Real-Time Estimation of the Parameters of Long-Range Dependence

Matthew Roughan, *Member, IEEE*, Darryl Veitch, and Patrice Abry

***Abstract*—An on-line version of the Abry-Veitch wavelet-based estimator of the Hurst parameter is presented. It has very low memory and computational requirements and scales naturally to arbitrarily high data rates, enabling its use in real-time applications such as admission control, and avoiding the need to store huge data sets for off-line analysis. The performance of the estimator as a function of the length of data processed is demonstrated using simulated data. An implementation for 10-Mb/s Ethernet based on standard hardware supporting sampling rates of 1 data point per millisecond is described, and results of its operation presented, as is an implementation for 155-Mb/s asynchronous transfer mode networks. Finally we illustrate the power of on-line measurements by collecting measurements over a period of five months, and using them to look for diurnal trends in scaling properties of the data.**

***Index Terms*—Estimation, fractal, Hurst parameter, long-range dependence, on-line, real-time, self-similar, traffic modeling, wavelets.**

I. Introduction

REAL-TIME traffic measurement is necessary to support network management tasks such as call admission control, rate adaptation, and network monitoring. As such activities must take place on the small time scales implied by the high bandwidth of modern telecommunications systems, the extent of such measurements and the complexity of the algorithms which use them are limited by hard processing constraints, a situation which is unlikely to change. Even in the less demanding case of off-line processing, the ever increasing volume of data that can be collected over a given time interval poses huge storage and processing problems. Such limitations are particularly serious if parameters crucial to meaningful traffic characterization have high computational complexity, say of $O(n^2)$ where n is the length of the data.

In the last few years the discovery of the *self-similar* nature of many kinds of packet traffic [1], [2] has inspired a small revolution in the way that high-speed traffic is viewed. Although no single model is accepted as definitive, the *Hurst* parameter H, which describes the degree of self-similarity, holds a central place in the description of such traffic. Its accurate measurement is therefore of considerable importance for the provision of quality of service as well as for the dimensioning of networks. Unfortunately, methods for the estimation of this parameter from data have suffered from poor statistical performance and/or high computational complexity inappropriate for large data sets or real-time use.

Recent work based on wavelets, however, has provided a semi-parametric estimator for H which gives unbiased estimates together with significant computational advantages, notably a run time complexity of only $O(n)$. Details of this estimator are summarized in Section II, and can be found in [3], [4] (see also [5]–[7]). In [8] it was shown how these computational advantages can be exploited to allow H to be estimated in real-time simply, inexpensively, and with very low memory requirements. Section III describes the real-time implementation of the estimator, which is the subject of an Australian provisional patent (application number PP1692).

The aim of the present paper is to extend [8] to give a more complete account of the capabilities of the on-line estimator, by validating the claims made in [8] regarding its practicality and utility. In [8] the method was illustrated on a 10-Mb/s Ethernet (see Section V), and it was explained how the method scales to arbitrary size with respect to both memory and processing requirements, so that it will remain applicable as data rates increase. This claim is supported by the successful application to asynchronous transfer mode (ATM) traffic at 155 Mb/s (see [9]).

Another advantage highlighted in [10] is that a real-time estimator is not only useful in traditional real-time settings but also, by performing estimation at the point of measurement, radically reduces the volume of data that needs to be stored for off-line analysis. We illustrated some of the potential of this idea in Section VI by examining more than five months of Ethernet data to address the issue of diurnal or daily variation in H. New links between the variations in load and H are presented here for the first time. Finally, it is shown how not only the Hurst parameter but much more generally the variations in scaling behavior as observed in the *Logscale Diagram* can be studied in the real-time framework, and important time-varying features extracted. This paper therefore constitutes a synthesis of [8], [9], and [10], together with more detailed descriptions of the use of the method and the structure of Ethernet traffic.

II. Abry-Veitch (AV) Estimator

In any data measurement situation a basic theoretical framework is required through which to view the data, to select important parameters which describe it, and to propose and evaluate estimators of them. In our case the time-varying *rate* $x(t)$

Manuscript received January 8, 1999, revised June 9, 1999, and December 21, 1999; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor S. Floyd. This work was supported by Ericsson Australia.

M. Roughan and D. Veitch were with the Software Engineering Research Centre, RMIT University, Melbourne, Victoria 3053, Australia. They are now with EMUlab, an Ericsson-sponsored laboratory, Department of Electrical and Electronic Engineering, University of Melbourne, Victoria 3010, Australia (e-mail: m.roughan@ee.mu.oz.au; d.veitch@ee.mu.oz.au).

P. Abry is with CNRS URA 1325, Laboratoire de Physique, Ecole Normale Supérieure de Lyon 46, Lyon, Cedex 07, France (e-mail: pabry@physique.ens-lyon.fr).

Publisher Item Identifier S 1063-6692(00)06791-1.

of incoming traffic is the data of interest, and we model it as a stationary stochastic process. Basic features of this process are its mean $\mu_x = E[x]$, variance $\sigma_x^2 = E[(x-\mu_x)^2]$, and correlation function $\gamma_x(k) = E[(x(t+k)-\mu_x)(x(t)-\mu_x)]$. In this context the self-similar properties of traffic manifest themselves in a particular form of $\gamma_x(k)$, namely a decrease with lag k so slow that the sum of all correlations downstream from any given time instant is always appreciable, even if individually the correlations are small. The past therefore exerts a long-term influence on the future, exaggerating the impact of traffic variability and rendering statistical estimation problematic. This phenomenon is known as *long range dependence* (LRD), and is commonly defined by $\gamma_x(k) \sim c_\gamma |k|^{-(1-\alpha)}$, $\alpha \in (0,1)$, or equivalently as the power-law divergence at the origin of its power spectrum: $f_x(\nu) \sim c_f |\nu|^{-\alpha}, |\nu| \to 0$.

The Hurst parameter describes the (in practice asymptotic) self-similarity of the cumulative traffic process $\int_0^t x(s)\,ds$ while α describes the LRD rate process $x(t)$. It is nonetheless common practice to speak of H in relation to LRD via the relation $H = (1+\alpha)/2$, and we follow this convention here.

In [4], [11] a semi-parametric joint estimator of (α, c_f) is described based on the *discrete wavelet transform* (DWT). Wavelet transforms in general can be understood as a more flexible form of a Fourier transform, where $x(t)$ is transformed, not into a frequency domain, but into a time-scale wavelet domain. The sinusoidal functions of Fourier theory are replaced by wavelet basis functions $\psi_{a,t}(u) \equiv \psi_0((u-t)/a)/\sqrt{a}, a \in I\!R^+, t \in I\!R$ generated by simple translations and dilations of the *mother wavelet* ψ_0, a band pass function with limited spread in both time and frequency. The wavelet transform can thus be thought of as a method of simultaneously observing a time series at a full range of different scales a, whilst retaining the time dimension of the original data.

Multiresolution analysis theory shows that no information is lost if we sample the continuous wavelet coefficients at a sparse set of points in the time-scale plane known as the *dyadic grid*, defined by $(a,t) = (2^j, 2^j k), j, k \in I\!N$, leading to the DWT with discrete coefficients $d_x(j,k)$ known as *details*. Intuitively, the dyadic grid samples the wavelet domain at a resolution appropriate to the scale. Henceforth we will deal exclusively with the details of the DWT. The *octave* j is simply the base 2 logarithm of scale $a = 2^j$, and k plays the role of time (although a time whose rate varies with j). For finite data of length n, j will vary from $j = 1$, the finest scale in the data, up to some $j_{\max} \approx \log_2(n)$. The number of coefficients available at octave j is denoted by n_j, and approximately halves with each increase of j.

The estimator has excellent computational properties due to the fast "pyramidal" filter bank algorithm [12] for calculation of the discrete wavelet transform, which has a complexity of only $O(n)$. The number of wavelet coefficients $d_x(j,k)$ thus generated is also of order n, and subsequent computations required to form the estimate of H from them have only this complexity. The overall complexity therefore remains $O(n)$, which clearly scales satisfactorily.

The main feature of the wavelet approach which makes it so effective for the statistical analysis of scaling phenomena such as LRD is the fact that the wavelet basis functions themselves possess a scaling property, and therefore constitute an optimal "coordinate system" from which to view such phenomena. The main practical outcome is that the LRD in the time domain representation is reduced to residual **short** range correlation in the wavelet coefficient plane $\{j,k\}$, thus removing entirely the special estimation difficulties. Thus for each fixed j, the series $d_x(j,\cdot)$ can be regarded as a stationary process with weak short-range dependence, and these series can be regarded as independent of each other.

