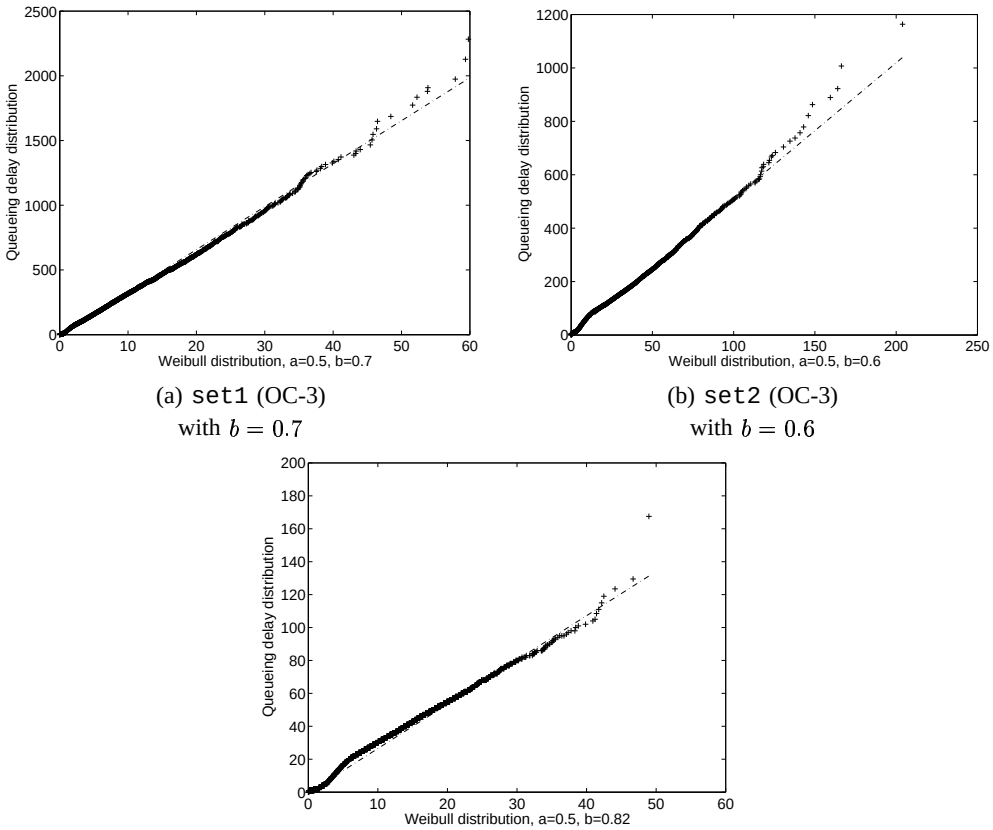
results indicate that our delay distributions do not have the power-law tail like the Pareto distribution, and are *not heavy tailed*.

We then look into whether our queuing delay distributions are long-tailed. As already mentioned, a Weibull distribution with a shape parameter β less than 1 belongs to the long-tailed distributions. We fit a Weibull distribution to our queuing delay distributions, and present our results in Figure 13 for the first three data sets. `Set4` is omitted because it is characterized by input and output link utilization of less than 10 Mbps out of the 622 Mbps of the link's capacity. As a consequence, the respective queuing delay distribution is characterized by a 99th percentile equal to 15 μs. This means that the number of samples we have in the tail of this particular distribution is very limited. Moreover, given their magnitude, the sample values are sensitive to our clock accuracy and our 1 μs granularity. Figure 13 shows that the queuing delay

	original set 2,781,201 matches					filtered set 2,596,486 matches (6.6% filtered out)				
in1 to *out1* (μs)	min.	avg.	90%	99%	max.	min.	avg.	90%	99%	max.
$d_{tx}^{-}(m)$	26	112	226	754	35,342	26	106	217	603	3,937
$d_{tx}^{-}(m) - d_{min}(l_m)$	0	70	183	710	35,309	0	65	174	558	3,903
	non IP-options set 1,175,665 matches					filtered set 1,145,170 matches (2.5% filtered out)				
in2 to *out2* (μs)	min.	avg.	90%	99%	max.	min.	avg.	90%	99%	max.
$d_{tx}^{-}(m)$	26	63	116	352	1,547	26	56	112	230	1,547
$d_{tx}^{-}(m) - d_{min}(l_m)$	0	33	87	321	1,520	0	27	82	200	1,520
	non IP-options set 17,613,183 matches					filtered set 17,613,018 matches (0.0009% filtered out)				
in3 to *out3* (μs)	min.	avg.	90%	99%	max.	min.	avg.	90%	99%	max.
$d_{tx}^{-}(m)$	7	20	35	60	546	7	20	35	60	546
$d_{tx}^{-}(m) - d_{min}(l_m)$	0	8	23	48	527	0	8	23	48	527
	non IP-options set 70,423,140 matches					filtered set 69,710,887 matches (1% filtered out)				
in4 to *out4* (μs)	min.	avg.	90%	99%	max.	min.	avg.	90%	99%	max.
$d_{tx}^{-}(m)$	19	24	34	63	16,091	19	24	33	49	362
$d_{tx}^{-}(m) - d_{min}(l_m)$	0	2	2	39	16,045	0	1	2	16	160

TABLE V

STATISTICS FOR THE OC-3 AND OC-12 DATA SETS BEFORE AND AFTER FILTERING.



