Local Buffering and Correlations in Internet Traffic

Richard G. Clegg, Yann Golanski

Abstract

This paper describes a simple method based upon Markov chains for generating a traffic stream with particular selected correlation properties and its use in testing ideas for reducing correlations in data streams. The Markov model generates data exhibiting the statistical phenomenon known as long-range dependence (LRD) which describes a time series where the data has siginificant correlations even at high lags. This mathematical model is then implemented in simulation and buffering techniques are tested to reduce the correlations found in the traffic. It is found that local buffering can be used successfully to break up correlations in traffic and to selectively reduce correlations with a specific lag.

1 Introduction and Background

In the last ten years much attention has been given to the correlation structure of internet traffic. Consider a time series of packets per unit time travelling along a particular link in a network. Intuitively it would seem likely that correlations in this time series could have great effects on the queuing performance. For example traffic which tended to stay high for long periods and low for long periods might well cause more problems for a network than traffic which was less correlated. Long-range dependence (LRD) is a statistical phenomenon which describes a time series where the correlations persist even in widely separated points in the series. In 1993 a paper was published [9] which showed that measurements of data on the internet showed the presence of LRD. In the subsequent years, a huge number of papers have been published on this subject. The interest is mainly fuelled by the realisation that the statistical nature of the traffic can have a marked effect on the queuing performance.

Local buffering strategies have long been used in networks to improve queuing performance by smoothing data (or by admissions control). This paper introduces a novel (and extremely simple) method for generating a time-series of packets which exhibit LRD. This is used as a source of traffic with a correlation structure exhibiting correlations over a wide range of lags. A new method of local buffering is used on the data to influence correlations at specific lags. It is important to realise that local buffering cannot reduce LRD itself (and this is not the goal of this paper) but it could be used to influence correlations in the data.

1.1 A Brief Introduction to LRD

A good introduction to the topic of LRD is given by [2]. An introduction in the context of internet data measurement is given by [3, chapter one]. Several definitions of LRD are in common use in the literature (not all of them equivalent). A commonly used definition is given below.

Definition 1. A weakly stationary time series exhibits LRD if its autocorrelation function (ACF) does not have a finite sum. That is,

$$\sum_{k=-\infty}^{\infty} \rho(k) = \infty,$$

where $\rho(k)$ is the ACF.

It is often assumed that the ACF has the specific asymptotic form,

$$\rho(k) \sim c_\rho k^{-\alpha}, \tag{1}$$

for some positive constant c_ρ and some real $\alpha \in (0, 1)$. The symbol $\sim$ is used to mean asymptotically equal to so that $f(x) \sim g(x)$ means $f(x)/g(x) \to 1$ as $x \to \infty$. The best known measure of LRD is the Hurst parameter H where usually $H \in (1/2, 1)$ with $H = 1/2$ representing data which is short-range or independent. The parameter α is related to H by $H = 1 - \alpha/2$.

The constant α is related to the Hurst parameter A large number of papers have shown the presence of LRD in time series from internet traffic measurements [9, 16, 10] and it has also been shown that the effect on queuing performance can be considerable. Traffic which exhibits LRD can experience more packet drops and delays than traffic which does not. It is claimed [6] that the Hurst parameter is "...a dominant characteristic for a number of packet traffic engineering problems...". Estimates of queuing performance are given by [13, 14] but [12] notes that the effect is not a simple one and in some cases a high Hurst parameter may improve network performance.

1.2 Buffering and Correlations

It has long been known that local buffering in the internet can improve the performance of a network — notably for admission control at the edge of a network. The classic example of this is the much studied "Leaky Bucket" [4] and its variants. Other buffer management techniques such as RED [8], or BLUE [7] rely on feedback mechanims in the TCP protocol to affect data sending rates. Such non-local techniques will not be discussed in this paper. The idea of this work is to experiment on smart buffering where the rate of sending varies to achieve a specific aim. It can be shown [11] that local buffering cannot reduce the Hurst parameter of traffic. However, local buffering certainly can reduce correlations. This work provides an experimental investigation of this using simulation results.