We can now outline the estimator as consisting of the following four stages:

1) **Wavelet decomposition:** A discrete wavelet transform of the data is performed, generating the details $d_x(j,k)$ over the dyadic grid.
2) **Detail variance estimation:** At each fixed octave j the details are squared and averaged across "time" k to produce μ_j, an excellent estimate of the variance of the details[1]. For LRD processes the μ_j follow a power-law in j with exponent α.
3) **Analysis using the Logscale Diagram:** From the plot of $y_j = \log_2(\mu_j)$ against j, the *Logscale Diagram*,[2] the scaling range (j_1, j_2) where scaling occurs (i.e., where the y_j fall on a straight line) is determined.[3]
4) **LRD parameters estimation:** The LRD parameters H and c_f [4] are extracted by performing a weighted linear regression[5] over the scaling region.[6]

Matlab source code for the AV estimate is available at http://www.emulab.ee.mu.oz.au/~darryl.

An example of the regression fit using a simulated data set is given in Fig. 1. The 95% confidence intervals for each y_j, shown as vertical lines at each octave j, are seen to increase with j. A plot such as this of y_j against j, complete with confidence intervals about the y_j, has been termed the *Logscale Diagram* [4], [7], and constitutes an effective starting point for the analysis of scaling phenomenon. The estimator, being semi-parametric, requires an analysis phase prior to estimation to determine the scaling range where alignment is observed in the Logscale Diagram (see [7] for further details on the reading of Logscale Diagrams).

III. On-Line Estimator

The AV estimator summarized above is gaining acceptance as the method of choice for measuring LRD in traffic [14]–[16], and wavelets are even being used to measure multifractal properties of traffic [17], [18]. Until now however, the AV estimator

[1]Since the expectations of the details are all identically zero μ_j [13], [12], the average of the squares of the details at a given j is an estimate of the variance at that j.

[2]In forming the Logscale Diagram small corrective terms $g(n_j)$ are in fact subtracted from $\log_2(\mu_j)$ to compute y_j to account for the fact that $E[\log](\cdot) \neq \log(E[\cdot])$.

[3]If the data is truly LRD then the upper cutoff scale j_2 should always be the largest possible given the length of the data, ie. $j_2 \approx \log_2(n)$, however scaling in a finite range is also observed in data [7].

[4]H is related to the slope of the plot, and c_f to a power of the intercept.

[5]The weights are functions of the known variances of the y_j and do **not** depend on the data.

[6]Confidence intervals for H are derived from the standard variance formulae for weighted linear regression with mutually independent y_j, and so again are **not** functions of the data.

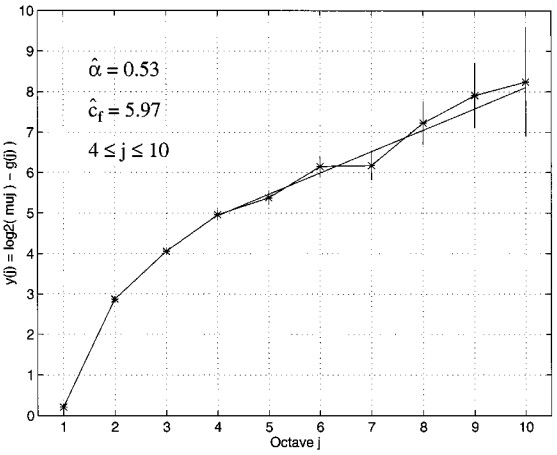

Fig. 1. Stages 3 and 4: estimation from the Logscale Diagram. An example of the y_j against j Logscale Diagram and regression line for a LRD process with strong SRD. The vertical bars at each octave give 95% confidence intervals for the y_j. The series is simulated $\mathrm{fARIMA}(0,d,2)$ with $d = 0.25$ $(\alpha = 0.50)$ and $\Psi = [-2,-1]$ implying $c_f = 6.38$. Selecting $(j_1, j_2) = (4, 10)$ allows an accurate estimation despite the strong SRD: $\hat{\alpha} = 0.53 \pm 0.07$, $\hat{c}_f = 6.0$ with 95% confidence interval $4.5 < \hat{c}_f < 7.8$.

has been used as a batch estimator, that is, where a data set is collected and analyzed off-line. It is ideally suited to on-line use however, making it usable within network elements such as switches as well as network monitoring systems. By on-line estimation we mean a data processing method whereby new fragments of data are processed as they arrive. In what follows we concentrate on the estimation of H, although the second LRD parameter c_f and indeed the entire Logscale Diagram (as illustrated in Section VI), is also provided by the method.

On-line estimation has two main requirements:

1) that an algorithm be devised such that newly acquired data elements can be processed individually and merged with existing processed data, rather than requiring complete re-computation;
2) that the algorithm be efficient enough to implement the above at the rate that new data arrives.

The first requirement is critical for on-line estimation, whereas the second is an issue of the necessary computing power versus its cost. Because of the steadily increasing bandwidth of networks however, the method must be scalable, so the second requirement is in fact principally an issue of the time and memory complexity of the algorithm.

The AV algorithm can be adapted to satisfy both requirements. The first stage of the estimator, the wavelet decomposition, is implemented in an on-line fashion using a pyramidal filter bank as shown in Fig. 2. Indeed, such filter banks were devised with on-line applications in mind. The second stage can be performed on-line as follows. Let the current stored sum of squares at octave j calculated from the first n_j values be $S_j = \sum_{k=1}^{n_j} d_x(j,k)^2$. Assume that the arrival of the new data point $x(n)$ results in a new coefficient $d_x(j, n_j + 1)$ at octave j from the filter bank. The sum is then updated as follows:

$$n_j \leftarrow n_j + 1,$$
$$S_j \leftarrow S_j + d_x(j, n_j)^2.$$

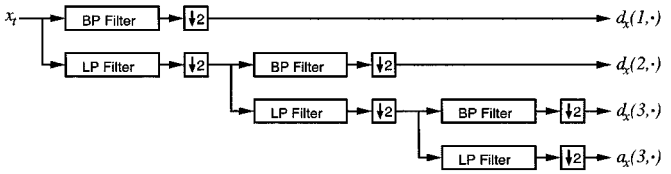

Fig. 2. Filter bank. At each level in the recursive structure, the bandpass (BP) output $d_x(j,\cdot)$ and the low pass (LP) output $a(j,\cdot)$ occur at half the rate of the input $a_x(j-1,\cdot)$.

When the variance estimate at octave j is required for the third stage it can be calculated as $\mu_j = S_j/n_j$.

The final two stages of the estimation algorithm need not be adapted to on-line versions, as there is no need to compute the Logscale Diagram or H every time a new data point is acquired. They may be recalculated only as needed, typically at "human" time-scales several orders of magnitude larger than the data collection rate. In any case the complexity of the final two stages is only $O(\log_2(n))$, with memory requirements of only $O(\log_2(n))$ also, and therefore they pose no computational difficulties. The great advantage of keeping stage 3 separate from stage 4 is that, at no extra computational cost, the nature of the data as a function of scale can be freely examined over all the scales available up to the current time, rather than simply assuming that the data is LRD and estimating blindly. The need to choose the scaling range by examining the Logscale Diagram can be automated when required to obviate the need for manual intervention in real-time applications. This could be done either by fixing values of (j_1, j_2) based on prior "off-line" studies, or by implementing heuristics which identify the scaling range automatically.

Some explanation is required to explain why the first stage of the on-line estimator is scalable. The on-line filter bank, illustrated in Fig. 2, consists of a number of filters of fixed size K connected in series (typically the size of these filters is small, say $K = 6$). Because the output rate of each filter is only half of its input rate, data of length n is effectively summarized and held in the filter banks in the form of $K\log_2(n)$ "half-processed" values. These numbers are the only ones which must be stored in memory, not the full set of historical input data $x(t)$. Regarding the run-time complexity, on average each new data point $x(n)$ results in $2(K+1)$ operations, a number independent of n. The maximum possible number of operations resulting from a single new data point scales as $O(\log_2(n))$, however this does not occur very frequently, and if problems of processor load arise the filter bank can be naturally implemented in digital signal processing (DSP) hardware.

Regarding the scalability of the second stage, the number of operations is $O(n)$, and the memory requirements are only $O(\log_2(n))$ as only the sums of squares S_j are kept, and *not* the full set of detail coefficients. The third and fourth stages are based on simple processing of the S_j, and so they scale satisfactorily with respect to both computation and memory.

Section V shows how a quite modest computer is capable of performing the AV estimation algorithm, on-line and in real-time on 10-Mb/s Ethernet data sampled every millisecond.

