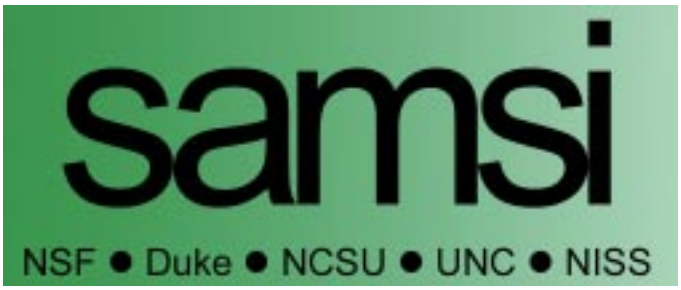


Long-Range-Dependence in a Changing Internet Traffic Mix

Cheolwoo Park, Felix Hernandez-Campos,
J.S. Marron, David Rolls and F.D. Smith

Technical Report #2004-9
March 26, 2004

This material was based upon work supported by the National Science Foundation under Agreement No. DMS-0112069. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Statistical and Applied Mathematical Sciences Institute
PO Box 14006
Research Triangle Park, NC 27709-4006
www.samsi.info

Long-Range-Dependence in Changing Internet Traffic Mixes

Cheolwoo Park

Felix Hernandez-Campos

J. S. Marron

David Rolls

F. Donelson Smith

1. Introduction and Motivation

This paper provides a deep analysis of long-range dependence in continually evolving Internet traffic mixes by employing a number of recently developed statistical methods. Surprisingly large and consistent differences in the behavior of packet-count time series are observed between data from 2002 and 2003. A careful examination based on stratifying the data according to protocol, reveals that the large difference is driven by a single UDP application that was not present in 2002. Another result is that the observed large differences between the two years shows up only in packet-count time series, and not in byte counts (while conventional wisdom suggests that these should be similar). We also explore some of the implications for queue management in routers resulting from these findings.

The seminal papers reporting empirical evidence for long-range dependent and self-similar properties in network traffic first appeared around ten years ago [refs]. Their findings were a "disruptive" event in networking research. They discovered a remarkable characteristic of Internet traffic – its *high variability across a wide range of scales, and how that variability changes as scale increases*. If we plot the number of packets or bytes that arrive at a network link, say every 10 milliseconds, we observe a highly variable process where the number of arrivals is constantly changing. Interestingly, if we plot these arrivals at coarser scales, say every 0.1 second, every second or every 10 seconds, etc., we obtain a rather unexpected result. Instead of smoother and smoother arrivals as we would expect, we always observe a process that is almost as variable as the one observed at the finer scales. This property of the variance in packet or byte arrivals in Internet traffic, which is known as *self-similarity* or *scale-invariance*, holds true for scales from a few hundred milliseconds up to hundreds of seconds. Quantitatively, the decay in the variance of arrivals for self-similar traffic is proportional to m^{2H-2}, where $m \geq 1$ represents

scale as the aggregation of arrival counts, and H is known as the *Hurst* parameter. For the time series of bin counts generated using a Poisson process $H=0.5$, while $H \in (0.5,1)$ for a stationary, long-range dependent process. The closer the value of the Hurst parameter is to 1, the more slowly the variance decays as scale (m) increases, and the traffic is said to be more *bursty*. The slow decay of the arrival variance, as scale increases in self-similar traffic, is in sharp contrast to the mathematical framework provided by Poisson modeling, in which the variance of the arrivals process decays as the square root of the scale (see [LTWW94, PF95]).

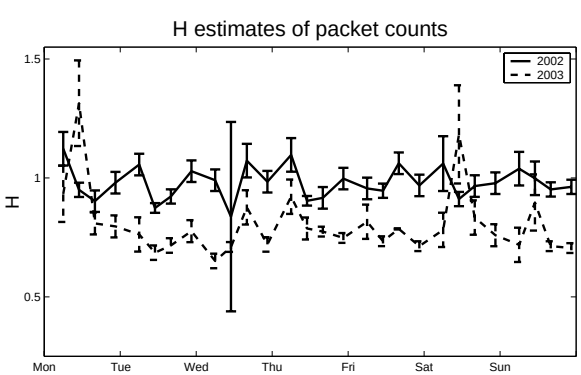
Self-similarity also manifests itself as *long-range dependence* (or *long memory*) in the time-series of arrivals. This means that there are non-negligible correlations between the arrival counts in time intervals that are far apart. More formally, the autocorrelation function, $\rho(k)$, of long-range dependent time-series decays in proportion to $k^{-\beta}$ as the lag k tends to infinity, where $0<\beta<1$. The Hurst parameter is related to β via $H=1-\beta/2$, so the closer the value of the Hurst parameter is to 1, the more slowly the autocorrelation function decays. In contrast, Poisson models are short-range dependent, *i.e.*, their autocorrelation decays exponentially as the lag increases. The implied "failure of Poisson modeling" for Internet traffic spawned an active field of research in analysis of network traffic. Some of the research closely related to this paper are reviewed in section 2.

One of the major strengths of the early studies [refs] was that they were based on a significant number of high-quality network traces (high quality in the sense that they captured hours or days of operation on production networks and were recorded with reasonably accurate and precise timestamps for each packet). In recent years, however, there have been few studies that examined network traffic in the modern Internet using empirical data comparable in quantity and quality to the earlier studies (a few exceptions, notably from Sprint Labs, are described in Section 2). There are several reasons for this decline in studies based on empirical data. One is that network links have dramatically increased in speed from the 10 Mbps Ethernets monitored for the early studies to the 1000 Mbps Ethernets and 2500 Mbps (OC-48) or faster technologies commonly deployed in today's access and backbone network links. Capturing traces, even when such links are lightly loaded and the traces include only packet headers, is costly in terms of the required monitoring equipment and disk storage. Another important issue is that network service providers regard as proprietary or trade secrets any empirical data that would reveal information about the operational characteristics of their networks. Similarly, the service providers, including those that are part of universities or other research organizations, have to be concerned about privacy issues. Because of these factors, there

are relatively few suitable network traces publicly available to researchers (the NLANR repository [ref] does contain a few such traces that have been used in some of the studies reviewed in Section 2).

In this paper we present the results from analysis of an extensive set of two-hour long traces collected in 2002 and 2003 from a high-speed (1000 Mbps) access link connecting the campus of the University of North Carolina at Chapel Hill to its Internet service provider (ISP). The traces were started at the same four times each day for a week-long period. In aggregate, these traces represent 56 hours of network activity during the second week of April in 2002 and 56 hours during the same week of April in 2003. The 2002 traces include information about 5 billion packets that carried 1.6 Terabytes of data while the 2003 traces represent almost 10 billion packets and 2.9 Terabytes (the load on our Internet link almost doubled in one year). The details of the trace acquisition methods and summary statistics of the traces are presented in Section 3.

Figure 1 provides a visual summary of the estimated Hurst parameter (H) and 95% confidence intervals (CI) for each of the traces at the day and time (x- axis) that the trace was obtained. The most surprising observation is that the 2003 packet-count time series generally had significantly lower estimated H values than in 2002. The H estimates for the byte counts were, however, virtually identical between the two years which is also unexpected because previous studies have shown packet and byte counts to be closely related. Investigating the reasons for these differences in H estimates is the major theme of this paper. We also observed a number of traces that exhibited large H values (> 1.0) that indicated non-stationarity (other than simple linear trends) or even more complex structure, or that had very wide confidence intervals for the H estimates. Several examples of these are examined in detail in section 4.



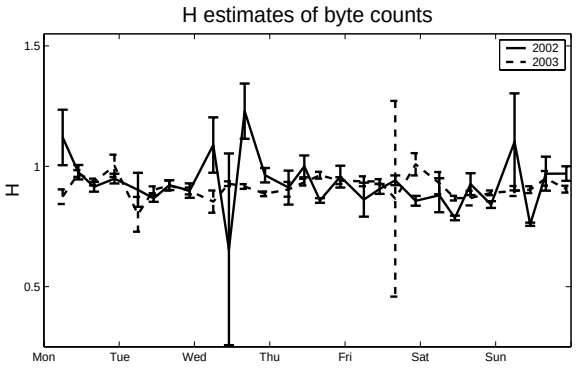


Figure 1. Hurst estimates and confidence intervals for packet and byte counts by day of week and time of trace start. Clearly the H estimates for packet counts are significantly lower in 2003 while the H estimates of byte counts show little differences between the two years.

The statistical tools used include wavelet-based spectrum and Hurst parameter estimation as a simple quantification of relative long range dependence, dependent SiZer analysis for specific investigation of structure beyond that indicated by the Hurst parameter, a cross trace summary plot of SiZer structure for use in the simultaneous analysis of a large data set, and the study of queueing processes that are driven by empirical time series. We used wavelet-based analysis tools for working with long-range dependent time series that were developed by Abry and Veitch [refs]. SiZer was invented by Chaudhuri and Marron [ref], and the dependent data version that is used in Section 4 was developed by Park, Marron and Rondonotti [ref]. The queueing analysis approach treats an empirical network trace as the input to an abstract infinite-buffer queueing system using traffic intensity (the ratio of arrival rate to service rate) as the primary parameter that can be varied. With this approach, important insights come from the tail distributions of queue sizes (using a complementary cumulative distribution function, CCDF) generated by long-range-dependent traces at different traffic intensities.

Our primary findings were:

- Hurst parameter estimates for the packet-counts in 10 millisecond intervals decreased significantly between 2002 and 2003 while the Hurst parameter estimates for byte counts remained about the same.
- A single peer-to-peer file sharing application (*Blubster*) that suddenly became popular between 2002 and 2003 had a strong influence on the Hurst parameter for packet counts in the aggregate traffic and caused the differences in H between the two years. This happened because the application sends a large number of small packets with high frequency that, in turn, produces high variance at certain time scales in the wavelet analysis.
- With equal traffic intensities, queue lengths in simulations driven with the actual traces of byte counts were consistently larger with 2003 traces (although the mean and variance of the 2003 traces is also higher). Qualitatively, the decay of the queue length CCDF for bytes counts appears similar between 2002 and 2003 traces. Using packet counts, on the other hand, shows noticeable qualitative differences between the years
- Moderate to strong long-range dependence (Hurst parameter estimate in the range [0.65, 0.99]) existed in the vast majority of traces.
- In several of the 56 traces studied, we found evidence of highly bursty traffic in both packet and byte counts that could not be modeled as Fractional Gaussian Noise.

- The long-range dependence, as measured by the Hurst parameter estimate, was not influenced by the day-of-week or time-of-day. This implied that changes in the ***active*** user population (and, by implication, types of applications used) did not influence long-range dependence.
- The long-range dependence as measured by the Hurst parameter estimate was not influenced by the traffic load (link utilization) or the number of active TCP connections. In particular we found no indication of declining strength in long-range dependence as more active TCP connections are aggregated.

.

The remainder of this paper is organized as follows. Section 2 briefly reviews some prior work that is closely related to the themes of this paper. Section 3 describes details of the tracing methods and presents summary statistics of all the trace data. Section 4 gives a brief summary of long-range dependent time series and the wavelet-based methods for analysis. It also presents the results for aggregated packet- and byte-count time series for all the traces. In Section 5 we present results for protocol-dependent subsets focusing on the differences between the time series of all packets and bytes compared with just TCP or UDP. We also show how a single peer-to-peer application influences the Hurst parameter. Section 6 gives results from a queueing analysis to assess the implications for router's buffers of the long-range dependence found in our traces. Section 7 summarizes our findings.

2. Review of Related Work

Various research directions were pursued, including those seeking evidence for pervasive long-range dependent properties [refs], finding physical explanations for the causes [refs], investigating the implications for network and protocol designs as well as performance analysis [refs], and examination of scaling behavior at different time scales [refs]. The book by Park and Willinger [25] (and the references therein) is an excellent resource and guide to the results in this area.

3. Traffic Measurements and Data

The two trace collections were obtained by placing a network monitor on the high-speed link connecting the University of North Carolina at Chapel Hill (UNC) campus network to the Internet via our Internet service provider (ISP). All units of the university including administration, academic departments, research institutions, and a medical complex (with a teaching hospital that is the center of a regional health-care network) used a single ISP link for Internet connectivity. The user population is large (over 35,000) and diverse in their interests and how they use the Internet — including, for example, email,

instant messaging, student "surfing" (and music downloading), access to research publications and data, business-to-consumer shopping, and business-to-business services (e.g., purchasing) used by the university administration. There are tens of thousands of on-campus computers (the vast majority of which are Intel architecture PCs running some variant of Microsoft Windows). For several years the university has required all incoming freshmen to have a personal computer, typically a laptop with wired or wireless network interfaces. There are over 250 wireless access points operating at 11 Mbps throughout the campus and a significant fraction of Internet traffic transits the wireless network. Wired connections are 100 Mbps and are pervasive in all buildings including the dormitories. The core campus network that links buildings and also provides access to the ISP is a switched 1 Gbps Ethernet.

