

Copyright © 2002 IEEE. Reprinted from
Proceedings of the IEEE, 2002; 90 (5):800-819

This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of the University of Adelaide's products or services. Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by writing to pubs-permissions@ieee.org.

By choosing to view this document, you agree to all provisions of the copyright laws protecting it.

Self-Similar Traffic and Network Dynamics

ASHOK ERRAMILI, MEMBER, IEEE, MATTHEW ROUGHAN, MEMBER, IEEE,
DARRYL VEITCH, MEMBER, IEEE, AND WALTER WILLINGER, MEMBER, IEEE

Invited Paper

One of the most significant findings of traffic measurement studies over the last decade has been the observed self-similarity in packet network traffic. Subsequent research has focused on the origins of this self-similarity, and the network engineering significance of this phenomenon. This paper reviews what is currently known about network traffic self-similarity and its significance. We then consider a matter of current research, namely, the manner in which network dynamics (specifically, the dynamics of transmission control protocol (TCP), the predominant transport protocol used in today's Internet) can affect the observed self-similarity. To this end, we first discuss some of the pitfalls associated with applying traditional performance evaluation techniques to highly-interacting, large-scale networks such as the Internet. We then present one promising approach based on chaotic maps to capture and model the dynamics of TCP-type feedback control in such networks. Not only can appropriately chosen chaotic map models capture a range of realistic source characteristics, but by coupling these to network state equations, one can study the effects of network dynamics on the observed scaling behavior. We consider several aspects of TCP feedback, and illustrate by examples that while TCP-type feedback can modify the self-similar scaling behavior of network traffic, it neither generates it nor eliminates it.

Keywords—Chaotic maps, congestion control, dynamical systems, feedback, long-range dependence, scaling, self-similarity, wavelets.

I. INTRODUCTION

Traffic characterization and modeling are generally viewed as important first steps toward the analysis and control of network performance. At the same time, there is little disagreement that the resulting understanding of and solutions to network performance-related problems are only as good and complete as the underlying assumptions

on the traffic flows in the network, and the manner in which they interact with network elements. However, traffic measurement studies over the past 10 or so years have continued to demonstrate our limited understanding of actual network traffic by revealing “emergent” phenomena—measurement-driven discoveries about the dynamic nature of actual network traffic that come as a complete surprise, defy conventional wisdom, and cannot be explained nor predicted within the framework of the traditionally considered traffic models. An example of such an emergent phenomenon was the discovery of the self-similar or fractal nature of network traffic [18], [42], [54].

The focus of this paper is the observed scaling behavior of network traffic, and the use of dynamical systems models to understand how feedback mechanisms affect it. In particular, we survey here the development of self-similar traffic and performance modeling over the past 10 years, paying special attention to the radical changes that have resulted from a gradual improvement of our understanding of the causes and origins underlying the self-similarity phenomenon [32], [53], [57], [72]–[74], [77]. This work motivates the development of innovative new approaches for modeling and evaluating the performance of large-scale complex communication networks such as today's Internet. As an illustration, we outline a dynamical systems modeling approach that shows some promise in tackling future traffic performance modeling problems, where feedback and the large-scale nature of networks must be addressed for accurate analysis and control.

A. Self-Similarity: Dealing With the Phenomenon

In the case of data networks, high time-resolution packet-level traffic measurements are generally recorded from the physical link over which the data is sent, by copying either an initial part of each packet (i.e., the packet header) or every single bit of each packet (i.e., header plus payload) over to a high-performance storage device. Along with every packet that is stored, additional information is usually saved, notably an accurate time stamp (packet arrival time), packet

Manuscript received July 5, 2001; revised November 20, 2001. This work was supported in part by Ericsson Research.

A. Erramilli is with Qnetworx, Inc., Morganville, NJ 07751 USA (e-mail: ashok@qnetworx.com).

M. Roughan and W. Willinger are with AT&T Laboratories-Research, Florham Park, NJ 07932-0971 USA (e-mail: roughan@research.att.com; walter@research.att.com).

D. Veitch is with EMUlab, University of Melbourne, Victoria 3010, Australia (e-mail: d.veitch@ee.mu.oz.au).

Publisher Item Identifier S 0018-9219(02)05244-1.

size, and other status information. The last decade has seen an enormous increase in empirical studies of high-quality and high-volume data sets of traffic measurements from a variety of different data networks, but especially from different links within the global Internet. These studies typically describe pertinent statistical characteristics of the temporal dynamics of the “packet” or bit rate processes (i.e., the time series representing the number of packets or bits per time unit, over a certain time interval) as seen on a link within the network. They provide ample evidence that measured packet traffic exhibits extended temporal correlations [i.e., *long-range dependence (LRD)*], and hence when viewed within some range of (sufficiently large) time scales, the traffic appears to be *fractal-like* or *self-similar*, in the sense that a segment of the traffic measured at some time scale looks or behaves just like an appropriately scaled version of the traffic measured over a different time scale. In effect, this empirically-based effort toward describing actual data network traffic has demonstrated that self-similarity provides an elegant and compact mathematical framework (see Section II-A) for capturing the essence behind the wide range of observed “burstiness” or scale-invariance in measured traffic traces (see [77] for further references).

The observed self-similar behavior of measured traffic was in sharp contrast to what the conventional models for data traffic predicted, models that in general lacked validation against measured traffic traces. A hallmark of these traditional voice-based data traffic models is a correlation function which decays exponentially fast [i.e., *short-range dependence (SRD)*], implying that time-aggregation quickly results in white noise traffic characterized by the absence of any significant temporal correlations, and capable only of reproducing the observed bursty behavior of measured traffic over a narrow range of time scales. In fact, during the first few years following the discovery of self-similarity, much of the existing research effort focused on purely descriptive models and followed the line of traditional time series analysis, where the context in which the data are measured and collected is often of little or no significance. As surveyed in Section II-B, the main emphasis was on statistical inference, parameter estimation, and model fitting. In this sense, almost the only impact that the self-similarity finding had in the short term was to add self-similar models to an already long list of existing “open loop” data traffic models. Here “open loop” refers to the fact that while the traffic characteristics impact queueing behavior, the impact of queueing performance in shaping the incident traffic is not modeled. Moreover, while tractability was the primary motivation for many of the traditional models, the new self-similar modeling alternatives were the result of a measurement-driven research effort; they also revealed very clearly the restrictive conditions under which traditional data traffic modeling operated when trying to cope with the generally accepted “bursty” nature of data traffic.

As far as the engineering significance of self-similarity is concerned, the standard question that was asked immediately after its discovery was “what are the implications of self-similarity on queueing performance and network

engineering?” Put differently, without fully understanding the causes and origins of the phenomenon at hand, these researchers needed to know the impact that self-similar traffic would have on the analysis and engineering of high-speed networks already under development and deployment, such as frame relay, ATM and Internet access/backbone networks [53]. To answer this FAQ, a good starting point in understanding the impact of self-similarity was provided by Norros [50], who developed a formula that can be used to estimate buffer overflow probabilities at network switches and routers. The Norros results showed that the queueing backlogs were in general worse with self-similar traffic, in the sense that the buffer sizes to achieve a certain loss objective could be significantly greater. This agrees with the common intuition that the presence of positive correlations in the incident traffic can only aggravate queueing delays. The long-range correlations manifest themselves in extended periods of time over which the incident traffic exceeds link capacity, leading to heavy queueing backlogs. In contrast, a small amount of buffering is sufficient to smooth out the peaks and valleys in SRD traffic. In another engineering sense, the presence of self-similarity could have been termed “good news.” The large deviations principles that underpins the Norros’ formula, and its refinements [22], [47], [49], directly relate the traffic characteristics (e.g., distribution of arrival counts) to performance measures (e.g., queue length distributions, loss rates). They indicate that to obtain the performance estimates necessary to accomplish even basic network engineering tasks, one must in principle characterize an infinite family of distributions of the arrival counts. This is clearly impossible to do in practice. However, this is precisely where the observation of self-similarity in network traffic is of practical relevance—it buys parsimony; the ability to specify the relevant statistical details of arrival counts with as few parameters as possible. In theory (i.e., assuming an idealized Gaussian setting), it enables the representation of the infinite family of distributions by three parameters over the entire scaling region—an enormous reduction in the description complexity. These three parameters are the mean and the variability of the traffic process, and the self-similarity or Hurst parameter (see Section II-C for details).

Self-similar traffic descriptions that capture traffic fluctuations over many time scales of interest are also attractive in view of the large deviations notion of a “critical time scale,” which dominates queueing behavior. If one knew *a priori* what the critical time scale was, there would be no need to model the traffic on other time scales. This has led some to argue that because queueing phenomena are dominated by essentially a single time scale, self-similarity is irrelevant since it is used to describe traffic over a wide range of time scales. In practice, traffic fluctuations over a wide range of time scales are important in engineering—ranging from milliseconds, through seconds, minutes, and beyond. Even in determining queueing performance, buffer sizes in modern packet networks correspond to several round-trip times, or several hundred milliseconds or more. Secondly, the buffer levels and capacity of the link can vary widely

in an end-to-end flow (say, from modem speeds of 9.6 kb/s in wireless access through OC-48 or 2.5 Gb/s in the Internet backbone). Third, the critical time scale varies with link utilization. Thus, basing a traffic description on an *a priori* expectation of critical time scales limits the model's utility, requiring a specialized set of models for each potential combination of queuing parameters. In such cases, a self-similar traffic description is clearly a more parsimonious and also more viable alternative.

The above discussion does not imply that self-similar models are always appropriate. As described in [25], there are three fundamental conditions that must be satisfied for a Gaussian self-similar traffic description, corresponding to the observation of long-range dependence in traffic, to apply:

- 1) the network traffic should be sufficiently aggregated so that the marginal distributions of counts are at least approximately Gaussian;
- 2) the long-range dependent scaling region should span the engineering time scales of interest;
- 3) the impact of network controls on the traffic flows must not be significant over the engineering time scales of interest.

These three conditions together suggest a feasibility regime for the standard self-similar traffic model discussed in Section II-A, based on *fractional Brownian motion (fBm)*. It is delineated by: moderate to heavy traffic (so that the aggregation levels are sufficient for a second-order description to be valid), aggregation from a large number of low-activity sources (so that no one source is dominant), and moderate to large buffer sizes (so that the scaling region covers the time scales of interest). While this may seem like a restrictive set of conditions, it nevertheless holds for a range of practical network engineering problems.

B. Self-Similarity: Understanding the Phenomenon

In parts of the Internet research community, the self-similarity discovery was met from the very beginning with a degree of curiosity that went beyond the obvious question of how to model a traffic trace found to be consistent with LRD. In short, these researchers wanted to know if there exists a physical explanation for the observed self-similar nature of data traffic—an explanation that makes sense in the networking context and which can be phrased and, more importantly, validated in terms of more fundamental traffic-related entities. To move beyond the conventional time series analysis perspective and to succeed in explaining the physics behind self-similar traffic, it is essential to exploit more purposefully the high semantic context contained in the measured data, thereby explicitly accounting for certain aspects behind the design and architectural principles of today's data networks [16]. In fact, realizing that it is difficult to think of any other area in the sciences where the available data provides such detailed information about so many different facets of the phenomenon under study suggests that there exists great potential for discovering intuitively appealing, conceptually simple and mathematically rigorous statements

as to the causes and origins of all sorts of “emergent” phenomena in data networking, including self-similarity.

Over the last few years, the challenge to develop networking-related explanations for self-similarity has resulted in a number of studies that convincingly demonstrate that the empirically observed scaling behavior of measured network traffic over large time scales is caused by the high-variability of the individual sessions (or transmission control protocol (TCP) connections—see Section II-E—or IP flows) that make up the aggregate traffic. More precisely, these studies show that aggregate packet-level network traffic exhibits self-similar scaling behavior over sufficiently large time scales if the durations (in time) or sizes (in bytes) of the individual sessions, or TCP connections, or IP flows that generate the aggregate traffic have a *heavy-tailed distribution with infinite variance* (that is, range from extremely short/small to extremely long/large with nonnegligible probability). The resulting *structural* modeling approaches are discussed in Section II-D and provide a physically-based, networking-related understanding of the observed self-similarity phenomenon in terms of more fundamental properties of the traffic patterns generated by the individual users and/or applications. They have impacted our understanding of actual network traffic to the point where we can explain why it exhibits self-similar scaling properties over large enough time scales. The technical arguments are in full agreement with the networking researchers' intuition, can be explained readily to nonnetworking experts and/or nonmathematicians, and can be validated against measurements at the different layers in the networking hierarchy where the more fundamental quantities can be inferred or observed. These developments have helped immensely in demystifying self-similar traffic modeling and have given rise to new insights and physical understanding of the effects of LRD on the design, management and performance of modern data networks [18], [29], [73], [76], [83].

On the one hand, the fact that we can explain self-similar scaling in terms of the statistical properties of the individual sessions or connections that make up the aggregate link traffic suggests that the LRD nature of network traffic is mainly caused by user/application characteristics such as Poisson arrivals of sessions and heavy-tailed distributions for the session durations/sizes, and has little to do with the network, i.e., with the protocol-specific mechanisms that determine the actual flow of packets as they traverse the Internet (as long as the available bandwidth is required to be shared in a fair manner among the competing connections). Assuming that the nature of the information to be transported by networks does not itself change drastically, self-similar scaling at large scales is, therefore, likely to remain with us. On the other hand, realizing that LRD leaves the small-time scale behavior (or, equivalently, the small-lag correlation structure) essentially unspecified, has recently motivated researchers to renew their investigations into the small time scale structure of network traffic, in an attempt to relate the observed complex and highly time-localized traffic patterns to the most important features of the most commonly encountered protocols, and to understand

their mutual interactions in the large-scale Internet. Thus, while for the purpose of explaining the physical causes behind the observed LRD behavior over large time scales it was necessary to have only a rough understanding of the protocol architecture of today’s data networks, relevant studies of the fine-time scale structure require a more detailed knowledge of the dominant protocols and/or their most pronounced features and rules. To this end, we focus in Section II-E on the most commonly used transport layer protocol, TCP, and some of its critical functionalities.

