Network Traffic Behaviour in Switched Ethernet Systems

Tony Field, Uli Harder & Peter Harrison[1]

Department of Computing
Imperial College London
Huxley Building
180 Queen's Gate
London SW7 2AZ
England

Abstract

Measurements on a high-performance switched Ethernet system are presented that reveal new insights into the statistical nature of file server and web server traffic. Both file sizes and data requested from the web server are shown to match well a truncated Cauchy distribution. This is a distribution with heavy tails similar in nature to the commonly used Pareto distribution but with a much better fit over smaller file/request sizes. We observe self similar characteristics in the traffic at both servers and also at a CPU server elsewhere on the network. Traffic from this server is predominantly targeted at the file and web servers, suggesting that self-similar properties at one point on a network are being propagated to other points. A simple simulation model of a web server network link is presented where the service (packet transmission) demands have the same Cauchy distribution as the observed request sizes. The power spectrum of the departure process is shown to follow a power law and is shown to match extremely well that of the observed traffic. The simulation is also used to investigate the link between the power laws in the request size distribution and the network traffic by using Lévy distributions for the request sizes. This supports the suggested link between file/request size distribution and self-similarity. The resulting implication that self similarity and heavy tails are primarily due to server-nodes, rather than being inherent in offered traffic, leads to the possibility of using conventional queueing network models of performance.

[1]e-mail: ajf@doc.ic.ac.uk, uh@doc.ic.ac.uk, pgh@doc.ic.ac.uk

1. Introduction

In preparation for the building of performance models for network performance we have been measuring and analysing network traffic at various parts of an academic departmental network, specifically the Computing Department of Imperial College, London. The motivation is to provide a better understanding of networking issues and to provide a body of data that can be used to validate future models of networks and network traffic.

Network traffic has been measured by many researchers and been analysed in many different ways ever since the seminal papers of Leland et al. and Eramili et al. [1, 2]. There are many papers debating the reason for the apparent self-similarity in modern communications traffic. The explanations put forward range from ON/OFF models of heavy-tail distributions [1, 2], through the file size distribution in file systems and web servers [3, 4] to user behaviour, higher level network protocols, back-off algorithms in the Ethernet [5], buffers in routers and the TCP congestion avoidance algorithms [6, 7, 8, 9, 10].

Like many earlier papers the main motivation here is to obtain a better understanding of the characteristics of network traffic and to explain the sources of self-similarity. In addition, however, we want to build accurate models of network traffic, together with the web-servers with which it interacts. This requires that such models should be able to reconstruct the type of self-similar patterns that are observed in real networks. To this end, we develop a simple simulation reference model, carefully parameterised using insights from various observations of real traffic, which is shown to recreate very accurately the power laws observed.

A key property that the simulation reproduces is the distribution of both web server and file server response sizes, in terms of the number of Ethernet packets required to transmit them. It is well known that these often exhibit power laws [1, 2] but previous work has tended to match them to Pareto distributions. Our analysis suggests that the data may be better represented by a (truncated) Cauchy distribution and the success of the simulation model in reproducing observed traffic patterns bears this out. The paper thus adds to the growing body of evidence linking heavy-tailed distributions with self-similarity, here backed up with a supporting model.

Curiously the Cauchy distribution appears elsewhere in our analysis, specifically when we consider the *changes* in the packet transmission rates over time. This type of measurement is commonplace in the analysis of stock prices, for example [11, 12], but not, to date, for network traffic. Interestingly, stock price changes have also been found to be well approximated by Lévy distributions (of which the Cauchy distribution is one particular example). There is also a model for the distribution of the sizes of e-mails, reported in [13], that uses a Cauchy distribution.

The measurements we have performed are cheap and easy to reproduce at other sites. In particular, in our discussions we promote the use of `/proc/net/dev` files when analysing over very long time periods. This is both economical on resources and can be done without system privileges. A comparison of `/proc/net/dev` and `tcpdump` measurements is presented later in the paper.

The paper makes the following contributions:

- We propose `/proc/net/dev` as an effective and efficient alternative to `tcpdump` for monitoring network traffic. This is supported by comparing observations made using both methods.
- We show that measured traffic taken at three locations on a state-of-the-art switched Ethernet fits closely various (truncated) Cauchy distributions, which are heavy-tailed. The data captured has been made publicly available via the world-wide web [14].
- We demonstrate that changes in packet rates (first difference of the packet-density time series) also conforms to a Cauchy distribution, revealing a possible link with the observations of the server traffic.
- We show that sample traces from the internet traffic archive display similar characteristics to our network, particular in the inter-event time distribution, packet rate change distribution and power spectra.
- We present results from a simulation model of a network link assuming Poisson arrivals and Cauchy-distributed request sizes; we explicitly model the packetisation process and observe in the output process a power law that corresponds closely to that of observed network traffic.
- Based upon the various observations of real and simulated network traffic we propose a possible internet modelling strategy based on processor sharing and conventional (product-form) queueing networks.

The rest of the paper is organised as follows: Section 2 discusses the two main mechanisms for real-time data capture within the Linux operating system. Section 3 describes the architecture of the network in question and the type of monitoring that has been performed. Section 4 describes the analysis techniques that have been applied to the captured traces and Section 5 presents the results of applying that analysis. The simulation model is detailed in Section 6 and the potential role of queueing models in the analysis of internet components is summarised in Section 7. The conclusions and future work are given in Section 8. The results of our analysis of the Internet Traffic Archive are presented separately in the Appendix.

2. Data Capture

This paper refers to data measured in two different ways. One uses the `tcpdump` program that monitors traffic at the packet level. The other method reads the contents of the file `/proc/net/dev` periodically.

2.1. `tcpdump`.

The `tcpdump` [15] utility attaches itself to the network socket in the Linux kernel and makes a copy of each network packet that arrives at the network interface. To be more precise it allows all Ethernet packets arriving at the kernel to be logged. We have used the program to obtain for every frame that is transmitted:

- A timestamp indicating when the packet reaches the kernel;
- Source and destination IP addresses and port numbers;
- The size of the frame (only the user data is reported, the headers for various protocol layers have to be added on to recover the actual size of the frame);
- The traffic type (`tcp`, `udp`, `icmp`, etc).

The timestamp is the time the kernel first "sees" the packets rather than the time it seen by the Ethernet card. The output of `tcpdump` can be written to a file for later analysis. This allows monitoring over long periods of time.

2.2. **`/proc/net/dev`.** The Linux operating system keeps track of the number of packets and bytes sent and received in a virtual file called `/proc/net/dev` using counters. The counters get reset at machine reboot; the values are modulo 2^{32}. We have used a `PERL` program to query this file at regular intervals of 1 second and record the value of the counters and a time stamp. Occasionally this process fails to record the counter within 1 second. In the processing phase we then linearly interpolate the missing values and get a time series of the counter values for every second. The data is similar to the aggregated data of the previous section although the time resolution is by no means as good. This method has three advantages over `tcpdump` : it usually requires no special operating system privileges, it can be run for a much longer time as only summary data is collected, and it incurs lower intrusion overheads. The tradeoff between resolution and efficiency, both space and time, is addressed later in this paper.

All raw data discussed in this paper has been anonymised and is freely available for inspection and use in other research—see [14] for the url.

3. Network Architecture and Monitoring

The monitored system is a departmental switched Ethernet. We focus attention on three components of this network for the purposes of this paper, as illustrated in Figure 1: The router, which connects the network to the outside world, an arbitrarily chosen CPU server (named MOA), and the departmental web server which services both internal and external web page requests.

3.1. **Router.** The department is connected to the Internet via a Black Diamond router from Extreme Networks [16]. The Black Diamond is used as both a top-level *switch* for the internal Ethernet and as a router for all outgoing traffic. Those two functions are completely separate. All outgoing packets from machines in the department are duplicated at media access control (MAC) level to the second

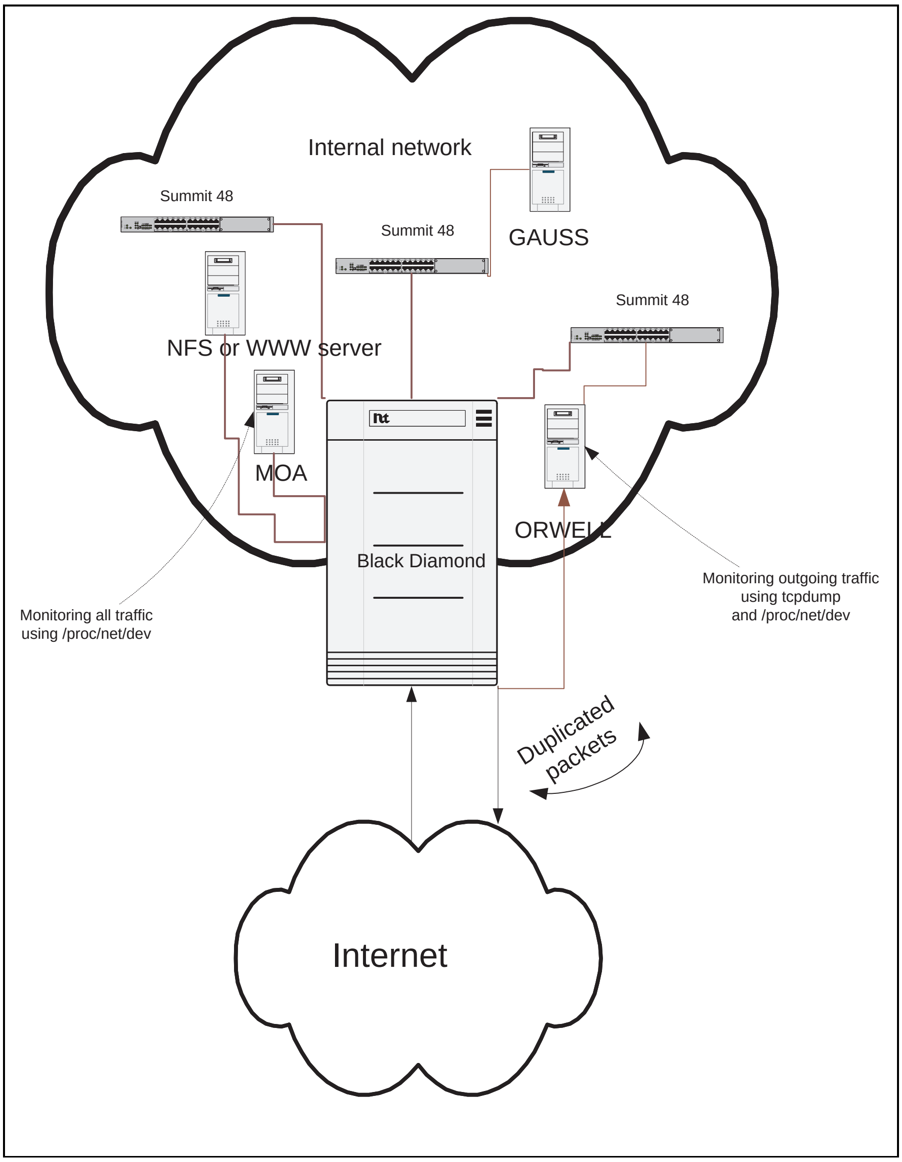

FIGURE 1. Overview of the departmental network. Note that the activity of the core router due to the internal network is not recorded.The Black Diamond is a router for external traffic and a top-level switch for the internal traffic. Both parts are implemented by different pieces of hardware and are independent of each other.

network interface card (NIC) of a dedicated monitoring machine (ORWELL[2]).

[2]The Black Diamond provides a link of 1 Gbit/s. The monitoring machine runs Linux 2.2.x, has 256 Mbytes RAM and 4 SCSI disks used in rotation for the log files. ORWELL can monitor at a rate of 100Mbit/s

Log files are transferred to a different machine (GAUSS) for analysis. We have used `PERL` scripts to turn the `tcpdump` output into text files. Also, we made the data anonymous by replacing the IP numbers with an indication whether the number was internal or external. This paper focuses in part on data gathered in this way over a two hour interval on 22 March 2002 between 12pm and 2pm.

When interpreting the results, the reader should keep in mind what time scales are involved. A 100Mbits/sec network manages to transmit about 13 Bytes per microsecond. Following is a list of important Ethernet data sizes and their transmission times for a 100Mbps network:

- Inter-frame gap: 12 Bytes or 0.92 μseconds
- Smallest frame: 64 Bytes or 4.9 μseconds
- Largest frame: 1500 Bytes or 115 μseconds.

The inter-frame time gap is the minimum gap between two consecutive Ethernet frames. The inter-event time histograms show peaks at about 6 and 120 μseconds, as these are the most likely inter event times on a busy network.

We also measured the outgoing traffic as seen by the monitoring machine using the `/proc/net/dev` file starting on 6 March 2002 at 18:48 and finishing on 25 March 2002 at 16:29. This observation period overlaps the two-hour period captured above using `tcpdump` this enables a comparison of both measurement methods.

3.2. **CPU server.** On one of the departmental CPU servers, MOA, we ran the `/proc/net/dev` monitor for twelve days at a measurement interval of one second. The measurement started at 1 February 2002 at 16:24:47. A majority of the traffic generated by the CPU servers are targeted at the internal file and web servers.

3.3. **Web Server.** Web server traffic was filtered so as to collect only outgoing data generated by external requests. The web server itself provides information on the individual web requests, in particular bytes shipped per request. This data is sent to the outside world via the router (Black Diamond) and individual (www) packets are monitored as they pass through the same router. We thus obtain the measured request size distribution and a time series representing the instants the corresponding stream of Ethernet packets pass through the router. Again we use the interval on 22 March 2002 between 12pm and 2pm.

4. Analysis Methods

We concentrate here on the point processes formed by packet departure events happening at times t_i, $i \in I \subseteq I\!N$. The event that occurs at time t_i is called E_{t_i}. We assume that there are $n \in I\!N$ events (or observations of events), the first happening at t_1 and the last one at t_n. The observation period may begin before the first event and end after the last, so we define it to be $T = [t_0, t_{n+1}] \subset I\!R$ with $t_0 \leq t_1 \leq \ldots \leq t_n \leq t_{n+1}$, for arbitrary t_0, t_{n+1} bounding the set of event-instants.

