Analysis of Measured Single-Hop Delay from an Operational Backbone Network

Konstantina Papagiannaki, Sue Moon, Chuck Fraleigh, Patrick Thiran, Fouad Tobagi, Christophe Diot

***Abstract*—**

We measure and analyze the single-hop packet delay through operational routers in a backbone IP network. First we present our delay measurements through a single router. Then we identify step-by-step the factors contributing to single-hop delay. In addition to packet processing, transmission, and queueing delays, we identify the presence of very large delays due to non-work-conserving router behavior. We use a simple output queue model to separate those delay components. Our step-by-step methodology used to obtain the pure queueing delay is easily applicable to any single-hop delay measurements.

After obtaining the queueing delay, we analyze the tail of its distribution, and find that it is long tailed and fits a Weibull distribution with the scale parameter, $a = 0.5$, and the shape parameter, $b = 0.58$ to 0.6. The measured average queueing delay is larger than predicted by M/M/1, M/G/1, and FBM models when the link utilization is below 70%, but its absolute value is quite small.

I. Introduction

DELAY is a key metric in network performance and quality-of-service perceived by end users. In today's best-effort Internet, packets experience delay due to transmission and propagation through the medium, as well as queueing due to cross traffic at routers. The characteristics of the traffic have a significant impact on the queueing delay. In a ground-breaking work, Willinger et al. 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 queueing delay. The Fractional Brownian Motion (FBM) model has been proposed to capture the coarse time scale behavior of network traffic. It shows that the queueing behavior diverges from that of the Poisson traffic model significantly [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 queueing behavior can be approximated differently depending on the link utilization [5].

All the analyses of queueing behavior, however, have been based on packet traces collected from a single link and fed into an output buffer, whose size and service rate are varied. We are not aware of any measurement of the queuing delay on operational routers. No measurement of the actual delay has been taken and compared with analytical models, mainly because no such data has been available before.

K. Papagiannaki, S. Moon, and C. Diot are with Sprint ATL, 1 Adrian Ct., Burlingame, CA 94010. E-mail: {dina,sbmoon,cdiot}@sprintlabs.com. K. Papagiannaki is also with the University College London. C. Fraleigh and F. Tobagi are with the Department of Electrical Engineering, Stanford University, Stanford, CA 94305. E-mail: {cjf,tobagi}@stanford.edu. P. Thiran was with Sprint ATL on sabbatical leave from École Polytechnique Fédérale de Lausanne. E-mail: Patrick.Thiran@epfl.ch.

The difficulty in measuring single-hop delay in a real network is manyfold:

- Timestamps should be accurate enough to allow the calculation of the transit time through a router. This requires in particular that the measurement systems (1) have sufficient resolution so as to differentiate between the arrival times of two consecutive packets, and (2) are synchronized to each other in a way 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 for in-depth analysis 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 queueing behavior of any single output link is unrealistic in an operational network.

We have designed a measurement system to address the first two of the above difficulties, and deployed it in a commercial tier-1 IP backbone network to collect packet traces with accurate timestamps [6]. By splitting the optical signals, and tapping a part of them, the measurement systems are capable of capturing and timestamping every packet traversing the link (see details in Section II). Then, by comparing the differences in timestamps at the input and output links, we obtain the single-hop delay of packets. The third difficulty is not easy to overcome due to cost and space issues in deployment. Although this prevents us from characterizing the queueing experienced by *all* packets, we explain in Section II why our sample of packets is valid.

In this work we study the queueing delay through a single router in a backbone network, and compare it with known analytical models. In Section II we present delay measurements of more than three million packets winnowed down from more than one billion packets and 90 gigabytes of data collected from the Sprint IP backbone network[1]. In Section III we provide a methodology for quantifying the various elements in single-hop delay. On top of the expected factors, such as transmission, queueing, and processing delays, we observe very long delays not due to queuing. We use a single output queue model to isolate them. This step is necessary to separate delays not due to congestion in the study of the queueing behavior. In Section IV we analyze the tail behavior of the queueing delay, and compare it with estimates from various models. In Section V we summa-

[1] The focus of this paper is to study the queueing delay, not the vendor-specific router-internal operations. Thus we do not publish the information on the vendors and types of the routers.

rize our findings.

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 (2.5Gbps), using the DAG monitoring card [7]. However, only OC-3 monitoring systems were installed at the time when we collected the data presented in this paper. These monitoring systems have been deployed on various links in a Point of Presence (POP) of the Sprint E|Solutions IP backbone. We have collected hour- and day-long packet traces, and analyzed them off-line. Details of the measurement infrastructure can be found in [6].

A. Measurement Environment & Clock Synchronization

Each DAG card features a dedicated clock on board. This clock runs at a rate of 16MHz which provides a granularity of 59.6 ns between clock ticks. Packets are not timestamped immediately when they arrive at the DAG card. They first pass through a chip which implements the SONET framing, and which operates on 53 bytes ATM cells. Once this buffer is full, an interrupt is generated, and the packet is timestamped. In other words, timestamping happens on the unit of 53 bytes, thus introducing a timestamp error of 2 μs (the time needed for the transmission of 53 bytes on an OC-3 link).

Due to room temperature and the quality of the oscillator on board the DAG card, the oscillator may run faster or slower than 16 MHz. For that reason, it is necessary to discipline the clocks using an external stratum 1 GPS receiver located at the POP. The GPS receiver outputs a 1 pulse-per-second (PPS) signal which is distributed to all of the DAG cards located at the POP.

