Real-Time Measurement of Long-Range Dependence in ATM Networks

Matthew Roughan[1], **Darryl Veitch**[1], **Jennifer Yates**[2] **Martin Ahsberg**[1], **Hans Elgelid**[1], **Maurice Castro**[3], **Michael Dwyer**[1] and **Patrice Abry**[4],

1 - Department of Electrical Engineering, University of Melbourne, Victoria 3010, Australia.
E-mail: {m.roughan,d.veitch,m.dwyer}@ee.mu.oz.au

2 - AT&T Research Laboratories, 180 Park Avenue, P.O. Box 971, Florham Park, NJ 07932-0000, USA
E-mail: jyates@research.att.com

3 - SERC, RMIT University, Level 3, 110 Victoria St, Carlton, Vic 3053, Australia
E-mail: maurice@serc.rmit.edu.au

4 - CNRS UMR 5672, Laboratoire de Physique, 46, allee d'Italie, Ecole Normale Superieure de Lyon, 69364 Lyon, France.
E-mail: Patrice.Abry@ens-lyon.fr

Abstract— **We demonstrate the application of the on-line version of the Abry-Veitch wavelet based estimator of the Hurst parameter to ATM traffic carried over an OC3 link at 155 Mbps. The estimator 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.**

I. Introduction

Traffic measurements are vital to the provisioning of reliable, efficient Quality of Service (QoS) over data networks. Such measurements are useful for generating reference loads for testing equipment, and are a necessary ingredient in any application which tries to optimise network performance, from network planning and dimensioning tools to call admission control algorithms. Recent measurements of data traffic, for instance [1], [2], have discovered its fractal, self-similar, or long-range dependent nature – the traffic shows burstiness on many times scales: from milliseconds up to hours, or days. This behaviour has been proven to have important impacts on the performance of systems, however, until recently it has been prohibitively expensive to obtain measurements of these characteristics, preventing their use in day-to-day operations.

To gain a true understanding of the traffic carried on a network, detailed continuous measurements should be taken at many places in the network – ideally at any point where congestion might occur. Of course the volume of data generated through ubiquitous monitoring can be vast (consider a few seconds worth of data running at Gigabit/s speeds...), and so in-situ processing is greatly desirable to reduce the data to a managable and easily interpreted form. Such processing should be performed on-line and in real-time in order that network measurements be continuously available, and must be made with inexpensive equipment if it is to be almost ubiquitous. If this can be achieved then these results can be used in real-time applications such as Call Admission Control (CAC), rate adaptation, or congestion control.

The NLANR, in co-ordination with MCI, as part of the CORAL project [3], [4], [5], [6], [7], has developed a suite of OCXmon(itors), which allow non-invasive measurement of optical links (OC3 - 155 Mbs, OC12 - 622 Mps) carrying ATM traffic. We have adapted their work to allow the measurement of the fractal characteristics of traffic in real-time, as the data is collected.

The core of our work is based on the Abry-Veitch estimator of the parameters of Long-Range Dependence (LRD) which uses the wavelet transform to measure the degree of self-similarity in the traffic (details of which are summarized below and can be found in [8], [9]). The wavelet transform is a natural tool for such work because of the intrinsic scaling properties of wavelets, and the efficiency of the transform, which can even be computed in real-time as data is collected. In addition, the statistical properties of this algorithm (zero bias, 1/n efficiency, and robustness [10], [11]) are better than any currently known alternative. The estimator has very low memory and computational requirements and scales naturally to arbitrarily high data rates. It is therefore perfect for use in real-time applications such as CAC.

In [12] we showed that the estimator could be

[12]The majority of this work was conducted while the authors were at the Software Engineering Research Centre at RMIT University, Melbourne, Australia. The work was sponsored by Ericsson Australia.

applied to estimating H in real-time, simply, rapidly, and with very low memory requirements. The paper applied the method to Ethernet traffic at 10 Mbps, but we noted that the method should scale to arbitrary traffic rates with respect to both memory and processing requirements, so that it will remain applicable as data rates increase with time. This paper graphically demonstrates this scaling by applying the estimator to ATM (Asynchronous Transfer Mode) traffic at 155 Mbps.

This paper is organised as follows. In Section II we describe the Abry-Veitch joint estimator for the parameters of long-range dependence, and in Section III we describe the online implementation of the estimator. Both have been described in detail elsewhere [8], [12] so only a summary is included here. Section IV describes the implementation issues relating to the measurement of ATM traffic, and an architecture which satisfies all of the requirements for such a system. Section V then describes the actual measurement in real-time of the parameters of long-range dependence for artificially generated traffic.

