Impact of Flow Control Windows in TCP on Fractal Scaling of Traffic Exiting From a Server Pool

Heung-No Lee
Information Science Laboratory
HRL Laboratories, L.L.C.
3011 Malibu Canyon Rd.
Malibu, CA 90265
E-mail: heungno@hrl.com

Abstract—**We provide an analytical and network-systematic framework to characterize the self-similar, fractal scaling phenomenon which is believed to be ubiquitously present in modern high speed data network traffic. We show that the self-similar network traffic is due mainly to the use of closed-loop flow control at the transport layers, such as the use of the classes of protocols from the TCP family. For in-depth investigation on the subject, we inject synthetically generated application-level traffic which is completely short-range traffic, into a very simple network simulated with NS2.0, and examine the influence of different parameters of a TCP algorithm on a variety of different fractal scaling behaviors observed at the packet-level traffic traversing a link in the simulated network. We provide a very simple–but intuitive–mathematical explanation of the observed phenomenon using the shot-noise processes. Specifically, different kernel filters of the shot-noise are constructed to model the behavior of the window process *cwnd* in different stages of the congestion avoidance algorithms employed in a TCP algorithm. With the use of *exponential-law* shot-noise process, for example, we indicate that the *cwnd* process in the *slow start* phase results in a unique scaling behavior from RTT to a finer time scale, having a scaling slope of $\gamma = 2$. From RTT to a coarse time-scale, the more conventional fractal scaling behavior with the Hurst parameter less than 1.0 is observed; we compare this with the *rectangular-pareto* shot noise process.**

Keywords—**Shot-noise process, self-similar processes, long-range dependence, heavy-tailed distribution, pareto distribution, TCP congestion controls, and traffic characterization and modeling.**

I. Introduction

SINCE the salient discovery by Leland, Taqqu and Willinger [1] of a self-similar or fractal-like scaling phenomenon on the measured Ethernet LAN traffic, numerous empirical studies have been conducted based on high-speed and high-resolution network traffic measurements on a variety of different communications networks. This new discovery about network traffic has directly impacted the practice of network engineers in a variety of different ways, such as in developing more appropriate traffic models, in designing different traffic control functions, in evaluating the performance of a network, and in finding the capacity regions of a network. Before we attempt to accomplish these engineering objectives, we feel that it is a prerequisite to investigate the underlying causes more carefully and develop a more network-systematic understanding of the fractal scaling behavior marked in the trace data. Accordingly, we attempt to investigate why the network traffic displays fractal-like scaling behaviors over a broad range of time-scales, from a variety of perspectives which have not been considered in previous studies. These previous studies may include those works largely based on the following two major propositions: the superposition of heavy-tailed On/Off processes and the multifractal processes [1][2][3][4][5][6]. The scaling analysis for these works was performed mostly on a "live" traffic, collected off of a real network in operation. The other group of works may include those in [7][8][9] in which the traces analyzed were collected also from a simulated network using the Network Simulator(NS). With this "active observation" approach of investigating the underlying scaling behavior, we attempt to ascertain our novel proposition that the fundamental cause of data network traffic being self-similar is attributable more to the closed-loop congestion control and avoidance algorithms employed at the transport layer, such as the use of transport control protocols (TCP), than to the workload traffic being heavy-tailed in file-size distribution as has been widely believed.

A. Contributions

We introduce the use of the shot-noise process which helps construct an organized network-systematic traffic analysis and observation framework. We adopt the approach of *active observation* and conduct scaling analysis based on simulated network traces. Unlike previous works, the impact of TCP window mechanism on the fractal scaling phenomenon of the simulated traces are quantitatively explained, as well as qualitatively, in one-on-one comparisons with different versions of the shot noise process. Particularly, we attempt to provide a complete layout of fractal scaling mechanism in the data network traces, starting from the application-level traffic and all the way down to the packet-level aggregate traffic. We start with modeling the source (workload) traffic at the application layer, which is designed to have short-range correlation (in fact it is memoryless process). This short-range traffic will be injected into each of the TCP Senders in the simulated network. Many of the TCP flows are multiplexed at the router to which a bottleneck link is connected. Then, the scaling analysis is performed using the packet trace files collected off of the link. The scaling properties of the traces are compared with those of shot-noise processes which are excited by exactly the same workload traffic. Thus, the spectrum analysis of the packet traces can be directly compared with that of the shot-noise processes. This novel framework enables us to examine the influences of many different parameters, involved in shaping the overall spectral shape of the traces, including those in the workload level traffic, the transport layer, the network multiplexer, and in a TCP algorithm, and provides a more complete and systematic explanation of different scaling behaviors observed in the packet traces.

For NS2.0 simulation, we use a very simple network (see Fig.

0-7803-7206-9/01/$17.00 © 2001 IEEE

1; detailed explanation will be followed in simulation section) with a single bottleneck link to reduce the number of scenarios with different combination of parameters to be simulated. Our focus will be placed on investigating the impact of two bottle neck parameters C_{21} and D_{21}. The simplicity of simulation configuration greatly helps us to illustrate and compare the scaling behavior with the power spectrum of the shot-noise. From

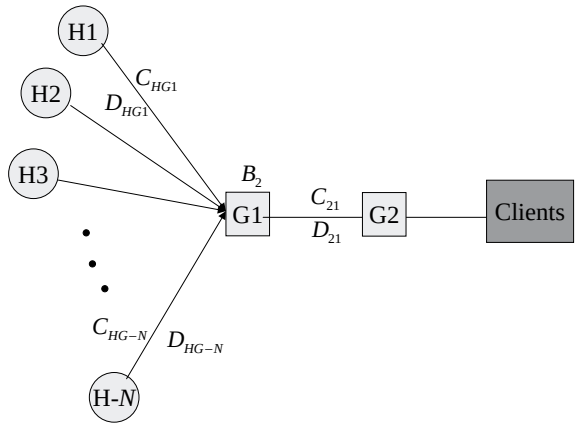

Fig. 1. One link configuration

the experiments, we produced the following list of novel contributions. First, we show that the TCP algorithm at the *slow start* phase brings about a unique scaling behavior over the range of time-scales from the time-scales of a packet transmission time T_p to those of *capacity* with the parameter H of 1.5 [1]. The cause is analyzed with the use of *exponential-law* shot noise. It is due to the *slow start* phase of TCP in which the window-size $cwnd$, the number of packets TCP sends without a reception of an ACK, grows exponentially. This unique scaling behavior would disappear at the time-scale above RTT since the size of the packet stream cannot be increased more than the size of $capacity$ (more explanation in section IV.). The spectrum analysis of exponential-law shot-noise process provides the reason of having the parameter $H = 1.5$. Second, we identify the second unique scaling region, from RTT to a higher time-scale (become more specific in section IV.), in which region the parameter H is again 1.5. This scaling behavior and the slope can be accounted for with the use of the *rectangular-law* shot noise. Third, we find that the more conventional fractal scaling behavior having the parameter $0.5 < H < 1.0$, immediately follows the second scaling region. To explain this, we use the *rectangular-pareto* shot noise process. Lastly, we show in general that the network parameters, the TCP parameters, the distribution of source traffic–all are actively involved in making a particular scaling behavior observable at a particular time-scale. This contradicts the previous statement made by Park [7] and Feldmann [9] that

[1] In this paper, we follow the sample spectrum definition of the $1/f^{\gamma}$ noise with $H = \frac{\gamma+1}{2}$ [10]. Thus, except for $0.5 < H < 1.0$–in which case the parameter H is the same as the Hurst parameter and the process is stationary so that we can safely define the autocorrelation function (from the use of the Wiener-Kinchine theorem) and the fractal dimension $D = 2 - H$–the use of parameter H in this paper merely points to the scaling slope γ by the linear relationship.

details of network does not affect the self-similarity of the network traffic.

B. organization

This paper is organized as follows. We omit explanation on TCP algorithms due to limited spaces and refer to [11]. Section II discusses the shot-noise process. In Section III, we discuss the distribution of response size to be used in making the source traffic in our NS2.0 simulations. Section IV provides NS2.0 simulations and discuss the results. Section V is the conclusion of the paper.