2 A Markov method for Simulating LRD

Figure 1 shows an infinite Markov chain which can be used to generate a time series exhibiting LRD. A chain with this topology but different transition probabilities is studied in [15] and [1] (who also studies the double sided version). The parameters f_i are the transition probabilities for reaching a given state i from state 0. Also π_i is defined as the equilibrium probability of state i. It is clear that $\sum_{i=0}^{\infty} f_i = 1$ and also that $\sum_{i=0}^{\infty} \pi_i = 1$. The chain is described in detail in [3, Chapter 2].

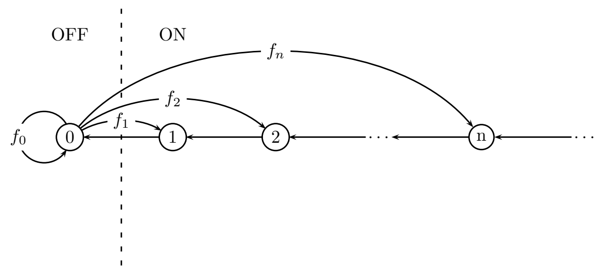

Figure 1: An infinite Markov chain which generates a time series exhibiting LRD.

The chain shown, given a starting state $X_0 \in \mathbb{Z}_+$, produces a Markov time series $X = \{X_i : i \in \mathbb{N}\}$ where all the $X_i \in \mathbb{Z}_+$. In turn, this chain can generate a time series $Y = \{Y_i : i \in \mathbb{N}\}$ where $Y_i = 0$ if $X_i = 0$ and $Y_i = 1$ otherwise.

It can be trivially shown that the chain is ergodic if $\sum_{i=0}^{\infty} i f_i < \infty$ and also $\forall i \in \mathbb{N}, \exists j > i : f_j > 0$. Further, it is simple to show that the equilibrium distribution π_i of the ith state is given by $\pi_i = \pi_0 \sum_{j=i}^{\infty} f_j$. Note that for $i = 0$ this simply says $\pi_0 = \pi_0$ since the transition probabilities must sum to one. In addition, since $\sum_{i=0}^{\infty} \pi_i = 1$ then $\pi_0 = 1 - \sum_{i=1}^{\infty} i f_i$.

An obvious way to introduce correlations of lag k into the series Y is to introduce unbroken sequences of k or more ones. Such an unbroken sequence can only occur if the X series is in a state k or higher (from the topology of the underlying chain). Therefore, the probability that $Y_{t+1} = \cdots = Y_{t+k} = 1$ is clearly given by $\sum_{i=k}^{\infty} \pi_i$. To introduce LRD the condition $\rho(k) \sim c_\rho k^{-\alpha}$ where $\alpha \in (0, 1)$ should be met if $\mathbb{P}[Y_{i+1} = 1 \ldots Y_{i+k} = 1] \sim Ck^{-\alpha}$. Hence the strict condition,

$$\sum_{i=k}^{\infty} \pi_i = Ck^{-\alpha},$$

is introduced where C is a constant (note that it has not yet been shown that this produces a valid ergodic Markov chain). By setting $k = 1$ it can be shown that $C = 1 - \pi_0$.

This is equivalent to setting

$$f_k = \frac{1 - \pi_0}{\pi_0} \left[k^{-\alpha} - 2(k+1)^{-\alpha} + (k+2)^{-\alpha} \right], \qquad (2)$$

for $k > 0$ and also, since $f_0 = 1 - \sum_{i=1}^{\infty} f_i$,

$$f_0 = 1 - \frac{1 - \pi_0}{\pi_0} \left[\sum_{i=1}^{\infty} i^{-\alpha} - 2 \sum_{i=2}^{\infty} i^{-\alpha} + \sum_{i=3}^{\infty} i^{-\alpha} \right],$$

which reduces to,

$$f_0 = 1 - \frac{1 - \pi_0}{\pi_0} \left[1 - 2^{-\alpha} \right]. \qquad (3)$$

These two equations (3) and (2) define a Markov chain which has two parameters α and π_0. The α parameter is the same parameter in (1) and is related to the Hurst parameter. The π_0 parameter is the equilibrium probability that the X series (and hence the Y series) is in the zero state and is hence given by $\pi_0 = 1 - \mu$ where μ is the mean of the Y series. It can be proved [3, chapter two] that this chain produces a time series with the given mean and Hurst parameter. The model is valid for $\alpha, \pi_0 \in (0, 1)$ if