The obvious advantage of computing estimates on-line is that results are immediately available, rather than after a lengthy

cycle of collection and analysis. As mentioned earlier, this is essential for real-time network management purposes, but also offers important advantages for traffic collection and analysis in general. For example, apart from reducing the analysis delay, this approach allows the decision as to whether enough data has been collected to be made as it arrives. It is also advantageous to be able to detect unusual events as they occur, enabling immediate modifications to the collection/analysis effort.

The other central advantage of on-line estimation is the reduction in memory requirements, both in terms of the algorithm itself and of the storage of data sets. Batch analysis requires the collection and analysis of *very* large data sets, and samples larger than any standard computer's memory space are easy to collect. For example, a traditional Ethernet sampled every millisecond over one week represents 604 million sample points, which stored as 4-byte integers requires approximately 2.4 GB of space. Thus capture of this data may be a problem, as the data cannot all be stored in memory and then saved to disk. Similarly for analysis, the data cannot be held in memory all at the same time resulting in large delays due to disk paging. In contrast, as explained above, on-line measurement does not have substantial memory requirements. Thus a traffic stream can be monitored and measured continuously for weeks at a time, without any delay in the estimation at the end of the process, and without a large memory.

The number of scales available for estimation increases with the length n of the data. Ideally the number of available octaves is simply $j_{\max} = \log_2(n)$, however edge effects limit the number in practice. Fig. 3 shows the number of octaves in the data and the number of octaves actually available as functions of n, for the Daubechies3 wavelets (implying a filter length of six taps) that are used here.

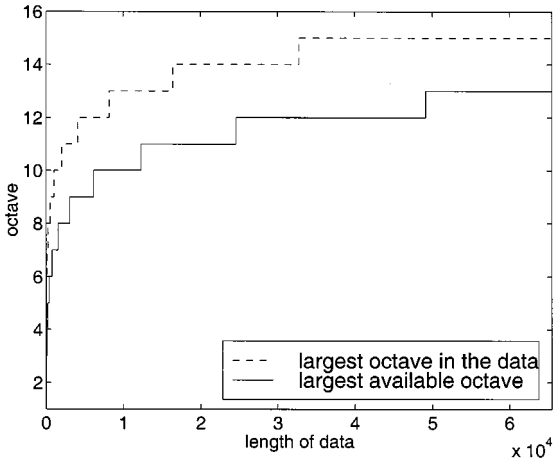

Fig. 3. Number of available scales as a function of n.

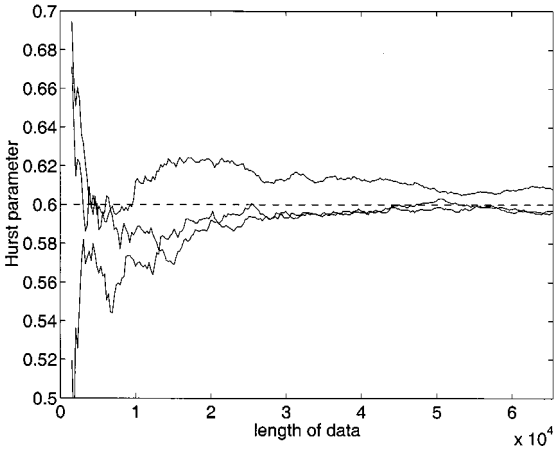

Fig. 4. Three example sample paths. The dashed line shows the true Hurst parameter while the solid lines show examples of the on-line Hurst parameter estimates.

IV. Performance

In a system designed to measure traffic on-line in real-time there are three key components: the packet capture process, the prefiltering process where the raw data is converted to a time series, and finally the on-line estimation itself. In this section we discuss and measure the performance of the on-line estimator only. The comments here are therefore valid regardless of the computational details of the lower two levels, which will vary according to the kind of network measured. The performance of the first two processes, and of all three together, is addressed in the next section in the context of Ethernet measurement.

The statistical performance of the batch joint AV estimator, and comparisons with other methods of estimating LRD parameters, have been described in detail elsewhere [4], [11] (for H only see also [5], [3], [6]). Briefly, the estimator offers excellent statistical performance: negligible bias, close to optimal variance, and robustness of various kinds including with respect to superimposed deterministic nonstationarities. It is not the aim of this paper to repeat these studies, but rather to illustrate the dependence of certain properties on data length, as the new feature of the on-line version is that the length of the data is constantly increasing.

The series used in this section were all realizations of the fractional Gaussian noise (fGn) process, precalculated using a standard spectral synthesis technique. In each case values from the series were piped to the on-line estimator one at a time, in order to simulate the arrival of raw measurements in real-time. Thus the estimator used here is identical to that used with the working on-line system described in the next section. The interval chosen between the actual estimations of H (stage 4 of the estimator) was every 2^8 data points. There is a warm up period at the beginning of the measurement run to wait for the octaves required for the analysis to become available (see Fig. 3).

For each of the values $H = 0.6, 0.7, 0.8$ and 0.9, 100 independent realizations of length $n = 2^{16}$ were generated. Fig. 4 shows three examples chosen randomly from the set with $H = 0.6$. The graph illustrates typical behavior of the estimator in time. Here we use prior knowledge of the fGn process to choose the lower end of scaling range to be $j_1 = 3$, and the upper end $j_2 = j_{\max}$ to be the largest octave available. A point of interest is that there is no immediate jump in accuracy when a new octave (scale) becomes available for use in the estimation. This is because when this occurs there are still relatively few data points at the new octave, and so the weighted regression gives little weight to it.

In Fig. 5, for each H, averages over the set of 100 realizations are plotted. The fact that the averaged estimates tend to the cor-

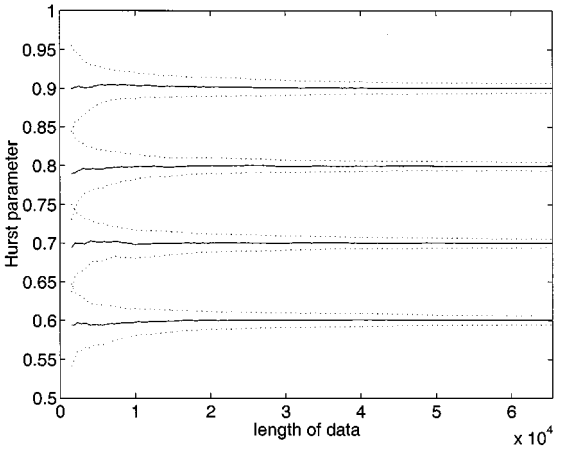

Fig. 5. Average (solid lines) of the estimates of the Hurst parameter, and the standard deviation around the average (dotted lines) for each set of 100 data sequences.

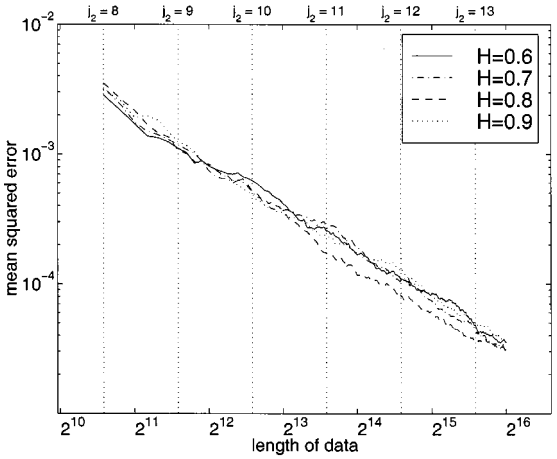

Fig. 6. MSE of the estimated Hurst parameter for each of the four sets of 100 sample paths.

rect values illustrates the lack of bias of the estimator. The speed of convergence to the correct value with increasing n is shown by the shrinking standard deviation of the 100 estimates shown to either side. These sample standard deviations constitute empirical estimates of the standard deviation of the estimator.