II. Shot Noise Processes

In this section, we briefly discuss a few mathematical analysis results from shot-noise theory to facilitate better understanding of scaling behavior marked in traces. We first discuss the shot-noise process as a network-systematic model. We then discuss the three kernel filters and their power spectral densities. Finally, we discuss the limitations of shot-noises as a complete network systematic model.

A. Network-systematic model

We define the shot-noise process $s(t)$ as a weighted and filtered Poisson process

$$s(t) := \sum_{n=-\infty}^{\infty} w_n h(t - t_n) \tag{1}$$

where
- t_n is a homogeneous Poisson point process with rate λ,
- w_n is an independent identically distributed (i.i.d.) random process,and is a weight assigned to each arrival,
- $h(t)$ is a linear time invariant kernel filter of the process.

This definition of the shot-noise process may be used to construct a structured network-systematic model to explain the user-level source traffic incoming to the network as well as the aggregate packet-level network traffic, and serves as the mathematical counterpart of the simulation using NS. In the following we address the shot-noise model from the perspectives of first source traffic, and then, aggregate traffic.

A.1 Source traffic perspectives

First, we propose a homogeneous Poisson arrival process to model the application level, session arrivals. At the application- or user-level the arrivals, the "weak law of small numbers" or "law of rare events" still holds for a number of independent users, and thus Poisson convergence follows. Specifically, in NS2.0 simulations, we have a total of N TCP Senders. First, the arrivals are generated according to the definition of Poisson process. And then, each arrival is assigned to a TCP Sender, uniform-randomly selected from N; Each selection is independent of any other selection.

Second, each arrival is modulated by a weight w_n which denotes the size of a single response associated with the arrival of a single application session or a single user. The random variable w_n is independent, identically distributed. We will discuss the distribution of w_n more in section III. With these two, the source traffic model is completed. It should be noted that the source

traffic process is *memoryless* due to Poisson arrivals as well as to $iid\ w_n$, and thus it is a *short*-range process. The source traffic process $z(t)$ can be defined as the weighted train of impulses

$$z(t) := \sum_{n=-\infty}^{\infty} w_n \delta(t - t_n) \tag{2}$$

where $\delta(\cdot)$ denotes the Dirac delta function. Now with $w_n = 1$ for all n, we have the followings. The expected value of the process is

$$E\{z\} = \lambda. \tag{3}$$

This follows from taking the derivative of the expected value of the Poisson count process λt. This can be easily follows from the fundamental theorem for any linear system which states that the linear operations are commutative–the expectation operation and differentiation are commutative. The autocorrelation function is given as follows

$$R_z(\tau) = \lambda^2 + \lambda\delta(\tau). \tag{4}$$

The Fourier transform of $R_z(\tau)$ gives the power spectral density

$$S_z(f) = \lambda^2\delta(f) + \lambda. \tag{5}$$

A.2 Aggregate traffic perspective

In this research, a set of kernels $h(t)$ will be used to give coarse imitations of the variation of window size *cwnd* in a single TCP session. As explained in [11], the *cwnd* process follows a set of rules provided in a particular TCP algorithm, exhibiting difference behaviours in a different phase of a TCP. In this research, in order to simplify our tasks, we shall only attempt to imitate the $cwnd$ process within the congestion avoidance region of a TCP with the use of three kernel filters. That is, $h(t)$ will model the $cwnd$ process of a TCP (TCP Reno is used in this paper) when it is poised in one of the following three situations:

- *slow start*, using the exponential-law kernel
- after $cwnd$ reached the maximum window size or the $capacity$ (defined in section IV) of the bottleneck link, using the use of the rectangular-law kernel
- many TCP Senders in competition, each with the Pareto distributed workload, using the rectangular-pareto kernel

The three kernel filters will be discussed in the following subsection, where the impulse and frequency responses of the filters are described. It should be noted that the frequency response of the filter determines the overall shape of the power spectrum of the shot-noise. For instance, the power spectral density $S_s(f)$ [2] of the shot-noise (1) can be obtained from

$$S_s(f) = S_z(f)\,|H(f)|^2 \tag{6}$$

$$= \lambda^2 H^2(0)\delta(f) + \lambda\,|H(f)|^2\,. \tag{7}$$

In addition, the mean and variance of the shot-noise process can be computed as

$$E\{s(t)\} = E\{w_1\}\lambda \int_0^\infty h(t)dt, \quad \text{and} \tag{8}$$

$$Var(s(t)) = Var\{w_1^2\}\lambda \int_0^\infty h^2(t)dt. \tag{9}$$

[2] We should note that (6) applies only when the autocorrelation function of the shot-noise process exists, because (5) is obtained from the Fourier transform of the autocorrelation function.

B. Three kernel filters and their power spectrums

We now discuss our three kernel filters, the exponential, the rectangular-deterministic and the rectangular-pareto kernels and their power spectral densities. Exponential-law kernel is to model the operation of the window $cwnd$ during the *slow start* phase of the TCP. Rectangular-law kernel is to model the behavior of the window $cwnd$ after it reaches its maximum bound, $window$ or $capacity$ (see further explanation in section IV.). The rectangular-pareto shot-noise is to model the aggregate process when the distribution of file size is heavy tailed. Accordingly, it should be noted that, we are targeting different regions of time-scales for each filter, which should become more clear with simulation examples in section IV.

B.1 Exponential-law kernel

The exponential-law kernel is defined as

$$h_e(t) := \mathrm{e}^{-\theta t}U(t), \tag{10}$$

where θ is the half-power frequency and $U(t)$ denotes the unit step function. We should note that if we let

$$\theta := \frac{log_2 \mathrm{e}}{RTT}, \tag{11}$$

then $h_e(t) = 2^{-\frac{t}{RTT}}$, which may represent the number of packets a TCP sender can transport for the duration of a single round trip time within the *slow start* phase (assuming a constant RTT for convenience of analysis). The filter's frequency response can be obtained from the Laplace transform evaluated at $s = j2\pi f$,

$$H_e(f) = \frac{1}{j2\pi f + \theta}, \tag{12}$$

and thus the power spectrum of the filter is

$$|H_e(f)|^2 = \frac{1}{4\pi^2 f^2 + \theta^2}. \tag{13}$$

Then, we have the following results

- $$E\{s(t)\} = \lambda/\theta \tag{14}$$
- $$Var(s(t)) = \frac{\lambda}{2\theta} \tag{15}$$
- $$S_e(f) = \frac{\lambda^2}{\theta^2}\delta(f) + \frac{\lambda}{\theta^2 + 4\pi^2 f^2} \tag{16}$$
- $$C_{exp}(\tau) = \frac{\lambda}{2\alpha}\mathrm{e}^{-\theta|\tau|} \tag{17}$$

The exponential-law shost noise (ELSN) will become very useful to be compared with the wavelet spectrum analysis results of the traces at the region of time-scales of the *slow start*. In general, ELSN does not possess a $f^{-\gamma}$ frequency spectrum. In fact, the kernel $h_e(t)$ is a low-pass filter as shown in Fig. 2. However, when the value of half-power frequency $\Omega := 2\pi f = \theta$ is small relative to the length of the trace, the low-frequency region of the flat-response may not be resolved with the spectrum analysis on a discrete-time sequence, due to the limited length of

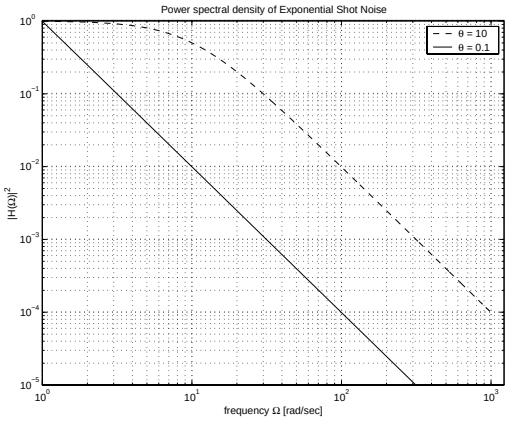

Fig. 2. Power spectrum of the kernel filter $h(t) = \mathrm{e}^{-\theta t}$

the sequence. In such a case, what we would observe may be the transition region only (ascending linearly in log-log scale), which extends from the minimum time-scale to the maximum time-scale–the length of the trace. This sample spectrum would be looking like a $f^{-\gamma}$ spectrum. In Fig.2, for example, when $\theta = 0.1$ rad/sec, what we observe is the sample spectrum of $f^{-\gamma}$ with $\gamma = 2$, which corresponds to $H = 1.5$.

