Dynamics of IP traffic: A study of the role of variability and the impact of control*

Anja Feldmann Anna C. Gilbert — AT&T Labs–Research, Florham Park, NJ

Polly Huang — USC/ISI, Los Angeles, CA

Walter Willinger — AT&T Labs–Research, Florham Park, NJ

{anja,agilbert,walter}@research.att.com, huang@catarina.usc.edu

Abstract

Using the *ns-2*-simulator to experiment with different aspects of user- or session-behaviors and network configurations and focusing on the qualitative aspects of a wavelet-based scaling analysis, we present a systematic investigation into how and why variability and feedback-control contribute to the intriguing scaling properties observed in actual Internet traces (as our benchmark data, we use measured Internet traffic from an ISP). We illustrate how variability of both user aspects and network environments (i) causes self-similar scaling behavior over large time scales, (ii) determines a more or less pronounced change in scaling behavior around a specific time scale, and (iii) sets the stage for the emergence of surprisingly rich scaling dynamics over small time scales; i.e., multifractal scaling. Moreover, our scaling analyses indicate whether or not open-loop controls such as UDP or closed-loop controls such as TCP impact the local or small-scale behavior of the traffic and how they contribute to the observed multifractal nature of measured Internet traffic. In fact, our findings suggest an initial physical explanation for why measured Internet traffic over small time scales is highly complex and suggest novel ways for detecting and identifying, for example, performance bottlenecks.

This paper focuses on the qualitative aspects of a wavelet-based scaling analysis rather than on the quantitative use for which it was originally designed. We demonstrate how the presented techniques can be used for analyzing a wide range of different kinds of network-related measurements in ways that were not previously feasible. We show that scaling analysis has the ability to extract relevant information about the time-scale dynamics of Internet traffic, thereby, we hope, making these techniques available to a larger segment of the networking research community.

1 Introduction

This paper provides new insights into the question "What aspects of user and network behaviors contribute to what characteristics of the dynamics in measured IP traffic?" by reproducing with a number of well-designed and fully-controlled network simulations a variety of scaling phenomena observed in measured ISP traffic. In particular, our empirical studies clarify what is meant by statements of the form "Self-similar scaling behavior over large time scales is mainly caused by user/session characteristics and has little to do with network-specific aspects" (e.g., see [8]; for related earlier work, see also [19]). In support of yet another conjecture that can be found in [8], we also present empirical evidence demonstrating that time scales on the order of a "typical" round-trip time within the network are directly related to a rather abrupt transition from self-similar scaling to a more complex scaling behavior; i.e., multifractal scaling. Finally, through experiments with the different components of a full-blown TCP implementation, we partly demystify the occurrence of this highly complex scaling behavior of measured Internet traffic over small time scales by reducing it to and pointing out a plausible explanation in terms of previously observed phenomena in the dynamics of TCP-type congestion control, among them ACK-compression; see for example [12, 27, 26, 32, 17], or the more recent study [20]. This empirical observation begs for a simple mathematical construction that incorporates the essence of flow control phenomena and leads to multifractal scaling behavior. Unfortunately, we have not yet succeeded in this endeavor and at this stage, referring to the observed fine-time scaling behavior of IP traffic as "highly complex" or "multifractal" makes little difference. However, we believe that aiming for an intuitive and rigorous physical explanation in the networking context of the mathematical concept of multifractals will shed new light on features of realistic IP networks that have largely gone unnoticed in the past. Succeeding in this endeavor would offer the attractive alternative of being able to avoid the notion of multifractals all together because the concept could be explained in genuine networking terms.

As a by-product of our empirical investigations into the dynamics of IP traffic, we present and advertise in this paper a class of wavelet-based scaling techniques and illustrate how their ability to localize a set of network measurements in time and scale enables one to uncover relevant information about the time-scale dynamics of network traffic. By doing so, we hope to make these techniques more readily available to a larger segment of the networking research community, thus drawing attention to the potential that these techniques have for analyzing network measurements in ways that are novel and that had not been feasible previously. In particular, we illustrate throughout this paper that these wavelet techniques are highly effective in identifying and extracting regular patterns in a way that cannot be easily accomplished with Fourier-based techniques. In combination with the ubiquitous nature of the observed scaling properties in network measurements, the wavelets' natural abilities to detect scaling behavior have made wavelet-based analysis the method-of-choice for studying various types of network measurements and for understanding some of their most useful

* A. C. Gilbert's research was partially supported by the NSF Grant DMS-9705665 at Yale University and AT&T Labs-Research, Florham Park, NJ; P. Huang's research was supported by the DARPA Grant DABT63-96-C-0054 at USC, Los Angeles, LA; and W. Willinger's research was partially supported by the NSF Grants NCR-9628067 at the University of California at Santa Cruz and ANI-9805623 at Boston University.

and relevant characteristics. We show in this paper how to interpret the results from such a time- and scale-localization approach and demonstrate how to relate the findings to underlying networking configurations and/or predominant user characteristics. To this end, we emphasize and exploit primarily the networking context in which these techniques are applied and focus less on the mathematical and statistical aspects and features of a time-scale analysis (e.g., multifractal formalism, estimation of multifractal spectra). Consequently, our findings are qualitative rather than quantitative in nature; that is, as far as, for example, self-similar scaling is concerned, we are mainly interested in using the proposed scaling techniques for the purpose of detecting self-similar scaling behavior over large time scales, which turns out to be a property that is highly robust under a variety of changes in the underlying network configuration. Our work relies on a set of measured traffic traces from an ISP environment and on various traces collected from a simulation environment that uses the *ns-2*-simulator [3] and exploits its ability to implement different network configurations. The ISP traces serve as benchmarks and are used for reality checks, while the *ns-2*-generated traces allow us to identify the effects that different aspects of user/session characteristics or network configurations can have on the dynamics of network traffic. Our analysis techniques, the measured IP traffic traces, and the simulation environments are described in detail in Section 2.

