Understanding the Limitations of Estimation Methods for Long-Range Dependence

Thomas Karagiannis, Mart Molle, Michalis Faloutsos
Department of Computer Science & Engineering
University of California, Riverside
{tkarag,mart,michalis}@cs.ucr.edu

Abstract

Over the last ten years, long-range dependence (LRD) has become a key concept in modeling networking phenomena. The research community has undergone a mental shift from Poisson and memoryless processes to LRD and bursty processes. Despite its popularity, LRD analysis is hindered by two main problems: a) it cannot be used by nonexperts easily, and b) the identification of LRD is often questioned and disputed. The main cause for both these problems is the absence of a systematic and unambiguous way to identify the existence of LRD. This paper has two main thrusts. First, we explore the (lack of) accuracy and robustness in LRD estimation. We find that the current estimation methods can often be inaccurate and unreliable, reporting LRD erroneously. We search for the source of such problems and identify a number of caveats and common mistakes. For example, some of the methods misinterpret short-range correlations for LRD. Second, we develop methods to improve the robustness of the estimation. Through case studies, we demonstrate the effectiveness of our methods in overcoming most known caveats. Finally, we integrate all required functionality and methods in an easy to use software tool. Our work is a first step towards a systematic approach and a comprehensive tool for the reliable estimation of LRD.

I. Introduction

The concepts of long-range dependence and self-similarity have revolutionized network modeling. During the last decade, there has been ample evidence of long-range dependence, scaling phenomena and heavy tailed distributions in various aspects of network behavior, such as packet interarrivals, traffic load, file sizes, and web traffic. For example, it has been observed that packet interarrival times are described by marginal distributions with heavier tail than that of the exponential distribution of a Poisson process. The consequence of these observations is profound: the research community has made a mental shift from Gaussian distributions and memoryless Poisson processes, to heavy-tail distributions and long-memory processes.

Intuitively, long-range dependence describes the memory of a process. In a LRD process, the current state has significant influence on its subsequent states far into the future. A consequence of long-range dependence is high variability in the behavior of the process over multiple time scales. In other words, when we aggregate the data to produce the average behavior of the process over large(r) time intervals, its observed behavior does not become smooth. The existence of LRD in a system affects significantly how we: a) model, b) provision for, and c) predict its behavior.

The problem with LRD is the difficulty in identifying and estimating its existence beyond doubt. This translates in two related questions:

- How accurate and robust are the current LRD estimators?
- How can we improve the reliability of the estimation?

The source of the complication is that there does not exist a definitive and systematic way to estimate long-range dependence. The question is simple: does a given a time series exhibit long-range dependence? The answer, however, is far from straightforward. The predominant way to quantify long-range dependence is the value of the *Hurst exponent*, which is a scalar number. However, the Hurst exponent cannot be calculated in a definitive way, it can only be estimated. Furthermore, there are several methods to estimate the Hurst exponent, but they often produce conflicting estimates. It is not clear which of the estimators provides the most accurate estimation. Consequently, studies can often arrive at arbitrary and misleading conclusions.

Limited previous work focuses on our ability to detect LRD correctly, while most efforts focus on observing LRD in real phenomena. However, without a robust methodology, we cannot be confident in our observations. An overwhelming number of studies is devoted to the identification and modeling of LRD in network behavior. Some of these efforts provide results that are difficult to reproduce or accept without concern. They sometimes report results: a) based on few estimation methods, b) without reporting the statistical confidence of the result, or the methods that were actually used. On the other hand, we find significantly less efforts that examine the limitations of robust LRD estimation. An even smaller number of studies takes the next step of providing practical guidelines to improve the estimation. The bottom line is that it is hard for a practitioner to acquire and master the necessary background, and develop the tools to estimate long-range dependence effectively.

In this paper, we identify the limitations of estimating LRD and provide tools and mechanisms to improve the robustness of the estimators. Our goal is to develop a systematic methodology in order to ensure accurate and unambiguous results. The first step towards this direction is to identify the limitations and the caveats in estimating long-range dependence. The second is to develop methods to stress-test and improve the accuracy of the estimation. Our contributions can be summarized in the following points:

- **Documenting the limitations of LRD estimation:** We examine the accuracy and robustness of the current estimators. We demonstrate that there does not exist an ultimate estimator, since each estimator can outperform the others depending on the scenario. In addition, we illustrate that estimation methodologies are likely to falsely identify LRD in a number of cases.
- **Identifying common errors and caveats:** Having establish the limitations of the estimation methodologies we identify their origins. We find three main types of factors that inhibit estimation: a) common processes such as noise, and periodicity, b) weaknesses of the estimators, such as misinterpreting short-range as long-range dependence, and c) the inherent difficulty in estimating asymptotic behavior which LRD and self-similarity describe.
- **Developing a test for the LRD:** We suggest a method of controlled randomization[1], **randomized buckets**, as an excellent tool to stress test the existence of LRD. This test separates the short-range from long-range dependence.
- **Improving the robustness of the estimation:** We propose several techniques and methods to increase the robustness of the identification and estimation of LRD. Each of these methodologies address specific limitations of the estimators, such as removing periodicities or short-range correlation.

The scope of this work. Our work is intended as a survival guide and how-to manual for practitioners. We identify several caveats and common pitfalls related to LRD estimation, and provide a systematic approach to avoid errors in estimating and interpreting the results. A few of the problems and techniques discussed are partially known in a variety of disciplines such as signal processing. However, they have not been presented before as part of a complete and comprehensive manual. In addition, our work goes much further than existing studies by examining thoroughly their effects and their significance in practice both analytically and experimentally. Earlier parts of this work have appeared in [14] [13].

All the above experience and learned lessons are integrated in our SELFIS tool [13], whose main goal is the estimation of LRD. SELFIS is a Java-based, portable, and user-friendly tool which we develop in order to: a) provide easy access to powerful and complex tools, b) provide a reference point for comparison purposes, c) prevent researchers from effort duplication. The 150 downloads of our tool validate our intuition that such a tool was long overdue. The experimental work presented here uses the SELFIS tool.

The rest of this paper is structured as follows. Sections II is a brief overview of self-similarity, long-range dependence and current methodologies used for LRD estimation. Section III summarizes previous findings of long-range dependence in the networking area. In section IV, we describe how randomized buckets can be employed to effectively identify LRD in a given time-series. Section V examines the limitations of LRD estimators in terms of accuracy and robustness. Section VI presents a set of suggestions and practical solutions towards robust LRD estimation and identification. In section VII we describe a systematic approach to LRD analysis. Finally, section VIII concludes this paper.

[1] This approach was first introduced in [8] to illustrate the effects of LRD on queueing delays, but it has not received the attention it deserves as a test for LRD analysis. In this paper, we expand this methodology to multiple levels of randomization: thus we can isolate short-, medium- and long-range dependencies. In addition, for the first time this methodology is theoretically analyzed. We provide analytical formulas and demonstrate its strong capabilities in LRD validation.

II. Basic Concepts and Definitions

This section explains the concepts of long-range dependence and self-similarity in the context of time series analysis. First, we present basic definitions that we are going to use throughout the paper. Next, we briefly describe self-similarity and long-range dependence , their intuition, and their differences. Finally, we present a description for many of the widely used Hurst exponent estimation methodologies.

A. Stochastic Time Series

Let $X(t)$ be a stochastic process. In some cases, X may take on the form of a discrete time series $\{X_t\}$, $t = 0, 1, \ldots$, either through periodic sampling or by averaging its value across a series of fixed-length intervals. We say that $X(t)$ is *stationary* if its joint distribution across a collection of times $t_1, \ldots t_k$ is invariant to time shifting. Thus, the dependence between the values of the process at different times can be characterized by evaluating its *autocorrelation function* (ACF), $\rho(k)$. The ACF measures the similarity between a series X_t, and a shifted version of itself, X_{t+k}:

$$\rho(k) = \frac{E[(X_t - \mu)(X_{t+k} - \mu)]}{\sigma^2} \tag{1}$$

where μ and σ are the respective sample mean and standard deviation for X. Note that the definition for ACF is normalized in a way that is *invariant* to all linear transformations. In other words, if $Y(t) = a \cdot X(t) + b$ for some constants a and b, then:

$$\rho_Y(k) = \frac{E[(Y_t - (a \cdot \mu + b))(Y_{t+k} - (a \cdot \mu + b)]}{a^2 \cdot \sigma^2} \equiv \rho_X(k)$$

Hence, $\rho(k)$ only provides us with the *relative size* of the dependence between values of the process as a function of the time shift k, and we have no way to determine its actual significance.

The *spectral density* of a time series provides an alternative approach to characterizing the dependence between values of the process at different times. Let $X_0, \ldots X_{N-1}$ be a discrete time series of length N, which therefore describes the behavior of the process X across a finite *measurement interval* whose length is $N - 1$ times the sampling period. If we assume that $X_0, \ldots X_{N-1}$ represents one iteration of an infinitely repeating pattern that describes the behavior of X over all time, then we can transform this “time domain” representation of X into an equivalent “frequency domain” representation using the discrete Fourier transform (DFT) algorithm. The result is a decomposition of X into the weighted sum of trigonometric functions using the set of frequencies $\nu_k = \frac{2\pi k}{N}$ radians per measurement interval, $k = 0, \ \ldots \lfloor \frac{N-1}{2} \rfloor$, that divide evenly into the measurement interval, Thus, the similarity between the original series, X_t, and a copy of itself that has been shifted by $1/k$th of the measurement interval, $X_{t+N/k}$, can be found by calculating the power spectral density, $I(\nu_k)$, which corresponds to the square of the weight assigned to frequency ν_k in the transformed series:

$$I(\nu_k) = \frac{2}{N} \left| \sum_{j=0}^{N-1} X_j e^{i\nu_k j} \right|^2$$

Of interest is also *the aggregated process* $X_k^{(m)}$ of a time-series, which is defined as follows:

$$X_k^{(m)} = \frac{1}{m} \sum_{i=km}^{(k+1)m-1} X_i, \quad k = 0, 2, ..., \left\lfloor \frac{N}{m} \right\rfloor - 1. \tag{2}$$

If $\{X_k\}$ were i.i.d., then clearly $Var(X^{(m)}) = \frac{\sigma^2}{m}$. More generally, however, if there is dependence among the values of X, then [2]

$$Var(X^{(m)}) = \frac{\sigma^2}{m} \left(1 + 2 \sum_{k=1}^{m-1} \left(1 - \frac{k}{m} \right) \rho(k) \right).$$

Thus, the variance of the aggregated process converges to zero at a much slower rate than $1/m$ if the values in the sequence are dependent.

B. Self-Similarity and Long-Range Dependence

We say that a stationary process X is *long-range dependent* if its autocorrelations decay to zero so slowly that their sum does not converge, i.e., $\sum_{k=1}^{\infty} \rho(k) = \infty$. Intuitively, the dependence between widely separated values from a long-range dependence process does not decay to zero, even across infinitely large time shifts.

We say that a stochastic process X is *self-similar* if

$$X(at) = a^{H} X(t)$$

where the self-similarity parameter, H, is often called the *Hurst exponent*. Intuitively, self-similarity describes the phenomenon where certain properties about the process are preserved irrespective of scaling in space or time.

Second-order self-similarity describes the property that the correlation structure (ACF) of a time-series is preserved irrespective of time aggregation. Simply put, the ACF of a second-order self-similar time-series is the same in either coarse or fine time scales. A stationary process X_t is second-order self-similar [25], if

$$\rho(k) = \frac{1}{2}[(k+1)^{2H} - 2k^{2H} + (k-1)^{2H}], \quad 0.5 < H < 1, \tag{3}$$

and asymptotically exactly self-similar if

$$\lim_{k \to \infty} \rho(k) = \frac{1}{2}[(k+1)^{2H} - 2k^{2H} + (k-1)^{2H}], \quad 0.5 < H < 1.$$

Second-order self-similar processes are characterized by a hyperbolically decaying autocorrelation function, and are extensively used to model long-range dependent processes. On the contrary, short-range dependence is characterized by quickly decaying correlations (e.g. ARMA processes). From these definitions, it is implied that a time-series is characterized by LRD, if $0.5 < H < 1$. As $H \to 1$, the dependence is stronger.

The most extensively used self-similar processes to simulate long-range dependence are fractional Gaussian noise (fGn) and fractional ARIMA processes. fGn is the increment of fractional Brownian motion (fBm). A stochastic process is fBm if it is Gaussian and its autocorrelation function is described by Eqn. 3. The fractional ARIMA(p,d,q) model is a fractional integration of the ARMA(p,q) model. ARFIMA processes describe long-range dependent series when $0 < d < 0.5$, where $H = d + 0.5$.

