On Aggregate Traffic Generation with Multifractal Properties

K. Kant

Intel Corporation

Beaverton, OR

Tel (503)677-6939, Fax (503)677-6700

kkant@co.intel.com

Abstract

This paper studies time scaling and queuing behavior of aggregate traffic that is asymptotically self-similar and also exhibits small time-scale irregularities introduced via a cascade construction. The work was motivated by the need for a web traffic generator that exercises HTTP servers in a lab environment and yet can capture traffic characteristics of a real-world web server with requests initiated over the WAN.

The main contribution of the paper is to provide some insight into how to choose an appropriate cascade construction in order to obtain desired scaling and queuing properties in a practical setting where the number of cascade levels is usually very limited. It is found that the largest time-scale of mass redistribution has far more influence on queuing behavior than the number of stages in the cascade. This, in turn, implies that a true multifractal construction (via a large number of stages) may be unnecessary, and thus appropriate queuing behavior might be achievable by a few stages and detailed modeling of the mass redistribution aspect of the network dynamics.

Symposium Area: High-Speed Networks

1 Introduction

The work reported here was motivated by the need to understand the implications of web traffic characteristics on the architectural aspects of a web server. It is well known that internet traffic, including web traffic, is extremely bursty and possesses characteristics (e.g., long range dependence and traffic irregularities caused by TCP window flow control) that make resource engineering very difficult. Apart from the usual issue of choosing buffer requirements, we are also interested in the utilization-delay characteristics for web-server internals that shed some light on the bandwidth requirements for the processor bus, I/O bus, memory channels, etc. In light of these needs, we needed a web traffic generator that exercises a web-server in a lab environment (to allow detailed hardware and O/S level measurements) and yet is powerful enough to emulate characteristics seen in real-life WAN traffic driving the web-server. In particular, the goal was to generate traffic that exhibits self-similarity which has been observed in all high-speed network traffic [13, 2, 3] and can be reasonably attributed to the on-off traffic pattern (high activity periods mixed with lull periods) typical of both the user-behavior and the web-browser behavior (request for a single web-page resulting in loading of multiple "files"). In addition, the traffic seen by a server is affected by the network dynamics (e.g., retransmissions, window-flow control, segmentation/reassembly, reencapsulation, etc.). Recently, a number of researchers have suggested that the impact of network dynamics in a WAN might be postulated as multifractal property [7, 14, 21]. Such a property also has a natural explanation in terms of the mass redistribution that occurs as a result of processing via various network layers and traffic passage through intermediate nodes. Our earlier study of web-traffic at HTTP level [11] confirmed asymptotic self-similarity and further analysis shows some multifractal characteristics as well. Consequently, the traffic generator was set up to generate asymptotically self-similar traffic with multifractal properties. This immediately brings the issue of how to control these aspects of the traffic. Generation of self-similar and asymptotically self-similar traffic is well understood by now (e.g., see [20, 23, 18, 12] among others); however, the use of multifractals for emulating the effect of network dynamics is new. The work reported here was an attempt to understand issues surrounding the generation of traffic with multifractal properties and its queuing properties.

The outline of the paper is as follows. Sections 2 and 3 briefly review self-similarity and multifractal behavior. Section 4 discusses our approach to traffic generation. Section 5 then studies the scaling and queuing behavior of the generated traffic and discusses how to choose the cascade generator parameters. Finally, section 7 concludes the work.

2 Self-similarity

Informally, self-similarity refers to the fact that the process aggregated over increasing time scales looks "similar", i.e., aggregation does not result in significant smoothing. Apart from the rather trivial case where the process has infinite variance and thus does not lend itself to smoothing, self-similarity is primarily caused by long-range dependence (LRD), i.e., significant correlations that persist over very large lags.

Let $\{X_i, i = 1, 2, \ldots\}$ denote a covariance stationary process with mean $\mu = E[X_i]$ and autocorrelation function $r(k) = E[(X_i - \mu)(X_{i+k} - \mu)]/\mathrm{Var}(X_i)$. Let $X_i^{(m)}$ denote the aggregated process with

block size m, defined as the sample means of blocks of size $m > 1$. More precisely:

$$X_j^{(m)} = \frac{1}{m} \sum_{i=m(j-1)+1}^{mj} X_i \tag{1}$$

Let $r^{(m)}(k)$ denote the autocorrelation function of the aggregated process $X_i^{(m)}$. The process X_i is called *exactly self-similar* if $r^{(m)}(k) = r(k)$, for all m and k, i.e., aggregation has no effect at all on the correlation structure. It is called *asymptotically self-similar* if $\lim_{m\to\infty} r^{(m)}(k) = r(k)$, i.e., self-similarity holds for large m (i.e., at large time scales). Fractional Brownian Motion (FBM) (traditional Brownian Motion except for the long-range dependence in successive increments) is an example of exactly self-similar process.

For an exactly self-similar process, the variance of the aggregated process $X_j^{(m)}$ goes down with m as $m^{-\beta}$ with $0 < \beta < 1$ as the decay rate. Thus, the *variance-time plot*, which plots $\log(\mathrm{Var}[X_j^{(m)}])$ against $\log(m)$ would imply asymptotic self-similarly if for large m the plot is linear [13]. Self-similarity is often characterized using the *Hurst parameter* H, defined as $1 - \beta/2$. H lies in the open interval (0.5,1.0), with H=0.5 for ordinary (i.e., short range dependent) random processes and higher values meaning more bursty traffic. Our analysis of Web request traffic from various sources showed H values ranging from 0.65 to 0.80. Note that much of the work on self-similarity reported in the literature concerns packet or byte traffic on a link. In contrast, our measurements concern application level traffic, i.e., HTTP requests and responses. Our measurements are consistent with the observation that while session arrival process is well modeled as Poisson, the number of requests within a session has a heavy tailed distribution which gives rise to self-similarity [7].

Self-similarity can be attributed to user behavior, which typically shows a very bursty on-off type of behavior (i.e., high activity periods punctuated by lull periods, both of which are heavy-tailed). Since user activities are expected to have dependencies over time scales of several minutes or even hours, a superposition of the traffic generated by a large number of users would exhibit *long range dependence* or *self-similarity* [23].

It is by now well established that if time-scales of interest are not limited, self-similarity can have a dramatic impact on performance, so much so, that one could even ignore short-term correlations. Reference [5] presents an interesting set of experiments to exhibit this. Narayan [15] has recently presented an exact analysis of a deterministic queue fed by FBM arrival process. It is shown that the complementary queue length distribution is Weibull multiplied by a very slowly decaying function $x^{-\gamma}$ where $\gamma = (1-H)(2H-1)/H$. That is, the queue-length distribution is very heavy-tailed, thus confirming the results in [5]. An earlier paper by Norros [16] provided a lower bound on queue length N, which confirms explosive queuing impact of self-similarity when large time-scales are admissible. The asymptotic queue length distribution for $M/G/\infty$ traffic is analyzed in [4] and is found to be of Pareto variety (which is much heavier tailed than Weibull). In particular, the complementary distribution decays as Cn^{2H-2} where C goes as $\rho^{2-2H}/(2-2H)(3-2H)$. However, studies (e.g., see [8, 10]) also show that if large time-scales are not captured (e.g., because of small buffers or timeouts), the impact of self-similarity may be much less significant than those of short-term correlations and marginal distribution.

3 Multifractal Properties

3.1 Need for Multifractal Models

Other than the user behavior, the network has the most influence in shaping the traffic seen by a server. This shaping may be deliberate (e.g., explicit enforcement of traffic parameters by means of a leaky-bucket or some other algorithm) or a result of processing or interaction with other types of traffic in the network. In particular, the traffic may encounter variable delays as it passes through intermediate nodes, packet losses and consequent retransmissions and window controls (e.g., TCP flow/congestion control), and segmentation/reassembly of packets at certain interfaces. The time constants associated with these activities range from short to medium term (sub-millisecond to a few hundred milliseconds), which are much less than the time constants involved in user activities. Consequently, asymptotic (i.e., large time scale) self-similarity aspects of the traffic are not affected by the network dynamics. On the other hand, the network effects introduce traffic irregularities at medium and short time scales that are important from a queuing perspective. The short-term irregularities can be captured by using appropriate marginal distributions and Markovian traffic models to represent short-term correlations. This still leaves medium-term irregularities that have a substantial impact on queuing. In fact, experiments involving queuing impact of WAN traffic show that the observed queuing delays are higher than would be observed if only the short-term correlations and marginal distribution are modeled. These effects are observed even when the buffers are not large enough to capture queuing effects of self-similar. Therefore, it is essential to represent medium term traffic irregularities in some controllable way.