Over the last few years, network-related measurements have become a rich source for observing interesting and at times surprising scaling behaviors; e.g., self-similar scaling [15, 21, 9] and multifractal scaling [24, 8]. Intuitively, the ubiquity with which some of these scaling phenomena occur in measurements from today's IP networks is related to the absence of an intrinsic scale wherever one looks: link speeds span an ever increasing range of scales (from Kbps modem access to Gbps optical fiber connections), as do latencies (on the order of microseconds for fiber optic links to seconds for satellite links), and packet round-trip times. At the same time, in today's Web-dominated Internet, the sizes or durations of sessions, number of HTTP requests/responses, TCP connections or IP-flows typically span up to six orders of magnitude (e.g., see [5, 29, 9]). Mathematically, the absence of an intrinsic scale is equivalent to high variability and can be captured in a parsimonious manner using heavy-tailed (also known as scale-invariant) distributions with infinite variance. Thus, one of the main objectives of this paper is to present a coherent picture of how this kind of variability of user/session- and network-related behaviors impacts the time-scale dynamics of network traffic. In particular, we identify in Section 3 those aspects of packet traffic that are affected by the absence of an intrinsic scale for certain user- and network configuration-related features; these aspects cover the self-similar scaling property of IP traffic over large time scales and the location of the (lower) cutoff scale(s) beyond which self-similarity ceases to exist and gives way to a richer and more complex scaling structure. In this sense, our studies suggest a clean separation between user- or session-related aspects and network-related features, at least as far as the physical explanations of self-similar and multifractal scaling are concerned.

Another major focus of our studies is to highlight the role that closed-loop flow control plays in providing a better understanding of the observed highly complex scaling behavior of IP traffic over small time scales [8]. To this end, we provide in Section 4 empirical evidence that TCP-like flow control in a heterogeneous network environment gives rise to actual packet flow patterns that exhibit a surprisingly rich mathematical structure consistent with multifractals. In contrast, open-loop controls such as UDP give rise to traffic patterns that are essentially smooth (i.e., regular) and lack significant local scaling behavior. By experimenting with the various components of a full-blown TCP-implementation, we can further clarify the contributions of congestion control and reliable transfer to the multifractal nature of Internet traffic. In this sense, our findings offer an initial physical explanation for the observed multifractal scaling property of measured IP traffic. In addition, our findings relate the observed multifractal scaling to a pronounced clustering effect of the packets belonging to individual TCP connections, which in turn is caused by the highly bursty dynamics of ACK packets (a well-known phenomenon called *ACK compression*). Our findings thus confirm an earlier conjecture made in [8], namely that a likely physical explanation for why measured Internet traffic over small time scales is highly complex will require a more detailed understanding of the TCP mechanism in a non-trivial networking environment. Moreover, by relating multifractal scaling to the physics of TCP (e.g., slow start, congestion control, retransmission, ACK compression), we have gained access to a substantial body of knowledge about various aspects of the dynamics of congestion control mechanisms; for example, see [13, 12, 25, 26, 32] and the empirical studies of Internet traffic dynamics [17, 20]. As a result, we believe to have set the stage for a physical explanation and understanding of the multifractal scaling phenomenon of measured IP traffic over small time scales that may be as plausible, intuitive, appealing and relevant as the one that has recently been found for the self-similar scaling (e.g., see [30, 29, 9]). This and other open problems, together with some practical applications of our scaling analysis and some limitations of our study and of the underlying network configurations are discussed in Section 5.

2 Towards a scaling analysis for network measurements

The special appeal for using wavelet methods for analyzing and understanding network-related measurements is that (i) wavelets are a natural mathematical tool for detecting, identifying and exploiting scaling phenomena, and (ii) scaling phenomena appear to be a dominant feature in variety of measurements from modern communication networks. In this section, we introduce and describe a set of wavelet-based scaling analysis techniques and show with a few toy examples and measured traffic traces from an ISP environment some of their most basic abilities for interpreting scaling-related characteristics and deviations. The last subsection contains a description of the simulation engine that we use throughout the rest of the paper for our empirical studies.