$$\pi_0 > \frac{2^\alpha - 1}{2^{\alpha+1} - 1}.$$

2.1 Computational Implementation of Markov Method

While the model described in this section cannot be thought of as a realistic model of network traffic, it is rich enough to provide a correlation structure with significant correlations at all lags (the key feature of LRD). Simulating the model on a computer is extremely simple and it allows the production of a stream of packets which exhibits LRD. If the series X can be generated then the series Y can be generated from it with $Y_t = 1$ representing the sending of a packet and $Y_t = 0$ representing an interpacket gap. It is clear that if $X_t > 0$ then $X_{t+1} = X_t - 1$ from the structure of the chain. The only difficulty comes in generating X_{t+1} if $X_t = 0$. A naive approach to this would be to generate a random number r, uniformly distributed in $(0, 1)$ and say that X_{n+1} is the smallest i such that $\sum_{j=0}^{i} f_i < r$. This procedure is successful for small i but becomes inaccurate since as i increases the sum gets nearer to one but the f_i get nearer to zero. Adding numbers near zero to numbers near one is a difficult problem for finite precision computer arithmetic. Hence the errors in each stage of addition get larger. However, by the very nature of LRD, large values of i are likely to come up and it is these which are important.

It can simply be shown that if $X_t = 0$ and $0 < k \le i \le j$,

$$\mathbb{P}[X_{t+1} \in [i, j] | (X_{t+1} \in [k, \infty], X_t = 0)] = \frac{i^{-\alpha} - (i+1)^{-\alpha} - (j+1)^{-\alpha} + (j+2)^{-\alpha}}{k^{-\alpha} - (k+1)^{-\alpha}}. \qquad (4)$$

Using this equation, Table 1 shows a procedure for generating the sequence $\{X_t : t \in \mathbb{N}\}$ given some randomly chosen X_0.

1. If $X_t > 0$ then $X_{t+1} = X_t - 1$. Exit here.
2. Explicitly calculate $\mathbb{P}[X_{t+1} > j]$ for values of $j \leq N$ where N is some small integer as described previously.
3. Generate a new random number R in the range $[0, 1]$.
4. Calculate $\mathbb{P}[X_{t+1} \in [N, 2N-1] | X_{t+1} \in [N, \infty]]$ from equation (4). If R is less than or equal to this probability then X_{t+1} is in the required range. Otherwise set $N := 2N$ and go to step three.
5. If X_{t+1} is in the required range then refine down by generating a new random number and seeing if X_{t+1} is in the range $[N, (3/2)N]$. Continue refining by a binary search (with a new random number each time) until X_{t+1} is found. Exit here.

Table 1: A procedure for finding X_{t+1} from X_t in the infinite chain.

3 Local Buffering to Reduce Correlations

A leaky bucket only allows traffic to exit from the buffer at a limited rate. The intelligent buffer described here will use a decision process to select whether to allow traffic to exit at a high, medium or low rate. This rate will be chosen to minimise a given statistic. For the purpose of this paper, the autocorrelation function at various lags will be used as the statistic to minimise.

Let $X_T = \{x_1, x_2, \ldots, x_T\}$ be T samples from a time series representing the traffic per unit time output from a buffer. Let $\hat{k}(X_T)$ be a statistic which can be estimated on a sample of T items of data and which it is desirable to reduce. (The k stands for "kipple" a word coined by the science fiction writer Philip K. Dick [5] to describe detritus which builds up of its own accord if you do not take actions to reduce it.) Now, assume that after l steps there are three possible continuations of the series representing high, medium and low traffic. There are therefore three separate T length samples,

$$\begin{aligned} X_T^H &= \{x_{l+1}, \ldots, x_T, x_{T+1}^H, \ldots, x_{T+l}^H\} \\ X_T^M &= \{x_{l+1}, \ldots, x_T, x_{T+1}^M, \ldots, x_{T+l}^M\} \\ X_T^L &= \{x_{l+1}, \ldots, x_T, x_{T+1}^L, \ldots, x_{T+l}^L\}, \end{aligned}$$

where the H, M and L stand for high, medium and low. Therefore, there are three separate estimates for the value of the $\hat{k}$ statistic after l new values have been added to the time series.