Clock synchronization on board the DAG card is achieved in the following fashion [8]. At the beginning of the trace collection the clock is loaded with the absolute time from the PC's system clock (e.g. 7:00 am Aug 9, 2000 PST). The clock then begins to increment at a rate of 16 MHz. When the DAG card receives the first 1 PPS signal after initialization, it resets the lower 24 bits of the clock counter. Thereafter, each time the DAG card receives the 1 PPS signal, it compares the lower 24 bits of the clock to 0. If the value is greater than 0, the oscillator is running fast and the DAG card decreases the frequency. If the value is less than 0, the oscillator is running slow and the DAG card increases the frequency.

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

There are several sources of error that may occur in the synchronization of the systems. We'll distinguish between two types of errors: (a) timestamping errors specific to a single DAG card, and (b) synchronization errors between multiple DAG cards.

As already mentioned, the DAG card uses an ATM cell buffer to store the captured packet until it is timestamped. That may introduce a maximum error of 2 μs. Given that the resolution of the time tick on board the DAG card is 59.6 ns, the use of this ATM cell buffer introduces a maximum timestamping error of 2 μs.

All DAG cards in a specific POP use the same GPS receiver for their clock synchronization. Therefore, synchronization errors between DAG cards in the same POP could be due to two possible reasons. The first one is the difference in propagation time for the 1 PPS signal. The 1 PPS signal is distributed to the DAG cards using a daisy chain topology. The difference in cable length between the first and the last system is 8 meters, which corresponds to a propagation delay of 28 ns. The second source of error is due to the fact that the clock synchronization mechanism cannot immediately adjust to changes in the oscillator frequency. Once the DAG card receives the 1 PPS interrupt, it has to increase or decrease the oscillator frequency depending on the clock offset. We measured in the lab the maximum clock offset observed when the card receives the 1 PPS interrupt. Its maximum value was 30 clock ticks, representing an error of 1.79 μs, while the median error was a single clock tick (59.6 ns). Accounting for those errors, the worst case skew between any two DAG clocks, participating in single-hop measurements, is less than 2 μs.

The total effect of both types of errors is a maximum clock skew of 6 μs. The lowest delay values we have measured in all our traces never go below 28μs. Therefore, a 6 μs skew represents a 20% error in the measurements.

B. Collected data

We capture packets on input links just before they enter a router, and on output links right after they leave a router. Let us 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: $d(n) = T_{out}(n) - T_{in}(n)$. It corresponds to the total time a packet spends in a router, including IP address lookup time at the input port, transmission time over the backplane switch fabric, waiting time at the output port, and transmission delay.

Packet traces from nine links have been analyzed. Due to space limitations, this paper shows measurements from four representative links only, collected on August 9th, 2000. Those four links include the pair of links that exhibits the highest delays observed among all of the monitored links. The first data set is 14 hours long, and the second data set 45 minutes long. Table I provides details about the four traces. We label a router-inbound link as `in`, and a router-outbound link as `out`, and refer to them as a data set in the rest of the paper. The first data set has been collected on the `in1-out1` pair of links: packets arrive at a core router from a public peering point, and leave for an access router. For the second data set on `in2-out2`, packets arrive at the same core router from the same access router and leave for the same public peering point. In other words, we measure both directions of the same router path. All links are POS (Packet-over-Sonet) OC-3 (155 Mb/sec). Figure 1 depicts the configuration of the monitoring systems. Dotted lines represent the traffic from the router's interfaces we do not monitor.

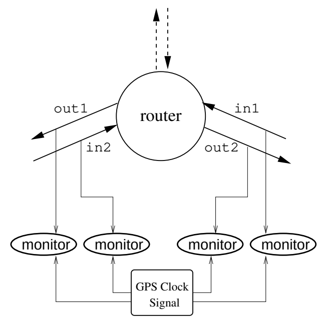


Fig. 1. Typical configuration of monitoring systems in a POP.

Figure 2 presents the link utilization averaged over a one minute interval for both data sets. It illustrates the daily fluctuations in the traffic as well as the wide variation in total volume. The link utilization ranges from 20% to nearly 70% in the first data set, and between 20 and 35% in the second.

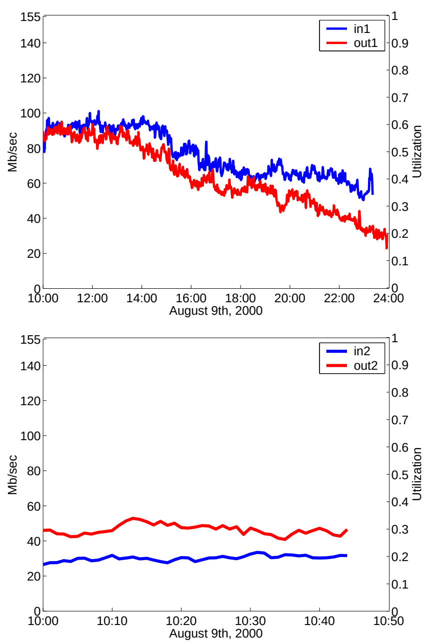


Fig. 2. Link utilization on `in1`, `out1`, `in2`, and `out2`

C. Matching Packets

From the measurements on input and output links, we need 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 carry almost identical information in all IP packets. 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, which is assumed to be the maximum delay value a packet can experience through a single node.

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, one by one, 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 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 the traces, and have paid special attention to matching them. Duplicate packets have all 44 bytes collected 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. By this method, 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, two traces of 2,781,201, and 1,175,674 *matched* packets are produced for the first and second data sets, respectively. We use these traces in the next section to analyze the elements that comprise the single-hop delay.

D. Representativeness of the Data

Our correlation traces provide us with complete information about the path between a specific incoming and a specific outgoing link. We have records of the arrival and departure time of each matched packet, as well as timestamps for all the other packets sharing the same incoming and outgoing link. We calculate the delays experienced by the matched packets. In this section, we would like to investigate how representative those delay results are for the rest of the traffic flowing on the same monitored links.