We used a network monitor to capture traces of TCP/IP headers from all packets entering the campus network from the ISP (we call these packets "inbound" traffic in this paper). All the traffic entering the campus from the Internet traversed a single full-duplex 1 Gbps Ethernet link from the ISP edge router to the campus aggregation switch. In this configuration, both "public" Internet and Internet 2 traffic were co-mingled on the one Ethernet link and the only traffic on this link was traffic from the ISP edge router. We placed a monitor on this 1 Gbps Ethernet by passively inserting a fiber splitter to divert some light to the receive port of a gigabit Ethernet network interface card (NIC) set in "promiscuous" mode. The *tcpdump* program was run on the monitoring NIC to collect a trace of TCP/IP packet headers. The trace entry for each packet includes a timestamp (with microsecond resolution), the length of the Ethernet frame, and the complete TCP/IP header (which includes the IP length field).

For the 2002 traces, the monitor NIC was hosted in a Intel-architecture server-class machine configured with a 1.0 GHz processor, 1 GB of memory, 64 bit-66 MHz PCI busses, six 10,000 RPM 31GB disks and running FreeBSD version 4.1. Buffer space of 32KB was allocated to the *bpf* device used by *tcpdump* to buffer transient overloads in packet arrivals. The *tcpdump* program reports statistics on the number of packets dropped by *bpf*. We found that in only 8 of the 28 traces was the drop rate 0.01% or less (1 packet per 10,000) and the drop rate was as high as 0.16% in one trace. For the 2003 traces we upgraded the monitor machine to a 1.8 GHz processor, 64-bit 100 MHz busses and five 15,000 RPM 35 GB disks. The *bpf* buffer was set to 16 MB and the *tcpdump* program modified to use block writes of 512 bytes. As a result no packets were dropped in any of the 2003 traces in spite of an almost two-fold increase in the load on the link.

The traces were collected during four two-hour tracing intervals each day for seven consecutive days of the week (28 traces per collection period). Both the 2002 and 2003 collection periods were during the second week in April, a particularly high traffic period on our campus coming just a few weeks before the semester ends. Collecting traces during the same week in April of both years allowed us to compare results from traces gathered approximately one year apart. The two-hour intervals were 5:00-7:00 AM, 10:00-12:00 noon, 3:00-5:00PM, and 9:30-11:30 PM. These periods were chosen somewhat arbitrarily to produce two traces near the beginning and end of the normal business day, and two during non-business hours when traffic volumes were the lowest (5:00-7:00 AM) or "recreational" usage was likely to be high (9:30-11:30 PM).

Appendix A provides summary information about the complete set of 56 traces including the total packets and bytes with a breakdown by transport protocol, TCP *vs* UDP. Also included are the average link utilization (relative to an idealized 1 Gbps transmission speed) and an estimate of the median number of TCP connections per minute over the tracing interval[1]. We found that the highest average link utilization almost doubled (from 10% to 18%) between 2002 and 2003 reflecting the growth in Internet usage on the campus. Likewise, the number of active TCP connections per minute also increased substantially in 2003 compared to 2002. Perhaps the most striking observation was the growth in UDP over the period of a single year from around 5% of packets and 7% of bytes to about 25% of packets and 14% of bytes. Clearly, the growth came from UDP applications that generated a smaller average packet size than had previously been present. Furthermore, data collected from this same link for the previous 5 years had shown that the average usage of UDP in both packets and bytes had remained nearly constant at levels similar to the 2002 data. We explore the causes and implications of this growth in UDP in Section 5.

4. Analysis of packet and byte counts

The concepts and definitions for self-similarity and long-range dependence given in section 1 assume that the time-series of arrivals is *second-order stationary* (a.k.a. *weakly stationary*). Loosely speaking, this means that the variance of the time-series (and more generally, its covariance structure) does not change over time, and that the mean is constant (so the time-series can always be transformed into a zero-mean stochastic process by simply subtracting the mean). The obvious question this raises is whether Internet traffic is stationary. This is certainly not the case at the scales in which the time-of-day effects

[1] Estimating valid TCP connections per minute is complicated by the presence of SYN-flooding and port-scanning attacks. We estimated the number of TCP connections per minute by computing the median of min(count of TCP segments with SYN flag, count of TCP connections with FIN or RST flag) for each 1 minute interval. SYN-flooding and port-scanning attacks usually have abnormally high counts of SYN or RST segments in one direction but not both.

are important (traffic sharply drops at night), so Internet traffic is characterized as self-similar and long-range dependent only for those scales between a few milliseconds and a few tens seconds. Furthermore, we often find trends and other effects even at these finer time scales. For example, many links show an increase in traffic intensity during early morning hours as more and more people get connected. However, it is still useful to study time-series using the self-similarity and long-range dependence framework, and this is possible using methods that are *robust* to non-stationarities in the data (i.e., methods that first remove trends and other effects from the data to obtain a second-order stationary time-series). In some cases, non-stationarities are so complex, that conventional models fail to accommodate them, which can result in estimates of the Hurst parameter greater than or equal to 1.

The wavelet-based tools for analysis of time series are important because they have been shown to provide a better estimator (and confidence intervals) than other approaches for the Hurst parameter that characterizes long-range dependent processes. These methods also are robust in the presence of certain non-stationary behavior (notably linear trends) in the time series. This is particularly important in working with traces having durations of two hours because we observed linear trends caused by diurnal effects in many of them (e.g., a ramp-up of network activity between 10:00 AM and noon).

For the results in this paper we used the Abry and Veitch methods (and MATLAB software) [30] to study the wavelet spectrum of the time series of packet and byte counts in 10 millisecond intervals. The output of this software is a *logscale diagram*, i.e., a *wavelet spectrum*, which provides a visualization of the scale-dependent variability in the data. Briefly, the logscale diagram plots the $\log_2$ of the (estimated) variance of the Daubechies wavelet coefficients, at the scale j, for the time series (I), against $\log_2(j)$. The wavelet coefficients are computed for scales up to 2^{16}. Since the scale effectively sets the time-aggregation level at which the wavelet analysis is applied, there is a direct relationship between scale and time intervals. For processes that are long-range dependent, the logscale diagram will exhibit a region in which there is an approximately linear relationship with positive slope at the right (coarser scale) side. An estimate of the Hurst parameter, H, along with confidence intervals on the estimate can be obtained from the slope of this line (H=(slope+1)/2). For a more complete summary of long-range dependence and wavelet analysis see also Chapter 2 of [25], [ref]. and [30].

4.1 Summary Results

Each of the traces was processed to produce two time series, one for packet counts and one for byte counts, both at 10 millisecond intervals. A ten millisecond interval is well within the accuracy of our monitoring infrastructure. For each of the

resulting time series, we used the wavelet analysis tools to determine if the time series exhibited long-range dependence and to compute the Hurst parameter estimate (H) along with 95% confidence intervals (CI). We set the parameter (N) that controls the number of vanishing moments of the wavelet to three as recommended in [ref] in order to avoid potential bias caused by the linear trends we observed in most of the two-hour traces. In the vast majority of cases the estimated H values and their confidence intervals fall in the range [0.65, 0.99] indicating moderate to strong long-range dependence. There are a few values of the estimated Hurst parameter in the range [1.0, 1.1] but where the CI range includes values < 1.0. We found a few others that represent more extreme cases in the sense that either the estimated H value was > 1.0 (indicating non-stationarity beyond a linear trend) or the confidence interval was extremely wide. We look at some of these in more detail in section 4.2

Figure 1 (in section 1) provides a visual summary of the estimated Hurst parameter (H) and confidence intervals (CI) for each of the time series at the day and time (x- axis) that the trace was obtained. There are several interesting features, notably that the 2003 packet count time series generally had significantly lower estimated H values than in 2002 even though the link utilizations and TCP connection rates were substantially higher. The H estimates for the byte counts were, however, virtually identical between the two years. The reasons for these differences in H estimates are further explored in Section 5. There did not appear to be any consistent relationships between the day of week or the time of day and the estimated value for H. This implied that changes in the makeup of the ***active*** user population at different days and times (i.e., changes in the proportions of faculty, staff and student users) and, by implication, types of applications used, did not influence long-range dependence. Further, night hours were expected to have more "recreational" uses of the network but this appeared to have no effect.

Figure 9 shows the estimated value of H for packet and byte counts as a function of the average link utilization during the trace interval. Figure 10 shows the estimated value of H for packet and byte counts as a function of the average number of TCP connections per minute. The H estimates for packet counts in 2003 again were lower than those in 2002 for ranges of utilization and TCP connection rates that overlap between the two years. For byte counts, the two years were again comparable in ranges of utilization and TCP connection rates that overlap. This indicates that the differences between 2002 and 2003 observed in Figure 1 are not related to the change in traffic loads on the link. Clearly both link utilization and TCP connection rates increased substantially between 2002 and 2003 but there did not appear to be any evidence of the Hurst estimate being influenced by changes in link utilization or TCP connection rates. Further, considering the 2003 traces alone, we find that even though the TCP connection rate ranged from about 13,000 per minute in one trace to 59,000 in another, no

influence on H is evident. In fact, the H values for packets and bytes in the trace with the lowest TCP connection rate were 0.78 and 0.93, respectively while the H values for the trace with the highest TCP connection rate were 0.88 and 0.92. Overall, we found no evidence that link utilizations or the aggregations of large numbers of TCP connections had any effect on long-range dependence as measured by the Hurst parameter.

`

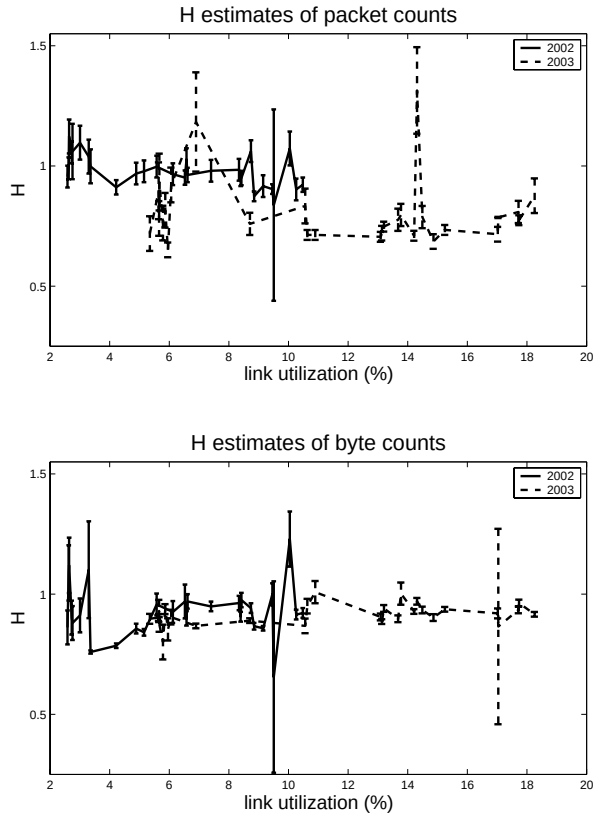


Figure 9. Hurst estimates and CI for packet and byte counts by link utilization

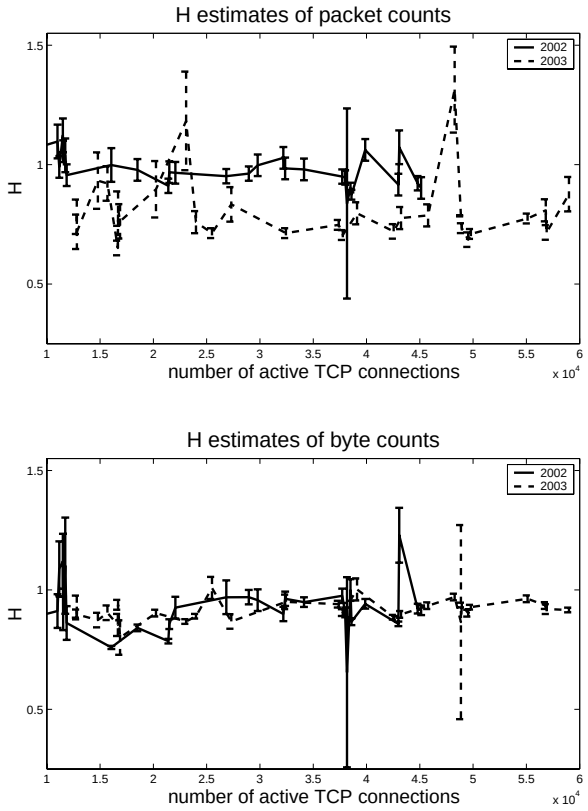


Figure 10. Hurst estimates and CI for packet and byte counts by TCP connections

4.2 SiZer Analysis of Extreme Cases in Hurst Estimation

We selected six of the most extreme examples of H estimates > 1.0 or wide confidence intervals to examine in more detail using the SiZer tools. They are:

2002 Wed 1000 packets	0.84 [0.44, 1.24]
2002 Wed 1000 bytes	0.65 [0.26, 1.05]
2002 Wed 1500 bytes	1.23 [1.11, 1.34]
2003 Mon 1000 packets	1.31 [1.13, 1.49]
2003 Fri 1500 bytes	0.87 [0.46, 1.27]
2003 Sat 1000 packets	1.18 [0.98, 1.39]

Table III. Traces with H estimates or CI ranges that represent extreme examples

We carefully examined these six time series for additional structure not consistent with long-range dependence using the SiZer approach to exploratory data analysis. The results of this analysis are shown in old Figures 1-4. The 2002 Wed 1000 bytes analysis is not shown here because it is very similar to 2002 Wed 1000 packets analysis. Similarly the 2003 Mon 1000 packets analysis is not shown here because the main ideas are similar to 2002 Wed 1500 bytes analysis. SiZer provides a useful method for finding "statistically significant trends" in time series. The original SiZer tool was developed by Chaudhuri and Marron (1999), and the dependent version, used here, is due to Park, Marron and Rondonotti (2004). SiZer is based on local linear smooths of the data, shown as curves corresponding to different window widths in the top panel of Figure 1 (a random sample of time series values is also shown). These curves are very good at revealing potential local trends in the time series. See, for example, Fan and Gijbels (1996) for introduction to local linear smoothing. Two important issues are: which of these curves is "the right one", and which of the many visible trends are "statistically significant" as opposed to "reflecting natural variation"?