2.1 Description of scaling analysis techniques

Consider a time series $X_{n,k}$, $k = 0, 1, 2, \ldots$, at the finest level of resolution 2^{-n} (or the finest scale n). This might represent the number of packets per 1 msec, for example. We coarsen X_n by averaging (with a slightly unusual normalization factor) over non-overlapping blocks of size two

$$X_{n-1,k} = \frac{1}{\sqrt{2}} \left(X_{n,2k} + X_{n,2k+1} \right) \tag{1}$$

and obtain the time series X_{n-1}, a coarser resolution picture of the original series X_n. The difference between these two pictures is

$$D_{n-1,k} = \frac{1}{\sqrt{2}} \left(X_{n,2k} - X_{n,2k+1} \right). \tag{2}$$

We can write the original time series X_n as the sum of the "blurrier" series X_{n-1} and the difference D_{n-1}, $X_n = 2^{-1/2}(X_{n-1} + D_{n-1})$. We can repeat this process (i.e., write X_{n-1} as the sum of yet another average X_{n-2} and the difference D_{n-2}, and iterate) for as many scales as are present in the original time series $X_n = 2^{-n/2} X_0 + 2^{-n/2} D_0 + \cdots + 2^{-1/2} D_{n-1}$. We refer to the collection of differences $D_{j,k}$ as the *discrete (Haar) wavelet coefficients* $d_{j,k}$, they make up what is commonly referred to as the *discrete wavelet transform*, and they may be calculated iteratively using Eqns. (1–2)[1].

We use the wavelet transform of a time series to study both its global and local scaling properties. We begin with the global properties, by which we mean the statistics of the time series viewed at each resolution level or scale, taken as a function of scale. In particular, we examine the average energy contained in each scale of the trace and examine how that quantity changes as we move from coarser to finer scales. The average energy at scale j is the average of the sum of the squared wavelet coefficients $|d_{j,k}|^2$; i.e.,

$$E_j = \frac{1}{N_j} \sum_k |d_{j,k}|^2,$$

where N_j is the number of coefficients at scale j. To determine the global scaling property of the data, we plot $\log(E_j)$ as a function of scale j, from coarsest to finest scales, and determine qualitatively over what range of scales there exists a linear relationship between $\log(E_j)$ and scale j; that is, over what range of time scales there exists self-similar scaling (see [1] for more details). In all of the figures in this paper, the scale j is on the bottom axis and the corresponding time (in seconds) is plotted on the top axis for reference.

The local scaling analysis is slightly more complicated than the global analysis in that we wish to gather information about local features (e.g., bursts of packets) rather than statistics about the time series viewed as a whole at each scale. In particular, we want to know how the number of packets in an interval of size ϵ about time t_0 is related to the size ϵ of the interval. The "stronger" and more concentrated the burst around time t_0, the less the number of packets depends on the size of the interval. The strength of the spike around time t_0 or the degree of "local burstiness" can be captured mathematically by a (possibly) time-dependent scaling exponent, and the goal of a local scaling analysis is to collect characteristic information about the strengths and locations of the various scaling exponents (for a more detailed presentation see, e.g. [8] or [2]). To gather the statistics of the local scaling exponents, we again use the discrete wavelet transform of the underlying data and define the partition function $S(q,j)$ as the sum over the local maxima of the (normalized) wavelet coefficients raised to the qth power at each scale j:

$$S(q,j) = \sum_{\max} |2^{-j/2} d_{j,k}|^q$$

(see [8, 11] and the references therein)[2]. Although we need several additional transformations of the partition function to quantify rigorously the distribution of scaling exponents, for an intuitive picture and the sort of qualitative interpretation of local scaling behavior considered in this paper, we rely upon the graphical features of the partition function to detect and assess the local scaling behavior of the data. To that end, our local scaling analysis consists of plotting, for each value of q, $\log S(q,j)$ as a function of scale j (from coarsest to finest) on one graph (throughout the paper, the local scaling plots will typically show the curves corresponding to $q = 0, 2, 4, \ldots, 18, 20$, with $S(0,j)$ being the straight reference line). This way we obtain a family of curves, and determining qualitatively, for a range of the smaller q-values, if there exists a more or less linear relationship between $\log S(q,j)$ and scale j over a range of the finest scales provides information about the nature of local scaling. In particular, "interesting" local scaling (i.e., scaling consistent with multifractal behavior) manifests itself, for a range of small-to-medium q's, in a linear relationship between $\log S(q,j)$ and scale j that extends over a range of fine time scales, where the slope of the linear regime depends in a non-linear fashion on q. The essence of multifractal analysis is to determine for a given data set how to infer these slopes and whether or not they depend in a non-linear manner on q. A linear dependence on q suggests a less interesting or "monofractal" local scaling where essentially one exponent characterizes the entire scaling behavior (e.g., self-similar processes with self-similarity parameter $0 < H < 1$).

[1] We use the Haar wavelets primarily for exposition and we use more general wavelets (e.g., compactly supported Daubechies wavelets [6]) for the scaling analysis.

[2] In practice, we slide a window of length five (this parameter can be varied) over the coefficients at each scale, extracting the local maxima.

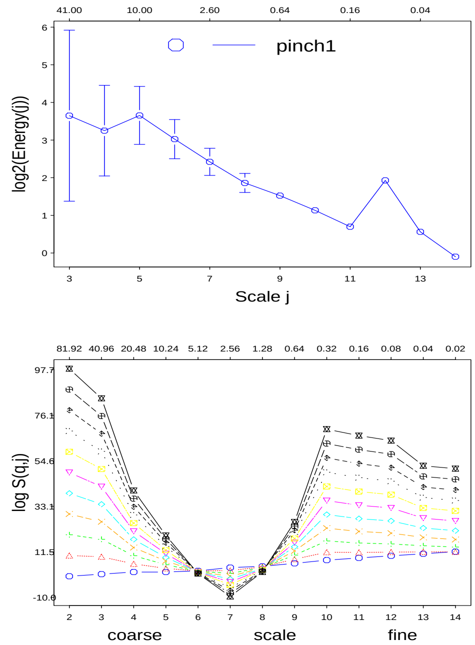

Figure 1: Global (top) and local (bottom) scaling analysis for the toy examples "pinch" (top) and "fold" (bottom).