An explicit modeling of network dynamics is clearly difficult; moreover, even if it could be done, the problem of deciding what "dials" to turn in order to achieve certain characteristics in the aggregate traffic is intractable. This has led to a direct analysis of WAN traffic for medium term scaling properties, and the analysis shows that the scaling phenomenon can be explained by using an extension of the self-similarity property, i.e., scaling phenomenon with different scaling coefficients in different time-scale region. A traffic showing such properties is known as *multi-fractal* and is defined more precisely in the next section.

3.2 Multifractal Properties

Multi-fractal traffic is defined as an extension of self-similar traffic by considering properties higher than second order characteristics. Recall that in characterizing exactly self-similar processes such as FBM we only need to consider a (wide-sense) covariance stationary process along with its autocorrelation function. If we were to examine third order properties, we also need to consider third-order moments of the marginal distribution and third order correlations (e.g., joint correlation between the samples X_i, X_{i+k_1} and X_{i+k_2} for all lag pairs (k_1, k_2)). Then, instead of just looking at the variance of the aggregated process $X_i^{(m)}$ as a function of m (as in the variance-time plots), we would also need to look at the skewness (or third central moment) of $X_i^{(m)}$ as a function of m. For an exactly self-similar process, a corresponding "skewness-time plot" will also yield a line with a scaled slope of $3\beta/2$. However, the more general multi-fractal process will still yield a linear plot, but with some arbitrary slope. More specifically, consider the qth order central moment at time-scale m:

$$\mu^{(m)}(q) = E\left[[X^{(m)} - E[X]]^q\right] \tag{2}$$

Then, a multi-fractal process is defined by the relationship:

$$\log \mu^{(m)}(q) = -\beta(q) \log m + C(q) \tag{3}$$

That is, the log of the moment decreases linearly with $\log m$ with certain slope β that depends on the order of the moment. However, for the special case of *mono-fractal* or self-similar process, $\beta(q)$ is linear in q, i.e.,

$$\beta(q) = q(1 - H) \tag{4}$$

where H is the Hurst parameter. That is, all moments show the same scaling behavior, and thus a log-log plot of any order moment against m will yield the same value of the Hurst parameter H.

It may not be clear from this discussion why the multi-fractal property deals with smaller time scales than the mono-fractal property. The connection is rather indirect and reflects the fact that we are dealing with an aggregate traffic property rather than saying something directly about mechanisms that operate at smaller time scales. The connection is as follows: recall that for any probability distribution, as we consider higher and higher order moments, we capture more and more irregularities in the distribution. In the present context, we batch size m corresponds to time-scale; thus, a higher order moment of $X^{(m)}$ captures events that happen at smaller time-scales. In an exactly self-similar process, there are no "irregularities" at smaller time-scales (i.e., correlation structure is no different at small/medium time scales than at large time scales), and therefore, every moment gives the same estimate of the H parameter. In a more general process, this is not the case. For example, a higher H value for third-order moment would indicate that the correlations decay more slowly over the medium term than over the long term.

The use of multi-fractals in network traffic modeling is very new and only a few results are currently available. In particular, it is shown in [7, 22] that while self-similarity is adequate for LAN traffic, WAN traffic traces are consistent with multifractal properties because in a WAN at small/medium time scales, the effect of network dynamics may dominate the user behavior. It is also shown that the scaling behavior shown by WAN traffic can be mimicked via a *cascade construction*, i.e., a multiplicative process that assigns mass to successively smaller time intervals according to some distribution (that depends on the level of subdivision). In the limit (i.e., as number of levels of subdivisions goes to infinity), such a procedure has been shown to generate a true multifractal [9].

4 Arrival Process Generation

The goal of arrival process generation is to generate times of successive request arrivals to a Web server such that the arrival process is asymptotically self-similar and shows appropriate multifractal behavior at smaller time scales.

There are a number of methods available in the literature to generate self-similar arrivals as surveyed in [20]. Of these, the one based on M/G/∞ queue [18] appears the most attractive for two reasons: (a) it is simple, efficient, and admits easy control of parameters, and (b) since the generated traffic is only asymptotically self-similar, it is possible to independently introduce multifractal behavior at small/medium time scales. Furthermore, given the result that a M/G/∞ process driving a deterministic queue yields heavier queue length tails than FBM for the same H parameter value, provides a more conservative process from queuing perspective.

An M/G/∞ process is defined as follows: Consider a discrete-time M/G/∞ queue with some time-

slot Δ as the time unit. All Poisson arrivals during a time-slot are put into service at the beginning of the next time-slot. Let $P(S = k), k = 1, 2, \ldots$ denote the mass function of the service time S in units of time-slots. Let $\hat{S}$ denote the residual service time of a customer. It is well-known that the queue-length distribution in this system at the end of each slot would be Poisson with mean $\lambda = \lambda_0 E[S]$ where λ_0 is average number of arrivals per slot to the M/G/∞ queue. However, the queue-lengths at the end of successive slots are correlated with autocorrelation function $\rho(k) = P(\hat{S} > k)$, the complementary distribution of the residual service time [18]. Thus, if we use this queue-length process to generate arrivals for the system of interest, we have the following arrival process A: the marginal distribution of A is discretised Poisson with rate λ per slot, and the $P(\hat{S} > k)$ as the autocorrelation function. In practice, we may want to achieve a given autocorrelation function $\rho(k)$, which can be used to compute the desired service time distribution. In particular,

$$P(S > k) = P(\hat{S} = k)E[S] = [\rho(k) - \rho(k + 1)]E[S] \tag{5}$$

Since $P(S > 0) = 1$ and $\rho(0) = 1$ by definition, $E[S] = 1/(1-\rho(1))$. Now, for long-range dependence, we let

$$\rho(k) = \alpha k^{-\beta}, \qquad 0 < \beta < 1 \tag{6}$$

where $\alpha = \rho(1) = 1 - 1/E[S]$. Then the generated arrival process is asymptotically self-similar with Hurst parameter $H = 1 - \beta/2$ [18].

Since the M/G/∞ system gives us only a discretised arrival process, the next step is to generate times of individual arrivals. This is done by collecting arrivals over $K \geq 1$ consecutive slots and then redistributing these over the entire interval of size $\tau_0 = \Delta K$ seconds. Let N denote the total number of arrivals over the span of K slots. Since N is Poisson, assigning each arrival to a point in the interval according to uniform distribution would yield exponentially distributed interarrival times within a slot. (The overall inter-arrival time distribution is still not exponential.) This is the base for comparison and creates a web page request process that is asymptotically self-similar with exponentially distributed increments.

The choice of continuous time domain in defining the arrival process is deliberate in view of the fact that we are interested in application level traffic generation and processing. For the same reason, the "time-slot" doesn't necessarily correspond to a transmission unit and can be chosen to be arbitrarily large. This necessarily implies dependence of results on slot-size, a topic discussed in the next section.

Following the approach in [7], multifractal properties can be induced in the arrival process by using a *cascade generator*. Let C_ℓ denote a symmetric RV in the range (0,1) with mean 1/2 at time-scale ℓ. (As discussed later in section 5, we identify the aggregation level as time-scale for convenience). Let $\tau_\ell = \tau_0 2^\ell$ denote the assignment interval size at time-scale ℓ. Let $L = \log_2(K)$ denote the largest (or the initial) time-scale for this subdivision. That is, the N arrivals are accumulated at time-scale L for redistribution. These N arrivals are split into portions $N.C_{L-1}$ and $N(1 - C_{L-1})$ and assigned to the left and right time intervals, each of size $\tau_1 = \tau_0 2^{L-1}$. This process is repeated recursively for certain number of time-scales, say L_C. At time-scale $\ell \in L - L_C..L - 1$, the number of arrivals during a subinterval, denoted N_ℓ, is given by the multiplication of appropriate C_i's, i.e.,

$$N_\ell = N\ C_{L-1}C_{L-2}\ldots C_\ell \tag{7}$$

As $L \to \infty$ (and finite L_C), this construction yields a true multifractal [9], which would have a log-

normal marginal distribution by virtue of multiplications of independent random variables. However, a number of issues need to be considered in practice. The first and most important issue is that the number of levels are necessarily very limited, and thus the asymptotic properties (e.g., lognormal marginals) may not hold at all. The second issue is how to control the irregularities resulting from the requirement of integral mass assignment when the mass to be assigned is small. The third issue is whether the cascade needs to be changed with the stage of subdivision, and if so, how that should be done. The construction in [7] yields monofractal behavior if all C_ℓ's are the same, and hence the authors suggest changing the variance as a function of ℓ. In particular, if C is the base generator, C_ℓ is constructed as follows:

$$C_\ell = \eta_\ell C + (1 - \eta_\ell)/2, \qquad 0 \leq \eta_\ell \leq 1 \tag{8}$$

Obviously, C_ℓ has a scaled variance of $\eta_\ell^2 \mathrm{Var}(C)$. Although η_ℓ's can be chosen to match scaling properties of a given trace, their choice in a traffic generator context is unclear. The next section addresses these and other issues.