Choice of the window width in curve smoothing is well known to be a very hard problem in statistics (and has been quite controversial, e.g., Jones, Marron and Sheather (1996) and Marron (1996)). SiZer avoids this problem by taking a scale space approach, where a large family of smooths, indexed by the window width is used. Essentially each of these curves represents a different "level of resolution of the data", and all are important, and thus should be used in data analysis. The second part of SiZer (shown in the lower panel) is a graphical device for doing the needed statistical inference, in particular flagging trends in the smooths when they are statistically significant. This is a color map, where the horizontal axis represents time, and thus is the same as the horizontal axis in the top panel. The vertical axis represents scale of resolution, i.e. window width of the local linear smooth, with the finest scales at the bottom. At each scale-time location (i.e. pixel in the map) statistical inference is done on the slope of the corresponding curve. When there is a significantly positive slope, the

dark color is used. Where the slope is significantly negative, the pixel is shaded light. When there is no significant slope, i.e. the variation is "what is naturally expected", an intermediate gray shade is used.

An important issue is the definition of "natural variation". The original SiZer of Chaudhuri and Marron (1999) tested against a null hypothesis of white noise (i.e. independent Gaussian). This is clearly inappropriate for Internet traffic, because such time series typically have large amounts of serial dependence. Thus conventional SiZer maps would show a large amount of "significant structure" (i.e. lots of light and dark regions). SiZer becomes more useful for such time series, when the null hypothesis is changed to a more typical model for network traffic, such as a Fractional Gaussian Noise (FGN). This is the idea behind the Dependent SiZer of Park, Marron and Rondonotti (2004). The SiZer map in the bottom panel of Figure 1 uses dependent SiZer, assuming an underlying FGN with Hurst parameter of H=0.8. This Hurst value is quite reasonable, because it is typical of those shown in Tables I and II. The FGN model also requires a variance parameter, which we estimate from the time series as discussed in Park, Marron and Rondonotti (2004) and Hernandez Campos, et al (2004).

The old Figure 1 shows a dependent SiZer analysis of the 2002 Wed 1000 packet count time series. Recall that this was seen to have a very wide confidence interval on the Hurst parameter estimate in old Figure 8. The SiZer map in the bottom panel shows darker color near the top, i.e. at the coarser scales. This shows that the upward trend visible in the family of smooths is statistically significant. There is also some significant fine scale behavior flagged as significant in the lower right of the SiZer map, that is seen to correspond to a substantial valley in the family of smooths. The colors in the SiZer map show that both the linear upward trend and the sharp valley are features that are not typically expected from FGN. However, the other spikes and valleys in the family of smooths correspond to medium gray regions in the SiZer map, and thus represent a level of variation that is natural for the FGN model. The significant non-FGN behavior is consistent with the poor behavior of the Hurst parameter estimation process. It is not clear if the overall upward trend, or the small sharp valley contributes more to the unusually wide confidence interval.

The SiZer analysis for 2002 Wed 1000 byte count time series, which also had an unusually large confidence interval for the Hurst parameter estimate, is quite similar to the one shown in old Figure 1. This is not surprising, because when packet sizes are nearly constant (very common for TCP traffic) the two time series tend to be correlated. As in Figure 1, there is a strong upward trend at the coarsest scales, but this time no small sharp valley. This suggests that the upward trend may be the main contributor to the wide confident confidence intervals. At first glance it is tempting to view the upward trend as linear, which

should have no effect on the Hurst parameter basis, because such effects are filtered out by the wavelet basis used in the estimation process. But a closer look shows that these trends are non-linear in nature, and coarse scale instability seems to be what is driving the large confidence intervals.

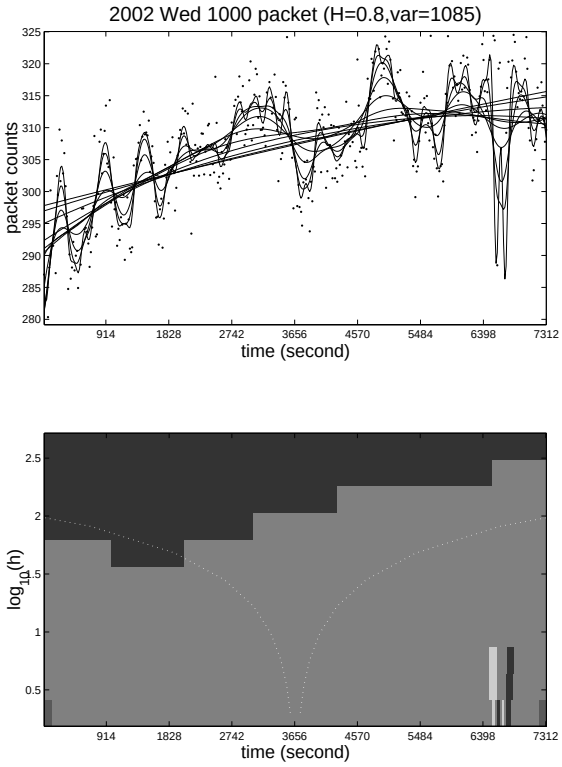
Old Figure 3 shows the corresponding dependent SiZer for the 2002 Wed 1500 byte count time series. This data set attracted attention because of the unusually high value of the estimated Hurst parameter highlighted in old Figure 8. Here there is less coarse scale trend, but more medium and fine scale color in the SiZer map. The family of smooths shows that this time the latter colors are generated by some large spikes. Again the SiZer maps shows that these spikes are much larger than would be typical for a FGN model, while the other spikes are consistent with FGN.

The SiZer analysis for the 2003 Mon 1000 packet count series, which also gave an unusually large Hurst parameter estimate, was qualitatively similar (and thus is not shown to save space). Again there were several sharp spikes, which were flagged as too large to be explained by FGN by SiZer. This suggests that a potential cause of unusually large Hurst parameter estimates is a number of sharp changes that are too large to be consistent with FGN.

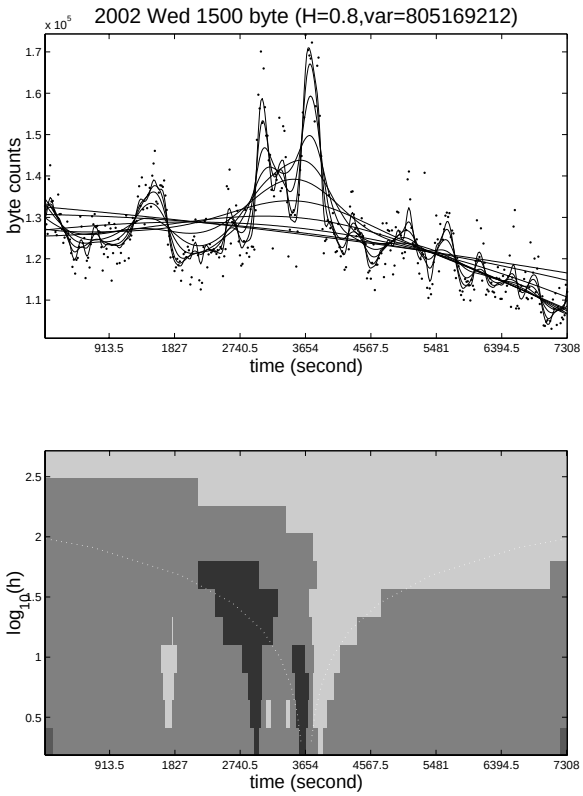
Old Figure 5 is a parallel analysis for the 2003 Fri 1500 byte count time series. This case was seen to be special in old Figure 8 because of an unusually wide confidence interval for the estimated Hurst parameter. Now the SiZer structure is quite different, with this time a number of smaller peaks and valleys being flagged as statistically significant. This shows that other departures from FGN can also generate a too wide confidence interval for the Hurst parameter.

Old Figure 6 is another case with very large H, which is generated by a much different type of anomaly. Here the data are the packet count time series from 2003 Sat 1000. The large Hurst parameter estimate seems to be driven by the substantial non-stationarity of a long increase, followed by a very large dropoff, followed by a nearly constant region.

The overall lesson from old Figure 1 – old Figure 6 is that a number of types of departures from the FGN model can drive the observed large Hurst parameter estimates, and wide confidence intervals. Dependent SiZer provides a clear analysis of which aspects of the data drive this behavior

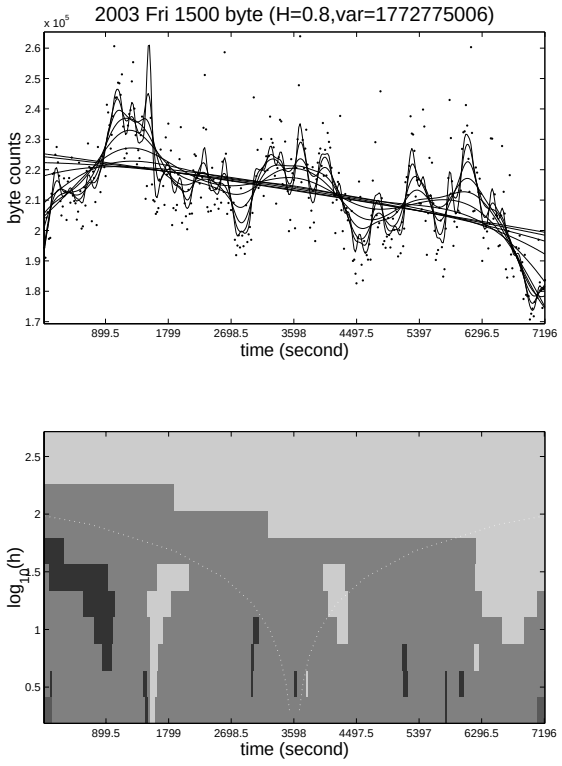


Old Figure 1. Dependent SiZer analysis of 2002 Wed 1000 packet counts. Shows wide confidence interval is due to significant, nonlinear upward trends

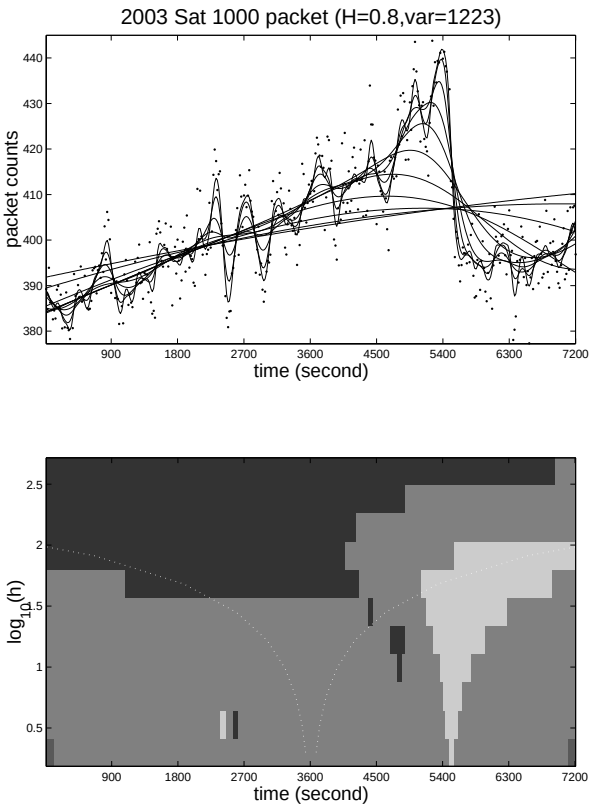


Old Figure 3. Dependent SiZer analysis of 2002 Wed 1500 byte counts. Shows large Hurst parameter is due to spikes larger than expected for FGN

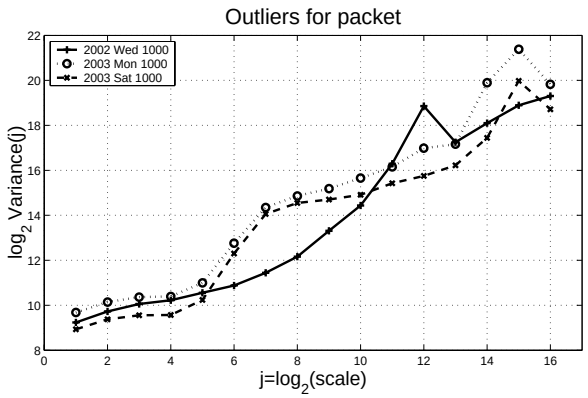
.



Old Figure 5. Dependent SiZer analysis of 2003 Fri 1500 packet counts. Shows wide confidence interval is due to many small scale statistically significant peaks and valleys.