The matched packets form a subset of the traffic on the output link; matched packets in `set1`, and `set2` constitute 0.5%, 2.4% of the total packets on links `out1`, and `out2` respectively. Although this subset results from a single input port, it will be equivalent to a pure random sampling if the matched packets on the output link are geometrically distributed and independent [11].

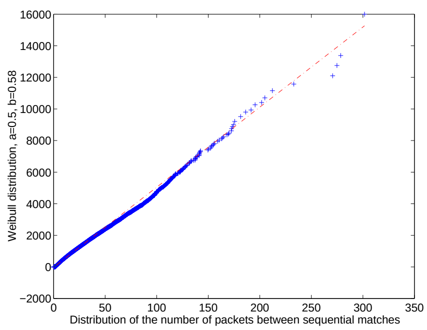
We first analyze the distribution of the distance between matched packets in terms of packet counts. We find that it fits a Weibull distribution[2]. Figure 3 shows the Quantile-Quantile plot

[2]The probability density function of a Weibull distribution is given by $f(x) =$

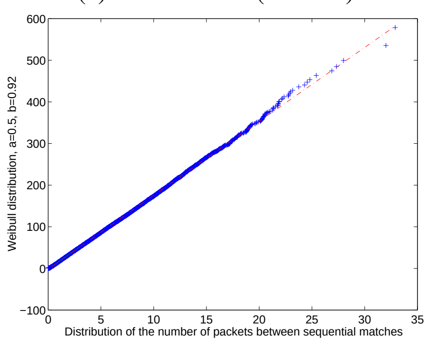
Set	Link	Date	Start Time	End Time	No. of packets
1	`in1`	Aug. 9, 2000	09:56:33 PDT	19:56:07 PDT	793,528,684
	`out1`	Aug. 9, 2000	09:56:00 PDT	19:56:07 PDT	567,680,718
2	`in2`	Aug. 9, 2000	09:56:03 PDT	10:41:04 PDT	28,213,976
	`out2`	Aug. 9, 2000	09:56:04 PDT	10:41:04 PDT	48,886,948

TABLE I

DETAILS OF TRACES

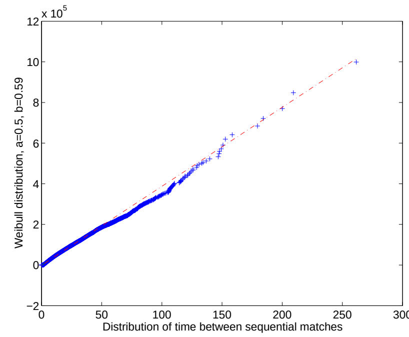


(a) First data set ($b = 0.6$)

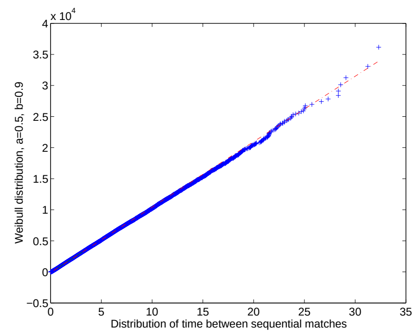


(b) Second data set ($b = 0.92$)

Fig. 3. QQ-plot of the distribution of the number of packets in the output trace `out` between two sequential matched packets (x-axis) versus a Weibull distribution.



(a) First data set ($b = 0.59$)



(b) Second data set ($b = 0.9$)

Fig. 4. QQ-plot of the inter-packet time distribution of the matched packets in the output trace `out` (x-axis) versus a Weibull distribution.

of the distribution of the number of packets between sequential matches and a Weibull distribution with a given b parameter. A shape parameter of $b = 1$ makes the Weibull distribution coincide with the exponential distribution, and indicates a pure random sampling (as the discrete equivalent of the continuous exponential distribution is a geometric distribution). If b is close to 1 though, a sample set is not purely random, but is close to random with occasional large gaps between matched packets.

Our two data sets exhibit $b = 0.6$ and $b = 0.92$, respectively. The inter-packet distribution of the second data set is therefore very close to an exponential distribution, while for the first data set is not.

We also analyze the distribution of the distance between two matched packets in terms of time. Figure 4 shows the Quantile-Quantile plots for the distribution of time between the packets matched on the output. We observe similar Weibull shape parameters (0.59 and 0.9 respectively).

$\frac{bx^{b-1}}{a^b}e^{-(\frac{x}{a})^b}$, with $a > 0,\ b > 0$; α is called the scale parameter, while b is called the shape parameter.

We further look into the sample autocorrelation function (ACF) of the inter-packet distance to investigate any correlation. In both time and number of packets, the ACF of the first data set exhibits significant correlation. The second data set exhibits much less correlation, but some correlation still exists at the lag of 50. Thus, though the distribution of the inter-packet distance in the second data set is close to an exponential distribution, it is difficult to assess how close our sample is to a pure random sample, due to some correlation structure in the data set.

In summary the first data set cannot be considered as a pure random sample of the queueing behavior at the output link, while the second data set is very close. Thus we consider our data sufficient for studying the queueing behavior. Moreover, in the case of the second data set, our conclusions will be very close to the complete queueing behavior at the output link.

III. DELAY ANALYSIS

We start this section with general observations on the delay measurements. Then we plot the empirical probability density function of the measured single-hop delay, and quantify step-by-step the factors that contribute to the single-hop delay. The goal of this step-by-step analysis is to isolate the queueing delay, which is analyzed in Section IV.