II. The 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* $x(t)$ 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 as $\gamma_x(k) \sim c_\gamma |k|^{-(1-\alpha)}$, $\alpha \in (0,1)$, or equivalently as the power-law divergence at the origin of its spectrum: $f_x(\nu) \sim c_f |\nu|^{-\alpha}$, $|\nu| \to 0$. The Hurst parameter, H, describes the (asymptotic) self-similarity of the cumulative traffic process corresponding to $x(t)$ which generates the LRD of $x(t)$, itself described by α. It is nonetheless common practice to speak of H in relation to LRD. The two are related as $H = (1+\alpha)/2$.

In [8], [9] (see also [13], [11], [14], [12], [10]) a semi-parametric joint estimator of (α, c_f) is described based on the *Discrete Wavelet Transform*. 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 $\psi_{a,t}(u) \equiv \psi_0(\frac{u-t}{a})/\sqrt{a}$, $a \in I\!R^+$, $t \in I\!R$ generated by simple translations and dilations of the 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 Discrete Wavelet Transform with discrete coefficients $d_x(j,k)$ known as *details*. Henceforth we will deal exclusively with the details of the Discrete Wavelet Transform. 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 [15] 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 phenomenon such as LRD is the fact that the wavelet basis functions themselves possess a scaling property, and therefore constitute an optimal 'co-ordinate 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 almost independent of each other.

We can now outline the estimator as consisting of the following three 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 then averaged across 'time' k to produce an (excellent) estimate of the variance of the wavelet coefficients, called μ_j. It has been shown that the μ_j follow a power-law in j with exponent α.

3. **LRD parameters estimation** A plot, referred to as the Logscale Diagram, is made of $y_j = \log_2(\mu_j)$ against j complete with confidence intervals, and from it the range of octaves $[j_1, j_2]$ where scaling occurs is determined. The LRD parameters H and c_f are then extracted by performing a weighted linear regression over those scales.

Notes:

- Since the expectations of the details are all identically zero [13], the average of the squares of the details at a given j is an estimate of the variance at that j.
- In forming y_j small corrective terms $g(j)$ are in fact subtracted from $\log_2(\mu_j)$ to account for the fact that $E\,[\log]\,(\cdot) \neq \log(E\,[\cdot])$.
- H is related to the slope of the plot, and c_f to a power of the intercept.
- The weights are the known variances of the y_j and do **not** depend on the data.
- 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.

An example of the Logscale Diagram and regression fit using a simulated data set is given in *Figure 1.* The 95% confidence intervals for each y_j, shown as vertical lines at each octave j, are seen to increase with j.

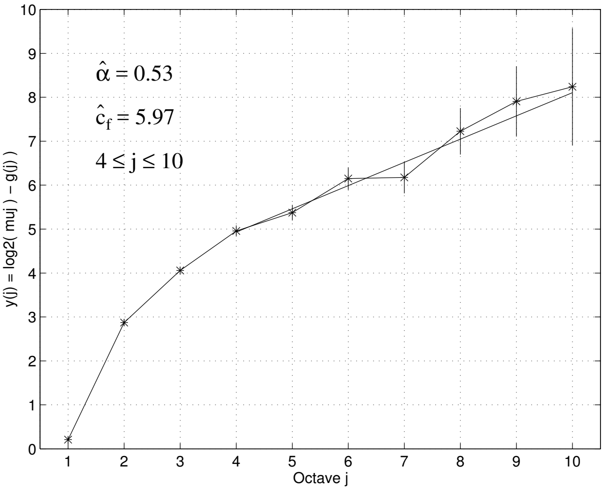

Fig. 1. An example of the Logscale Diagram for a LRD process with strong SRD. The vertical bars at each octave give 95% confidence intervals for the y_j. The series is simulated FARIMA(0,d,2) with $d = 0.25$ ($\alpha = 0.50$) and $\Psi = [-2, -1]$ implying $c_f = 6.38$. Selecting $(j_1, j_2) = (4, 10)$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

III. The on-line estimator

The AV estimator summarized above is gaining acceptance as the method of choice for measuring LRD in traffic [16], [17]. Until recently, however, it 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, is also estimated by the method, and is very important in applications: for example, the confidence intervals of mean estimates for a LRD process are asymptotically proportional to $\sqrt{c_f}$.