1. First allow the buffer to output traffic at its normal rate for T time periods to get the series X_T.
2. Estimate the three samples X_T^H, X_T^M and X_T^L assuming that the buffer rate is set to high, medium or low respectively and with the assumption that the input to the buffer for the next k time steps is the same as it was for the most recent k time steps.
3. Calculate the values of $\hat{k}(X_T^H)$, $\hat{k}(X_T^M)$ and $\hat{k}(X_T^L)$.
4. Set the buffer output rate to high, medium or low according to which of the three statistics is lowest.
5. Wait one time step. Move the time series along by one time step so all the T length samples are moved one step to the right. Go to step two.

Table 2: A procedure for minimising a given statistic using a buffer.

Now, if the prediction in step two is good then this procedure should minimise the parameter $\hat{k}$. This will be tested in the simulation in the next section. The parameter l can be thought of the amount of time the simulation looks ahead. In the experiments described, this is kept as one.

4 Simulation Method

The network simulator ns-2 (`www.isi.edu/nsnam/ns/`) is open source and freely available online. It simulates packet based networks using either UDP or TCP protocols. The Markov based LRD generation mechanism was added to the ns-2 simulation as well as the intelligent buffering technique. In this case the simulation is kept extremely simple and uses UDP. The topology for the experiments described here is shown in figure 2. Nodes one to eight are sources generating long-range dependent traffic according to the Markov method previously described. They all send their traffic through to the output (labelled router node) via the shaper node which implements the procedure above to reduce correlations. In the results reported here the router node remains unused.

All links between the sources and the shaper have a capacity of $256kb/s$, the link between the shaper and the router has a capcity of $2048kb/s$ and the router to the exit is half of that. The sources are all sending UDP packets of size $256b$ at most every millisecond to give a maximum

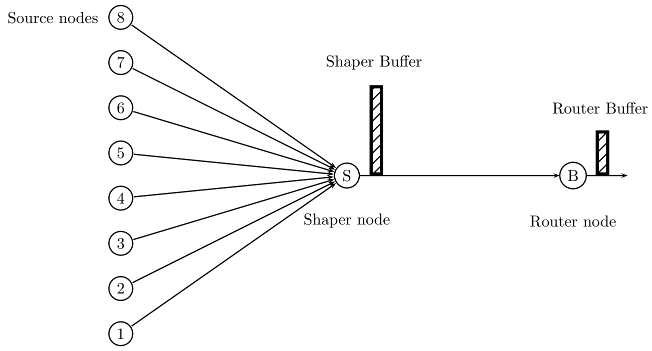

Figure 2: Network Topology for Testing in ns-2

rate of $256kb/s$ (that is, if the Markov chain generated a continual stream of ones then there would be 1000 packets of 256 bits giving a stream of $256kb/s$ kilobits per second). The rates were chosen such that if all the links send at exactly half their capacity then the router which carries traffic to the sink will be exactly full. Note that, in fact, if the traffic was this heavy (a mean of 0.5 for the Markov chain) then considerable packet loss would result since the system could only cope if the traffic were completely evenly distributed.

The system was tested with l (the "look ahead") parameter was set to one for these tests. The low, medium and high send rates for the shaper were $575kb/s$ (low), $1075kb/s$ (medium) and $1575kb/s$ (high). For the experiments described, the Hurst parameter was 0.75 and the mean is 0.4. The experiments are compared with a "leaky bucket" queue which simply sends out traffic at a maximum rate of $1575kb/s$. The time interval chosen to collect the data for the intelligent buffer was 0.32 seconds. These figures were chosen to produce a system which had some congestion, an interesting correlation structure in the leaky bucket case yet remained relatively simple.

5 Results

Initial results are with the $\hat{k}$ statistic measuring just the value of the autocorrelation function at a single lag. Figure 3 shows attempts to suppress each individual autocorrelation lag from one to twelve and compared to the base case (leaky bucket). The figures should be interpreted as follows, the curve LB AC shows the autocorrelation function of the output of the shaper node with the leaky bucket in place. This curve is present on each of the six graphs and shows the characteristic slow decline of the ACF which is characteristic of LRD. The line AC 1 shows the autocorrelation function with the intelligent buffer and the $\hat{k}$ statistic being an estimate of the autocorrelation function at lag one. By supressing $\hat{k}$ the autocorrelation function at lag one is suppressed and this can be clearly seen on the graph. Similarly for the AC 2 line, the autocorrelation function is suppressed at lag two.