The inter-event times, Δt_i, $1 \leq i \leq n-1$, are defined as

$$\Delta t_i = t_{i+1} - t_i$$

For a finite observation of a point process we can easily generate a histogram that, when correctly scaled, approximates the probability density function (pdf) of inter-event times.

4.1. **Power Laws.** The probability density function $p(x)$ is said to follow a power law if

$$p(x) \propto x^{\gamma}$$

as $x \to \infty$, for $\gamma < -1$. When investigating the existence or otherwise of a power law we use exponentially growing bin sizes for the histograms. Apart from the histogram, we also compute the mean and variance of the inter-event times which are useful for distribution fitting.

4.2. **Aggregation.** Starting from a point process one can investigate the behaviour of the corresponding *aggregated* time series. The observation period T is divided into N contiguous intervals of size $T_N = T/N$. In each of these intervals, we count the number of events or, if appropriate, we aggregate the properties of the events. So the time series consists of N values

$$a_i = \Big| \{E_t | t_0 + iT_N \leq t < t_0 + (i+1)T_N\} \Big|.$$

for $i = 1, 2, \ldots, N$. Sometimes it is preferred to use the quantity $A_i = a_i/T_N$.

4.3. **Self Similarity.** An aggregated time series can be subjected to many analyses, one of which is its *scaling behaviour*. This is also known as testing self-similarity. Two of the first investigations of the statistical nature of network traffic were [1, 2]. The authors found evidence that the observed traffic was distinctly non-Poisson and thus not amenable to conventional traffic modelling techniques. They instead used methods that had been developed earlier by Hurst who was investigating the "ideal" size of reservoirs (a good summary of Hurst's work is given in [17, 18]). In particular, Hurst introduced the rescaled range statistic R/S which gives an idea of the self-similarity or long-range dependence of a time series. Many other statistics, which are all proved or conjectured to be related to the Hurst parameter, have since been introduced. A good review of the estimators and their relationships can be found in [19, 20, 21]. The next section is based on the material found in those papers. In this investigation, we will use the *power spectrum* to analyse the correlation of the monitored data and inter-event histograms to analyse the inter-arrival time distribution.

4.4. **Power Spectrum.** For a time series $X(t)$ with zero mean the auto-correlation function at lag τ is defined as

$$C(\tau) = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} dt X(t+\tau)X(t).$$

The power spectrum (density) is defined as the Fourier transform of the auto correlation function

$$S(f) = \frac{1}{2\pi} \int_{-\infty}^{\infty} d\tau C(\tau) e^{-2\pi i f \tau}.$$

In fact, due to the Wiener-Khintchine theorem [22, 23] it is easier to work with the Fourier transform of the auto-correlation function. By the Wiener-Khintchine theorem, under certain assumptions, the power spectrum is the same as the absolute square of the Fourier transform of the time series signal [3] which can be efficiently calculated using for instance the Fast Fourier Transformation (FFT)

$$S(f) = \lim_{T\to\infty} \frac{1}{4\pi T} \Big| \int_{-T}^{T} dt X(t) e^{-i2\pi f t} \Big|^2.$$

The power spectrum, or spectral density, shows the distribution of the signal strength for different frequencies. Note that this expression can be discretised for discrete time series. Since the point process tends to be rather sparse it is better to turn the time series into an aggregate time series of counts. In our investigation we have used 10ms bins for the aggregation in line with previous research [1, 2]. Again one is looking for power laws where the power spectrum $S(f)$ behaves like $S(f) \propto 1/f^\alpha$, where f is the frequency. The exponent α turns out to be 0 for white noise and 2 for a Brownian motion.

Because the auto correlation function is the inverse Fourier transform of the power spectrum, $1/f^\alpha$ noise corresponds to the following correlation function

$$C(\tau) \propto |\tau|^{\alpha-1} \quad \text{for } 0 < \alpha < 1$$

it also follows that an exponent α close to but smaller than 1 corresponds to long term correlations.

By itself the power spectrum (also known as periodogram) is not a consistent estimator for the auto-correlation function. This means that using more data points does not lead to a more accurate estimate. The only change is that lower frequencies can be probed. However, by using filters or windows the power spectrum can be turned into a consistent estimator, for details see for instance [24]. The power spectra were computed using standard methods published in the *Numerical Recipes in C* with overlapping windows [25]. In addition we averaged the values of the power spectra in logarithmic intervals to reduce the noise that is usually seen for higher frequencies. When interpreting a power spectrum it is useful to keep in mind that the leftmost (highest) frequency is determined by

[3] We should note that this is not a very rigorous way of defining the power spectrum, as the time series has to fulfil certain criteria for the integral to be well defined, for instance.

sampling rate and corresponds to the shortest time, while the rightmost (lowest) frequency is determined by the sampling length and corresponds to the longest time.

Other methods to investigate the correlation of the data are the rescaled range statistic, the log-variance plot, [17, 18, 26] detrended fluctuation analysis, wavelet transformations [27],the Fano factor and the Allan factor [19, 20, 21, 27].

4.5. **Scaling Behaviour and Power Laws.** Suppose a time series $X(t)$ exhibits the scaling law $X(t\alpha) = g(\alpha)X(t)$ for some function $g(\alpha)$. Then it is known that $X(t) = bg(t)$ and $g(\alpha) = \alpha^c$, for real constants b, c, is the only non-trivial solution to the above functional equation. This property is related to what is known as self-similarity. The exponent c is related to the Hurst parameter and, in fact, there is a firm belief – even a mathematical proof in some cases – that all exponents retrieved from power laws discovered in statistics of self-similar or fractal time series are related. So, the choice of the method used to analyse a time series is, up to a point, a matter of taste and ease of use.

If one assumes that a time series is generated by a stochastic process, such as Brownian motion, for example, one can investigate the distribution of the *changes* of the time series values $a_i,\ i = 1, 2, ..., N$ from one time step to the next, i.e. the differenced time series $\{\Delta a_i = a_{i+1} - a_i \mid 1 \leq i \leq N - 1\}$. Alternatively one can work out the distribution of the $a_i, 1 \leq i \leq N - 1$ as they are the changes of the underlying counting process. Depending on the statistical nature, one also expects a different kind of scaling law for these distributions with respect to the length of the aggregation interval T_N.

5. Measurements

5.1. **Outgoing Router Traffic.** We begin with a routine analysis of the outgoing traffic at the Black Diamond router. We expect no surprises here and find, as has been observed numerously elsewhere, that there is strong evidence of a power law in the underlying time series for packet departure times over the measurement interval, as measured using `tcpdump`. In particular it is clear that the inter-event times are distinctly non-exponential as evidenced in figure 2. Notice the two humps corresponding to the shortest and longest packet sizes (4.9μseconds and 115μseconds respectively). The power spectrum of this outgoing traffic shows a power law between 1Hz and 10^{-3}Hz, as indicated by the straight line in figure 3. This corresponds to a time range of 1 second to 16 minutes. Note that for frequencies above $10^{1.5}$ Hz or below 30mseconds the power spectrum is approaching that of white noise (straight line with gradient 0).

5.1.1. *Comparison of `tcpdump` and `/proc/net/dev`.* The data gathered provides the opportunity to compare data obtained from measurements with `tcpdump` with that in the `/proc/net/dev` file. Whilst gathering the `tcpdump` data, we also used `/proc/net/dev` to measure network activity over the same period as

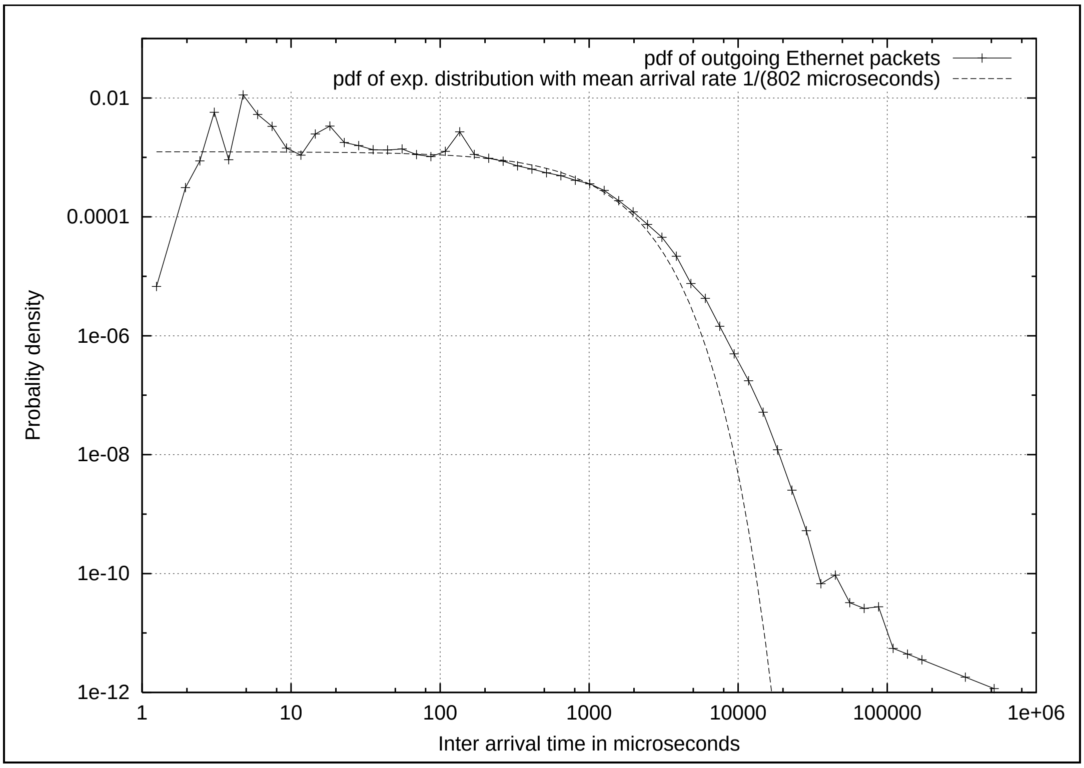

FIGURE 2. The inter event time histogram for the outgoing network traffic on a log-log scale. For comparison, we have drawn an exponential pdf with the same mean as that measured. The measurements were taken on the router using `tcpdump`.

reported in Figure 4 and 5. Aggregated over two minute intervals, the packet rates are almost identical for both methods. The power spectrum obtained by `/proc/net/dev` is rather jerkier due to the smaller number of points. Still, we can observe a power law behaviour similar to the one observed with `tcpdump`. The results show that data obtained by `/proc/net/dev` provides an excellent alternative to `tcpdump` particularly when monitoring over very long time periods. `/proc/net/dev` is used exclusively in the analysis of the CPU server which was monitored continuously over a twelve-day period.

Looking at the changes in the packet rates, Δ_i, we find that they are not Normally distributed which we would expect if the time series were a Wiener Process (Brownian Motion)—see figure 5. The distributions instead show a power law in a double-logarithmic plot.

5.2. **CPU Server.** The power spectrum of a twelve-day observation period clearly shows peaks corresponding to the daily cycles ($86,000 \approx 10^{4.93}$seconds or $10^{-4.93}$Hz), and also has peaks between one and two hours ($7,200 \approx 10^{3.85}$seconds or $10^{-3.85}$Hz

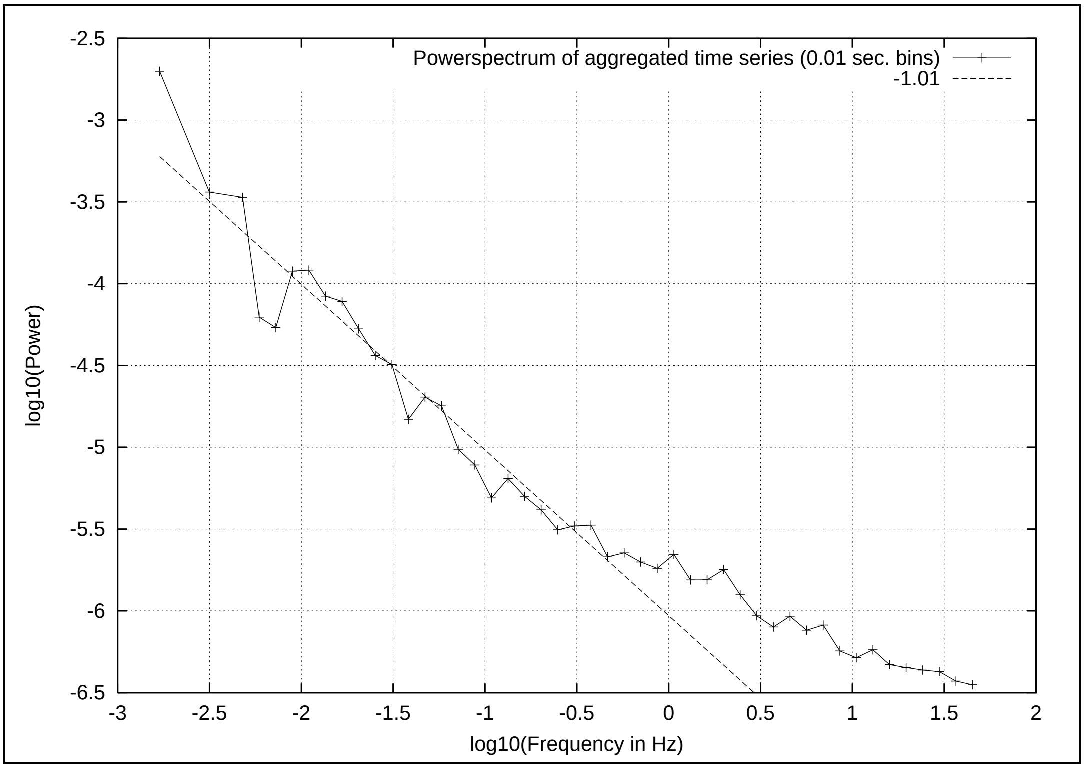

FIGURE 3. A plot of the power spectrum of the outgoing network traffic time series measured in packets per second, aggregated over 0.01 second intervals. These measurements were taken on the router using `tcpdump`.

and $3,600 \approx 10^{3.55}$seconds or $10^{-3.55}$Hz), an interval that will correspond to typical session lengths of student users during term time, see figure 6. The main interest lies in the plots which show the *changes* in the observed packet rate. We see a distribution that is distinctly leptokurtic and certainly non-Gaussian, see figure 7, 8 and 9.