C. Hurst exponent Estimators

Hurst exponent estimators can be classified in two main general categories; Estimators operating in the time domain and those operating in the frequency or wavelet domain. In more detail, the following bullets describe the widely used estimation methodologies:

1) Time-domain estimators: These estimation methodologies are based on investigating the power-law relationship between a specific statistical property of the time-series and the time aggregation block size m.

- *Absolute Value method.* The log-log plot of the aggregation level versus the absolute first moment of the aggregated series $X^{(m)}$ is a straight line with slope of $H - 1$, if the time-series is long-range dependent (where H is the Hurst exponent).
- *Variance method.* The method plots in log-log scale the sample variance versus the block size of each aggregation level. If the series is long-range dependent then the plot is a line with slope β greater than -1. The estimation of H is given by $H = 1 + \frac{\beta}{2}$.
- *R/S method.* This method uses the rescaled range statistic (R/S statistic). The R/S statistic is the range of partial sums of deviations of a time-series from its mean, rescaled by its standard deviation. A log-log plot of the R/S statistic versus the number of points of the aggregated series should be a straight line with the slope being an estimation of the Hurst exponent.
- *Variance of Residuals.* The method uses the least-squares method to fit a line to the partial sum of each block m. A log-log plot of the aggregation level versus the average of the variance of the residuals after the fitting for each level should be a straight line with a slope of $H/2$.

2) Frequency-domain/wavelet-domain estimators: These estimators operate in the frequency or the wavelet domain. Intuitively, they examine if the spectrum or energy of a time-series follows power-law behavior.

- *Periodogram method*. This method plots the logarithm of the spectral density, $I(\nu)$, versus the logarithm of the frequency, ν. The slope provides an estimate of H.
- *Whittle* estimator. The method is based on the minimization of a likelihood function, which is applied to the periodogram of the time-series. It gives an estimation of H and produces a confidence interval. It does not produce a graphical output. Whittle estimator assumes a priori that the power spectrum of the underlying process of the dataset is known. That is, the Whittle estimator should only be used if a time-series has already been shown by other means to be consistent with a specific process e.g., fGn. *It should not be used to detect LRD.*
- *Abry-Veitch (AV)*. The Hurst exponent is estimated by using the wavelet transform of the series [1]. A least-squares fit on the average of the squares of the wavelet coefficients at different scales is an estimate of the Hurst exponent. The method produces both a graphical output and a confidence interval.

III. Previous Work

Research on self-similarity and long-range dependence of Internet traffic can be classified in two general categories. The first includes studies on the manifestation of such phenomena, their origins, and their impact on performance. The second relates to the accurate estimation and efficient identification of these complex properties and their actual modeling for simulation and design purposes.

There has been ample evidence of long-range dependence and scaling phenomena in many different aspects of network traffic. The first experimental evidence of self-similar characteristics in local area network traffic were presented in the pioneering work of [18]. The authors perform a rigorous statistical analysis of Ethernet traffic measurements and were able to establish its self-similar nature. Similar observations were presented for wide area Internet traffic in [27], where it was also shown that packet interarrival times follow distributions with a tail heavier than that of the exponential. Heavy tails have been widely considered as the main cause of this scaling behavior. It was shown that file and transfer sizes are in agreement with this type of distributions and their shape was directly related to the self-similar properties of traffic [6] [39] [24]. On the other hand, TCP congestion control has been suggested as another underlying cause of long-range dependence. Studies have argued on the possibility of TCP mechanisms to result and propagate self-similar behavior in the network [36] [37] [19]. Furthermore, it has also been observed that traffic presents even more sophisticated properties such as multi-fractal scaling. This was usually associated with either the Round Trip Time(RTT) or intrusive “fast” flows with small interarrival times [9] [35] [42] [30] [11] [32]. Finally, in [10] [31] [23] the relevance and the effects of the self-similarity on various metrics of network performance are examined.

All these overwhelming observations lead to a shift from the traditional Poisson modeling and independence assumptions [16] which were discarded as unrealistic and overly simplistic [38]. However, there have been a few indications that these models could still be applicable as the number of sources increases in fast backbone links with large volumes of traffic multiplexing [4] [5]. In addition, other studies [40] [41] point out that a number of end-to-end network properties seem to agree with the independence assumptions in the presence of nonstationarity.

The second major aspect of research on self-similarity and long-range dependence is estimation of the Hurst exponent. An overview of a large number of these estimation methodologies can be found in [34] [2]. Relatively little effort has been devoted to study the accuracy of the estimation methodologies [34] and point out difficulties in long-range dependence estimation [21] [14]. These studies present pitfalls when estimating the intensity of long-range dependence in the presence of trends, non-stationarity, periodicity and noise. Furthermore, the limitations of the variance-time estimator have been analyzed in [17]. Of major importance is also the development of models for simulating long-range dependence. Proposed models like the one in [30] or generators for long-range dependent time-series [26] are hard to evaluate in practice. Thus, there are hardly any studies that assess the various models or compare the different generators. In general, this suggests the need for practical tools and a systematic methodology to estimate, validate and generate long-range dependent time-series.

IV. LRD Identification via Randomized Buckets

We propose *Randomized Buckets* as an intuitive methodology for the stress-testing and validation of long-range dependence. Randomized buckets is based on randomizing specific parts of a time-series in order to independently control the amount of correlation at different time-scales. The methodology is analyzed in detail in the following sections. In section IV-A we describe randomized buckets. In section IV-B we demonstrate examples on synthesized

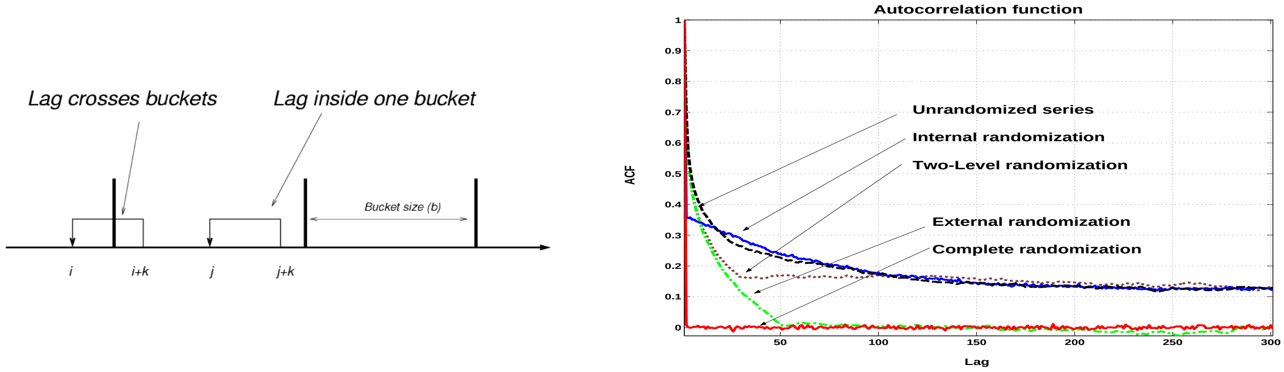

Fig. 1. LEFT: Pairs separated by lag k can belong to the same bucket or not in which case they are inbucket or outbucket pairs respectively. RIGHT:Autocorrelation function (ACF) for a fGn series with H=0.8, before and after bucket randomization. Note that the ACF of the initial fGn series, as well as the ACF after internal and 2-Level randomization all exhibit LRD for large lags. On the contrary, the ACF after external or complete randomization is flat for small and for all lags respectively.

time-series. Finally, in section IV-C we provide a set of analytical results on the effect of randomized buckets on the correlation structure of a time-series.

A. *Methodology*

Randomized buckets is based on the idea of decoupling short- from long-range correlations in a time-series. This is achieved through partitioning of the time series into a set of “buckets” of length b. Thus, we define the contents of the uth bucket to be items $X_{u \cdot b}, \ldots, X_{(u+1) \cdot b-1}$ from the series, and the **home** of item X_i to be bucket $H(i) \equiv \lfloor i/b \rfloor$. Also, we say that two items (X_i, X_j) form an **inbucket** pair if $H(i) = H(j)$; otherwise, they form an **outbucket** pair with an **offset** of $|H(i) - H(j)|$ buckets. Note that this classification depends on the (fixed) locations of the bucket boundaries, and not just the separation between two items in the time series. For example, fig. 1 (left), shows that two items separated by lag k could form either an inbucket or outbucket pair.

Once the series has been partitioned in this way, we can then apply one of the following bucket randomization algorithms to reorder its items:

External Randomization (EX): The order of buckets is randomized, whereas the content of each bucket remains intact. This can be achieved by labelling each bucket with a bucket-id between 0 and $\lfloor \text{SeriesLength}/b \rfloor$, and randomization the bucket-ids. External randomization preserves all correlations among the inbucket pairs, while equalizing all correlations among the outbucket pairs with different offsets. Thus, if the series is sufficiently long, the ACF should not exhibit significant correlations beyond the bucket size.

Internal Randomization (IN): The order of the buckets remains unchanged while the contents of each bucket are randomized. As a result, correlations among the inbucket pairs are equalized, while correlations among the outbucket pairs are preserved, but rounded to a common value for each offset. Thus, if the original signal has long-memory, then the ACF of the internally-randomized series will still show power-law behavior.

Two-Level Randomization (2L): Each bucket is further subdivided into a series of “atoms” of size a. Thereafter, we apply external randomization to the block of $\lfloor b/a \rfloor$ atoms within each bucket. As a result, both short-range correlations (within each atom) and long-range correlations (across multiple buckets) are preserved, while medium-range correlations (across multiple atoms within the same bucket) are equalized. This novel type of randomization allows us to also study the effect of medium-range correlations.

B. *Examples*

We show how long-range dependence can be detected using randomized buckets in the following case study. We synthesized a fGn series of length 64K (2^{16}) and Hurst exponent 0.8. The sample autocorrelation function (ACF) for this series is shown in fig. 1(right), which clearly shows a power-law like behavior and implies long-range dependence. The same figure also shows the ACF function after the series is randomized with four different ways:

- **Internal Randomization** (using $b = 50$) significantly lowers and flattens the ACF at small values of the lag compared to the original (unrandomized) series. However, for large values of the lag, internal randomization has no effect on the ACF. Observe that long-range dependence seems to dominate the original series, since the effect of equalizing the inbucket correlations on ACF is minimal.
- **External Randomization** (using $b = 50$) causes the ACF to drop smoothly from the the initial value for the unrandomized sequence to zero as the lag increases, reaching zero at exactly the bucket size!
- **Complete Randomization** (using $b = 1$) removes all correlations by creating a completely random series. As one would expect, the ACF shows that no correlation exists.
- **Two-Level Randomization** (using $a = 30$, $b = 300$) exhibits similar behavior to external randomization for small values of the lag (i.e., less than a), along with similar behavior to internal randomization for large values of the lag. These two limiting values also match the ACF for the original (unrandomized) series, but for intermediate values the two-level randomization significantly reduces the correlations.

C. Analysis of Randomized Buckets

In this section, we provide analytical results that explain the shape of the autocorrelation function in the each different case of randomized buckets.

Suppose we apply randomized buckets to an existing series X_0, X_1, ..., X_N to create the modified series $\hat{X}_0$, $\hat{X}_1$, ... $\hat{X}_N$. How will this randomization be reflected in the sample Autocorrelation function for the the modified series, $\hat{\rho}(\cdot)$? In the remainder of this section, we derive analytical formulas for computing the autocorrelation function of the randomized series from the autocorrelation function of the initial (unrandomized) series. Note that a practitioner does not need to understand this analysis in order to use bucket randomization, but it does provide the understanding and theoretical proof of the effect of bucket randomization on a series.

Our analysis relies on the randomization of the values. For example, picking a *specific* item $\hat{X}_i$ from bucket $H(i)$ *after* internal randomization is equivalent to randomly selecting an item from the same bucket *without* randomization.

During the calculation of $\hat{\rho}(\cdot)$, we must eventually evaluate all possible **pair-correlations** of the form:

$$\frac{E[(\hat{X}_i - \mu)(\hat{X}_j - \mu)]}{\sigma^2}$$

Out of the complete set of pair-correlations, we find the subset, of size N_k say, for which $|i - j| = k$ and average them to obtain $\hat{\rho}(k)$, the ACF for lag k. Since randomization the series preserves both its mean μ and variance σ^2, we recognize that we must have previously evaluated *exactly the same set of pair-correlations* in order to calculate $\rho(\cdot)$. However, due to the relabeling of items during the randomization, we must average *different subsets of size* N_k to obtain $\rho(k)$ and $\hat{\rho}(k)$.