(a) `set1` (OC-3) with $b = 0.7$

(b) `set2` (OC-3) with $b = 0.6$

(c) `set3` (OC-12) with $b = 0.82$

Fig. 13. Quantile-Quantile plot of the queuing delay distribution against a Weibull distribution.

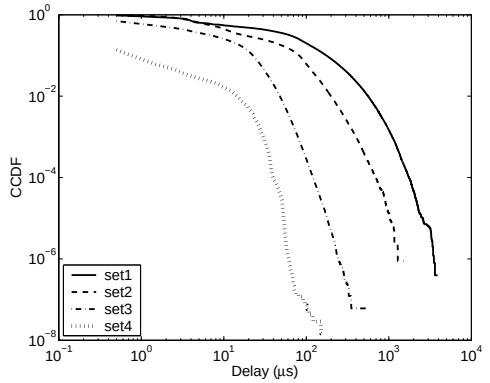


Fig. 12. Log-log plot of CCDF for the queuing delay measured for all four data sets (data set 1, 2: OC-3, data set 3, 4: OC-12)

distribution for `set1` and `set2` fits to a Weibull distribution with a shape parameter b equal to 0.6, and 0.7 respectively. The OC-12 distribution for queuing delay inside the same linecard (`set3`) can be approximated with a Weibull distribution with a shape parameter b equal to 0.82. Therefore, the distribution of queuing delay is *long tailed*, confirming the finding in [2].

We further sort the three data sets in order of increasing output link utilization, i.e. `set2`, `set1` and `set3` (Table I). We notice that data sets characterized by higher output link utilization are also characterized by greater values of b for their output queuing delay distribution. Thus, it appears that the output queuing delay distribution gets closer to an exponential distribution for higher output link utilizations. A similar finding was also reported in [16]. Nevertheless, further analysis is needed to confirm such a statement.

V. Conclusions

To the best of our knowledge, this work is the first to provide data about actual delays incurred through a single router in the backbone. We measure single-hop delay as experienced by packets in the Sprint IP backbone network. We develop a methodology to identify the contributing factors to the single-hop delay that is simple and applicable to *any* single-hop delay measurements. We demonstrate its applicability on OC-3 and OC-12 packet traces. In addition to packet processing, transmission, and queuing delays, we identify the presence of very large delays that cannot be explained within the context of a work-conserving FIFO output queue. We provide a simple technique to remove these outliers from our measurements, and offer possible explanations regarding the events that may have led to such extreme delays through a single node.

According to our results, 99% of the packets in the backbone experience less than 1 ms of delay going through a single router when transmitted at OC-3 speeds. At OC-12 line rates, the single-hop delay drops to less than 100 μs. After the extraction of the queuing delay component in our measurements, we show that the largest part of single-hop delay experienced by a packet is *not* due to queuing, but rather to the processing and transmission of the packet across the switch fabric. In addition, we observe a small number of packets (less than 1% in our measurements) that may experience significantly larger delays, either because they carry IP options or because they are affected by idiosyncratic router behavior.

The analysis of the queuing delay distribution reveals that it can be approximated by a Weibull distribution with a scale parameter $a = 0.5$, and a shape parameter $b = 0.6 \sim 0.7$ for transmission of packets across two different OC-3 linecards. When packets are forwarded within the same linecard, i.e. they do not transit the switch fabric, and at OC-12 link speeds, the queuing delay distribution can be approximated with a Weibull distribution with a higher shape parameter $b = 0.82$. Thus, the output queuing delay distribution is *long-tailed* confirming previous analytical findings by Norros [2]. We believe that identification and modeling of the several components comprising single-hop delay allow for more realistic backbone router models, that could easily be used in simulation environments.

In summary, packets in the Sprint IP backbone network experience edge-to-edge delays that are dominated by the propagation delay, and face minimal jitter. This result, though, should be evaluated within the context of Sprint's backbone design principles that dictate moderate link utilization across the network; in our measurements all links were utilized less than 70% even at a 10 ms time scale.

Acknowledgment

We would like to thank Mark Crovella for providing us with the `aest` tool, Jim Kurose, Rene Cruz, Jon Crowcroft and Saleem Bhatti for their comments on the paper.

References

[1] W. Leland, M. Taqqu, W. Willinger, and D. Wilson, "On the self-similar nature of Ethernet traffic," *IEEE/ACM Transactions on Networking*, vol. 2, no. 1, pp. 1–15, Feb. 1994.

[2] I. Norros, "On the use of fractional brownian motion in the theory of connectionless networks," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 6, pp. 953–962, Aug. 1995.

[3] A. Erramilli, O. Narayan, and W. Willinger, "Experimental queuing analysis with long-range dependent packet traffic," *IEEE/ACM Transactions on Networking*, vol. 4, no. 2, pp. 209–223, Apr. 1996.