2.2 Detecting and identifying scaling behavior

Rather than focusing on the quantitative aspects of self-similarity estimation using wavelet-based techniques (as, for example, emphasized in [1]), we suggest a more qualitative usage of the above-mentioned scaling techniques. To this end, we show with simple toy examples how, by manipulating certain aspects of a time series, the global scaling behavior can be changed in quite drastic ways. Starting with an exactly self-similar trace, we modify the wavelet coefficients at a fixed scale, e.g., scale $j = 12$, by adding to each coefficient at that scale a fixed multiple of $\max_k\{|d_{j,k}|\}$. The global scaling plot for the resulting toy example time series "pinch" is given in Figure 1 (top) and shows a pronounced "spike" at scale 12. "Dips" can be introduced in a similar way, and by explicitly manipulating the wavelet coefficients over a range of different scales, the "spikes" and "dips" can be made wider or narrower. Clearly, data that exhibit such features will give rise to intricate global scaling plots, and blindly applying quantitative methods (e.g., wavelet-based estimation of the self-similarity parameter) in these situations can easily lead to wrong conclusions.

To demonstrate how certain features in the data can have an impact on the local scaling analysis, we begin with a time series derived from a conservative cascade with fixed generator W (we take W to be a truncated normal on $[0,1]$ with mean $1/2$ and variance 0.01; for details, see [8, 11, 23]). From this sequence, we construct the toy example time series "fold" by targeting a selected range of scales and replacing the wavelet coefficients at those scales by appropriately chosen fixed (possibly scale-dependent) quantities. From the corresponding local scaling plot shown in Figure 1 (bottom) we see that adding in this way periodic components at the specified scales causes a pronounced dip, while removing selectively variability by inserting smaller coefficients gives rise to the apparent "folding over;" that is, the structure function plots become negative for some scales, indicating that the corresponding wavelet coefficients have become very small. As in the case of the global scaling analysis, even a qualitative assessment of multifractality of a data set with such distinguished features becomes a non-trivial task and requires extreme care.

While these examples are simplifications of real-life situations, they do highlight the effects that adding certain types of local periodicities can have on the scaling behavior of data. Moreover, note that such disturbances often cannot be easily identified using Fourier-type transforms of actual data, but they can be detected by extracting the wavelet coefficients at the desired scale(s). However, with these toy examples, such periodicities are readily apparent in the Fourier spectrum.

2.3 Measured Internet traffic dynamics

Next we illustrate the global and local scaling analysis techniques described earlier with two data sets of measured IP traffic from an ISP environment (see Appendix for a detailed description of the data). The results serve mainly as benchmarks and reality checks for our simulation work described in the rest of the paper. However, they also show that even though the measurements were taken about one year apart, their statistical characteristics as far as the global and local scaling are concerned are quite similar. This observation supports earlier conjectures about self-similar large time scaling and multifractal scaling over small time scales representing two invariants for Internet traffic; that is, characteristics of the dynamic nature of IP packet traffic that are robust under a wide range of possible networking- and application-related changes.

On the one hand, the right plot in Figure 6 which depicts the global scaling behaviors for DIAL1 and DIAL2 shows that both data sets exhibit very similar global scaling behavior (i.e., self-similar scaling over time scales larger than a few hundreds of milliseconds – look for approximately linear behavior on the left half of the plot, for scales 1–10; emergence of a different regime for scales 11–18). On the other hand, looking at the right plot in Figure 7 which shows the local scaling analysis for DIAL2 measured at the 1 msec scale, we observe non-trivial local scaling behavior over small time scales (i.e., over time scales on the order of a few hundred milliseconds and below) which, upon further investigations, can be shown to be consistent with multifractal scaling (look for approximately linear behavior of the partition function plots for scales 15–19 or, by "cutting across the spike" at scale 14, for scales 12–19). Similar results (not shown here) apply for the data set DIAL1.

Finally, to hint at things to come, we show in Figure 2 the results of a local scaling analysis for three subsets of the trace DIAL2. The subsets represent traffic that is transmitted between three different networks and the ISP clients and can be obtained using IP-header-information. (Two IP addresses are considered to belong to the same network if they have the same high-order 16 bit IP addresses.) NET1 turns out to consist mostly of traffic between the ISP clients and the ISP web servers, NET2 is traffic between the ISP and major news servers, and NET3 consists mainly of realaudio UDP traffic. The observed differences in the corresponding local scaling plots are telling. There is a folding-over in the local scaling behavior, similar in nature to the toy example "fold" shown at the bottom of Figure 1, and it becomes more pronounced as we move from NET1 to NET2: There is more RTT variability in NET2 than in NET1 because packets have to travel across ISP boundaries, and the traffic patterns show higher regularities (i.e., small wavelet coefficients, and hence small values of $\log S(q, j)$) over a substantially wider range of the medium time scales. In contrast, the local scaling plot for NET3 shows trivial structure: Much in the spirit of realaudio UDP, packets are essentially sent at constant rate, with a periodicity on the order of about 40 msec (i.e., all partition functions coincide roughly at scale $j = 15$, they are all roughly linear, and their slopes are approximately linear in q). These observations give an indication that local scaling analysis is capable of performing "detective" work in identifying and explaining which aspects of network behavior contribute to what features observed in the measured traces.