B.2 Rectangular-determinstic kernel

We now consider the rectangular-law kernel,

$$h_r(t) := U(t) - U(t-T), \tag{18}$$

where T denotes the length of the rectangular pulse. This filter may be useful to model the TCP window process after $cwnd$ is grown to the maximum allowed window size $window$ or the $capacity$ of the link.

The Laplace transform of $h_r(t)$ is $(1 - \mathrm{e}^{-\mathrm{sT}})\frac{1}{\mathrm{s}}$. Evaluating at $s = j2\pi f$, we have $h_r(t) = T \cdot \mathrm{e}^{-j\pi fT}\frac{sin(\pi fT)}{\pi fT}$. Then, the power spectrum of the rectangular pulse is

$$|H_r(f)|^2 = T^2(\tfrac{sin(\pi fT)}{\pi fT})^2. \tag{19}$$

It is to be noted that (a) the envelop of the power spectrum decays as $f^{-\gamma}$ with $\gamma = 2.0$ and that (b) the width $\frac{2}{T}$ of the main lobe shrinks as the length of the pulse T increases. From the second observation, it follows that as the length of the pulse T increases, f^{-2} behavior is extended toward the lower and lower frequencies. In section IV., this rectangular-law shot noise process will be compared with the network traces collected off of a bottleneck link in which a TCP sender takes a prolonged traffic ON-time to complete the transport of a response either due to a limited window size and interruption of other TCP traffic or due simply to a very large response size .

B.3 Rectangular-Pareto kernel

The kernel of the rectangular-pareto shot noise process is defined as

$$h_p(t) := U(t) - U(t-T). \tag{20}$$

where T is Pareto distributed with the probability density function

$$\Pr\{T = t | T > t_o\} = \alpha t_o t^{-\alpha-1} \tag{21}$$

where $1 < \alpha < 2$ and $t_o > 0$. The power spectral density of the Rectangular-Pareto shot-noise can be obtained by taking the *expectation* of $S_s(f)$ (6) with respect to $H(f)$, and can be written as

$$S_p(f) = \lambda^2\delta(f)|E\{H_p(0)\}|^2 + \lambda E\{|H_p(f)|^2\}. \tag{22}$$

Again the second term, $E\{|H_p(f)|^2\}$, determines the overall shape of the power spectral density. Now we evaluate the integral to investigate the scaling slope in the frequency-domain. Let's use $H_p(f; T = t)$ to denote the rectangular window of size t. Then, we have

$$\begin{aligned} E\{|H_p(f)|^2\} &= \int_0^\infty H_p(f; T = t)\Pr\{T = t\}dt \\ &= \int_0^\infty (\tfrac{sin(\pi ft)}{\pi f})^2 \alpha t_o t^{-\alpha-1} dt \\ &\simeq \Gamma(-\alpha)\,|cos(\alpha\pi/2)|\,(2\pi f)^{-(2-\alpha)}, \end{aligned} \tag{23}$$

where $\Gamma(\cdot)$ denotes the complete Gamma function [3]. Thus, we have found the relationship between the Pareto parameter α and the spectral scaling slope γ, which is $\gamma = 2 - \alpha$. Then, using the relationship $H = \frac{1+\gamma}{2}$, we have the following relationship between the parameter H and the Pareto parameter

$$H = \frac{3-\alpha}{2}, \tag{24}$$

where the parameter $H \in (0.5, 1.0)$–the Hurst parameter–for $\alpha \in (1.0, 2.0)$. We should note that the same relationship between the Hurst parameter and the Parato parameter was obtained from the limit theorem [1].

The Rectangular-Pareto shot-noise process will be useful to explain the scaling behavior observed from the corresponding time-scale of transporting w_o (defined precisely in section III.) and the larger time-scales.

C. Limitation of shot-noise as the complete model

It would be worthwhile to mention a couple of points on the fundamental limitations of the shot-noise framework as a complete model. First, the kernel filters are time-invariant and thus cannot exactly model the dynamic, time-varying nature of the TCP window processes. We settled for having three different filters for there different operation regions. Second, the maximum traffic rate any aggregate traffic can reach is limited in network traces, while it is unlimited for the shot-noise. For examples, the exponential-law shot noise will approach a Gaussian in amplitude distribution as the Poisson driving rate λ increases beyond $1/\tau_h$, where $\tau_h := \frac{|\int h(t)dt|^2}{\int h^2(t)dt}$. For TCP traffic, when λ increases beyond a certain rate the aggregation of TCP flows will create a network congestion and in effect each TCP will reduce the traffic rate in accord with the feedback mechanism. This time-varying nature of the TCP cannot be accurately captured with the proposed framework based on the time-invariant filters. It should be noted that our purpose of using shot-noise is not an attempt to model the TCP window process exactly, but to study the fractal scaling behavior more systematically.

[3]Hint: Use $sin(at) = \frac{\mathrm{e}^{\mathrm{i}at} - \mathrm{e}^{-\mathrm{i}at}}{2\mathrm{i}}$ and $\int_0^\infty \mathrm{e}^{-ct}t^{b-1}dt = \Gamma(b)c^{-b}$, $b \neq 0, -1, -2, \ldots$

III. Generation of Response Random Variable w_n

This section discusses the distribution of response data size w_n and the method to generate samples. It is perhaps well known that the tail distribution of response size in bytes is heavy-tailed, or more precisely pareto distributed with $1.0 < \alpha < 2.0$. For example, Crovella et al [12] report some of the response data size distribution based on their collection at Boston University, which consists mainly of file transfers with the estimate of α close to 1.1. Willinger et al [13] also report the estimates of α, 1.1 for FTP session sizes and 1.35 for HTTP session sizes. Based on these findings, we have no doubt that the tail distribution of response size is a Pareto with $1 < \alpha < 2$.

Figure 3 shows the complementary CDF of HTTP response data size collected at UC Berkeley Web Trace 8468. It should be noted that the upper figure uses a linear-log scale to emphasize the distribution for small response sizes and about 90% of the response is smaller than 10 Kbytes, whereas the second figure shows a log-log plot and we can clearly see it follows the parato distribution. The smallest value x_o of the pareto reference curve is determined as $x_o = \bar{X}\frac{\hat{\alpha}-1}{\hat{\alpha}}$, where $\hat{\alpha}$ is estimated from the least square curve fitting in the region of 2^{12} through 2^{23}. From this, we obtained $\hat{\alpha} = 1.49$ and $x_o = 2449$.

In this paper, we certainly use the pareto-tail distribution to generate response samples of w_n. We have chosen the mean value, i.e. $\alpha = 1.5$ for the generation of pareto w_n. In addition, we have also considered the followings: Firstly, we have noticed that in general only 10% of the responses follows the heavy-tailed distribution and the rest of 90% does not. For example, consider the top figure of Fig.3, a linear-log plot emphasizing the non-pareto region. It is noticeable that the empirical curve significantly deviates from the pareto-reference. Since this implies that about 90% of the responses are completed within the *slow start* phase of a TCP connection, it might be beneficial to carefully reflect the empirical distribution in generating the random samples. Thus, we modified the pareto random variable to create a new compound random number generator, named $downrnd$. CCDF of *downrnd* is constructed from two different distributions each of which has a separate domain of support. The first part is an exact copy of the empirical distribution, from 0 up to w_o, where w_o denotes the response size with $Pr[w_n > w_o] = 0.1$. The second part simply uses the pareto random number generator with $\alpha = 1.5$. Fig.4 verifies the generator $downrnd$ where the complementary distribution constructed from the histogram of 60,000 $downrnd$ samples is compared with the pareto-reference.