These results show that this method can be used to suppress correlations at any lag. What is, perhaps, equally interesting is that there are such striking harmonic effects. For example, suppressing a correlation at a lag of one also suppresses all odd lags but raises the correlation at even lags (the AC 1 line). In fact, the rule seems to be that if we suppress at a lag k then lags with odd multiples of k are also suppressed but lags with even multiples of k actually gain in correlation. This can be clearly seen for the graph of AC 5 which shows a dip at lags five and fifteen but a peak at ten. This periodicity in the ACF should be no suprise and is even more striking in the next experiments.

This was followed by attempts to suppress correlations at multiple lags which was achieved simply by making the $\hat{k}$ parameter the sum of the ACF estimated over several lags. These experiments are shown in Figure 4 — graph (a) shows attempts to suppress all even lags, (b) shows attempts to suppress all odd lags, (c) shows attemps to suppress lags one to five and (d) shows attempts to suppress lags six to ten. The results show interesting behaviour of the autocorrelation. Graph (a) shows suppression of correlation at most lags except for multiples of twelve. Graph (b) shows that odd lags can be suppressed extremely successfully and this leads to increased correlations at even lags. Graph (c) shows that the lags one to five are suppressed very well indeed and this causes increased correlations at a lag of six and multiples of six. Finally, graph D shows that the procedure can successfully suppress the lags from six to ten at the expense of increased correlations at other lags.

6 Conclusions

An intelligent buffering method has been described which can be used to alter the nature of correlations in data. This could be part of a more general shaping scheme for data and allow finer control over the statistical nature of traffic on networks. If all network buffers performed such shaping and tweaking of traffic it seems likely that the network administrator could have a firm control over the nature of the traffic. However, much more work is needed in this area. There are a large number of parameters to be explored. In particular, it should be noted that the $\hat{k}$ statistic could be any statistic that we wished to reduce in the network which could in principle be reduced by buffering. The choice of the autocorrelation function was based upon the authors' interest in the subject of long-range dependence and it is clear that there are a large number of possibilities to explore.

References

[1] M. Barenco and D.K. Arrowsmith. The autocorrelation of double intermittency maps and the simulation of computer packet traffic. *Dynamical Systems*, 19(1):61–74, 2004.

[2] J. Beran. *Statistics For Long-Memory Processes.* Chapman and Hall, 1994.

[3] R. G. Clegg. *Statistics of Dynamic Networks*. PhD thesis, Dept. of Math., Uni. of York., York., 2004. Available online at: www.richardclegg.org/pubs/thesis.pdf.

[4] R. L. Cruz. A calculus for network delay, part i: Network elements in isolation. *IEEE Trans. Inform. Theory*, 37(1):114–131, January 1991.

[5] P. K. Dick. *Do Androids Dream of Electric Sheep?* Del Rey, 1968.

[6] A. Erramilli, O. Narayan, and W. Willinger. Experimental queueing analysis with long-range dependent packet traffic. *IEEE/ACM Trans. on Networking*, 4(2):209–223, 1996.

[7] W. Feng, K. Shin, D. Kandlur, and D. Saha. The BLUE active queue management algorithms, August 2002.

[8] S. Floyd and V. Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, 1993.

[9] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. On the self-similar nature of Ethernet traffic. In D. P. Sidhu, editor, *Proc. ACM SIGCOMM*, pages 183–193, San Francisco, California, 1993.

[10] R. Morris and D. Lin. Variance of aggregated web traffic. In *Proc. IEEE INFOCOM*, pages 360–366, 2000.

[11] A. L. Neidhardt, F. Huebner, and A. Erramilli. Shaping and policing of fractal traffic. *IEICE Trans. Commun.*, E81-B(5):858–869, 1998.

[12] A. L. Neidhardt and J. L. Wang. The concept of relevant time scales and its application to queuing analysis of self-similar traffic (or is Hurst naughty or nice?). In *Proceedings of the 1998 ACM SIGMETRICS joint international conference on Measurement and Modeling of Computer Systems*, pages 222–232, 1998.