5 Cascade Choice and Queuing Properties

The *semi-random cascade* generator used in [7] and discussed in the last section, is just one of many choices for mass redistribution. Perhaps a generator based on the real network dynamics would be most appropriate, but investigation of that aspect is beyond the scope of this paper. The binary subdivision used by this cascade can be extended to k-ary subdivision, however, in terms of matching scaling and queuing properties, no real advantage is gained by doing so. Thus, the basic generator of [7] is retained. To handle the rounding problem mentioned in the last section, the following approach is used: If n arrivals are to be assigned to an interval at some level, the assignment of round(nC) and $n-$round(nC) arrivals is alternated between the left and right subintervals. Also, given an integer L, arrivals are accumulated over 2^L consecutive slots, and then distributed according to the binary cascade C for all L levels.

Other alternatives were also considered, but were rejected. For example, it was originally thought that an important parameter to control is the number of levels L_C (starting at the given time-scale $L \geq L_C$) for which the mass is distributed according to the cascade, and below which an exponential interarrival time is enforced (by a uniformly distributed location of points as discussed above). This would limit the mass redistribution to time scales $L - L_C + 1$ through L and thus allow a better control over scaling and queuing properties. However, it was found that *the choice of L_C has very little influence on scaling properties; it is the absolute value of L that really matters*. For this reason and the fact that we would like to get as close to a true multifractal as feasible, we chose to carry the subdivision down to a single slot (which amounts to choosing $L_C = L$). *In terms of queuing impact, or perhaps analytic modeling convenience, one could as well consider the other extreme, i.e., $L_C = 1$.*

Yet another alternative considered was to keep one variance for a few levels and then change over to another value as suggested in [7]. However, we found that the C used at the highest level is the only one that affects performance, and the effect of changing the variances is completely negligible. In view of this, none of the results shown here vary C with the cascade level.

In order to examine the impact of the distribution of cascade generator C on traffic characteristics, several sample distributions were considered. However, *it was found that the scaling and queuing properties depend primarily on the variance of the cascade generator C, rather than on its shape*. (In the limit as $L \to \infty$, the dependence can be shown to be only on the variance). In view of this, all results shown in

this paper only concern *Uniform* distribution, which has a coefficient of variation (CV) of 0.577. Smaller variances can be obtained by simply using equation (8).

To examine multifractal scaling properties, we consider a few low-order moments of the aggregated process, i.e., $\mu^{(m)}(q)$ for $q = 2, 3, 4$ in equation (2). It is convenient (but not necessary) to use non-overlapping batches in the aggregation and increase the batch size by the same factor at each level. In fact, we start with a batch size of 2 and double it at each step. In view of this, we shall henceforth replace batch size m with aggregation level k, where $k = \log(m)$. Since aggregation level directly corresponds to the time-scale, we shall use the two interchangeably. We generated arrivals over 10 million interarrival times and aggregated them in blocks of size 1, 2, 4, 8, ... slots until the sample size goes down to 100 slots. The queuing properties were examined by feeding the traffic to a deterministic single-server queue. Note that unlike the ATM-like environments where a deterministic server with the rate of one service per time-slot is a natural choice [4, 24], no such clear choices exist for the application level models that we are interested in. We still chose deterministic service in order not to pollute the queuing results with effects of service-time variability. Even so, the service time does not necessarily equal a single slot duration; in fact, because of preference for large slot-sizes (discussed later), the service time is merely some fraction of the slot-size.[1]

It is found that from these experiments that $\log(\mu^{(k)}(q))$ as a function of aggregation level k shows a sharp drop at precisely $k = L$, i.e., at the largest time-scale at which mass redistribution is introduced. That is, the slope $\beta(q)$ changes substantially between time scales L and $L + 1$. This change in scaling behavior can be shown more dramatically by computing a local estimate of the H parameter as a function of k using equation (4) and the local two-point estimate of $\beta(q)$. (This local estimate is undefined at $k = 1$, and was arbitrarily chosen to be the same as for $k = 2$.) The graphs in Fig. 1 show this estimate for $L \in 0..8$.

Fig. 1.1 essentially shows the M/G/∞ traffic since there is no mass redistribution in this case (i.e., $L = 0$). In this case, the traffic shows stronger short-term correlations but eventually becomes consistent with a self-similar process with $H = 0.8$. In other cases, the traffic is again asymptotically self-similar with $H \approx 0.8$, but shows markedly different behavior around L, the highest time-scale at which mass redistribution is introduced. In particular, at time-scale $L + 1$, the local H estimate dips very sharply but recovers quickly on both sides (i.e., smaller and larger time-scales.) Furthermore, the extent of the dip generally increases monotonically with L. The dip also correlates very well with the queuing behavior — a large dip results in large queuing effects. An immediate consequence of these observations is that the absolute value of L is far more important in terms of scaling and queuing behavior than L_C, the number of levels over which the mass redistribution is done. In fact, $L_C > 1$ results in only marginally worse queuing behavior than $L_C = 1$. Another point to note is that higher moments show essentially the same behavior as variance around the dip, and thus it suffices to concentrate on the variance itself.

The significance of the dip in H estimate can be understood from the mechanics of mass redistribution. The mass redistribution at time-scales $L..L - L_C$ has two major impacts:

1. Correlations over time scales $L..L - L_C$ are compromised. Since the original traffic has strong correlations, the mass redistribution causes the variance (and higher central moments) to drop substantially.

[1] If a discrete-time model is preferred in certain applications, one could define the service time duration as a "mini-slot" and discretize everything else accordingly.

2. A considerable batching of arrivals occurs at smaller time-scales, i.e., the local marginal distribution becomes much more irregular. Consequently, the traffic leads to a much worse queuing behavior at short to medium time scales.

In light of this, mass redistribution can be thought of as a mechanism that locally weakens medium-term correlations at the cost of lumpier marginals at shorter time-scales. This impact is largest at time-scale L and goes down exponentially at lower time-scales. This explains the quick recovery of H estimate to the left of time-scale L.

Fig. 2 shows the queuing behavior of the traffic. In particular, Fig. 2.1 shows the average queue length at 70% utilization as a function of maximum cascade level L for standard deviation modifiers [η in equation(8)] as given. It is clear that as L increases, η has an increasingly larger impact on queuing behavior. In other words, one could match almost any queuing behavior that might arise in practice by fixing L at some sufficiently large value and "dialing" an appropriate η value. To this end, Fig. 2.2 re-displays the information in Fig. 2.1 with η along the x axis. These curves can be used to choose an appropriate L and η so that the desired average queue length (perhaps determined from live measurements) is achieved. Finally, Fig. 2.3 shows the average queue-length as a function of utilization, with L as a parameter. The tremendous influence of L on queuing behavior should be clear from this. In particular, at 90% utilization, $L = 0$ (i.e., no mass redistribution) yields a queue-length of only about 55, whereas for $L = 8$, the queue length approaches 1260!