[4] A. Feldmann, A. Gilbert, and W. Willinger, "Data networks as cascades: Investigating the multifractal nature of Internet WAN traffic," *ACM Computer Communication Review*, vol. 28, no. 4, pp. 42–55, Sept. 1998.

[5] A. Erramilli, O. Narayan, A. Neidhardt, and I. Saniee, "Performance impacts of multi-scaling in wide area TCP/IP traffic," in *IEEE Infocom*, Tel Aviv, Israel, Mar. 2000.

[6] C. Fraleigh, C. Diot, B. Lyles, S. Moon, P. Owezarski, K. Papagiannaki, and F. Tobagi, "Design and deployment of a passive monitoring infrastructure," in *Passive and Active Measurement Workshop*, Amsterdam, Apr. 2001.

[7] "Dag 3.2 SONET network interface," http://dag.cs.waikato.ac.nz/dag/dag4-arch.html.

[8] 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 *IEEE Infocom*, New York, June 2002.

[9] D. Knuth, *The Art of Computer Programming, Volume I: Fundamental Algorithms*. Second Edition, Addison-Wesley Publishing Company, Reading, 1973.

[10] V. Paxson, "Measurements and analysis of end-to-end Internet dynamics," Ph.D. dissertation, University of California Berkeley, April 1997.

[11] K. Thompson, G. Miller, and R. Wilder, "Wide-area Internet traffic patterns and characteristics," *IEEE Network*, vol. 11, no. 6, pp. 10–23, November/December 1997.

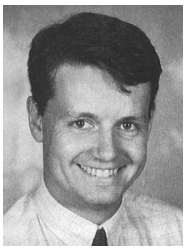
[12] S. Keshav and S. Rosen, "Issues and trends in router design," *IEEE Communications Magazine*, vol. 36, no. 5, pp. 144–151, May 1998.

[13] N. McKeown, "iSLIP: A Scheduling Algorithm for Input-Queued Switches," *IEEE Transactions on Networking*, vol. 7, no. 2, pp. 188–201, Apr. 1999.
[14] D. Cox, "Long-range dependence: a review," in *Statistics: An Appraisal*, H. A. David and H. T. David, Eds. Ames, IA: Iowa State University Press, 1984, pp. 55–74.
[15] M. Crovella and M. Taqqu, "Estimating the heavy tail index from scaling properties," *Methodology and Computing in Applied Probability*, vol. 1, no. 1, pp. 55–79, July 1999.
[16] J. Cao, W. Cleveland, D. Lin, and D. Sun, "Internet traffic tends *toward* Poisson and independent as the load increases," in *Nonlinear Estimation and Classification*, D. Denison, M. Hansen, C. Holmes, B. Mallick, and B. Yu, Eds. New York, NY: Springer Verlag, Dec. 2002, pp. 83–110.

Konstantina Papagiannaki received her first degree in electrical and computer engineering from the National Technical University of Athens, Greece, in 1998, and her PhD degree from the University College London, U.K., in 2003. She has been a member of the IP research group at the Sprint Advanced Technology Laboratories since April 2000. Her research interests are in Internet measurements, modeling of Internet traffic, and backbone network traffic engineering.

Sue Moon received her B.S. and M.S. from Seoul National University, Seoul, Korea, in 1988 and 1990, respectively, all in computer engineering. From 1990 to 1991, she worked for IMIGE Systems, Inc. in Seoul, Korea. She received a Ph.D. degree in computer science from the University of Massachusetts at Amherst. Since 1999, she has been with Sprint ATL in Burlingame, California. Her main research interests are network performance measurement and monitoring: network delay, traffic scaling behavior analysis, and network anomalies.

Chuck Fraleigh received a B.S. degree in computer and electrical engineering from Purdue University in 1996. He received an M.S. degree in electrical engineering from Stanford University in 1998, and a PhD degree in electrical engineering from Stanford in 2002. From 1998 to 2002, he was a student visitor at Sprint Advanced Technology Labs and is now with NetVmg. His research interests include Internet traffic measurement and modeling, network provisioning, and the design of network measurement systems.



Patrick Thiran received the electrical engineering degree from the Université Catholique de Louvain, Louvain-la-Neuve, Belgium, in 1989, the M.S. degree in electrical engineering from the University of California at Berkeley, USA, in 1990, and the PhD degree from the Swiss Federal Institute of Technology at Lausanne (EPFL), in 1996. He became a professor at EPFL in 1998, and was on leave with Sprint Advanced Technology Laboratories, Burlingame, CA, in 2000-01.

His research interests are in communication networks, performance analysis, dynamical systems and stochastic models. He served as an associate editor for the IEEE Transactions on Circuits and Systems in 1997-99. He is a Fellow of the Belgian American Educational Foundation, and he received the 1996 EPFL Doctoral Prize.

Christophe Diot 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. Diot moved to Sprint Advanced Technology Laboratories in October 1998 to take the lead of the IP research group. His current interest is in the passive monitoring of the Sprint IP backbone in order to study IP traffic characteristics and to design new analytical models and traffic engineering solutions for packet networks (ipmon.sprint.com).