Secondly, we have also noted that the distribution of FTP essentially has the same shape as that of HTTP, but with a few orders of magnitude larger in size. Thus, we modified $downrnd$ to generate the sample for FTP. The first part, up to w_o, is generated by multiplying the empirical distribution of HTTP with 10^d, where $d = 1, 2, \ldots$ is the order of magnitude shift. The second part is created from the pareto with $\alpha = 1.5$. Fig.5 shows the verification of the method with two order of magnitude shift, i.e.,$d = 2$.

Thirdly, we also created completely artificial distributions, named skewed-HTTP and skewed-FTP, for the convenience of investigating the impact of pareto tail distribution in response size on fractal scaling of the network traffic. They were created

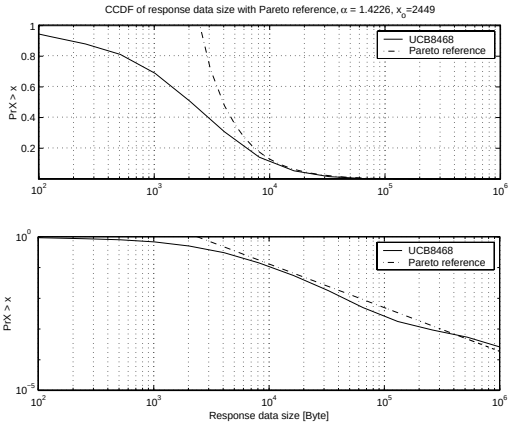

Fig. 3. Complementary CDF of workload size, UC Berkeley Web Trace 8468

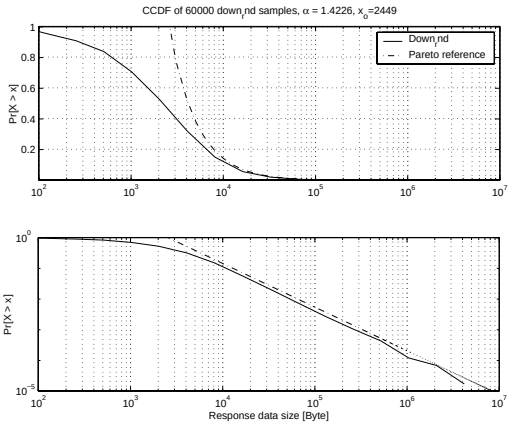

Fig. 4. Complementary CDF of 60000 downrnd samples for HTTP-distribution

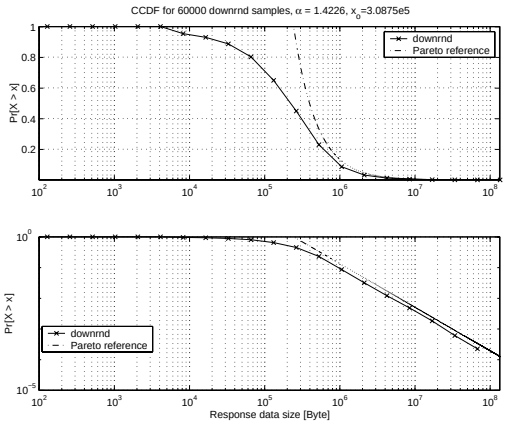

Fig. 5. Complementary CDF of 60,000 downrnd samples for FTP-distribution

basically by lumping all tail probabilities in the pareto region, i.e. $w_n \geq w_o$, to w_o. That is, $\Pr\{w_n = w_o\} = 0.1$.

IV. Investigation on Traffic Self Similarity

In this section, we discuss NS2.0 simulation results. It is our intention to have simple experiments, and thus to stay intuitive. We first provide some of simple rules of scaling analysis based on the concept of *sample spectrum*, to effectively–but not rigorously, address the problem of having multiple distinct scaling regions. We then move on to conduct some *single shot* analysis. The single shot experiments are designed to investigate the impact of the *slow start* in TCP on the scaling behavior of smaller time-scale regions, typically around a round trip time (RTT) delay. The spectrum of a single shot shall make a good comparison with the scaling laws observed in the *exponential-law* shot noise. Finally, *multiple-shot* analysis will be given. They are designed to investigate the larger time-scale behaviors having a longer traces. From multiple-shots experiments, we should be able to examine the causes of self-similar behavior existing at the larger time-scales, which may be compared with the *rectangular-pareto* shot noise processes.

A. Simulation Configuration

Fig.1 is the network configuration with a single bottleneck link. We inject the systematic source traffic discussed in section III.A, which is basically a Poisson request arrival process with independent response sizes. Each of N TCP servers is connected to a server node Hi, $i = 1, \ldots, N$, each Hi is connected to the gateway router G1, and G1 connects all the flows to the clients. We intend to observe the traffic flows from G1 to G2 from which we can collect the server-to-client traces, but not the packet transitions from G2 to G1 which would be consisted only of ACKs and requests. We want the simulation to be very simple and thus we have the following simulation setup:

- Each of the links from Hi, $i = 1, \ldots, N$, to G1 has the same link speed and delay, to have a single RTT condition for all TCP flows [4].
- The delay and link-speed from Hi to G1 is set to have a zero delay 0.0 sec and 10 Gbps respectively, for all i, $i = 1, \ldots, N$. This is to ensure the link from G2 to G1 is the bottleneck.
- The buffer size at G1 for the outgoing link G1-G2 is set to a very large value, i.e. 10,000 packets. This is to ensure no packet drop so that the TCP is to be remained in the congestion avoidance region throughout the duration of a simulation. Thus, in this paper we will not deal with the TCP behavior in congestion control phases such as time-out, fast retransmit and fast recovery, which involves other feedback window mechanisms and differs from what the three shot-noise filters discussed in this paper can justify.
- The buffer employs drop-tail and FIFO rule.
- The link, G1-G2 is a duplex link and thus the up-stream traffic from G2 to G1, mostly ACKs, do not interfere with the down stream traffic.

In such simulation setup, we vary the following parameters:

- C the link-speed of G1-G2, ($C := C_{21}$ shown in the figure)
- D the propagation delay of G1-G2, ($D := D_{21}$ shown in the figure)

Before moving forward, we briefly introduce the notion of *capacity*, which is defined in this paper as,

$$capacity \text{ [packets]} = \frac{C_{21} \text{ [bits/sec]} \times \text{2D[sec]}}{8\text{[bits/byte]} \times 1000\text{[bytes/packet]}}. \quad (25)$$

Thus, the *capacity* is twice the bandwidth-delay product (BDP) $C \cdot D$ of the bottleneck link, expressed in the units of a packet. BDP provides a measure of an ideal TCP window size, such as how big the window size should be, to send the data at the maximum rate without causing the packets queued up at the buffer of the bottleneck router. Then, what is the need of defining the *capacity*? The $capacity$ provides the notion of the minimum round trip delay of the link: It provides the measure of the time duration in units of T_p how long it takes to receive an ACK from the instance a packet is sent. If the window size is greater than the capacity, the link is fully occupied, such as serving the stream of packets back to back. Later, we will observe a sharp slope change in the wavelet power spectrum at the time-scale located around the *capacity*. $capacity$ is also used as a handle to either increase or decrease the level of contention among TCP Senders. In *multiple-shot* analysis section, we fix the number of TCP Senders and vary the size of *capacity* in two cases, *small capacity* and *large capacity*.

B. Some clarification items on scaling analysis

The self-similar processes which have been more frequently referred in the literature of traffic-engineering such as fractional Gaussian, M/G/∞, FARIMA, etc. are defined only for $0.5 < H < 1.0$. On the other hand, we are about to observe a couple of unusual scaling analysis results from the traces obtained from simulation. They often indicate values of the parameter H greater than 1.0. In fact, it is possible to generate the sample path of a self-similar process which has the value of parameter H outside the region $0.5 < H < 1.0$ (for example see [14]), using the frequency-domain approach [15]. These discrete-time sample path of a self-similar process with any value of parameter H can be well described by the more general class of $f^{-\gamma}$-noise [16] where the power exponent γ is related with the Hurst parameter as,

$$H = \frac{1+\gamma}{2}. \quad (26)$$

It should be noted that for the non-stationary process with $H > 1.0$ the time-domain analysis tools which is based on the assumption of stationary process does not work; the frequency domain analysis tools such as the periodograms and wavelets continue to work. Thus, we chose the basic spectrum analysis tools developed by Avry and Veitch [17] with the choice of vanishing moment of 3.