To further illustrate this convergence, Fig. 6 shows a log–log plot of the mean squared error (MSE) of the estimates as a function of n. The MSE corresponds closely to an empirical measure of the variance of the estimator, as we know the bias to be negligible. The fact that an approximately straight line is seen suggests that the variance of the estimator decreases as a power law. It is also noteworthy that the MSE seems to have very little dependence on H. Both of these facts are in agreement with the theoretical results of [4] which state that there is no H (nor c_f, nor μ_x, nor σ_x^2) dependence in the variance of the estimate of H, and that asymptotically the variance goes as

$$\mathrm{var}(\hat{H}(n)) \sim \frac{2^{j_1-3}}{\ln^2 2} n^{-1} \tag{1}$$

where j_1 is the smallest scale used in the estimation. The $1/n$ rate of decrease is a result typical of the variance of estimators in a **short** range dependent context (for example independent

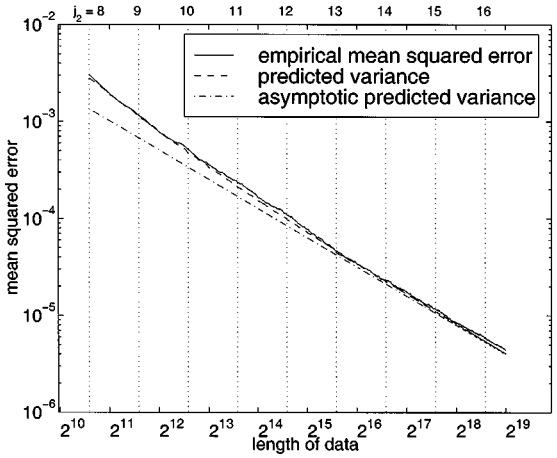

Fig. 7. Dependence of the MSE of the estimates on n.

random variables), and it is therefore particularly noteworthy: the AV estimator obtains short range dependent statistics from long-range dependent data. The hypotheses used to obtain the theoretical results are never exactly satisfied in practice however, not even for a "model" LRD process such as the fGn. We therefore repeated the test for $H = 0.6$, shown in Fig. 7, this time over 500 realizations each of length 2^{19}, in order to examine more closely the decay rate of the MSE. In the plot the MSE is compared to the asymptotic theoretical prediction given above, and the full theoretical variance prediction of [4]. Performing a linear regression on the MSEs in the plot leads to a slope of -1.04, which agrees with the predicted rate of n^{-1}, the minor discrepancy being easily accounted for by the asymptotic nature of the $1/n$ dependence together with statistical fluctuations in the MSE.

The computational performance of the estimator is also excellent. For instance we profiled the algorithm running on a 133-MHz Pentium PC (running `FreeBSD` and compiled with `gcc`), on test data 100 000 elements long (piped directly to the estimator), where estimates of the LRD parameters were made every 1000 data elements (100 estimations in total). The wavelet transformation and sum of squares computation consumed 1.8 s of CPU time, and the estimation 4.8 s. Thus the transform took on average 0.018 ms per data point, and the estimation 48 ms per estimate. Taking the most conservative view, it is unnecessary to perform the estimation more than once per second, implying a negligible impact from the estimation phase and therefore allowing sampling rates of 50 000 samples/s or better. Of course this will be reduced if competition from other processes is significant, notably from the packet capture process if it is running on the same machine.

V. Real-Time Estimation for Ethernet

We have explained in Section III how the on-line estimator is scalable, and in Section IV demonstrated the estimator using simulated on-line data. In this section we use the estimator to analyze 10-Mb/s Ethernet data on the local area network at the Software Engineering Research Centre (SERC) without the use of high performance hardware, proving that it is efficient enough to be used on real, low-cost systems.

An Ethernet was chosen for two reasons. First, it was the first type of data network where self-similar traffic was shown to exist [1]. Second, it is relatively easy to extract traffic from an Ethernet because of the broadcast nature of the medium. The Berkeley Packet Filter (BPF) [19], which is part of the kernel of `FreeBSD` (a variant of the `Unix` operating system suitable for Intel PC's), was used to capture and time stamp packets. The packet capture and the estimation algorithm were all run on a 133-MHz Pentium computer. Though the timestamping of the BPF on this system is not as accurate[7] as that obtained in the original Bellcore study [1], that same study showed that timestamping accuracy of the order of 1 ms (significantly worse than the accuracy obtained here) is quite sufficient to measure self-similarity.

The output from the BPF passes through a simple prefiltering program which generates a sequence of data values corresponding to the number of bytes transferred over the Ethernet during each sampling interval. This sequence is the raw data series $x(t)$ to be analyzed by the on-line estimator. In [16] the authors demonstrated that a sampling interval as large as 100 ms was adequate for measurements of self-similarity in real traffic. A sampling interval as fine as 1 ms posed no computational problems for the 133-MHz processor. At rates much faster than this the packet capture and prefiltering processes can fail to keep up with the data, whereas the load due to the estimation algorithm, as detailed in the previous section, is small at such rates. The bottleneck in this Ethernet measurement system is therefore in the lower layers, not the on-line estimation.

Fig. 8(a) shows Ethernet data originally sampled at 1 ms intervals and averaged over 2^{16} ms $\simeq 1$ minute intervals. Fig. 8(b) shows the on-line estimation for the same data, using $(j_1, j_2) = (8, j_{\max})$, based on visual inspection of Logscale Diagrams. The latter figure shows real-time output, where at each t the estimate is based on all the data in $[0, t]$. The graph therefore becomes smoother with increasing t as new data has proportionally less impact on the growing weight of past data. It is important to note that such long-term estimates are not usually meaningful, due to nonstationarity in the data. The intent of these figures is not to give useful H estimates, and certainly not to demonstrate self-similarity in Ethernet traffic as this has been demonstrated many times before. Rather the results are intended to show that the algorithm is efficient and robust enough to apply to real data, in real time.

We mention briefly here that we have extended our on-line monitoring efforts to 155 Mb/s ATM traffic, also on inexpensive hardware. In this system the on-line estimation is identical to that described here, only the lower layers have changed. The details of the ATM traffic monitor are fully described in [9]. The lower layers of the system are based around the OC3MON [20] which has been used successfully to monitor the vBNS[8] as part of the CORAL project [21]–[23].[9]

[7]The exact accuracy of timestamps on a time-sharing system depends on what else is running on the system.

[8]Very high performance Backbone Network Service. Online. Available: http://www/vbns.net/

[9]Coral Network Traffic Analysis. Online. Available: http://moat.nlanr.net/Coral/

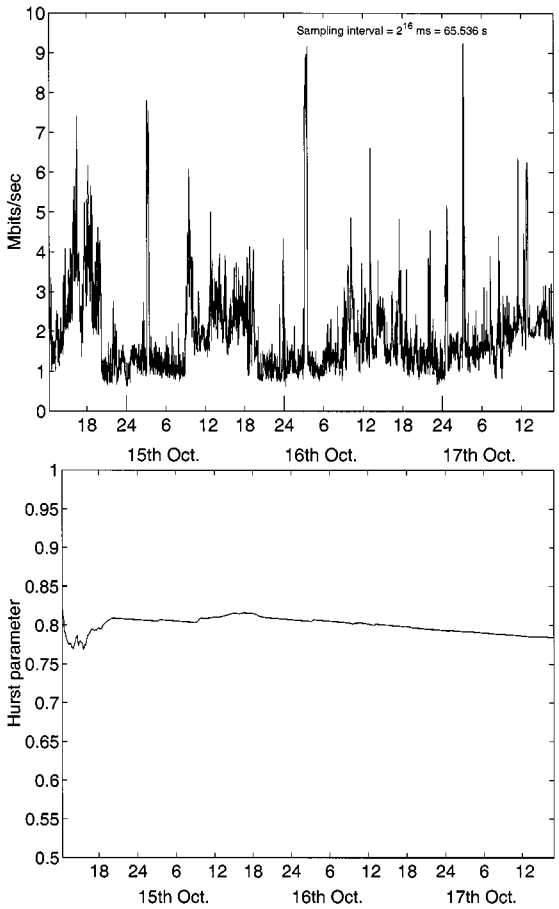

Fig. 8. SERC local Ethernet data, October, 1997. (a) Ethernet byte data averaged over approximately one-minute intervals. (b) Corresponding on-line estimates.

VI. Long-Term Measurements

One of the major advantages of on-line measurement is that measurements can be made over long periods. This is possible because the data is not collected, but analyzed as it arrives, enabling it to be stored in a far more compact and useful form. Thus the most onerous parts of "off-line" analysis, the data collection and initial processing, can be performed in real-time, leaving only the higher level aspects, for example the examination of Logscale Diagrams, as truly off-line. This "fast off-line" (FOL) analysis method is illustrated in this section.

Note that not collecting data is also a disadvantage in that new questions involving analysis outside of the standard suite cannot be answered. For example, we are unable to return to the original data to repeat our analysis on a per-application basis. It is necessary in real-time analysis to know in advance all of the questions of interest, and to have on-line algorithms for each.

As part of an ongoing data collection effort at SERC, Ethernet data from the local network was collected from March 4th to August 24th, 1998. A major reconfiguration of the local network at the end of this period was a natural breakpoint motivating a study, reported in [10], whose aims were twofold: first, to illustrate the benefits of FOL analysis by performing an analysis which would not have been possible using ordinary off-line methods without great effort; and second, to begin to investigate the important practical question of

the diurnal or daily variation of the Hurst parameter. Here we summarize the results of [10] and present additional data on the link between load and H. The data was collected by running a set of monitors almost continuously over the period (small gaps occurred as the monitors were also used for other purposes). Note also that the network size and configuration was significantly different from the previous October 1997 data (Fig. 8) during the long-term study.