For concreteness, suppose the two items $\hat{X}_i$ and $\hat{X}_j$ from the randomized series were originally labeled $X_{i'}$ and $X_{j'}$ in the unrandomized series, and the lag between these two items was $k' = |i' - j'|$. Then the associated pair-correlation value will be used to calculate both $\rho(k')$ for the unrandomized series and $\hat{\rho}(k)$ for the randomized series, and in general it will *not* be equal to k.

Thus, the key result we need is the following: for all combinations of $k \geq 1$ and $l \geq 1$, how many pair-correlations come from pairs that were separated by lag k in the original (unrandomized) series and lag l in the randomized series, $N_{k,l}$? Once we find $N_{k,l}$, we can calculate $\hat{\rho}(\cdot)$ directly from $\rho(\cdot)$, without having to reexamine the raw data. The trick is finding a way to "steal" (the right combination of) pair-correlations from the earlier calculation of $\rho(\cdot)$, until we have accumulated the N_L we need for calculating $\hat{\rho}(l)$. Clearly, we need $N_{1,l}$ of the pair-correlations that "belong to" the calculation of $\rho(1)$, plus $N_{2,l}$ that "belong to" the calculation of $\rho(2)$, etc. Fortunately, we don't need to worry about stealing the correct items from each calculation. If the relative position of the paired items was unchanged by the randomization (i.e., an inbucket pair under external randomization), then $k = l$ and we must steal all the pair-correlations from the calculation of $\rho(k)$. Otherwise, the relative positions of the paired items must have been changed by the randomization, and to remain faithful to a particular sample path we would need to steal certain pair-correlations but not others. However, since the sample Autocorrelation function is already defined as an expected value, we may as well consider average behavior of bucket randomization over all possible ways to shift the bucket boundaries, and all possible outcomes for the randomization. In this case, we can assume each of the $N_{k,l}$ pair-correlations we "steal" contributes the average value, $\rho(k)$, to the calculation of $\hat{\rho}(l)$, so that:

$$\hat{\rho}(l) = \sum_{k=1}^{\infty} \frac{N_{k,l}}{N_l} \rho(k)$$

Rather than attempting to derive the set of coefficients $N_{k,l}/N_l$ directly for each bucket randomization algorithm, we find it convenient to first classify each pair of items $(\hat{X}_i, \hat{X}_j)$ as an inbucket pair if they belong to the same bucket (i.e., $H(i) = H(j)$), or an outbucket pair with an offset of $|H(i) - H(j)|$ buckets, as illustrated in fig. 1(left). Intuitively, since each bucket randomization algorithm affects the inbucket and outbucket pairs differently, it is much easier to calculate a conditional ACF for each classification using the associated pair-correlations, than to tackle the final result directly. After calculating a relative weight for each conditional ACF, we obtain the final ACF as a weighted sum of the conditional ACFs.

1) **ACF for External Randomization:**

- *Conditional ACF:* Under external randomization, the b items from bucket u are moved as unit to a new home in some randomly selected bucket u'. Suppose we pick two items from the externally randomized series, $\hat{X}_i$ and $\hat{X}_j$, which were originally labeled $X_{i'}$ and $X_{j'}$ in the unrandomized series. If the two items belong to the same bucket, then both of them will experience the same label shift of magnitude $i - i' = j - j' = C \cdot b$ for some integer C, and their lag $k = |i - j|$ is unchanged by the randomization. Therefore, the correlations among all inbucket pairs are unchanged by external randomization, and hence **the conditional ACF at lag k using only inbucket pairs is just $\rho(k)$.**

  On the other hand, if the two items belong to different buckets, then each one will experience a different label shift. Since external randomization shuffles the ordering of all m buckets in the series, the offset between their respective new homes will be randomly distributed across the entire length of the randomized sequence — independent of the size of their original offset. Therefore, the correlations among outbucket pairs must be independent of the offset and, providing the the series is sufficiently large, it will also be negligible small, and **the conditional ACF at lag k using only outbucket pairs will be zero everywhere, independent of k.**
- *Bucket Offset as a Function of Lag:* The key to analyzing external randomization is to find the distribution of the offset, $Z(k)$, as we walk through the sequence using all pairs $(\hat{X}_i, \hat{X}_{i+k})$ separated by lag k. Because of the periodicity induced by using deterministic values for both k and b, we obtain the same result by restricting our attention to bucket 0, i.e., $0 \le i \le (b-1)$. Clearly, if $k < b$ then both values from the first $(b-k)$ pairs, i.e., $(\hat{X}_0, \hat{X}_k)$, ..., $(\hat{X}_{b-1-k}, \hat{X}_{b-1})$, will all be selected from bucket zero, and the values in the remaining k pairs will be separated by an offset of exactly one bucket. Conversely, if $k \ge b$ then the two selected values in each pair must belong to different buckets. As the left index i sweeps across positions 0 to $b-1$ of bucket zero, the right index j sweeps across the remainder of bucket $\lfloor k/b \rfloor$ from position k mod b to position $b-1$, followed by the first $(k \text{ mod } b) - 1$ positions of bucket $\lfloor k/b \rfloor + 1$. Thus, the distribution for $Z(k)$ is given by the following:

$$Prob\{Z(k) = z\} = \begin{cases} \frac{b-(k \text{ mod } b)}{b} & z = \lfloor k/b \rfloor \\ \\ \frac{k \text{ mod } b}{b} & z = \lfloor \frac{k}{b} \rfloor + 1 \end{cases}$$

  In particular, note that the probability that $Z(k) = 0$ drops linearly from $(b-1)/b$ to zero as the lag, k, increases from 1 to b. Notice that $Z(k) = 0$ whenever we consider an inbucket pair and $Z(k) > 0$ for whenever we consider an outbucket pair. Thus, the probabilities for $Z(k) = 0$ and $Z(k) > 0$ are exactly the weights we need for summing the conditional ACF for inbucket and outbucket pairs to give us the ACF for external randomization, namely:

$$\hat{\rho}_{EX}(k) = \begin{cases} \frac{b-k}{b}\rho(k) & 1 \le k \le b-1 \\ \\ 0 & k \ge b \end{cases} \tag{4}$$

  Fig. 2(left) illustrates the shape of the conditional ACF for inbucket and outbucket pairs as well as the unconditional ACF. The figure presents curves resulted from both the above analytical formulas and also from experimental data for a fGn series. Note that the analytical results closely match the experimental ACF for all cases.

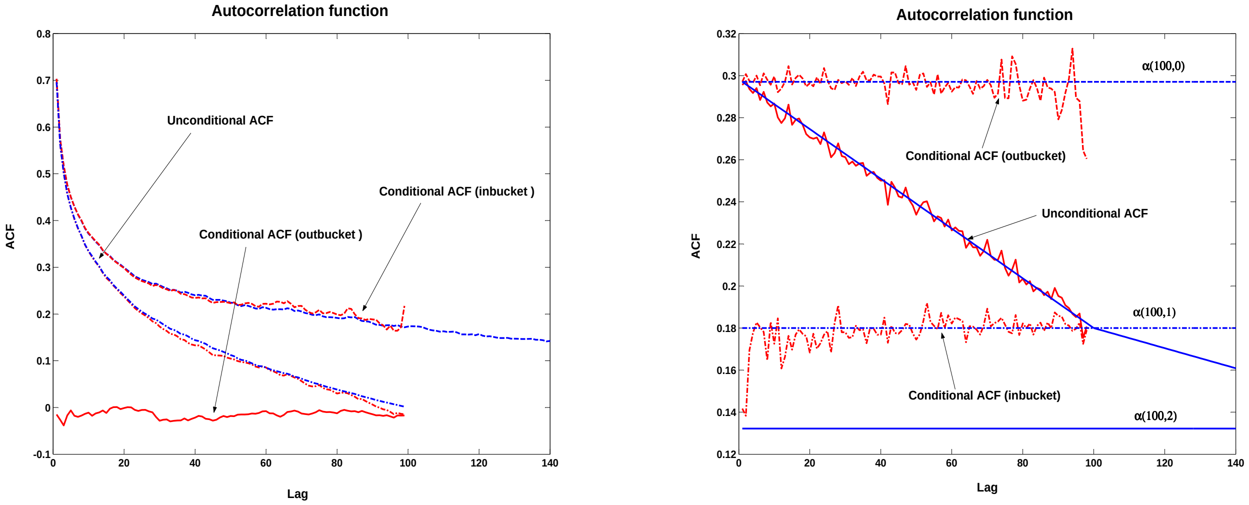

Fig. 2. Detailed view of the ACF under external (LEFT) and internal(RIGHT) randomization. The smooth curves correspond to the theoretical analysis of the randomization, while the more noisy curves to synthesized fGn series.
LEFT: Notice how the conditional ACF for inbucket pairs closely follows the ACF for the original series until it ends at $k = b-1$; it becomes very noisy near the end because much less data is available. The analytical model closely follows the unconditional ACF, and both of them drop smoothly to zero at exactly $k = b$, due to the linear drop in $Prob\{Z(k) = 0\}$.
RIGHT: Notice how the conditional ACF for inbucket pairs closely follows the constant $\alpha(b, 0$ until it ends at $k = b - 1$. Meanwhile the conditional ACF for outbucket pairs (all of which have offset 1 in the measured data) closely follows the constant $\alpha(b, 1)$. Finally, the unconditional ACF follows the analytical model, which is a a straight line dropping from $\alpha(b, 0)$ to $\alpha(b, 1)$ as the lag increases from 1 to b.

2) ACF for Internal Randomization:

- *Conditional ACF for inbucket pairs:* Under internal randomization, each of the b items from bucket u is moved to a random location within the same bucket. Suppose we pick two items from the internally randomized series, $\hat{X}_i$ and $\hat{X}_j$, which were originally labeled $X_{i'}$ and $X_{j'}$ in the unrandomized series. If the two items belong to the same bucket, then each of them will be randomly assigned to a different label from the set $\{u \cdot b, (u \cdot b) + 1, \ldots (u+1) \cdot b - 1\}$, and their pre-randomization lag, $k = |i - j|$ is independent of their current lag, k'. Therefore, the correlations among all inbucket pairs are equalized by internal randomization.

  Since randomly selecting an item from an ordered set is equivalent to deterministically selecting an item from a randomized set, the pre-randomization lag, k', for each inbucket pair must have the same distribution as the distance, d, between two points randomly selected without replacement from a sequence of length b. Since there are $b(b-1)/2$ distinct two-point combinations, but only the following $b - d$ of them (X_0, X_d), (X_1, X_{k+d}), …, X_{b-d-1}, X_{d-1} have distance d, the probability of obtaining distance d is:

$$Prob\{ \text{ distance } = d\} = \frac{2(b-d)}{b(b-1)} \quad 1 \le L \le b-1 \tag{5}$$

  Thus, for every inbucket pair we evaluate in calculating the ACF at lag k for an internally randomized series, we know that its pair-correlation will always have the same value, $\alpha(b, 0)$, which is independent of k and given by:

$$\alpha(b, 0) \equiv \sum_{d=1}^{b-1} \frac{2(b-d)}{b(b-1)} \cdot \rho(d) \tag{6}$$

  (The parameter 0 in equation 6 is introduced for notational consistency, as we will see below, where it represents the bucket offset between the selected items.) Moreover, $\alpha(b, 0)$ must also be the conditional ACF for all inbucket pairs under internal randomization.
- *Conditional ACF for outbucket pairs:* Let us now return to the opposite case, i.e., where the two items $\hat{X}_i$ and $\hat{X}_j$ belong to different buckets. In this case, the bucket offset $Z = H(i) - H(j)$ will remain constant under internal

randomization, but each item experiences a different label shift within its own bucket, for example, from position i' mod b to position i mod b say.