Old Figure 6. Dependent SiZer analysis of 2003 Say 1000 packet counts. Shows large Hurst parameter estimate can be caused by other phenomena, such as large temporary increase, followed by a large drop



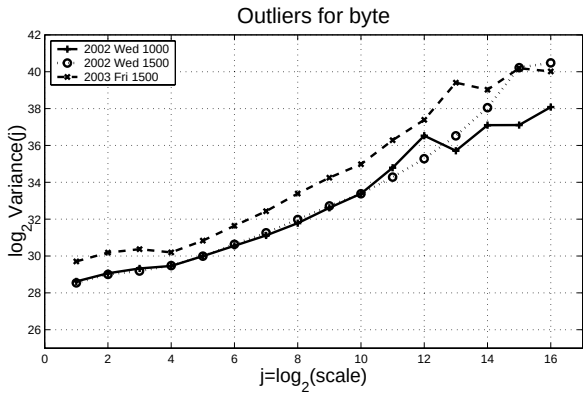


Figure 7. Logscale diagrams for the six traces that represent extreme examples of H estimates or CI width.

4.3 Analysis of Packet Counts Using SiZer Tools

We now consider possible reasons why the H estimates for the packet count time series were consistently higher for 2002 than for 2003. As we have seen in the previous section, the wavelet tools may sometimes be unable to provide a good estimate of the Hurst parameter in cases where there is significant structure or non-stationarity in the time series. One possibility for the differences in H estimates for packet counts between the two years is that some structural feature of the data is the cause. To investigate these issues we used SiZer to test each of the traces for statistically significant differences from a FGN process with H=0.80. We present here the results for only four of the traces, the time intervals on Thursdays at 3:00-5:00 PM (a heavy traffic period) and Saturdays at 9:30-11:30 PM (a light traffic period) for both 2002 and 2003. These traces are representative of the results we obtained from examining the complete set. Figures 11-14 show the SiZer outputs for the time series of packet counts from these four traces.

In Figure 11, the SiZer map shows "natural variation" for the first two thirds of the time span. However the shape decrease about three-quarters of the way through the time interval, around 16:30, is statistically significant, as indicated by the large light region, across a broad range of scales. The smaller dark region shows that there was also a significant increase around 16:40. Figure 11 shows that the FGN model gives a reasonable fit over the first two-thirds of this time series, but is inadequate to explain the large valley in the latter third (i.e.. the deep valley is a type of variation that is larger than that

which is typical of FGN. Figure 12 shows the same analysis for Saturday, 21:30-23:30. While the family of smooths in the top panel looks qualitatively similar to that in Figure 11, the SiZer map in the bottom panel shows a rather different structure. In particular, there is nearly no statistically significant trend, i.e., FGN is essentially appropriate over this entire time series.

Figure 13 shows the same analysis, for data from the following year, 2003, for the same time period as Figure 11, Thursday 15:00 – 17:00. Compared to Figure 11, a very noticeable change is the clear downward trend in the coarsest scale smooths (which are close to being lines). This trend is seen to be statistically significant by the large amounts of light shade in the top rows of the SiZer map. Perhaps this downward trend represents an important difference in either user or network behavior between 2002 and 2003. Another difference between Figures 11 and 13 is that the total traffic has doubled, from around 300 packets in each 10-millisecond bin, to around 600 packets. Figure 14 provides a similar analysis, for Saturday, 21:30-23:30, 2003 (same time slot as Figure 12). While the family of smooths appears similar between 2002 and 2003, once again there is a significant downward trend, representing a major change in user or network behavior.

We have done Dependent SiZer analyses for all the time blocks shown in Tables I and II. The main observations are generally similar to what is illustrated here: there are time periods where the FGN model is appropriate (i.e. the SiZer map is all gray), and periods where it is not (dark and light features show up in the SiZer map). A summary of these, which allows some comparison of 2002 with 2003 is shown in Figure 15. This is a gray level plot, where the pixels represent scale and time using the same coordinates as in the individual SiZer maps. Each pixel has a gray level which represents the number of times that it was significant (either dark or light) over the entire collection. Darker shades of gray indicate more frequent significance. Comparison of the Dependent SiZer summaries for 2002 (top panel) and for 2003 (bottom panel) shows roughly equal amounts of non-stationary burstiness, of a type that can not be explained by FGN, for both years. The only difference is that there are more statistically significant trends at the coarsest scales for 2003, which is quite consistent with what was observed in Figures 11-14. This suggestion that levels of burstiness are similar for 2002 and 2003 is a sharp contrast with the results suggested by the Hurst parameter analysis, where there was an indication of significantly less burstiness in 2003 than in 2002. Therefore, we have to look elsewhere for explanations of the shift in H between 2002 and 2003. Since the most obvious difference was the growth of UDP usage as a transport protocol, we did a protocol-dependent analysis of the time series as described in the following section.

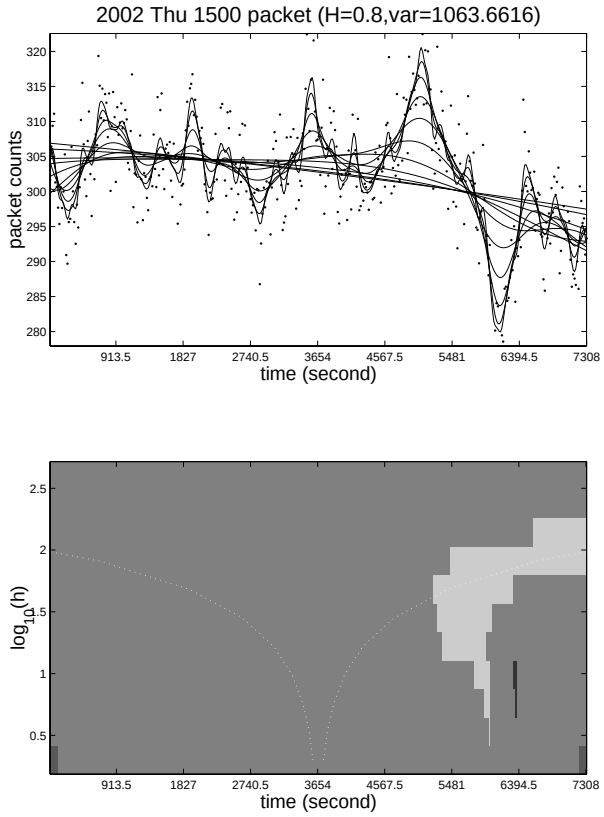


Figure 11. SiZer test for 2002 Thursday 3:00-5:00PM

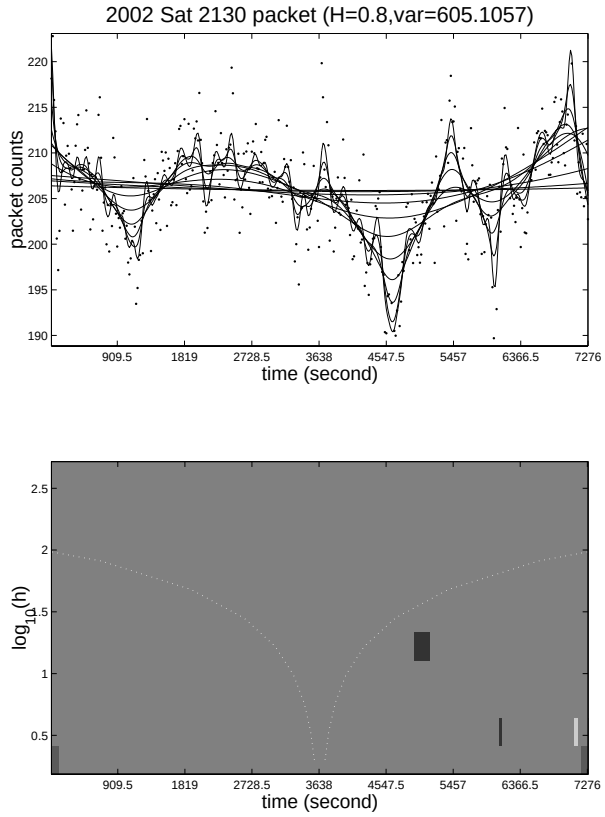


Figure 12. SiZer test for 2002 Saturday 9:30-11:30 PM

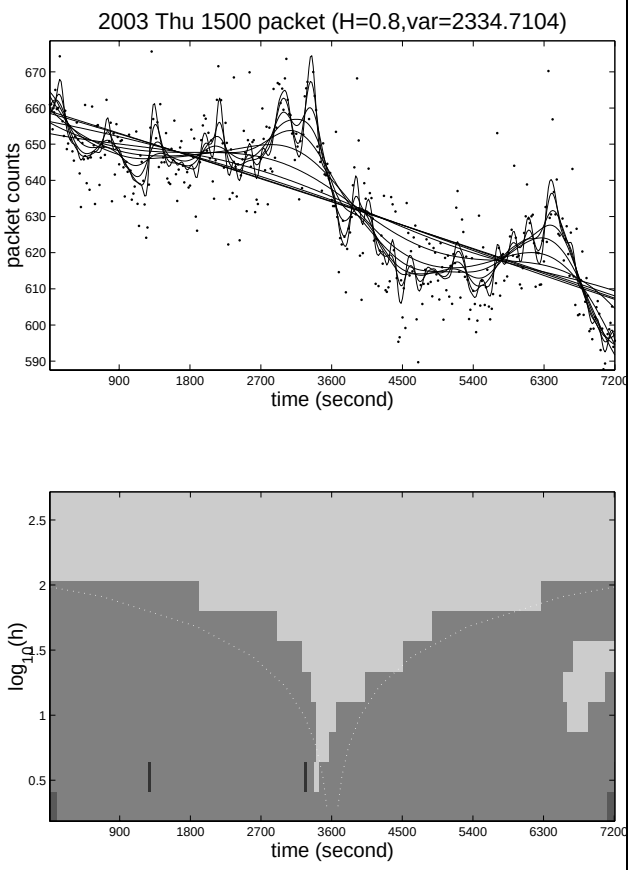


Figure 13. SiZer test for 2003 Thursday 3:00-5:00PM

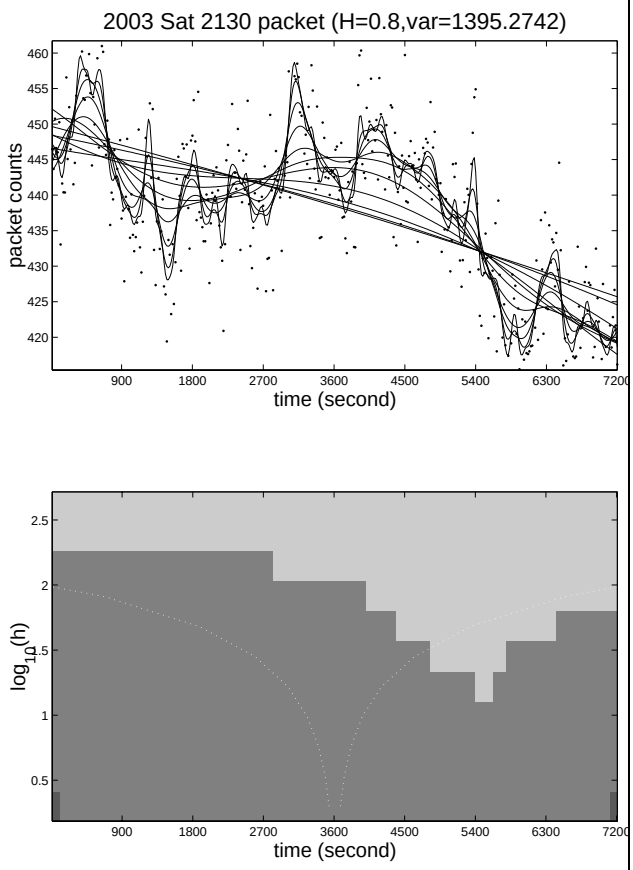


Figure 14. SiZer test for 2003 Saturday 9:30-11:30PM

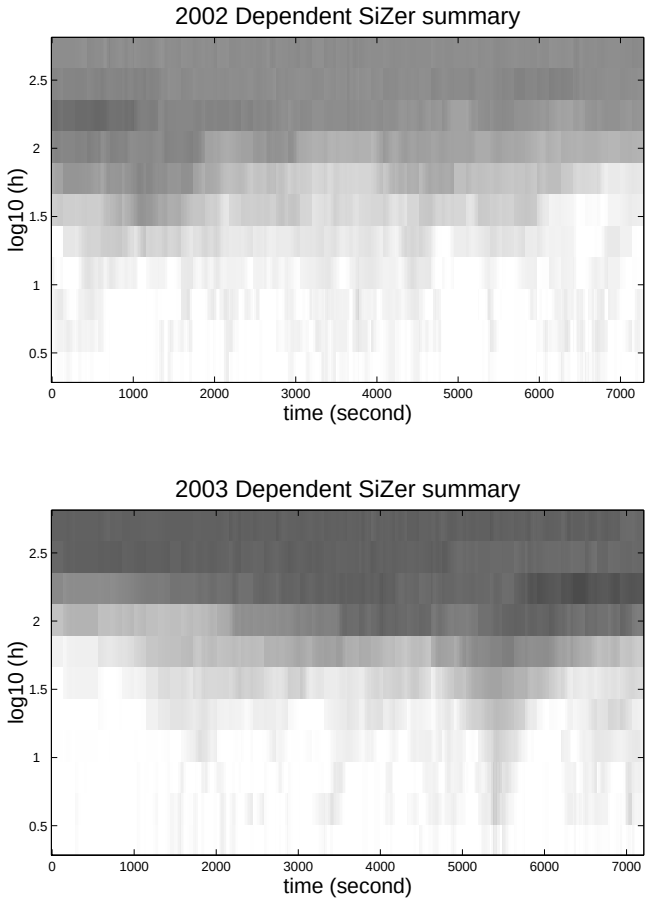


Figure 15. SiZer summary of all traces in 2002 and 2003

5. Protocol-Dependent Analysis

To examine more closely the factors that might affect the shift in long-range dependence in packet counts between 2002 and 2003, we also processed the traces to extract individual time series for packet and byte counts for the two major transport protocols of the Internet, TCP and UDP. These protocols can influence packet and byte arrivals, especially at small time scales (e.g., less than one second) [ref]. TCP mediates the data transmission rates that applications can achieve through window-based flow- and congestion-control mechanisms [ref]. UDP provides no flow or congestion control at the transport layer so applications using UDP must implement their own mechanisms. At larger time scales (above one second), traffic sources that have heavy-tailed sizes of application-defined data units (e.g., file sizes) which lead to heavy-tailed "ON" times, or that have heavy-tailed durations of "OFF" times, have been shown to be the basic causes of long-range dependence in packet counts at a link that aggregates packets from many such sources[ref].