In current models of traffic, the parameter H is a constant describing the scaling nature of arriving traffic which is deemed to be *stationary*, that is, constant in time in the statistical sense. Naturally in real data this assumption will hold only approximately, and in some cases not at all (for instance see [24]). For example diurnal variation in load is a recognized feature of traffic in most contexts (for examples see [25], [26], [23]), is this also true of H? In the seminal paper [1] the authors speculate that this is so, and further that there is a correlation between the load and H.

We performed monitoring separately over blocks of data 1, 4, and 24 hours long, with the intention of studying the diurnal and weekly variations of the load and Hurst parameter, and their correlations. It is important to bear in mind that with each measurement, the implicit assumption made is that H is well defined and constant over the block. If this is not the case then **no** estimator can return a meaningful single value. For example if measurements over 24 contiguous one-hour blocks reveal that H changes, then we know that although a measurement over the entire 24-hour period may still be a valuable summary of scaling information, it cannot be taken as a meaningful estimate of a constant H.

On the other hand, if H *is* well defined (and constant) over a block, then we wish to measure it robustly, that is, despite the possible presence of nonstationarities in other aspects of the data, such as the mean. Although such nonstationarities can cause problems for standard estimators of the Hurst parameter [27], the AV estimator is remarkably robust to changes in the mean or variance of a process [5], [28], and therefore we believe that, provided the scaling exponent is itself well defined (see [28], [24] for further discussions on this issue), reasonable estimates can be made even in the presence of common nonstationarities, for example, slow periodic changes in the mean. There are circumstances where the estimator can fail. These include when there are *very* large and sharp jumps in mean, the presence of overwhelming additive noise masking the LRD component, or of power-law trends whose exponent corresponds to an H larger than that of the data. There is no evidence for nonstationarities of these types here.

Fig. 9 gives an example of Hurst parameter estimates over the three different time-scales of 1, 4, and 24 hours during a week in April. The graph shows considerable variation in the estimates over this time period. In fact, we can immediately conclude that H is not constant, as the theoretical confidence interval [assuming constant H and Gaussian data, see (1)] for each of the 24-hour estimates (not shown on the figure) is orders of magnitude smaller than the variation observed. The confidence intervals corresponding to the four and one-hour measurements are also very small, however the data is highly non-Gaussian at 1 ms resolution, and we must be careful about interpreting the

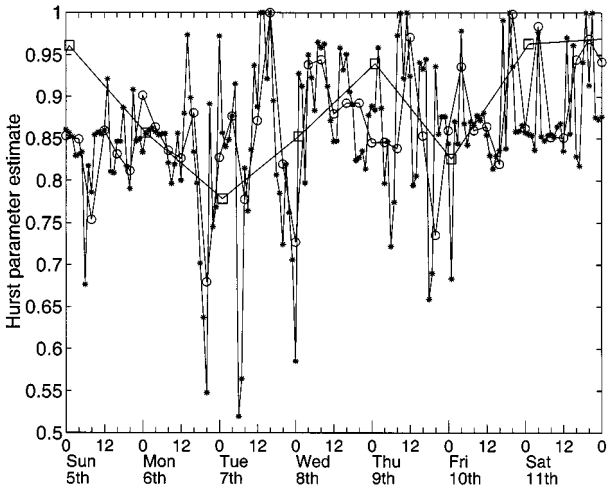

Fig. 9. Example sample paths for Hurst parameter estimates. The three curves are based on 1 ($*$), 4 ($\circ$), and 24-hour blocks of data ($\square$). Start time = 05-Apr-1998 00:00:00.

Gaussian based confidence interval too strictly. Nonetheless, although some of the large variation in one-hour estimates can certainly be attributed to statistical fluctuation, the broad correspondence between the estimates over the three time-scales suggests that the variation reflects real changes below the 24-hour time scale. Furthermore the four-hour measurements seem to be more consistent with the one-hour measurements than the 24-hour are with the four-hour (though still with high variation), hinting that H may be taken as reasonably constant over intervals close to, but below four hours. It is also noteworthy that there is no obvious correlation between the load estimates over the same time-scales (not shown) and the Hurst parameter estimates.

How could such variations in H occur? One way is suggested by [29], [30], in which the authors demonstrated that LRD in aggregated traffic may arise from the high variability of the individual ON/OFF sources of which it is composed. However when one measures the source of this variability in individual applications such as WWW traffic [31], SS7 traffic [32], or Unix file sizes [33], one finds that they have different degrees of variability. In theory, for aggregate traffic formed by the mix of a set of applications, the most variable source will eventually dominate, and determine the asymptotic properties of the aggregate (including H). However, in practice, for finite data series, some applications may not make an impact on the measurements. For instance, a source running at very low rate might remain in the OFF state for the entire period of measurement. Thus the effective Hurst parameter which we measure may be dominated by a specific group of applications. This group may change significantly over time due to nonstationarities in the mix of applications, leading to a variable Hurst parameter estimate. Despite these difficulties, the Hurst parameter may still be a viable and important parameter for modeling the aggregate traffic.

Given that H may vary, the next question we ask is whether it varies with time of day. Before considering this question we examine the diurnal behavior of the load in our network. Fig. 10(a) and (b) displays estimates of the average load

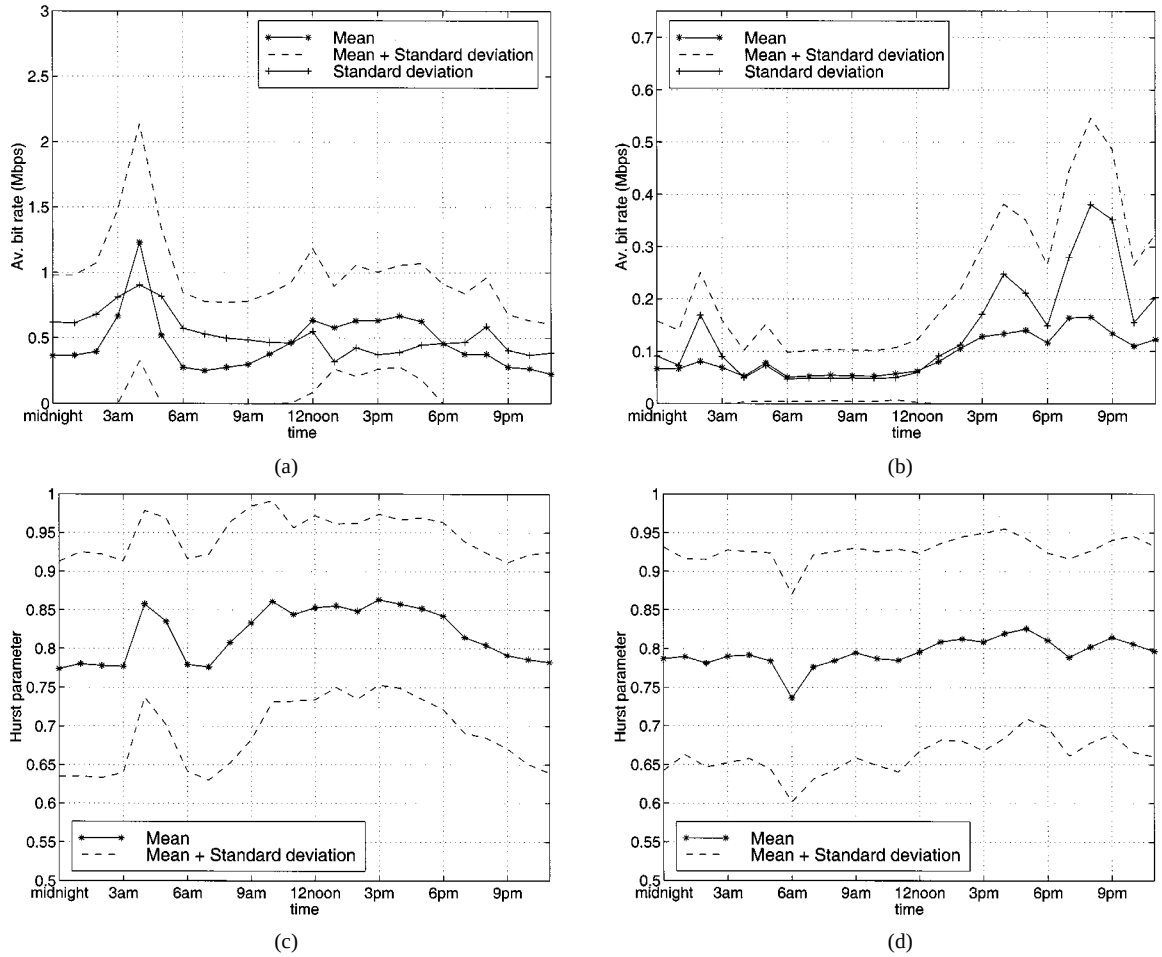

Fig. 10. Diurnal cycle in load and Hurst parameter estimates. The results plotted for each hour are averages over many days. (a) Weekday load. (b) Weekend load. (c) Weekday Hurst parameter. (d) Weekend Hurst parameter.