A. Observations

Let us denote the m-th matched packet as m, and the total number of matched packets by M. Fig. 5 plots the minimum, average, and maximum values of $\{d(m)\}$ per minute interval for the first data set. We observe first that the minimum delay is almost constant throughout the trace, while the average delay exhibits more oscillations and decreases by a few tens of microseconds as the link utilization decreases toward midnight (Figure 2). 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 queueing in each one minute interval.

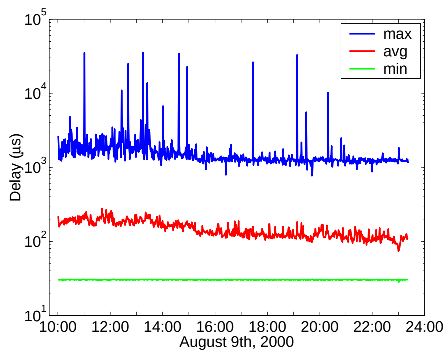


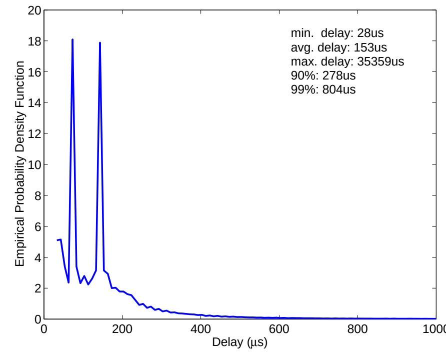
Fig. 5. Minimum, average, and maximum delay per minute for the matched packets of the first data set.

The maximum delay is more variable than the average delay. It shows occasional spikes of a few milliseconds reaching up to 35 ms. Spikes of more than 10 ms are more frequent in the first half of the trace, when the links are more utilized. We also note that the maximum delay remains consistently above 1 ms, even as the average delay decreases. We return to this phenomenon with an explanation in Section III-B.4.

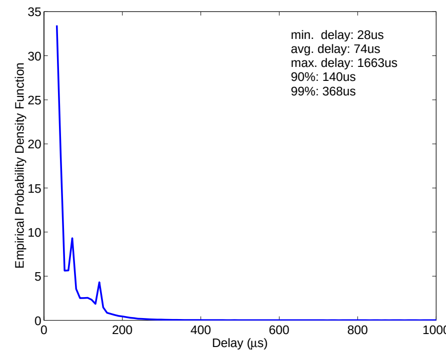
B. Step-by-Step Analysis of the Single-Hop Delay

1) Empirical Probability Density Function of Single-Hop Delay: We plot the empirical probability density function of $\{d(m)\}, 1 \leq m \leq M$, in Fig. 6. It shows that 99% of packets experience less than 804 μs of delay in the first data set and less than 368 μs in the second data set. Only 0.001% of matched packets experience a delay larger than 5 ms in both data sets, while the maximum delay observed is 35 ms in the first data set and 1.6 ms in the second data set.

There are three distinct peaks at the beginning of each curve. The peaks are located between 0 and 200 μs. Previous work by Thompson et al. reports that packets in the backbone do not have a uniform size distribution, but three unique peaks at 40 to 44, at 552 to 576, and at 1500 bytes [12]. The sizes of 40 to 44 bytes correspond to the minimum TCP acknowledgement packets and *telnet* packets of a single key stroke; 552 and 576 to default MTU sizes when path MTU discovery is not used by a sending host;



(a) First data set



(b) Second data set

Fig. 6. Empirical probability density function of delay of matched packets $\{d(m)\}$.

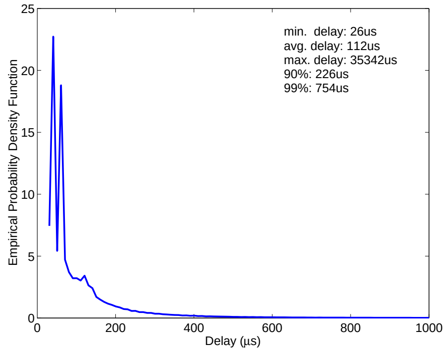
and 1500 to the Ethernet MTU size. In our traces more than 70% of packets have three sizes: 40, 576, and 1500 bytes. We conjecture the three peaks at the beginning of the delay distribution to be related to the packet size. To verify the above conjecture, we group the packets of those three sizes, and plot three separate empirical probability density functions. Each distribution has a unique peak that matches one of the three peaks in Figure 6. This size dependence will be used in the next section to identify factors contributing to single-hop delay.

2) Transmission Delay on the Output Link: We now turn our attention to what might contribute to the same amount of delay for packets of the same size. A first cause is the *transmission delay* on the output link. It 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[3]. 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*, and denote it 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 7.

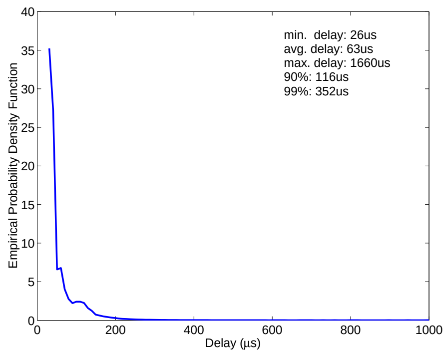
There still are three distinct peaks in the distribution, though they are less pronounced than in Figure 6. This indicates that there is still a part of the router transit time that depends on the packet size.

3) Minimum Router Transit Time: When a packet arrives at a router, its destination address is looked up in the forwarding table and the appropriate output port is determined. Then, the

[3]Throughout this paper, we set $C_{out} = 150.336 Mbps$, which is the effective payload of POS OC-3.