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 easily implemented in an on-line fashion using a real-time pyramidal filter-bank (*Figure 2*). Indeed, such filter-banks were devised with on-line applications in mind. The second stage is trivial and 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:

$$\begin{aligned} n_j &\leftarrow n_j + 1, \\ S_j &\leftarrow S_j + d_x(j, n_j)^2. \end{aligned}$$

When the variance estimate at octave j is required for the final stage it can be calculated as $\mu_j = S_j/n_j$. The final stage of the estimation algorithm need not be adapted to an on-line version, as there is no need to compute H every time a new data point is acquired. It may be re-calculated 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 stage is only $O(\log_2(n))$.

Some explanation is required to explain why the first stage of the on-line estimator is scalable. The on-line filter-bank, illustrated in *Figure 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-bank in the form of $K \log_2(n)$ 'half-processed' values. Crucially, 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. If this processor load were to great then the filter-bank can be naturally implemented in Digital Signal Processing (DSP) hardware, but we have not found this to be necessary.

Section V shows how a quite modest computer is capable of performing the AV estimation algorithm, on-line and in real-time on ATM data sampled at 155 Mbps.

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 1 week represents 604 million sample points, which stored as four 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 available scales 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. Note that the on-line algorithm allows all of the scales available in the data to be seen and used, rather than deciding a-priori which scales will be examined.

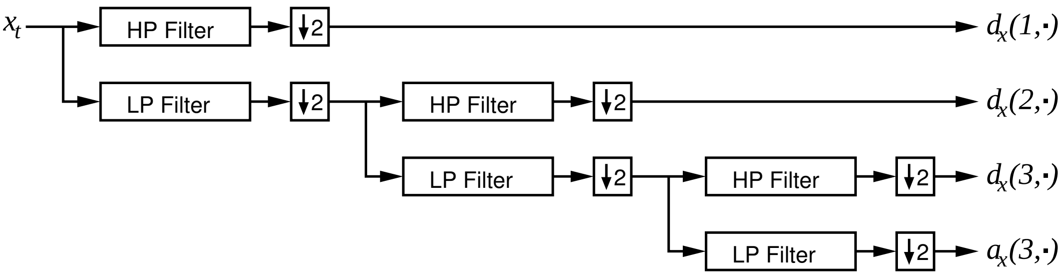

Fig. 2. The filter-bank. At each level in the recursive structure, the High Pass (HP) output: $d_x(j,\cdot)$, and the Low Pass (LP) output: $a(j,\cdot)$, occur at half the rate of the input $a_x(j-1,\cdot)$.

IV. Architecture for measuring ATM traffic

The seminal study of self-similarity in data traffic was conducted on an Ethernet [1], which is a relatively easy medium to measure because of its broadcast nature. Furthermore, at 10 Mbps there are no problems collecting or storing the traffic data. Hence our initial study of real-time measurement of the parameters of LRD was conducted on Ethernet data [12], [18].

ATM is a much more demanding medium. We consider here, ATM over 155 Mbps OC3c SONET (Synchronous Optical NETwork) which has emerged as a common ATM implementation. The demands of such a network are, in particular, a much higher data rate and the fact that SONET is a point-to-point technology. We shall concentrate on measuring IP traffic transmitted over ATM, as this is the type of data to which we have access, though the methods described here could easily be applied to other types of traffic.

In addition to requiring that our ATM monitor measure IP traffic transmitted over ATM at OC3 rates (nominally 155 Mbps), there are a number of other requirements, namely that it

1. be passive; that is, have no effect on network performance;
2. be implemented using cheap, commodity off-the-shelf (COTS) components;
3. allow remote access and control;
4. allow accurate time stamping of packets;
5. be able to measure either the aggregate traffic on a link, or only the traffic on a particular Virtual Circuit (VC), Virtual Path (VP) combination, or only the traffic between specific IP source and destination addresses, or only the traffic transmitted to a specific TCP (Transmission Control Protocol) port.

The first requirement is dictated by practicalities – no network operator would allow access to their network without a guarantee that traffic would not be disrupted. The second requirement was in part dictated by budget, but was further motivated by the philosophy that network monitoring should be almost ubiquitous, rather than a tool for chasing problems. Ubiquity requires cheap and easily assembled monitors, and also creates the need for remote access and control to aid in their management. The next requirement is one of accurate measurement, particularly important for LRD measurement. The final requirement is dictated by many of the applications of network monitoring, for instance billing.