With the desire for a more complete description of measured network traffic that is also capable of accounting for the possibility of interesting phenomena associated with small time scales comes the need for a mathematical framework that extends beyond fractional Gaussian noise and is especially geared toward a flexible and effective handling of highly irregular (or regular) behavior that may be well-localized in time. This leads us to the study of differentiability, fractality and regularity of stochastic processes; that is, properties that are inherently related to the high-frequency behavior or small-lag correlations. These have little to do a priori with low-frequency components and/or high-lag correlations, except that the latter dominate as soon as smoothing or aggregating in time has wiped out any interesting small-time peculiarities, leaving us with LRD or its close relative, asymptotic self-similarity, as predominant and universal characteristics. However, this desire for more complete traffic models is in serious conflict with the realization that because of its *feedback* or “closed-loop” nature, TCP (or any other form of end-to-end congestion control mechanism) constantly shapes (with some delay) the packet flow emitted from the different sources, which in turn alters the rate process that arrives at the queue or IP router for buffering, which in turn impacts the level of congestion, and so on [39]. The key point is that TCP traffic is *shaped* by current networking conditions, resulting in TCP traffic in turn *dynamically adapting* to changing network conditions [29], [83].

The above observation, and the fact that many of the measured traffic traces which show strong scaling behavior are derived from networks in which TCP is dominant, has led researchers to a conflicting set of conclusions. On the one hand there are statements to the effect that the impact of network feedback is to eliminate self-similarity, at least to the extent that some of the dire predictions of the Norros model are not observed in detailed simulations of TCP behavior [7]. Note however that the conclusions made in [7] apply more generally to all open loop models, self-similar or not. On the other hand, there are other studies which claim that TCP mechanisms can create and propagate the self-similarity in network traffic [69], [70] (for an earlier discussion concerning protocol mechanisms and self-similarity, see [55]). We discuss the existing confusion over this issue and the resulting dilemma in more detail in Section II-F. Clearly, the diversity of opinions on the impacts of network dynamics on scaling behavior in traffic motivates an in-depth investigation of closed-loop models, including our approach (see Sections III and IV) based on dynamical systems.

C. Self-Similarity: A Dynamical Systems Approach

The structural modeling approach discussed thus far relates source behavior at the application level to observed traffic characteristics at the network layer. In order to extend this approach to scenarios in which network feedback significantly shapes source behavior, we need a more versatile approach to source modeling. The work of [23], [24], [56] demonstrated one way in which models based on chaotic maps can be used to model traffic sources. The basic concept is to associate the state of a chaotic map to traffic source activity. By carefully choosing the underlying maps, this work established that it is possible to model a full range of “open loop” data traffic source behavior.

Beyond mimicking conventional open loop sources, we show in this paper that the chaotic map approach can be extended to explicitly incorporate network feedback in its evolution, thereby capturing source behavior shaped by such feedback. The dynamical systems approach readily lends itself to such interactions, achieved through coupling the equations which describe source and network evolution. We can then relate specific aspects of TCP behavior to changes in observed traffic characteristics. This approach is detailed in Sections III and IV. We currently simulate the resulting sets of equations to evaluate the impacts of network feedback on the observed scaling behavior, though there is considerable scope for deriving analytical results that can yield additional insights.

II. SELF-SIMILAR SCALING AND NETWORK TRAFFIC

A. The Mathematics of Large-Time Scaling

To describe the character of the observed scaling properties of traffic data more precisely, we introduce a second-order stationary time series $X(t)$. We can think of the series as the number of bytes arriving in time interval $t = 1, 2, 3 \dots$, where a typical time interval could be 1 ms.

Let the mean, variance and covariance function of this series be denoted by $\mu = E[X(t)]$, $v = \text{Var}[X(t)]$, and $c_X(k) = E[(X(t+k) - \mu)(X(t) - \mu)^2]$, respectively. The question of the nature of fractal traffic is also that of the true nature of *burstiness*, of variability, which at a fundamental level could be due to variations in value, that is in $X(t)$, or to dependence of values over time (or to both). Here we consider that the former does not play the central role, indeed for simplicity we will allow X to be a Gaussian process. The onus is, therefore, on $c_X(k)$ to explain the scale invariance of data via temporal correlations.

A simple example of a time series X with a characteristic time-scale, which dominates its statistical behavior, is that of an autoregressive process of order one, where $c_X(k) = \phi^k$, $\phi \in (-1, 1)$. The memory of this process decays exponentially fast. In contrast, a *Long-Range Dependent* (LRD) series can be defined via a slow, power-law decay of $c_X(k)$:

$$c_X(k) \sim c_\gamma k^{-\beta}, \quad 0 < \beta < 1 \quad (1)$$

where c_γ is a finite positive constant, and the symbol \sim means “behaves asymptotically as,” or more precisely, that

the ratio of the two sides tends to one in the limit of large k . Once in the asymptotic regime, the reigning power-law precludes the existence of any characteristic “time constant.” To see this more clearly, consider the variance of the sample mean estimator \bar{X} based on n values of X

$$v^{(n)} = \text{Var}[\bar{X}] \sim \frac{2c_\gamma}{(1-\beta)(2-\beta)} n^{-\beta} \quad (2)$$

from which two facts emerge. First, the decay in variance is radically slower than the $1/n$ scaling one would expect from familiar *Short-Range Dependent* (SRD) processes which are characterized by covariances that decay exponentially fast. We moreover observe that the prefactor c_γ plays the important role of the *size* of the LRD effect, whereas the *character* of the LRD effect is set by the *scaling exponent* β . Indeed, confidence intervals on mean estimates will be roughly proportional to $\sqrt{c_\gamma}$. Second, for large n the property $v^{(m)}/v^{(n)} \sim (m/n)^\beta$ holds: only the ratio of the scales m and n matters, the decay is controlled by β and does not depend on any particular scale! Finally, the variance $v^{(n)}$ has another life, as the variance of the integrated form of X : $Z(n) = \sum_{t=1}^n X(t)$. This process has stationary increments, and its property $v^{(m)}/v^{(n)} \sim (n/m)^\beta$, if it held exactly, would be exactly that of a *self-similar process in the strict sense* [46], [61]. Such processes have the greatest possible degree of statistical self-similarity, making them ideal as canonical models. They are defined by the requirement that any random vector of $Z(i)$ s at different times has a joint distribution which is identical to that of a rescaled and normalized version of the random vector. For the one-dimensional distributions, this is simply

$$Z(t) \stackrel{d}{\sim} a^{-H} Z(at) \quad (3)$$

which—assuming zero mean—immediately implies that $\text{Var}[Z(t)] = at^{2H}$, where $H > 0$ denotes the *self-similarity parameter* or the *Hurst parameter*. This reveals that the LRD parameter β and the Hurst parameter of a self-similar process are related as $H = (1 - \beta)/2$. The unique continuous Gaussian process with this property and with stationary increments is the well-known *fractional Brownian motion* (fBm) [44]. The (standard) fBm family of processes, $B_H(t)$, is indexed by a single parameter $H \in [0, 1]$ which controls all of its properties. Larger values of H correspond to stronger large scale correlations, which make the sample paths appear more and more smooth.¹ In contrast, $B_{1/2}$ is the memoryless Brownian Motion, which in discrete time is just a random walk. The increments of fBm with $H \in [0, 1]$ define the well known *fractional Gaussian Noise* (fGn) family of stationary processes [44]. For Brownian Motion these increments are i.i.d., which is clearly SRD. fGn processes with $H \in [0, 1/2)$ are also SRD, though qualitatively different to white noise, whereas fGn processes with $H \in (1/2, 1]$ are LRD.

¹Of course, this appearance can be modified by allowing for two additional parameters, namely mean and variance, and allows capturing a wide range of “smoothness” scenarios, from “highly bursty” to “very smooth.”

For analytical studies, the canonical process for modeling LRD traffic rate processes is fGn with $H \in (1/2, 1]$ (fGn with $H \in [0, 1/2)$ has not as yet played a major role in traffic modeling). When describing the observed self-similar behavior of measured traffic rate we can be more flexible. The standard model is a *self-similar process in the asymptotic sense* or, equivalently, an LRD process.²

B. Detection and Measurement of Large-Time Scaling

We have just seen that the statistics of \bar{X} are nonstandard in the presence of LRD, and this is true very generally. A different but related issue, which is key to the rigorous detection and measurement of scaling phenomena, applies to the estimation of the scaling exponent itself. Almost any statistic from an LRD process will involve the exponent, and could, therefore, be used as the basis of an estimation method. For example from equation (2) one could attempt to extract β by measuring the slope of $\log v^{(n)}$ against $\log n$. This “variance-time plot” method is in fact well known, however, such a simple time domain estimator has, in general, poor properties, precisely because of the high correlations between samples. Notably, it is biased and has poor robustness with respect to nonstationarities. This latter point is very important. For some time after the discovery of scaling in traffic, it was debated as to whether the data was indeed consistent with self-similar scaling, or that the finding was merely an artifact of poor estimators in the face of data polluted with nonstationarities. The introduction of wavelet based estimation techniques to traffic in [1] helped greatly to resolve this question, as they convert temporal LRD to SRD in the domain of the wavelet coefficients, and simultaneously eliminate or reduce certain kinds of nonstationarities. It is now widely accepted that scaling in traffic is real, and wavelet methods have become the method of choice for detailed traffic analysis (see e.g., [28], [29], [68], [83]). A survey of estimation methods for the scaling exponent can be found in [65], [84], and comparisons of the wavelet estimator against others is given in [67] (see also [3]), where a joint wavelet estimator is given for (β, c_γ) with excellent statistical properties. Another key advantage of the wavelet approach is its $O(n)$ computational complexity (memory complexity is also $O(\log(n))$ in an on-line implementation [59]), which is invaluable for analyzing the enormous data sets of network-related measurements.³ However, even wavelet methods have their problems when applied to certain real-life or simulated traffic traces. An important rule-of-thumb continues to be to use as many different methods as possible for checking and validating whether or not the data at hand is consistent with any sort of hypothesized scaling behavior, including self-similarity.

As we use the wavelet approach extensively in our analyzes below, we provide a brief practical introduction to the method. Wavelet analysis is a joint *time-scale* analysis. It replaces $X(t)$ by a set of coefficients, the *details* $d_X(j, k)$, j ,

²In this paper—unless stated otherwise—the term “self-similar” always refers to “self-similar in the asymptotic sense.”

³Corresponding Matlab code is publicly available from <http://www.emulab.ee.mu.oz.au/~darryl>.

$k \in \mathcal{Z}$, where 2^j denotes the scale, and $2^j k$ the instant, about which the analysis is performed. In the wavelet domain, equation (2) is replaced by $\text{Var}[d_X(j, k)] = c_f C 2^{(1-\beta)j}$, where the role of m is played by *scale*, of which j is the logarithm, where c_f is the frequency domain analog of c_γ , and C is independent of j . The analog of the variance-time diagram, that is the estimates of $\log(\text{Var}[d_X(j, \cdot)])$ against j , we call the *Logscale Diagram*. It constitutes an estimate of the spectrum of the process in log–log coordinates, where the low frequencies correspond to large scales, appearing in the right in the figures below. The global behavior of data as a function of scale can be efficiently examined in the Logscale Diagram, and the exponent estimated by a weighted linear regression over the scales where the graph follows a straight line. What constitutes a straight line can be judged using the confidence intervals at each scale which can be calculated or estimated. The important fact is that the estimation is heavily weighted toward the smaller scales, where there is much more data. For further details on the method, its use and robustness properties, we refer the reader to [67], [2].

C. Impact of Large-Time Scaling on Performance

Performance studies of networks, whether they be analytic or simulation based, as well as the interpretation of traffic measurements themselves, require traffic models. Looking back over the history of such models, two main drivers stand out: mathematical tractability, and intuition based on telephony network roots. Traffic processes were typically chosen to be Markov processes, sometimes of high complexity, capable of incorporating a hierarchy of network protocol and traffic source effects, but ultimately impacting a finite (usually small) number of time scales. Examples include the Poisson process, renewal processes (with finite variance inter-arrival times), Markov chains modulating the state transitions of alternating renewal processes, autoregressive processes of order N , etc. The realization that over large scales packet traffic was scaling lay each of these drivers to rest for many performance problems. It was necessary to select new models which incorporated fractal properties, reflecting other essential properties of real data, while retaining parsimony. This was and remains a challenge, because the need for tractability is still very much alive. Presenting a black box statistical model for data is one thing, however resolving a useful performance question using it is quite another! We briefly describe here the main model which satisfies the above criteria and has proven useful in the quest to understand some of the fundamental implications of fractal traffic. We describe its main performance implications when used as an input model for simple queueing systems.

The model was originally introduced by Norros [50] and is based on the fBm process. Begin with the stationary traffic rate process $X(t)$, then sum it to form the (nonstationary) process measuring the total number $Y(n) = \sum_{t=1}^n X(t)$ of bytes arriving in the interval $[0, n]$. This can be decomposed according to the average rate μ , a factor σ_Y controlling the

size of fluctuations, and a process $W(t)$ describing their “shape,” as

$$Y(t) = \mu_Y t + \sigma_Y W(t) \quad (4)$$

where $W(t)$ has zero mean. If X is LRD, and forgiving a switch to continuous time, a natural candidate to model $W(t)$ is the fBm B_H defined above, which possesses perfect scaling at all scales but in particular has increments with LRD. Physically, such a model makes sense if conditions (1)–(3) in Section I-A are satisfied. Following Norros, the “fractional Brownian storage” consists of an infinite reservoir, with a constant maximum output rate C , fed by $Y(t)$ above, seen as a highly irregular “fluid” input. The quantity of interest is the marginal distribution $Q(x)$ of the (stationary) queueing process, and the principal result is that it is asymptotically of Weibull type, that is

$$\log Q(x) \sim \kappa x^{2(1-H)} \quad (5)$$

where κ is a slowly varying prefactor [49]. The slow “stretched exponential” decrease with x implies a far higher loss probability than is the case in the traditional Markovian modeling context where exponentially fast decaying tails are the norm.