Once again using the result that randomly selecting an item from an ordered set is equivalent to deterministically selecting an item from a randomized set, we recognize that $\hat{X}_i$ could have occupied any of the pre-randomization positions $0, \ldots, b-1$ within block $H(i)$, and at the same time $\hat{X}_j$ could have occupied any of the pre-randomization positions $0, \ldots, b-1$ within block $H(j)$, for a total of b^2 distinct combinations. Notice that b of these combinations give us a lag of exactly $Z \cdot b$, and thereafter the number of combinations giving progressively larger or smaller lags falls off linearly in both directions, until we are left with only one way to generate either the maximum lag of $(Z+1) \cdot b - 1$, or the minimum lag of $(Z-1) \cdot b + 1$. Thus:

$$\begin{aligned} Prob\{ \text{ lag } &= b \cdot Z + L | \text{ values } = m \cdot b, \\ &\text{bucket } = b, \text{ offset } = Z\} \\ &= \frac{b - |L|}{b^2} \quad (-b+1) \leq L \leq b-1. \end{aligned} \tag{7}$$

Just as we saw in the zero offset case, this distribution for the lag applies to every outbucket pair with offset Z we select using this method. Therefore, we have immediately that the conditional ACF outbucket pairs with offset Z is given by:

$$\alpha(b, Z) \equiv \sum_{L=(-b+1)}^{b-1} \frac{b-|L|}{b^2} \cdot \rho(b \cdot Z + L) \tag{8}$$

Unconditional ACF for Internal Bucket Randomization: Each time we select an inbucket pair will be equivalent to randomly selecting two values from a single bucket of size b, and the resulting contribution to $\hat{\rho}_{IN}(k)$ will be $\alpha(b, 0)$, as defined in equation (5). Similarly, each time we select an outbucket pair separated by an offset of Z buckets, the resulting contribution to $\hat{\rho}_{IN}(k)$ will be $\alpha(b, Z)$, as defined in equation (8) independent of k. By combining these results with equation (IV-C.1), we have:

$$\hat{\rho}_{IN}(k) = \begin{cases} \frac{b-k}{b}\alpha(b,0) + \frac{k}{b}\alpha(b,1) & 1 \leq k \leq b-1 \\ \\ \frac{b-(k \text{ mod } b)}{b}\alpha(b, \lfloor k/b \rfloor) & \\ +\frac{k \text{ mod } b}{b}\alpha(b, \lfloor k/b \rfloor + 1) & k \geq b \end{cases} \tag{9}$$

Fig. 2(right) illustrates the curves for the conditional and unconditional ACF for internal randomization. As in external randomization, both theoretical and experimental curves are presented.

3) **ACF after Two-Level Bucket Randomization:** To calculate the ACF after applying two-level bucket randomization to the series, we must distinguish between three levels of separation between the paired values. The analysis is similar in flavor with the two previous approaches, but it is omitted due to space limitations.

V. The Limitations of the Estimators

In this section, we study the performance of the estimators described in section II. To explore their limitations we use different types of time-series. Specifically, section V-B describes the accuracy of the estimators in synthesized long-range dependent time-series. In section V-C we examine how sensitive the estimations are to the presence of periodicities, noise and nonstationarity. Finally, section V-D describes the effect of short-range correlations on the estimation.

A. Overview

We examine three properties of the estimators. First, we test their accuracy on LRD series. Second, we test their ability to discriminate LRD behavior when applied to datasets that are not LRD. Third, we examine how short-range correlations can influence long-range dependence estimation. More specifically, we use three types of time-series: a) synthesized LRD series with known Hurst exponent, b) synthesized non-LRD series, c) randomized LRD series.

Our findings show that there is no definite estimator that could be consistently used in every case. Each estimator evaluates different statistics of the signal to estimate the Hurst exponent. Thus, different processes (e.g. noise,

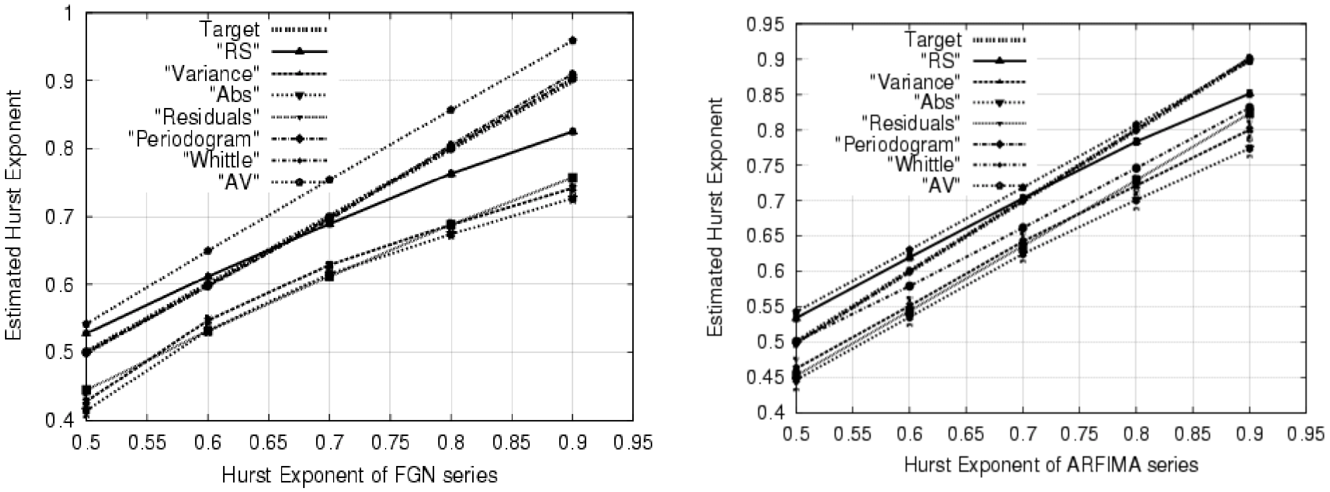

Fig. 3. The performance of the estimators using FGN and fractional ARIMA synthesized time-series. The “Target” line presents the optimal estimation expected by the estimators.

periodicity) have different effect on each estimator. In more detail, our results can be summarized in the following points.

- Estimates of the Hurst exponent vary between the estimators even in synthesized LRD time-series with known Hurst exponent value.
- When the data are generated by fractional Gaussian noise (fGn), Whittle and Periodogram seem to give the most accurate estimation for the Hurst exponent.
- When the data are synthesized using the ARFIMA model, Abry-Veitch, Periodogram and R/S have the best performance.
- Periodicity and noise may obscure the identification of LRD.
- Noise, trend and periodicity can mislead the estimators into erroneously reporting long-range dependence.
- “Frequency-based” estimators (Abry-Veitch, Whittle and Periodogram) depend mainly on short-memory to derive the Hurst exponent estimate.

B. Estimator Accuracy on Synthesized LRD Series:

Each of the estimators was tested against two different types of synthesized long-memory series: a) fractional ARIMA and b) fGn. For each Hurst value between 0.5 and 1 (using a step size of 0.1), we generated 100 fGn and 100 fractional ARIMA synthesized datasets of length 64K (65,536). Fig.3 reports the average of the estimated Hurst value for these datasets for each estimator. Together with the average the 95% confidence intervals are shown. However, the confidence intervals are so close to the average that they are barely discernible. The fGn generator used in this figure is described in [26]. However, these results were verified using also the fGn generator described in [28].

Fig.3 shows that there is significant variation in the estimated Hurst exponent value between the various methodologies, especially as the Hurst exponent tends to 1. Frequency-domain estimators seem to be more accurate. In the case of fGn synthesized series, Whittle and Periodogram estimators fall exactly on top of the optimal estimation line. Note however that Whittle estimator has the a priori the advantage of being applied to a time-series whose underlying structure matches the assumptions under which the estimator was derived. The wavelet Abry-Veitch estimator always overestimates the value of the Hurst exponent (usually by 0.05). Time-domain estimators fail significantly to report the correct Hurst exponent value, underestimating by more 20% as the exponent approaches one. In fig.3, time-domain estimators are represented by the lines clustered significantly under the optimal estimation line. When fractional ARIMA is used to synthesize time-series, estimations are in general closer to the optimal estimation line. However, none of the estimators follows consistently the optimal line across all Hurst values.

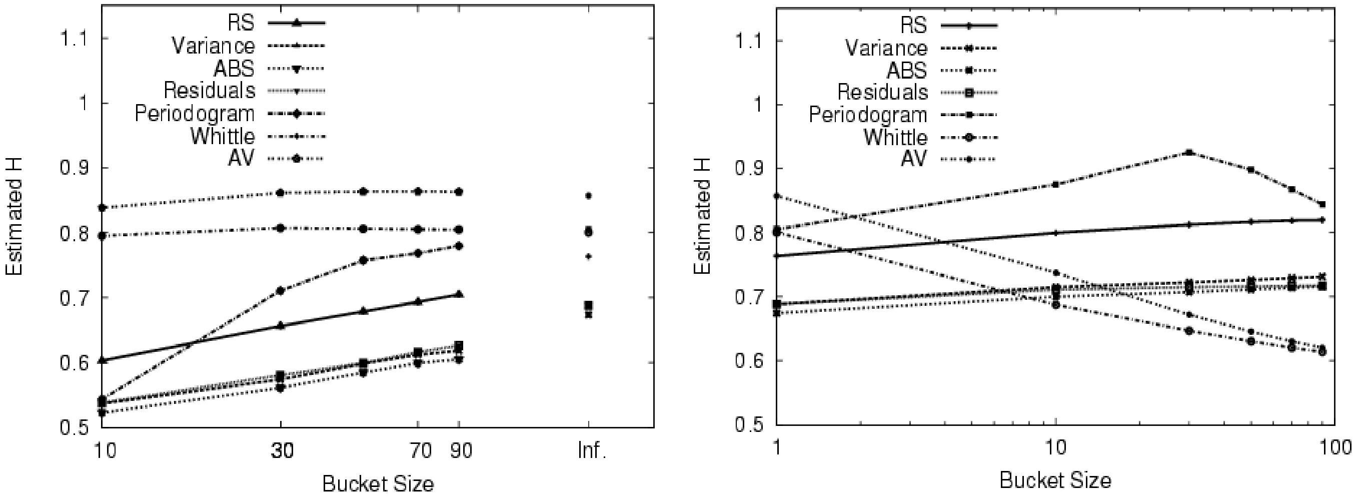

Fig. 4. External (left) and internal (right) randomization with various bucket sizes (10,30,50,70,90). The X axis is presented in log scale. “Frequency based” (AV, Whittle, Periodogram) estimators fail to capture that long-term correlations are distorted in external randomization. Their estimations are the same as before randomizing (the bucket size of the initial ACF can be considered to be infinity in external randomization). In addition, these estimators fail to capture internal randomization. Their estimations after internal randomization deviate significant from the initial estimations (the bucket size of the initial ACF can be considered to be 1 in internal randomization).

C. Discrimination of LRD behavior in LRD and Deterministic Series:

In order to study the sensitivity of the estimations we examined the effect of various phenomena common to time-series analysis, such as periodicity, noise or trend, on the estimators. Our analysis revealed that estimators are significantly affected by the presence of such processes. Furthermore, most of the methodologies fail to distinguish between long-range dependence and such phenomena and falsely report LRD in deterministic nonLRD time-series. We examined the following cases (detailed results for all the following points can be found in [14]):

- *Cosine + white Gaussian noise:* The estimators are applied to periodic datasets. The series are synthesized by white Gaussian noise and the following cosine function : $Acos(\alpha x)$. Periodicity can mislead the Whittle, the Periodogram, the R/S and the Abry-Veitch methods into falsely reporting LRD. The Hurst exponent value depends mainly on A. Increasing A results in the Hurst exponent approaching 1. Especially, if the amplitude is large and the period small, then Whittle always estimates Hurst to be 0.99.
- *fGn series + white Gaussian noise:* We examine the effect of noise on LRD data. All estimators underestimate the Hurst exponent in the presence of noise. However, with the exception of Whittle and the wavelet estimator the difference is negligible. Depending on the signal to noise ratio and the Hurst exponent value of the fGn series, these two estimators significantly underestimated the Hurst exponent by more than 20%.
- *fGn series + a cosine function:* The effect of periodicity in LRD data is considered. Our findings show that periodicity affects all estimations. Depending on the amplitude of the cosine function, time-domain estimators tend to underestimate the Hurst exponent. On the other hand, frequency-based methodologies overestimate the Hurst exponent. As the amplitude of the cosine function is increased, estimates approach one.
- *Trend:* The definition of LRD assumes stationary signals. To study the impact of non-stationarity on the estimators, we synthesized various nonLRD series with different decaying or increasing trends. We also examined combinations of previous categories (white Gaussian noise and/or cosine functions) with trend. In every case only Whittle gives an estimation for Hurst which is always .99. Also the Periodogram estimates Hurst to be greater than 1.