In fact figure 8 shows that the asymptotic power law has a gradient of -2. And figure 9 shows that the distributions at different aggregation levels fall into one *master* curve when plotted on top of each other, as they should do for a Cauchy distribution[4]. These plots were inspired by related work in the analysis of stock prices where changes in prices have been shown to follow similar patterns, see for example [11, 12].

In order to trace possible causes of this behaviour we next study the characteristics of the file sizes stored at the file server and the request sizes generated

[4]See p. 92 in Voit's book, p. 71 in Mantegna's book and Honerkamp [11, 12, 24] for details of the scaling behaviour of Lévy distributions.

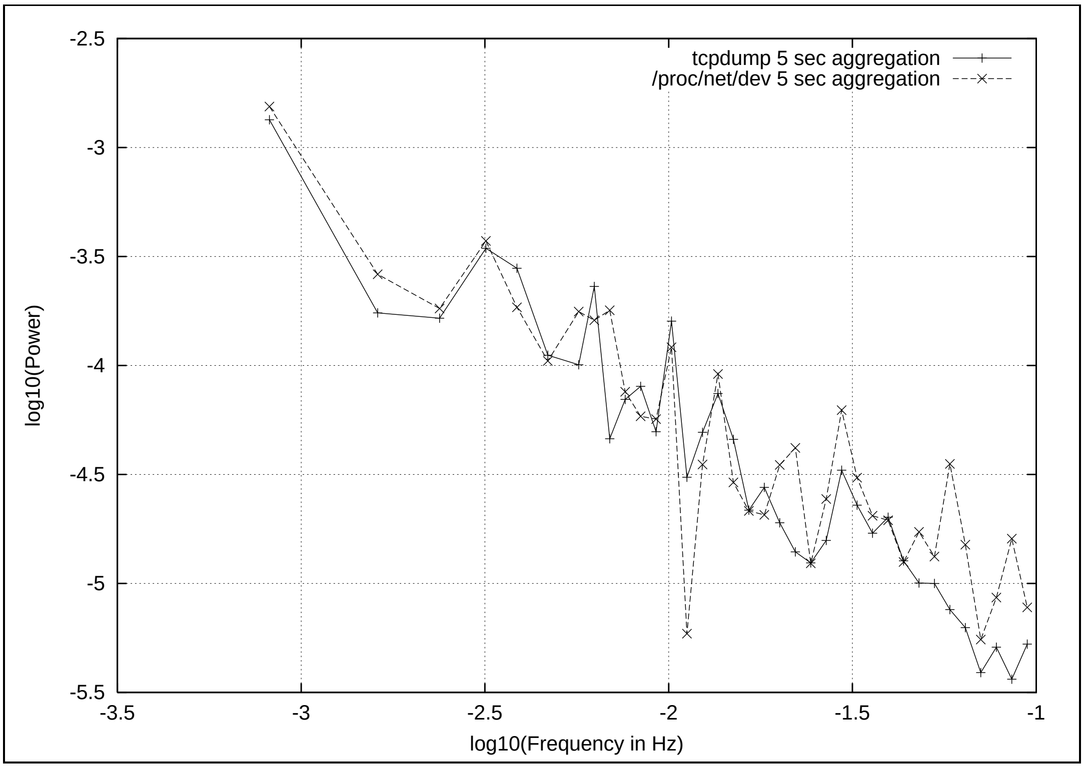

FIGURE 4. The plot shows power-spectra from data collected over the same period and aggregated to 5 seconds in both cases. Though not identical the plots are similar and show definite signs of a power law. These measurements were taken on the router using `tcpdump` and `/proc/net/dev`.

by the web server, as requests to these two servers constitute the main source of network traffic to/from the CPU server.

5.3. **File Size and Webserver Request Distribution.** Many authors have repeatedly shown that these distributions follow Zipf's law to a good approximation, which says that

$$P(\text{file or request size} > x) \approx \frac{1}{x}.$$

One pdf that can exhibit this behaviour, and which has been used elsewhere, is the Pareto distribution

$$p(x) = \alpha k^{\alpha} x^{-\alpha-1},$$

where $\alpha, k > 0$ and $x \geq k$. If $\alpha = 1$ the Pareto distribution shows the behaviour of the Zipf law for large x. In a double logarithmic plot, this distribution is a straight

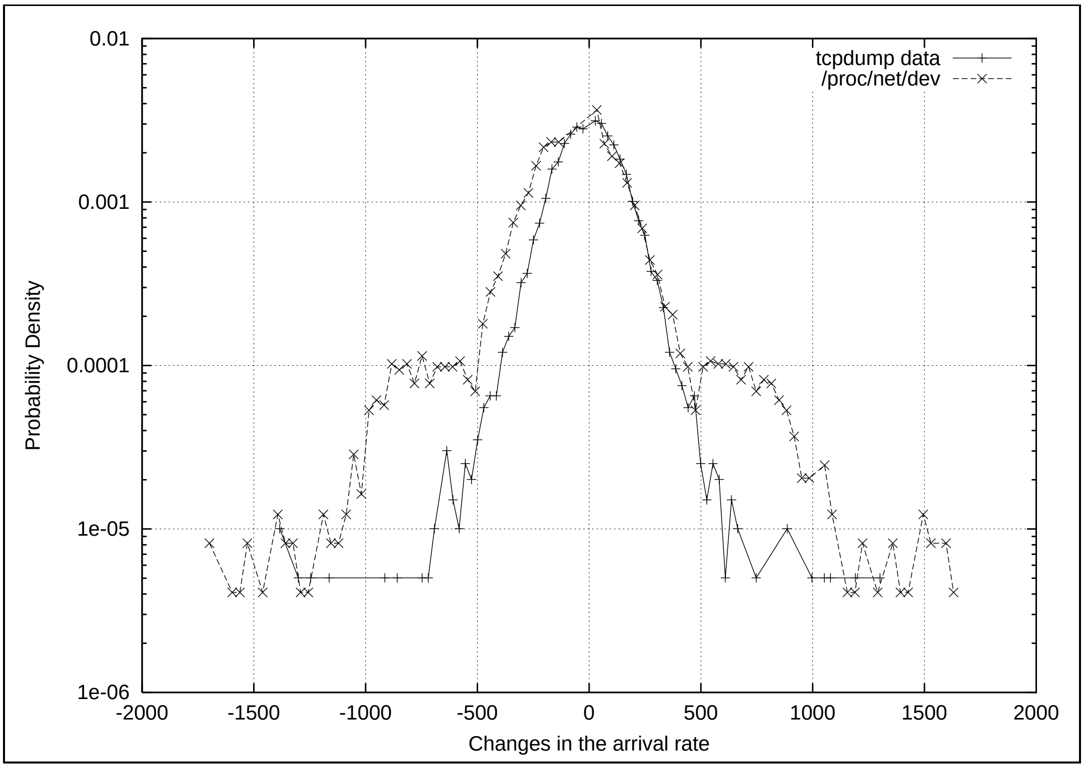

FIGURE 5. The plot displays the distribution of the changes in the packet event rate, measured over the same two-hour period using both `tcpdump` and `/proc/net/dev`.

line with gradient $-(1+\alpha)$. In [4] the authors use the *log-log complementary distribution* plots to estimate the distribution of request sizes.

We have measured both the file size distribution on several Linux/UNIX machines (using an adapted script from [3]) and the distribution of the request size of external requests made to the departmental web server. In doing so, we adopt a different approach by plotting the histogram of the measured pdf using exponentially increasing bin sizes. This exposes more detail in the lower end of the distribution. The results are shown in figure 10 for the files hosted on the SunSite server and figure 11 for the requests to the webserver. Measuring file size distributions for users file spaces reveals remarkably similar patterns.

These measurements are distinctly non-Pareto but we find that they are extremely well approximated by a Cauchy distribution. The positive Cauchy distribution has a pdf given by

$$p(x) = \frac{2}{\pi} \frac{s}{s^2 + x^2} \tag{1}$$

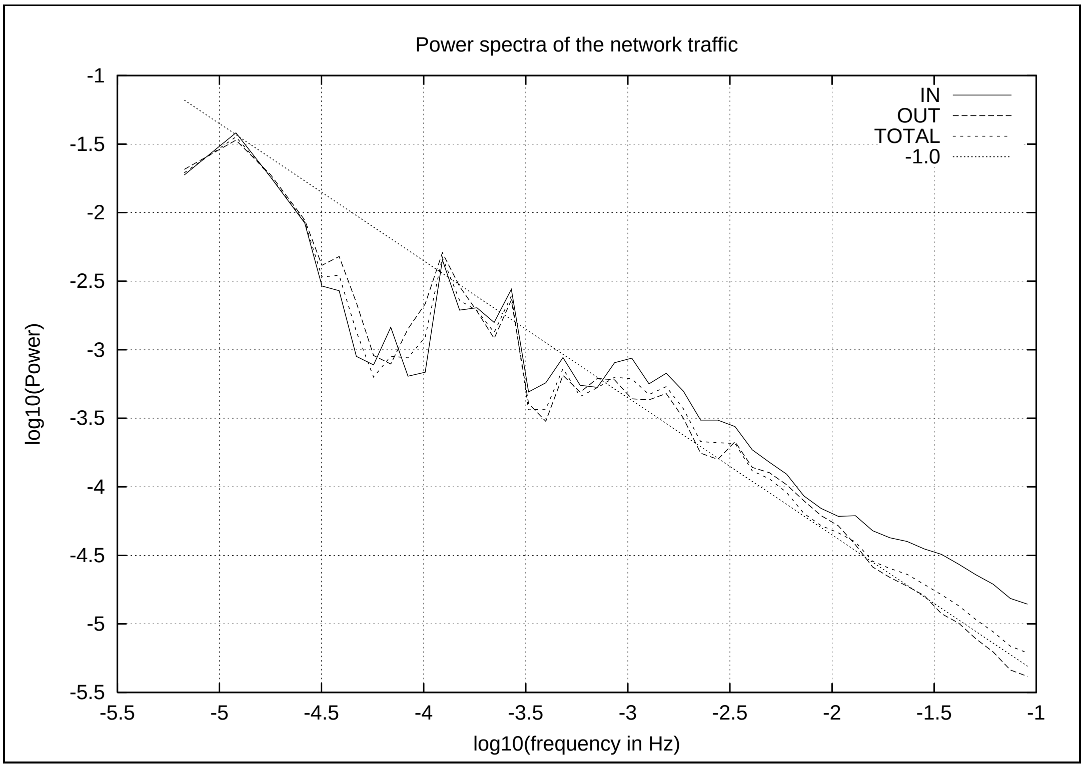

FIGURE 6. Power spectrum of the network traffic time series measured in packets per second, aggregated over 5 second intervals. As described in the text, the `/proc/net/dev` measurement lasted 12 days on the CPU server, starting on 1 February 2002 at 16:24:47

where $s > 0$. The plots in figures 10 and 11 show a Cauchy distribution truncated at the largest file or request and parameters $s = 4900$ and $s = 29000$ respectively. For those values of s the distributions match the measured mean and variance well. For large x the density behaves like $1/x^2$ so the complementary distribution function (cdf) therefore obeys Zipf's law. The distribution is distinguished from the Pareto distribution by its behaviour for small x. This is easily missed when linearly sized bins are used to create the histogram – hence the value of choosing exponentially scaled bin sizes. Although qualitatively the approximation for small file sizes is not brilliant, it is significantly better than the Pareto distribution. On a log-log plot, this is just a straight line. So, although it can be used to model well requests above approximately 1Kbyte (figure 11) it cannot capture the smaller request sizes at all. This has already been pointed out by Downey [28] who puts forward a very simple and elegant model for file size distributions that has a log-normal distribution. Figures 10 and 11 show lognormal distributions

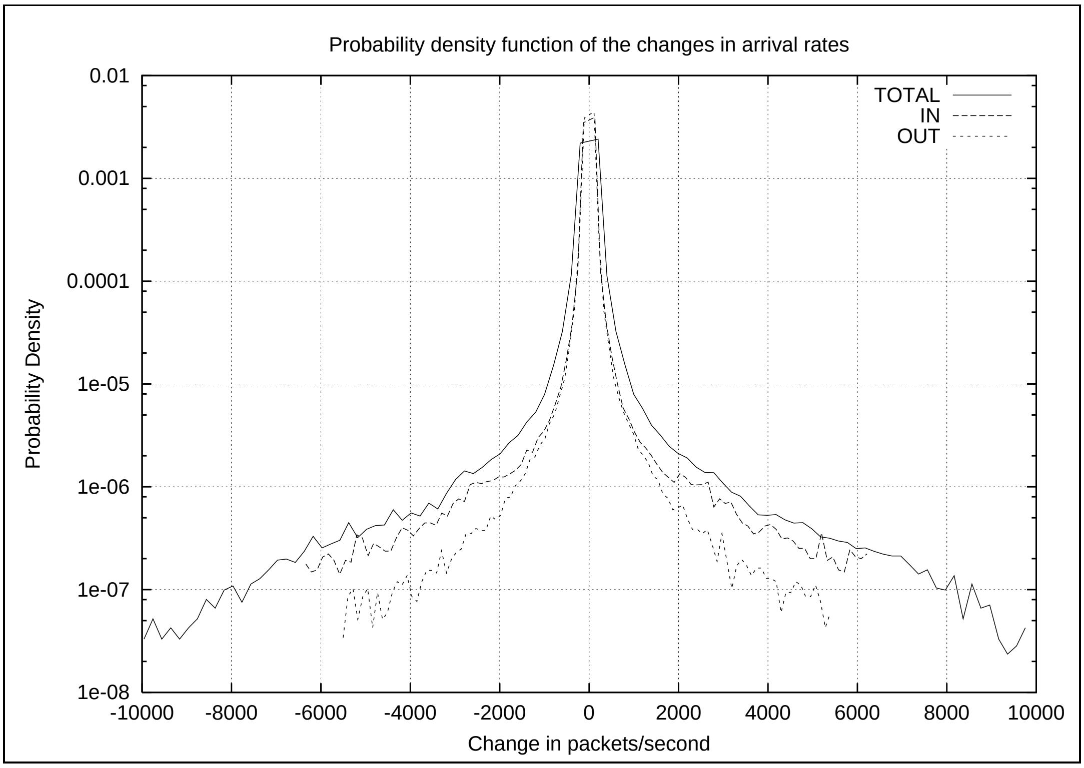

FIGURE 7. Probablity distribution function of the changes in the event rates of packets for the incoming, outgoing and total traffic in a log-linear plot. As described in the text the `/proc/net/dev` measurement lasted 12 days on the CPU server, starting on 1 February 2002 at 16:24:47

superimposed. These were parametrised using the values of logarithmic mean and variance of the observed data.