In practice we have to deal with queues of finite length, which clearly leads, via loss, to bounded queueing distributions whose moments not only exist, but are bounded. However, the effects of LRD are not so easily avoided. In order to improve loss rates, say by a given ratio, buffer sizes must be increased significantly. This is in contrast to the simple Markovian case where—once buffers are larger than a size corresponding to the characteristic scale of the input—losses drop off exponentially. This “buffer insensitivity” effect is present even in the relatively mild LRD queueing of equation (5). Far heavier tails in the queueing distribution are possible, as we explore further in Section II-D. In short, the effect of LRD on performance can be summarized as being mildly negative [33], [53] through to very severe, depending on the circumstances. Often little can be done about the impacts, as they are intrinsic to the sources themselves. The network can try to control traffic flow and smooth it out, but this in turn introduces, in general, other performance penalties. A key lesson learned is that multiplexing of traffic is highly desirable, since the variance reducing effects of adding *independent* flows will reduce the prefactor σ_Y , in turn reducing the impact of the temporal burstiness. Another lesson is that a *power-law [in this case, it concerns the shape of the covariance function in (1)] is highly contagious and hard to eradicate*, which suggests that learning to live with and manage scaling effects is a skill network engineers need to acquire.

D. Causes and Origins of Large-Time Scaling

The models described above (including fGn or other self-similar processes) are “black boxes” in the sense that they ignore nearly all of the gathered traffic information (contained in the header of each recorded IP packet) and, therefore, can offer only a limited contribution to an improved

understanding of data networks and data network traffic. They allow us to talk and think in concrete terms about what some of the measured phenomena are “like,” to formulate reliable measurement methodologies, and to investigate their *a priori* impacts and implications on network performance. However, black box models remain unsatisfying, especially in the networking context, where the highly engineered aspects of the underlying system should be reflected in a proposed model of traffic to provide direct insight into the causes and origins of the emergent phenomena of interest, and their implications for traffic engineering and network management.

To illustrate, consider the following mathematical construction that can be seen at once to reflect the layering architecture of the Internet. At the application layer, *sessions* (i.e., FTP, HTTP, TELNET) arrive at random (i.e., according to some stochastic process) on the link and have a “lifetime” or session length during which they exchange information. This information exchange manifests itself for example at the IP layer, where from the start until the end of a session, IP packets are transmitted in some bursty fashion. Thus, at the IP layer, the aggregate link traffic measured over some time period (e.g., 1 h) is made up of the contributions of all the sessions that—during the period of interest—actively transmitted packets. Mathematically, this construction, also known as *immigration death process* or *M/G/∞ queueing model* and originally due to Cox [17], gives rise to LRD or, equivalently, asymptotic second-order self-similarity, provided the session arrivals follow a Poisson process and, more importantly, the distribution $F(x)$ of their sizes (or lifetimes) T are *heavy-tailed with infinite variance*. Here, we call a distribution F (or the underlying random variable T) *heavy-tailed* with index $\alpha > 0$ if, as $x \rightarrow \infty$,

$$1 - F(x) = P[T > x] \sim c_\alpha x^{-\alpha}.$$

Note that for $1 < \alpha < 2$, T has a finite mean but exhibits *infinite variance* or “high variability.” For $\alpha \leq 1$, T does not even have a finite mean. Thus, the main ingredient of Cox’s construction is that the session sizes (or durations) are heavy-tailed with $1 < \alpha < 2$. Intuitively, this property implies that there is no “typical” session size but instead, the session sizes are highly variable (i.e., exhibit infinite variance) and fluctuate over a wide range of scales, from Kilo-bytes to Megabytes and Gigabytes and beyond. It is this basic characteristic at the application layer that causes the aggregate traffic at the IP layer to exhibit self-similar scaling. A closely related earlier construction, originally due to Mandelbrot [45] (see also [43], [76]), relies on the notion of an *on–off process* (or, more generally, a *renewal-reward process*), but uses the same basic ingredient of heavy-tailedness to explain the self-similarity property of the aggregate link traffic. An *on–off source* typically refers to the mutually independent, alternating on-periods (during which packets are emitted at a constant rate h), and off-periods (during which no packets are sent). The on-times are samples from a random variable B with $\mathbb{E}[B] = 1/\mu$, and the off-times are distributed as A with $\mathbb{E}[A] = 1/\nu$, resulting in an average rate for the

on–off source of $\lambda = h\nu/(\nu + \mu)$. Similar to the Cox construction, LRD can be induced in the rate process $X(t)$ of such an on–off process in a very simple, direct, and computationally effective manner by simply requiring the on- and/or off-periods to be heavy-tailed with infinite variance. Hence, aggregating many such on–off sources results in aggregate link traffic that exhibits self-similar scaling behavior and can be shown to converge to fGn. For details about Mandelbrot’s construction, its relationship with Cox’s construction, and proofs and generalizations of the basic results we refer to [74].

The beauty of structural models such as Cox’s or Mandelbrot’s construction is that in stark contrast to the conventional black box models mentioned earlier, they not only explain the self-similarity phenomenon in simple terms (i.e., heavy-tailed connections), but they also clearly identify the data sets that need to be extracted from the available IP packet-header traces to validate the proposed explanation; that is, to “close the loop” between the discovery of the self-similar scaling behavior of aggregate link traffic on the one hand, and its explanation in terms of infinite variance phenomena at the application or connection level on the other. For example, because of the way many applications are structured, determining session-related entities such as arrival times and sizes or durations from packet-level measurements is straightforward. For FTP and TELNET, these entities have been shown to be consistent with Cox’s construction; see [54]. For HTTP (i.e., Web sessions), obtaining session information is generally more involved, but the empirical evidence for the heavy-tailed characteristic of Web-related entities (e.g., HTTP request sizes and durations) has been well-established to date; see for example [18], [27], [75]. Typically, these empirical studies rely on more or less sophisticated methods for inferring heavy-tailed behavior, including simple log–log plots of $1 - F(x)$ versus x (the so-called *complementary cumulative distribution function* or CCDF plots,⁴ where the objective is to identify a linear region in the tail and use its slope as an estimate of the heavy-tail index α), or *Hill* plots [57]. It is worthwhile noting that as far as the self-similarity phenomenon is concerned, the explanation in terms of high variability remains to date the only approach that has been able to “close the loop.” In particular, as demonstrated in [71] and [30], alternative explanations that rely either on phase transition arguments from statistical mechanics [64], or on network protocols such as TCP [70], fall well short of satisfying this crucial “closing the loop” requirement.

The ability to explain the observed self-similarity phenomenon of aggregate traffic in terms of the high variability characteristics of the individual components that make it up has far-reaching and at times unexpected implications. For one, it suggests that the self-similarity is mainly caused by user/application characteristics (i.e., Poisson arrivals of sessions, highly variable session sizes or durations) and is hence likely to remain a feature of network traffic for some time

⁴When estimating the heavy tail index α , we do not recommend relying on the commonly-used log–log plots of the probability density function $f(x)$ versus x , as they can be easily misinterpreted and often result in highly unreliable estimates.

to come—assuming the way humans tend to organize information will not change drastically in the future [81]. This observation has resulted in a substantial body of literature on performance modeling with heavy-tailed input models, the main driver being the unexpected finding that on–off-type sources with heavy-tailed on-periods ($1 < \alpha < 2$) can lead to an infinite average queue content even though the average input rate λ is less than the capacity C of the queue [15]. Essentially, the infinite variance of the on-times allows very long periods where the input rate exceeds C , leading to correspondingly large buffer levels with appreciable probability. Many variants of these on–off models and related fluid inputs (e.g., superpositions of many on–off sources, mixtures of infinite and finite variance on-times) have been studied, with similar results [14], [53]. Processor sharing scheduling disciplines can mitigate somewhat the service degradations seen from the individual users’ point of view, as it means that if they encounter a very large queue they nonetheless begin to receive service immediately [11]–[13], [82]. However, from a system or network operator’s point of view, simply changing scheduling disciplines cannot make congestion disappear, and we are again reminded of the lesson learned earlier that “a power-law is highly contagious and hard to eradicate” and that learning to live with and manage scaling effects is a skill network engineers need to acquire. Examples that illustrate how the ubiquity of traffic-related high-variability phenomena can be exploited and has led to the design, analysis, and implementation of new algorithms for practical engineering problems include 1) CPU load balancing in networks of workstations [35]; 2) connection scheduling in Web servers [20]; 3) load-sensitive routing in IP networks [63]; and 4) detecting certain kinds of network attacks [79], [80].

Realizing that self-similarity concerns large time scales and leaves the small scale behavior essentially unspecified has recently motivated researchers to renew their investigations into the small time scale structure of network traffic. This current effort and the paradigm shift that it implies for future traffic and performance modeling will be the focus of the rest of this section. In addition, the modeling approaches we have described so far are of the “open loop” variety, where a source’s characteristics are specified, and the performance of the network is calculated. What is missing from this, very traditional approach, is that in the actual network, there exist protocols which allow the source and the network to interact, and indeed this is very common. In “closed loop” approaches, the source is not viewed as an independent agent but reacts to network conditions, and of course the opposite direction is true. However, such interactions operate on time scales of limited range. One, therefore, expects, and this is borne out below in Section IV, that beyond these interaction or feedback time scales, the LRD exponents will continue to make themselves felt. The same argument does not hold true at small scales, as we explore after first providing some background on the leading candidate of network feedback (or, equivalently, of creating interesting small-time scale behavior), the TCP protocol.

E. A TCP Primer

The Transmission Control Protocol (TCP) has for some time carried the bulk of data in the Internet; for example, currently all Web, FTP (file transfer), and NNTP (network news) services, or some 70%–90% of all traffic, use TCP. For this reason, it is a natural choice of protocol on which to focus attention. There are a number of versions of TCP (Tahoe, Reno, Vegas, New Reno) but the predominant one by far is TCP Reno, which we focus on here; for more TCP-specific details, see, e.g., [5], [36]–[38], [78].

TCP aims to provide a reliable delivery service. To ensure this, each packet has a sequence number, and its successful receipt must be signaled by a returning *acknowledgment* (ACK) packet. The second aim of TCP is to provide efficient delivery. An adaptive window flow control is employed where only a single *window* of data is allowed to be transmitted at any one time.⁵ Once one window’s worth of data has been sent, the sender must wait until at least part of it has been acknowledged before sending more. This method is powerful as it is controlled by the sender, requiring no additional information from either the receiver or the network. TCP has two control windows, the sender based *cwnd*, and the (advertised) receiver window *rwnd*. Sources use the minimum of the two. Here we assume that *rwnd* is constant, playing the role of the maximum value of *cwnd*, the dynamical variable of interest. We assume that all packets are of *Maximum Segment Size* (MSS).

The data send rate of a source with window size w , and packet+acknowledgment Round-Trip Time of RTT, is

$$\text{send rate} = \frac{w}{\text{RTT}}. \quad (6)$$

The interesting design issue in the flow control lies in how to choose *cwnd*. Ideally, the window is exactly tuned to the available bandwidth in the network. If it is too small, then the network is used inefficiently, while if too large, congestion may result. However, in the Internet bandwidths can vary from kb/s to Gb/s, while RTTs can range from less than 1 millisecond to 1 second, allowing a variation in send rate of 10 orders of magnitude [32]. The TCP flow control attempts to adaptively choose the window size using four algorithms, *Slow start*, *Congestion avoidance*, *Fast Recovery*, and *Fast Retransmission*. To illustrate TCP’s potential for generating interesting packet arrival patterns over small time scales, we briefly discuss these four algorithms, concentrating on the evolution of the window during the slow start phase which plays a key role in Section IV.

Slow Start and Congestion Avoidance: Slow start begins by setting *cwnd* to an initial value. We use the common value of 1 packet.⁶ It then sends the first window of data, and increases *cwnd* by 1 MSS for each ACK received. As illustrated in Fig. 1, this results in a doubling of *cwnd* every RTT—effectively an exponential increase, despite the name “slow start.” The increase is fast to allow TCP to quickly adapt

⁵TCP data flow is defined in bytes, however some implementations, and many models, use packets. We use both, depending on context.

⁶Requests For Comment RFC2581 [5], the *de facto* standard, sets a limit of $2 \times \text{MSS}$ bytes.

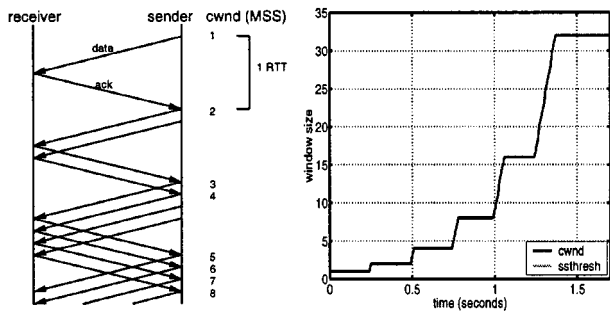


Fig. 1. Conceptual diagrams of slow start behavior. Left: initial packet transfers, Right: geometric increase in window size before *ssthresh* is reached (simulation with $RTT \approx 0.25$ s).

to the available network bandwidth, which is *a priori* unknown. Slow start finishes when *cwnd* exceeds the slow start threshold *ssthresh*. The *congestion avoidance* phase is then entered, where window growth slows to linear, via $cwnd \leftarrow cwnd + MSS^2 / cwnd$. The initial value of *ssthresh* is not prescribed by the standards and varies between different implementations from 1 (i.e., no slow start), to ∞ . An important fact is that for small transfers slow start is the dominant part of this protocol. In the current Internet, short TCP connections are very common, for instance, the median WWW file transfer is around 3–4 kB, which means that at least half of web transfers take less than 5 packets. This means that the download bandwidth is strongly constrained by the RTT, and has very little to do with the transmission speed of the link for most sources. This is particularly true for wireless links where RTTs are larger.