We encounter traces having multiples of different scaling regions each of which can be better described by a unique scaling-slope. For this, we would like to introduce following clarification items which is based on the concept of *sample spectrum*.

- Define the basic region of support: The smallest time shall be the packet transmission time T_p. The largest time shall be the duration of the trace T_D. Then we overcome the problems of infrared and ultraviolet catastrophe.

[4] It would be very interesting to investigate the effect of distributed RTTs as it also has a high variability in the real Internet case. However it is beyond the scope of this research effort.

• Within the time-scale of T_p to T_D, there may be many distinct scaling regions. We describe each of them with a certain region of support and an associated scaling slope γ or H.

C. Single-shot analysis

In this section, we want to focus on the small time-scale behavior of the traces, especially coupled with the *slow start* of TCP. Every time a TCP starts a session, it must start with *slow start* before moving into the next phase of congestion avoidance or congestion control. In addition, as discussed in section III, about 90 percent of responses in HTTP are involved with the response size smaller than 10 packets and 1000 packets for FTP. This implies that about 90 percent of the responses would likely be completed within the phase of *slow start*. From the shot-noise theory, we learned that the frequency characteristics of the modulating window, the filter $h(t)$, determines the power spectrum of the shot-noise process. They motivate us to characterize the frequency response of the TCP window algorithm within the *slow start* phase. We may investigate this by sending one impulse, which is equivalent to a single transfer of a fixed size response data. In order to examine consistency of our proposition, we have varied the size of $capacity$ and the size of the maximum bound on window size, *window*.

In order to facilitate a better understanding of the scaling analysis results, we now consider a spectrum analysis on the $slow\ start$ window process based on the Z-transform. Let us consider a link serving a flow of a single shot in the *slow start* phase. First, we choose a parameter P to denote the $capacity$ for the link. Now, let's consider the number of packets that are consecutively transmitted over the link. At the first round, the TCP sender sends one packet and gets an ACK in P T_p, at which moment it sends 2 packets in two consecutive units of time T_p. Thus the link will be busy for two consecutive T_p and off for $(P-2)T_p$, and so on. Representing this operation in Z-transform, using T_p as one unit of discrete time we have

$$S(z) = \sum_{k=0}^{l-1} Z^{-kP} \frac{1 - Z^{-2^k}}{1 - Z^{-1}}, \tag{27}$$

where l denotes the power exponent in response size w_1 as

$$w_1 = 2^{l+1} - 1. \tag{28}$$

Fig. 6 is the results of Z-transform evaluated at $Z = \mathrm{e}^{j\omega}$ for $P = 8$, and $l = 4, 5, 6$, and 7. We should note that the slope parameter γ increases to 2.0, and thus $H = 1.5$ from (26), as l increases to P and all the low-pass filter characteristics visible at $l = 4$ disappears. This provides, together with the exponential-law shot noise in section III., the necessary explanation of the unique scaling behavior from the time-scale T_p to $capacity$ having the parameter $H = 1.5$. We will see this again with NS2.0 simulated traces.

We now discuss the NS-simulation results. For the first experiment, the maximum allowed window size *window* was set to 20 packets. $C = 3$Mbps and $D = 60$msec. Thus, $T_p = 2.7$msec and *capacity* of the link is 45 packets. The response size w_1 of a single shot is controlled by (28) with $l = 2, 3, \ldots$

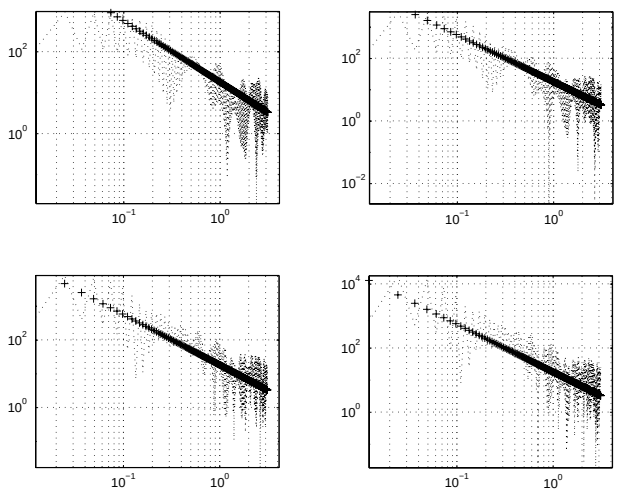

Fig. 6. Sample spectrum using the Z-transform of *slow start* shots, $l = 4, 5, 6$ and 7, clockwise starting from the upper left figure, at *capacity* $= 2^8$

Fig. 7 shows the scaling analysis on the single shot transportation of w_1 with $l = 4$. The first figure indicates the evolution of window size *cwnd* over time. At $t = 0$, *cwnd* starts with 1 packets. RTT is about $2D_{21} = 120$msec. At the first RTT delay, the TCP sender receives the first ACK for packet number 1 and increases *cwnd* by 1. Thus, TCP sender sends two packets. Around the second RTT delay, the TCP sender receives two ACKs for packet number 1 and 2, and increases *cwnd* to 3 and 4, each offset by T_p. This process goes on until *cwnd* reaches the maximum allowable window size, *window* = 20. The second figure shows the wavelet transform results of the $cwnd$ process. The third figure indicates the byte time-series which is obtained by recording the total number of bytes traversed the bottleneck link at every T_p. This is obtained from the two-column time-series, the first column of which denotes the time-stamp of packet and the second denotes the number of bytes of the packet which were received at $G2$. Since each tick represents one T_p and every packet has the same size of 1000 bytes. The ordinate value of the time series should take only two values, 0 or 1000. The fourth figure is the wavelet transform of the byte time-series.

We first note that the *cwnd* process possesses almost the same scaling characteristics as the byte time-series does. As in convention, we estimate the parameter H from the scaling slope of the wavelet power spectrum of the byte time-series. It should be noted that as our analysis using the Z-transform indicated already, the fractal scaling with $H = 1.5$ is obtained again. In fact, the ascending time-scales extends from $k = 1$ to $k = 5$ with $H = 1.381$. Finally note that $log_2\{window\} = 4.3219$, which is around $k = 5$.

Now, let us consider a larger response with $l = 7$, i.e., a shot with 255 packets to see the impact of a small $window$ with a longer time-series. At each RTT delay period, the window size $cwnd$ increases exponentially such as 1, 2, 4, 8 and 16, but the increment stops when $cwnd$ reaches 20. This provides the reason why the fractal scaling stops at $k = 5$ as shown in Fig. 8.