Note that two of the spikes in the file size distribution correspond to the size of directories on UNIX machines (near 1 and 8 kbyte). For the file size distributions both the logmormal and Cauchy distributions fit the data well. For requests to the website, the log-normal model does much worse than the Cauchy distribution. However, this is not too surprising as the model is not meant to cover this situation as it describes how file size distributions develop.

Recently Marron et al. [29] have developed a model where they use an extension of the Pareto model and overlay up to four Pareto distributions to fit the measured distribution. The numerical results appear to be good, but the fit has to be performed by hand each time. Also, a model just having one of two parameter like the Cauchy or log-normal distribution seems more appealing due to its simplicity.

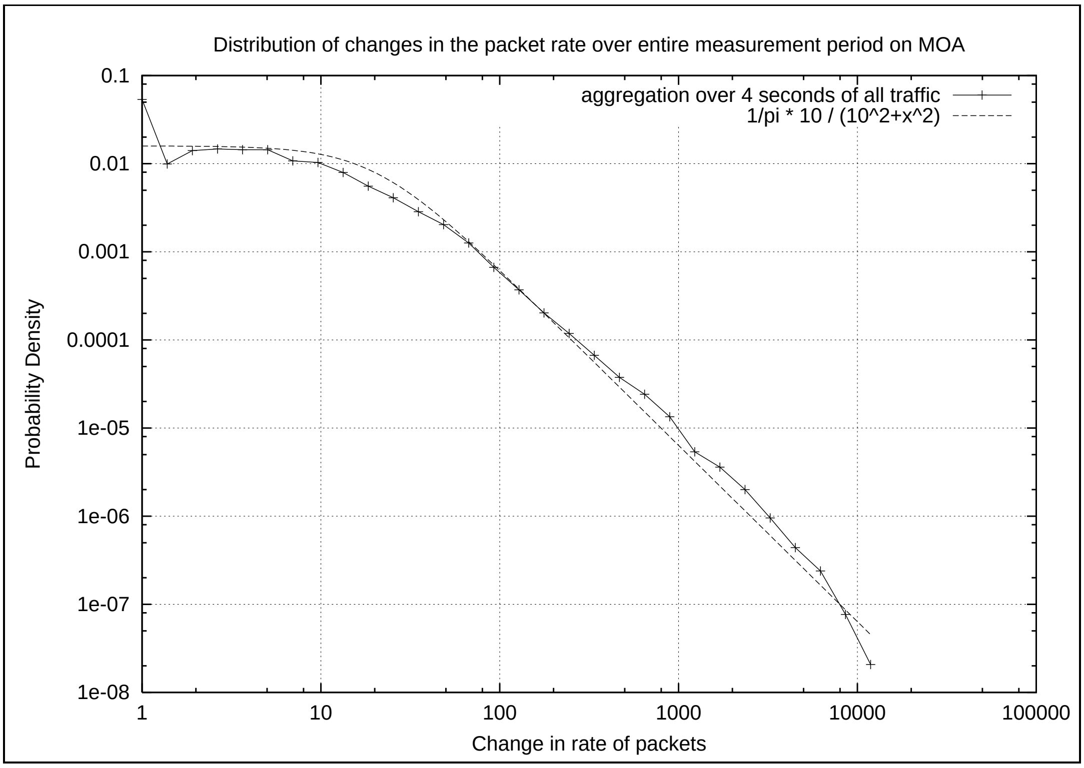

FIGURE 8. Probablity distribution function of the changes in the event rates of packets of the entire traffic compared to a Cauchy distribution in double-logarithmic plot. As described in the text the `/proc/net/dev` measurement lasted 12 days on the CPU server, starting on 1 February 2002 at 16:24:47

A generalization of the Cauchy distribution are the α-stable Lévy distributions. They are defined by their Fourier transform, or characteristic function [24]

$$G(k) = e^{-|k|^{\alpha}\gamma}. \tag{2}$$

The scale parameter γ has to satisfy $\gamma > 0$. The distributions are stable for $0 \leq \alpha \leq 2$. In the family of the Lévy distribution spans all the way from the normal distribution, $\alpha = 2$, to the Cauchy distribution, $\alpha = 1$. Thus by varying the exponent α we can go smoothly from a heavy tailed distribution to one without a heavy tail. We are going to use that property later in our simulation. This distribution may also be useful if a measured turns out to have a different tail behaviour.

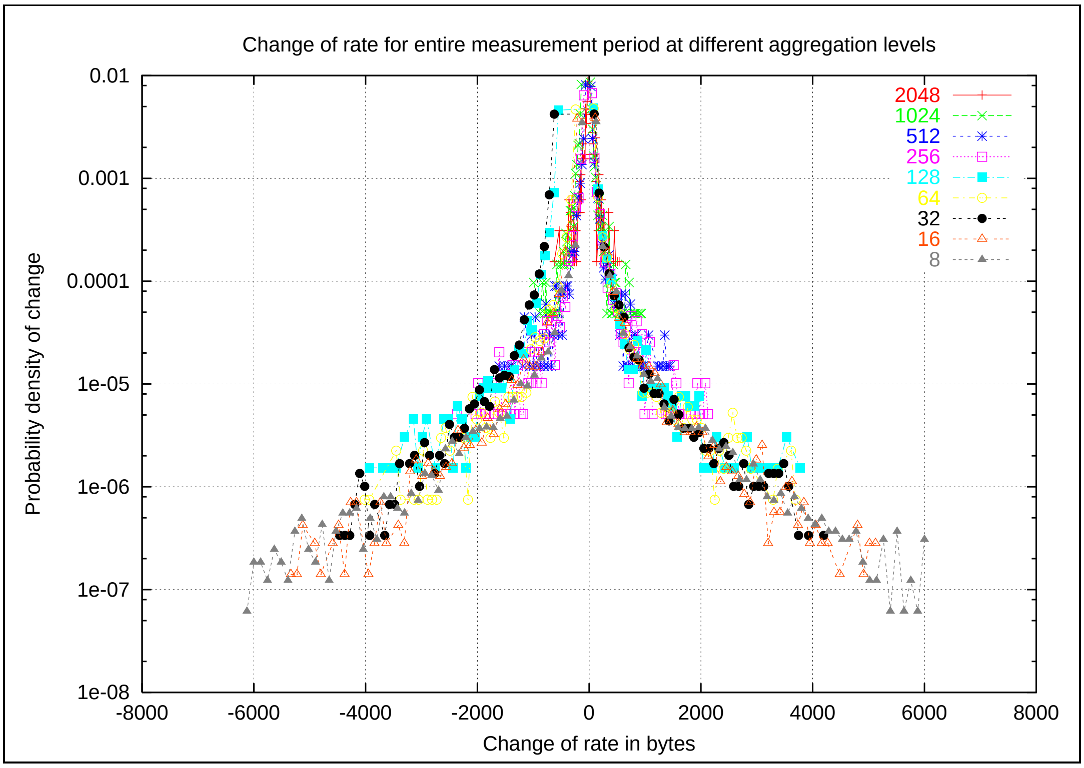

FIGURE 9. Probablity distribution function of the changes in the event rates of packets of the entire traffic at different aggregation levels. As described in the text the `/proc/net/dev` measurement lasted 12 days on the CPU server, starting on 1 February 2002 at 16:24:47

6. Modelling the Traffic

We have now characterised the measured network traffic with its inter-event time histogram, the power spectrum of the aggregated time series and the change in the rates of packets seen on the network.

In this section we describe a simulation study that makes simple assumptions in line with our measurements. We focus attention on the network link at the output of the web server. We model this link using a single server queue where jobs arrive according to a Poisson process and where the service requirement (network transmission) is distributed according to a truncated Cauchy distribution which has a pdf $c(x)$ defined by:

$$c(x) = \begin{cases} p(x)/C & 0 \leq x \leq x_{\max} \\ 0 & \text{else} \end{cases}$$

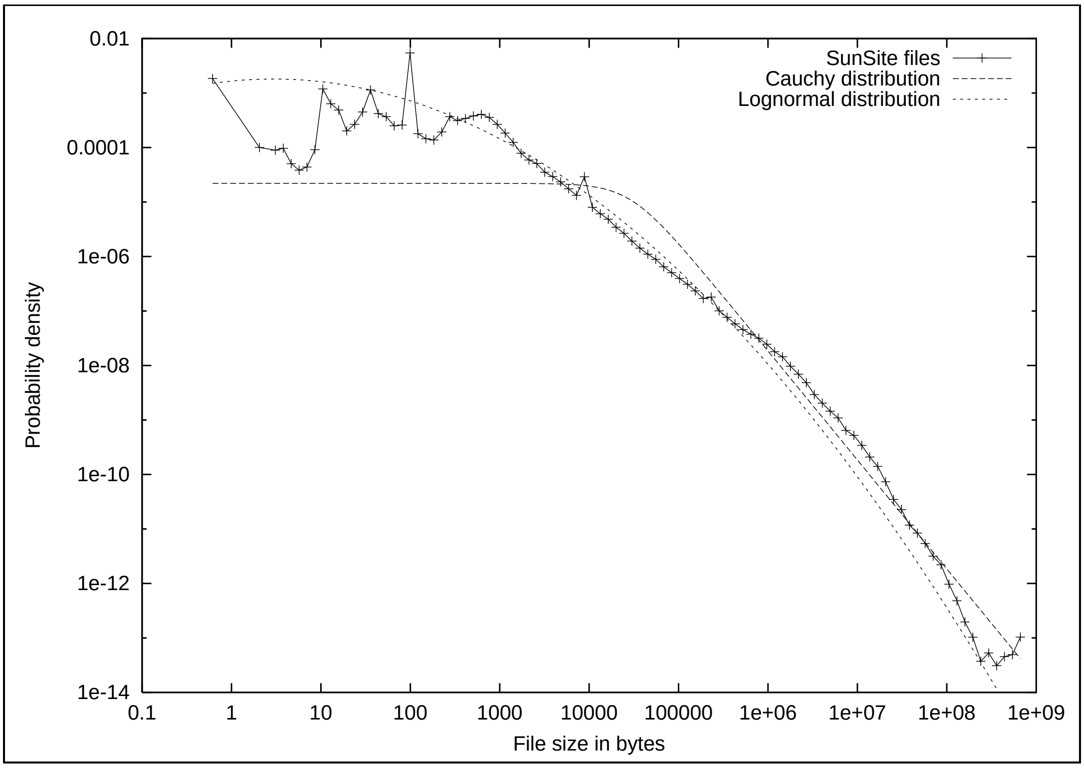

FIGURE 10. The file size distribution of all files at the SunSite server hosted at Imperial College London. In the plot we also show a log-normal distribution and a Cauchy distribution.

where $p(x)$ is given by equation 1 and C is a normalisation constant

$$C = \int_0^{x_{\max}} p(x)\,dx.$$

Note that the truncation of the Cauchy distribution gets rid of its usually prohibiting features like infinite moments.

It is important to understand that we are *not* trying to model a web server here. We are not concerned with multi-threading, process scheduling, disk and CPU accesses etc. caused by a request. Also, we do not take the TCP connection build-up, close down or acknowledgements into account and do not distinguish UDP-based (like most NFS implementations) and TCP-based (like web servers) systems. Instead we focus on the server output where, ultimately, requests are blocked into frames which are added to the input queue of a separate server that represents the network link. Crucially, what we have done is model explicitly the packetisation of requested files into individual frames prior to transmission. In this sense it is *not* a conventional M/G/1 queueing model.

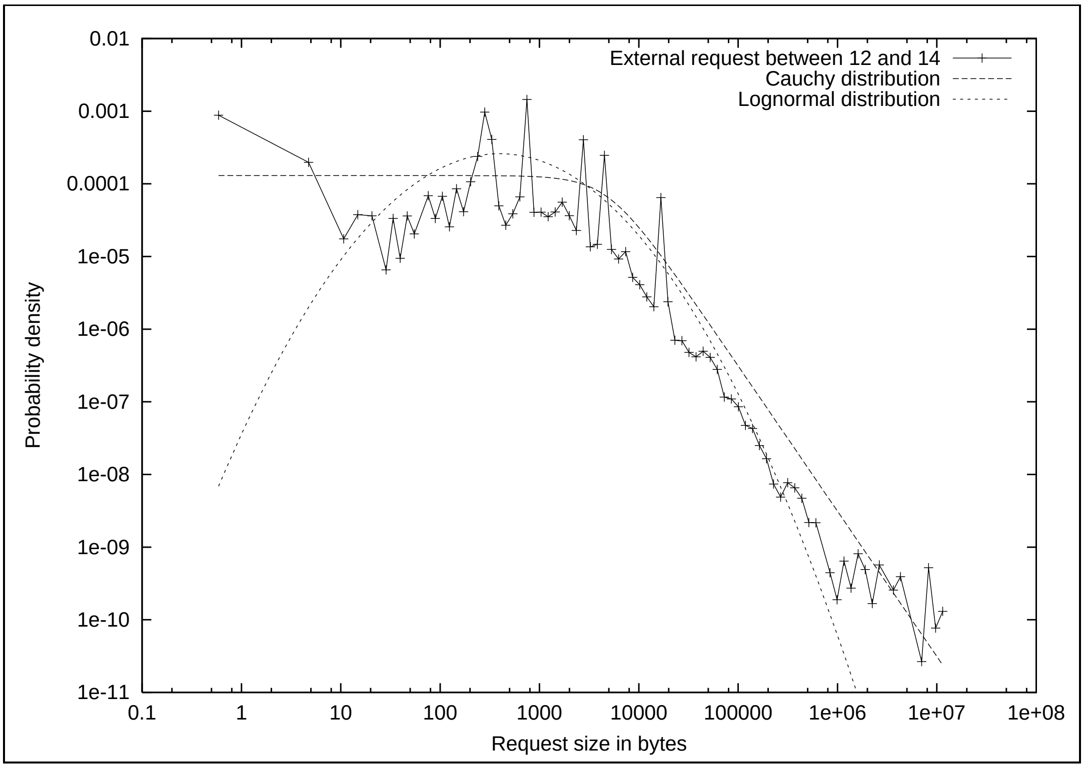

FIGURE 11. This plot compares the request size distribution of external requests on 22 March 2002 between 12.00 - 14.00 to a Cauchy distribution. Though it is clearly not a perfect fit, it describes the nature of the distribution fairly well, especially the feature of requests below one kilobyte. In addition there is a corresponding log-normal distribution displayed.