Loss Indications, Retransmissions, and Recovery: Slow start and congestion avoidance describe how the window size is increased when packets are acknowledged, assuming no loss. If a packet is lost however, TCP assumes this is due to congestion and reacts by decreasing the window size, as well as retransmitting the lost packets. There are two types of packet loss indications: *retransmission timeout* (RTO) which is reliable but slow, and *duplicate acknowledgments*, which is much quicker but less reliable. With RTO's, the sender simply assumes that if a packet has not been acknowledged within some timeout then it has been lost. The methods of choosing the RTO value are described in [78]. The second method is based on the fact that if packets arrive out-of-sequence the receiver can only acknowledge the *contiguous* part of received data, but is also required to send a duplicate acknowledgment for this part of the data to indicate the out-of-sequence packet. Thus, detection of multiple duplicate acknowledgments suggests packets arriving after a loss. Reaction to loss indications consists of retransmission of packets, and resetting of *cwnd* and *ssthresh*. In either case *ssthresh* is reset to $\max(\text{FlightSize}/2, 2 \times MSS)$, where *FlightSize* is the amount of sent data as yet unacknowledged. Timeouts tend to occur when congestion is severe, and duplicate ACK's when congestion is light. The two events are, therefore, treated differently. When a timeout occurs the sender returns *cwnd* to the initial value and begins again with slow start. With duplicate acknowledgment the sender need not be so drastic. *Fast Recovery/Fast Retransmission*

are two intimately related algorithms for effectuating a quick recovery from isolated losses. For details of their operation we refer the reader to [5].

F. There Must be More to Network Traffic Than LRD

The very definition of LRD given above is asymptotic, and, therefore, makes no statement on the nature of correlations over small time scales, nor what constitutes “small” or “large” scales. There is no question that short-range correlations in traffic can be very strong; for example it is well known that this is true of variable bit-rate video data, and very significant for networking performance problems. It is, therefore, important to examine the causes and origins of the small time scale correlations in traffic and understand how they impact (and are impacted by) various aspects of network performance. In doing so, one can hardly avoid being critical toward traditional “open loop” traffic modeling and instead emphasizing the need to deal with closed-loop network scenarios and models, as network feedback has the potential to impact or entirely remodel the “short” to “medium” term dependencies.

The first step must again be to turn to data to see what actually occurs over scales below the asymptotic LRD regime. Several studies of TCP traffic over wide-area networks (WANs) have shown an extraordinary degree of variability on small time scales, from milliseconds or below up to the scales where LRD dominates, say, 1 second and above, depending on the networking setting where the traces were collected. For some traditional open loop queueing models, the effects of such variability have been shown to be strongly negative [26]. In each of [4], [28], [66], [68], unconventional but nevertheless open-loop scaling models were investigated to describe the observed type of variability, which can be thought of as local irregularity. Indeed, when looking at Logscale Diagrams over small scales, apparent straight lines are observed, the signatures of scaling, with slopes that are quite different to those found at large scales. Furthermore, with remarkable regularity, a sharp change point is seen at scales that roughly correspond to the round-trip times experienced by the packets. Unlike the case for large scales however where the scale invariance is described through a single parameter, the scaling seen over small scales seems to be more consistent with richer scaling models, such as *Multifractals*, where an entire spectrum of exponents are required; in [4], [68] an even richer class, *Infinitely Divisible Cascades*, was proposed.

The case for richer scaling models is a topic of current research. It remains inconclusive, partly because the statistical evidence is unclear as to whether the spectrum of exponents is really a superior description, but particularly because a network-derived explanation for such behavior has proven elusive. In addition, as mentioned above, at such scales the closed-loop feedback components of traffic, notably that of TCP, can be expected to have a significant effect on the dependence structure. We, therefore, focus on the opposite direction, beginning with dynamic network effects due to TCP which, in turn, appear capable of modifying to a certain degree the small scale variability. There is a small but growing

literature on the nature and effects of TCP dynamics, by which we mean deterministic effects, although we allow hybrid deterministic/stochastic approaches. We now provide a short, and necessarily incomplete, survey of this work.

There are a number of publications [41], [48], [51], [52] which have demonstrated that under suitable assumptions the average window size of a source goes as the inverse square root of the loss probability: $cwnd = k/\sqrt{p}$. Although the assumptions in any one case are somewhat restrictive (random independent losses, persistent sources, low loss rates) the result holds for a range of different conditions (independent losses to periodic losses, TCP-Tahoe, Reno, Delayed Acknowledgments) and nonnegligible loss rates, and seems to apply to persistent traffic on real networks, with $k \approx 1$. Note that by *persistent* or *greedy* sources we mean those which are always trying to transmit at the maximum rate the protocol will allow. There are also more complex formula that apply over a wider range of values of p [52], or for more complex loss patterns [6]. This body of work however, is largely restricted to coarse measures of traffic such as long term throughput rates and avoids dealing with the large-scale aspect of real networks.

A much finer analysis is possible using the Max+ machinery [8], [9], which is in principle capable of providing rigorous results over quite a wide range of scenarios, including tandem networks. Thus far, only throughput-type measures have been derived using it. It remains to be seen if this framework allows the derivation of a full time-scale analysis in the context of a large-scale network. In a recent paper [10], the same authors also show that the competition between TCP sources under a uniformly shared loss assumption can generate a radical dispersion between the individual instantaneous rates, whose sum is bounded by the link bandwidth. It will be interesting to see if this dispersion persists under more realistic loss and networking assumptions, and if it can provide a convincing mechanism for temporal scaling.

A dynamical systems approach, in the sense of chaos theory, has also been attempted. In [70], it is shown (using simulation) that TCP dynamics, at least in the case of a simple network with small numbers of greedy sources and small buffers, is capable of generating very complex chaos-like behavior, as well as (for some parameter values) apparent scaling over a range of scales (for an earlier references, see [55]). It seems unlikely however this is the cause of the observed scaling in large-scale networks, where the highly heterogeneous mix of traffic sources, with their random arrivals and typically small data volumes, add a great deal of random noise to the system, which would tend to erase whatever dynamics is behind the chaos. Furthermore in real networks large buffers are the norm.

Note that the dynamical systems approach we employ in the next two sections is totally different, in a way opposite, to that of [70]. We *define* a traffic source to have chaotic features, and use it to show how it can *impact*—using the terminology introduced in Section II-A—the “size” of LRD behavior, and its onset scale, but not its “character.” The chaos we consider is of a very extreme *mixing* type, and, there-

fore, models a complex system whose dynamics is stable to perturbations, and even has stochastic features. In this sense, our approach is related to the work by Kelly and co-workers [40], [34], who describe dynamical systems that represent TCP-type rate control algorithms and establish stability and fairness properties by showing that, with an appropriate formulation of a network-wide optimization problem, the network’s implicit objective function provides a Lyapunov function for the dynamical systems defined by the rate control algorithms. It would be interesting to establish a possible connection between Kelly’s approach and that outlined below.

As we have seen, there are a variety of avenues currently being explored in the quest to understand and quantify the structure of TCP-rich network traffic data on small time scales. We briefly outline here some of the challenges, before moving to add our own contribution in the remaining two chapters.

Measurement, Inference and Estimation: The richness of traffic is such that one is always in need of more powerful data gathering and processing infrastructures on the one hand, and statistical analysis methods on the other. For existing estimation techniques, the most urgent requirement is increasing their robustness to nonstationarities of various types, which will always be present, despite the luxury of huge data sets which allow apparently stationary subsets to be selected. Closely related to this is the need for formal hypothesis tests to more rigorously select between competing conclusions, and closely related in turn is the need for reliable confidence intervals to be computable, computed, and used intelligently. The statistical questions themselves include investigating the degree of validity of scaling models at small scales, and understanding the relations between the small scale behavior one finds in *different* time series extracted from the same raw traffic data. A key question is, which effects can be seen to “control” which others? If an answer to this can be found at a statistical level, it would suggest corresponding connections at a network causal level.

The Mathematics of Modeling: Many of the approaches discussed so far: multifractals, infinitely divisible cascades, Max+ algebra, dynamical systems, involve significant mathematical challenges at several levels. These include problems of description and understanding of the traffic source models themselves, properties of estimators for their parameters, and difficulties in extending results to cover the range of effects important in the network. These also include taking into account the heterogeneity of sources, details of router scheduling disciplines, the finer points of the TCP protocol, features of traffic at the application or even user levels, and aspects that are crucial for capturing the essence of the interactions present in large-scale TCP-type closed loop systems.

Finding the Grail: The final category of challenges is to integrate the different elements which impact (and are impacted by) traffic, to synthesize a viewpoint which is as parsimonious and robust as it is accurate and useful. Current models seem unlikely to provide the “final” answer, and they lack the correct intuitive vision which allows us to see when they are predominant, and how they combine with

other mechanisms. A central component of any such answer must be: can it tell us if the nature of traffic as it stands today is based on robust mechanisms, which will cause it to propagate into the future despite the furious evolution of the phenomenon, or is it a confused mix of effects which will evolve beyond recognition in a few years, and if so, to what?

III. CHAOTIC MAP MODELS OF PACKET TRAFFIC

A. Definitions

A chaotic map $f: V \rightarrow V$ can be defined by the following three criteria [21]:

- 1) f shows *sensitive dependence on initial conditions* (SIC) which is to say that typical trajectories starting from arbitrarily close initial values nevertheless diverge at an exponential rate. SIC is the basis of the “deterministic, yet unpredictable” property of chaotic systems, because errors in the estimates of initial states become amplified and prevent accurate prediction of the trajectory’s evolution.
- 2) f is *topologically transitive* or *strongly mixing*, which implies that V is irreducible, that it can not be decomposed into subsets that remain disjoint under repeated action of the map.
- 3) periodic points are dense in V .

From a modeling perspective, even low order chaotic maps can capture complex dynamic behavior. It is possible to model traffic sources that generate fluctuations over a wide range of time scales using chaotic maps that require a small number of parameters.

The application of chaotic maps to model packet traffic was developed in [23], [24], [56]. We outline the basic approach proposed in these papers, which is to associate packet source activity with the evolution of a chaotic map. Consider the one dimensional map in Fig. 2 in which the state variable x evolves according to $f_1(x)$ in the interval $[0, d]$ and according to $f_2(x)$ in $[d, 1]$.

One can model an on-off traffic source by stipulating that the source is active and generates one or more packets whenever x is in the interval $[d, 1]$, and is idle otherwise. By carefully selecting the functions $f_1(x)$ and $f_2(x)$ one can generate a range of on-off source behavior.

The traffic generated in this manner can be related to fundamental properties of the map. As the map evolves over time, the probability that x is in a given neighborhood is given by its *invariant density* $\rho(x)$ which is the solution of the Frobenius–Perron equation [62]:

$$\rho(x) = \int_0^1 \delta(x - f(z))\rho(z) dz. \quad (7)$$

If the source is assumed to generate a single packet with every iteration in the on-state ($d \leq x \leq 1$), the average traffic rate is simply given by:

$$\lambda = \int_d^1 \rho(x) dx. \quad (8)$$

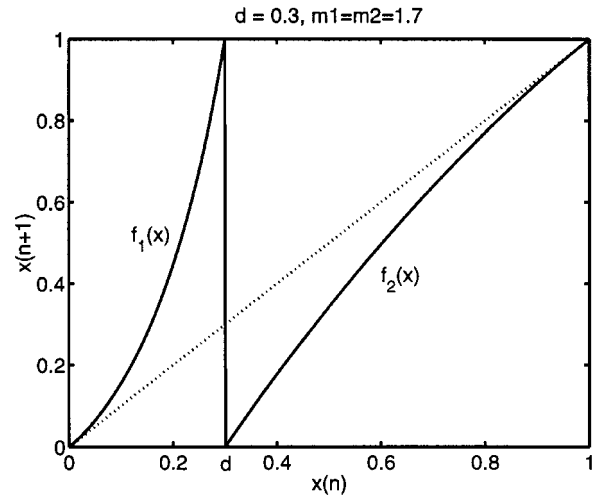


Fig. 2. The chaotic map with $d = 0.3$ and $m_1 = m_2 = 1.7$.

Another key source characteristic is the “sojourn” or run times spent in the on- and off-states. Given the deterministic nature of the mapping, the sojourn time in any given state is solely determined by the reinjection or initial point x_0 at which the map reenters that state. For example, let the map reenter the on-state at $x_0 (\geq d)$. The sojourn time there is then the number of iterations k it takes the map to leave the interval $[d, 1]$:

$$f_2^{k-1}(x_0) \geq d, \quad f_2^k(x_0) < d. \quad (9)$$

One can then derive the distributions of the on- and off-periods from the reinjection density $\psi(x)$ [where the probability that the map reenters a state in the neighborhood $(x_0, x_0 + dx)$ is given by $\psi(x_0) dx$] and (9). The reinjection density $\psi(x)$ can be derived from the invariant density:

$$\psi(x) = \rho(f^{-1}(x)) \frac{df^{-1}(x)}{dx}. \quad (10)$$

One can use these relations to demonstrate the following.

- 1) The sojourn times in the two states are geometrically distributed if $f_1(x)$ and $f_2(x)$ are linear:

$$\begin{aligned} f_1(x) &= \frac{x}{d}, & 0 \leq x < d \\ f_2(x) &= \frac{1-x}{1-d}, & d \leq x \leq 1. \end{aligned} \quad (11)$$

This follows from the fact that the invariant density in this case is uniform, and substituting for the linear mappings in (9).

- 2) In order to match the heavy-tailed sojourn time distribution behavior observed in measurement studies one can choose $f_1(x)$ such that as $x \rightarrow 0$

$$f_1(x) \sim x + cx^m \quad (12)$$

where $3/2 < m < 2$. Note that this function evolves very slowly in the neighborhood of 0 (a fixed point). This results in sojourn times in the off-state that are characterized by distributions with infinite variance.

One can similarly generate extended on-times by choosing $f_2(x)$ so that it behaves as (12) in the neighborhood of 1 (another fixed point). In this way one can match the heavy-tailed sojourn time distribution observed in measurement studies.