All results discussed thus far are for slot-size of 25 (i.e., 25 average interarrival times). We now discuss the sensitivity with respect to slot-size that was alluded to in the above. Given some value for L, the time-period over which the mass-redistribution occurs increases with the slot-size. This immediately implies that for $L > 0$, queuing results will always depend on the slot-size. When the time-scale is small, only very short-term correlations are disturbed by the mass redistribution, thereby resulting in smaller drop in local H estimate at time-scale $L+1$. This is illustrated in Fig. 3 for slot-size of 5. On the contrary, a small slot-size results in considerable "edge effects" in the process of locating arrivals within a single time-slot. In fact, with $L = 0$ (i.e., no mass redistribution), one would expect the coefficient of variation of interarrival time (CV_{ia}) to be 1 if there were no edge effects. With slot-size=1, we instead find that $CV_{ia} = 2$. As the slot-size increases, CV_{ia} decreases rapidly to about 1.06 for slot-size=5, and becomes almost 1 at slot-size$\geq$ 10. Because of large CV_{ia}, the queue lengths are very large at slot-size=1 (see Fig. 2.6), but decrease rapidly as slot-size increases (as shown in Fig. 2.5 for slot-size 10). The chosen maximum value of 25 for slot-size, was motivated by the fact that the edge-effects become almost zero at this slot-size. Notice that at small slot-sizes, the queuing behavior is rather complex (i.e., smaller L doesn't necessarily mean smaller queue-length). Thus, large slot-sizes are preferred from a modeling perspective, as it is easy to find an appropriate L value for matching the observed queuing behavior. Note that when edge effects are not an issue, at large L, we can expect queuing results to be about the same so long as the total mass-redistribution interval remains the same. Thus, for example, with a slot-size of 50 and $L = 128$ (see Fig. 2.4), the average queue can be expected to be about the same as with slot-size of 25 and $L = 256$. (The correspondance is not exact because in the first case, mass redistribution proceeds down to a time-scale which is twice that in the second case.)

It was mentioned above that the at time-scale $L + 1$, the local H estimate dips sharply and the extent of the dip controls the queuing behavior. Fig. 4 illustrates this phenomenon. It shows the log-log plot of average queue length vs. variance at time-scale $L + 1$ for a couple of values of the standard deviation modifier η. (Time-scales 0-3 are excised from these figures since from Figs. 1 and 3, there is no clear dip at those time-scales). It is seen that the behavior is roughly linear; the dotted lines shows the linear regression fit and the computed slope. In other words, the average queue-length varies as some power of

the coefficient of variation at time-scale $L+1$. Furthermore, the slope appears to decrease with increasing cascade variance. It remains to verify these results analytically.

6 Wavelet Analysis

In the previous section, the effect of cascade construction on the scaling properties of the traffic was studied only by considering a simple 2-point estimate of the H-parameter across successive scales. While such a technique is fine for illustrative purposes, a meaningful statistical estimation of H necessarily requires it to remain invariant over a range of scales. Also, strong correlations between $X^{(m)}$'s for different m values makes a direct estimation of H unreliable [1]. In this section, we present the scaling analysis of the generated traffic using wavelets in order to further reveal the effect of the perturbation caused by the cascade construction.

Wavelets are well suited for studying the scaling (i.e., time vs. frequency) properties of traffic because of their ability to zoom in on the desired range of time and frequency. As in previous sections, the measure of interest here is the number of arrivals during a time period. The wavelet decomposition of such a "signal" characterises its behavior both over time and over successive scales (where the scale is synonmous with the aggregative level for the purposes of this paper). Let $\psi(t)$ denote the "mother wavelet", and $\psi_{m,k}(t)$ its scaled and dilated variants defined as follows:

$$\psi_{m,k}(t) = \sqrt{2^{-m}}\psi(2^{-m}t - k) \qquad m = 0, 1, 2, \ldots, K, qsk = 0, 1, 2, ldots \tag{9}$$

where the index m refers to behavior over time-scales of duration 2^m time-slots. Then if $X(t)$, $t = 1, 2, 3, \ldots$ denotes the arrival process, its wavelet decomposition is given by:

$$X(t) = \sum_{\forall m} \sum_{\forall k} d_X(m, k)\psi_{m,k}(t) \tag{10}$$

where $d_X(m, k)$'s are the coefficients of the wavelet expansion and are given by the following inner product:

$$d_X(m, k) = \sum_{\forall t} X(t)\psi_{m,k}(t) \tag{11}$$

Given the coefficients $d_X(m, k)$'s, one can study both the time localized behavior of the traffic (by choosing a specific time of interest k) and its global properties (by aggregating over the time parameter k). In this paper, we are concerned with only the global properties of traffic. Considering $d_X(m, k)$'s for a fixed m as distribution of a RV over the time k, we can examine its qth order moment, henceforth called *partition function*:

$$E(q, m) = \frac{1}{n_m} \sum_{k=0}^{n_m - 1} |d_X(m, k)]^q \tag{12}$$

In particular $q = 2$ gives the variance of $d_X(m, .)$'s, or the energy of the signal. It can be shown that if X has stationary increments, $d_X(m, k)$'s (for a given m) are stationary, and have only short-term correlations. This makes estimators such as $E(q, m)$ more reliable than the corresponding time-domain estimators such as $\mu^{(m)}(q)$ considered in the previous sections. It can be shown that for a self-similar process, $E(q, m)$ shows a power law behavior similar to the one for $\mu^{(m)}(q)$ in equation (3). That is, for

some slope $\alpha(q)$ and intercept $C'(q)$, we have:

$$\log E(q,m) = \alpha(q)\log m + C'(q) \tag{13}$$

where the function $\alpha(q)$ is given by:

$$\alpha(q) = q(H - 1/2) \tag{14}$$

Furthermore, for multifractal processes, it can be shown that equation (13) holds for small time-scales, but the function $\alpha(q)$ could be arbitrary. Thus plots of $\log E(q,m)$ and $\log m$, called as *Logscale diagrams* in [1], provide another mechanism for studying the scaling properties of the traffic generated using the technique discussed in this paper.

Figures 2–5 show some sample curves for $q = 2 - 5$. Based on the discussion in section 5, the curves here are restricted to the slot size of 25. The mother wavelet used in all cases is db3, i.e., Daubechies wavelet with 3 vanishing moments. Note that with n vanishing moments, the spectral density of the wavelet goes down as f^n as $f \to 0$. Also, n vanishing moments would decimate any nonstationarity in the data that can be described by an $(n-1)$th order polynominal. We conducted experiments using wavelets with $n = 1..10$, and found that $n = 3$ is quite adequate. Large values of n smooth out the scaling behavior but require larger sample sizes at large values of j because of increased support.

As a starting point, figure 1 shows the logscale diagrams for pure $M/G/\infty$ traffic (i.e., for the traffic generated as in section 4 but without any mass redistribution). The vertical lines at each scale show the 97.5% confidence interval of the estimated value at that scale. The dashed-dotted line shows the regression line that includes the correction factor for the fact that $E[\log(X)] \neq \log[E(X)]$, as suggested in [1]. Keeping in mind that $M/G/\infty$ traffic is only asymptotically self-similar, the regression line considers only the time scales 5-14. The estimated α and H values are shown. Note that the estimated H value is pretty close to the expected value of 0.8. Furthermore, all moments show an almost identical behavior, as expected for a self-similar process.

Figure 2 shows the logscale diagrams for $L = 5$, i.e., when mass distribution occurs only for scales 1-5. It is seen that the partition function gets severely perturbed around time scale 5. Between scales 1 and 5, the partition functions increases with a slope of around 1.3, which appears characteristic of the cascade chosen here. Between scales 5 and 9, the partition function goes down steadily, which indicates decreasing influence of the mass redistribution at larger time scales. Between time scales 9 to 14, the traffic is essentially unaffected by the mass redistribution at small timescales, and shows linear scaling (i.e., asymptotic self similarity) with H estimate of about 0.86.

Figure 3 shows the logscale diagrams for $L = 6$. As expected, the partition function increases for scales 1-6 with a slope of about 1.3, and then mostly decreases until scale 11. Beyond this time scale, it should show asymptotic self-similarity, except that the diagrams don't have enough scales with sizable samples to discern any specific behavior.

Figure 4 shows the extreme case where cascade construction starts at $L = 12$. In this case, the traffic characteristics over almost the entire range (i.e., up to scale 11) are primarily governed by mass redistribution. The partition function depends on the moments of the cascade generator, and thus the use of same cascade generator at all scales implies linear scaling behavior across the scales as confirmed by Figure 4. The influence of M/G/∞ mechanism comes into play in form of the variability of the total starting mass. A major consequence of this variability is that equation (14) does not hold. In fact, $\alpha(q)$ decreases with increasing q, which means that the traffic does not show a simple monofractal behavior.