The list of requirements above strongly suggests that the monitor be based around a PC architecture, using a COTS ATM Network Interface Card (NIC). The FORE Systems PCA-200EPC ATM NIC has been used for network monitoring by MCI on the vBNS (very high performance Backbone Network Service [19]). In fact the CORAL group [3], [4], [5], [6], specifically the OC3MON project [7], has made the drivers for this NIC freely available. Hence this was an ideal starting point (we have recently learnt there is at least one other project based at the University of Waikato in New Zealand which has developed similar hardware [20]). We therefore based our monitor around the OC3MON, though minor modifications to their drivers were required for our application.

The architecture of our system is shown in Figure 3. As in the CORAL monitor, the SONET optical link is tapped using an optical splitter, which in our case splits 50% of the light off to our NIC. The NIC reads the ATM cells, and collects and time stamps the initial cell of each IP packet, transmitted using AAL5 (ATM Adaptation Layer 5), and discards all of the other cells of the packet. The first cell contains the TCP/IP headers[1] and can therefore be used to select traffic on IP source/destination or TCP port (which may determine the application through the well known TCP ports), while the cell VC/VP numbers could also be used for traffic selection. The OC3MON project has also investigated *IP flows* though we have not applied this feature.

The time stamp precision is 40 ns relative to the clock on the NIC (approximately 1/70th of the cell transmission time). The actual accuracy of the timestamps may not be quite this good due to processing delays on the ATM card, but these inaccuracies are small compared with the sampling intervals of interest (see below). The cell capture process then produces a "packet count" time series by counting the number of IP headers received during each time interval – referred to as the sampling interval. A second "bit rate" time series of the is generated by examining the AAL5 Protocol Data Units (PDUs) to obtain the length of the PDUs. Obviously this is not the actual line transmission rate, as the ATM cell overhead, and the SONET overhead are not taken into account – we are more interested in the higher layer transmission rate in any case. The packet rate time series is the input to the estimation routines used later, though previous work [11] has shown that either time series should produce equivalent results.

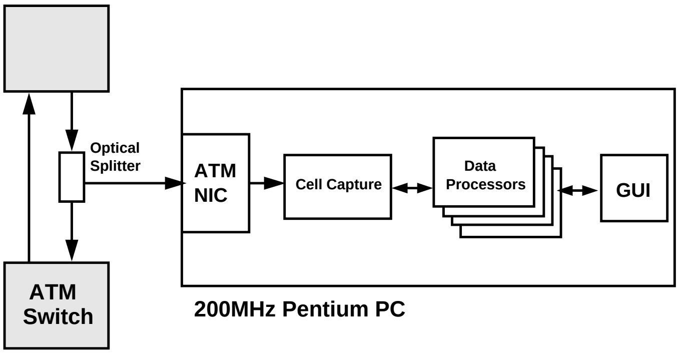

Fig. 3. The architecture of the ATM monitor.

The time series produced by the capture process is then transmitted to a *data processor* which takes the times series and performs the estimation procedure described above. In addition simple statistics of the data, such as the mean bit and packet rates, are also collected. These estimates are then fed to a Graphical User Interface (GUI) front end which displays the results. The GUI also allows control (starting, stopping, and changing of parameters) of the data processors and cell capture processes. More than one data receiver can run in parallel allowing simultaneous monitoring of more than one VP/VC on the same link, or simultaneous estimation on the same data stream using different scales/time intervals.

The communication between cell capture, data processor, and front end is over TCP/IP sockets, allowing for distributed operation. For instance the front end could display results and control the capture and data processor processes from a remote location. Note though, that if this is the case, care should be taken to make sure that the traffic from the monitor itself is not being carried on the monitored link!

The cell capture process was written in C++, the data processors in C and the front end in JAVA, further illustrating the flexibility gained by using the TCP/IP sockets for communication between processes. In principal the JAVA front end could even be run as an applet, though in the cases we considered, the standard JAVA security model would prevent some of the desired actions of a monitor front end, such as logging to a local disk.

Altogether the system costs significantly less than 5000 Australian dollars ($\sim$ \$3000 US). There are a number of adjuncts that could be included for ideal operation, for instance GPS (Global Positioning System) based time calibration can be used to correct for clock drift and ensure that the time stamps generated by the clock are accurate with respect to UTC (Coordinated Universal Time), but as our current measurements are only at a single point this is not necessary.