For our purposes, it is convenient to use the following map, referred to as the *Fixed Point Double Intermittency* (FPDI) map in [56]:

$$f(x; m_1, m_2, d) = \begin{cases} \frac{x}{(1-c_1x^{m_1-1})^{\frac{1}{m_1-1}}}, & \text{if } x < d \\ 1 - \frac{1-x}{(1-c_2(1-x)^{m_2-1})^{\frac{1}{m_2-1}}}, & \text{if } x \geq d \end{cases} \quad (13)$$

where

$$c_1 = \frac{1-d^{m_1-1}}{d^{m_1-1}} \quad (14)$$

$$c_2 = \frac{1-(1-d)^{m_2-1}}{(1-d)^{m_2-1}}. \quad (15)$$

See Fig. 2 for an example. Note that $f_2(x)$ is a reflected and shifted version of $f_1(x)$.

There are several reasons for choosing this particular map:

- 1) One can model a wide range of on-off behavior by suitably choosing m_1 and m_2 . For example, setting $m_1 = m_2 = 1$ and evaluating the limits using L'Hopital's rule, the FPDI map reduces to the Bernoulli Shift (11), characterized by geometrically distributed sojourn times.
- 2) In the neighborhood of the fixed points 0 and 1, $f(x)$ possess the intermittency behavior of (12), so that for m_1 and/or m_2 in the range $(3/2, 2)$ the map generates sojourn times that have heavy tailed distributions.
- 3) The "fixed point" in its name refers to a unique property of $f_1(x)$ and $f_2(x)$, in that their functional form is invariant under composition, whereas typically even simple nonlinear functions quickly escalate in complexity. For the FPDI map, applying $f_1(\cdot)$ to x repeatedly w times yields precisely the same form: if $x, f_1(x), \dots, f_1^{(w-1)}(x) < d$

$$f_1^w(x; m_1, m_2, d) = \frac{x}{(1-wc_1x^{m_1-1})^{\frac{1}{m_1-1}}}. \quad (16)$$

This is a special property that we exploit extensively in modeling TCP windowing behavior, as well as variable RTTs.

One useful result that follows immediately from this invariance property is an expression for the remaining sojourn time of a source $s_i(n)$ in an on- or off-state, given the current value of $x_i(n) \geq d$

$$s_i(n) = \left\lceil \frac{(1-x_i(n)) \left[1 - \left(\frac{1-x_i(n)}{1-d} \right)^{m_i-1} \right]}{c_2(1-x_i(n))^{m_i}} \right\rceil \quad (17)$$

otherwise if $x_i(n) < d$

$$s_i(n) = \left\lceil \frac{x_i(n) \left[1 - \left(\frac{x_i(n)}{d} \right)^{m_i-1} \right]}{c_1x_i(n)^{m_i}} \right\rceil \quad (18)$$

where $\lceil x \rceil$ denotes the smallest integer larger than x .

The FPDI on-off source model defined in Fig. 2 can be aggregated to model network traffic flows derived from a number of independent on-off sources. In particular, aggregating a large number of on-off sources based on nonlinear segments satisfying (12) generates an approximation of fBm [56]. The Hurst parameter H of the resultant traffic is

$$H = \frac{3m-4}{2m-2} \quad (19)$$

where $m = \max(m_1, m_2)$.

IV. CHAOTIC MAP MODELS OF TCP FEEDBACK

A. Motivation

The on-off source models described in Section III are "open loop" and evolve independently of the network conditions created by the offered traffic. In particular, they do not account for the variety of TCP flow control mechanisms which regulate source behavior in networks—such as the exponential window increase, the response to losses, and the impacts of round-trip times.

Of particular interest to us is the impact of such TCP dynamics on the scaling behavior of network traffic. As we indicated earlier, the current views on this subject run the gamut of possibilities—from some studies that claim that TCP feedback eliminates self-similarity to the extent that fBm queueing predictions do not hold (e.g., [7]; but see also [39]); to other studies that claim that self-similarity can in fact be generated by such feedback [55], [70].

Note that any network scenario in which flow controls substantially alter source behavior violates condition (3) for the validity of the fBm model (which is consistent with the observations of [7]). Note also that most traffic traces used in measurement studies that establish the existence of self-similarity are predominantly TCP. These two apparently conflicting observations can be reconciled on the basis of several insights.

- 1) Self-similarity has been observed in a wide range of networking scenarios, including ones in which TCP is not used, for e.g., in aggregates of ISDN traces [77], which suggests that at least one of the sources of self-similarity is fundamental to the nature of data communications—specifically, heavy-tailed distributions of the files transferred over the Internet; or heavy-tailed "think" times in interactive computing. While we do not rule out other causes (such as protocol interactions), we believe that this is the most robust mechanism of the many that have been proposed to explain self-similarity.
- 2) TCP flow controls cannot eliminate the heavy-tailed on-off sojourn time distributions, but they can modulate the on-times. In particular, the effect of TCP dy-

namics can be expected to be most significant on finer time scales (of the order of a round-trip time or lower). One can, therefore, characterize the impact of TCP as changing the parameters of self-similar scaling—the fBm parameters, together with the lower cutoff time scale that marks the beginning of the scaling region.

- 3) Because TCP feedback modifies the self-similarity in the offered traffic, “open loop” modeling approaches, in which traffic is described in isolation from the performance analysis, will not accurately predict TCP performance. This is not a limitation of the fBm queueing model alone; all open loop modeling approaches will fail when network feedback is a significant consideration.

What is required, therefore, is an integrated description and analysis approach that on the one hand captures the impact of the traffic on the network (such as queueing) and on the other captures the impact of network state on source behavior.

Up to this point, analytical models of TCP analysis have been largely based on persistent sources that in effect offer infinite file sizes (see for instance [34], [58]). As can be expected, traffic aggregated from such sources shows no scaling behavior. Chaotic maps allow the modeling of complex source behavior (e.g., heavy-tailed file size distributions) and the ability to couple these with other dynamical equations describing the evolution of network state.

We first show how window evolution can be incorporated into the basic source model. In particular, we focus on slow start (as opposed to congestion avoidance) because:

- 1) it affects all connections;
- 2) it is the dominant effect for the majority of connections;
- 3) it has a more dramatic effect on the dynamics than congestion avoidance.

B. Zero Loss Dynamics

In order to isolate the impacts of the various mechanisms in TCP, we start by considering the case of zero loss. In effect we assume buffers to be so large in relation to the cumulative window size across all connections as to render loss very rare. Second, as a starting point, we assume that the RTTs are constant, and do not change across sources and from packet to packet. This is representative of the case in which RTTs are dominated by fixed propagation delays, the effect of queueing delays is negligible, and there is a single dominant route. The only aspect of TCP feedback modeled then here is the increase of window size, either up to its maximum value, or up to a point when the “file transfer” is completed.

We base our source model on the following correspondence between the networking scenario and the chaotic map model.

- 1) There are N identical, independent sources, each with its own “dummy” state variable $x_i(n)$, which governs its traffic generation behavior [e.g., a source is “on” when $x_i(n) \geq d$].

- 2) Every iteration of the map corresponds to one RTT, which is, therefore, the smallest time scale modeled; in particular we do not attempt to generate or analyze phenomena that may be observed on time scales smaller than one RTT (e.g., multifractality).
- 3) One sojourn in the on-state corresponds to one TCP session (this is the simplest abstraction of web browsing behavior).
- 4) The starting point in the on-state determines the length of the “file” in packets to be downloaded. Hence, the file sizes are random variables determined by the map.
- 5) If source i is in the on-state, and its window size is $w_i(n)$, the source will send up to this many packets during a single iteration of the system. It actually transmits $p_i(n)$, which is the lesser of $w_i(n)$ and the residual file size, $s_i(n)$, so that if source i is in the on-state

$$p_i(n) = \min(w_i(n), s_i(n)). \quad (20)$$

- 6) The source does not generate packets in the off-state, during which it evolves one iteration at a time. The off-states, which correspond to the “think times” between download requests, are assumed to be independent of the TCP and network states.

With these correspondences in mind, we can write a coupled set of dynamical equations describing the evolution of TCP windows and the source states. If for source i , $x_i(n) \geq d$, then $\text{state}_i(n) = \text{On}$, and

$$w_i(n+1) = \begin{cases} 1, & \text{if } \text{state}_i(n-1) = \text{off} \\ \min(2w_i(n), w_{\max}), & \text{otherwise} \end{cases} \quad (21)$$

$$x_i(n+1) = f^{(p_i(n))}(x_i(n); m_1, m_2, d) \quad (22)$$

otherwise if $x_i(n) < d$, then $\text{state}_i(n) = \text{off}$, and

$$w_i(n+1) = 0 \quad (23)$$

$$x_i(n+1) = f(x_i(n); m_1, m_2, d) \quad (24)$$

where $w_i(n)$ is the window size at time n , w_{\max} is the maximum window size, determined in most cases by the advertised receive window, and $p_i(n)$ is the number of packets sent for source i in RTT n (20). $\text{State}_i(n)$ is used to represent the state of the system after the iteration, $f^{(j)}(\cdot)$ denotes iterating the map j times, $f(\cdot)$ is the FPDI map above, and the s_i are the calculated sojourn times from (17) and (18). As before, we initialize the x_i by a uniform random variable on the interval $(0, 1)$. Note also that we are iterating the chaotic map a number of times at each step, which can be readily evaluated using (16).

The window size determines the number of packets generated by a single source in a round-trip time. The function $f(\cdot)$ has to be, therefore, iterated this many times in a single time step to produce the residual “file size.” The total traffic on the link (given N sources) is then:

$$t(n) = \sum_{i=1}^N p_i(n). \quad (25)$$

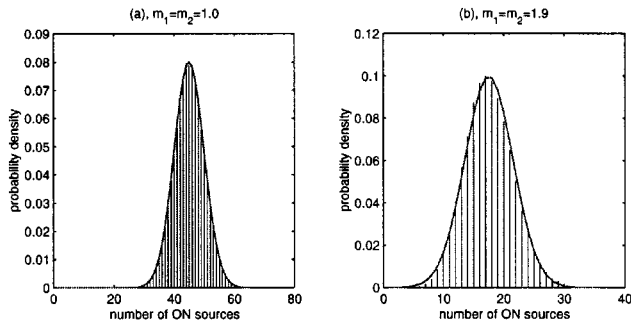


Fig. 3. The distribution of the number of on-sources. Vertical bars are from the simulation. The line shows a Gaussian fit.

1) *Results:* The above set of equations has been implemented in a C program, and results generated for a variety of cases.⁷ In this section, we present results for two simulations, each over 100 000 iterations long, to illustrate the behavior of the above system. The two cases considered are

- a) geometric on-off times: $m_1 = m_2 = 1.0$;
- b) heavy-tailed on-off times: $m_1 = m_2 = 1.9$,

with $d = 0.5$, $w_{\max} = 64$, and $N = 100$ for both.⁸

We first examine in Fig. 3 the distribution of the number of on-sources, which has a limiting Gaussian form, as in the simple on-off models. However, when we examine the marginal distribution of the traffic rate over one RTT (in Fig. 4), it remains Gaussian only for case a), while in case b) we see what appears to be multiple superposed Gaussian distributions with some skewness in the means. This can be explained by noting that the peaks of the superposed distributions are 64 packets apart, which is the maximum window size. To further clarify, consider Fig. 5, which shows a log-log plot of the window size distribution. Case a) corresponds to geometrically distributed file sizes, where there is a negligible probability that a source will be on for long enough to reach the maximum window size, but in case b) there is a significant probability of this happening, and in fact, there will be cases where a single source is “on” for a long time (see following figures). Effectively, for a period of time there is a net increase in the traffic rate equal to $w_{\max} = 64$. This causes the secondary peak in the traffic rate, with subsequent peaks corresponding to two or more persistent sources.

As link capacities and traffic volumes increase, and the maximum window size becomes relatively small, the skewness becomes less noticeable, and the aggregate more closely resembles a Gaussian. This is an illustration of Condition (1) in Section I-A.

Another effect of the window increase is the “compression” of the on-times. Fig. 6 shows log-log plots of the distribution of the off-times, and as expected, we see in case a) a geometric decrease, and in case b) a power-law decay. As should be expected from the symmetry of our chosen param-

⁷Initial values of x are chosen from a uniform distribution on $[0, 1]$. The first 1000 iterations have been discarded to eliminate transients.

⁸We have in fact simulated many other cases, including some with $m_1 \neq m_2$, but it is convenient here to consider the symmetric case to contrast the behavior of the on- and off-times.

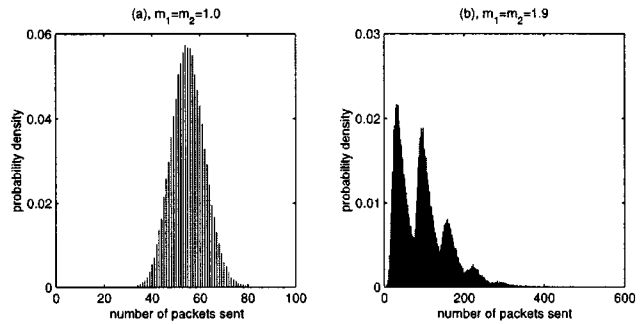


Fig. 4. The distribution of the marginal traffic rate (over one RTT).

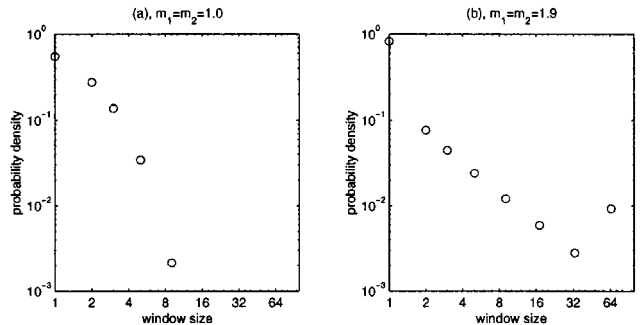


Fig. 5. Log-log plots of the window size distributions.