Figure 1:

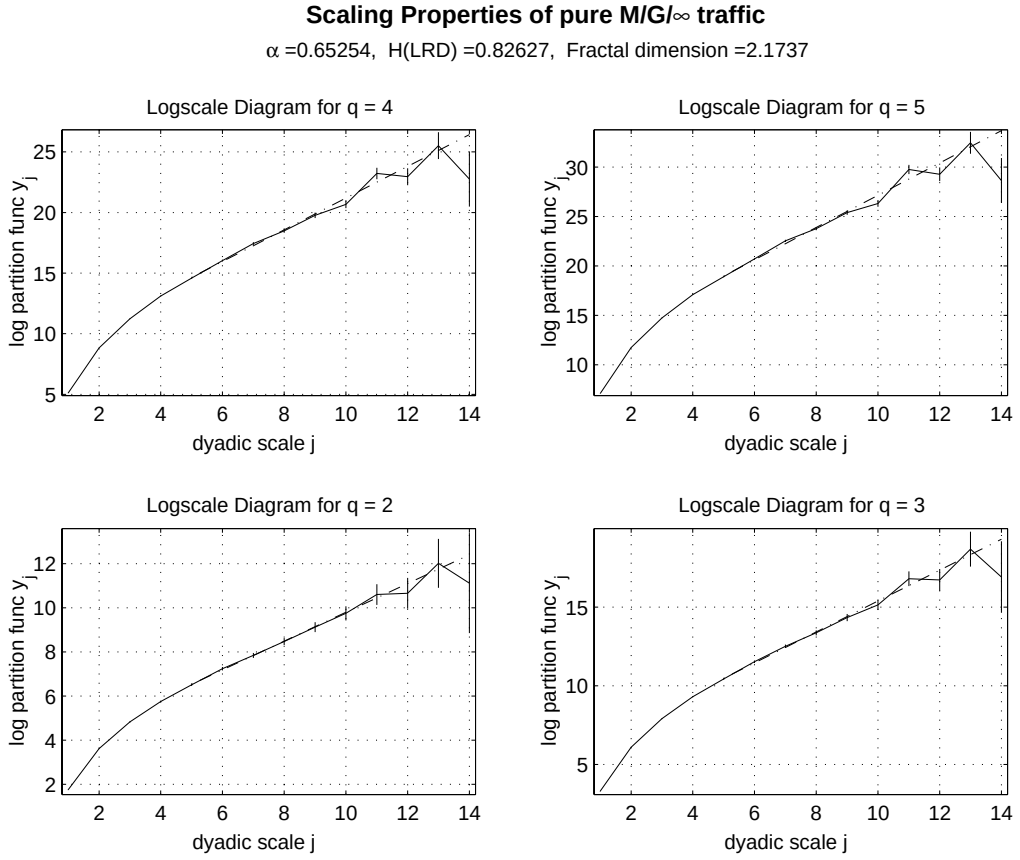
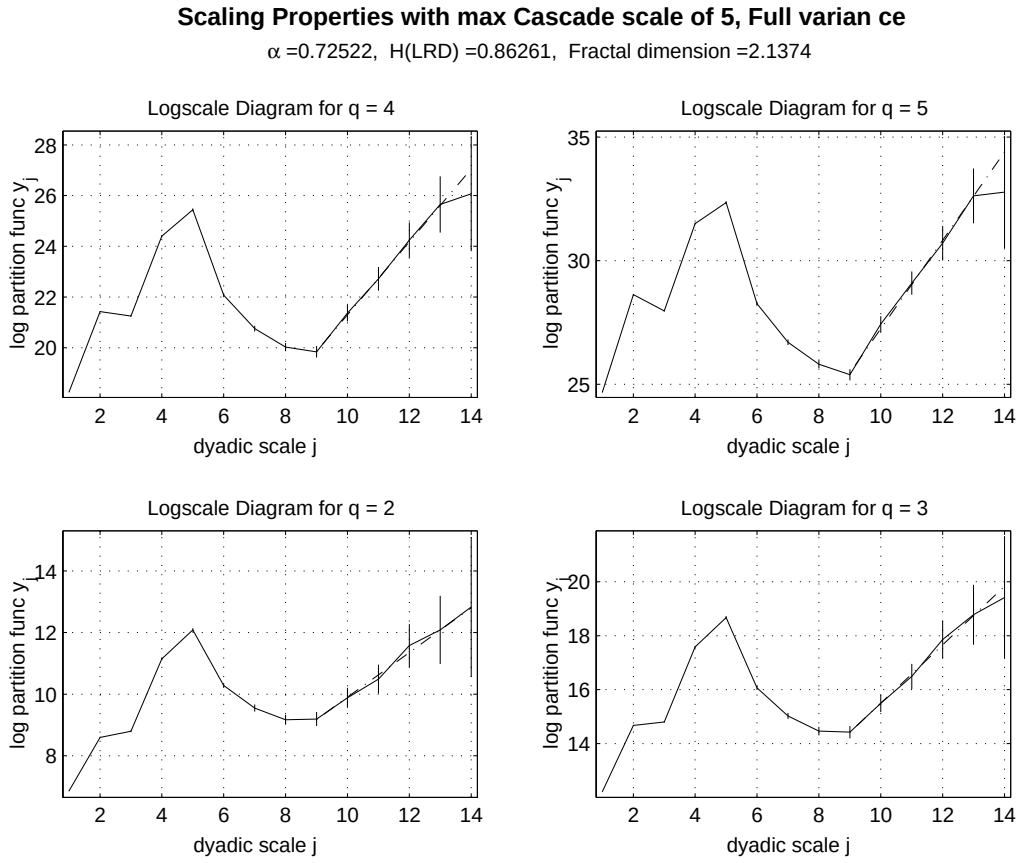


Figure 5 shows the impact of cascade generator variance on the scaling properties. Here again $L = 12$, but the variance is reduced to zero, thereby making it a deterministic cascade. If the starting mass had zero variability, the partition function will be zero (actually a very small value) for the entire range of scales. Instead, it is seen that the partition function has a small value up to about scale 9 and then rises sharply. In fact, at large time scales the behavior is almost the same as for the full variance case, since in these cases, the scaling properties are dominated by the variance of the original M/G/∞ process.

Note that in all cases, the effective H parameter value due to cascade construction exceeds 1. This is an effect of our traffic generation scheme (i.e., a M/G/∞ traffic collected over successive slot blocks, as opposed to starting with a single large mass which is divided down using the cascade). In the extreme case of a deterministic cascade, the H estimate is 0.5 for small time scales but is again governed by other aspects of the traffic generation technique for large time scales. For examples, with long-range dependent M/G/∞ traffic, at large time scales, H still exceeds 1; however, with short range dependent traffic, the H estimate becomes 0.5 for large time scales as well. An $H > 1$ over the mass redistribution scales coupled with a basic M/G/∞ process indicates that the cascade construction will lead to a substantially worse queuing performance than without mass redistribution as observed in the last section.

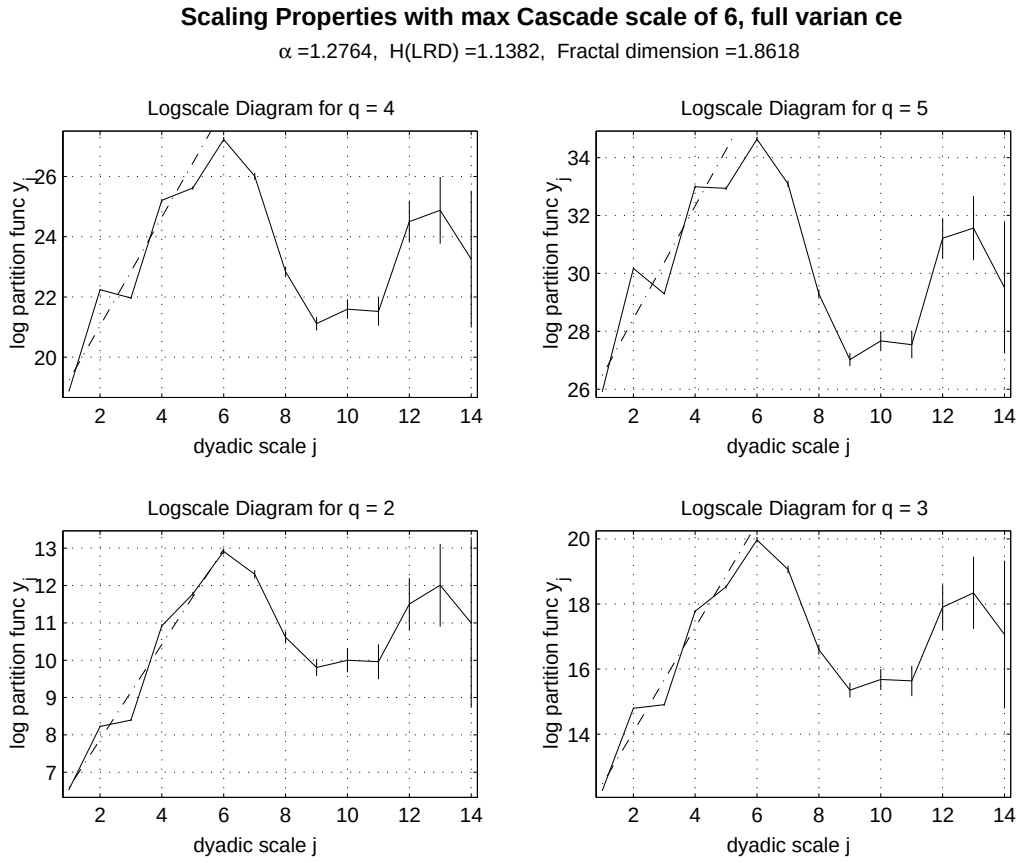
Figure 2:



7 Conclusions

This paper studies queuing behavior of aggregate traffic that is asymptotically self-similar and also exhibits small time-scale irregularities introduced via a cascade construction. The main contribution of the paper is to provide some insight into how to choose an appropriate cascade construction in order to obtain desired scaling and queuing properties in a practical setting where the number of cascade levels is usually very limited. It is found that the largest time-scale of mass redistribution has far more influence on queuing behavior than the number of stages in the cascade. Also, the queuing and scaling behavior is primarily governed by the second order properties of the cascade. Third, the local irregularity introduced by the cascade appears to be simply related to the queuing behavior, but it remains to pursue this result analytically. In view of the fact that just one or two mass redistributions can seriously affect the scaling and queuing aspects, it is natural to base the mass redistribution on the real networking mechanisms. In fact, such mechanisms may not even pass as real cascades, but the results in this paper suggest that a faithful modeling of their overall effect could result in traffic with realistic queuing behavior. We also showed that in general, the scaling and queuing properties depend on the slot-size, and discussed issues in choosing an appropriate slot-size.

Figure 3:



Acknowledgements

The author would like to acknowledge helpful discussions with Ashok Erramilli and Y. Chandramouli on multifractal traffic characteristics.

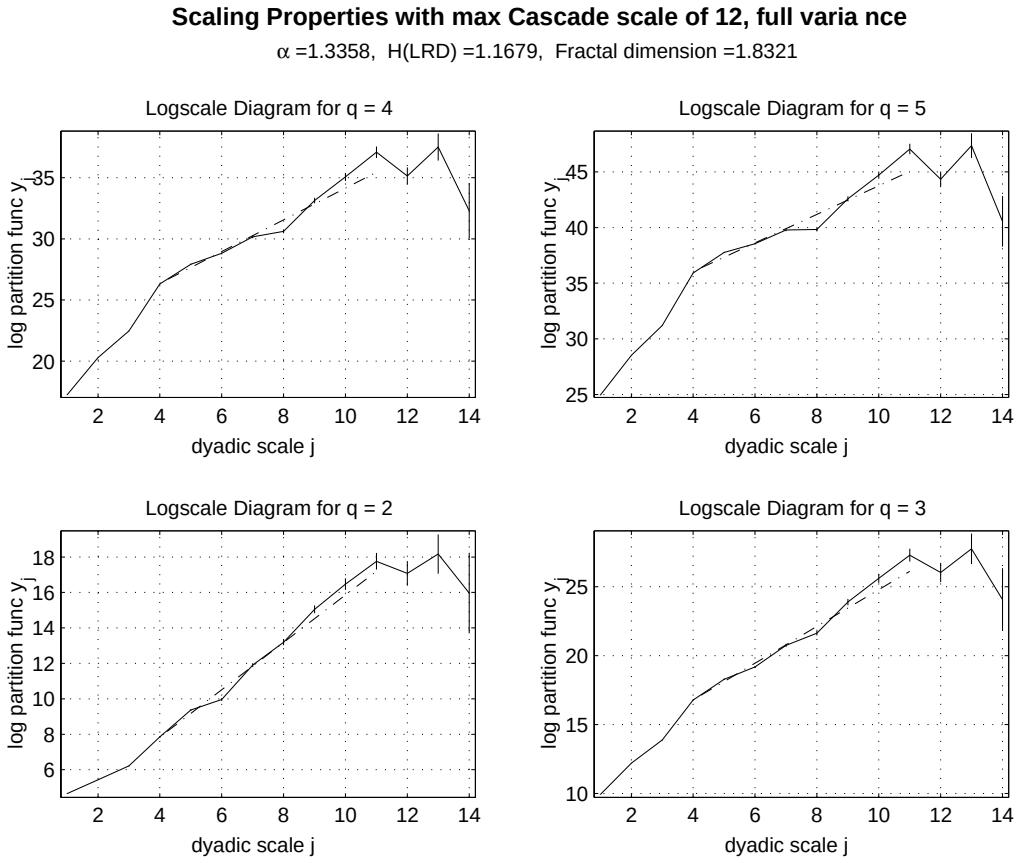
References

[1] P. Abry, P. Flandrin, M.S. Taqqu, and D. Veitch, “Wavelets for the analysis, estimation, and synthesis of scaling data”, report available from `www.serc.mit.edu.au/ darryl`.

[2] J. Beran, R. Sherman, and W. Willinger, “Long range dependence in variable bit rate (VBR) video traffic”, IEEE trans on communications, Vol 43, No 3, pp1566-1579, Feb 1995.

[3] M.E. Crovella and A. Bestavros, “Explaining World-wide web self-similarity”, Technical Report, Dept of Computer Science, Boston University, Oct 1995.

Figure 4:



[4] T. Daniels and C. Blondia, "Asymptotic Behavior of a Discrete Time Queue with Long Range Dependent M/G/∞ Input", Technical Report, Dept of Math and Computer Science, University of Antwerp, Belgium, April 1998.

[5] A. Erramilli, O. Narayan and W. Willinger, "Experimental queuing analysis with long range dependent packet traffic", IEEE/ACM trans on networking, April 1996.

[6] R. Felding, J. Gettys, et. al., "Hypertext Transfer Protocol — HTTP 1.1", IETF RFC 2068.

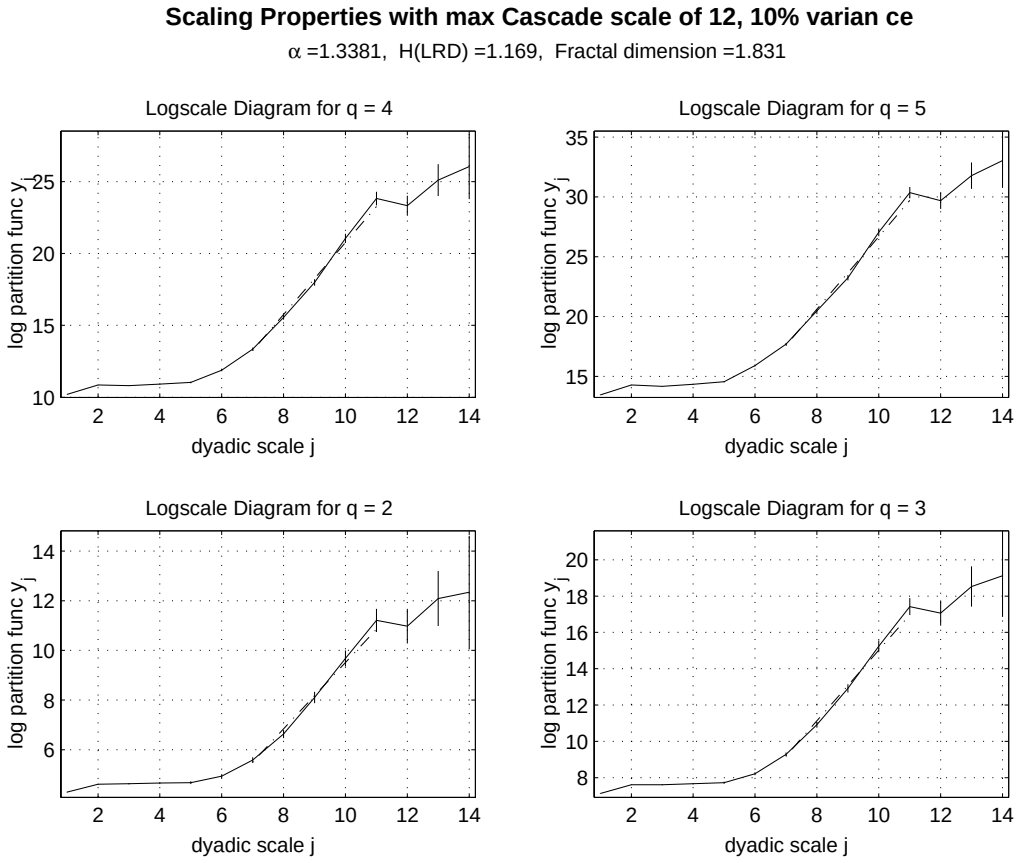
[7] A. Feldmann, A.C. Gilbert and W. Willinger, "Data Networks as Cascades: Investigating the multifractal nature of Internet WAN traffic", Proc. 1998 ACM SIGCOMM, pp42-55.