2.4 Using *ns-2* to replicate realistic IP traffic dynamics

The simulation engine used throughout this study is *ns-2* (Network Simulator version 2) [3]. This discrete event simulator provides a rich library of modules such as different flavors of TCP, scheduling algorithms, routing mechanism, and trace collection support.

Using the measured ISP traces as benchmarks and road map for the experimental studies described below, our choices of network topologies and types of clients are basically determined by attempting to replicate a reasonably realistic ISP environment. Since roughly 60-80% of all packets and bytes measured in our ISP environment are Web-traffic, our primary user is a consumer accessing the network through an ISP via a modem bank to browse the Web. To accurately simulate HTTP transfers, we extend the existing *ns-2* HTTP modules to accommodate for the variability that is inherent in the Web.

In a typical HTTP 1.0 transaction, a web client sends a request to the Web server for a web object after establishing a TCP connection. The server responds with a reply header (sometimes attaching data) and then continues to send data. However, the original *ns-2* TCP connection module failed to send the connection set-up and tear-down packets. In fact, the TCP connection modules allow the transfer of data in only one direction. To circumvent this problem, we emulated the exchange of the HTTP header information with two *ns-2* TCP connections that have the same "port" numbers which facilitates object identification[3]. During a Web session a user usually requests several Web pages and each page may contain several web objects (e.g. `jpg` images or `au` files). To capture this hierarchical structure and its inherent variability, we allow for different probability distributions for the following user/session attributes: inter-session time, pages per session, inter-page time, objects per page, inter-object time, and object size (in KB). For each of these distributions, we can choose from the many built-in distributions (such as constant, uniform, exponential, Pareto, etc.) or we may define our own. Details about the parameters required for these distributions and used in our studies can be found in the Appendix. We base our choice of distributions (including the specific parameters) on the work surrounding SURGE [4], a Web workload generator designed to generate realistic Web traffic patterns, and upon [7, 18]. Note that we simulate HTTP without pipelining and without persistent connections.

The protocol stack, network topology (including delays and bandwidths), and the sequence of Web requests define a simulation. Since TCP Reno and HTTP 1.0 [28] are assumed to be the predominant protocols in the ISP environment at hand, we emulate them in our simulations. We vary the number of sessions from 100 (low load scenarios) to 300 or 400 for high load scenarios. Each session consists of a fixed number (300) of Web pages. This ensures that for almost all simulations all sessions are active for the duration of the simulation (4200 seconds). We discard an initial segment of each simulation run during which we randomly activate all sessions.

As far as choosing a network topology is concerned, we are again motivated by the ISP environment where we obtained our measurements. To find out how various attributes of network topology and web request sequence affect the traffic characteristics, we

[3]The latest releases of *ns-2* support two-way TCP with detailed connection establishment and teardown.

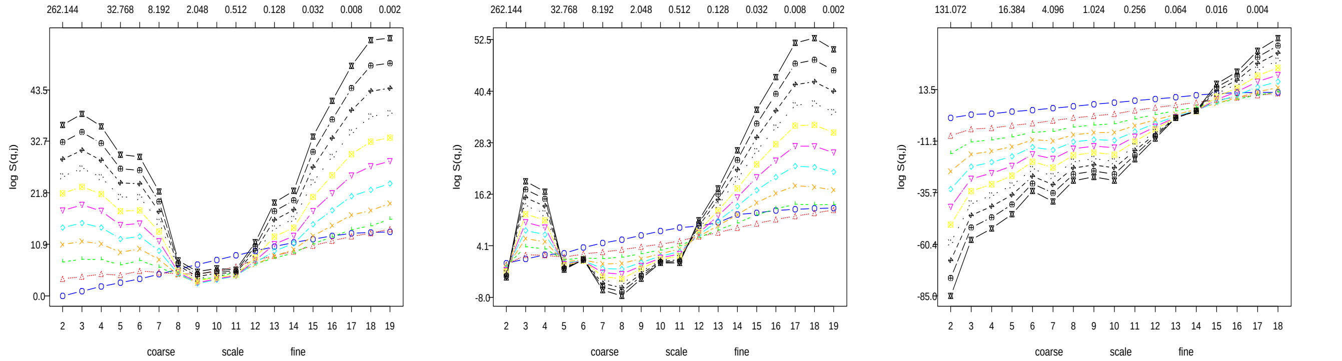

Figure 2: Local scaling analysis plots for three subsets of trace DIAL2: NET1 (left, ISP Web server traffic), NET2 (middle, traffic to news server), NET3 (right, realaudio UDP traffic).

experiment with a set of network topologies. We concentrate on simulation environments that consist of a set of clients connected to an access network which in turn provides connectivity to a set of servers, in effect creating a "dumbbell." We map each user session to a single client node (either home computer connected via slow speed modems or office workstation connected via switched Ethernet), referred to as MODEM NODES or HIGH-SPEED NODES respectively. This means that the simplest architecture is one where a set of client nodes is connected to an access node that is connected to a single Web server Figure 3 (top). To understand the impact of congestion on the middle link, we split it into two separate links that can have different delays and bandwidths. If no bottleneck is introduced we refer to the topology as CAPBELL (unlimited capacity). If the link labeled **A** in the top plot in Figure 3 has lower capacity than the link labeled **B**, we refer to the topology as SINGLEBELL. To increase the variability of the delays and/or bandwidths to different servers, we expand the single servers into a set of servers as shown in the bottom plot in Figure 3 and refer to the topology as FLEXBELL. To experiment with cross traffic, a set of clients and servers can be added to either the link labeled **A** in Figure 3 (bottom) or the links labeled **B** and **C**. This topology is called CROSSBELL. To ensure that the modem clients are not the bottleneck links, the buffers in the queues on those links are configured to have sufficient space.