V. Real-time measurement of ATM traffic

The aim of this paper is *not* to prove the existence of LRD in ATM traffic. Many papers have demonstrated the existence of LRD in data traffic both for instance in Local Area Networks (LANs) [1] and for Wide Area Networks (WANs) [2] and even in specifically ATM networks [17]. The data pre-

[1] Assuming IPv4 with no options.

sented here is intended to show that the monitor is capable of real-time measurement of ATM traffic. Note that the data reduction gained through the on-line monitor allows us to study network behaviour over truly long scales – months to years!

The tests presented here were conducted using artificially generated traffic. For this purpose we built a specific traffic generator [21]. The traffic generator can generate loads of up to 140 Mbps of data traffic (close to the line rate), with LRD characteristics. It uses the superposition of On/Off sources model [22] to generate this traffic. In this case we used 100 sources each running as a separate process which generated pseudo-files (of random size, padded with zeros) and transferred the files across the link. The file sizes were generated using the Pareto distribution, as were the times between file transfers. The value of the shape parameter in the Pareto distribution was 1.5 in both cases, corresponding to a Hurst parameter of 0.75. The files are broken into IP packets which are transferred using the AAL5 across the ATM link. AAL5 takes the packet, adds a minimal tail, and passes the packet to the segmentation and reassembly layer which segments the packets into 48 byte cells and attaches the 5 byte cell headers which are then transmitted over the optical link.

The results are shown in Figures 4 to 7. Figure 4 shows the packet rate, while Figure 5 shows the bit rate; both are averaged over 1 second intervals. The x-axis shows the time relative to the time of the measurement – hence the total displayed period is the final 5 minutes of the experiment. The experiment was carried out over approximately 30 minutes, but in order to be able to clearly see the results we display only the final part of this time series. Note that the figures are very similar. The transmitted files tend to be large, and so the packets transmitted are almost all the same size (the maximum PDU size 65,535 bytes), and therefore the bit rate depends very closely on the packet rate. Note also that the bit rate shown is the data bit rate, that is, the number of data bits transferred, including IP and TCP headers, but excluding ATM and SONET overhead which add approximately 10% to the bit rate displayed.

We can see from the figures that the traffic is highly variable, as desired. However, the true test of self-similarity is to apply the estimator. Figure 6 shows the Hurst parameter estimates over the same time interval. As the time interval displayed is near the end of the data set, the estimate changes only very slowly as each new measurement is relatively unimportant compared to the large body of preceeding data incorporated in the estimates. We may immediately note that the Hurst parameter estimate is not as accurate as its confidence intervals indicate (the measured value is around 0.8 while the theoretical value is 0.75).

The difference between the theoretical and measured values of H does not arise because of errors in the measurement process. It is in fact an artifact of the generation process. While the On/Off model has simple theoretical properties, in practice generation of LRD processes using On/Off models is not as simple as one might expect. First, there is a little known fundamental problem with measuring the parameters of an On/Off process which can lead to bias in estimates even for very long series. It concerns the sampling of the longer On and Off events which leads to a truncation in the sample correlation structure of any sample path. The phenomena has been described in [23], and the bias observed using other estimators of the Hurst parameter has been noted elsewhere [24]. Unfortunately, these truncation effects intrinsically limit the range of scales over which the generated traffic displays self-similarity – so that simply extending the length of the time series, which would seem to be an obvious solution, is of limited help. Furthermore there is the practical problem that in any real application the length of the measurement interval will be limited by time constraints due to non-stationarity in the data [25]. Second, our implementation uses separate user processes on a multitasking operating system, and the resulting kernal controlled scheduling prevents us controlling the behavior of the traffic on fine time scales. This last point however is not so important from the point of view of H measurement, as LRD processes are only asymptotically self-similar. Thus the scaling behaviour begins only above some lower cuttoff scale, and so the lowest scales are not used in the estimation in any case, as noted below.

Such effects highlight the importance of examining Logscale Diagrams, and not simply taking a blind estimate of a single parameters such as H.