D. Short-Range Correlations

How sensitive are the estimators to short-range correlations? To address this question we used the randomized buckets methodology. Intuitively, the estimations should not be affected after internal randomization. Note that internal randomization breaks the short-term correlations, while preserving the long-term. On the contrary, external randomization should significantly influence estimations, since long-memory is distorted. Estimates of the Hurst exponent in this case should be close to 0.5 since long-range dependence has been removed.

We synthesized numerous fGn series for Hurst exponent values between 0.5 and 1. These synthesized series were then randomized for different bucket sizes. Fig.4 presents our findings. For both figures, the Y axis shows the estimated Hurst value while the X axis presents the bucket size. The estimation of the initial (unrandomized) time-series is represented with bucket sizes of infinity and one for external and internal randomization respectively. The figure shows the mean of 100 estimations of different fGn series with Hurst exponent 0.8 after being externally and internally randomized with bucket sizes ranging from 10 to 90.

Intuitively, after external randomization (Fig.4, left), estimations should be be close to 0.5. This should be especially true for small bucket size. In general, this is the case for all the estimators except AV, Whittle and Periodogram. In particular, estimations by AV and Whittle are not affected by external randomization irrespective of the bucket size. Their estimates are the same with the initial estimation (bucket size of infinity), even though long-range dependence has been eliminated from the series.

Similar counter-intuitive behavior holds for internal randomization. Except AV and Whittle, all estimates of the Hurst exponent value remain relatively constant before and after internal randomization as expected (Fig.4, right). On the contrary, AV and Whittle estimations drop significantly as the bucket size increases!

Considering the effect of randomized buckets on long-memory, we claim that these two estimators — AV and Whittle — depend more on the short-term characteristics of the time-series to derive an estimate for the Hurst exponent. Note, that while these are the most accurate estimators when applied to synthesized LRD series (see V-B, they are also the ones which are mostly affected by short-term correlations.

VI. Towards A Robust Hurst Exponent Estimation

In this section, we present methodologies and algorithms to improve the accuracy and robustness of the estimators. We target the weaknesses of the estimators that we identified in the previous sections. Specifically, we tackle the following issues:

- **The limited number of samples in large scales.** We propose several methods to increase the number of samples at large time scales without affecting the properties of the signal.
- **The effect of periodicity and noise.** We suggest methods to exclude the effect of periodicity and noise in order to increase the accuracy of the estimation.
- **The inaccuracy of regression.** The regression that several estimators use can suffer from statistical artifacts, when the sample points are not equally distributed among the various scales.This is a significant problem in the case of the Periodogram estimator, for which we develop a special "log-averaging" procedure to preprocess the periodogram data before applying the regression method.
- **The effect of short-range correlation.** Several estimators can be mislead by short-range correlations. We propose methods to alleviate these effects and enable the estimators to focus on the long-range correlation, which is of interest here.
- **The effect of nonstationarity.** Stationarity is assumed in the definition of LRD and self-similarity. We propose methods to alleviate the effect of nonstationarity by using a moving average as a reference point in the calculation instead of the global series average.

In most of the above issues, we provide a case study to demonstrate the effectiveness of our approach. We use three datasets of significantly different nature in our case studies:

- **Series 1.** A fractional Gaussian noise series synthesized by the generator by Paxson [26], (LRD with target Hurst exponent 0.9).
- **Series 2.** The well-known trace, *BC-pAug89*, of a local area network link, which started the LRD research [18], and has been widely used since [27] (LRD with target Hurst exponent estimated at 0.8).
- **Series 3.** A recent backbone link trace[2] from January 15, 2003 (10am), by CAIDA [3] on a SONET OC48 (2.5Gbps) link that belongs to MFN, a US Tier 1 Internet Service Provider.

[2] In this trace, we count to the number of bytes per msec for 20 minutes worth of traffic, unless otherwise specified.

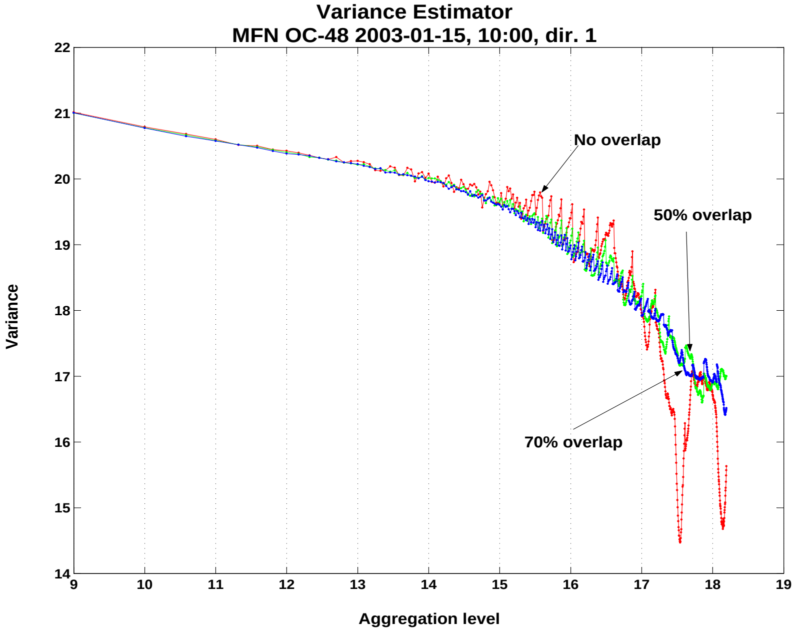

Fig. 5. The effect of using overlapping segments with the variance estimator for a 64K long dataset with H=0.9. The segment length is increased linearly in steps of 512, rather than in powers of two, which gives rise to points that fall between aggregation levels. Notice how increasing the number of segments using overlap reduces the uncertainty at large scales, effectively eliminating the two large dips that appear beyond aggregation level 17 without overlap.

A. *The Limited Number of Samples in Large Scales*

At large scales, we have relatively few samples due to the aggregation, which leads to statistical inaccuracies.

Let us examine the problem in more detail. Most time domain estimators operate on segments of the aggregated time-series $X^{(m)}$. For instance, the variance estimator measures the sample variance across all aggregated data values $\{X^{(m)}\}$ that belong to a given aggregation level. The rate of change of the sample variance as a function of the aggregation level is used to find an estimate for H. Typically, the length of the aggregation segment doubles at each level. Thus, assuming an initial sequence length of $2^{16} = 64K$ values, the second aggregation level will use a segment length of $m = 2^2$ to generate $64K/2^2 = 16K$ aggregated data values at this scale. However, by the time we reach the 14th aggregation level, the segment length has grown to $m = 2^{14}$ and only 4 aggregated data values can be constructed. This lack of samples introduces uncertainty into the calculations associated with large aggregation levels.

We propose two different methods to increase the number of samples at larger time scales.

1) **Segment Overlap***:* In this method, we increase the number of samples available at large scales by allowing segments to overlap. In other words, we decouple the *segment length*, m, from which we generate one aggregated data value from the *shift in the starting point*, s, between adjacent segments. Note that even though it creates some (additional) dependence between adjacent aggregated data values, segment overlapping does not cause any bias in the results because we don't need independent samples to estimate the population variance for $X^{(m)}$. (In the classical confidence interval calculation, the problem with dependent samples is converting the population variance into an estimate of the variance of the mean, which is not required here.)

When overlapping segments are used, our previous definition for the aggregated process must be modified as follows:

$$X_k^{(m,s)} = \frac{1}{m} \sum_{i=ks}^{ks+m-1} X_i, \quad k = 0, 2, ..., \left\lfloor \frac{N-m}{s} \right\rfloor . \tag{10}$$

For $s = m$, adjacent segments are disjoint and the definition reduces to Eqn. 2. For $s = \frac{m}{2}$, there is 50% overlap between adjacent segments. In this case, the aggregate process includes $\frac{N-m}{m/2} + 1$ segments, which is one fewer than twice the number of segments we would have had without overlap.

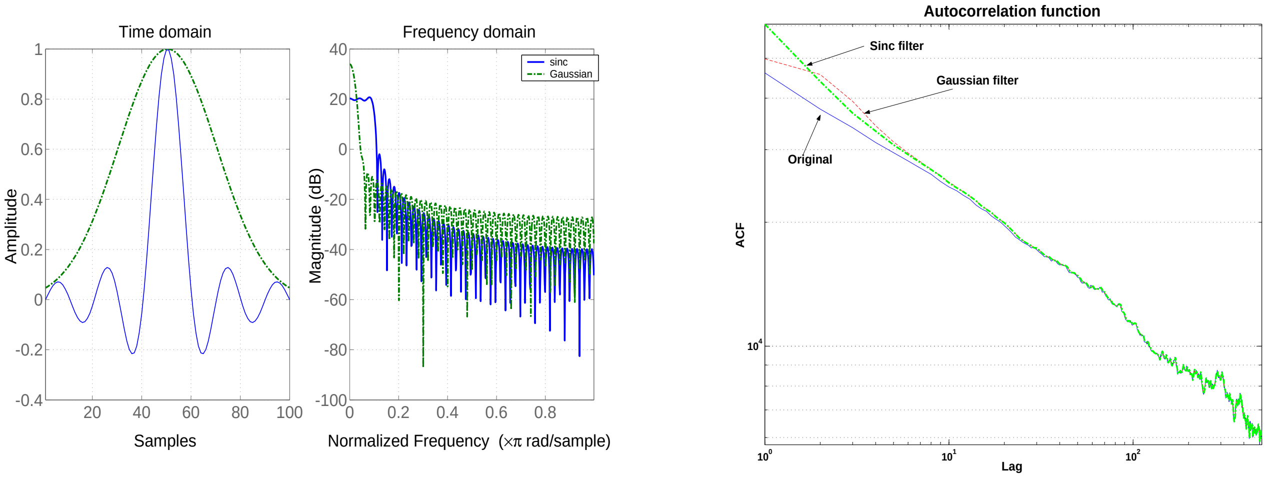

Fig. 6. LEFT: Gaussian (dotted line) and sinc (solid line) filters and their responses in the frequency domain. RIGHT: ACF after 4-times oversampling and low-pass filtering (log-log scale).

Case Study: The effectiveness of segment overlap. The effect of overlapping is illustrated in fig. 5. The figure shows the variance estimator for Series 3, the backbone link, starting from aggregation level 9 (i.e., $m = 512 = 2^9$). Without overlap, we observe that the estimated values vary significantly; observe the two dips at the right most corner of the plot at large scales. The reason is that the few samples at large scales make the estimation statistically sensitive. On the other hand, estimations with overlap of 50% and 70% are considerably smoother.

This figure also highlights the dichotomy in the scaling behavior of Internet traffic in agreement with previous observations (e.g., [42]). The scale of change for the Hurst exponent is within a few hundred milliseconds. The exponent shifts from approximately 0.62 to 0.8 from small to large scales. However, the characteristics of the scaling of Internet traffic are beyond the scope of this paper.

2) **Oversampling***:* Oversampling involves increasing the sampling rate for a signal using interpolation to generate redundant data points between the original measurements. These extra samples can be used to reduce aliasing artifacts. In particular oversampling is widely used in digital audio [7], to significantly improve audio quality by increasing the signal-to-noise ratio, and in computer graphics, to prevent the edges of objects from appearing jagged [20].

In time series analysis, oversampling can be used to increase the number of samples available at each aggregation level, and hence to produce a more robust estimate of the range and variance of these samples. Note that, by definition, oversampling by a factor of r means reducing the sampling interval to $1/r$th of its original value. Thus, the state of the original process at step t corresponds to the state of the oversampled process at step rt.

Starting from the series $\{X_t\}$, we can generate the corresponding r-times oversampled series $\{\hat{X}_t^{(1/r)}\}$ in the following way:

1) We expand the series by a factor of r via *duplicate insertion* to create $\{X_t^{(1/r)}\}$, where we have replaced each sample point by a block of r consecutive sample points that share the same value.[3]
2) We apply *digital low-pass filtering* to the expanded sequence to smooth out the sharp corners created by duplicate insertion. In this case, each point in the expanded series, $X_t^{(1/r)}$, is replaced by a moving average of nearby points, $\hat{X}_t^{(1/r)}$, using a symmetrical bell-shaped set of weights.

Filtering can be performed by various low-pass filters, such as Gaussian or sinc filters. Fig. 6 (left) shows a Gaussian window and a sinc function and their responses in the frequency domain. The sinc function has a value of 1 where the offset x is zero, and a value of $\frac{sin(\pi x)}{\pi x}$ for all other x.