from March 4th to August 24th, 1998, during each one-hour period of the day. Fig. 10(a) shows the weekday load, while Fig. 10(b) shows the weekend load[10] —the two are substantially different, and it does not seem appropriate to combine them. The figure also shows the empirical standard deviation of the load about the mean, which reflects the range of truly different values of load observed in a given hour of the day over different days, not the variation of the load within a particular hour. The implicit assumption behind these graphs is that there is stationarity on weekly time-scales, so that meaningful averages can be taken to reduce the variability of the measurement at each hour, allowing the diurnal variation to emerge from the background variation.

The first notable feature of the mean load during the week is a weak busy cycle—the load increases during the day and decreases during the night—which we will refer to as the *user busy cycle*. The peak of this cycle appears to occur at 4 pm. The second notable feature is a large peak early in the morning, which results from the nightly backups. These backups start at $\sim$3 am each morning from Tuesday to Saturday.

We refer to the user busy cycle as weak because its magnitude is not large compared with the natural variation during the day, and between days. Regarding the connection between the average diurnal variation of Fig. 10(a) and (b) and the size of daily fluctuations, note in Fig. 10(a) that the standard deviation of the results does not appear to correlate well with the user busy cycle, but that it does appear to be correlated with the backup peak.

The traffic on the weekends is significantly lower than during the week—not surprisingly. There is also a user busy cycle during the weekends which has a later peak, around 7 or 8 pm. The weekend cycle is also more variable (with respect to the load). However we shall see presently, when considering the Logscale Diagrams of Fig. 11, that this low load coincides with a breakdown of scaling behavior at high scales.

Fig. 10(c) and (d) show the equivalent picture for Hurst parameter estimates. The standard deviation itself has not been plotted in the results, as it remains roughly constant with a value slightly larger than 0.1, but we do show the average $\pm$ the standard deviation.

Fig. 10(d) seems to indicate that the weekend traffic has a variable Hurst parameter, but that it does not have a strong systematic dependence on the time of day (except for a possible

[10]Backups on our system begin at approximately 3 am and may extend past 6 am after weekdays and therefore occur on Saturday morning but not Sunday or Monday morning. We consider the backups to be part of the weekday workload, and hence we have adjusted the measured begining and end of the weekend to 7 am on Saturday morning and 7 am on Monday morning respectively.

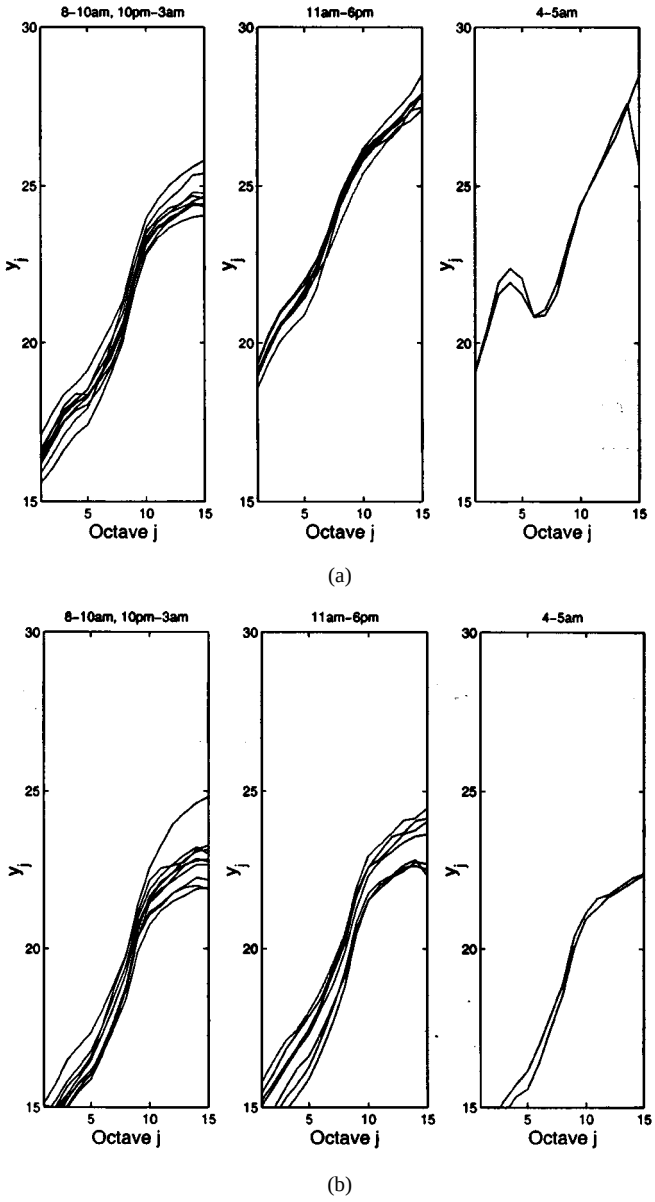

Fig. 11. Diurnal cycle in Logscale Diagrams. (a) Three classes of LD's can be found during weekdays, corresponding to (from left to right) low load, high load and backup periods. (b) On weekends the shape of the LD is roughly constant.

blip at 6 am). Fig. 10(c) seems to indicate that during the week the Hurst parameter does have some dependence on time of day—though it is not very strong. In comparison, two standard deviations around the estimate covers almost the entire range of values from 0.5 to 1.0 indicating that at any time of day, any of the range of possible values of H can occur.[11] Thus averaging over several months was needed to render this daily cycle visible.

There is a peak at about 4 am which appears to be strongly correlated with the backup peak, and also a "Hurst user busy cycle" which follows the user busy cycle fairly closely: the Hurst parameter seems to be connected to the network load. Interestingly, the peak due to the backups coincides with the backup load peak, while the busy cycle behavior of the Hurst parameter seems to begin earlier, and persist for longer than that of the load. It is plausible that this behavior relates to the type of traffic present, for instance the applications being used (as discussed above). To investigate this further new studies must be conducted partitioning the data by application—unfortunately on-line analysis does not allow one to reprocess the data to extract new information if this was not obtained in the initial analysis. It is the central feature of FOL analysis that it discards all the original data, and though this is an advantage from most points of view, it prevents re-examination of the original data.

In the above estimates of H the full range of scales were used to estimate H, that is $j_1 = 1$, and j_2 as large as possible, rather than selecting the scales after examination of the Logscale Diagram. As a result the estimates should be thought of as a rough measure of the behavior of the data as a function of scale, rather than unbiased estimates of the scaling parameter of LRD. This choice was motivated by the fact that the significant nonstationarities observed in the data precluded the choice of universally valid fixed values of $[j_1, j_2]$, and automated selection is beyond the scope of this paper.

In fact it is necessary to examine the scaling behavior in detail, and not only to take measurements of the Hurst parameter over some fixed scale range. We therefore investigated the time variation of the Logscale Diagram itself. Again differences between weekdays and weekends were found, and diurnal variations within these two. In Fig. 11 hourly Logscale Diagrams, again averaged over different days, are grouped into three classes corresponding to different time periods—low load times (8 am–10 am and 10 pm–3 am), high load times (11 am–6 pm) and times when backup load dominates the network (4 am–5 am). During the weekdays each class is characterized by a qualitatively different shape whereas during the weekend there is little qualitative variation. Although during high load periods the curves are approximately straight lines which means that the Hurst parameter estimates give useful information both at large and small scales, at low load periods the curves are far from straight and the meaning and usefulness of the estimates can be called into question.

Since during the busy hours of the day the workload is dominated by applications controlled by human interaction, whereas at night, and times of low load such as the weekend, applications controlled by automatic computer interaction dominate, we speculate (as did the authors of [1]) that human interaction plays an important role in the nature of the scaling behavior, rather than load as such. This would also explain how the user busy cycle is not identical to the Hurst busy cycle. Finally, we note that when backup load dominates the network the Logscale Diagrams shows a peak at octave 4—implying that the dominant phenonomena in the backup load occurs at time scales $\sim$16 ms—not a surprising conclusion given the machine-generated nature of this traffic.