3 On the role of variability

In this section we explore the role variability in its many forms plays in determining the scaling properties of network traffic. We divide the types of variability into two main categories: user- or session-centered variability (e.g., sizes of Web sessions or sizes of HTTP data transfers, number of requests per session) and network-related variability (including delays, bandwidths, and topology). To understand the scaling behavior observed in the measured data, we hold all but one of the above forms of variability fixed and explore the effects of the remaining element of variability. In doing so, we sometimes simulate artificially simple networks; nevertheless, we are able to find clear "fingerprints" in the measurements that are caused by the different aspects of variability. We start with user/session variability and its effects on the scaling properties of the time series of packet counts. Then we examine how network variability impacts the scaling behavior of traffic.

3.1 User- and session-related variability

One of the least complex forms of variability is that of the users and their sessions. It is expressed in terms of the distributions of the number of objects per page, the number of packets per object, the interarrival times of pages, etc. By high user variability, we mean that at least one of the "workload-specific" distributions (i.e., number of objects per page or number of packets per object) must be chosen from the class of heavy-tailed distributions with infinite variance (e.g., Pareto-type tail behavior), while low user variability reflects the fact that all these distributions are either exponential or trivial (i.e., constant).

We use the CAPBELL configuration with its essentially unlimited bandwidth constraints and with 1 msec link delays to illustrate the difference between how low user variability and high user vari-

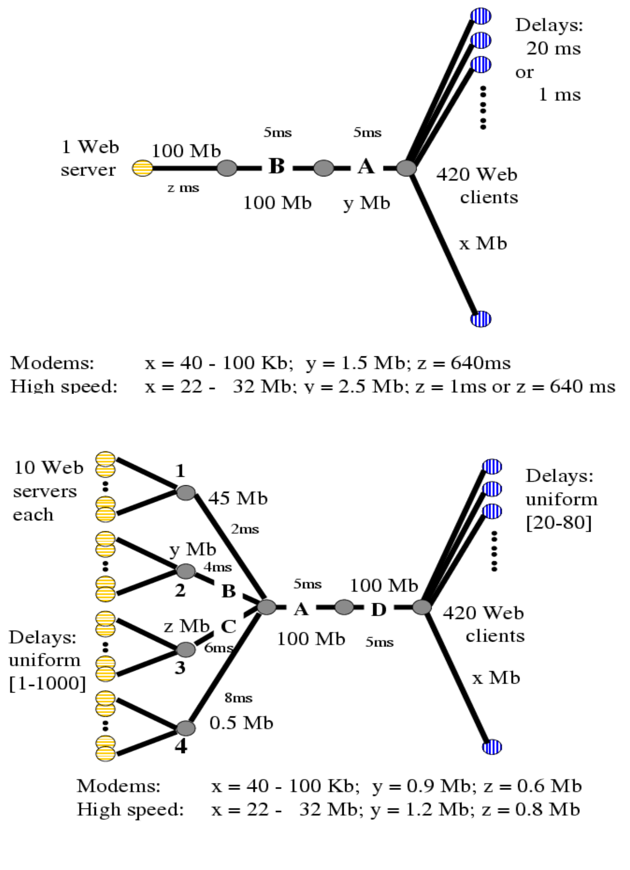

Figure 3: Network topologies: CAPBELL/SINGLEBELL (top), FLEXBELL (bottom); Mb=Mbps, Kb=Kbps, ms=millisecond.

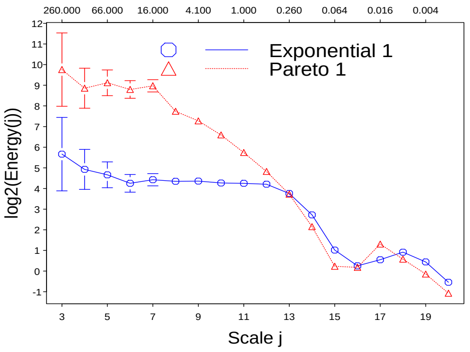

Figure 4: Impact of user variability on global scaling analysis: low user variability (CAPBELL, 400 HIGH-SPEED NODES, EXPONENTIAL 1); high user variability (CAPBELL, HIGH-SPEED NODES, PARETO 1).