Next, we want to investigate the impact of having a larger *window* = 256. We also increase *capacity* of the link by having

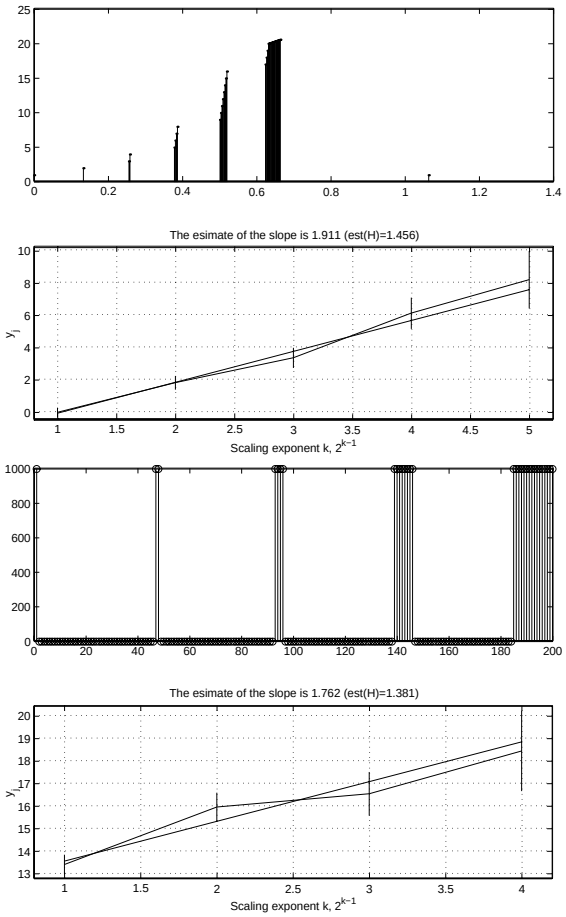

Fig. 7. *cwnd* process, packet series and their wavelet transforms for $D = 60$msec, $l = 4$, $C_{21} = 3$Mbps

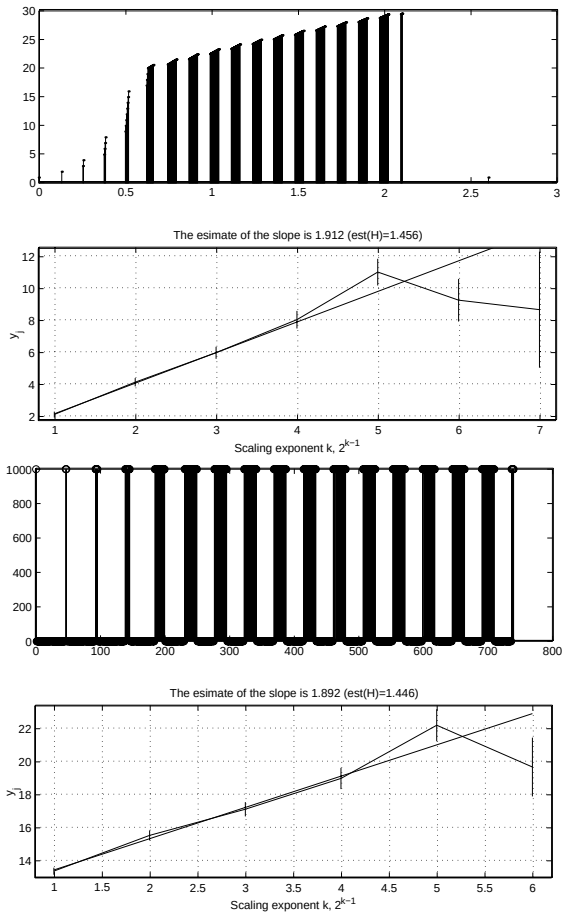

Fig. 8. *cwnd* process, packet series and their wavelet transforms for $D = 60$msec, $l = 7$, $C_{21} = 3$Mbps, $w_1 = 255$ and $window = 20$

$D = 120$msec and $C = 6$Mbps. Thus, $capacity = 180$. A single shot with 511 packets, $l = 8$, is now used. The third figure of Fig. 9 clearly illustrates the operation of packet transports in *slow start* such that at every 180 ticks the sender increases the number of packets with a power of 2. Thus, we shall observe the same scaling behavior with $H = 1.5$ as was clearly indicated by the examples of Z-transform and the exponential-law shot noise. It should be noted that since $window > capacity$ in this case, capacity takes the limiting factor determining the upper most time-scale at which the small scaling law stops, i.e. around the time-scale of $log2(capacity) = 7.5$. We clearly see the extension of fractal scaling region having $H = 1.396$, upto $k = 8$. When we lower capacity by having $C = 1$Mbps and $D = 30$msec, we observe that the scaling law with $H = 1.5$ stops at $k = 2$ (Not presented in figures). After the *slow start* phase, the link is 100 percent utilized serving the packets back to back. Thus, the fractal scaling law disappears for $k > 2$. This is natural since the time-series becomes a constant rate process, taking only one value of 1000 after the time instant at which $cwnd$ exceeds the value of $capacity$.

From the results of single-shot analysis, what we have learned shall be clear such that $min\{window, capacity\}$ determines the upper most time-scale up to which, starting from the time-scale of T_p, the exponential-law scaling behavior with $H = 1.5$ presides. Each of the single-shot traces obtained in this section symbolizes the characteristic kernel $h(t)$ which determines the PSD of a shot-noise. Thus, the characteristics of single-shot spectrum shall translate exactly to the spectrum of multiple-shot simulations which involves with multiple TCP connections and large file transfers.

D. Multiple-shot analysis

Single-shot analysis reveals the scaling law at the small time-scale, from T_p to $capacity$. Multiple-shot analysis shall shed lights on the scaling behavior at the larger time-scale. The characteristic scaling signatures at the small time-scales shall be remained to be the same in multiple-shot analysis as discussed. The filter characteristics of *slow start* is basically a exponential-law shot window. Thus, it will reveal the low-pass filter behavior, as shown in Fig. 2, including the responses of low frequency regions (or the larger time-scales of interest in this section) when observed with a longer trace. Thus, it would not interfere much with the self-similar scaling law at the large time-scales which may be developing with multiples of TCP transfers with different response sizes.

The source traffic is designed to be the same as that of the shot-noise processes. TCP-request arrivals are modeled as Poisson process with rate λ. The size of response w_n will be gen-

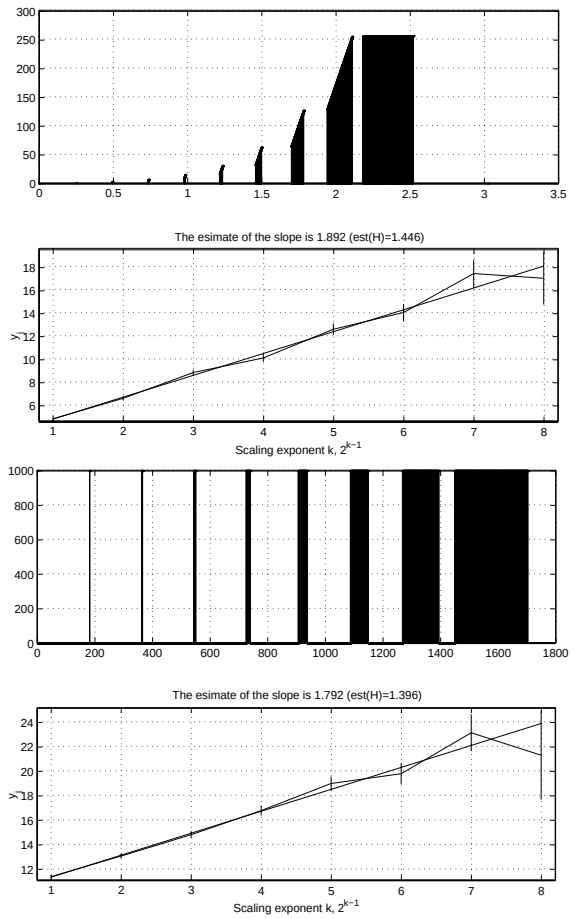

Fig. 9. *cwnd* process, packet series and their wavelet transforms for $D = 120$msec, $l = 8$, $C_{21} = 6$Mbps, $w_1 = 511$ and $window = 256$

erated with the four different distributions, HTTP and FTP with $\alpha = 1.5$, and skewed HTTP and FTP developed in section III. It is of convenient to define the offered loading factor ρ

$$\rho := \frac{\lambda E\{w_1\}}{C_{21}/8}. \tag{29}$$

where λ is the arrival rate of the Poisson process: In simulation, we use the sample mean, $\bar{w}_1$, instead of $E\{w_1\}$.

Thus, we vary the number of TCP flows, the TCP-request arrival rate λ and the distribution of w_n for source traffic generation. In addition, we vary the $capacity$ of the network by changing the link-speed C and the link-delay D. Throughout this section, we fix *window* = 20 for convenience. In the Internet, there may be a variety of different $window$ sizes employed by different versions of TCPs as well as by different users and host machines. However, in this paper, we fix it to be 20 and vary the $capacity$ of the bottleneck link since what we are interested in is the relative ratio between the sizes of $window$ and the $capacity$, but not the precise numbers.