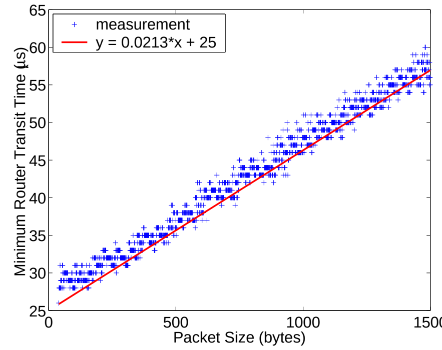


(a) First data set

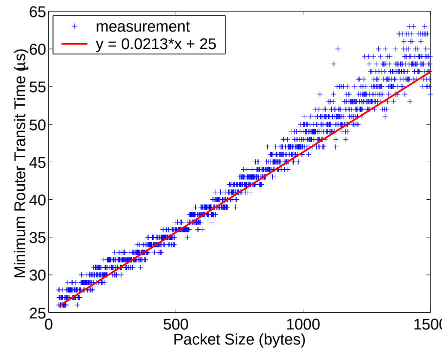


(b) Second data set

Fig. 7. Empirical probability density function of router transit time, $d_{tx}^{-}(m) = d(m) - l_m/C_{out}$



(a) First data set



(b) Second data set

Fig. 8. Minimum router transit time versus packet size: $\min_m d_{tx}^{-}(m)$ for $m \in \{i | l_i = L\}$, versus the packet size L

packet is transferred to the output port through the backplane of the router. All the core routers in today's market do store-and-forward, as opposed to cut through. This operation along with address lookup imposes on *every* packet a minimum amount of delay, proportional to its size, which is likely to explain those remaining peaks in Figure 7.

The architectural details of the router determine exactly where packets are delayed inside a router, and they vary from one router to another. The goal of our work is not to discover router-dependent delay behavior, but rather the queueing delay due to interfering cross traffic. Below we quantify the minimum router transit time experienced by packets in our data sets.

To study if this minimum time is dependent on the packet size, we plot the minimum router transit time per packet size, $d_P(L)$, versus the packet size L in Figure 8:

$$d_P(L) = \min_{1 \leq m \leq M} \{d_{tx}^{-}(m) | l_m = L\}.$$

This figure shows that there exists a linear relationship between the two metrics. This relationship is made explicit through a linear regression. Given that both data sets feature an order of magnitude more packets of 40, 576, and 1500 bytes, those three packet sizes are more likely to provide us with accurate minimum router transit times. For that reason, we use only the measurements for those three packet sizes in linear regression, and obtain the following equation for the minimum router transit time per packet size:

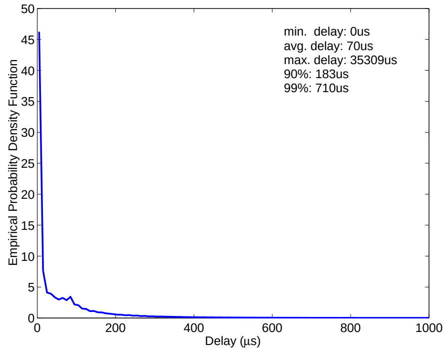
$$\hat{d}_P(L) = 0.0213 \cdot L + 25 \ \text{(in } \mu\text{s)} \tag{1}$$

Subtracting $\hat{d}_p(l_m)$ from the router transit time, $d_{tx}^{-}(m)$, we obtain the delay distribution presented in Figure 9, which represents the actual amount of time packets have to wait in the output queue.

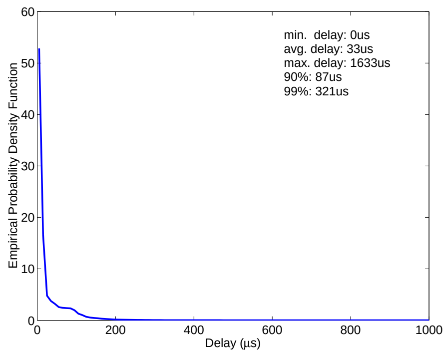
Peaks have now disappeared and the delay distributions look very similar for both data sets[4]. The distribution is characterized by very low delays: 45% of the packets in the first data set, and almost 50% of the packets in the second data set experience zero queueing delay. Some slight differences in the average delay could possibly be explained by the packet size distribution of the two sets: the first data set is dominated by packets larger than 500 bytes, while the second data set contains mostly 40 bytes packets. Also the link utilizations on the output links are different. However, the maximum delays for both data sets are much higher than the 99% percentiles, and in the case of the first data set, the maximum delay still reaches 35 ms.

A key observation is that the tail of the delay distribution is very long, accounting for the presence of very large delays in the output queue. 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 some long delays are not caused by congestion on the output link. We conjecture that there are components of the delay that do not stem from queueing due to cross traffic, but rather come from idiosyncratic router behavior. In the next section, we identify how those idiosyncrasies contribute to delay.

[4]Considering that the transmission delay of a 1500 byte packet is 80 μs, we conjecture that the flat region around 100 μs is due to packets queueing behind a maximum-sized packet. However, since we do not have input timestamps of all the packets, we cannot verify the conjecture.



(a) First data set



(b) Second data set

Fig. 9. Empirical probability density function of $(d^{-}_{tx}(m) - \hat{d}_P(l_m))$

4) Filtering Based on a Single Output Queue Model: When packets arrive at a router, they contend for resources to be transferred to the destination output port. 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. 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 enable us to determine if the delay of a packet is due to queueing or not, using *only* the measurements we have at our disposal.