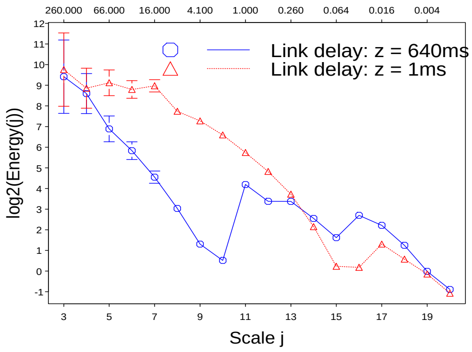

Figure 5: Impact of delay variability on global scaling: short delays (CAPBELL, 400 HIGH-SPEED NODES, PARETO 1); long delays (CAPBELL, HIGH-SPEED NODES, PARETO 1).

ability contribute to the dynamics of the measured traffic. We consider the environment with high-speed access nodes and perform a global scaling analysis of the time series representing the number of packets per 1 msec recorded at link **A**. The results are shown in Figure 4[4]. While the low user variability simulation yields a trivial global scaling plot (i.e., horizontal line, consistent with the absence of long-range dependence), the high user variability setting gives rise to a pronounced global scaling behavior over large time scales; that is, the packet counts exhibit long-range dependence and the traffic is asymptotically self-similar. Of course, these empirical results are in full agreement with the rigorous physical explanation of the self-similar scaling of network traffic over large time scales in terms of the infinite variance or high-variability property of user session sizes (for details, e.g. see [21, 9]). In this sense, global scaling plots such as the ones shown in Figure 4 (see also below for global scaling plots where we explicitly change a variety of network-related features, without any significant effect on the large time scaling behavior) illustrate what is meant by saying that "self-similar scaling over large time scales is primarily caused by user/session characteristics and has little to do with network-specific aspects" [8] (for earlier related findings, see [19]).

3.2 Network-related variability: Delays

In the above discussion of the results of our global scaling analysis, we focused solely on the large time scale features and checked whether or not there exists a more or less linear relationship in Figure 4, and if so, whether or not the slope is zero or strictly negative. In particular, we ignored two other prominent features in those global scaling plots: an apparent departure from linearity at some more or less pronounced medium-to-small time scale, and the emergence of some structure other than self-similarity below that scale. In this subsection, we identify a variability aspect that is not user- or session-related but network-specific and that is primarily responsible for the observed departure from self-similar scaling at some specific time scale. The question about what structure emerges when considering time scales below (i.e., to the right of) that specific time scale will be discussed later in this section.

Using the same high-access CAPBELL configuration as before, the only network-related aspect that we change is link delay, which of course impacts the round-trip time (RTT) behavior of the packets sent over the network. More specifically, we consider CAPBELL with a low link delay of $z = 1$ msec (resulting in a packet RTT of 24 msec) and compare it with a high link delay of $z = 640$ msec (i.e., RTT= 1.3 sec). In both situations, we collect the time series of number of packets per 1 msec at link labeled **A** and compute the global scaling plots shown in Figure 5. Notice that the time scale where self-similar scaling breaks down is the smallest scale (scale 10 = 2.0 seconds) that is larger than the data packet RTT in the respective networks[5]. Moreover, the type of breakdown of self-similar scaling (i.e., a pronounced "dip" at time scale 10) matches that of the toy example presented in Section 2.2, for the simple reason that the RTT behavior in this network configuration identifies a distinct and significant periodic component in the traffic. Also note that the additional dips at finer time scales can be attributed to the presence of periodic components caused by, for example, the time it takes for an ACK packet to travel to a client and the TCP packet released by the server to get to the monitored link, or for a TCP data packet to travel to a client and the corresponding ACK to return to the monitored link.

3.3 Network-related variability: Congestion I

We have seen that the CAPBELL environment imposes a rigid RTT behavior that has essentially the same effect on the global scaling plot as manipulating the wavelet coefficients at the time scale corresponding to the packet RTT in the network to introduce a pronounced periodic component in the resulting packet trace (see Section 2.2). To illustrate the effect of adding variability to the RTT behavior, we use the high-access SINGLEBELL configuration which is identical to the above CAPBELL environment except that the capacity on the middle link A has been decreased to $y = 2.5$ Mbps. Keeping the same number of user sessions, we introduce in this way a single bottleneck and create congestion (resulting in a loss rate of 3.85%). The resulting global scaling plot (not shown here) differs in three ways from the global scaling plot for the corresponding CAPBELL scenario. First the self-similar scaling breaks down earlier (i.e., coarser scale) than in the non-bottleneck scenario; second, the transition from self-similar scaling is smoother (i.e., pronounced dips essentially disappear, or are "smoothed out" over a range of time scales) than in the non-congested environment; and third, the energy in the trace is substantially smaller at each scale than in the CAPBELL configuration (i.e., smaller wavelet coefficients throughout, resulting in a scaling plot that lies below its counterpart, except for the finest scale). The first two observations are a result of the higher variability in RTT due to the presence of congestion, while the third feature simply reflects a decrease of variability in the overall trace (i.e., filling up the link leaves little

[4] The convention used throughout the paper is to indicate the simulation environment associated with each plot by given the triple (CONFIGURATION, LOAD, WORKLOAD SCENARIO); for WORKLOAD SCENARIO refer to Appendix A.2.

[5] Note that if $2^{-j-1} < \text{RTT} \leq 2^{-j}$, the break down will occur at scale j because there every wavelet coefficient includes at least one packet from the added periodic component (and hence, these coefficients are less variable than those at scale $j-1$, some of which do not include packets from the added periodic component).