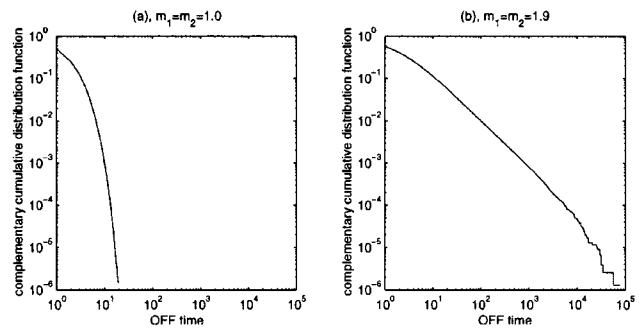


Fig. 6. Log-log plots of the off-time distribution (in RTTs).

eters the distributions of file sizes also reflect these characteristics. However when one examines the distribution of the on-times (shown in Fig. 7) in case b) we see a noticeable deviation from a strict power-law. This is explained by the rate doubling slow start algorithm which acts like a logarithmic compression of the number of RTTs needed to transmit the desired number of packets. The number X of RTTs required to transmit a file of N packets is given by

$$X(N) = \begin{cases} \lfloor \log_2(N) \rfloor + 1, & \text{if } X < 2w_{\max} \\ \left\lceil \frac{N - 2w_{\max} + 1}{w_{\max}} \right\rceil + X(2w_{\max} - 1), & \text{if } X \geq 2w_{\max}. \end{cases} \quad (26)$$

The circles in Fig. 7(b) shows the distribution that would result from (26) based on the actual file size distribution. We can see that the curve matches the on-time distribution almost exactly. The lower part of the on-time distribution remains power-law because in these cases the maximum window size ($w_{\max} = 64$) is reached.

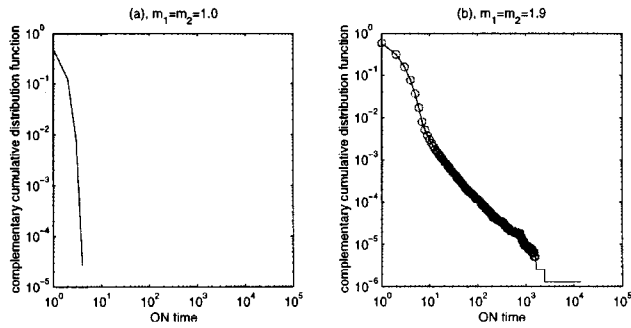


Fig. 7. Log-log plots of the on-time distribution (in RTTs). The circles in plot (b) show the predicted on-time distribution derived from the file size distribution via (26).

One can extrapolate from this that *if there were no maximum window size and no network bottlenecks, the power-law decay in the file size distribution would be compressed to a geometrically distributed on-time, and in principle, the LRD in the resulting traffic would be eliminated.* This does not necessarily bode well from a network queuing viewpoint, because in effect, the long-range correlations are compressed into heavy-tailed marginals, and there would be very large batch arrivals into network queues. However, in practice, there are limits on the window size and link capacities, and the net effect is to move the onset of the power-law decay in the on-time distributions further out.

The compression of the on-times is also reflected in traffic aggregated from a number of such sources. We apply the wavelet based estimation method to traffic aggregates, and one can see from the Logscale Diagrams plotted in Fig. 8 that in the geometric case there is no evidence of LRD ($H = 0.5$), while in the heavy-tailed case the evidence is clearly visible ($H > 0.5$). The estimated value of H is also close to that predicted in (19). Further, it is of interest to see that the small scale behavior (from scales 1–4) does not follow the strict linear asymptote corresponding to LRD. This is the type of midscale effect characteristic discussed in [60], and we attribute this at least in part to the on-time compression caused by the exponential window increase of TCP slow start. It is a topic for further research to determine the relation between lower cutoffs in the scaling behavior of the aggregated traffic, and the on-periods.

We have also looked at asymmetric maps ($m_1 \neq m_2$). This is analogous to the situation in which the on- and off-sojourn times have different distributions. We have also examined scenarios in which these values vary by source. Individually, and in aggregate, the asymptotic scaling behavior is dominated by the higher value (corresponding to the distribution with the slowest decay). On intermediate time scales, the aggregate traffic may show scaling as well, with the slope determined by a weighted average of the individual scaling parameters. The overall conclusions of this section remain however unchanged by allowing for this type of heterogeneities.

C. Finite Buffer

The next aspect of TCP behavior we model is the response to packet losses. When the buffer is finite, there can be packet losses due to buffer overflows. For simplicity, we model this

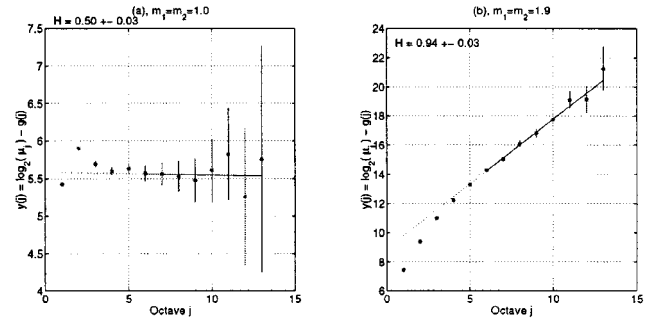


Fig. 8. Log-scale diagrams of the traffic.

here as though packet loss is always detected by a timeout; we are ignoring the “duplicate ack” mechanism outlined in Section II-E. After a time-out, the window size is reduced to 1, and the TCP session reenters slow start. We must also model the buffering of packets which modifies the equations above to include:

$$l(n+1) = [b(n) + t(n) - C - B]^+ \quad (27)$$

$$b(n+1) = [b(n) + t(n) - C - l(n)]^+ \quad (28)$$

where $l(n)$ is the total number of lost packets during RTT n , $b(n)$ is the buffer content at the end of RTT n , C is the bottleneck link capacity in packets per second, B is the buffer size in packets, and $[\cdot]^+$ takes the positive part of the argument.

If $l(n+1)$ is positive, then we choose which packets are lost randomly, thus distributing the lost packets to each source in proportion to the number of packets it sent. Any source which loses packets reduces its window size to 1 in the following RTT.

We do not model retransmission of lost packets in this paper, nor the RTO behavior. It appears that these mechanisms can introduce correlations of their own over some time scales [30], and we do not wish this to obscure the impacts of the window evolution.

1) *Results:* Once again, we simulate the equations above in a C program. The major difference between these results and those with zero loss is that the window sizes are now adjusted to reduce the rate when it gets too large. The result is that fewer sources hit the maximum window size, and the peaks in the traffic rate marginals are less pronounced (though there is still considerable skewness). However, the majority of observations in the zero loss case carry over, at least qualitatively. We can examine a number of features of the aggregated traffic to gain additional insights on how TCP affects scaling behavior, and we illustrate this through a couple of examples.

Relationship Between \mathbf{m} and \mathbf{H} : The first question of interest is whether the relationship in (19) still holds. Fig. 8 indicates that the relationship is at least approximately correct. Here we simulate over a range of values of $m = [1, 2]$, and measure the value of H using the wavelet estimator. Fig. 9 shows the results (using 10 independent simulations to obtain the 95% confidence intervals shown as vertical bars in the graph). The solid line shows (19), and we see that there is some deviation from it, in particular around $m = 1.5$. This can be explained on the basis of earlier observations [56] that convergence to the theoretical value of 0.5 is very slow for

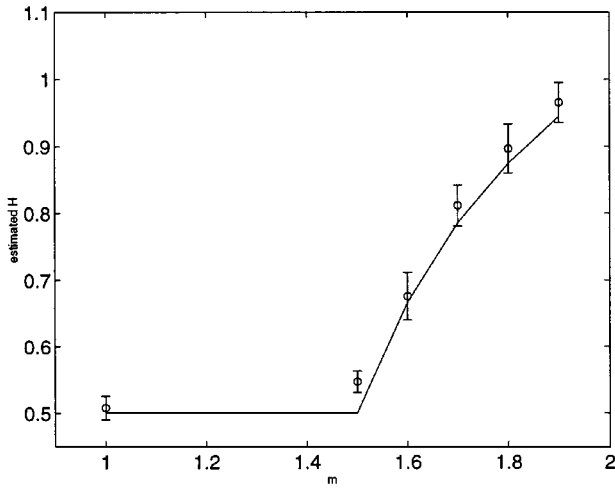


Fig. 9. The relationship between input m and the estimated Hurst parameter for the traffic. The solid line shows (19), while the circles show the mean estimated $H\%$ for 10 simulations, and the bars give 95% Gaussian confidence intervals.

$m = 1.5$; this is the borderline case between short range and long range dependence. Orders of magnitude longer simulations are needed to see the asymptotic properties correctly. For the most part, the predicted value lies within the 95% confidence intervals for the estimated Hurst parameters.

Varying w_{\max} : Fig. 7 shows the effect of TCP slow start's exponential window increase. We can gain further understanding of this effect by varying w_{\max} , which we do in Fig. 10(a) where we examine the on-time distributions. As can be expected, there is no modification of the on-time distribution for $w_{\max} = 1$, because this is equivalent to the open loop case. As w_{\max} increases we see increasing deviation from a straight line, toward a geometric distribution with a power-law tail. This once again supports the hypothesis that the geometric part of the on-time distribution is due to the logarithmic compression of the on-times because of slow start, and the maximum window size determines the cutoff for this effect.

Fig. 10(b) shows the variation in the Logscale Diagrams as w_{\max} increases. When w_{\max} is small we see close to straight lines (except at the smallest scales), as we would expect.⁹ As w_{\max} increases we can see that the curves are translated upward, which is natural as the variation in the traffic increases (increasing the variance of a process by a factor raises the corresponding Logscale Diagram by twice the log of that factor), but more importantly the curve changes shape at fine scales (up to scale 4), which corresponds to changes in the dynamics of TCP at scales up to $2^4 = 16$ times the RTT. Overall, one can conclude that the impact of TCP response to packet losses does not change the conclusions from the zero loss case.

D. Variable RTTs

Forcing the iteration time step of the map to correspond to one RTT is restrictive because it forces all sources to have the same, unvarying RTT. In fact, sources can and do have

⁹Remember that variation of the order of the confidence intervals plotted as vertical bars is to be expected.

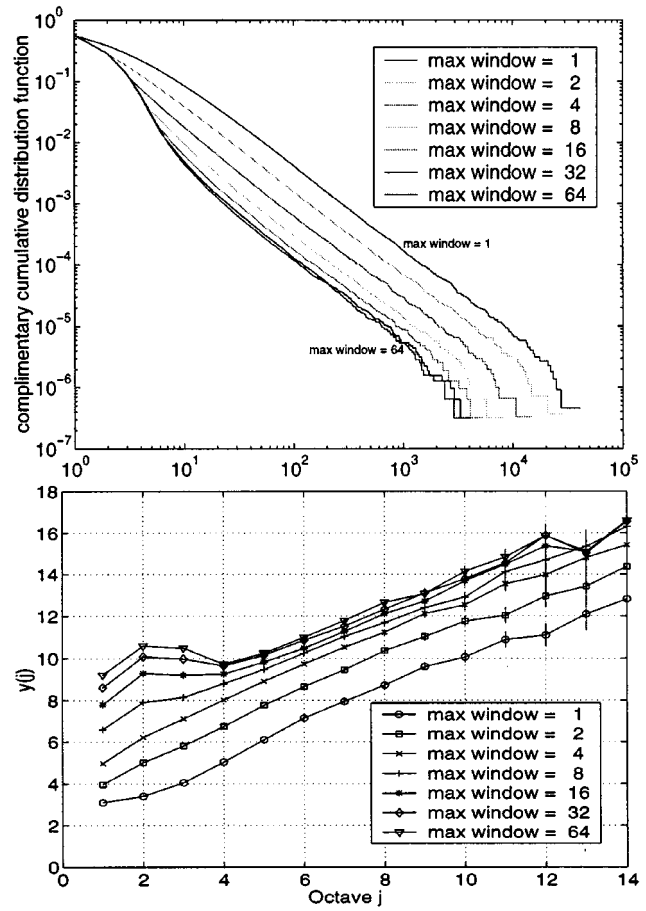


Fig. 10. The effect of varying w_{\max} , the maximum window size: The on-time distribution for a variety of values of w_{\max} (top), the Logscale diagrams for a variety of values of w_{\max} (bottom).

different RTTs, and this can have a strong impact on their relative performance [31]. Moreover, the RTT can vary dynamically within a TCP session. For instance, as the load increases, the queue lengths increase, and, therefore, the RTT increases. This can have as dramatic an impact on the generated traffic as packet losses.

In order to model the impacts of variable RTTs, we need to decouple the time index of the map evolution from a round-trip time. We can still study the map evolution in discrete time, with the time quantum equal to a convenient value, such as the average RTT. Note that as the actual RTT fluctuates over n , it can be lesser or greater than the time step. If a map iterates w steps in an actual $\text{RTT}(n)$, this is equivalent to iterating $w/\text{RTT}(n)$ in a time step. Note that in general $w/\text{RTT}(n)$ will not be an integer. However, the special structure of the FPD map allows the computation of fractional iterations of the map. We can, therefore, explicitly include the effect of RTT using (6) and scale the rate at which sources evolve correspondingly, by replacing w in (16) with $w/\text{RTT}(n)$. The number of packets sent in a fixed interval of time is then:

$$p_i(n) = \begin{cases} \min\left(\frac{w_i(n)}{\text{RTT}_i(n)}, s_i(n)\right), & \text{if state}_i(n) = \text{on} \\ 0, & \text{if state}_i(n) = \text{off} \end{cases} \quad (29)$$

There are two complications.

- 1) When fractional packet loss occurs, there is no obvious way to distribute the losses between sources or identify what the loss means and, hence, we will examine the zero loss case here.
- 2) The reinjection probabilities tend to be skewed by the partial iterations past the threshold d , and so an alternative reinjection method must be used (we have used uniformly distributed reinjection here).

The new system is defined by using the modified equation for transmitted packets, (20), in (21)–(24), in conjunction with the following set of relations:

$$t(n) = \sum_{i=1}^N p_i(n) \quad (30)$$

$$b(n+1) = [b(n) + t(n) - C]^+ \quad (31)$$

$$\text{RTT}_i(n+1) = \frac{b(n+1)}{C} + P_i \quad (32)$$

$$\text{state}_i(n) = \begin{cases} \text{on,} & \text{if } x_i(n) \geq d \\ \text{off,} & \text{if } x_i(n) < d \end{cases} \quad (33)$$

where P_i is the propagation based delay for source i , and b is the number of packets in the buffer. In the high bandwidth, small buffer case with P_i normalized to 1, this reduces to the first case discussed with fixed, equal RTTs.