D.1 Small *capacity* Link

We now present the results for small *capacity* simulations where "small" implies the situation when the *capacity* is smaller than *window* the maximum allowed window size, which we fix it to be 20. We chose $D = 30$msec and $C = 2.5$Mbps. Thus, $capacity$ is 18.75 which is smaller than $window = 20$.

First, the skewed FTP response size distribution is generated with the shift $d = 2$. For this, we know that the maximum size w_o of the response is 1,620 in packets with 10% chance of being selected in the skewed FTP. There are total of 100 counts of arrivals, 10 arrivals for each TCP connection. The offered load is 0.3457 with $\lambda = 0.2083$. Since $window > capacity$, the smallest possible completion time of transporting the maximum response is 1,620 T_p. Note that since $log2(1620) = 10.7$, we expect to see a change of scaling law at $k = 11$ or $k = 12$. This is the case of rectangular-law shot noise discussed in section II, which would has f^{-2} spectrum behavior. That is $\gamma = 2$ and $H = 1.5$ from (26). In fact, Fig.10 indicates that the scaling region with $H = 1.5$ extended up to $k = 11$, with a little break at $k = 4$ which is also expected from single-shot analysis.

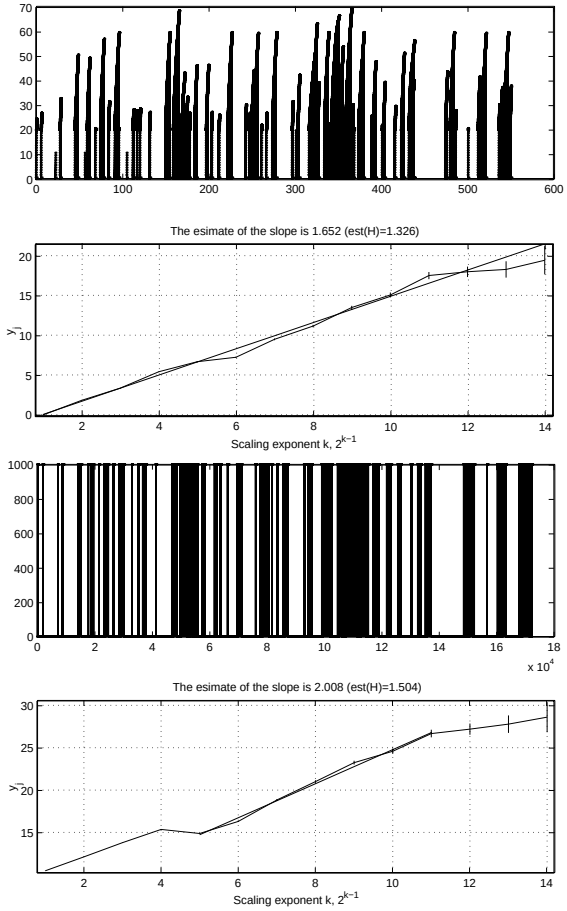

Fig. 10. Small *capacity* simulation, $N = 10$, $D = 30$msec, $C = 2.5$Mbps, skewed FTP-profile with shift=1, 10 arrivals for each connection

The break is at the time-scale of $2^4 = 16$ which is comparable to *capacity*=18.75. For scales with $k > 11$, the scaling curve starts to develop the low-pass filter characteristics. With a longer simulation, it actually becomes one showing a flat response beyond $k > 11$. If we have the shift $d = 1$ so that the maximum response size becomes 162, $2^{7.34}$, the scaling region with $H = 1.5$ ends at $k = 8$, and a flat response region dominates the rest $k > 8$. Both of the simulations, the offered load

were kept at the value of $\rho = 0.3457$. When we change the value of ρ to 0.0346, the same scaling behavior was observed such that up to $k = 11$ the trace show $H = 1.5$. This validates our approach of explanation the scaling law with the rectangular-law shot noise.

When we switch back to the use of pareto-tail with $\alpha = 1.5$, the flat-spectrum region disappears. Instead, yet another scaling region with slope $H = 1.013$ appears from $k = 11$ to $k = 16$. With the use of pareto-tail, a response of huge size is possible with a non-negligible probability. Thus, the fractal scaling region tends to extend to the lower frequency regions, but with a significantly reduced slope such as $H \leq 1$, which we attempt to explain the behavior by comparison with the rectangular-pareto shot noise.

We now switch to the use of HTTP-distribution having the pareto-tail with $\alpha = 1.5$. Since HTTP-distribution starts the pareto-tail at around the response size of 10 packets, we will not observe a significant development of fractal scaling region with $H = 1.5$ beyond $k > 5$ unlike the case with the FTP-distributions. Two simulations were conducted with (1) $\rho = 0.1638$ and (2) 0.9332, and observed were the scaling behaviors at the lower frequency. The simulation results for $\rho = 0.1638$ is in Fig. 11. Note that hereafter, only the wavelet transforms for $cwnd$ and the byte time-series are shown, the top and the bottom respectively. At this relatively low traffic intensity, the flat-frequency region starts to develop at a early time-scale as shown. When we increase $\rho = 0.9332$, we observe a straight

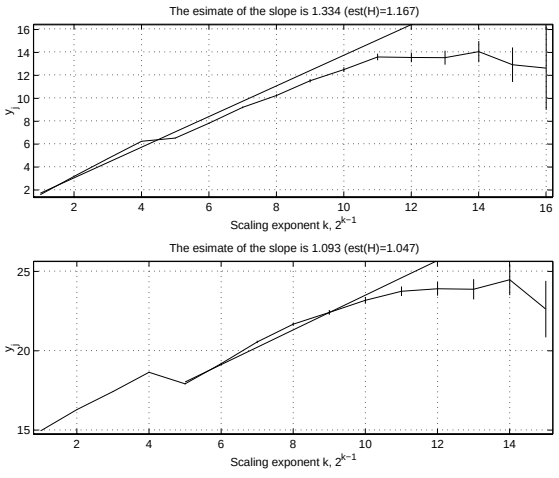

Fig. 11. Small *capacity* simulation, $N = 10$, $D = 30$msec, $C = 2.5$Mbps, HTTP-distr., 100 arrivals for each connection

scaling behavior with $H = 0.997$ extends throughout the entire time-scales (Not shown in figures). This indicates that the contention among TCP senders effectively enlarges the completion time of a response. This will be emphasized by our next experiment. We now use the skewed HTTP-profile. The result is in Fig. 12. It should be noted that it again exhibits a fractal scaling region extending from $k = 5$ to $k = 13$ where the slope estimate of the trace from $k = 5$ to $k = 14$ gives $H = 0.757$. From these experiments, it becomes very clear that the effective completion time, which depends on the network parameters and the TCP parameters, plays the determining factor having a fractal scaling region even without having the heavy-tail distribution in response size.

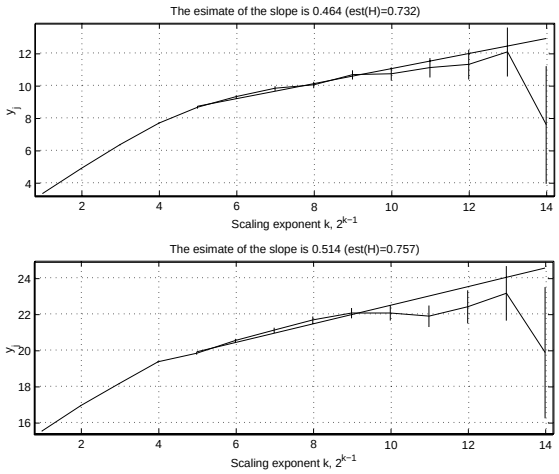

Fig. 12. Small *capacity* simulation, $N = 10$, $D = 30$msec, $C = 2.5$Mbps, Skewed HTTP-distr., 100 arrivals for each connection

D.2 Large *Capacity* Link

We summarize by applying what we have learned so far to longer traces excited by many arrivals of responses. For this, we increase $capacity$ of the link by having $C_{21} = 25$Mbps and $D_{21} = 30$msec. This makes $capacity$=187.5, which is $2^{7.55}$. In this setup, a break is expected at $k = 5$ for $window = 20$ and another one at $k = 8$ for $capacity$. We use HTTP with pareto-tail with $\alpha = 1.5$. Thus, we expect there will be a single scaling region for $k > 8$. We first investigate the trace obtained at a simulation with low traffic intensity, $\rho = 0.0102$. The shape of the resulting spectrum is very similar to Fig.13, with the estimate of H parameter from the packet-trace is 0.709. Fig.13 shows the wavelet spectrum for $\rho = 0.2101$. We note that the estimate of the Hurst parameter is 0.903, which tends to get larger as ρ increases.