[13] I. Norros. A storage model with self-similar input. *Queueing Systems*, 16:387–396, 1994.

[14] Z. Sahinoglu and S. Tekinay. On multimedia networks: Self similar traffic and network performance. *IEEE Communications Magazine*, January(1):48–52, 1999.

[15] X. J. Wang. Statistical physics of temporal intermittency. *Phys. Rev. A*, 40(11):6647–6661, 1989.

[16] 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 Trans. on Networking*, 5(1):71–86, 1997.

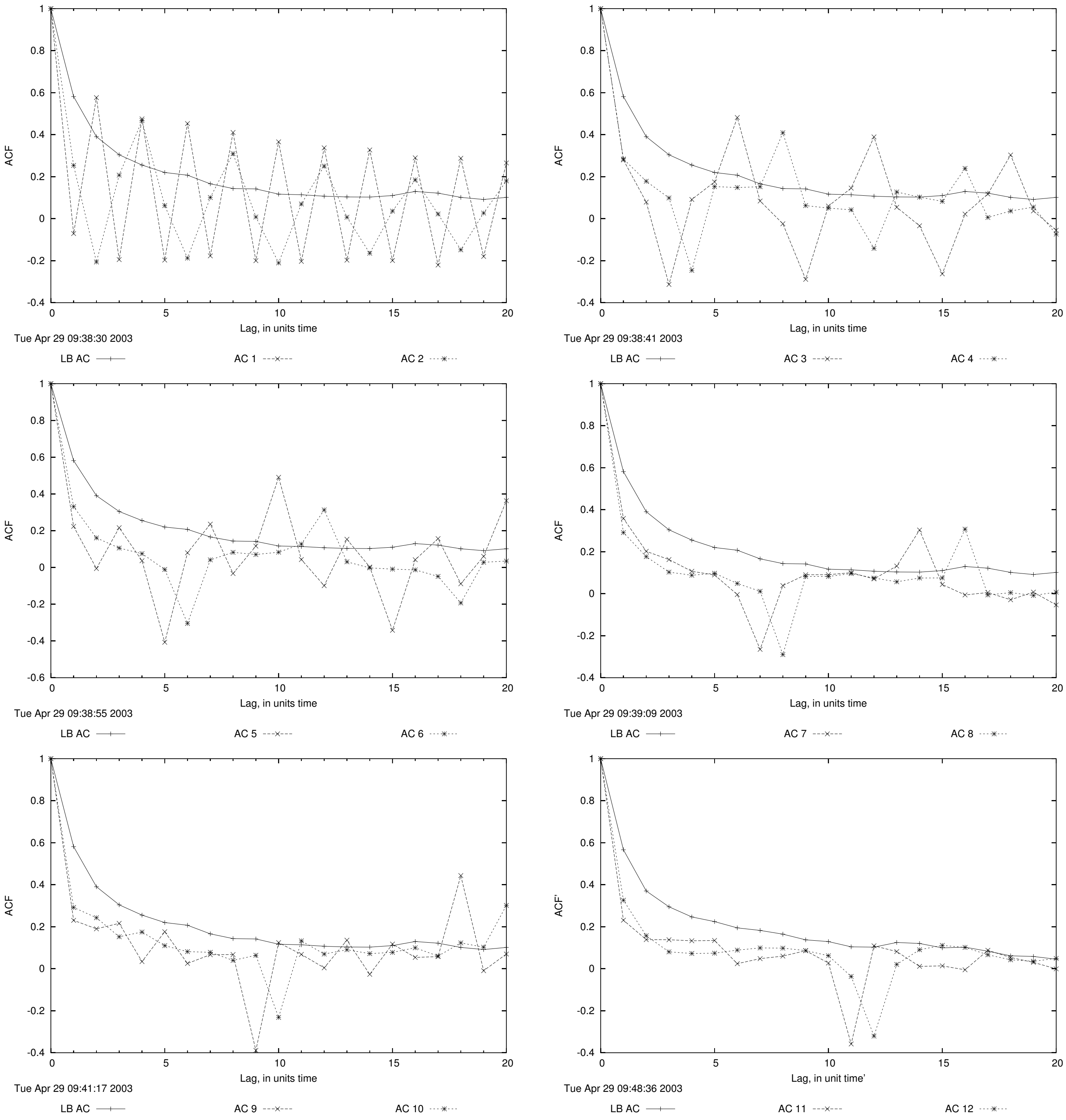

Figure 3: Autocorrelation function with different lags suppressed.