[8] M. Grosslauser and J.C. Bolot, "On the relevance of long-range dependence in network traffic", Proc ACM SIGCOMM 96, pp15-24.

[9] R. Holley and E.C. Waymire, "Multifractal Dimensions and Scaling Exponents for Strongly Bounded Random Cascades", Annals of Applied Probability, Vol 2, pp 819-845, 1992.

[10] D. Heyman and T.V. Lakshman, "What are the implications of long-range dependence for VBR video traffic engineering", IEEE/ACM trans on Networking, Vol 4, pp301-317, June 1996.

Figure 5:



[11] K. Kant and Y. Won, "Server Capacity Planning for Web Traffic Workload", to appear in IEEE trans on Data and Knowledge Eng.

[12] M. Krunz and A. Makowski, "A source model for VBR Video traffic based on M/G/∞ Input", Technical Report, Univ of Maryland.

[13] W.E. Leland, M.S. Taqqu, W. Willinger and D.V. Wilson, "On the self-similar nature of ethernet traffic", IEEE/ACM trans on networking, Vol 2, No 1, pp 1-15, Fen 1994.

[14] P. Mannersalo and I. Norros, "Multifractal analysis: A potential tool for teletraffic characterization?", COST257 report 97/32.

[15] O. Narayan, "Exact asymptotic queue length distribution for fractional brownian motion", Advances in Performance Analysis, Vol 1, pp39-63, 1998.

[16] I. Norros, "A buffer with self-similar input", Queuing systems, Vol 16, No 2, pp382-396, Feb 1994.

[17] K. Park, G. Kim, and M. Crovella, "On the effect of traffic self-similarity on network performance", Technical report CSD-TR 97-024, Dept of computer science, Purdue university.

[18] M. Parulekar and A. Makovski, "M/G/∞ input processes: A versatile class of models for network traffic", Proc of IEEE Infocom 97, April 1997.

[19] J.D. Pruneski and S.Q. Li, "The linearity of low frequency traffic flow: An intrinsic I/O property in queuing systems", IEEE Globecom, 1995.

[20] V. Paxson, "Fast, approximate synthesis of fractional gaussian noise for generating self-similar network traffic", Computer Communication Review, Vol 27, No 5, pp 5-18, Oct 1997.

[21] R.H. Reidi and J.L. Vehel, "TCP Traffic is Multifractal: a numerical study", To appear in IEEE Trans on Networking.

[22] M.S. Taqqu, V. Teverovsky and W. Willinger, "Is network traffic self-similar or multifractal?", To appear in *Fractals*.

[23] 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", Proc of SIGCOMM 95, pp100-113.

[24] S. Wittevrongel and H. Bruneel, "Correlation Effects in ATM Queues due to Data Format Conversions", Performance Evaluation, Vol 32, 1998, pp35-56.

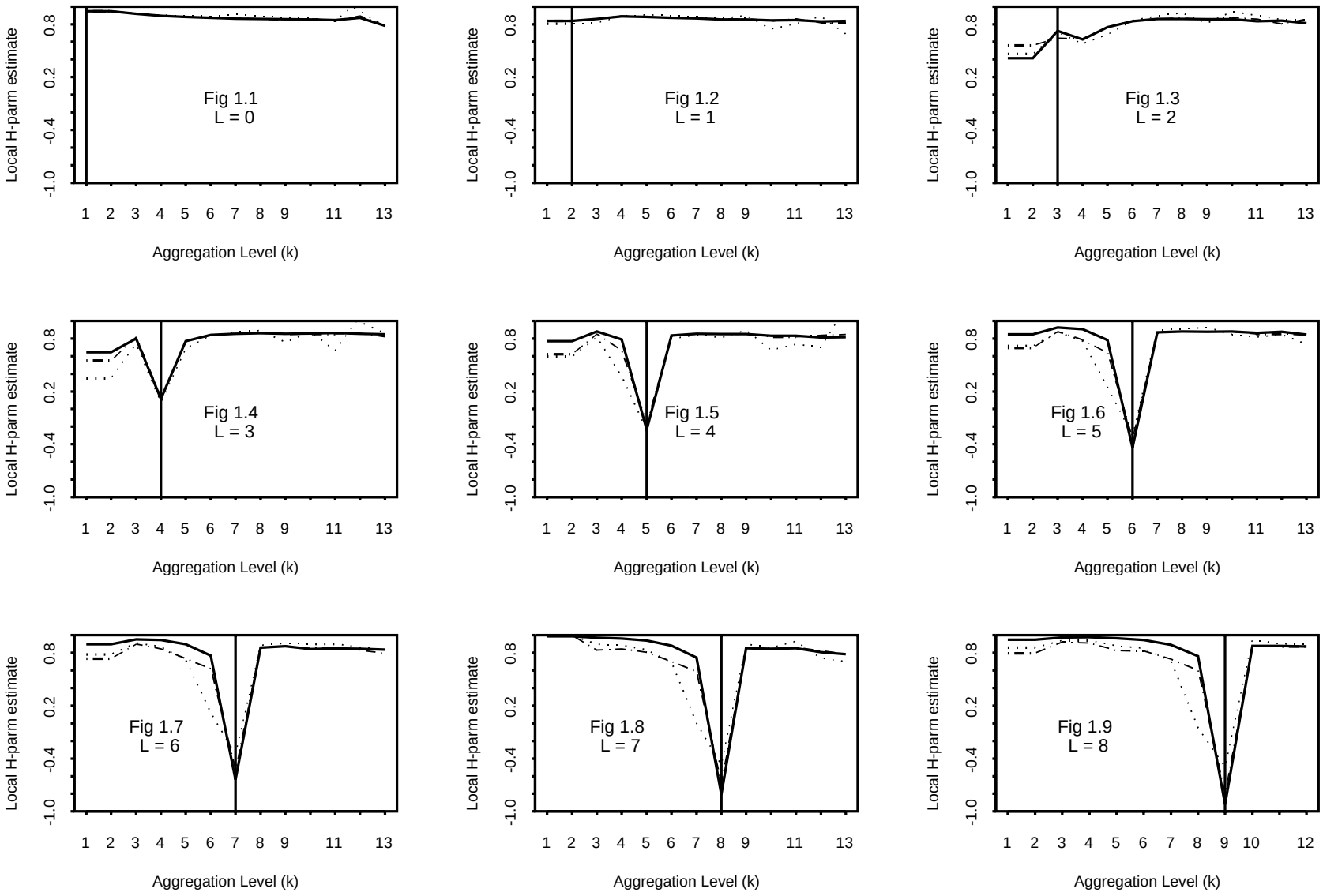


Fig. 1: Local H parameter vs. aggregation level for slot_size=25 (q=2: solid line, q=3: dotted line, q=4: dashed line)

Vertical lines drawn at L+1 where L is the maximum time-scale for mass redistribution