We filtered each trace to extract a time series of packet or byte counts in 10 millisecond intervals for TCP packets only and likewise for UDP packets only. Because the link utilization was quite low for all traces and we are using 10 millisecond intervals, there should be little influence of TCP packets on UDP packets and *vice versa*[2]. We then computed the logscale diagram for each of the TCP and UDP packet- and byte-count time series. Figures 16-19 show the logscale diagrams for TCP and UDP along with the logscale diagrams for the complete packet traces containing both TCP and UDP ("All"). In these figures, all 28 time series for each year are overlaid in one plot. For both packet and byte counts in 2002 we found that the logscale diagram for "All" in each trace corresponds almost exactly to the one for TCP extracted from that trace. Each of the UDP logscale diagrams is completely distinct from its corresponding "All" or TCP trace. This was also the case for byte counts in 2003. Clearly, in these three cases UDP and UDP-based applications have little or no influence on the overall long-range dependence and Hurst parameter estimates reported above.

There was, however, a dramatically different result for 2003 packet counts. Notice that in Figure 18, the logscale diagram for "All" packet counts follows closely the shape of UDP especially in the time scales between 2^5 and 2^{10} (300 milliseconds to 10 seconds). Figures 20 and 21 illustrate this effect in more detail for two of the representative traces (Saturday at 9:30PM in 2002 and 2003) that were introduced earlier. Further, the steep increase in the variance of wavelet coefficients at these time scales happens near the beginning of the linear scaling region used to estimate the Hurst parameter. This implies that the slope will be smaller with a correspondingly smaller Hurst parameter estimate for packet counts. This explains the surprisingly smaller Hurst parameter estimates for 2003 as shown in Figure XX. In order to explain the reasons for these changes in UDP packet traffic between 2002 and 2003 and how they affect H, we examined the UDP traffic in more detail as described in the following.

Our 2003 traces show a much larger percentage of UDP packets and the UDP packets exhibit two interesting characteristics not present in UDP packets found in the 2002 traces:

- Most UDP datagrams have very small payloads (20-300 bytes).
- Port number 41,170 is either the source or the destination of a large fraction (e.g., 70%) of the UDP datagrams in each trace.

[2] At the 1Gbps speed of the monitored link, 2,500 packets of the average size 500 bytes can be transmitted in 10 milliseconds

Port number 41,170 is known to correspond to *Blubster* (a.k.a. *Piolet*)[ref], a music file-sharing application that became quite popular in early 2003. This suggests that the network behavior of this application is responsible for the remarkable change in the nature of UDP traffic in our 2003 traces, and this motivated us to have a deeper look at its characteristics. *Blubster* uses a proprietary protocol, so its network behavior is not documented in any RFC or public document. We had, however, examined numerous traces and a working copy of the program, and found that hosts running *Blubster* (i.e., peers) use UDP for constructing and maintaining a network of neighboring peers and for performing searches. Peers also use TCP for downloading web pages and banners, for bootstrapping[3], and for performing the actual file transfers.

The way *Blubster* uses UDP creates an unusual pattern of burstiness in traffic. Each member of the *Blubster* network sends a significant number of *ping* datagrams each second, between 40 and 125, with the purpose of updating its list of neighbors. The destination port number of each of these ping datagrams is 41,170, and they have very small payloads (20 bytes). Active neighbors reply to each ping with another small UDP datagram. We also observed that *Blubster* peers continued pinging our test peer even hours after the program was closed (the number of arrivals slowly decreases over time). *Blubster's* search mechanism employs UDP to exchange queries and responses, following a flooding algorithm similar to one implemented in *Gnutella*. Query datagrams contains little more that a search string, so their payload is generally very small too. If a peer has a file matching the search string, it replies with a UDP datagram that includes information about the quality of the found file and the connectivity of the peer in which it is located. This reply datagram is slightly larger than the other types of *Blubster* datagrams, but it is still relatively small (100-300 bytes). *Blubster* makes no other use of UDP (downloads use TCP), so the ping and search operations are fully responsible for the type of burstiness in the packet arrivals of the traffic. Furthermore, this application uses very small packets, so its impact is negligible in the time-series of byte counts (Figure 19) but very significant for packet counts (Figure 18).

Next, we analyze time-series of packet and byte counts for *Blubster* traffic. We obtained these time-series by first filtering packet header traces for UDP datagrams with one port being 41,170 and the other one being greater than 1023. The second port number condition is needed to filter out legitimate datagrams from other applications, such as DNS, that use well known

[3] Bootstrapping is a basic operation in peer-to-peer systems that do not rely on a central server to perform searches. During the bootstrapping phase, peers contact some specially designated node to obtain a first list of neighbors. In the case of *Blubster*, this node is actually a central web server that keeps tracks of the peers in the network. This host does not serve files or take part in the search algorithm.

port numbers (0-1023, [ref]). These applications can make use of UDP port 41,170 as a client port, but *Blubster* does not use well-known port numbers for its clients. As an example of the use of this protocol, between 3 pm and 4 pm on the Thursday trace, we observed UDP datagrams going to more than 3,000 IP addresses in the UNC campus sent from almost 40,000 IP addresses outside it[4].

We filtered the 2003 traces based on the *Blubster* port number to extract time series of packet and byte counts in 10 millisecond intervals for *Blubster* only and for all TCP and UDP minus *Blubster* ("Rest"). The logscale diagrams for these two time series, along with the original trace ("All"), from one representative trace (Saturday at 9:30 PM) from 2003 are shown in Figure 22. Clearly, the logscale diagram for the packet counts in the original trace is completely driven by the *Blubster* packets in the time scales between 2^5 and 2^{10}. Note that while there is a similar shape in the logscale diagram for *Blubster* byte counts, it is not reflected in the logscale diagram for the "All" byte counts. This is probably because this application uses a large number of small packets and contributes far less to the total bytes.

We recomputed the logscale diagrams for 2003 packet counts using only the non-*Blubster* packets (the "Rest" time series) and used them to estimate the corresponding Hurst parameter. We found that these new Hurst estimates were very comparable to those from 2002, mostly falling in the range [0.90, 0.99]. For example, the Hurst estimates and confidence intervals for "Rest" in the two representative traces were 0.94[0.92, 0.96] and 0.94[0.90, 0.99] instead of the estimates of 0.77[0.75, 0.79] and 0.76[0.71, 0.81] computed from the original ("All") trace. We used the SiZer method to investigate why the *Blubster* application produced such distinctive scaling behavior in the logscale diagram. Figures 23 and 24 show the SiZer analysis for one representative trace considering only the *Blubster* packets. We show three levels of detail; the full trace, a zoomed-in view of a 40 second interval at approximately 1000 seconds into the trace, and a zoomed-in view of a 10 second interval between 15 and 25 in the above 40 second interval. The SiZer views reveal a statistically significant (relative to white noise) high-frequency variability in the *Blubster* traffic with periods in the 1-5 second range, consistent with the time scales with heightened variance in the logscale diagram. Thus it is clear that a single peer-to-peer application with a particular UDP-based flooding-search algorithm strongly influenced the estimated Hurst parameter for an entire trace.

[4] Probably there were not 3,000 active users of *Blubster* during this interval because external peers are slow to purge their peer lists and continue to use IP addresses that may not be currently running *Blubster* or have been reassigned by DHCP.

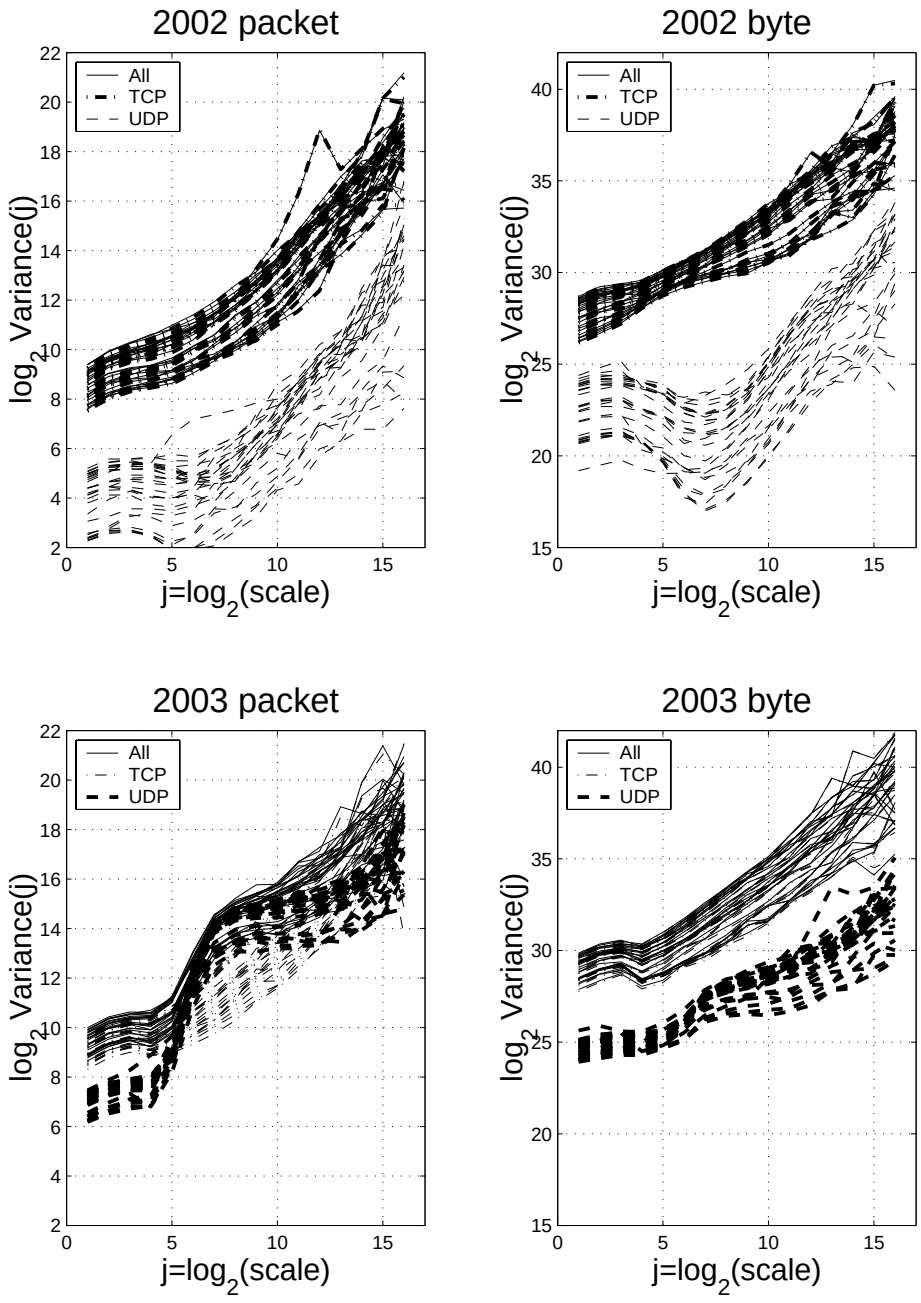


Figure 16-19. Logscale Diagrams for 2002 and 2003 Packet and Byte Counts (all traces)