The authors of [1] also noted an apparent correlation between load and H, and so the next question we ask is "Given a much longer series of data can we deduce a simple relationship between the two?" Fig. 12 illustrates the correlations between the load and the Hurst parameter by plotting them against each other. The plots are based on the hourly load and

[11]Values near 1 may be artifacts of nonstationarity, but not necessarily. The estimator can take any real value, even when H is well defined and in $(0.5, 1)$.

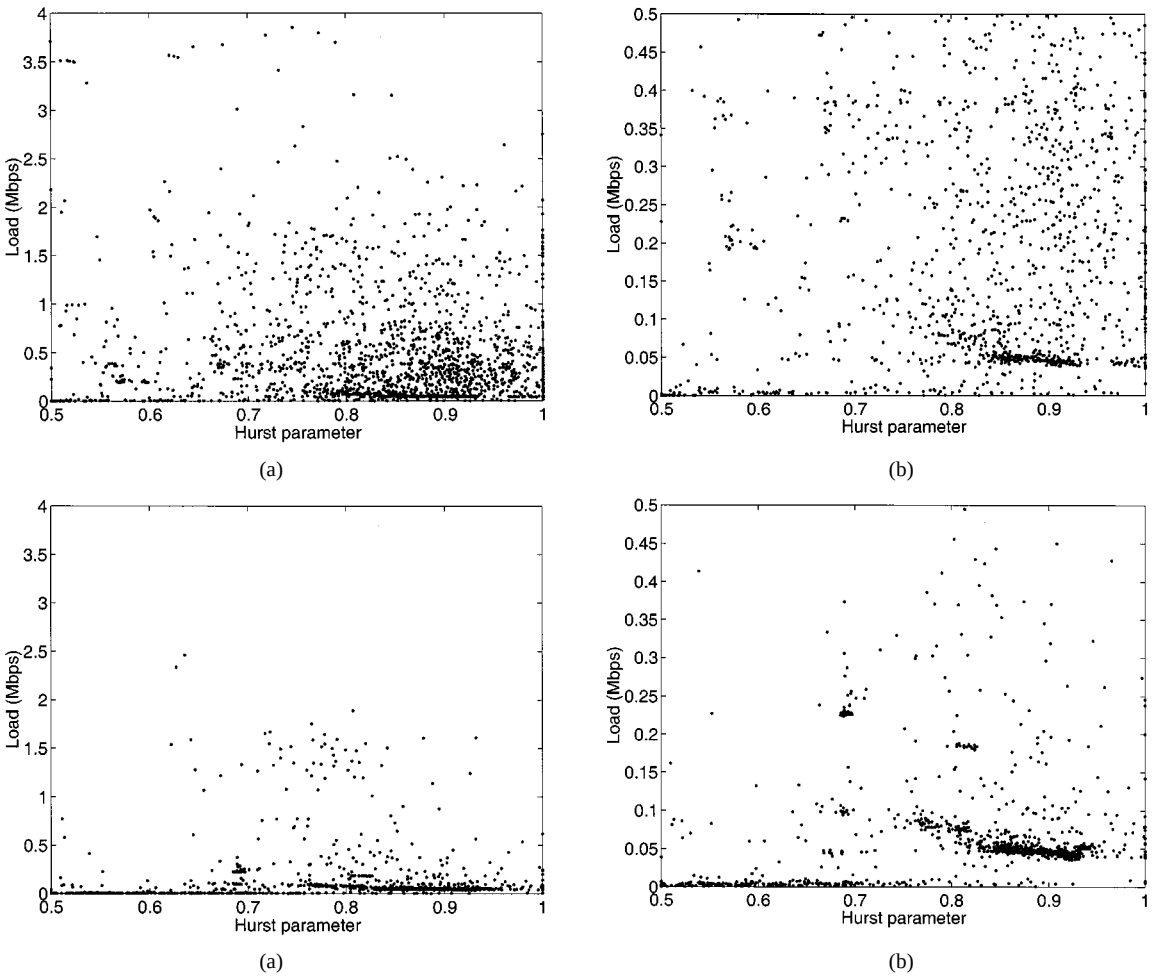

Fig. 12. Correlation between the Hurst parameter and load. (a) Weekday correlations. (b) Zoom of (a). (c) Weekend correlations. (d) Zoom of (c).

Hurst parameter measurements. Again the data is divided into weekday and weekend data. Plots (b) and (d) show closeups of regions of plots (a) and (c) respectively. It is clearly seen that there are correlations between load and the Hurst parameter, but that they are not at all simple. For instance plot (d) shows clear clustering, though at very low loads. This could be due to a particular type of traffic dominating during such time periods. Plot (b) appears also to have clustering, though it is not as obvious—this could simply be because during the week, the type of traffic which generates such clusters cannot so easily dominate the other forms of traffic. More work is required to determine appropriate classes of models to account for such observations.

There are a great many things remaining to study in this data, however this is not the intention of this report, and the study is ongoing. Rather it is intended to provide a taste of the possibilities created by the cheap, ubiquitous monitoring allowed by an on-line estimator.

VII. Conclusion

We have shown that the Abry-Veitch estimator for the measurement of the parameters of long-range dependence, including the Hurst parameter, can be successfully applied on-line, in real-time, enabling their use in real-time applications such as measurement-based admission control (for instance see [34]). Furthermore, the immediate analysis of data at the point of measurement avoids the storage of huge data sets for off-line analysis. The scalability of the method was demonstrated both with respect to memory requirements, which are very modest, and processing complexity. The algorithm's performance was demonstrated by applying it to simulated on-line data, and found to be excellent and in agreement with theoretical results. The algorithm was also demonstrated in a working system using a modest PC to make real-time measurements of both Ethernet traffic and ATM traffic. Thus the method is efficient enough to deal with high data rates on inexpensive hardware. If, as network speed increases, a point is reached where processing requirements exceed the capacity of the processor chips available at the time, the algorithm could be implemented using DSP hardware, to which it is ideally suited. Such a solution would be able to cope with any data rates currently envisaged with room to spare.

Finally we illustrated some of the possibilities opened up by the on-line estimator in its role as a fast off-line analysis method. It was used to simply and easily analyze the local area network at SERC over a five-month period to assess diurnal variability in the Hurst parameter, as well the behavior of the data as a function of scale via diurnal changes in the Logscale Diagram.

An interesting novel observation was that the scaling properties could be ordered into a small number of groups which appear to be correlated with the traffic load, and we speculate by the origin—human or machine—of the traffic. Another key observation was that the Hurst parameter does vary with time, and time of day, but the variations in individual days greatly outweigh the diurnal variation measured by averaging over many days.

There is much scope for future work, notably:

1) use of smoothing, perhaps using a Kalman filter, in order to discard old data, which might be more appropriate than windowing as used here;
2) use of an adaptive choice of the scaling range (j_1, j_2);
3) further study of the data presented in Section VI, notably with respect to the connection between load and the form of the Logscale Diagram;
4) collection and study of a substantial set of ATM data;
5) collection of data by application type in order to study the origins of the correlations between load and Hurst parameter;
6) application of the statistical test for the constancy of scaling parameters to better judge when the exponent changes [26];
7) application to call admission control and congestion control.

REFERENCES

[1] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of ethernet traffic (extended version)," *IEEE/ACM Trans. Networking*, vol. 2, no. 1, pp. 1–15, Feb. 1994.

[2] V. Paxson and S. Floyd, "Wide-area traffic: The failure of poisson modeling," *IEEE/ACM Trans. Networking*, vol. 3, no. 3, pp. 226–244, 1994.

[3] P. Abry, P. Gonçalvès, and P. Flandrin, "Wavelets and Statistics," in *Lecture Notes in Statistics*. New York, NY: Springer-Verlag, 1995, vol. 105, ch. Wavelets, spectrum estimation, 1/f processes, pp. 15–30.

[4] D. Veitch and P. Abry, "A wavelet based joint estimator of the parameters of long-range dependence," *IEEE Trans. Inform. Theory–Special Issue on Multiscale Statistical Signal Analysis and Its Applications*, vol. 45, no. 3, Apr. 1999.

[5] P. Abry and D. Veitch, "Wavelet analysis of long-range dependent traffic," *IEEE Trans. Info. Theory*, vol. 44, no. 1, pp. 2–15, 1998.

[6] P. Abry, D. Veitch, and P. Flandrin, "Long-range dependence: Revisiting aggregation with wavelets," *J. Time Series Analysis*, vol. 19, no. 3, pp. 253–266, 1998.