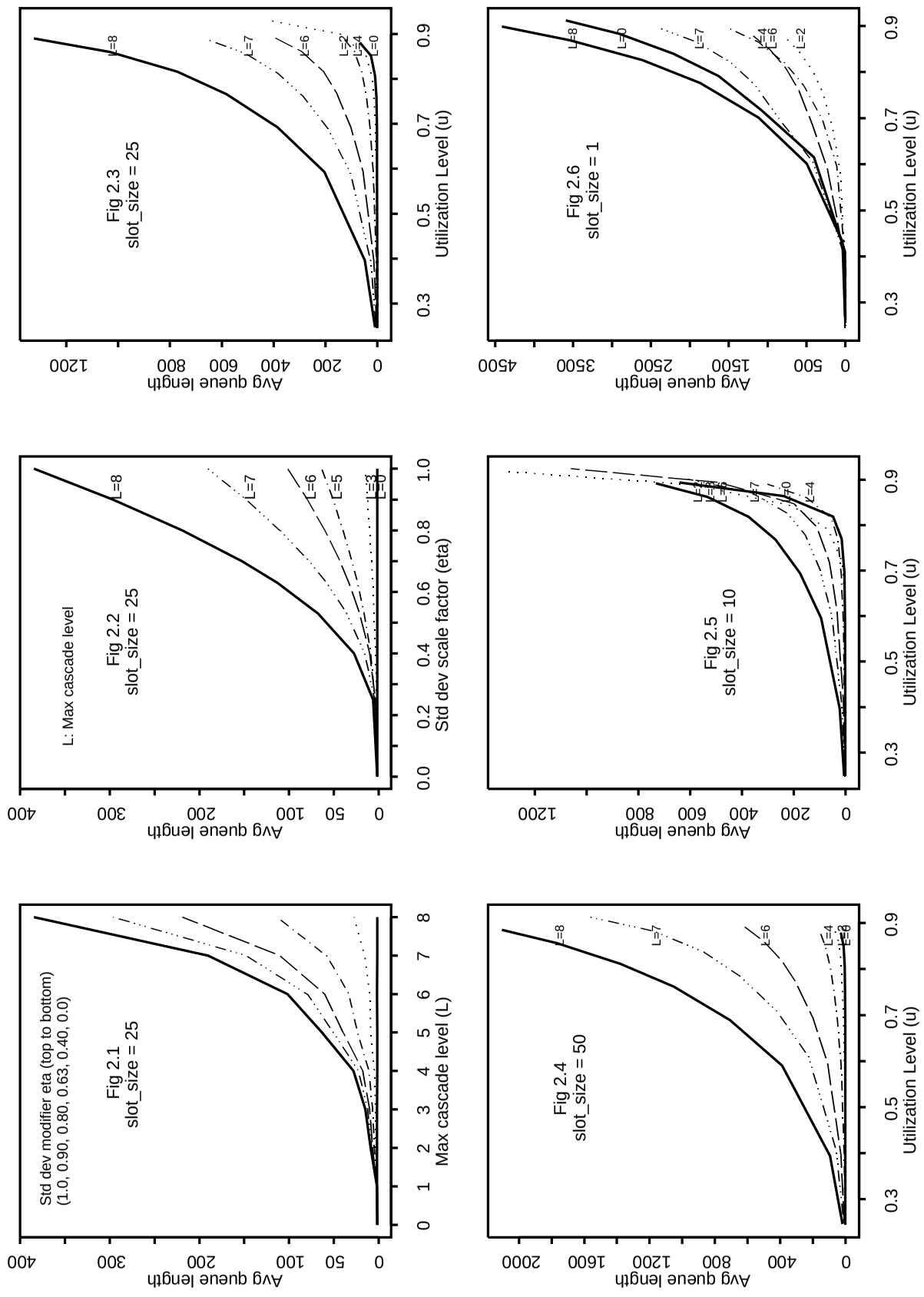


Fig. 2: Avg Queue Length vs. misc system parameters

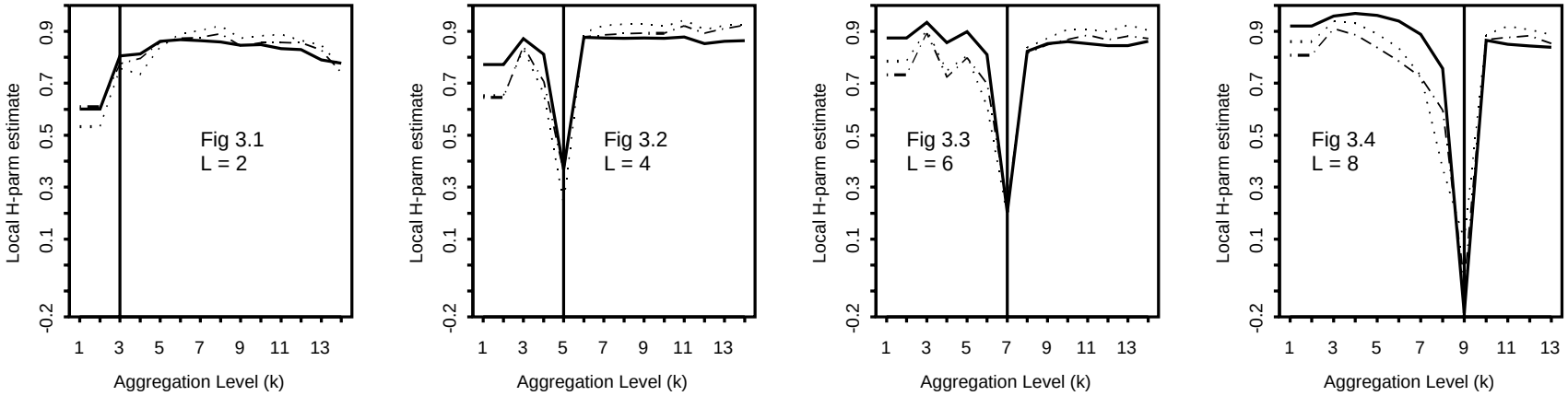


Fig. 3: Local H parameter vs. aggregation level for slot_size=5 (q=2; solid line, q=3: dotted line, q=4: dashed line)
Vertical lines drawn at L+1 where L is the maximum time-scale for mass redistribution

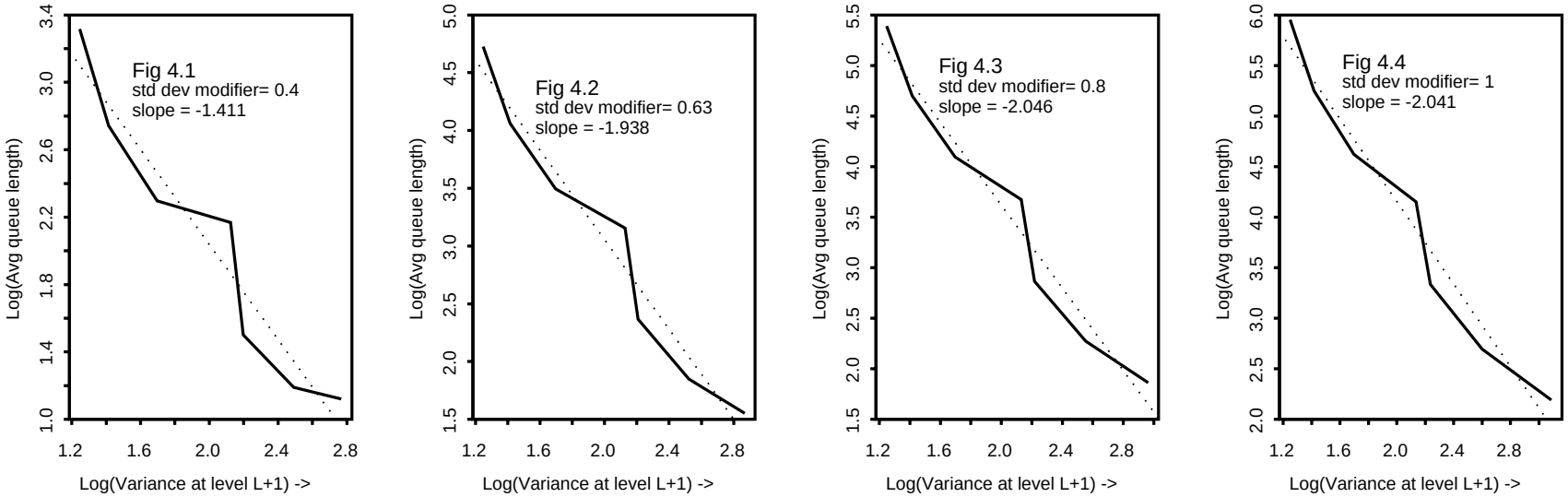


Fig. 4: Log(Avg queue length) vs. variance at time-scale with smallest H value