The server (link) removes work from the queue in blocks (Ethernet frames) and waits after each frame for a short period (inter-frame gap). The blocks are all of the maximum size or less than 1518 bytes. The link server spends the same amount of time serving each block and then waits before the next block is processed, similar to the effect of the inter-frame gap on an Ethernet. We assume that all blocks leave the server at a fixed rate determined by the network speed. The model is illustrated in figure 12 and has been implemented in JAVA.

We measure the cumulative queue length at the server to make sure the system is in equilibrium. Each departure from the system is logged and the time series of departures is analysed like those of real network traffic in previous sections. For the power spectra, we aggregated packet counts in 0.01 second bins.

To parametrise the model we used the log file of the departmental webserver covering the period from 12pm to 2pm on 22 March 2002. As the reader will

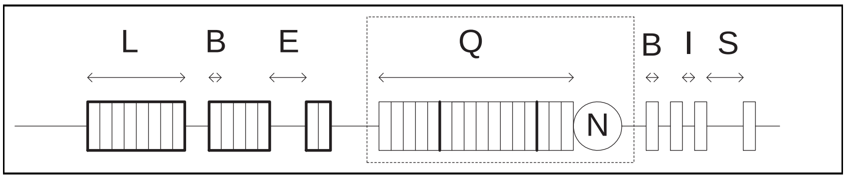

FIGURE 12. This is a sketch of our model. Requests arrive at the left. Their sizes L are measured in bytes and are distributed according to a Cauchy distribution. Each request is then packaged into blocks (frames) of size 1518 bytes or less. The time between requests E is exponentially distributed. The server N emits each block after a service time corresponding to the network speed. The minimum time between blocks after service is the inter-frame gap I of twelve bytes. The time between blocks can be larger I, namely S if the server was idle. The queue-length Q is measured as the number of blocks awaiting service.

recall, we have already presented an analysis of the traffic going through the Black Diamond router. Figure 11 shows the request size distribution for that period and compares it to a Cauchy distribution with $s = 4900$. If we truncate this distribution at 12 Mbytes we get a mean of 24 kbytes per request and a standard deviation of 192 kbytes. This compares to a mean 24 kbytes and a standard deviation of 376 kbytes with the measurements. We chose this value of s as it fits the mean request size extremely well and produces a standard deviation that is in the same order of magnitude as the observed one. Empirically there were 4.6 external requests per second; the same value was used in the simulation.

We make the assumption that the request arrival process is Poisson, partly motivated by the desire to keep the model as simple as possible, given that we cannot tell from the logfile when the requests arrived, only when their service finished. However, there are also sound theoretical reasons for this assumption, see Section 7.

We ran the simulation 10 times with a simulation time of about one day for three different network speeds: 1Mbit/sec, 10Mbit/sec and 100Mbit/sec for the same arrival process specified above. In each case it took at least hours if not days to reach equilibrium. The average queue lengths were 4300, 45 and 3.8 packets with 90% probability confidence intervals of $[3427, 5173]$, $[38, 52]$ and $[3.0, 4.6]$ blocks respectively. The lowest network speed may well have to be run over an even longer time to get a better estimate of queue length.

We analysed the last two hours of each simulation run similarly to the analysis of real network traffic before. To compare our simulation at the network level with the measurements made at the Black Diamond router, we filtered the traffic trace discussed in section 5 for packets originating from the internal webserver which are bigger than one Kbyte. In total there were 556,907 packets fulfilling these criteria. Their mean inter event time was 13 msec with a standard deviation of 68 msec. The simulations were run assuming a network speed of 100Mbps, to enable direct comparisons with the measurements.

6.1. **Departure Process.** A log-log plot of the inter-event time histogram is shown in figure 13. This is worthy of some explanation. Like the measurements the model shows a peak corresponding to the time between transmission of consecutive long packets. This is a common case as large requests are blocked into frames of size 1518 bytes. For requests less than 1518 bytes the request can be serviced by a single frame. The distribution here shows another peak toward zero reflecting the lower end of the (positive) Cauchy distribution; this matches well the observed data which contains, as expected, a relatively large number of minimum-sized frames (4.9μsecond + 0.92μsecond inter-frame gap). The two smaller peaks seen in the real traffic are not directly reproduced in the modelled traffic. The very short inter-event times evident in the real data are timing errors in the `tcpdump` timings. This has been verified by comparing `tcpdump` timings with those obtained by instrumenting the Linux kernel directly using the GILK tool [30].

Beyond 115.92μseconds (largest frame + inter-frame gap) the inter-event times correspond to times where the queue empties and hence includes a request inter-event time. Given our assumption of Poisson arrivals, it is not surprising that the simulation trace in figure 13 beyond 115.92μseconds follows approximately an exponential distribution (the interval in clues both an inter-event time and a block transmission time). Superficially the real traffic follows a similar trend with, interestingly, a tail that appears *not* to be consistent with a heavy-tailed distribution. The sharper slope of this distribution is accounted for by the way the data is measured. Recall that we are measuring the output from the server after it has passed through the main router. Here the traffic is interspersed with frames from elsewhere in the network (the filtered packets are only 8% of the total traffic). The sharp 'on/off' (i.e. high variance) process which we would expect at the server output link is dispersed by the other traffic at the switch. This reduces the variance of the inter-frame times whilst preserving the mean. This explains the shape of the measured distribution.

Figure 14 shows the power spectra for each of the network speeds simulated. There is clear evidence of a power law in case. The increase in network speed appears to shift the frequency interval governed by a power law to the higher frequencies. This is presumably caused by the blocks being processed at a higher rate. The shift is approximately one 'decade' (order of magnitude) for each speed increase by a factor of ten. The slope of the power law is not affected at all by the network speed. When we compare the power spectra, we find that the simulation corresponding to a network speed between 1 and 10Mbps fits best, see figure 14. However, the inter event time histogram would of course not fit terribly well. Still, the simulation shows a much more distinct power law. This is likely to be related

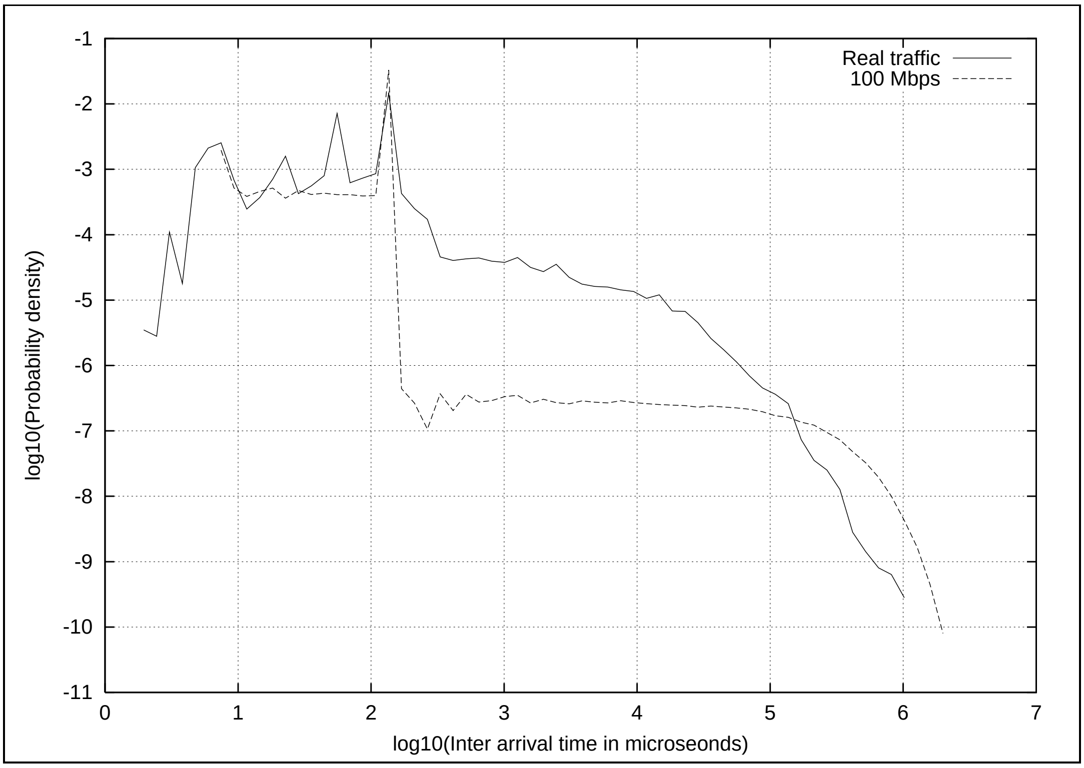

FIGURE 13. Plot comparing the inter event times of the simulation model and the real traffic.

to factors that our model has neglected, like CPU, I/O waits of the webserver and competition with other network traffic. Such traffic might also account for the fact that the empirical data agrees more with the 1-10Mbps simulation than with the 100Mbps, even though the webserver has a 100Mbps network connection. The filtered traffic used to parametrise the simulation accounted for about seven to eight percent of the total traffic. However, the slope of the power law in the power spectrum based on the simulation and the empirical data are very similar.

To demonstrate the effect of the power law in the distribution of the request sizes, we ran our simulation with an exponentially distributed request size. The mean request size was 24 kbytes per request. While the model manages to reproduce the mean inter event time well by construction, it fails to capture the power spectrum, see figure 15. In the simulation, the power law behaviour of the request size distribution can also be seen in the distribution of changes in the packet rate, figure 16. Compare this with the measurements we made at the CPU server, figure 8. The pictures look very similar, supporting the hypothesis that the self-similarity seen at the CPU server is largely caused by the file size distribution.

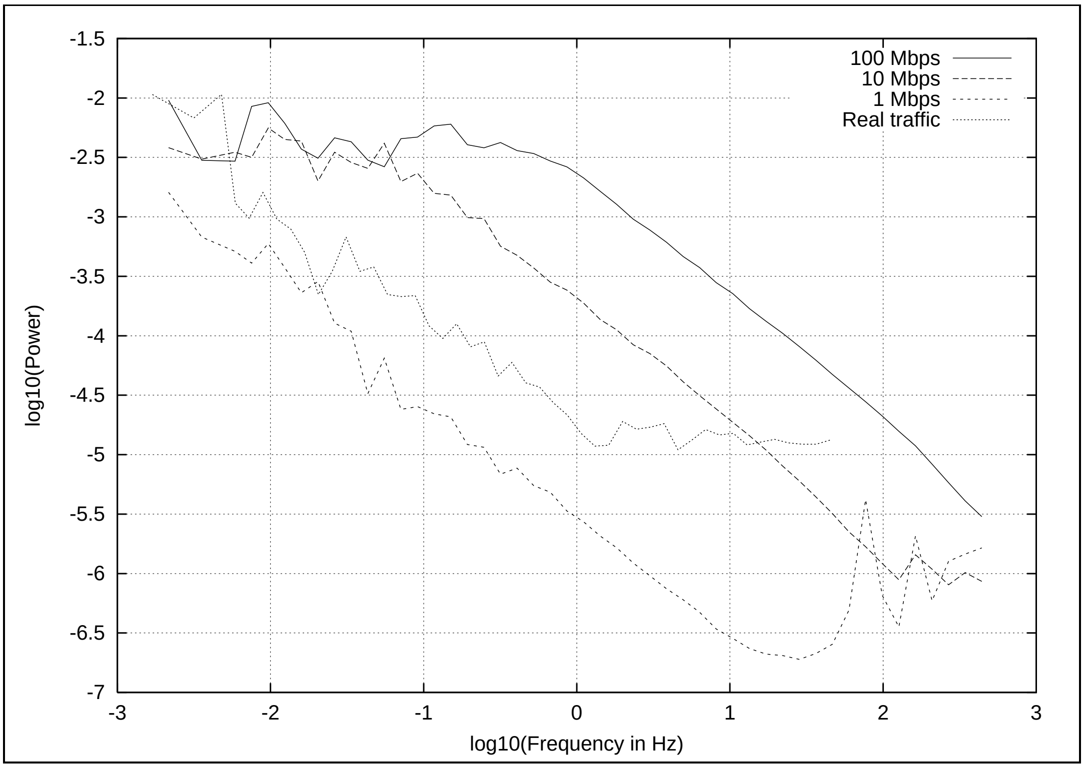

FIGURE 14. The plot compares the power spectra of the departure process of the simulation for different network speeds with that of the real traffic. The real traffic was measured with `tcpdump` on 22 March 2002 between 12pm and 2 pm.