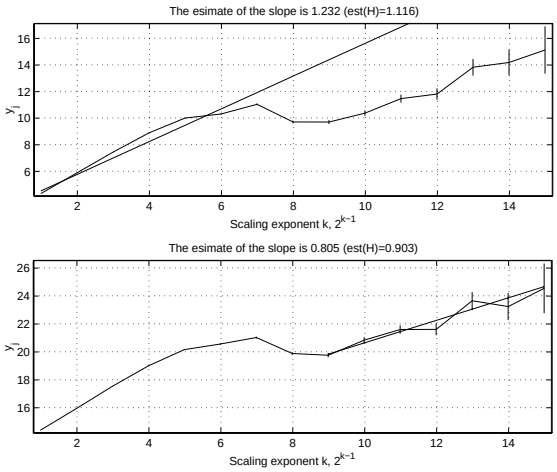

Fig. 13. Large *capacity* simulation, $N = 10$, $D = 30$msec, $C = 25$Mbps, HTTP-distr., 100 arrivals for each connection

Lastly, we increase the number of connections to $N = 10000$, which is about 53 times greater than the *capacity*and 1000 ar-

rivals for each connection. The motivation for this experiments is to observe the impact of contenting TCPs on scaling behavior. We again have two loading conditions, $\rho = 0.1010$ for one and 0.4010 for the other. As the results, we observe no noticeable change in the scaling behavior as N being brought to a large number. We again observed the impact of ρ in the change of fractal scaling slope as was seen in previous experiments.

E. Summary statements

The following statements serve as a summary of what we have learnt in section IV.

- The scaling region due to *slow start* is from T_p to $min\{window, capacity\}$, having the parameter H of 1.5.
- From the time-scale of *window* to *capacity*, there exists a separate, blurred scaling region with a slope close to 1.0.
- For large response sizes such as the use of skewed FTP or FTP, there exists another scaling region with $H = 1.5$, from the time-scale of $capacity$ to that of maximum response size w_o in the skewed distribution (or the starting point of a Pareto-tail in FTP-distribution).
- The pareto-tail region of response size is responsible for the fractal scaling behavior toward the low frequency (or the larger time-scales) having the parameter H generally less than 1.0.
- The reason for f^{2H-1} behavior at the lower frequencies, such as for $k > log_2(capacity)$ in HTTP or for $k > log_2(w_o)$ depends more on the closed-loop flow-control of a TCP, than the source being heavy-tailed.
- The heavy-tailedness of response size is not the necessary condition for trace to exhibit fractal scaling responses.

V. Conclusion

We illustrated with many NS2.0 simulations how the parameters of TCP algorithm such as $cwnd$ or $window$, the different phase of TCP such as *slow start*, and the network parameters such as link-speed and link-delay, are inter-related to make influences on the variety of different self-similar scaling behaviors observed in the network traffic. We showed that it is possible for a single trace to display multiples of different scaling regions with different scaling slopes. The range of each region and the associated scaling slope were shown to be strongly associated with the key parameters described above. For example, we showed that the *slow start* causes a unique scaling behavior, characterized by $H = 1.5$, starting from the time-scale of T_p to that of min$\{window$,$capacity\}$. For time-scales greater than $capacity$, the transport of a large file, regardless of a specific shape of the tail distribution, may cause the completion time of a response to be enlarged, causing a significant extension of the fractal scaling behavior–the $1/f^2$ behavior–toward the lower and lower frequency. This is especially so when the incoming source traffic rate is relatively higher. The operation of workload transfer gets interrupted by the other on-going TCP traffic, and thus the completion time of a response–the traffic ON period–gets enlarged. The larger the original workload size is the higher the chance is of getting more number of interruptions while being transported. Having said this, we can carefully conjecture a newer proposition that the heavy-tailed parameter of the traffic ON-period which actually determines the self-similarity of the traffic can become greater than the original α of the workload size distribution; this may explain why we observe the value of parameter H significantly greater than $\frac{3-\alpha}{2}$ when the link is very busy.

In this paper, we have used a single round trip time setup by having the exact same link speed and delay for all TCP connections. In practice, the distribution of RTTs for different flows are highly likely to have high variation. The small time-scale behavior studied in this paper for a single RTT setup might be extended to explain the more complex phenomenon of multifractal scaling law. That is, the multifractal phenomenon might be explained with a superposition of multiples of unique scaling laws related with each RTT for each connection.

Acknowledgments

The author would like to acknowledge Dr. Bo Ryu at HRL for useful discussions and suggestions of unknown reviewers.

References

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

[2] R.H. Riedi and J. Lévy-Véhel, "Tcp traffic is multifractal: A numerical study," 1997.

[3] R.H. Riedi and W. Willinger, *Toward an Improved Understanding of Network Traffic Dynamics*, vol. Self-similar Network Traffic and Performance Evaluation, Wiley, 1999.

[4] M.S. Taqque, V. Teverovsky, and W. Willinger, "Is network traffic self-similar of multifractal?," *Fractals*, pp. 63–73, 1997.

[5] A. Feldmann, A.C. Gilbert, and W. Willinger, "Data networks as cascades: Investigating the multifractal nature of internet wan traffic," *Proc. of the ACM/SIGCOMM'98*, pp. 25–38, 1998.

[6] A.C. Gilbert, W. Willinger, and A. Feldmann, "Scaling analysis of conservative cascades, with applications to network traffic," *IEEE Trans. on Info. Theory*, vol. 45, no. 3, pp. 971–91, 1999.

[7] K. Park, G. Kim, and M.E. Crovella, "On the relationship between file sizes, transport protocols, and self-similar network traffic," *Proc. of IEEE Int. Conf. on Network Protocols*, pp. 171–80, 1996.

[8] A. Veres and M. Boda, "The chaotic nature of tcp congestion control," *Proc. of IEEE Infocom 2000*, 2000.

[9] A. Feldmann, A.C. Gilbert, P. Huang, and W. Willinger, "Dynamics of ip traffic: A study of the role of variability and the impact of control," *Proc. of the ACM/SIGCOMM'99*, 1999.

[10] Gregory W. Wornell, *Signal Processing with Fractals: A wavelet-based approach*, Prentice Hall, Upper Saddle River, NJ, 1995.

[11] W. Richard Stevens, *TCP/IP Illustrated*, vol. 1, The Protocols, Addison-Wesley, Reading, Massachusetts, 1994.

[12] Crovella et al, "Heavy-tailed probability distributions in the world wide web," *A Practical Guide to Heavy Tails: Statistical Techniques and Applications*, pp. 27–53, 1998.

[13] W. Willinger, V. Paxson, and M.S. Taqque, "Self-similarity and heavy tails: Structural modeling of network traffic," *A Practical Guide to Heavy Tails: Statistical Techniques and Applications*, pp. 27–53, 1998.

[14] Heung-No Lee and Yongmin Choi, "Monitoring the degree of self-similarity in network traffic with concept of sample spectrum," *preprint*, 2000.

[15] V. Paxson, "Fast, approximate synthesis of fractional gaussian noise for generating self-similar network traffic," *Computer Communications Review*, vol. 27, pp. 5–18, 1997.

[16] Benoit B. Mandelbrot, *Multifractals and 1/f noise*, Springer-Verlag, NY, 1998.

[17] P. Abry and D. Veitch, "Wavelet analysis of long-range dependent traffic," *IEEE Trans. Info. Theory*, pp. 2–15, 1998.