To gain insights into the effects of dynamically varying RTTs, we examine in detail what happens to a single source under these equations. When multiple sources are multiplexed, an individual source's contribution to the overall growth in the buffer length is small. Therefore, many sources acting independently will not be dramatically affected by allowing RTT to vary dynamically with buffer length. Hence, a single source shows better the effect of buffer dynamics on the TCP window flow control algorithm.

Fig. 11 shows the sample path of a single case. The numbers of packets sent per fixed time interval (iteration of the map), the queue length, and the window size are each plotted on two time scales (the right-hand plots show a longer time scale).

The window size plots show a short initial nonlinear growth in the window (up to iteration $n = 36$), followed by an almost linear growth up to the maximum window size (reached at iteration $n = 69$). Note that the window increase is substantially slower than in slow start with fixed RTTs [see the circles in Fig. 11(a)]. During the short-period of nonlinear growth there is no buffering required, and so the RTTs are simply the propagation delay, and the rate at which packets are sent increases dramatically. However, when the queue begins to fill (at $n = 36$), the RTTs increase, which decreases the rate of window growth, resulting in only linear growth of the window. With approximately linear growth of the window and queue length, with a corresponding linear increase in RTTs, the resulting packet rate (6), levels off, and appears to approach a limit. However, when the window size reaches its maximum and stops growing, the queue and the RTT continue to grow, and so the packet rate decreases.

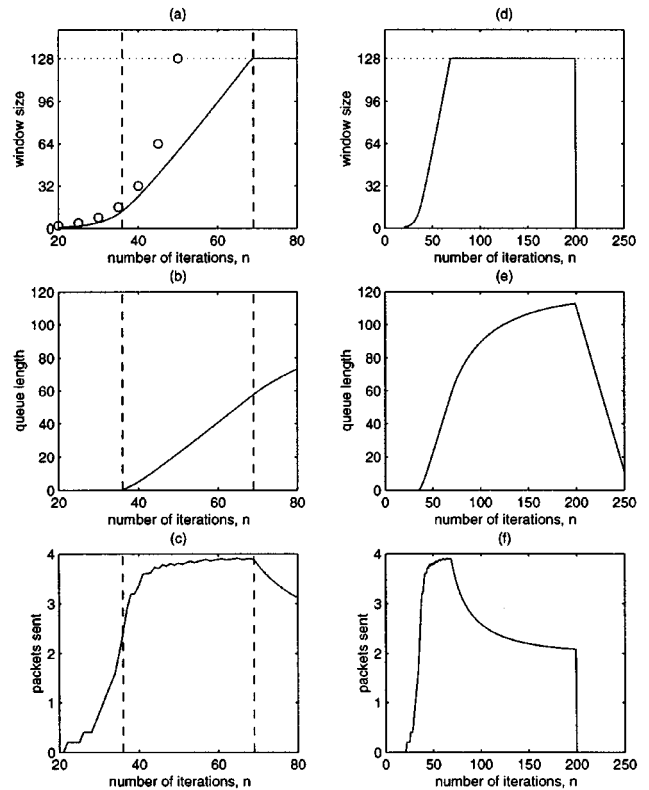


Fig. 11. An example iteration of the variable RTT model ($C = 2$, $P = 5$, $w_{\max} = 128$). The left-hand plots (a)–(c) show a finer time scale plot of the initial part of plots (d)–(f). The three plots are, (a) and (d): the window size, (b) and (e): the length of the queue, and (c) and (f): the number of packets sent per fixed time interval. Circles in (a) show the projected window size for slow start with fixed RTTs. The horizontal dotted line shows the maximum window size, and the vertical dashed lines show two epochs of interest ($t = 36$ and $t = 69$).

The decreasing packet rate, leads to sub-linear growth in the queue, and once again, for $n < 200$ the packet send rate appears to be approaching a limit which corresponds to the link capacity $C = 2$ packets per interval. Finally at $n = 200$ the file transfer terminates, and the packet rate drops to zero, while the queue empties over the next 50–60 time intervals.

Such complex transient behavior is clearly quite different from the fixed RTT case. The plateau effect in the queue length, and the limiting rate for packets appear as natural rate limits. However, note that the queue length at the (second) limit is near the maximum window size which, by the window flow control, would be the maximum number of packets that could be in the network at any one time. Hence, this dynamic limit is approximately what the window flow control would actually achieve, making this simulation much more realistic than those with fixed RTTs. Qualitatively, one can observe that increasing RTT in response to increasing window size has the effect of limiting the offered traffic rate; this stretches out the duration of the ON period while lowering the transmission rate in the on-period. The resulting aggregate can, therefore, be expected to have lower variability with more extended correlations. Quantifying the impacts of varying RTTs on the scaling behavior of the traffic is left for further study.

V. CONCLUSION

It is now widely accepted that packet network traffic exhibits self-similarity over a range of time scales. Secondly, research over the last decade has provided strong evidence that the self-similarity arises from the heavy-tailed distributions that characterize data traffic source activity, though this does not rule out the potential for other sources of self-similarity. A subject of current research is the impact of network feedback (through for example, the TCP-type mechanisms) on network traffic characteristics. After surveying what is currently known on network traffic self-similarity and its origins, we examined the impacts of network feedback using dynamical systems models. We indicated how the joint evolution of a traffic source and the network state can be modeled by a coupled set of equations that are based on chaotic maps. We identified the impacts of distinct aspects of the TCP flow control mechanisms on the traffic. For example, the impact of the slow start window mechanism is to logarithmically compress the ON states, which has the effect of moving out the scaling region in both the aggregated traffic and the tail distribution of the ON state. The aggregated traffic is nevertheless self-similar over a range of time scales. The presence of losses does not substantially change these conclusions. The impact of variable RTTs can significantly alter the dynamics of the source, the network queues, and some aspects of the scaling behavior. In conclusion, TCP-type feedback appears to have the effect of modifying the self-similarity behavior of network traffic, but it neither generates it nor eliminates it.

There is considerable scope for further work, in particular relating parameters that describe scaling behavior in network traffic to source characteristics and the network state. There is considerable scope as well to investigate and extend the use of chaotic map models as closed loop traffic sources—for example, alternate source interpretations of the maps; alternate maps, in particular those which automatically achieve uniform reinjection probabilities in the two states, without the need for external randomization.

ACKNOWLEDGMENT

A. Erramilli and D. Veitch wish to thank Ericsson Research for their generous support.

REFERENCES

- [1] P. Abry and D. Veitch, "Wavelet analysis of long-range dependent traffic," *IEEE Trans. Inform. Theory*, vol. 44, pp. 2–15, Jan. 1998.
- [2] P. Abry, M. S. Taqqu, P. Flandrin, and D. Veitch, "Wavelets for the analysis, estimation, and synthesis of scaling data," in *Self-Similar Network Traffic and Performance Evaluation*, K. Park and W. Willinger, Eds. New York: Wiley, 2000.
- [3] P. Abry, P. Flandrin, M. S. Taqqu, and D. Veitch, "Self-similarity and long-range dependence through the wavelet lens," in *Long-Range Dependence: Theory and Applications*, P. Doukhan, G. Oppenheim, and M. S. Taqqu, Eds. Cambridge, MA: Birkhäuser, 2001.
- [4] P. Abry, D. Veitch, L. Huang, S. Roux, P. Flandrin, and J. Mischeel, "Statistical scaling analysis of TCP/IP data," presented at the ICASSP 2001, Special Session, Network Inference and Traffic Modeling, Salt Lake City, UT, May 7–11, 2001.
- [5] M. Allman, V. Paxson, and W. Stevens, "TCP congestion control," IETF Network Working Group, RFC 2581, 1999.
- [6] E. Altman, K. Avrachenkov, and C. Barakat, "A stochastic model of TCP/IP with stationary random losses," in *Proc. ACM SIGCOMM'2000*, 2000, pp. 231–242.
- [7] A. Arvidsson and P. Karlsson, "On traffic models for TCP/IP," in *Proc. ITC-16*, 1999, pp. 457–466.
- [8] F. Baccelli and D. Hong, "TCP is max-plus linear, and what it tells us on its throughput," in *Proc. ACM SIGCOMM'2000*, vol. 30, Stockholm, Sweden, 2000, pp. 219–320.
- [9] F. Baccelli, "Effect of cross traffic on the performance of tcp," presented at the Workshop on the Modeling of TCP, (École Normale Supérieure), Dec. 1998.
- [10] F. Baccelli and D. Hong, A.I.M.D., fairness and fractal scaling of TCP traffic, Preprint, Apr. 2001.
- [11] S. Borst, O. Boxma, and P. Jelenković, "Generalized processor sharing with long-tailed traffic sources," in *Teletraffic Engineering in a Competitive World, Proceedings of ITC-16*, P. Key and D. Smith, Eds. Amsterdam, The Netherlands: North-Holland, 1999, pp. 345–354.
- [12] S. Borst, O. Boxma, and P. Jelenković, "Asymptotic behavior of generalized processor sharing with long-tailed traffic sources," in *Proc. IEEE INFOCOM'2000*, vol. 2, 2000, pp. 912–921.
- [13] —, "Induced burstiness in generalized processor sharing queues with long-tailed traffic flows," presented at the Allerton Conf., 2000.
- [14] O. J. Boxma and J. W. Cohen, "The single server queue: Heavy tails and heavy traffic," in *Self-Similar Network Traffic and Performance Evaluation*, K. Park and W. Willinger, Eds. New York: Wiley, 2000.
- [15] G. L. Choudhury and W. Whitt, "Long-tail buffer-content distributions in broadband networks," *Perform. Eval.*, vol. 30, pp. 177–190, 1997.
- [16] Computer Science and Telecommunications Board (CSTB), National Research Council, *The Internet's Coming of Age*. Washington, DC: Nat. Academy Press, 2001.
- [17] D. R. Cox, "Long-range dependence: A review," in *Statistics: An Appraisal*, H. A. David and H. T. David, Eds. Ames, IA: Iowa State Univ. Press, 1984, pp. 55–74.
- [18] M. E. Crovella and A. Bestavros, "Self-similarity in world wide web traffic—evidence and possible causes," *IEEE/ACM Trans. Networking*, vol. 5, pp. 835–846, Dec. 1997.
- [19] M. E. Crovella, M. S. Taqqu, and A. Bestavros, "Heavy-tailed probability distributions in the World Wide Web," in *A Practical Guide to Heavy Tails: Statistical Techniques and Applications*, R. Adler, R. Feldman, and M. S. Taqqu, Eds. Cambridge, MA: Birkhäuser, 1998, pp. 3–25.
- [20] M. Crovella, R. Frangioso, and M. Harchol-Balter, "Connection scheduling in Web servers," in *Proc. USITS'99*, Boulder, CO, Oct. 1999.
- [21] R. L. Devaney, *An Introduction to Chaotic Dynamical Systems*, 2nd ed. Reading, MA: Addison-Wesley, 1989.
- [22] N. G. Duffield and N. O'Connell, "Large deviations and overflow probabilities for the general single-server queue, with applications," *Math. Proc. Cambridge Philos. Soc.*, vol. 118, pp. 363–374, 1995.
- [23] A. Erramilli and R. P. Singh, "Application of deterministic chaotic maps to characterize broadband traffic," in *Proc. 7th ITC Specialists Seminar*, Morristown, NJ, 1990.
- [24] A. Erramilli, R. P. Singh, and P. Pruthi, "An application of deterministic chaotic maps to model packet traffic," *Queueing Syst.*, vol. 20, pp. 171–206, 1995.
- [25] A. Erramilli, O. Narayan, and W. Willinger, "Experimental queueing analysis with long-range dependent packet traffic," *IEEE/ACM Trans. Networking*, vol. 4, pp. 209–223, Apr. 1996.
- [26] A. Erramilli, O. Narayan, A. Neidhardt, and I. Sanjeev, "Performance impacts of multiscaling in wide-area TCP/IP traffic," in *Proc. IEEE INFOCOM'00*, vol. 1, 2000, pp. 352–359.
- [27] A. Feldmann, A. C. Gilbert, W. Willinger, and T. G. Kurtz, "The changing nature of network traffic: Scaling phenomena," *Comput. Commun. Rev.*, vol. 28, pp. 5–29, Apr. 1998.
- [28] A. Feldmann, A. C. Gilbert, and W. Willinger, "Data networks as cascades: Explaining the multifractal nature of Internet WAN traffic," in *Proc. ACM SIGCOMM'98*, Vancouver, Canada, 1998, pp. 42–55.
- [29] 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," *Comput. Commun. Rev.*, vol. 29, no. 4, pp. 301–313, 1999.
- [30] A. Feldmann, D. R. Figueiredo, B. Liu, V. Misra, D. Towseley, and W. Willinger, "Does TCP produce self-similar traffic?," Preprint, 2001.
- [31] S. Floyd, "Connections with multiple congested gateways in packet-switched networks—Part 1: One-way traffic," *Comput. Commun. Rev.*, vol. 21, pp. 30–47, 1991.
- [32] S. Floyd and V. Paxson, "Difficulties in simulating the Internet," *IEEE/ACM Trans. Networking*, vol. 9, pp. 392–403, Aug. 2001.