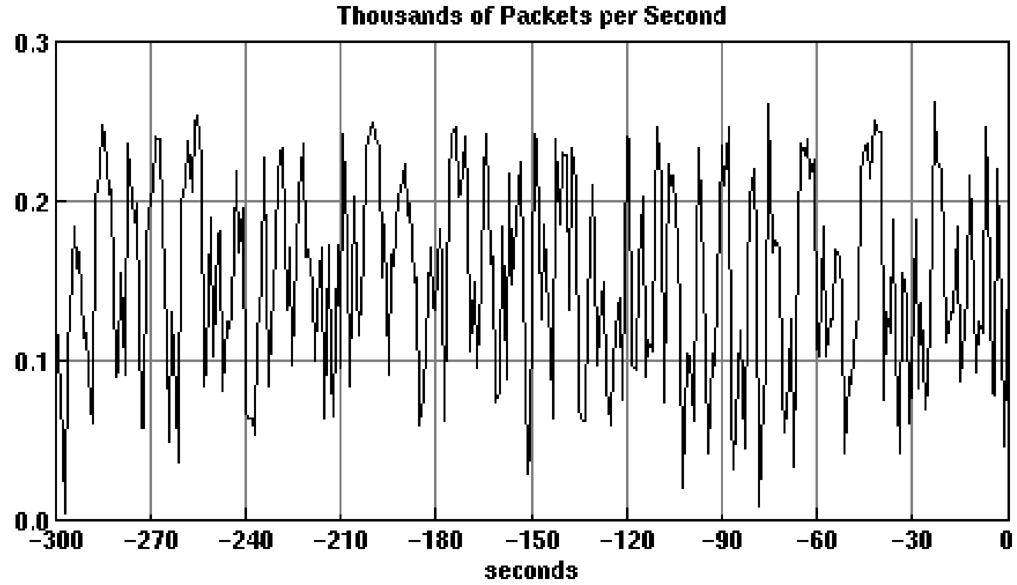

Fig. 4. An example of artificially generated traffic – in thousands of packets per second over the last 5 minutes of traffic generation.

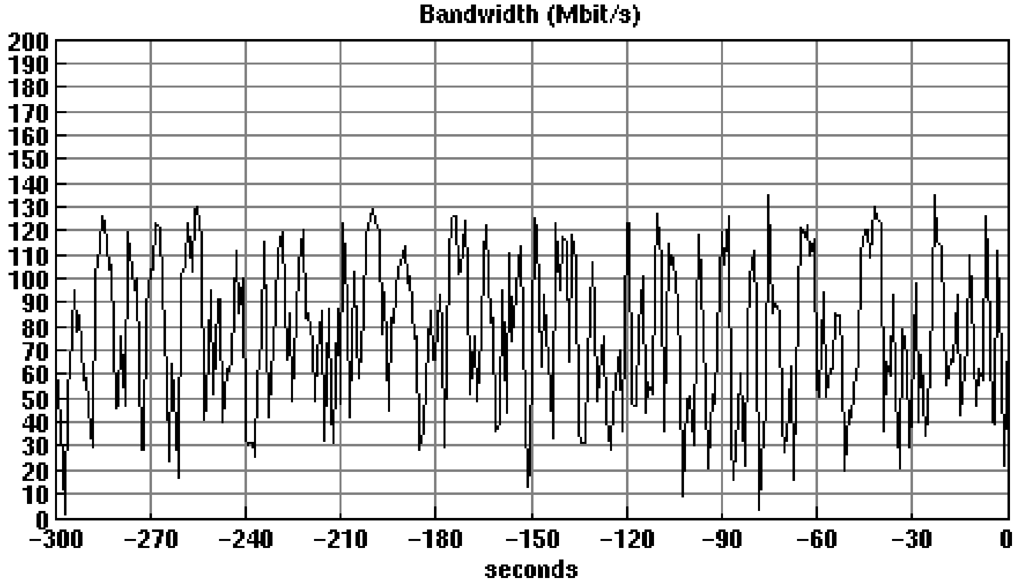

Fig. 5. An example of artificially generated traffic – in Mbits per second over the last 5 minutes of traffic generation.

Figure 7 shows the Logscale Diagram of the data in question. Note in *Figure 7* the x-axis has been modified to display the time-scale rather than the values of j. As described in Section II the Logscale Diagram shows the values y_j which are used in a weighted regression to perform the estimate of H. We can immediately see that the y_j do not all fall on a straight line, and so cannot all be used in the estimate. The Logscale Diagram at the finer time scales is quite non-linear. It becomes approximately linear above a time scale of around 1 second, and it is in this region (shown by the vertical lines) that we perform our estimate of the Hurst parameter. Finally at time-scales around 1 minute there appears to be the beginnings of the truncation effect noted above. These higher scales should be excluded to avoid the associated bias, but this would leave us with a very narrow range of scales to make our measurements.

We have recently developed a superior method for generating traffic which has been documented in [21], however it is useful to describe the behavior of the simpler On/Off generator here. We have thereby demonstrated that an examination of Logscale Diagrams, based on accurate measurement, can clearly display the surprisingly complex behaviour of On/Off generators.