In previous sections we have identified three contributing factors to single-hop delay: transmission delay, minimum router transit time, and queueing delay. In modern router architectures, the packet processing is heavily pipelined so that the minimum router transit time of a packet should not introduce extra queueing for the next packet arriving at the input port. It can be considered as having the packet arrival delayed at the output queue. We thus modify the packet arrival time as $T^{'}_{in}(m) = T_{in}(m) + \hat{d}_p(l_m)$, and set the service rate of the single output queue to the transmission rate of the output link, as illustrated in Figure 10.

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 11 we plot the number of bytes transmitted during the time interval of $[T^{'}_{in}(m), T_{out}(m) - l_m/C_{out}]$ versus the size of the interval. All data points lie below a line that corresponds to the link speed. Most of those points that fall off the line are bounded by another line below, of the same slope, which allows for the transmission of an extra maximum-sized packet. The line is: $y = C_{out} \cdot x - 1500$, where x is the size of the time interval, and y is the number of bytes. 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 increase the margin of error in our analysis. We thus allow one maximum-sized packet as the error margin in our waiting time calculation. 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 most of the time. We consider as the *filtered* data set those packets that lie between the two bounding lines in the figure. Other packets are considered to have experienced delay not due to queueing beyond the error margin, and are filtered out. From the first data set, 9.2% of the matched packets are filtered out, and from the second data set, 3.1%.

We summarize the statistics of the router transit time and queueing delay of filtered and non filtered packets in Tables II and III. As we can see, the average delay, the 90th, and the 99th percentiles of the filtered data set are now lower in both data sets. Moreover, all of the delays larger than 5 ms in the first data set have disappeared, and the maximum delay drops from 35 ms to 3.9 ms. On the other hand, the maximum delay for the second data set remains the same. In other words, measured delays over 5 ms are not due to queueing, and our single queue model is effective in filtering them. The average, 90th, and 99th percentile delays of the first data set are larger than those of the second data set. Packets of 40 bytes take up about half of the packets in the second data set, and explain the relatively small values in delay. We plot the minimum, average, and maximum values of the filtered delays for the first data set in Figure 12. Compared to Figure 5, we notice that the maximum delay does not stay over

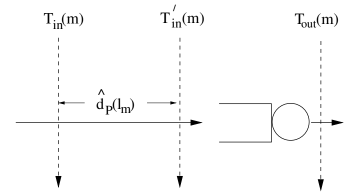


Fig. 10. Single output queue model of a router

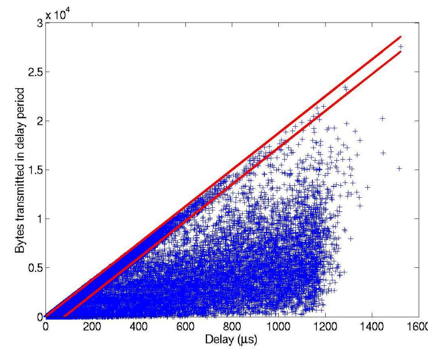


Fig. 11. Number of bytes transmitted between $T^{'}_{in}(m)$ and $T_{out}(m) - l_m/C_{out}$ on `out1`.

1 ms, but decreases as the link utilization of the output link decreases. We believe that the 1 ms maximum delays in Figure 5 are due to a known periodic task that introduces a 1 ms delay at least once per minute. Indeed, the autocorrelation function for the filtered data set confirms such a periodicity in the 1 ms delay values. Packets delayed for larger periods of time had no further distinguishing characteristics.

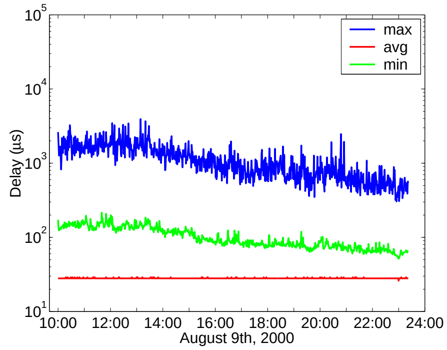


Fig. 12. Minimum, average, and maximum delays per minute of the matched and filtered packets for the first data set.

With the single output queue model, we have been able to remove non-queueing delays[5]. We speculate on the origin of the non-queueing delays in the next section.

C. Possible Causes for Very Large Delay

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 5 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 time constant, both of which we checked in our traces. We also observe the same phenomenon on traces collected on other links.

Therefore, we conjecture that very large delays are caused by implementation biases in the routers. We have identified the following potential sources of non-queueing delays:

- Routers can stop forwarding packets for a short period of time if they are busy with some other resource-intensive task, *i.e.* routing table updates[6].
- It is also known that not all packets experience the same processing overhead at a router. Most routers are designed to optimize the performance for the majority of packets. IP packets with options require extra processing in the IP protocol stack to look into the option field, and might therefore travel through a slower path in software than other packets without options (analysis of the collected data showed that our correlation traces did not include any option packets).
- SNMP requests and garbage collection in memory management in the routers.
- Finally, router interface cards with multiple ports or backplane switch fabrics may allow head-of-line blocking. The traces presented in this paper were collected for interfaces belonging to quad-OC3 linecards. Therefore, it is highly likely that the linecard was busy serving one of the other three interfaces while a largely delayed packet was waiting for transmission at the monitored link. Unfortunately, given that the other three interfaces of the same linecard were not monitored, we cannot prove such a statement.

We should note at this point, that our measurements are in line with testing results published in [13]. In this report routers from several vendors for OC-48 and OC-192 link speed were tested, and were reported to have significant variations in the delay of fixed-sized packets. We believe some of the large delays we see in our traces are from similar router architecture design constraints.