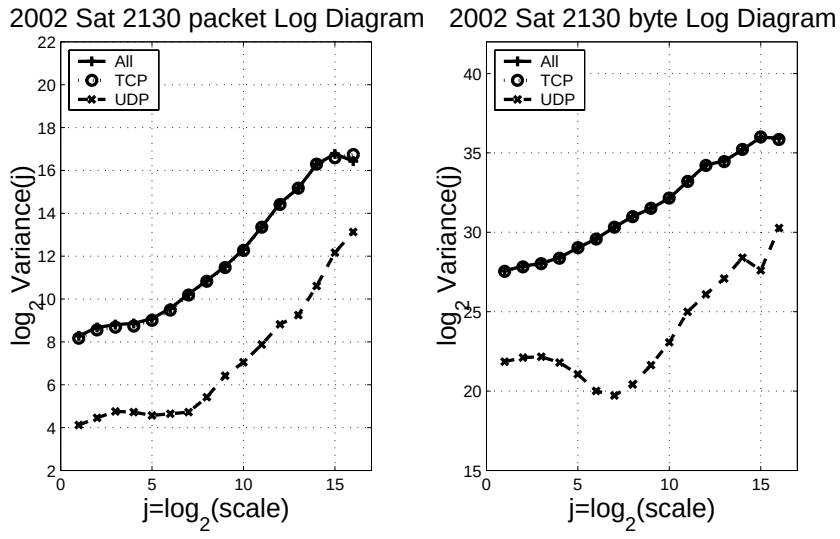


Figure 20. Logscale Diagram for 2002 selected traces

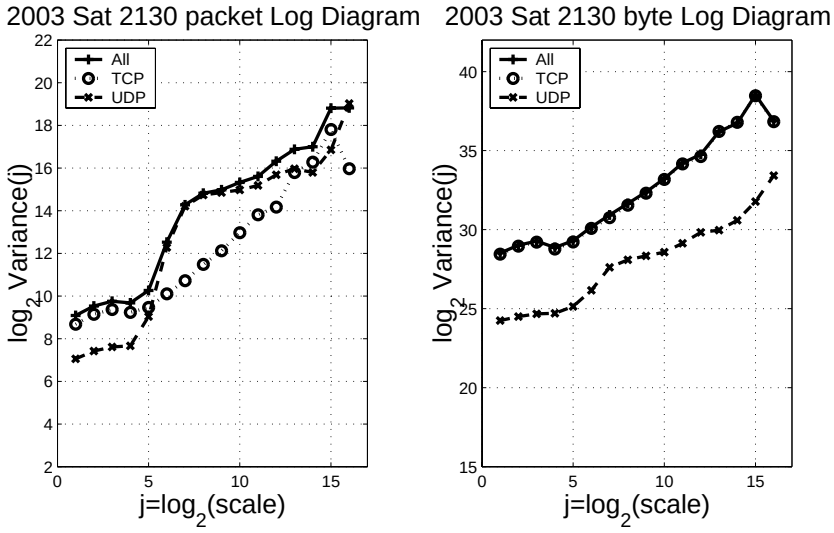


Figure 21. Logscale Diagram for 2003 selected traces

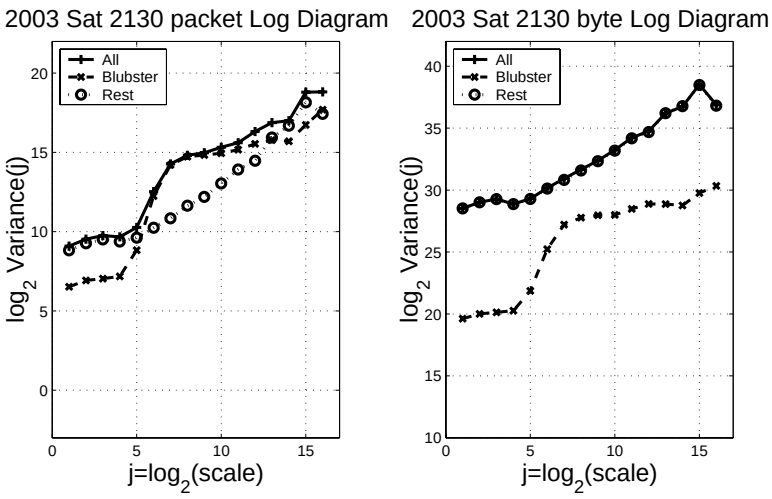


Figure 22. Logscale Diagram for 2003 *Blubster* packets and bytes

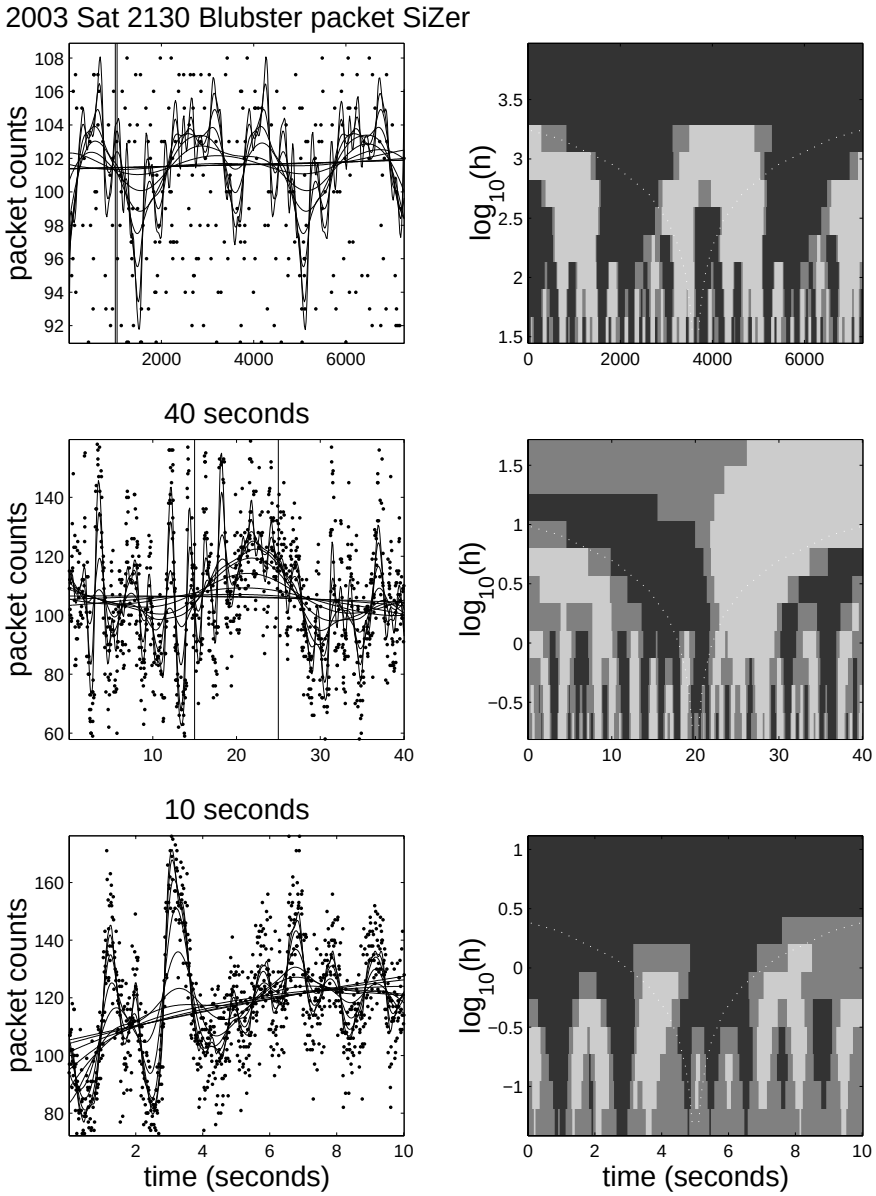


Figure 24. SiZer of 2003 Saturday 9:30 PM *Blubster*

6. Trace-Driven Queuing Analysis

The idea behind trace-driven queuing analysis is simply to consider traces in their natural setting. Since the behavior of queues within network elements has principal importance, how queues behave with traffic data is important. In other fields, like economics, non-queuing considerations are more important. Queuing behavior is influenced by the mean rate, variance,

marginal distribution, dependence structure and other factors in the input traffic. Exact theoretical results to obtain a precise understanding may not be straightforward, or may not be known.

6.1 Introduction to Trace-Driven Queuing Analysis

The traces analyzed here are time series of byte counts (or packet counts), say $\{X_n, n = 1, \ldots N\}$ at 10 millisecond resolution. That is, X_1 counts the number of bytes in the first 10 ms interval, X_2 counts the number in the second 10 ms interval and so on. A queue is simulated using the standard Lindley recursion formula

$$Q_n = \max(Q_{n-1} + X_n - C, 0), Q_0 = 0, n = 1, 2, \ldots$$

Here C is the server rate and specifies the number of bytes (packets) that can be served in one time unit. This formula imposes no upper limit on the queue length and one would need to modify it to simulate a buffer of finite capacity B.

To obtain a steady-state queue length one typically discards some number of initial values $Q_0, Q_1, \ldots Q_r$. The goal is to reduce the effect of Q_0 which propagates through the queue lengths. For the simulations used in this study the first 25,000 values were discarded and no upper limit on the queue size was imposed.

Although byte and packet counts are used here, it is also possible, and maybe preferable, to use the timestamps of packets along with their sizes instead. Timestamp data gives more precise simulations because each packet can be 'scheduled' to arrive precisely as it was measured. Timestamp data also allows study of the delay experienced by each arriving packet which isn't really possible with bin counts. Inherent in the idea of using counts is that features below the sampling resolution cannot be distinguished. A recent analysis of Internet data [**Error! Reference source not found.**] has suggested using timescales below 100 ms to see the effects of so called "micro-congestion episodes". Unfortunately, files with timestamp data can be quite large and unwieldy. A recent four-hour UNC trace produced a file of timestamps, packet sizes and a sequential connection ID numbers, with over 100 million packets, exceeding 2 GB.

If a stationary sequence $\{X_n\}$ has mean rate $m = \mathrm{E}[X_n]$ the utilization ρ is given by

$$\rho = \frac{m}{C}.$$

Generally the queue is called "stable" if $\rho < 1.$ If $\rho > 1$ the queue is "unstable" because the high average rate of incoming traffic overwhelms the server. It is common to specify C based on hardware specifications for a link (the bandwidth). Then the utilization is determined by m and C. In the analysis performed here the goal is to compare datasets from 2002 and 2003. The growth in the mean (and variance) complicates the comparison. An appropriate server rate for a 2002 trace could lead to an unstable queue using a 2003 trace, simply because the 2003 trace has higher mean. So the approach used here is to specify m and ρ and let C be determined to give the desired utilization. A technique to deal with different variances is shown in the following section in which the simulation results are discussed.

To study the queuing properties of data it is desirable to have a non-empty queue, at least some of the time. (In a queue that is always empty there is nothing to see.) For a given trace the way to cause more queuing is to increase the utilization, which is equivalent to lowering C closer to the mean rate m. With queuing simulation then it is not uncommon to use utilizations at 60%, 70% or higher, unlike real networks where utilizations may not exceed 25%. Again the goal is to cause queuing to see how the mean, variance, marginal distribution, dependence structure, etc., interact to affect queuing.

The distribution of the queue length is one of the quantities which can be studied using queuing simulations. (Another important quantity is the distribution of the delay). Theoretical results exist for the steady-state queue length complementary cumulative distribution function (CCDF). Assume that the input process of counts has finite variance. Then independence and weak-dependence in the input process is commonly associated with exponential decay in the queue length CCDF [**Error! Reference source not found.**, **Error! Reference source not found.**, **Error! Reference source not found.**]. Long-range dependence is commonly associated with "Weibull-like" decay in the queue length CCDF [**Error! Reference source not found.**, **Error! Reference source not found.**]. The Hurst parameter, H, manifests in the shape parameter such that higher H leads to slower decay in the queue length CCDF (i.e. larger queues are more likely). A common way to visualize the queue length CCDF is on a graph with log y vs. x axes. Exponential decay appears as a decreasing straight line while long-range dependence appears as a decreasing curve, concave up.

6.2 Queuing Analysis

This section presents results from trace-driven queuing simulations using the four datasets described above (Thursday at 3 PM, Saturday at 9:30 PM; 2002 and 2003). Queuing simulation provides a natural setting to see performance impacts of

differences in the traces. To minimize non-stationary effects, only the middle one hour of the two hour traces was used. For simplicity, we refer to the traces as above, using the starting time of the two-hour recording period.

The trace-driven queuing results described here provide further support for the conclusions reached above using wavelet-based and SiZer analyses. After allowing for an increase in the mean and variance, the queue length CCDF using byte counts shows a very similar shape between 2002 and 2003. This is consistent for the nine simulated utilizations between 50% and 95%. In contrast, the shape changes between 2002 and 2003. It is further demonstrated that using the aggregate, TCP-only, and UDP-only byte counts, the queue length CCDF curves generally have similar shapes in 2002 and also in 2003, for each utilization (allowing for changes in scale). Using the packet counts, the aggregate and TCP-only again give similar shapes while UDP-only gives a different shape for each utilization. In 2002 traces with a low percentage of UDP packets, this difference is very noticeable. (That is, UDP looks very different from the aggregate.) With 2003 traces, with a higher percentage of UDP packets, the difference in shape is less extreme.