[7] P. Abry, P. Flandrin, M. S. Taqqu, and D. Veitch, "Wavelets for the analysis, estimation and synthesis of scaling data," Self Similar Network Traffic Analysis and Performance Evaluation, 1999, submitted for publication.

[8] M. Roughan, D. Veitch, and P. Abry, "On-line estimation of parameters of long-range dependence," in *IEEE GLOBECOM'98*, Sydney, Australia, Nov. 1998, pp. 3716–3721.

[9] M. Roughan, D. Veitch, M. Ahsberg, H. Elgelid, M. Rumsewicz, M. Castro, and M. Dwyer, "Real-time measurement of long-range dependence in ATM networks," presented at the PAM2000 Workshop on Passive and Active Networking, New Zealand, 2000.

[10] M. Roughan and D. Veitch, "A study of the daily variation in the self-similarity of real data traffic," in *Proc. 16th Int. Teletraffic Congress (ITC 16)*, vol. 3b, P. Key and D. Smith, Eds., Amsterdam, The Netherlands, 1999, pp. 67–76.

[11] D. Veitch and P. Abry, "Estimation conjointe en ondelettes des paramètres du phénomène de dépendance longue," in *Proc. 16ième Colloque GRETSI*, Grenoble, France, 1997, pp. 1451–1454.

[12] I. Daubechies, *Ten Lectures on Wavelets*. Philadelphia, PA: SIAM, 1992.

[13] S. Mallat, *A Wavelet Tour of Signal Processing*. New York, NY: Academic, 1998.

[14] A. Feldmann, A. C. Gilbert, W. Willinger, and T. G. Kurtz, "The changing nature of network traffic: Scaling phenomena," *Comput. Commun. Rev.*, vol. 28, no. 2, 1998.

[15] J. L. Jerkins and J. L. Wang, "A measurement analysis of ATM cell-level aggregate traffic," presented at the IEEE GLOBECOM'97.

[16] J. L. Jerkins, A. L. Neidhardt, J. L. Wang, and A. Erramilli, "Operations measurement for engineering support of high-speed networks with self-similar traffic," in *Proc. 16th Int. Teletraffic Congress (ITC 16)*, vol. 3b, P. Key and D. Smith, Eds., Amsterdam, The Netherlands, 1999, pp. 895–906.

[17] A. Feldmann, A. C. Gilbert, and W. Willinger, "Data networks as cascades: Explaining the multifractal nature of internet WAN traffic," presented at the ACM/SIGCOMM'98, Vancouver, B.C., Canada, 1998.

[18] R. H. Riedi, M. S. Crouse, V. J. Ribeiro, and R. G. Baraniuk, "Multifractal wavelet model with application to TCP network traffic," *Inform. Theory*, no. 45, pp. 992–1018, Apr. 1999.

[19] G. R. Wright and W. R. Stevens, *TCP/IP Illustrated*. Reading, MA: Addison-Wesley, 1995, vol. 2, ch. 31.

[20] J. Apisdorf, K. Claffy, K. Thompson, and R. Wilder, "OC3MON: Flexible, affordable, high performance statistics collection," presented at the INET'97 Conf., June 1997.

[21] J. Dugan. Coral—Flexible, Affordable, High Performance Network Statistics Collection. [Online]. Available: http://www.caida.org/Tools/Coral/

[22] G. J. Miller, K. Thompson, and R. Wilder, "Performance measurement on the vBNS," in *Proc. Interop'98 Engineering Conf.*, Las Vegas, NV, May 1998.

[23] K. Thompson, G. J. Miller, and R. Wilder. (1997) Wide-area Internet traffic patterns and characteristics. Extended Version. IEEE Networks. [Online]. Available: http://www.vbns.net/presentations/papers/index.html

[24] D. Veitch and P. Abry, A Statistical Test for the Time Constancy of Scaling Exponents. preprint.

[25] Engineering and Operations in the Bell System, R. F. Rey, Ed., AT&T Bell Labs, ch. 5, 1983.

[26] J. F. Shoch and J. A. Hupp, "Measured performance of an Ethernet local network," *Comm. ACM*, vol. 23, no. 12, pp. 711–721, 1980.

[27] S. Molnár, A. Vidács, and A. A. Nilson, "Bottlenecks on the way toward fractal characterization of network traffic: Estimation and interpretation of the Hurst parameter," presented at the Int. Conf. Performance and Management of Complex Communications Networks (PMCCN'97), Tsukuba, Japan, Nov. 1997.

[28] M. Roughan and D. Veitch, "Measuring long-range dependence under changing traffic conditions," in *Proc. IEEE INFOCOM'99*, New York, NY, Mar. 1999.

[29] W. Willinger, M. S. Taqqu, R. Sherman, and D. V. Wilson. Self-similarity through high-variability: Statistical analysis of ethernet LAN traffic at the source level. presented at ACM/SIGCOMM'95. [Online]. Available: http://www.acm.org/sigcomm/sigcomm95/sigcpapers.html

[30] F. Brichet, J. Roberts, A. Simonian, and D. Veitch, "Heavy traffic analysis of a storage model with long range dependent on/off sources," *Queueing Systems*, vol. 23, pp. 197–225, 1996.

[31] M. E. Crovella and A. Bestavros, "Self-similarity in world wide web traffic: Evidence and possible causes," *IEEE/ACM Trans. Networking*, vol. 5, no. 6, Dec. 1997.

[32] D. E. Duffy, A. A. McIntosh, M. Rosenstein, and W. Willinger, "Statistical analysis of CCSN/SS7 traffic data from working CCS subnetworks," *IEEE J. Select. Areas Commun.*, vol. 12, no. 3, Apr. 1994.

[33] G. Irlam. (1998, June) Unix File Size Survey—1993. [Online]. Available: http://www.base.com/gordoni/ufs93.html

[34] A. Erramilli and J. L. Wang, "A connection admission control algorithm for self-similar traffic," presented at the IEEE GLOBECOM'99.

Matthew Roughan (M'97) received the B.Sc. with Honors and the Ph.D. degrees in applied mathematics from the University of Adelaide, Australia, in 1990 and 1994, respectively.

From 1994 to 1996 he worked at the the Cooperative Research Centre for Sensor Signal and Information Processing, Adelaide. From 1997 to 1999 he worked at the Ericsson/RMIT Software Engineering Research Centre, Melbourne, Australia. He now works as a Senior Research Fellow in the Ericsson funded EMUlab, Department of Electrical and Electronic Engineering, University of Melbourne, Australia.

Darryl Veitch was born in Melbourne, Australia in 1963. He received the B.S. with Honors degree at Monash University, Melbourne, in 1985, and the Ph.D. degree in mathematics in dynamical systems at the University of Cambridge, U.K., in 1990.

In 1991 he joined the research laboratories of Telecom Australia (Telstra), Melbourne, where he became interested in long-range dependence as a property of tele-traffic in packet networks. In 1994 he left Telstra to pursue the study of this phenomenon at the CNET, Paris, France (France Telecom). He then held visiting positions at the KTH, Stockholm, Sweden, INRIA, Sophia Antipolis, France, and Bellcore, Morristown, NJ, before taking up a three-year position as Senior Research Fellow at RMIT, Melbourne. He is now a Senior Research Fellow at the Ericsson funded EMUlab, Department of Electrical and Electronic Engineering, University of Melbourne. His research interests include scaling models of packet traffic, parameter estimation problems and queueing theory in a long range dependent context, and the statistical and dynamic nature of Internet traffic.

Patrice Abry was born in Bourg-en-Bresse, France, in 1966. He received the degree of Professeur-Agrege de Sciences Physiques in 1989 at Ecole Normale Supérieure de Cachan and received the Ph.D. degree in physics and signal processing at Ecole Normale Supérieure de Lyon and Université Claude-Bernard Lyon I in 1994.

Since October, 1995, he has been a permanent CNRS Researcher at the Physics Laboratory, Ecole Normale Supérieure de Lyon. He is the author of the book *Ondelettes et Turbulences—Multirésolution, algorithmes de décompositions, invariance d'échelle et signaux de pression*, Paris, France: Diderot, 1997. His research interests include theoretical developments of wavelet decompositions and time-scale analysis, studies of the scale invariance phenomenon and related topics (self-similarity, fractal, $1/f$ processes, long-range dependence, and local regularity of processes). The applications of current interest are physics of disordered systems (turbulence, chaos, etc.) and analysis and modeling of packet traffic.

Dr. Abry received the AFCET-MESR-CNRS prize for best Ph.D. in signal processing for the years 1993–1994.