It is important to note that neither oversampling nor low-pass filtering alter the characteristics of the time-series that

[3] Zero-value insertion is commonly used in the signal processing literature, rather than the duplicate-insertion we describe here. The two methods differ only by a linear scale factor, and with duplicate-insertion the sample mean is unchanged by oversampling, rather than decreasing by a factor of r if zero-value insertion had been used.

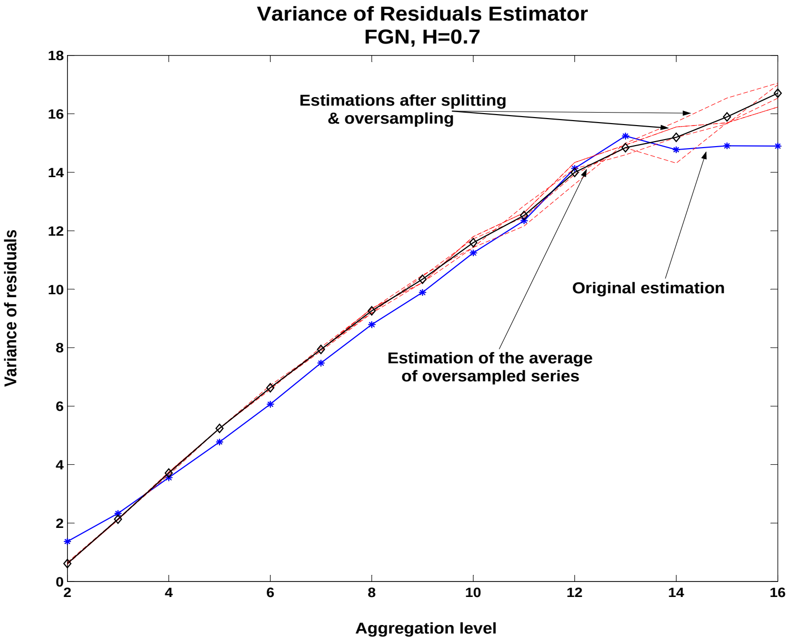

Fig. 7. The variance of residuals estimator applied to an anomalously-bad fGn sequence produced by Paxson's generator. Even though the data set has somehow completely confused the estimator at high aggregation levels, the average results across $D = 4$ oversampled sub-series are quite reasonable.

describe its long-term behavior. We use Series 1, fGn, to demonstrate this result in Fig. 6 (right). Here we show a log-log plot of the autocovariance function of the fGn series: a) without oversampling and b) after 4 times oversampling and low-pass filtering using either the Gaussian or sinc filters. Note that we use the (unnormalized) autocovariance function to make it easier to compare the curves because low-pass filtering shifts the (normalized) autocorrelation function by reducing the variance. In addition, the lag and the coefficients of the 4-times oversampled curves have been are divided by 4 to correct for the change in length for the sampling interval. It is clear from this figure that the ACF is not affected by oversampling, and that low-pass filtering only affects the ACF at small lags that fall within the window of the filter.

Since our primary goal is to determine the behavior of the process X_t over large scales, and oversampling generates additional intermediate points without extending the overall range covered by the series, this "obvious" method for applying oversampling does not look very promising. However, consider what happens if we first distribute the elements of X_t among D interleaved sub-series in round-robin ordering. In this case, we define the elements of the dth sub-series, $0 \leq d < D$, as:

$$Y_{t,d} = X_{Dt+d}, \quad t = 0, 1, \ldots .$$

Note that the sampling interval for $\{Y_{t,d}\}$ has been increased by a factor of D relative to the sampling interval for $\{X_t\}$, but each sub-series covers the same overall range as the original series. Thus, if we now apply D-times oversampling to each sub-series, we obtain D distinct oversampled sub-series, $\{\hat{Y}_{t,d}^{(1/D)}\}$, each of which shares the sampling interval and covers the same overall range as the original series, $\{X_t\}$. Since the distinctions among different sub-series are due to small time shifts in the sampling interval boundaries, they must represent short-range properties of the original series. Hence, by *averaging* the results obtained from different sub-series we can identify and/or reduce the impact of short-range properties on our analysis. To summarize, the final methodology is as follows:

1) Distribute the elements of X_t among D interleaved sub-series, $Y_{t,d}$, $0 \leq d < D$.
2) Apply duplicate insertion to each sub-series by a factor of D to create the expanded sub-series $Y_{t,d}^{(1/D)}$.
3) Smooth each of the expanded sub-series using digital low-pass filtering to create the expanded and smoothed sub-series $\{\hat{Y}_{t,d}^{(1/D)}\}$.
4) Separately analyze each of these sub-series, and average the results.

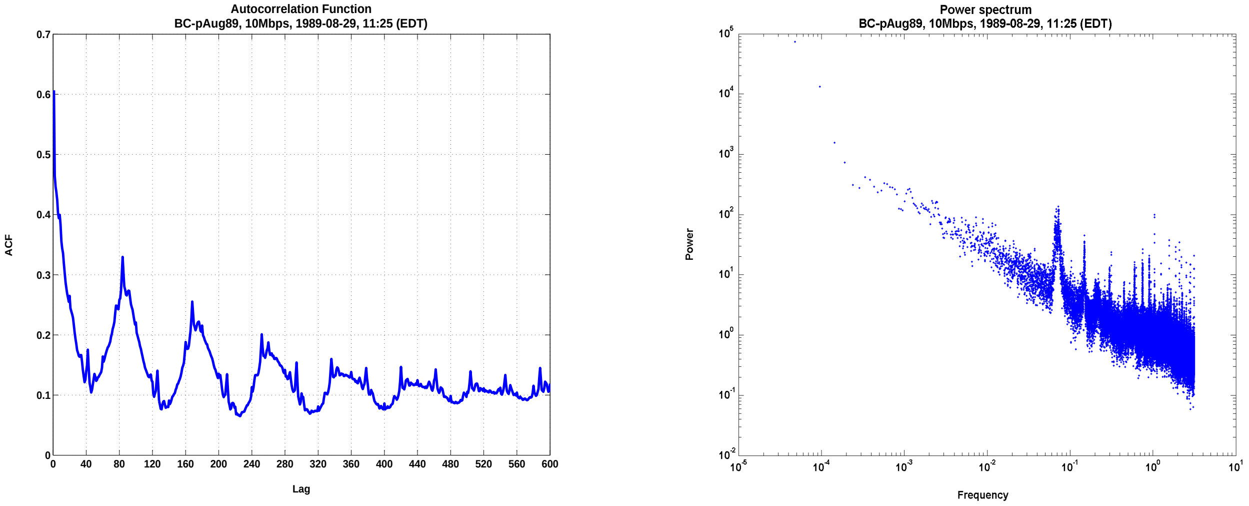

Fig. 8. ACF and power spectrum (log-log scale) for the BC-pAug89 trace. There is a strong periodic component at 420ms. Periodicities are also observed at higher frequencies.

Case study: The effectiveness of oversampling. Fig. 7 shows the results of applying our oversampling approach to the variance of residuals method. The original time series is synthesized fGn produced with Paxson's generator. In this case, however, we present the results generated by an anomalously-bad random seed, which somehow managed to completely confuse the variance of residuals algorithm at high aggregation levels. However, the results obtained by applying the variance of residuals algorithm to the $D = 4$ oversampled sub-series are far better, and their average is quite good. Thus, we see that by allowing us to average the results obtained by applying a Hurst estimator to D distinct sub-series, our oversampling methodology can reduce its sensitivity to anomalies in the data.

B. *The Effect of Periodicity and Noise*

Section V demonstrated how periodicity and noise can affect the estimation of Hurst exponent. This section presents effective methodologies to alleviate the effects of such phenomena.

1) **Periodicity***:* We propose two different ways of removing the periodicity: a) internal randomization with bucket size larger than the period, and b) stop-band filtering to filter out the frequencies of the observed periodicity [22].

Case study. Removing the periodicity can result in more robust and more accurate estimations of the Hurst exponent. To demonstrate these methodologies, we use Series 2, the LAN data trace. This data sets has strong periodic components, which we see by examining the power spectrum and autocorrelation function of this series. These periodicities are mainly results of request-response protocols of Ethernet traffic [33] [12]. Fig. 8 shows the autocorrelation function and the power spectrum of packet arrivals. Similarly, fig. 9 shows the ACF and power spectrum after removing the periodic components with internal randomization and filtering. Internal randomization with window 100 smoothes the power spectrum. On the other hand, a stop-band filter is used to filter out two certain bands of frequencies that contribute most to the periodicity. A *hamming* window [22] was used for the filtering; however, different windows have similar effects. Removing the periodicity results in an increase of the Hurst exponent from approximately 0.8 to 0.86 for all the time-domain and Whittle estimators, and from 0.87 to 0.93 for periodogram and Abry-Veitch estimators.

2) **Noise***:* Noise affects the accuracy of the Hurst exponent and mainly Whittle and Abry-Veitch estimators (see section V-C). The reason is that noise distorts the higher frequencies of the power spectrum. Thus, low-pass filtering and internal randomization with a small window should denoise the signal. However, because filtering distorts the original shape of the power spectrum, Whittle and Abry-Veitch estimators should consider only the low frequencies of the spectrum in their estimations.

Case study: The previous subsections as well as section IV present the results of low-pass filtering and internal randomization. Thus, further examples are not shown in this case.

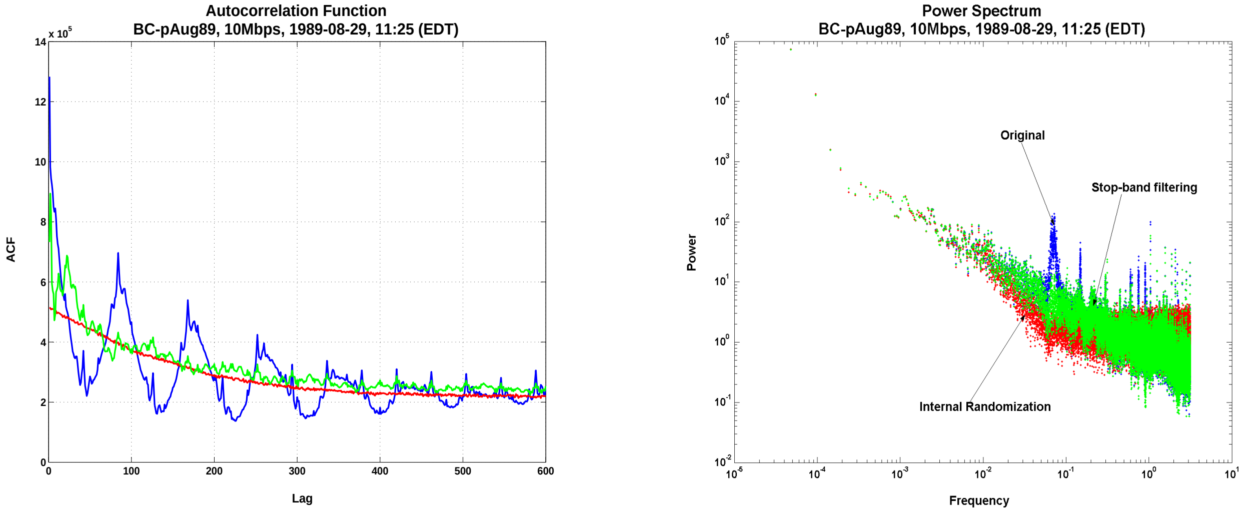

Fig. 9. ACF and power spectrum after removing periodicities with internal randomization and stop-band filtering from the BC-pAug89 trace.

C. *The Pitfalls of Linear Regression*

All estimation methodologies use regression to obtain a “best fit” estimate of the Hurst exponent. However, minimizing the mean squared error across all sample points may be counterproductive if all sample points are not equally important. This is especially true in the case of the periodogram, where the vast majority of the sample points represent high frequencies whereas for evaluating LRD properties we must estimate the slope at low frequencies. This problem is further confounded by the fact that high frequency power estimates are inherently very noisy. Indeed, increasing the length of the data sequence merely allows us to compute noisy power estimates over a broader range of frequencies, without reducing the error on any of them. And, finally, the very act of estimating the best-fit straight line through a logarithmically transformed data set needs to be approached with extreme caution. For example, suppose the data set contained only two points, with respective values $2^1 = 2$ and $2^9 = 512$. Clearly, the best single-point approximation that minimizes the mean square error is simply the arithmetic average of the two data points, namely $257 \approx 2^8$. However, if we attempt the same calculation using the logarithmically transformed data set, then we would erroneously select $2^5 = 32$ as the “best fit” approximation by averaging the exponents of the two data points.