Comparison of queue lengths using traces from 2002 and 2003, matched with the same weekday, time, (and even protocol) reveals consistent similarities using bytes that are not present using packets. Figure 101 shows the empirical CCDF curves for the queue lengths for Saturday at 9:30 PM from both 2002 and 2003 using simulated 80% utilizations. That is, the server rate was set to give 80% utilization based on the mean rate of the respective (2002 or 2003) time series of counts. With bytes counts, buffer sizes are larger with higher probabilities using 2003 traces. This is a consistent feature of direct comparisons with simulated utilizations from 50% to 90%. The most likely explanation is the higher variance of the traffic in 2003 (see below for additional explanation). Replacing the packet counts with byte counts (not shown) does not show the same consistency. Rather, with equal utilizations and the same horizontal axis, sometimes the queue length CCDF from 2002 is larger. There is not even a consistent pattern with increasing utilization. Rather, the year giving larger buffers changes back and forth between 2002 and 2003 with increasing utilization, in contrast to the consistency of 2003 byte counts always giving larger buffers, as shown in Figure 101.

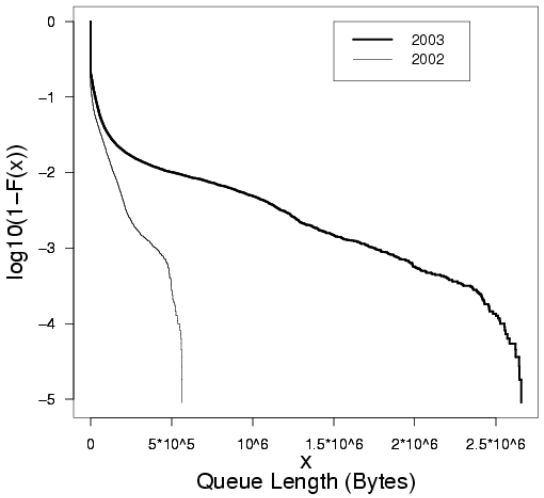


Figure 101: Empirical Queue Length CCDF for Saturday 2002 & Saturday 2003 at 9:30 PM, Aggregate Byte Counts, 80% Utilizations. 2003 shows larger buffers are more likely.

It is important to remember that the traffic in 2003 is "scaled up" from that of 2002. The mean of the aggregate and TCP-only byte counts are scaled up by a factor of 1.5–2.0. Variances are scaled up by a factor of 1.7–3.1. The UDP portion is scaled up more, by up to 4.6 times the mean, and 6.6 times the variance. To investigate the shape of the CCDF decay (while allowing for differing scales in the traces), the same curves from Figure 101 are shown again in Figure 102, but as overlaid curves, each with its respective horizontal axis, using scales in the ratio of 4.7:1. In this way the decay of the curves is revealed as qualitatively similar, which is consistent with the similarity of estimated Hurst parameters (0.84 in 2002, 0.89 in 2003). This similarity in the shape of the CCDF curves is also present in the Thursday traces. Using the packet counts, overlaid CCDF curves do not consistently show similar decay between 2002 and 2003 (not shown).

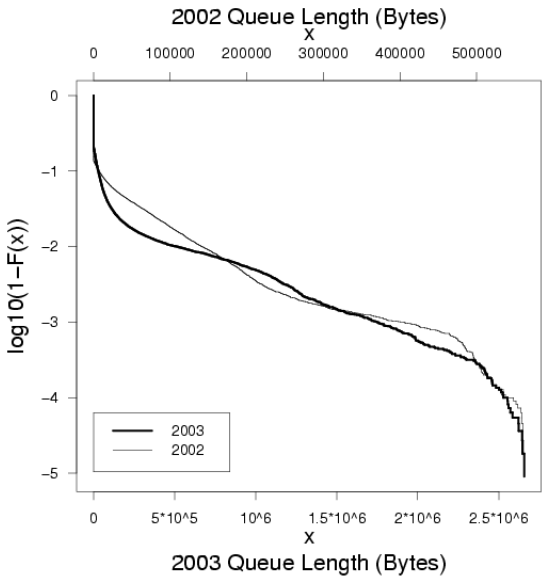


Figure 102: Empirical Queue Length CCDF for Saturday 2002 & Saturday 2003 at 9:30 PM, Aggregate Byte Counts, 80% Utilizations, individual horizontal axes. Although 2003 is 'scaled-up', both curves show similar shape.

Queuing comparison of the aggregate (ALL) with TCP-only (TCP) or UDP-only(UDP) subset time series is also revealing. When byte counts are used, at equal utilizations ALL and TCP have queue length CCDF curves with similar shape, allowing for differing scales. Figure 103 shows an overlay of the queue length CCDF using ALL and TCP traces from Saturday 2003 at 9:30 PM with simulated 80% utilizations. Buffer sizes are shown for ALL (bottom axis) and TCP (top axis) with scales in the ratio of 0.9:1. Although TCP is a subset of ALL, because utilization (not server rate) is fixed, TCP can cause more queuing, as reflected in the large TCP axis scale. The similarity in the shape of the CCDF curves is observed consistently across simulated utilizations between 50% and 95%. It is also observed consistently across all traces analyzed by trace-driven queuing methods: four traces discussed here and an additional trace each from 2002 and 2003 from different days of the week.

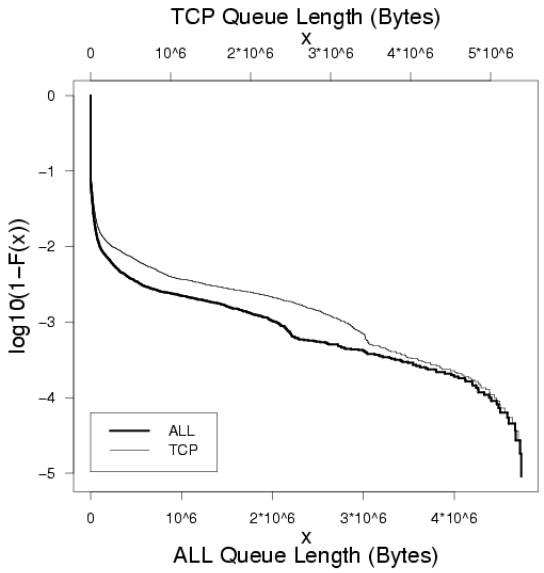


Figure 103: Empirical Queue Length CCDF for Saturday 2003 at 9:30 PM, Aggregate and TCP Byte Counts, 80% Utilizations. ALL and TCP curves using bytes show similar shape.

Figure 104 shows a similar comparison, but between ALL and UDP, instead of TCP. The scale of the horizontal axes is in the ratio of 6.7:1. The simulated utilizations are also 80%. Allowing for differing scales, again the shape of the curves is similar. This is also consistent across simulated utilizations between 50% and 95% where UDP makes up at least 10% of the total bytes.

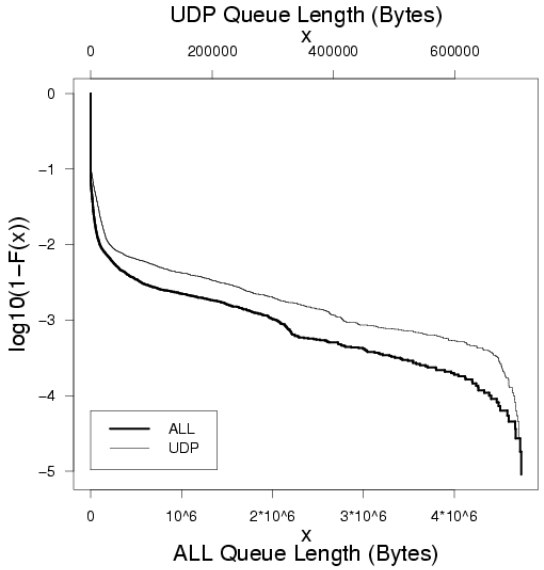


Figure 104: Empirical Queue Length CCDF for Saturday 2003 at 9:30 PM, Aggregate and UDP Byte Counts, 80% Utilizations, UDP 15% of ALL. ALL and UDP curves using bytes show similar shapes.

A comparison of queue length CCDF curves using packet counts (allowing for differing scales) reveals ALL and TCP give rise to similar shapes (not shown). This is the same as with byte counts. Interestingly, it is true even using Saturday 2003 when UDP makes up 31% of the packets, where it might be expected the high percentage of UDP packets would effect the shape. (In an additional dataset not shown with 30% UDP packets, the queue length CCDF curves for ALL and TCP are similar between 50% and 70% utilization. Above 70%, ALL shows similarity with UDP instead). However, unlike with byte counts, ALL and UDP queue length CCDFs are consistently different in shape, although the degree of that difference varies. Figure 105 shows a queue length CCDF overlay for ALL and UDP from Thursday 2002 at 3 PM with simulated 80% utilizations. In this trace UDP comprises 9% of the total packets. The shape of the queue length CCDF between ALL and UDP is quite different UDP. This is consistent across nine simulated utilizations between 50% and 95%.

In 2003, with a higher percentage of UDP packets, the differences in the shape of the CCDF curve are not as large as with 2002. Figure 106 shows a queue length CCDF overlay of ALL and UDP for Saturday 2003 at 9:30 PM with simulated 80% utilizations. In this case UDP packets are 31% of the total. The most noticeable difference is that the CCDF for UDP is displaced vertically two orders of magnitude (so, larger probabilities) from the CCDF for ALL. It is important to remember that the x-axes have different scales so this is not a direct quantitative comparison of two CCDF curves. Rather, since the utilizations are the same, relative to their respective mean rates, UDP packets appear more queuing intensive than the overall aggregate.

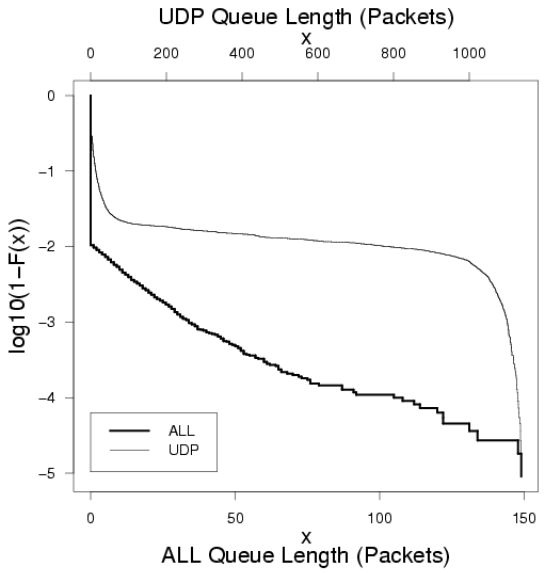


Figure 105: Empirical Queue Length CCDF for Thursday 2002 at 3 PM, Aggregate and UDP Packet Counts, 80% Utilization, UDP 9% of ALL. ALL and UDP packet counts give quite different shapes.

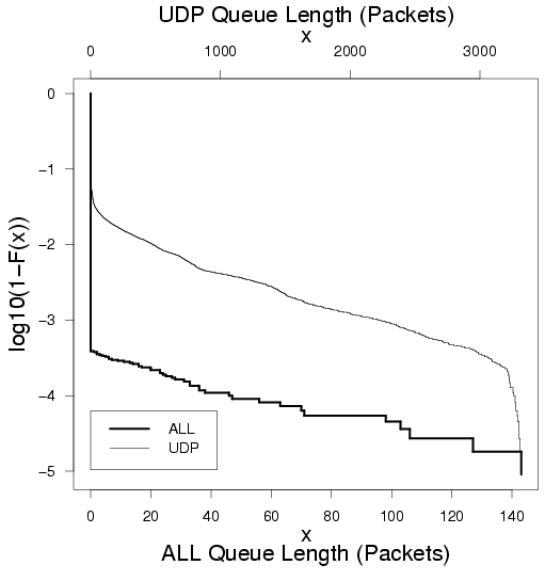


Figure 106: Empirical Queue Length CCDF for Saturday 2003 at 9:30 PM, Aggregate and UDP Packet Counts, 80% Utilization, UDP 31% of ALL. ALL and UDP packet counts give somewhat different shapes.

In summary, using byte counts and equal simulated utilizations, queue lengths are consistently larger with 2003 traces, although the mean and variance of those traces is also higher. Allowing for the increase in mean and variance, the shape of the queue length CCDF appears similar between 2002 and 2003 traces, and also between the aggregate and TCP-only traces.

The shape of the queue length CCDF is also similar between the aggregate and UDP-only traces when the percentage of UDP bytes in the aggregate is at least 10%. Since these are inbound traces, this similarity may be due, in part, to something experienced in the network. Using packet counts, the queue length CCDFs do show differences between 2002 and 2003. Qualitatively, the aggregate and TCP-only traces give similar queue length CCDF decay while UDP-only traces give decay different from that seen with the aggregate.