6.2. **An generalisation of the model.** To investigate the influence of the power law of the request size distribution on the on the inter arrival time distribution and the power spectrum we changed our simulation. Instead of using a Cauchy distribution we use Lévy distribution with different exponents ranging from $\alpha = 0.5$ to $\alpha = 2$ (normal distribution). To generate the random number we used the method published in [31]. To generate Lévy distributed random numbers $l_\alpha(x)$ with exponent α

$$X = \frac{\sin \alpha\gamma}{(\cos\gamma)^{1/\alpha}}\left(\frac{\cos((1-\alpha)\gamma)}{W}\right)^{(1-\alpha)/\alpha}$$

with γ uniformly distributed in $(-\pi/2, \pi/2)$ and W exponentially distributed; W and γ are independent. Figure 17 illustrates the difference in the resulting different request size distributions.

As we can see the inter arrival time distribution does not change much in its overall behaviour. This would imply that it is mainly determined by the network speed and the arrival process.

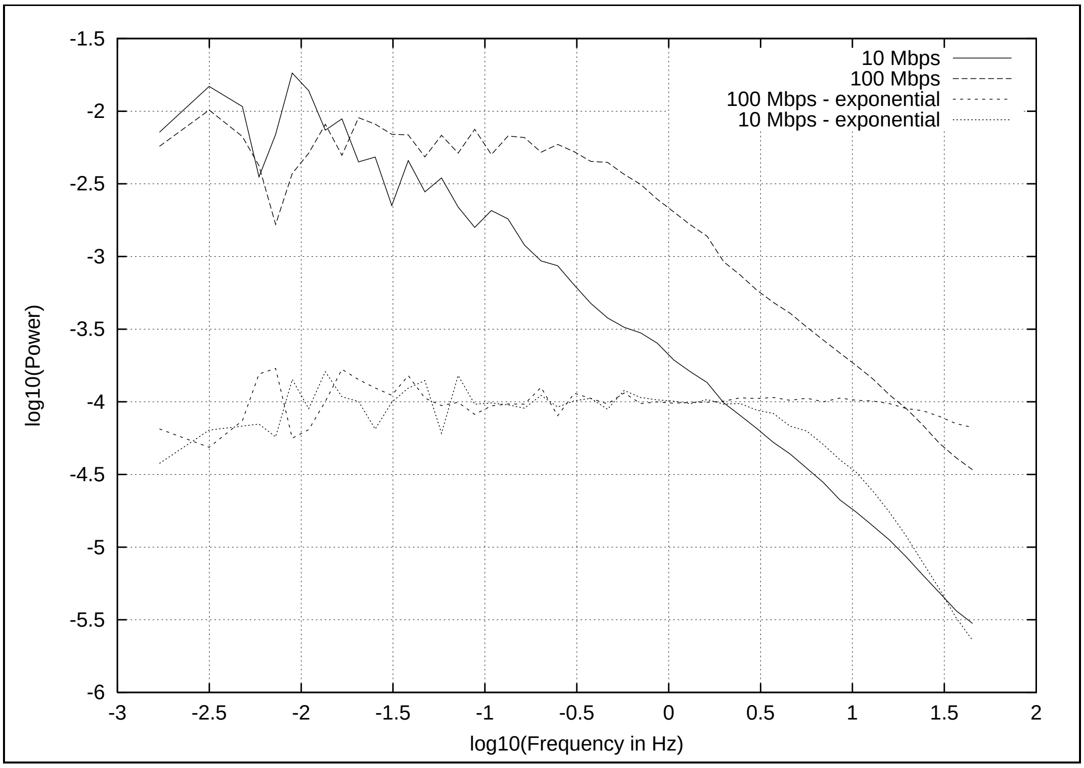

FIGURE 15. This plot compares the power spectra of the simulation using Cauchy distributions to ones using exponential distributions for the request sizes.

In contrast to that the power spectrum changes its behaviour dramatically. The request distribution with no heavy tail produce a flat power spectrum, as expected. With increasing the parameter α the slope of the power spectrum increases as well. This suggests a direct link between the slope of the power law of the request size distribution and the power spectrum.

7. ANALYTICAL MODELLING

The comparison between the properties of the traffic generated by the simulation and that observed at the Black Diamond router suggest that file size distribution can be a sufficient cause of power laws, self-similarity and long range correlation. Significantly, this is the case even for Poisson external arrivals. Although we have not tested the hypothesis that external request instants approximate Poisson streams (we have not been able to measure this directly), this has been found to be the case for many decades in a wide range of teletraffic systems. In fact, the hypothesis has sound theoretical foundations. It can be proved that a superposition of arrival processes is asymptotically Poisson under appropriate

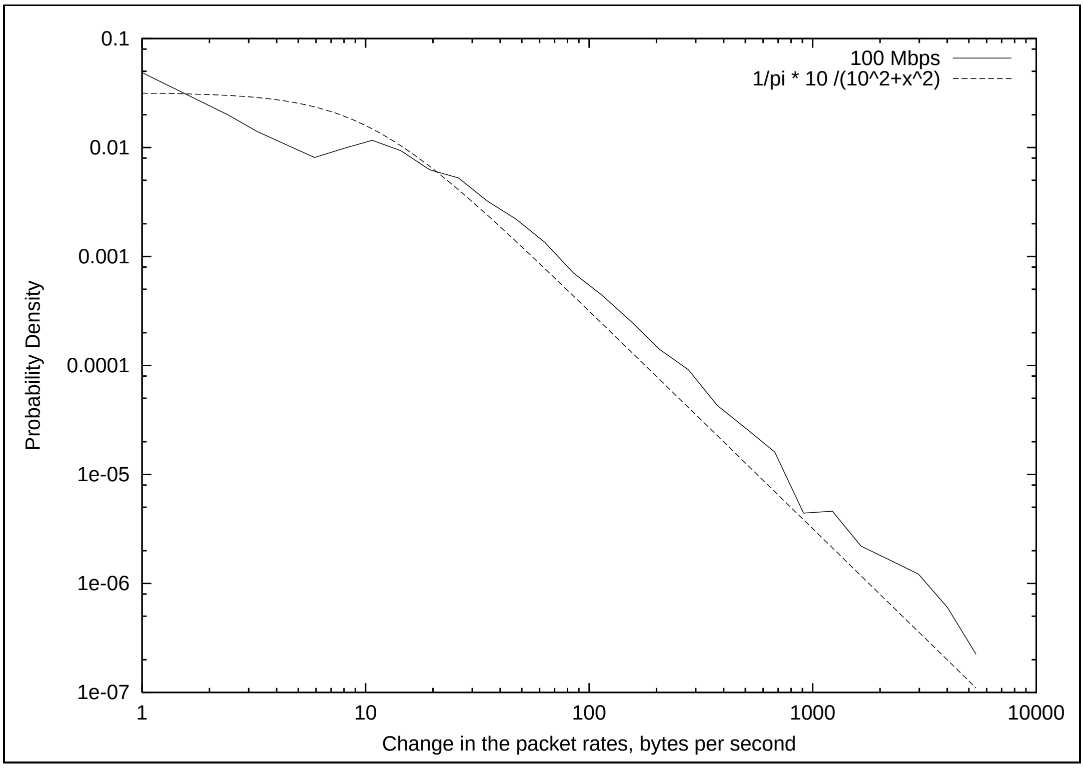

FIGURE 16. This plot compares the changes in the packet rate for the simulation running at 100 Mbps to that of a Cauchy distribution.

conditions; for example a large number of independent renewal processes of which none is dominant, i.e. has a much higher rate than the average – see [32, 33] for example.

In any packet-based communication network, the output traffic from a server comprises (a) a sequence of constant-size packets separated by the small inter-frame gap when the server is busy (with *any* work); (b) a null stream, when the server is idle. Put in queueing terms, the output is an on-off process with streamed packet arrivals during the on-periods, which are the *busy periods* of the queue, and off-periods distributed according to the inter-arrival time distribution. Clearly, for heavy-tailed service times, the busy period must also be heavy tailed, although it is hard to calculate precisely the busy period distribution in general.

If we consider the output from the simulation model above, each busy period comprises a consecutive sequence of packet streams, one for each request in the (FIFO) queue. When we measure real data on a network the packets we see are an *interleaving* of the packet representations of each input request—an artefact of the operation of the servers themselves. If it is indeed the case that requests to

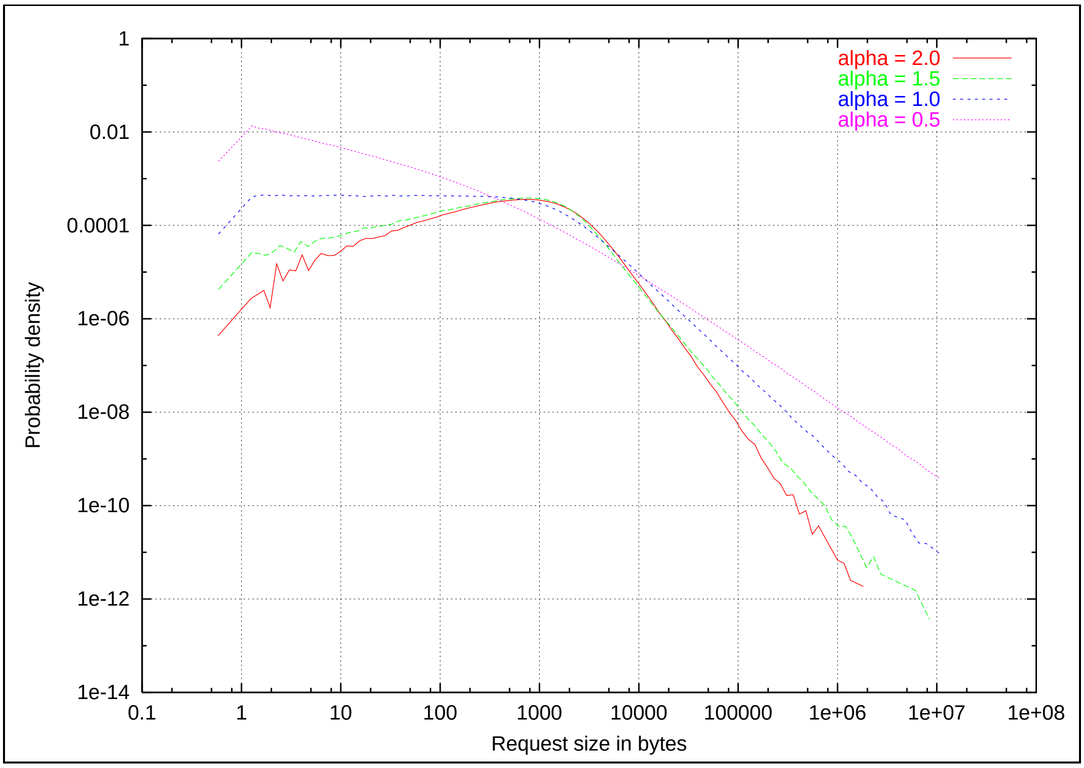

FIGURE 17. The request size density distribution using different Lévy distributions.

the server are approximately Poisson and that the request sizes are approximately Cauchy then the observed packet stream and that produced by the simulation should be approximately the same in terms of these on/off processes.

The instrumentation in our network precludes direct measurement of the output process so it is not possible to measure directly the busy periods at the server output. However, if they can be shown to match the busy period of an M/G/1 queue with Cauchy-distributed service times [5] then a conventional M/G/1 queue with processor sharing would appear to be a good model of the server as a whole. Moreover, if the external arrivals really are well approximated by Poisson processes, and if the nodes' service times are independent, then whole networks would be amenable to analysis using conventional queueing networks with *processor sharing* (PS) queueing discipline. It is because of the interleaving of packet streams, referred to above, that we believe PS discipline to be the most appropriate. In these circumstances, by the BCMP Theorem [34], product-form solutions will exist, from which performance measures like mean queue length, throughput

[5]The fragmentation of requests into frames (packets) introduces a small gap between frames but this adds less than 1% to the busy period.

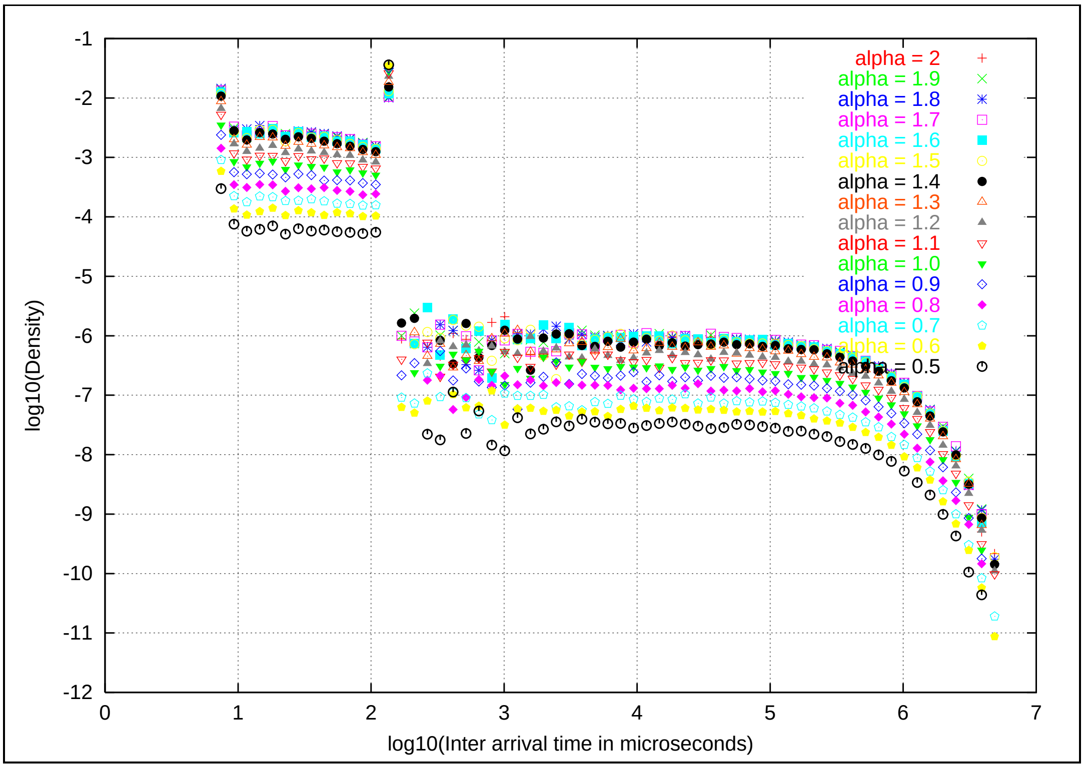

FIGURE 18. The inter event time histogram for the simulation for different Lévy distributions.

and mean response time follow via recursive algorithms that can be implemented efficiently. Such models can easily accommodate multiple classes of traffic (with different demands at each node and different paths through the network). The extent to which BCMP queueing networks provide adequate approximate models of networks of this sort is a key next step in our proposed research.