For a more robust estimate of its slope, the periodogram should be averaged to eliminate the triangular “cloud” of points at high frequencies. A number of parametric and non-parametric methodologies have been proposed to smoothen the power spectrum [29], but they do not seem to be widely known nor used routinely. Non-parametric methodologies are based on either separately evaluating the periodogram over at different segments of the series and averaging the results (i.e., the Welch method), or on windowing the time-series or the autocorrelation function before estimating the power spectrum.

We propose an aggregation methodology that creates a *“log-average”* representation of the periodogram, using a sequence of variable-size buckets to account for the difference in the number of samples between lower and higher frequencies. More specifically, if $\{I(\nu_k)\}$ is the periodogram for the time series,[4] $\max\{\nu\}$ and $\min\{\nu\}$ represent upper and lower bounds on the desired range of periodogram frequencies, and B is the desired number of buckets, then the low-frequency cutoff for bucket b, $\nu^*(b)$, is defined as:

$$\begin{aligned}
\nu^*(0) &= \min\{\nu\} \\
\nu^*(B) &= \max\{\nu\} \\
\nu^*(b) &= \frac{\nu^*(b-1)}{\delta}, \qquad\qquad b = 1, 2, \ldots, B-1
\end{aligned}$$

[4] Actually, it is customary to restrict $\{I(\nu)\}$ to include only the bottom 10% of the available frequencies from the entire periodogram, because they occupy 90% of the entire frequency range due to the $1/k$ progression of frequencies, and because the long-range scaling behavior of the process is described by frequencies closer to zero.

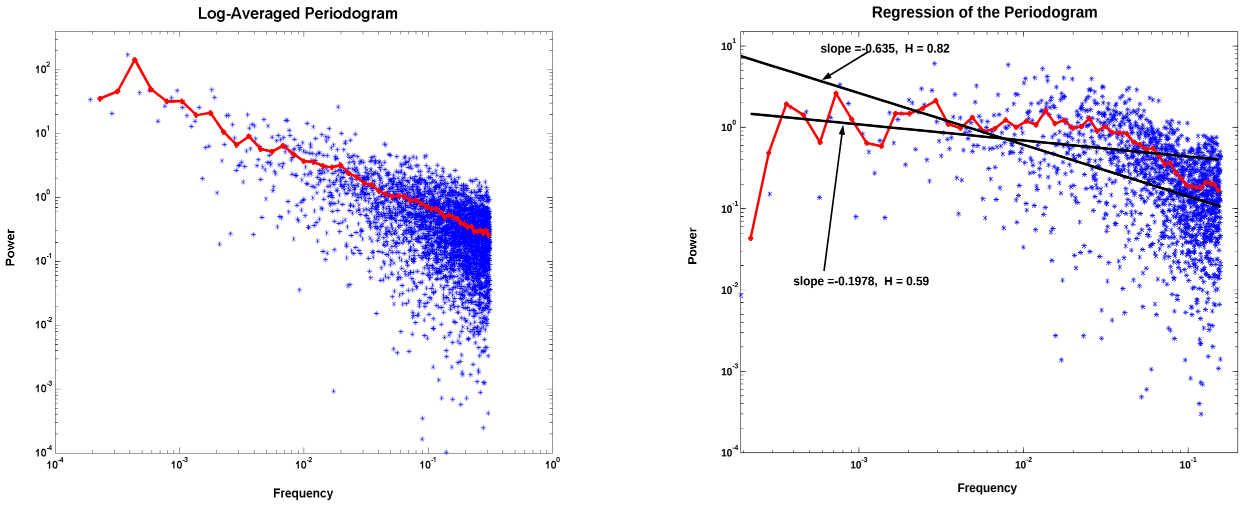

Fig. 10. Left: Log-averaging smoothes the periodogram for a fGn series dramatically, making it very easy to estimate its slope. RIGHT: Regression of the periodogram with and without log-averaging, after the fGn time-series has been externally randomized to remove long-range correlations. Note that without log-averaging, the Hurst exponent estimation falsely identifies long-range dependence.

$$\delta \quad = \quad e^{\ln\left(\frac{\min\{\nu\}}{\max\{\nu\}}\right)\cdot\frac{1}{B}}.$$

Later on, we will represent each bucket by its midpoint frequency, $\nu^m(b) = \frac{\nu^*(b)+\nu^*(b+1)}{2}, b = 0, \ldots, B-1$. To "fill" each bucket, recognize that each term in the periodogram, $I(\nu_k)$, is an estimate of the average power across a band of frequencies that starts half way to the previous frequency, ν_{k-1}, and ends half way to the next frequency, ν_{k+1}, which therefore represents a frequency band of width:

$$\frac{\nu_k - \nu_{k-1}}{2} + \frac{\nu_{k+1} - \nu_k}{2} = \frac{\nu_{k+1} - \nu_{k-1}}{2}.$$

Hence, we can estimate the average power in bucket b by summing the total power contributed by each of its constituent frequencies and dividing by its bucket width to obtain:

$$I(\nu^m(b)) = \frac{1}{2} \sum_{\nu^*(b) \le \nu_k < \nu^*(b+1)} I(\nu_k) \frac{\nu_{k+1} - \nu_{k-1}}{\nu^*(b+1) - \nu^*(b)},$$

for all $b = 0 \ldots B-1$. The final step is to exchange approximately 9% of the averaged spectrum for each bucket with its immediate neighbors to account for leakage of power between nearby frequencies.[5]

Case study: The effectiveness of periodogram averaging. In this experiment, we use the power of randomized buckets to demonstrate the effectiveness of our improved regression method. We use Series 1, synthesized fGn with target Hurst exponent 0.9, but *modified* through the application of external randomization with bucket size 20, to create a series that has *short-range dependencies only*. We show that the basic periodogram estimator erroneously reports LRD even in the modified signal, while our log-averaging method is able to detect the absence of LRD.

In Fig. 10 (left), the raw periodogram for the unmodified version of Series 1 appears as a triangular "cloud" of points that are very widely scattered for $\nu > 10^{-2}$. On the other hand, the log-averaged periodogram with 50 buckets is shown as a polygonal line, which shows a fairly straight linear trend except at the very lowest frequencies. Clearly, log-averaging eliminates the large variations between samples individual periodogram samples, making it very easy to estimate the Hurst exponent.

Fig. 10 (right) shows the results of substituting the modified version of Series 1 after external bucket shuffling. In this case, the raw periodogram still looks like a triangular "cloud" of points, although it is obvious that the left-hand

[5] This is an artifact of using a rectangular window to define the boundaries of each frequency bucket, and means the final value for $I(\nu^m(b))$ will be approximately 81% of the average power calculated above, together with 9% of $I(\nu^m(b-1))$ and $I(\nu^m(b+1))$.

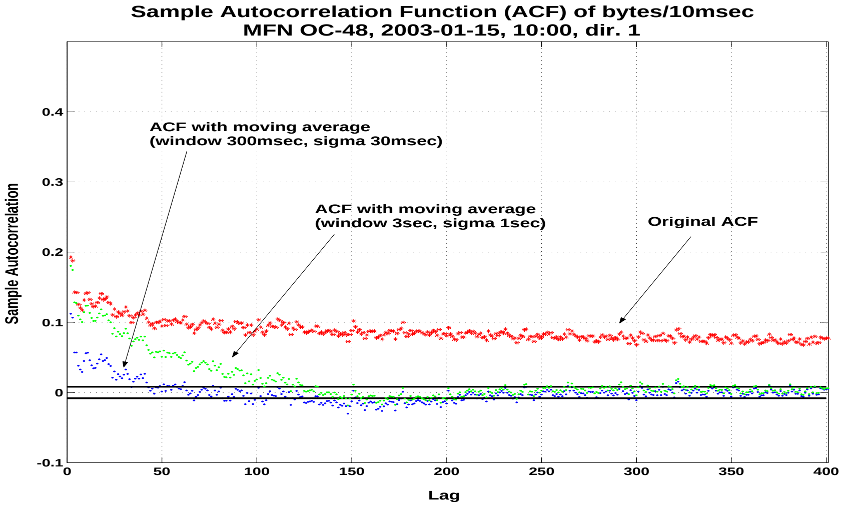

Fig. 11. ACF calculated with global mean and moving average mean based on a Gaussian window. Correlation coefficients drop to zero after first lags when local nonstationarities are removed with moving average.

tail has been bent downwards. If the periodogram method is applied to the raw data, the resulting estimate of the Hurst exponent is almost unchanged by external bucket shuffling because the regression is dominated by the large number of higher frequency samples. On the other hand, applying the periodogram method to the log-averaged data produces a Hurst exponent estimate that is much closer to its true value of 0.5.

D. *The Effect of Short-Range Correlation*

We propose to alleviate the effect of short-term correlations or high frequency phenomena with the use of: a) internal randomization from the randomized buckets method, or b) low-pass filtering. Short-range correlation can bias the estimators towards reporting long-range dependence as we have seen earlier (section V-D). Removing short-range correlation does not affect the shape of the autocorrelation function, if the time-series is truly LRD. The bucket size of the internal randomization or the band of frequencies to be removed depend on how short-range is defined on the specific process. Especially, suggestions such as the one in [15], where the Hurst exponent is estimated using explicitly only short-range correlations are bound to produce misleading results in any realistic dataset. This approach is based on solving Eqn. 3 for $k = 1$, and thus estimating the Hurst exponent using only the magnitude of lag-one correlations! This algorithm can *only* be applied if the process is known to be LRD and agree with fGn.

Case study: In our previous analysis, we have demonstrated that estimation based on small lags of the autocorrelation function can lead to erroneous conclusions (see section V-D). Examples of the ACF or power spectrum after removing short-range correlations should be in agreement with figures described in section IV and the power spectrum after low-pass filtering in previous subsections and thus are not repeated here.

E. *The Effect of Non-stationarity*

Nonstationarity can be misinterpreted as long-range dependence, since all estimators assume stationarity to estimate the Hurst exponent. Because of the use of a global mean both in the calculation of the autocorrelation function and also in the estimators distant events in time may seem highly correlated; however, this is only a side-effect of the fluctuations of the mean of the series with time. Recall from Eqn. 1 that the ACF expresses the relative dependence between values of a process using the distance from the global mean μ. To alleviate this effect, we propose the use of moving average, instead of the global average of the process, if nonstationarity is suspected. In more detail, non-stationarity effects are removed from the estimation of the ACF by replacing the *global* average used Eqn. 1, by a *moving average* calculated using a Gaussian window.

Case Study: Using a moving average improves the accuracy of the estimation if nonstationarity exists. Not accounting for nonstationarity can erroneously lead to detection of LRD in a nonLRD series. We illustrate this by studying the autocorrelation function of Series 3, the backbone link. In Fig. 11, we plot the autocorrelation function calculated first using Eqn. 1, and second by using moving average, with a Gaussian window of 300msec and 3sec. The initial method shows high correlation for large lags suggesting LRD. However, when nonstationarity is removed with the moving average windows, correlations become insignificant beyond lags of 150. Namely, they fall within the 95% confidence interval of zero. Thus, by using the moving average to remove nonstationarities, we conclude that Series 3 does not exhibit LRD. The lag where each of the two moving-average ACF curves in Fig. 11 drops to zero, is also related to the the standard deviation (sigma) of the Gaussian window, and also on its size. Thus, we must be careful when calculating the moving average of the process not to use too small or too large Gaussian window. A small Gaussian window will result in removing important characteristics of the process besides the nonstationarity. On the oher hand, too large a Gaussian window will have no effect on removing nonstationarities since it will be too smooth. Thus, visual inspection of the Gaussian moving average is required to avoid artifacts caused by too large or too small window.

VII. Towards a Systematic Approach

This section binds together all the previous pieces of this paper in one unified methodology to facilitate the detection of LRD by providing a series of steps and practical tips. Recall that one of our goals is to make LRD analysis: a) systematic, and b) accessible to practitioners. We use the term “systematic” to refer to several important traits such as: a) well defined, b) robust, and, c) repeatable.

To the best of our knowledge, this is the first effort towards a systematic approach. In addition, all the functionality below is supported by our software tool SELFIS [13]. Therefore, our tool and the steps below provide a manual to nonexpert inclined practitioners. Our tool has already been downloaded by approximately 150 researchers around the globe spanning multiple disciplines from networking to social and biological sciences. This shows the need for such a systematic approach.