VI. Conclusion

In summary, this paper demonstrates that cheap, scalable, and ubiquitous monitoring of data traffic is possible, and can include measurements of the fractal nature of the traffic. Furthermore, such measurements can be successfully performed in real-time, enabling their use in real-time applications such as CAC, congestion control, and network monitoring.

Acknowledgments

The authors gratefully acknowledge the support of Ericsson Australia.

References

[1] Will E. Leland, Murad S. Taqqu, Walter Willinger, and Daniel V. Wilson, "On the self-similar nature of Ethernet traffic (extended version)," *IEEE/ACM Transactions on 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 Transactions on Networking*, vol. 3, no. 3, pp. 226–244, 1994, http://ee.lbl.gov/nrg-papers.html.

[3] "Coral network traffic analysis," online: http://moat.nlanr.net/Coral/.

[4] J. Dugan, "Coral – flexible, affordable, high performance network statistics collection," online: http://www.caida.org/Tools/Coral/.

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

[6] Kevin Thompson, Gregory J. Miller, and Rick Wilder, "Wide-area Internet traffic patterns and characteristics," *IEEE Networks*, 1997, Extended Version: http://www.vbns.net/presentations/papers/index.html.

[7] J. Apisdorf, K. Claffy, K. Thompson, and R. Wilder, "OC3MON: Flexible, affordable, high performance statistics collection," in *INET'97 Conference*, June 1997.

[8] Darryl Veitch and Patrice Abry, "A wavelet based joint estimator of the parameters of long-range dependence," *IEEE Transactions on Information Theory special issue on "Multiscale Statistical Signal Analysis and its Applications"*, vol. 45, no. 3, April 1999.

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

[10] Matthew Roughan and Darryl Veitch, "Measuring long-range dependence under changing traffic conditions," in *IEEE INFOCOM'99*, NY, NY, March 1999, IEEE Computer Society Press, Los Alamitos, California.

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

[12] Matthew Roughan, Darryl Veitch, and Patrice Abry, "On-line estimation of parameters of long-range dependence," in *IEEE GLOBECOM'98*, Sydney, Australia, November 1998, pp. 3716–3721.

[13] P. Abry, P. Gonçalvès, and P. Flandrin, *Wavelets and Statistics*, vol. 105 of *Lecture Notes in Statistics*, chapter Wavelets, Spectrum estimation, $1/f$ processes., pp. 15–30, Springer-Verlag, New York, 1995.

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

[15] I. Daubechies, *Ten Lectures on Wavelets*, SIAM, Philadelphia (PA), 1992.

[16] A.Feldmann, A.C.Gilbert, W.Willinger, and T.G.Kurtz, "The changing nature of network traffic: Scaling phenomena," *Computer Communications Review*, vol. 28, no. 2, 1998.

[17] Judith L. Jerkins and Jonathan L. Wang, "A measurement analysis of ATM cell-level aggregate traffic," in *IEEE GLOBECOM'97*, 1997.

[18] Matthew Roughan, Darryl Veitch, and Patrice Abry, "Real-time estimation of the parameters of long-range dependence (extended version)," in *to appear in IEEE Transactions on Networking*, 2000.

[19] "Very high performance backbone network service," http://www.vbns.net/.

[20] "The university of waikato, atm group page," online: http://atm.cs.waikato.ac.nz/atm/.

[21] On line generation of fractal and multi-fractal traffic, "Darryl veitch and jon-andars backar and jens wall and jennifer yates and matthew roughan," in *PAM2000, Workshop on Passive and Active Networking, New Zealand*, 2000.

[22] Bong K. Ryu and Steven B. Lowen, "Point process approaches to the modelling and analysis of self-similar traffic - part i: Model construction," in *IEEE INFOCOM'96: The Conference on Computer Communications*, San Francisco, California, March 1996, vol. 3, pp. 1468–1475, IEEE Computer Society Press, Los Alamitos, California.

[23] M.Roughan, J.Yates, and D.Veitch, "The mystery of the missing scales: Pitfalls in the use of fractal renewal processes to simulate LRD processes," in *Applications of Heavy Tailed Distributions in Economics, Engineering and Statistics*, American University, Washington, DC, June 1999.

[24] Steven B. Lowen and Malvin C. Teich, "Estimation and simulation of fractal stochastic point processes," *Fractals*, vol. 3, no. 1, pp. 183–210, 1995.