- [33] M. Grossglauser and J.-C. Bolot, "On the relevance of long-range dependence in network traffic," in *Proc. ACM SIGCOMM'96*, 1996, pp. 15–24.
- [34] R. Gibbens, S. Sargood, C. V. Eijl, F. P. Kelly, H. Azmoodeh, R. Macfadyen, and N. Macfadyen, "Fixed-point methods for the end-to-end performance analysis of IP networks," presented at the 13th ITC Specialist Seminar on Internet Traffic Measurement and Modeling, Monterey, CA, Sept. 2000.
- [35] M. Harchol-Balter and A. B. Downey, "Exploiting process lifetime distributions for dynamic load balancing," in *Proc. ACM/SIGMETRICS '96*, Philadelphia, PA, 1996, pp. 13–24.
- [36] Information Sciences Inst., Univ. California, "Transmission control protocol," Internet Engineering Task Force, RFC 793, Sept. 1981.
- [37] V. Jacobson, "Congestion avoidance and control," *Commun. Rev.*, vol. 18, no. 4, pp. 314–329, 1988.
- [38] V. Jacobson, R. Braden, and D. Borman, "TCP extensions for high performance," Internet Engineering Task Force, RFC 1323, 1992.
- [39] Y. Joo, V. Ribeiro, A. Feldmann, A. C. Gilbert, and W. Willinger, "TCP/IP traffic dynamics and network performance: A lesson in workload modeling, flow control, and trace-driven simulations," *Comput. Commun. Rev.*, vol. 31, pp. 25–36, 2001.
- [40] F. P. Kelly, "Mathematical modeling of the Internet," in *Proc. 4th Int. Congress on Industrial and Applied Mathematics*, 1999, <http://www.statslab.cam.ac.uk/~frank/>, to be published.
- [41] T. V. Lakshman and U. Madhow, "The performance of networks with high bandwidth-delay products and random loss," *IEEE/ACM Trans. Networking*, vol. 5, pp. 336–350, June 1997.
- [42] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson, "On the self-similar nature of ethernet traffic (extended version)," *IEEE/ACM Trans. Networking*, vol. 2, pp. 1–15, Feb. 1994.
- [43] J. B. Levy and M. S. Taqqu, "Renewal reward processes with heavy-tailed interrenewal times and heavy-tailed rewards," *Bernoulli*, vol. 6, no. 1, pp. 23–44, 2000.
- [44] B. B. Mandelbrot and J. W. Van Ness, "Fractional Brownian motions, fractional noises and applications," *SIAM Rev.*, vol. 10, pp. 422–437, 1998.
- [45] B. B. Mandelbrot, "Long-run linearity, locally Gaussian processes, H-spectra and infinite variances," *Int. Economic Rev.*, vol. 10, pp. 82–113, 1969.
- [46] ———, *The Fractal Geometry Of Nature*. San Francisco, CA: Freeman, 1983.
- [47] L. Massoulié and A. Simonian, "Large buffer asymptotics for the queue with FBM input," *J. Appl. Probab.*, vol. 36, no. 3, pp. 894–906, 1999.
- [48] M. Mathis, J. Semke, J. Mahdavi, and T. Ott, "The macroscopic behavior of the TCP congestion avoidance algorithm," *Comput. Commun. Rev.*, vol. 27, pp. 67–82, July 1997.
- [49] O. Narayan, "Exact asymptotic queue length distribution for fractional Brownian traffic," *Advances in Perform. Anal.*, vol. 1, pp. 39–63, 1998.
- [50] I. Norros, "A storage model with self-similar input," *Queueing Syst.*, vol. 16, pp. 387–396, 1994.
- [51] T. J. Ott, J. Kemperman, and M. Mathis, "The stationary behavior of ideal TCP congestion avoidance," <ftp://ftp.bellcore.com/pub/tjo/TCPWindow.ps>, 1996.
- [52] J. Padhye, V. Firoin, D. Towsley, and J. Kurose, "Modeling TCP throughput: A simple model and its empirical validation," in *Proc. ACM SIGCOMM'98*, 1998, pp. 303–314.
- [53] K. Park and W. Willinger, *Self-Similar Network Traffic and Performance Evaluation*. New York: Wiley, 2000.
- [54] V. Paxson and S. Floyd, "Wide area traffic: The failure of Poisson modeling," *IEEE/ACM Trans. Networking*, vol. 3, pp. 226–244, June 1995.
- [55] J. M. Peha, "Retransmission mechanisms and self-similar traffic models," in *Proc. IEEE/ACM/SCS Communication Networks and Distributed Systems Modeling and Simulation Conf.*, 1997, pp. 47–52.
- [56] P. Pruthi, "An application of chaotic maps to packet traffic modeling," Ph.D. dissertation, KTH, Stockholm, Sweden, Oct. 1995.
- [57] S. I. Resnick, "Heavy tail modeling and teletraffic data," *Annals Statist.*, vol. 25, pp. 1805–1869, 1997.
- [58] M. Roughan, A. Erramilli, and D. Veitch, "Network performance for TCP networks—Part I: Persistent sources," in *Proc. 17th Int. Teletraffic Congr.*, Dec. 2001.
- [59] M. Roughan, D. Veitch, and P. Abry, "Real-time estimation of the parameters of long-range dependence (extended version)," *IEEE/ACM Trans. Networking*, vol. 8, pp. 467–478, Aug. 2000.
- [60] M. Roughan, J. Yates, and D. Veitch, "The mystery of the missing scales: Pitfalls in the use of fractal renewal processes to simulate LRD processes," in *ASA-IMA Conf. Applications of Heavy Tailed Distributions in Economics, Engineering and Statistics*. Washington, DC, 1999.
- [61] G. Samorodnitsky and M. S. Taqqu, *Stable Non-Gaussian Random Processes*. London, U.K.: Chapman and Hall, 1994.
- [62] H. G. Schuster, *Deterministic Chaos: An Introduction*, 2nd ed. Weinheim, Germany: Wiley-VCH, 1988.
- [63] A. Shaikh, J. Rexford, and K. Shin, "Load-sensitive routing of long-lived IP flows," in *Proc. ACM SIGCOMM '99*, Cambridge, MA, 1999, pp. 215–226.
- [64] R. V. Sole and S. Valverde, "Information transfer and phase transition in a model of Internet traffic," *Physica A*, vol. 289, pp. 595–605, 2001.
- [65] M. S. Taqqu, V. Teverovsky, and W. Willinger, "Estimators for long-range dependence: An empirical study," *Fractals*, vol. 3, no. 4, pp. 785–798, 1995.
- [66] J. L. Véhel and R. H. Riedi, "Fractional Brownian motion and data traffic modeling: The other end of the spectrum," in *Fractals in Engineering '97*, J. L. Véhel, E. Lutton, and C. Tricot, Eds. Berlin, Germany: Springer-Verlag, 1997.
- [67] D. Veitch and P. Abry, "A wavelet based joint estimator of the parameters of long-range dependence," *IEEE Trans. Information Theory (special issue on Multiscale Statistical Signal Analysis and its Applications)*, vol. 45, no. 3, pp. 878–897, Apr. 1999.
- [68] D. Veitch, P. Abry, P. Flandrin, and P. Chainais, "Infinitely divisible cascade analysis of network traffic data," in *Proc. ICASSP 2000*, Istanbul, Turkey, June 2000.
- [69] A. Veres, Z. Kenesi, S. Molnár, and G. Vattay, "On the propagation of self-similarity in the Internet," in *Proc. ACM SIGCOMM'2000*, Stockholm, Sweden, 2000, pp. 243–254.
- [70] A. Veres and M. Boda, "The chaotic nature of TCP congestion control," in *Proc. IEEE INFOCOM'2000*, 2000, pp. 1715–1723.
- [71] W. Willinger, R. Govindan, S. Jamin, V. Paxson, and S. Shenker, "Scaling phenomena in the Internet: Critically examining criticality," in *Proc. Nat. Acad. Sci. USA*, vol. 99, Feb. 2001, pp. 2573–2580.
- [72] W. Willinger, "The discovery of self-similar traffic," in *Performance Evaluation: Origins and Directions*, G. Haring, C. Lindemann, and M. Reiser, Eds. Berlin, Germany: Springer-Verlag, 2000, vol. 1769, Lecture Notes in Computer Science, pp. 493–505.
- [73] W. Willinger and V. Paxson, "Where mathematics meets the Internet," *Notices of the AMS*, vol. 45, pp. 961–970, 1998.
- [74] W. Willinger, V. Paxson, R. Riedi, and M. S. Taqqu, "Long-range dependence and data network traffic," in *Long-Range Dependence: Theory and Applications*, P. Doukhan, G. Oppenheim, and M. S. Taqqu, Eds. Cambridge, MA: Birkhäuser, 2001, to be published.
- [75] W. Willinger, V. Paxson, and M. S. Taqqu, "Self-similarity and heavy tails: Structural modeling of network traffic," in *A Practical Guide to Heavy Tails: Statistical Techniques for Analyzing Heavy Tailed Distributions*, R. Adler, R. Feldman, and M. S. Taqqu, Eds. Cambridge, MA: Birkhauser Verlag, 1998, pp. 27–53.
- [76] W. Willinger, M. S. Taqqu, R. Sherman, and D. V. Wilson, "Self-similarity through high-variability: Statistical analysis of Ethernet LAN traffic at the source level," *IEEE/ACM Trans. Networking*, vol. 5, pp. 71–86, 1997.
- [77] W. Willinger, M. S. Taqqu, and A. Erramilli, "A bibliographical guide to self-similar traffic and performance modeling for modern high-speed networks," in *Stochastic Networks: Theory and Applications*, F. P. Kelly, S. Zachary, and I. Ziedins, Eds. Oxford, U.K.: Clarendon Press, 1996, vol. 4, Roy. Stat. Soc. Lecture Note Series, pp. 339–366.
- [78] G. R. Wright and W. R. Stevens, *TCP/IP Illustrated, Vol. 2*. Reading, MA: Addison-Wesley, 1995.
- [79] Y. Zhang and V. Paxson, "Detecting stepping stones," in *Proc. 9th USENIX Security Symp.*, Aug. 2000.
- [80] ———, "Detecting backdoors," in *Proc. 9th USENIX Security Symp.*, Aug. 2000.
- [81] X. Zhu, J. Yu, and J. Doyle, "Heavy tails, generalized coding, and optimal Web layout," in *Proc. IEEE INFOCOM'01*, vol. 3, Anchorage, AL, 2001, pp. 1617–1626.
- [82] A. P. Zwart and O. J. Boxma, "Sojourn time asymptotics in the $M/G/1$ processor sharing queue," *Queueing Syst.*, vol. 35, pp. 141–166, 2000, to be published.

- [83] 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," in *Proc. ACM/SIGCOMM'99*, Cambridge, MA, Sept. 1999, pp. 301–313.
- [84] M. S. Taqqu, V. Teverovsky, and W. Willinger, "Estimators for long-range dependence: An empirical study," in *Fractal Geometry and Analysis*, C. J. G. Evertsz, H.-O. Peitgen, and R. F. Voss, Eds. Singapore: World Scientific, 1996.



Ashok Erramilli (Member, IEEE) received the bachelor's degree in physics from Poona University, India, in 1977, and the bachelor's degree in electrical communications engineering from the Indian Institute of Science in 1980, the M.S. and Ph.D. degrees in electrical engineering from Yale University, New Haven, CT, in 1981, and 1984, respectively.

He joined ITT's Advanced Technology Center in 1983, where he developed congestion and flow control strategies for high-performance packet switches. He joined Bellcore's Traffic Analysis groups in 1987, where he worked on the management of traffic flows across the full range of communications networks, with emphasis on the characterization, engineering and control of packet traffic. In 1995, he became manager of Bellcore's network design and traffic analysis research group, with responsibility for developing traffic management methods and tools needed to operate high-speed Frame Relay, ATM and IP networks. Since 1997, he has been an independent consultant, offering research, consulting and software development services for ATM, IP, 3G Wireless and optical networks. He has been a Visiting Professor at the Royal Institute of Technology in Stockholm, Sweden, between 1993–1996, and at the Royal Melbourne Institute of Technology in Melbourne, Australia (1999).

Dr. Erramilli has authored over 35 papers in the field of network traffic.



Matthew Roughan (Member, IEEE) received the Ph.D. degree in 1994 in applied mathematics from the University of Adelaide, in Australia.

From 1994 to 1996 he worked at the Co-operative Research Center for Sensor Signal and Information Processing, in Adelaide, on a diverse range of projects such as automatic interpretation of ionospheric data, and detection of land mines. From 1997 to 1999, he worked at the Ericsson/RMIT Software Engineering Research Centre in Melbourne, Australia, and in 2000,

continued to work with Ericsson at the new EMULab within the Department of Electrical and Electronic Engineering at the University of Melbourne. Since December 2000, he has been working at AT&T Labs-Research, as a member of the IP Network Management and Performance Department, on various projects involving traffic analysis and network performance modeling.



Darryl Veitch completed a B.S. (honors) degree at Monash University, Melbourne in 1985, and a mathematics doctorate in Dynamical Systems at the University of Cambridge, U.K., in 1990.

In 1991, he joined the research laboratories of Telecom Australia (Telstra) in Melbourne where he became interested in long-range dependence as a property of tele-traffic in packet networks. In 1994, he left Telstra to pursue the study of this phenomenon at the-CNET in Paris (France Telecom). He then held visiting positions at the

KTH in Stockholm, INRIA in the south of France, and Bellcore, Morristown, NJ, before taking up a three-year position as Senior Research Fellow at RMIT, Melbourne. He is now a Senior Research Fellow in the Electrical and Electronic Engineering department at the University of Melbourne where he directs the "EMULab," an Ericsson funded networking research group. His research interests include scaling models of packet traffic, parameter estimation problems and queueing theory for scaling processes, and the statistical and dynamic nature of Internet traffic.



Walter Willinger (Member, IEEE) received the Diplom (Dipl. Math.) from the ETH Zurich, Switzerland, and the M.S. and Ph.D. degrees from the School of ORIE, Cornell University, Ithaca, NY.

He is currently a member of the Information Sciences Research Center at AT&T Labs - Research, Florham Park, NJ. Before that, he was a Member of Technical Staff at Bellcore (1986–1996). He has been a leader of the work on the self-similar ("fractal") nature of data network

traffic. His research interests focus on providing a mathematical foundation for understanding and engineering large-scale complex communication networks such as the Internet.

Dr. Willinger is the co-recipient of the 1996 IEEE W.R.G. Baker Prize Award and the 1994 W.R. Bennett Prize Paper Award from the IEEE Communications Society for the paper titled "On the Self-Similar Nature of Ethernet Traffic."