We suggest the following steps to a practitioner, which we group into four phases. Given a time series, the problem is to detect and quantify the extent of the dependence.

Phase 1. Data inspection and processing. This phase is the equivalent of preprocessing. We propose to divide, if possible, the series into different components and study each one separately. It is in general more comprehensive and manageable to characterize components than the whole complex system.

- *Visually inspect the series at several different scales.* One can identify interesting properties or patterns that can help on the interpretation of results.
- *Decompose the series: detect and isolate periodicity, noise, and trend.* Such phenomena “confuse” the estimators.

Phase 2. Testing the existence of LRD. We want to answer ”yes” or ”no” with high confidence.

- *Run all estimators.* Each estimator provides a different perspective.
- *Focus on ”detection” estimators.* Note that the Whittle estimator presupposes the existence of LRD. Thus, its estimates are reliable only if LRD really exists.
- *Visually inspect the graphical output of the estimators.* This can often reveal problem areas that statistics such as the correlation coefficient of the regression cannot reveal. First, we can identify periodicities or multi-scale behavior, i.e., knee in the plot. This may indicate different behavior at different scales. Second, the accuracy of the estimation may significantly be affected by outliers at small or large scales.
- *If estimators cannot statistically support LRD, it is likely that the series is not LRD.*
- *Stress-test the existence of LRD with randomized buckets.* We recommend running several experiments with different randomization types to remove long-, medium-, or short-range correlations. LRD properties should remain unaffected by internal shuffling, and disappear in external shuffling.

Phase 3. Determining the exact Hurst value. Assuming that the previous steps detect LRD, we want to find and report an accurate estimation of the Hurst exponent.

- *Use all estimators, but now focus on the most reliable ones.* The wavelet and periodogram estimators are in general the most reliable if periodicities have been removed. Whittle is accurate provided the process agrees with fGn.
- *Use the proposed techniques to overcome sampling related problems.* If the number of samples is limited, oversampling can significantly increase the accuracy.

Phase 4. Reporting the results. Report the three types of information: a) the estimators used, b) the Hurst values, and c) their statistical confidence. We claim that failure to report any of these components, makes the reporting so incomplete that it becomes meaningless. This is critical to make research efforts repeatable and comparable.

The above steps provide a first systematic approach to explore and report LRD. It isolates most of the factors that lead to "confusion" such as periodicity, trend, and short-range correlations. However, the methodologies described in this paper cannot guarantee that there will not be any gray areas. The goal is to at least eliminate a number of "indecisive" instances. Our experience with LRD analysis revealed that this approach was able to provide a definitive result for most cases.

VIII. Conclusions

Our overarching goal is to popularize LRD analysis by making it conceptually accessible to and easy to use by engineers and practitioners. To this effect, we develop the basis for a systematic approach to LRD estimation. The absence of such a systematic approach has hindered the use of LRD. It is nontrivial for nonexperts to master the necessary background in order to develop the tools, and to interpret the results correctly.

In this paper, we identify the weaknesses of the current estimators and propose solutions towards a robust estimation of long-range dependence. In more detail, the work has two main thrusts.

The limitations of current LRD estimators. First, we provide an extensive documentation of the weaknesses of the current estimators. We show that sometimes estimators can mistakingly report LRD. Then, we go a step further and identify the causes of this misidentifications and inaccuracies. We find that the problem lies either in the limitations of the statistical methods or in properties of the signal, such as periodicity, which obfuscate the analysis.

Towards a robust LRD estimation. To alleviate these problems, we provide mechanisms to improve the statistical procedures in calculating LRD. Among those, we highlight the randomized buckets as an excellent stress-testing tool, that can isolate the effect of short-, medium- and long-term correlations. We demonstrate the effectiveness of our methods with experimental data. In several cases, our techniques detect correctly the absence of LRD, unlike currently widely used methods, which are mislead by statistical errors and artifacts.

As the culmination of this work, we provide the first how-to manual for practitioners. We list a set of steps to guide nonexperts in the correct use and interpretation of the estimation methods.

As a practical contribution, we develop and provide the SELFIS tool, which incorporates most of the functionality required in LRD analysis. The popularity of our tool so far has surpassed our expectations and verified beyond a doubt the need for such a tool. Our vision is to establish SELFIS tool as a point of reference, if not the de facto standard, for LRD analysis.

In conclusion, long-range dependence is a powerful mathematical tool, which has already revolutionized network and system modeling. However, ambiguities and misinterpretations can dilute its impact and its scientific credibility. Our work is a first step in developing a standardized methodology and common tools in order to help LRD analysis reach its full potential as a modeling tool.

Acknowledgments

We wish to thank Dr. R.H. Riedi for thorough discussions on the concepts of long-range dependence and self-similarity and their relevance to networking. In addition, thanks are also due to CAIDA and in particular to Andre Broido, Margaret Murray and kc claffy for making available the backbone trace which was used in the last part of the paper and for their helpful insights on characteristics of Internet backbone traffic.

References

[1] P. Abry and D. Veitch. Wavelet Analysis of Long-Range Dependence Traffic. In *IEEE Transactions on Information Theory*, 1998.

[2] J. Beran. *Statistics for Long-memory Processes*. Chapman and Hall, New York, 1994.

[3] CAIDA. Cooperative Association for Internet Data Analysis. http//www.caida.org/.

[4] J. Cao, W. Cleveland, D. Lin, and D. Sun. Internet Traffic Tends to Poisson and Independent as the Load Increases. *Bell Labs Technical Report*, 2001.

[5] J. Cao, W. Cleveland, D. Lin, and D. Sun. On the Nonstationarity of Internet Traffic. *SIGMETRICS/Performance*, 2001.

[6] M. E. Crovella and A. Bestavros. Self-Similarity in World Wide Web Traffic Evidence and Possible Causes. In *IEEE/ACM Transactions on Networking*, 1997.

[7] G. Erickson. A Fundamental Introduction to the Compact Disc Player, 1994. http://www.tc.umn.edu/ erick205/Papers/paper.html.

[8] A. Erramilli, O. Narayan, and W. Willinger. Experimental Queueing Analysis with Long-Range Dependent Packet Traffic. *IEEE/ACM Transactions on Networking*, 4(2):209–223, 1996.

[9] A. Feldmann, A. C. Gilbert, W. Willinger, and T. G. Kurtz. The Changing Nature of Network Traffic: Scaling Phenomena. In *ACM Computer Communication Review*, volume 28, pages 5–29, 1998.

[10] M. Grossglauser and J. Bolot. On the Relevance of Long-Range Dependence in Network Traffic. In *IEEE/ACM Transactions on Networking*, 1998.

[11] N. Hohn, D. Veitch, and P. Abry. Cluster Processes, a Natural Language for Network Traffic. *IEEE Transactions on Networking*, 2003.

[12] R. Jain and S. Routhier. Packet Trains: Measurements and a New Model for Computer Network Traffic. In *IEEE J. Select. Areas Commun.*, volume 4, pages 1162–1167, 1986.

[13] T. Karagiannis and M. Faloutsos. SELFIS: A Tool For Self-Similarity and Long-Range Dependence Analysis. In *1st Workshop on Fractals and Self-Similarity in Data Mining: Issues and Approaches (in KDD)* , Edmonton, Canada, July 23, 2002.

[14] T. Karagiannis, M. Faloutsos, and R.H. Riedi. Long-Range dependence:Now you see it, now you don't! In *IEEE GLOBECOM, Global Internet Symposium*, Taipei, Taiwan, 2002.

[15] H. Kettani and J. A. Gubner. A Novel Approach to the Estimation of the Hurst Parameter in Self-Similar Traffic. In *IEEE Conference on Local Computer Networks*, Florida, Novermber, 2002.

[16] L. Kleinrock. *Queueing Systems, Vol. II Computer Applications*. Wiley, New York, 1976.

[17] M. Krunz. On the limitations of the variance-time test for inference of long-range dependence. In *IEEE INFOCOM*, pages 1254–1260, 2001.

[18] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. On the Self-Similar nature of Ethernet Traffic. In *IEEE/ACM Transactions on Networking*, 1994.

[19] M. Liljenstam and A. T. Ogielski. Crossover Scaling Effects in Aggregated TCP Traffic with Congestion Losses. In *ACM Computer Communication Review*, 2002.

[20] D. P. Mitchell. Generating Antialiased Images at Low Sampling Densities. pages 21(4):65–72, 1987.

[21] S. Molnar and T. D. Dang. Pitfalls in Long Range Dependence Testing and Estimation. In *GLOBECOM*, 2000.

[22] A.V. Oppenheim and R.W. Schafer. *Discrete-Time Signal Processing*, pages 447–448,311–312. Prentice-Hall, 1989.

[23] K. Park, G. Kim, and M. E. Crovella. On the Effect of Traffic Self-Similarity on Network Performance. In *Proceedings of SPIE International Conference on Performance and Control of Network Systems*, 1997.

[24] K. Park, G. Kim, and M.E.Crovella. On the Relationship Between File Sizes Transport Protocols, and Self-Similar Network Traffic. In *International Conference on Network Protocols*, pages 171–180, Oct 1996.

[25] K. Park and W. Willinger. Self-similar network traffic: An overview. In *Self-Similar Network Traffic and Performance Evaluation*. Wiley-Interscience, 2000.

[26] V. Paxson. Fast approximation of self similar network traffic. Technical Report LBL-36750, 1995.

[27] V. Paxson and S. Floyd. Wide area traffic: the failure of Poisson modeling. *IEEE/ACM Transactions on Networking*, 3(3):226–244, 1995.

[28] E. E. Peters. *Chaos and Order in the Capital Markets*, page 211. John Wiley & Sons, New York, 1991.

[29] J. G. Proakis and D. G. Manolakis. *Digital Signal Processing : Principles, Algorithms and Applications* , pages 896 – 960. Prentice Hall, N.J., 1996.

[30] R. H. Riedi, M. S. Crouse, V. J. Ribeiro, and R. G. Baraniuk. A Multifractal Wavelet Model with Application to Network Traffic. In *IEEE Special Issue on Information Theory*, pages 992–1018, 1999.

[31] Z. Sahinoglu and S. Tekinay. On Multimedia Netowkrs: Self-similar Traffic and Network Performance. In *IEEE Communications Magazine*, volume 37, pages 48–52, 1999.

[32] S. Sarvotham, R. Riedi, and R. Baraniuk. Network Traffic Analysis and Modeling at the Connection Level. *IEEE/ACM SIGCOMM, IMW*, 2001.

[33] K. Sriram and W. Whitt. Characterizing Superposition Arrival Processes in Packet Multiplexors for Voice and Data. In *IEEE J. Select. Areas Commun.*, volume 4, pages 833–846, 1986.

[34] M. S. Taqqu and V. Teverovsky. On Estimating the Intensity of Long-Range Dependence in Finite and Infinite Variance Time Series. In R. J. Alder, R. E. Feldman and M.S. Taqqu, editor, *A Practical Guide to Heavy Tails: Statistical Techniques and Applications*, pages 177–217. Birkhauser, Boston, 1998.

[35] X. Tian, J. Wu, and C. Ji. A Unified Framework for Understanding Network Traffic Using Independent Wavelet Models. In *IEEE INFOCOM*, 2002.

[36] A. Veres and M. Boda. The Chaotic Nature of TCP Congestion Control. In *IEEE INFOCOM*, pages 1715–1723, 2000.

[37] A. Veres, Z. Kenesi, S. Molnar, and G. Vattay. On the Propagation of Long-range Dependency in the Internet. In *SIGCOMM*, 2000.

[38] W. Willinger and V. Paxson. Where Mathematics Meets the Internet. In *Notices of the AMS*, 1998.

[39] 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. *IEEE/ACM Transactions on Networking*, 5(1):71–86, 1997.

[40] Y. Zhang and N. Duffield and V. Paxson and S. Shenker. On The Constancy of Internet Path Properties. In *In Proceedings of ACM SIGCOMM Internet Measurement Workshop*, 2001.

[41] Y. Zhang and V. Paxson and S. Shenker. The Stationarity of Internet Path Properties: Routing, Loss, and Throughput. *ACIRI Technical Report*, 2000.

[42] Z. L. Zhang, V. Ribeiro, S. Moon, and C. Diot. Small-Time Scaling Behaviors of Internet Backbone Traffic: An Empirical Study. In *IEEE INFOCOM*, SF, 2003.