[25] Matthew Roughan and Darryl Veitch, "A study of the daily variation in the self-similarity of real data traffic," in *Proceedings of the 16th International Teletraffic Congress - ITC 16*, P. Key and D. Smith, Eds. 1999, vol. 3b, pp. 67–76, Elsevier, Amsterdam.

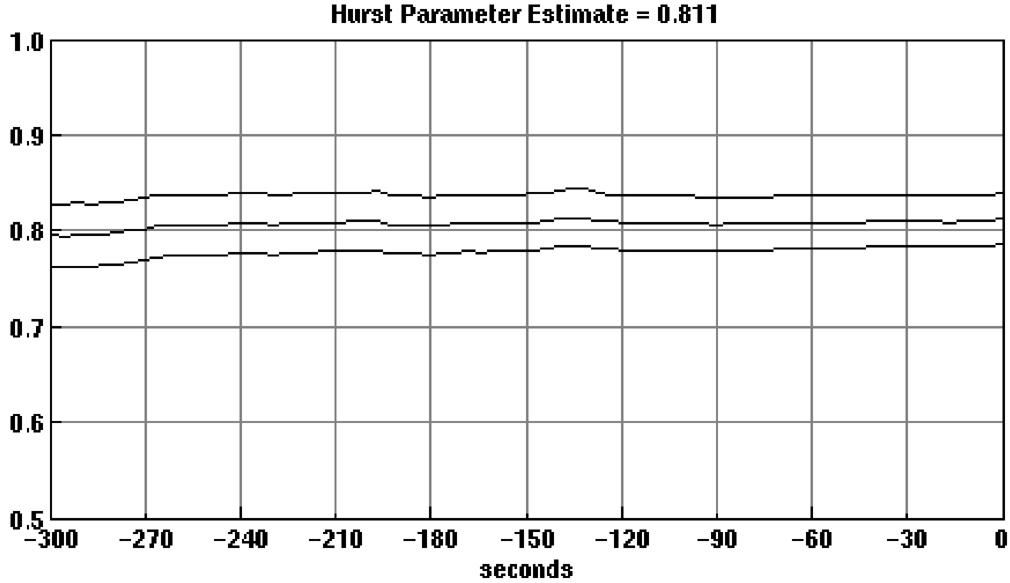

Fig. 6. An example output of Hurst parameter estimation. The three curves shown are, in order from the top, the upper 95% confidence interval, the estimate, and the lower 95% confidence interval.

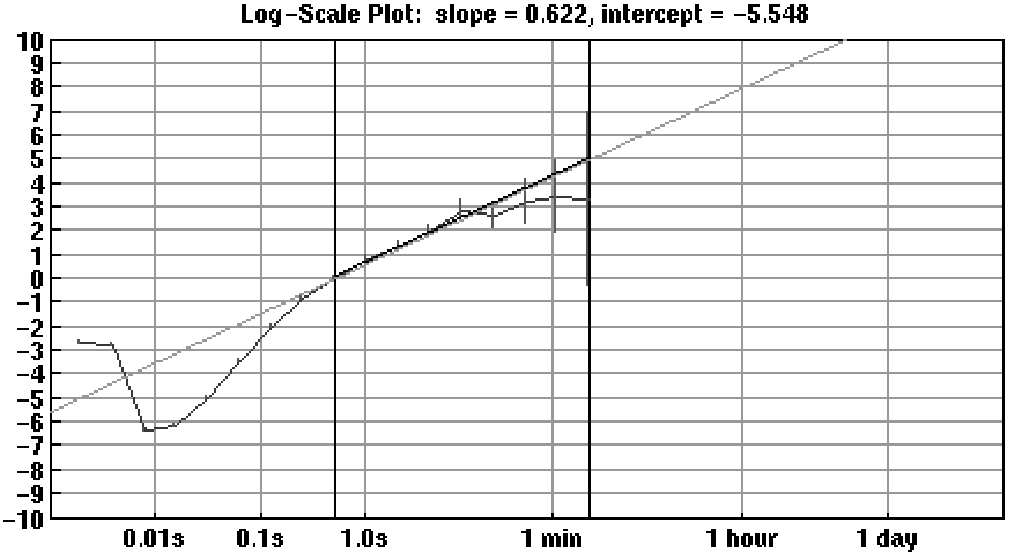

Fig. 7. The Logscale Diagram for the generated traffic showing a fitted regression line, and the scales used in the regression (shown as vertical lines across the entire y-range).