8. Conclusions and Future Work

In this paper we have reported results from the analysis of output traffic at the central router of a departmental switched Ethernet and have investigated the possibility of using queueing models to characterise the behaviour of such networks.

The measurement and analysis reported in this paper has shown that the power spectrum of the output process conforms to a power law, consistent with observations reported elsewhere in the literature. However, we have postulated that the output behaviour may be governed by the nature of the service processes involved, rather than the arrival processes. This was inspired by the observation that the various files requested externally are found to have a heavy-tailed

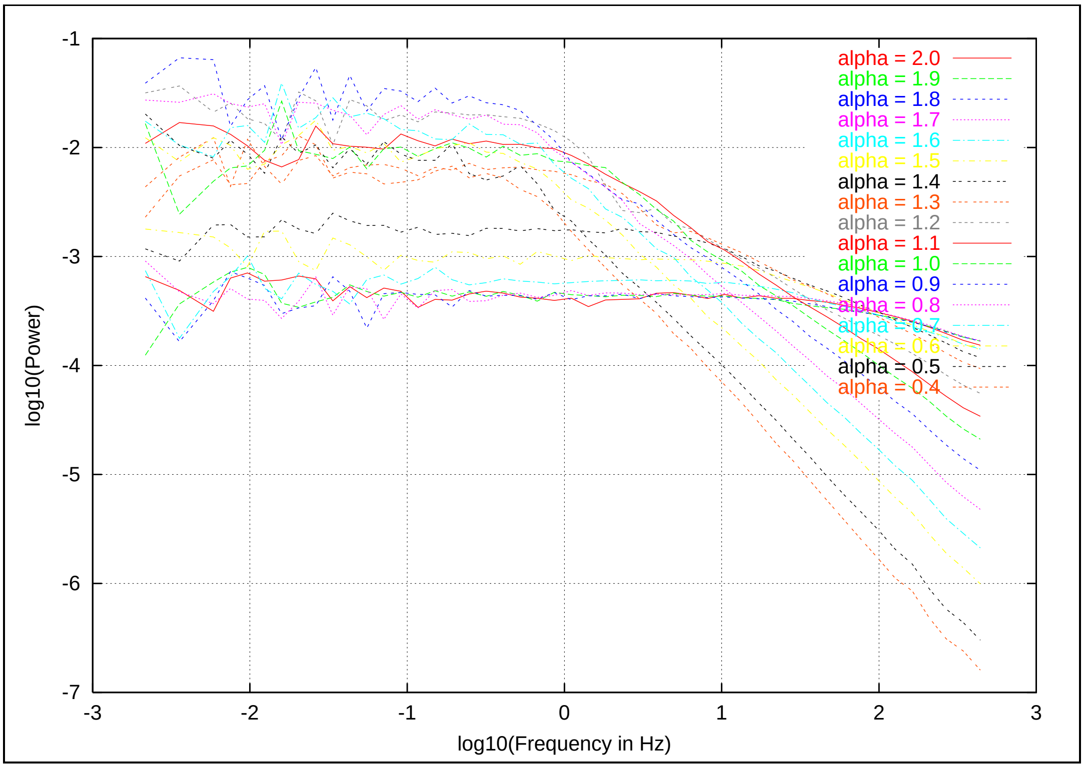

FIGURE 19. The power spectrum of the simulation for different Lévy distributions.

distribution. Similar observations have been reported by other authors, but in this paper we have proposed the Cauchy distribution as providing a superior fit to request sizes than those previously suggested; this is especially true for small file sizes. We have verified through simulation that power laws very similar to those observed in practice can be produced by a simulation model of a server network link servicing Cauchy-distributed request sizes, even assuming Poisson arrivals. Unlike a conventional M/G/1 queue the simulation models explicitly the transmission of requests by (spaced) Ethernet frames. This experiment does not tell us that the arrival processes *are* Poisson, but that if they are it is still possible to observe self-similar behaviour in the output process.

A detailed investigation into the applicability of BCMP queueing networks, in particular an assessment of the accuracy of the processor sharing and independence of nodes' service times assumptions, constitutes a major next step in the proposed research. In practice, of course, we would not always expect service times to be independent, particularly in situations where files are propagated through a series of routing nodes. However, it may be a reasonable approximation. We suspect, however, that process sharing is likely to provide a very good approximation of the behaviour of the nodes. This is certainly true for web servers and, very likely, file servers which typically operate NFS. These architectures are heavily multi-threaded, with the sharing of the available network resources among the various users being an important design objective.

Finally, we would like to investigate how we can incorporate results in [7, 8, 9, 10], which indicate that the congestion avoidance algorithm of TCP [35] introduces power laws, into to our model. Perhaps combining our findings with a model based on dynamical systems, e.g. [36], will shed more light on the nature of network traffic. In Mathematical and Theoretical Physics self-similarity has been studied widely over the last few decades, motivated by the widespread occurrence of $1/f$ noise in natural phenomena. There are many areas in science where $1/f$-noise has been seen: see for instance Jensen's book [37] for a good overview of the topic. A possible interpretation of $1/f$ noise is the notion of self-organised criticality (SOC) [37]. The word criticality is borrowed from physics where a critical state of a system is related to an infinite correlation length and the system going through a phase transition.

Acknowledgements

The authors would like to thank the Computer Support Group in helping with the data capture. We would also like to thank Jörn Davidsen of the Chemical Physics Theory Group (Department of Chemistry University of Toronto) for fruitful discussions on self-similarity and criticality, and Will Knottenbelt and David Thornley for stimulating conversations.

The research was funded by EPSRC (research grant QUAINT, GR/M80826).

References

[1] Will E. Leland, Murad S. Taqq, Walter Willinger, and Daniel V. Wilson. On the self-similar nature of Ethernet traffic. In Deepinder P. Sidhu, editor, *ACM SIGCOMM*, pages 183–193, San Francisco, California, 1993.

[2] Ashok Erramilli, Onuttom Narayan, and Walter Willinger. Experimental queueing analysis with long-range dependent packet traffic. *IEEE/ACM Transactions on Networking*, 4(2):209–223, 1996.

[3] Gordon Irlam. Unix file size survey. `http://www.base.com/gordoni/ufs93.html`.

[4] Mark Crovella and Azer Bestavros. Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes. In *Proceedings of SIGMETRICS'96: The ACM International Conference on Measurement and Modelling of Computer Systems.*, Philadelphia, Pennsylvania, May 1996. Also, in Performance evaluation review, May 1996, 24(1):160-169.

[5] Kensuke Fukuda, Hideki Takayasu, and Misako Takayasu. Origin of critical behaviour in ethernet traffic. *cond-mat/0007435*.

[6] Torsten Huisinga, Robert Barlovic, Wolfgang Knospe, Andreas Schadschneider, and Michael Schreckenberg. A microscopic model for packet transport in the internet. *Physica A*, 294:249–256, 2001.

[7] Andras Veres and Miklos Boda. The chaotic nature of TCP congestion control. *INFOCOM*, (3):1715–1723, 2000.

[8] Hans-Peter Schwefel. Behavior of tcp-like elastic traffic at a buffered bottleneck router. *http://citeseer.nj.nec.com/451117.html*.

[9] Biplab Sikdar and Kenneth S. Vastola. The effect of tcp on the self-similarity of network traffic. *http://citeseer.nj.nec.com/sikdar01effect.html*.

[10] D.E. Newman, Nathaniel D. Sizemore, B.A. Carreras, and V.E. Lynch. Growth and propagation of disturbances in a communication network model. *Hawaii International Conference on Systems Sciences*, January 2002.

[11] R. N. Mantegna and H. E. Stanley. *An Introduction to Econophysics.* CUP, ISBN 0 521 62008 2, 2000.

[12] J. Voit. *The Statistical Mechanics of Financial Market.* Springer, ISBN 3 540 41409 6, 2001.

[13] Götz Brasche and Bernhard Walke. Concepts, services, and protocols of the new gsm phase 2+ general packet radio service. *IEEE/Communications Magazine*, pages 94–106, August 1997.

[14] Measurement data. *http://www.doc.ic.ac.uk/~uh/QUAINT/data/*.

[15] Tcpdump and website. `http://www.tcpdump.org/`.

[16] ExtremeNetworks and documentation. `http://www.extremenetworks.com`.

[17] Benoit Mandelbrot. *The Fractal Geometry of Nature.* Freeman, 1982.

[18] Jens Feder. *Fractals.* Plenum, 1988.

[19] B. Pilgram and D. T. Kaplan. A comparison of estimators of $1/f$ noise. *Physica*, D 114:108–122, 1998.

[20] M.S. Taqqu, V. Teverovski, and W. Willinger. Estimators for long-range dependence: an empirical study. *Fractals*, 3(4):785–798, 1995.

[21] S. Thurner, S. B. Lowen, M. C. Feurstein, C. Heneghan, H. G. Feichtinger, and M. C. Teich. Analysis, synthesis, and estimation of fractal-rate stochastic point processes. *Fractals*, 5(4):565–595, 1997.

[22] N. Wiener. Generalized harmonic analysis. *Acta Mathematica*, 55:117, 1930.

[23] A. Khintchine. Korrelationtheorie der stationären prozesse. *Mathematische Annalen*, 109:604, 1934.

[24] J. Honerkamp. *Statistical Physics.* Springer, 2002.

[25] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C 2nd ed.* CUP, 1993.

[26] Jan Beran. *Statistics for Long-Memory Processes.* Chapman & Hall, 1994.

[27] P. Abry and D. Veitch. Wavelet analysis of long-range-dependent traffic. *IEEE Transactions on Information Theory*, 44(1):2–15, 1998.

[28] Allen B Downey. The structural cause of file size distributions. *SIGMETRICS/Performance*, pages 328–329, 2001.

[29] F. Hernández-Campos, J.S. Marron, G. Samorodnitsky, and F.D. Smith. Variable heavy tailed durations in internet traffic part I: Understanding heavy tails. *MASCOTS Proceedings*, pages 43–50, 2002.

[30] David Pearce, Paul Kelly, Uli Harder, and Tony Field. Gilk: A dynamic instrumentation tool for the linux kernel. *Modelling Techniques and Tools 12th International Conference, TOOLS 2002 London, UK, April 14-17, Proceedings (ISBN 3 540 43539 5)*, pages pp.220–226, April 2002.

[31] Gennady Samorodnitsky and Murad S. Taqqu. *Stable non-Gaussian Random Processes.* Chapman & Hall, 1994.

[32] H.J. Larson and B.O Shubert. *Probabilistic models in engineering sciences.* Wiley, New York, 1979.

[33] P.G. Harrison and N.M. Patel. *Performance Modelling of Communication Networks and Computer Architectures.* Addison-Wesley, 1993.

[34] Forest Baskett, K. Mani Chandy, Richard R. Muntz, and Fernando G Palacios. Closed and mixed networks of queues with different classes of customers. *Journal of the ACM*, 22(2):248–260, April 1975.

[35] Van Jacobson. Congestion avoidance and control. *ACM Computer Communication Review Proceedings of the Sigcomm '88 Symposium in Stanford, CA*, 18(4):314–329, August 1988.

[36] Frank Kelly. Dynamic thing. 29:370–377, 1987.

[37] P. Bak, C. Tang, and K. Wiesenfeld. Self organised criticality: an explanation of 1/f noise. *Physical Review Letters*, 59:381, 1987.

[38] The internet traffic archive. *http://www.acm.org/sigcomm/ITA/*.

APPENDIX: Comparison with the LBL data from the Internet Traffic Archive

In addition to data collected locally, we have also analysed the Ethernet data sets in the Internet traffic archive [38]. This is the data that has been measured and discussed in [1]. We used the following files: pAug.TL.Z, pOct.TL.Z, OctExt.TL.Z and OctExt4.TL.Z. The first two are shorter traces of one million events measured on the internal Ethernet, the latter two are longer traces measuring the external events only. For a more detailed description see [1].

The time-stamps in the files have microsecond granularity although accuracy is limited to around 10 microseconds by the hardware clock (four microseconds) and bus contention. However, this is still (just) enough to distinguish between packets as there is around a 10 microsecond silence between packets on a 10Mbps Ethernet and the smallest packet lasts about 60 microseconds.

We applied the same methods used to analyse our own data to the Bellcore data investigated in [1]. Looking at the inter-event time histograms in figures 2 and 20, the internal traces are similar to our data. This data also exhibits the same two peaks corresponding to the smallest and largest packet size on Ethernets. Note, though, that the peaks are shifted to the right by an order of magnitude due to the different speed of the measured Ethernets. This seems to suggest that the measurement errors for the time stamps are not as bad as the authors of [1] thought. It also suggests an interesting way to measure the speed of an unknown network.