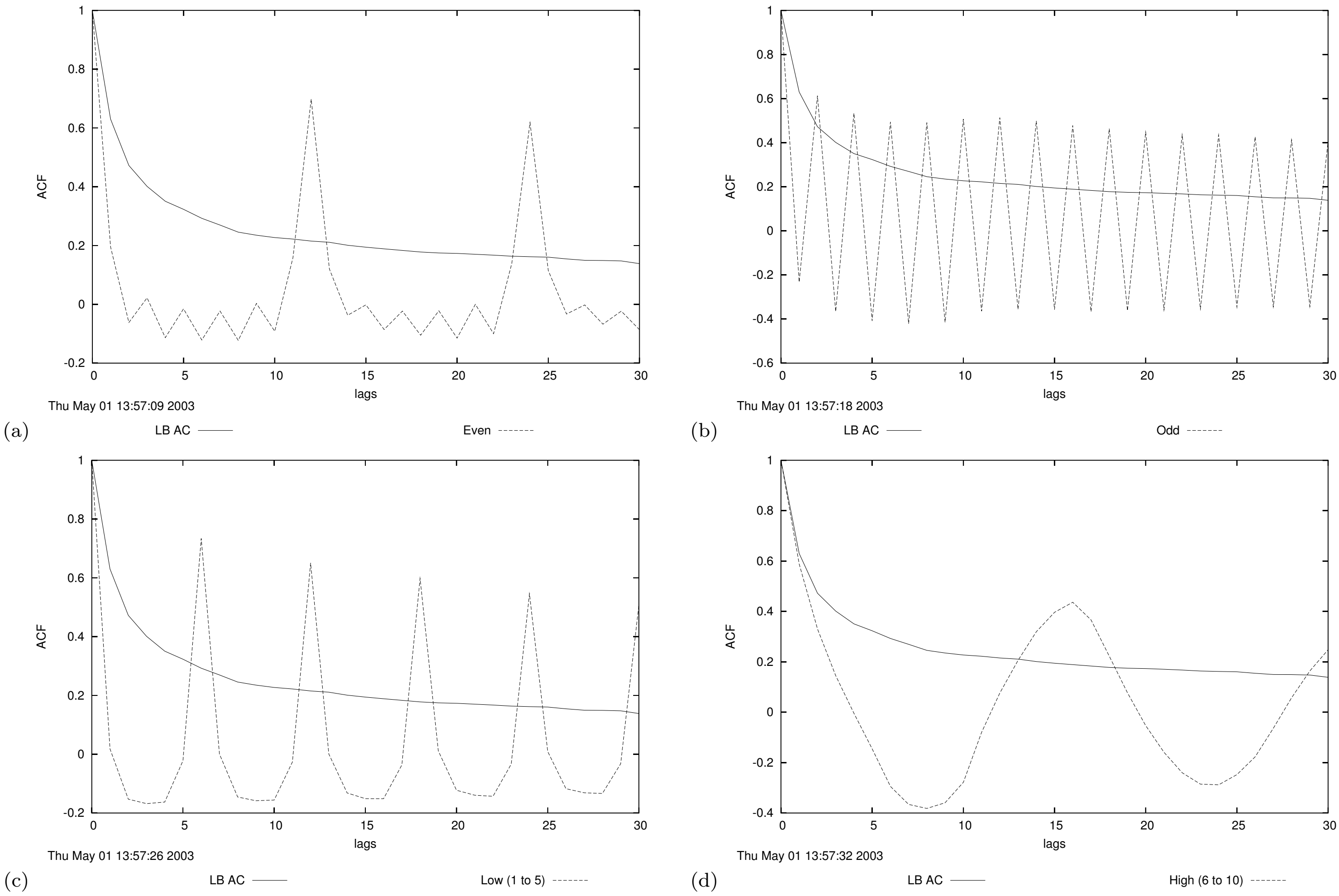

Figure 4: Suppressing only different combinations of lags; (a) is even, (b) is odd, (c) is 1 to 5 and (d) is 6 to 10.