7. Summary of Results

We have presented the results from a large-scale analysis of two-hour TCP/IP packet traces acquired in 2002 and 2003 from a 1 Gbps Ethernet access link connecting the entire University of North Carolina at Chapel Hill to its Internet service provider. Our major findings are:

- Hurst parameter estimates for the packet-counts in 10 millisecond intervals decreased significantly between 2002 and 2003 while the Hurst parameter estimates for byte counts remained about the same.
- A single peer-to-peer file sharing application (*Blubster*) that suddenly became popular between 2002 and 2003 had a strong influence on the Hurst parameter for packet counts in the aggregate traffic and caused the differences in H between the two years. This happened because the application sends a large number of small packets with high frequency that, in turn, produces high variance at certain time scales in the wavelet analysis.
- With equal traffic intensities, queue lengths in simulations driven with the actual traces of byte counts were consistently larger with 2003 traces (although the mean and variance of the 2003 traces is also higher). Qualitatively, the decay of the queue length CCDF for bytes counts appears similar between 2002 and 2003 traces. Using packet counts, on the other hand, shows noticeable qualitative differences between the years
- Moderate to strong long-range dependence (Hurst parameter estimate in the range [0.65, 0.99]) existed in the vast majority of traces.
- In several of the 56 traces studied, we found evidence of highly bursty traffic in both packet and byte counts that could not be modeled as Fractional Gaussian Noise.
- The long-range dependence, as measured by the Hurst parameter estimate, was not influenced by the day-of-week or time-of-day. This implied that changes in the ***active*** user population (and, by implication, types of applications used) did not influence long-range dependence.

- The long-range dependence as measured by the Hurst parameter estimate was not influenced by the traffic load (link utilization) or the number of active TCP connections. In particular we found no indication of declining strength in long-range dependence as more active TCP connections are aggregated.

8. References

Appendix A. Summary Data for 2002 and 2003 Traces

Tables I and II provide complete summary data for the 2002 and 2003 traces, respectively. The tables are organized with one row for each two-hour trace that gives the day and time the trace started, information about the packets and byte counts in the trace, the mean link utilization during the trace interval, the estimated number of TCP connections active per minute, and the % of packets reported as being lost by the monitor. For each of packet and byte counts, we show the total number in the trace, the % TCP, UDP and other (e.g., ICMP, ARP), and the estimated Hurst parameter with confidence intervals for that time series aggregated in 10 millisecond intervals.

UNC 2002	Packets					Bytes					% Util.	Active TCP	% Loss
	Total (M)	%			Hurst Param. and 95% C.I.	Total (GB)	%			Hurst Param. and 95% C.I.			
		TCP	UDP	Rest			TCP	UDP	Rest				
Sun 0500	118.8	97.87	1.89	0.24	1.04 [0.97, 1.11]	30.0	97.69	2.22	0.09	1.10 [0.90, 1.30]	3.30	11,732	0.01
Sun 1000	131.2	97.81	1.92	0.27	1.00 [0.93, 1.07]	30.5	97.02	2.86	0.12	0.76 [0.75, 0.77]	3.36	16,074	0.01
Sun 1500	185.6	95.05	4.70	0.25	0.95 [0.92, 0.98]	59.5	95.16	4.76	0.08	0.97 [0.90, 1.04]	6.53	26,890	0.06
Sun 2130	191.5	93.89	5.88	0.23	0.96 [0.93, 0.99]	60.3	94.30	5.63	0.07	0.97 [0.94, 1.00]	6.61	28,975	0.09
Mon 0500	118.5	97.82	1.86	0.32	1.12 [1.05, 1.19]	24.0	96.66	3.18	0.16	1.12 [1.00, 1.23]	2.64	11,506	0.01
Mon 1000	221.1	92.56	7.18	0.26	0.95 [0.92, 0.98]	76.9	88.03	11.86	0.11	0.98 [0.95, 1.01]	8.42	37,722	0.13
Mon 1500	243.3	93.12	6.58	0.30	0.90 [0.86, 0.95]	93.8	90.90	8.98	0.12	0.91 [0.89, 0.93]	10.26	45,126	0.14
Mon 2130	206.5	93.26	6.53	0.21	0.98 [0.93, 1.03]	67.6	94.50	5.43	0.07	0.95 [0.93, 0.97]	7.40	34,141	0.11
Tue 0500	111.9	97.77	1.94	0.29	1.06 [1.01, 1.10]	24.7	97.07	2.80	0.13	0.90 [0.83, 0.97]	2.73	11,534	0.01
Tue 1000	209.6	92.17	7.54	0.29	0.87 [0.85, 0.89]	80.8	89.04	10.87	0.09	0.87 [0.85, 0.88]	8.85	38,518	0.12
Tue 1500	253.6	91.79	7.86	0.35	0.92 [0.89, 0.95]	95.9	89.62	10.26	0.12	0.92 [0.90, 0.94]	10.48	44,818	0.16
Tue 2130	194.8	93.74	6.04	0.22	1.03 [0.98, 1.07]	60.0	94.39	5.53	0.08	0.90 [0.87, 0.93]	6.58	32,231	0.09
Wed 0500	120.9	98.11	1.63	0.26	0.99 [0.94, 1.04]	23.9	97.33	2.55	0.12	1.09 [0.97, 1.20]	2.63	11,172	0.00
Wed 1000	224.7	91.94	7.80	0.26	0.84 [0.44, 1.24]	86.9	87.50	12.41	0.09	0.65 [0.26, 1.05]	9.50	38,173	0.14
Wed 1500	245.0	91.34	8.38	0.28	1.07 [1.00, 1.14]	91.8	87.48	12.41	0.11	1.23 [1.11, 1.34]	10.04	43,098	N/A
Wed 2130	212.5	93.69	6.08	0.23	0.98 [0.94, 1.03]	76.1	93.21	6.72	0.07	0.96 [0.93, 0.99]	8.34	32,390	0.12
Thu 0500	113.2	97.18	2.50	0.32	1.10 [1.03, 1.17]	27.3	97.11	2.77	0.12	0.91 [0.84, 0.98]	3.01	11,002	0.01
Thu 1000	209.6	90.85	8.83	0.32	0.90 [0.88, 0.92]	86.3	87.44	12.45	0.11	1.00 [0.95, 1.04]	9.46	38,555	0.13
Thu 1500	221.2	91.13	8.57	0.30	0.92 [0.87, 0.96]	83.6	88.39	11.51	0.10	0.86 [0.85, 0.87]	9.15	43,020	0.11
Thu 2130	178.0	91.49	8.20	0.31	1.00 [0.95, 1.04]	50.8	93.03	6.86	0.11	0.96 [0.91, 1.00]	5.58	29,763	0.06
Fri 0500	110.2	94.34	5.29	0.37	0.96 [0.91, 1.00]	23.5	94.06	5.79	0.15	0.86 [0.79, 0.93]	2.59	11,843	0.00
Fri 1000	208.9	91.49	8.24	0.27	0.95 [0.92, 0.98]	76.6	86.70	13.19	0.11	0.91 [0.89, 0.93]	8.40	38,026	0.12
Fri 1500	214.7	90.95	8.63	0.42	1.06 [1.02, 1.11]	79.8	88.58	11.17	0.25	0.94 [0.92, 0.96]	8.74	39,856	0.10
Fri 2130	161.3	92.89	6.77	0.34	0.97 [0.92, 1.01]	44.5	94.34	5.54	0.12	0.86 [0.84, 0.88]	4.89	21,482	0.04
Sat 0500	108.1	94.37	5.19	0.44	1.06 [0.94, 1.18]	25.0	96.39	3.45	0.16	0.88 [0.81, 0.95]	2.75	8,246	0.00
Sat 1000	139.9	94.65	5.02	0.33	0.91 [0.88, 0.94]	38.4	95.79	3.97	0.24	0.79 [0.78, 0.79]	4.22	21,409	0.02
Sat 1500	168.8	93.14	6.50	0.36	0.97 [0.92, 1.01]	55.7	92.61	7.25	0.14	0.93 [0.88, 0.97]	6.12	22,032	0.04
Sat 2130	149.9	92.81	6.89	0.30	0.98 [0.93, 1.02]	46.9	94.63	5.28	0.09	0.84 [0.83, 0.85]	5.15	18,483	0.03

Table I. Trace Summary Data for 2002

UNC 2003	Packets					Bytes					% Util.	Active TCP	% Loss
	Total (M)	%			Hurst Param. and 95% C.I.	Total (GB)	%			Hurst Param. and 95% C.I.			
		TCP	UDP	Rest			TCP	UDP	Rest				
Sun 0500	233	74.73	25.00	0.27	0.72 [0.65, 0.79]	48.2	84.09	15.80	0.11	0.90 [0.88, 0.92]	5.35	12,840	0.00
Sun 1000	268	69.11	30.56	0.33	0.90 [0.78, 1.01]	50.8	79.55	20.28	0.17	0.90 [0.89, 0.92]	5.65	20,241	0.00
Sun 1500	342	67.79	31.92	0.29	0.71 [0.69, 0.73]	95.7	84.81	15.09	0.10	0.95 [0.92, 0.98]	10.63	32,466	0.00
Sun 2130	361	71.29	28.42	0.29	0.71 [0.68, 0.73]	117.8	88.61	11.30	0.09	0.90 [0.89, 0.92]	13.09	37,793	0.00
Mon 0500	233	77.93	21.78	0.29	0.93 [0.81, 1.05]	51.0	86.34	13.51	0.15	0.87 [0.84, 0.90]	5.67	14,795	0.00
Mon 1000	394	76.04	23.69	0.27	1.31 [1.13, 1.49]	128.8	85.02	14.87	0.11	0.97 [0.96, 0.98]	14.31	48,252	0.00
Mon 1500	449	73.48	26.34	0.18	0.81 [0.76, 0.85]	159.5	87.01	12.94	0.05	0.93 [0.92, 0.95]	17.72	56,839	0.00
Mon 2130	368	74.00	25.73	0.27	0.80 [0.75, 0.84]	124.0	89.21	10.72	0.07	1.00 [0.96, 1.05]	13.78	39,127	0.00
Tue 0500	254	80.65	18.94	0.41	0.76 [0.69, 0.83]	52.1	86.88	12.96	0.16	0.80 [0.73, 0.87]	5.79	16,871	0.00
Tue 1000	406	77.55	22.07	0.38	0.69 [0.66, 0.72]	133.8	85.73	14.17	0.10	0.90 [0.89, 0.92]	14.86	49,494	0.00
Tue 1500	455	74.91	24.77	0.32	0.72 [0.69, 0.75]	153.2	85.49	14.41	0.10	0.92 [0.90, 0.94]	17.02	56,913	0.00
Tue 2130	395	72.72	26.82	0.46	0.78 [0.73, 0.82]	123.1	88.53	11.34	0.13	0.90 [0.88, 0.91]	13.68	43,276	0.00
Wed 0500	250	78.94	20.44	0.62	0.65 [0.62, 0.68]	53.6	87.18	12.60	0.22	0.85 [0.81, 0.90]	5.96	16,671	0.00
Wed 1000	401	77.59	22.05	0.36	0.71 [0.69, 0.73]	127.9	84.64	15.26	0.10	0.93 [0.92, 0.94]	14.21	49,732	0.00
Wed 1500	456	74.84	24.78	0.38	0.88 [0.80, 0.95]	164.3	86.49	13.40	0.11	0.92 [0.91, 0.93]	18.26	59,064	0.00
Wed 2130	376	71.33	28.19	0.48	0.72 [0.69, 0.75]	118.2	88.16	11.64	0.20	0.89 [0.88, 0.90]	13.13	42,549	0.00
Thu 0500	250	79.67	19.82	0.51	0.92 [0.85, 0.99]	54.6	87.87	11.87	0.26	0.90 [0.87, 0.93]	6.06	15,697	0.00
Thu 1000	389	76.77	22.88	0.35	0.79 [0.74, 0.83]	130.4	84.93	14.96	0.11	0.93 [0.92, 0.95]	14.49	45,785	0.00
Thu 1500	456	74.39	25.25	0.36	0.77 [0.75, 0.79]	159.6	86.21	13.64	0.15	0.96 [0.95, 0.98]	17.73	55,104	0.00
Thu 2130	379	70.65	28.99	0.36	0.75 [0.73, 0.77]	118.8	87.00	12.86	0.14	0.94 [0.93, 0.95]	13.20	37,398	0.00
Fri 0500	257	76.80	22.71	0.49	0.82 [0.74, 0.89]	52.8	85.02	14.71	0.27	0.94 [0.92, 0.96]	5.87	16,674	0.00
Fri 1000	412	75.33	24.33	0.34	0.73 [0.71, 0.75]	137.2	84.88	14.98	0.14	0.94 [0.93, 0.95]	15.25	48,933	0.00
Fri 1500	443	73.16	26.54	0.30	0.79 [0.78, 0.79]	153.3	86.08	13.79	0.13	0.87 [0.46, 1.27]	17.04	48,869	0.00
Fri 2130	335	72.09	27.56	0.35	0.71 [0.69, 0.73]	98.1	87.73	12.13	0.14	1.01 [0.96, 1.05]	10.90	25,540	0.00
Sat 0500	241	74.47	25.06	0.47	0.78 [0.71, 0.85]	50.9	84.09	15.73	0.18	0.93 [0.88, 0.98]	5.66	12,824	0.00
Sat 1000	290	69.69	29.92	0.39	1.18 [0.98, 1.39]	62.1	81.86	17.93	0.21	0.87 [0.86, 0.88]	6.90	23,087	0.00
Sat 1500	340	69.61	30.02	0.37	0.83 [0.76, 0.91]	95.1	85.66	14.23	0.11	0.87 [0.84, 0.90]	10.56	27,332	0.00
Sat 2130	316	69.34	30.30	0.36	0.76 [0.71, 0.81]	78.4	84.87	15.01	0.12	0.89 [0.88, 0.90]	8.71	24,003	0.00

Table II. Trace Summary Data for 2003