Building a router to function as a perfect output queue is a challenge, and we expect most routers to have idiosyncrasies that deviate from an ideal output queue. For the study of queueing behavior generic to traffic characteristics, it is important to isolate delay due to router idiosyncrasies. Our step-by-step methodology provides us with a systematic approach to extract queueing delay from single-hop delay, and is applicable to any single-hop delay measurements.

IV. Analysis of Queueing Delay

A. Tail Behavior

In this section, we use the filtered data sets (i.e. without the packets that experience delay not due to queueing) to analyze the tail of the queueing delay distributions of $(d^-_{tx} - \hat{d}_P(l_m))$. This analysis will help us identify possible models for the queueing delay in the backbone, that could be exploited in simulation environments. We show that our results agree with previous studies.

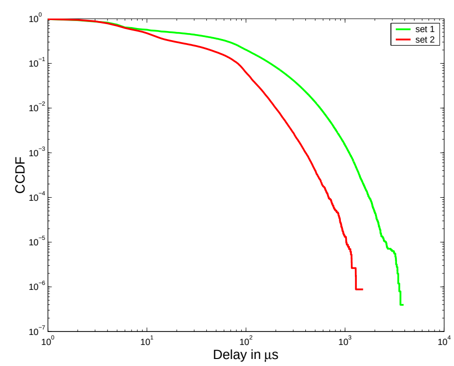


Fig. 13. Log-log plot of CCDF for data sets 1 and 2

The 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 more rapidly than the 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

[5] Strictly speaking, transmission and propagation delays are not due to queueing as well. However, we limit the use of non-queueing delay only to the delay that is not due to congestion, but to reasons we explore in the next section.

[6] Usually referred to as the *coffee break* effect.

time in μs	original set 2,781,201 matches						filtered set 2,525,643 matches (9% filtered)					
	min.	avg.	90%	99%	max.	var.	min.	avg.	90%	99%	max.	var.
$d_{tx}^{-}(m)$	26	112	226	754	35,342	24,011	26	107	219	606	3,937	13,980
$d_{tx}^{-}(m) - \hat{d}_p(l_m)$	0	70	183	710	35,309	23,760	0	66	176	561	3,903	13,607

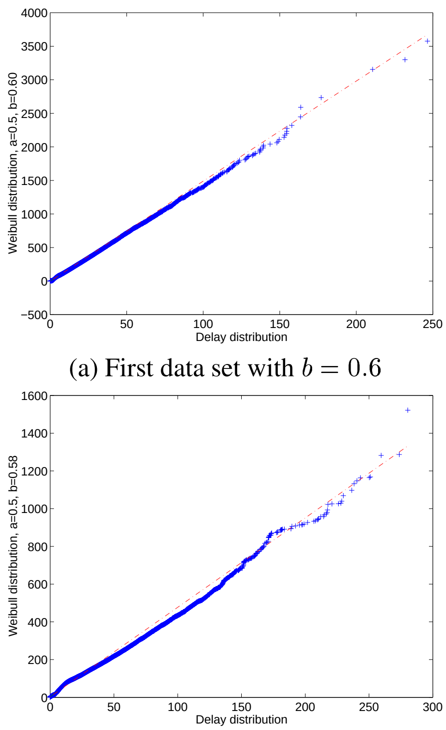
TABLE II

STATISTICS FOR THE FIRST DATA SET

time in μs	original set 1,175,674 matches						filtered set 1,139,608 matches (3% filtered)					
	min.	avg.	90%	99%	max.	var.	min.	avg.	90%	99%	max.	var.
$d_{tx}^{-}(m)$	26	63	116	352	1,660	6,651	26	56	113	230	1,548	2,004
$d_{tx}^{-}(m) - \hat{d}_p(l_m)$	0	33	87	321	1,633	6,616	0	27	83	200	1,521	1,965

TABLE III

STATISTICS FOR THE SECOND DATA SET



(a) First data set with $b = 0.6$

(b) Second data set with $b = 0.58$

Fig. 14. QQplot of the queueing delay distribution against a Weibull distribution.

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. Lastly, we call *long tailed* those distributions that are not strictly heavy tailed, but decay slower than exponential. Lognormal and Weibull distribution 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). Norros shows that the queueing delay distribution of the FBM traffic is approximated by a Weibull distribution [2]. We test the obtained queueing delay distributions against all three types identified above.

To examine what tail category our delay distributions fall into, we first plot the complementary cumulative distribution function (CCDF) of $(d_{tx}^{-}(m) - \hat{d}_P(l_m))$ of the filtered sets in log-log scale in Figure 13. If the distribution is exponential, the tail forms a straight line. Ours are clearly not exponential.

Next we use the `aest` tool to check if it is heavy tailed [15]. The results show our delay distributions do not have the power-law tail like the Pareto distribution, and are not heavy tailed. Lastly, we fit a Weibull distribution to the distributions, and present our results in Figure 14 for both data sets. Both distributions fit to a Weibull distribution with a shape parameter b close to 0.6. Thus the distributions of measured queueing delay are long tailed, confirming the finding in [2].

B. *Impact of Link Utilization on Queueing Delay*

In this section, we investigate the evolution of queueing delay with respect to link utilization in our backbone network, where link utilization ranges from 0 to 70%. Simple models, such as M/M/1, and M/G/1 fail to account for long-range dependence in the traffic, and therefore predict far smaller delays than the ones actually experienced. On the other hand, the Fractional Brownian Motion (FBM) model captures the characteristics of the observed traffic, but is mostly used in estimating the queueing delay for links which are utilized above 80%; a highly untypical operating region for backbone links[7]. We use our measurements to study the effect of link utilization on queueing delay, and to understand the delay guarantees that can hold inside a backbone network.

[7]Provisioning rules have evolved such that well engineered IP backbones very rarely operate links at utilization regions that would be considered congested [16].