Just like our data, the internal traces do not provide a power law in the inter-event time distribution. There is also a similar picture when we look at the power spectra in figures 21 and 3. The ranges of the power laws in the power spectrum are similar to those measured in our experiments. The external traces seem to provide a better power law than the internal ones. There is, however, a difference in the data when we look at the changes in the packet rates. Figure 22 shows a linear plot for the internal sources. In contrast to the data we measured, see figures 5,7,8 and 9, the internal data shows no sign of a power law for this metric. However, the external traces plotted in figure 23 show a similar leptokurtic behaviour [12]. The slope in the log-log plot is different to ours; in the Bellcore data it is approximately -2.4, see figure 24. In summary the data we

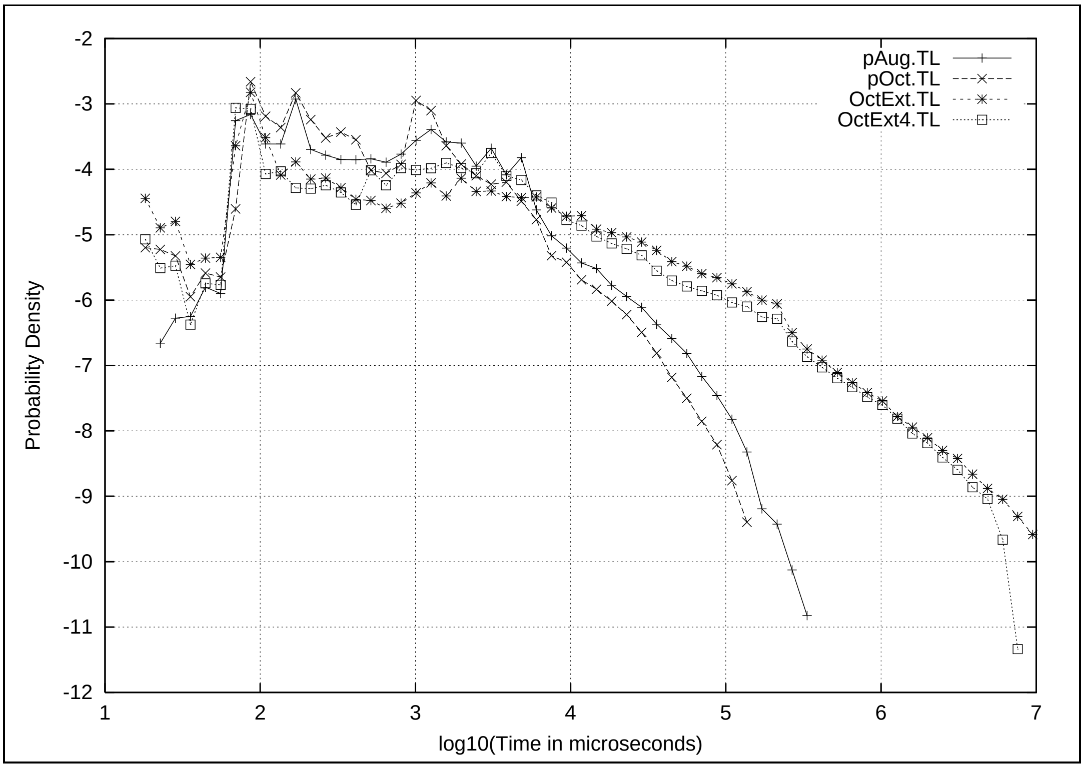

FIGURE 20. This plot shows the inter event time histograms for the four Bellcore traces.

have gathered shows a slightly different quantitative behaviour compared to the Bellcore data, especially in the change of the packets' rates. However, there are many qualitative similarities, for example in the inter-event time histograms and the power spectra.

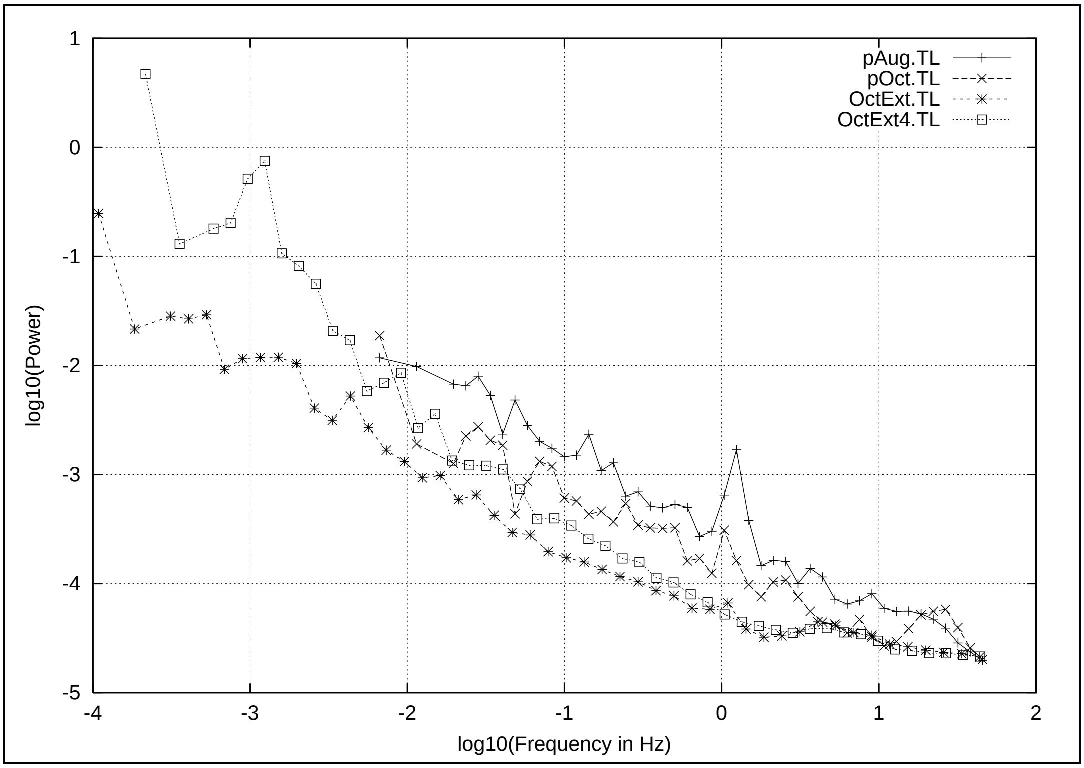

FIGURE 21. The power spectra of the four Bellcore traces, based on an aggregation of 10 msec.

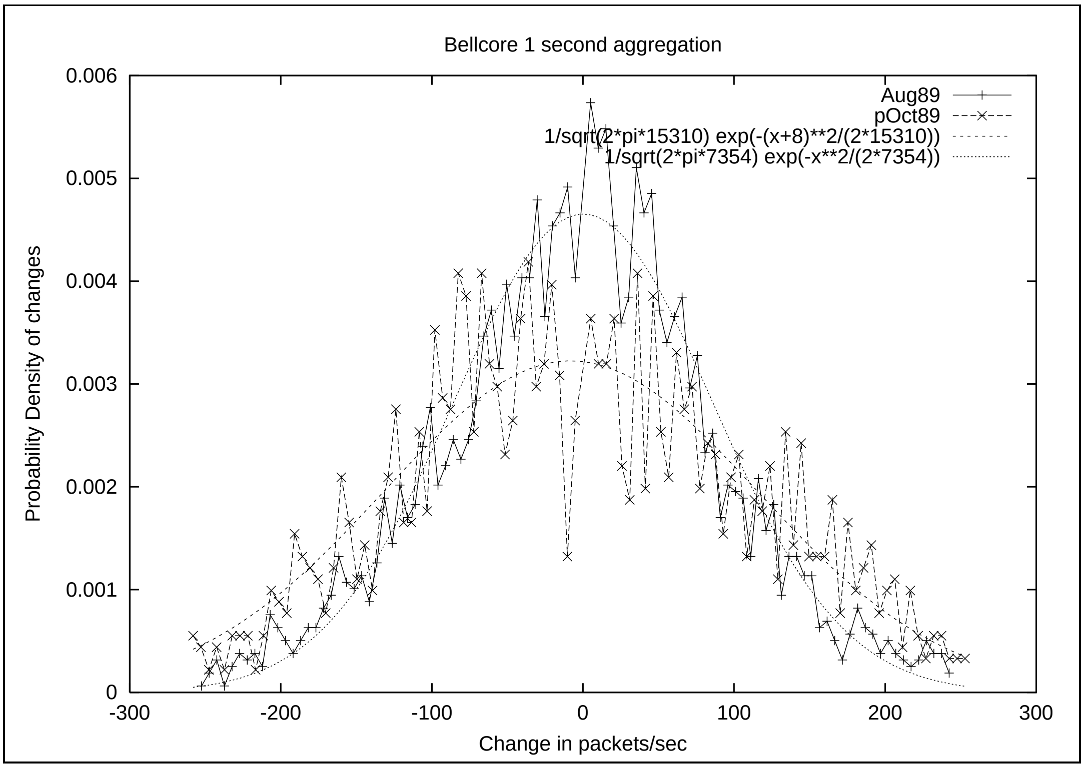

FIGURE 22. The change in the packet rates for the internal traces as a linear plot. The two lines are normal distributions with the same mean and average as the measured data.

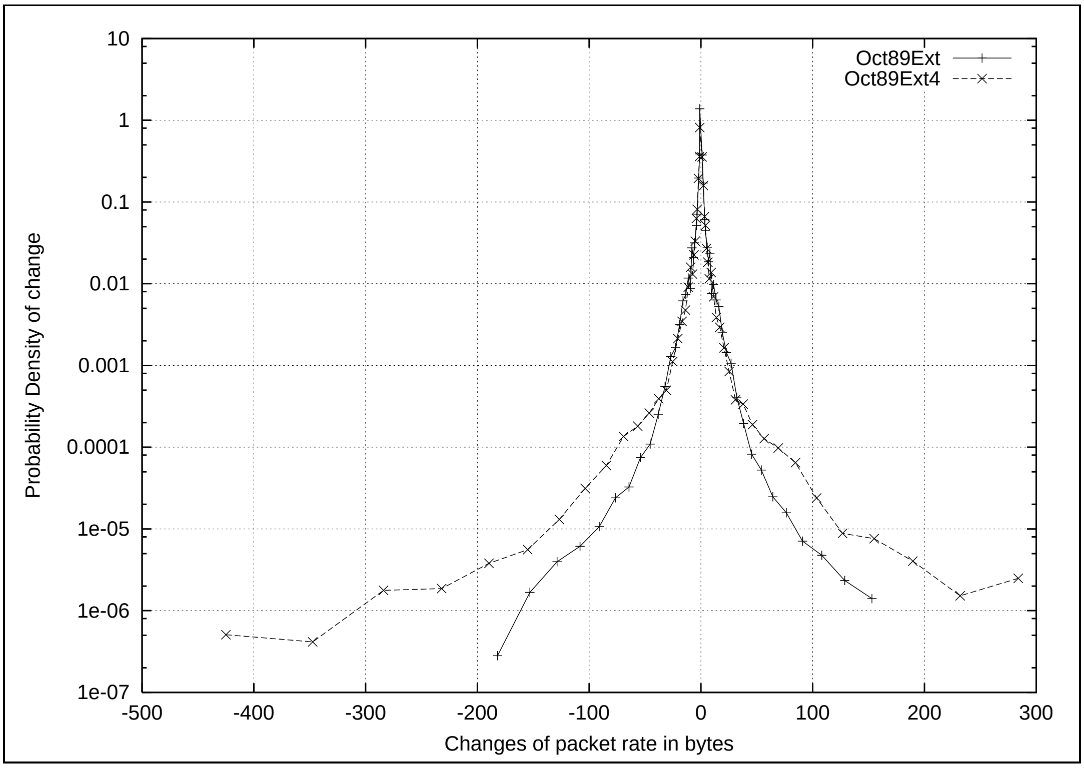

FIGURE 23. The change in the packet rates for the external traces as a log-linear plot.

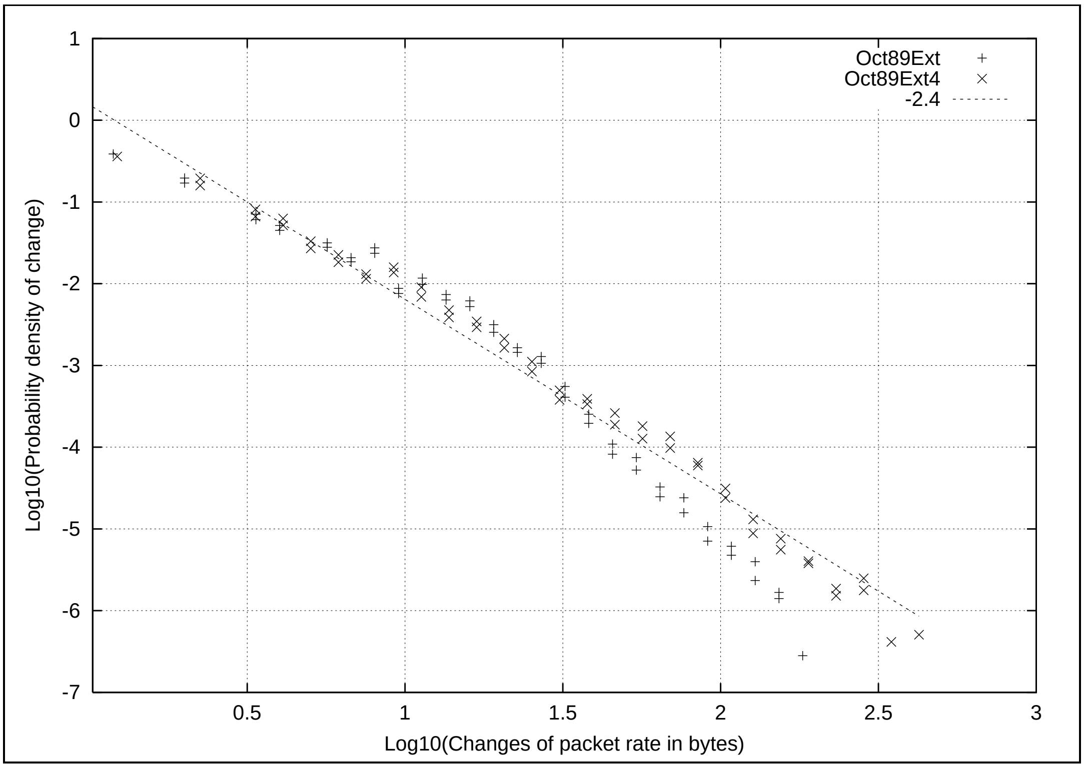

FIGURE 24. The change in the packet rates for the external traces as a log-log plot to determine the slope of the power law.