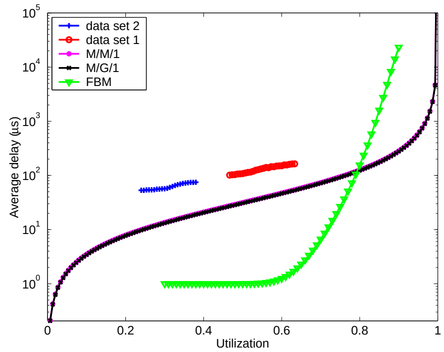


Fig. 15. Comparison of the average queueing delay vs. link utilization, derived from the data sets, the M/M/1 model, the M/G/1, and the FBM model.

We segment each trace into 5 minute intervals. We calculate average link utilization and average delay measured for each 5 minute interval. We proceed to correlate link utilization with delay and we calculate the average delay seen for each level of link utilization. In Figure 15, we present the average queueing delay versus link utilization for both data sets, and compare them with the M/M/1, M/G/1 and FBM models.

The parameters of the models are estimated from the first data set. The M/M/1 and M/G/1 take as input the average link utilization, the average packet size, and the capacity of the output link. For the M/G/1, specifically, the mean and covariance of the service time were estimated based on the packet size distribution and the link capacity, and the service times were found to be non-correlated. The variance in the service time for both data sets is very close to 1, and thus the graph of M/G/1 almost falls on M/M/1. The parameters for the FBM are estimated from the trace, and are equal to $m = 46.745\ Mb/sec, a = 350\ Kbit \times sec, H = 0.885$.

As can be seen from Figure 15, the M/M/1, and M/G/1 models capture the trend of the relationship between queueing delay and link utilization, but underestimate it by almost half an order of magnitude. It is well known, that the FBM model is designed to capture the queueing behavior at high link utilizations, while underestimating it at low and intermediate link utilizations. Indeed, FBM is performing poorly for link utilizations below 70%. Moreover, given that our links are never more than 70% loaded, we cannot compare the queueing delay predicted by the FBM in cases of high load, with actual measurements. In other words, for the operating regions that a large network would choose to utilize its links at, all M/M/1, M/G/1, and FBM underestimate the average queueing delay.

V. Summary

In this paper, we present single-hop delay measurements collected in an operational backbone network. We summarize our contributions.

- We offer a methodology to identify the contributing factors in single-hop delay. The methodology is simple and applicable to *any* single-hop delay measurements.
- We identify "non-queueing delay" as a factor in the single-hop delay. We provide a simple technique to remove these from our measurements.
- The delay measurements show that 99% of the packets in the backbone experience less than 1 ms of delay going through a single router.
- The queueing delay distribution is long tailed, and can be approximated by a Weibull distribution with a scale parameter $a = 0.5$, and a shape parameter $b = 0.58 \sim 0.6$.
- The average value of the measured queueing delay is larger than what proposed models would predict.

Lastly, we would like to point out that this work is the first to provide data about actual delays incurred through a single router in the backbone. We will extend the current work to study how different applications and protocols impact traffic statistics and queueing delay. Future work will also include the measurement and analysis of multi-hop delays from links of higher than OC-3/OC-12 speeds.

Acknowledgement

We would like to thank Mark Crovella for providing us with the `aest` tool, Allen Downey, Kavé Salamatian, Nick McKeown, Zhi-Li Zhang, Sang Lyul Min, and the Sprint E|Solutions network operations for their invaluable comments.

References

[1] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of Ethernet traffic (extended version)," *IEEE/ACM Transactions on Networking*, 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, 1995.
[3] A. Erramilli, O. Narayan, and W. Willinger, "Experimental queuing analysis with long-range dependent packet traffic," *IEEE/ACM Transactions on Networking*, 1996.
[4] A. Feldmann, A.C. Gilbert, and W. Willinger, "Data networks as cascades: Investigating the multifractal nature of Internet WAN traffic," in *ACM SIGCOMM '98*, 1998.
[5] A. Erramilli, O. Narayan, A. Neidhardt, and I. Saniee, "Performance impacts of multi-scaling in wide area TCP/IP traffic," in *Proceedings of INFOCOM*, 2000.
[6] C. Fraleigh, C. Diot, B. Lyles, S. Moon, P. Owezarski, D. Papagiannaki, and F. Tobagi, "Design and deployment of a passive monitoring infrastructure," in *Proceedings of Passive and Active Measurement Workshop*, Amsterdam, April 2001.
[7] "Dag 3.2 SONET network interface," http://dag.cs.waikato.ac.nz/dag/dag4-arch.html.
[8] "Dag synchronization and timestamping," http://dag.cs.waikato.ac.nz/dag-/docs/dagduck_v2.1.pdf.
[9] D. E. 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. thesis, University of California Berkeley, April 1997.
[11] R. W. Wolff, "Poisson arrivals see time average," *Operations Research*, vol. 30, pp. 223–231, 1982.
[12] K. Thompson, G. J. Miller, and R. Wilder, "Wide-area Internet traffic patterns and characteristics," *IEEE Network*, pp. 10–23, November 1997.
[13] "Internet core router test," http://www.lightreading.com/testing, March 2001.
[14] D. R. Cox, "Long-range dependence: a review," in *Statistics: An Appraisal*, H. A. David and H. T. David, Eds., pp. 55–74. Iowa State University Press, Ames, IA, 1984.
[15] M. E. Crovella and M. S. Taqqu, "Estimating the heavy tail index from scaling properties," *Methodology and Computing in Applied Probability*, vol. 1, no. 1, 1999.
[16] "Private communication with engineers at Sprint E|Solutions," .