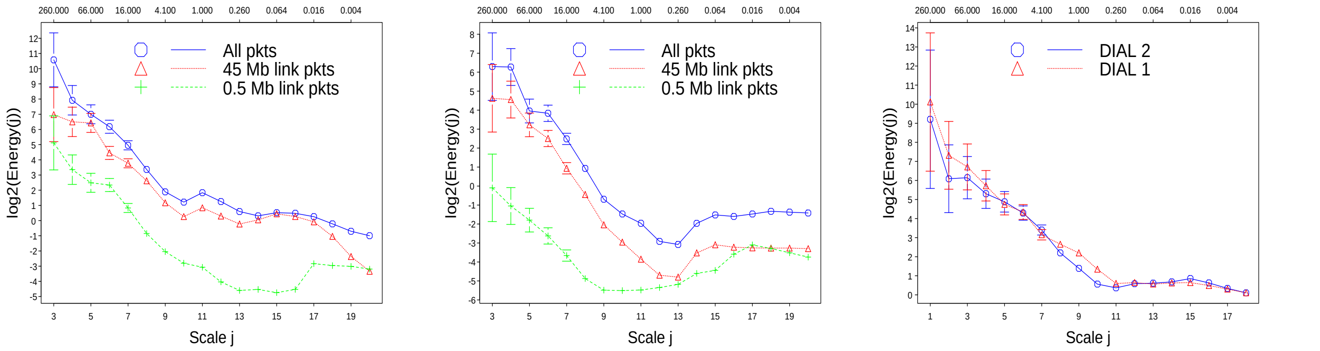

Figure 6: Impact of network bottleneck and load on global scaling; left (FLEXBELL, 400 HIGH-SPEED NODES, PARETO 1) generating losses of 0.46%; middle (FLEXBELL, 400 MODEM NODES, PARETO 2) generating a loss rate of 6.4%; right (global scaling plots for measured Internet traces DIAL1 and DIAL2, for comparison).

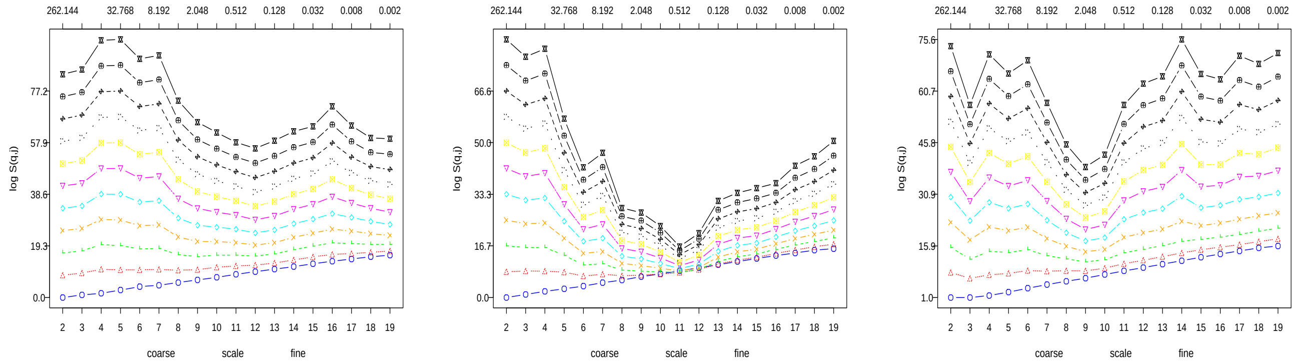

Figure 7: Local scaling analysis: left (FLEXBELL, 400 HIGH-SPEED NODES, PARETO 2); middle (FLEXBELL, 400 MODEM NODES, PARETO 2); right (local scaling plot for measured IP traffic trace DIAL2, for comparison).

room for even modest traffic fluctuations).

Another way to drive a network into congestion thus introducing more variability to the RTT behavior is by adding load. To this end, we consider the SINGLEBELL configuration with low-speed modem clients and run the simulation with 100 user sessions (low load) and 400 user sessions (high load), respectively. Simply by comparing the observed loss rates (i.e., 0.002% in the low load scenario, 8.72% in the high load scenario), we see that the 100 user case is essentially congestion-free while the 400 user case experiences significant congestion. The resulting global scaling plots are essentially the same as before, with one noticeable exception. The energy in the congested trace over the fine time scales is appreciably larger than in the non-congested case. This is a first indication that the more congested the link is (i.e., the more packets are lost) the larger the wavelet coefficients at time scales below the "typical" RTT will be. We will return to this issue shortly.

Finally, to add yet another component to RTT variability, we run the FLEXBELL configuration to identify the effect that the presence of different bottleneck links (possibly with different delays) within one and the same network has on the dynamics of the traffic. In Figure 6 we display the results of our global scaling analysis for the 400 nodes high-speed access case (left plot) and for the 400 nodes modem access case (middle plot). In each case, we show the global scaling plots for the aggregate traffic and for the traffic measured over 45 Mbps and 0.5 Mbps bottleneck links, respectively; i.e., over the links to nodes labelled 1 and 4 in the bottom plot of Figure 3, respectively. As we can see, the global scaling of the aggregate traffic is to a large degree determined by that of the "big" pipe (i.e., 45 Mbps link) and that the global scaling plot for the highly congested 0.5 Mbps link shows many of the features that we discussed earlier in this section in conjunction with congestion.

Note that the global scaling plot in Figure 6 (middle) of the aggregate traffic obtained from running the FLEXBELL configuration supporting 400 modem clients agrees reasonably well (slightly less energy, though, and a more pronounced dip at scales 12–13) with the global scaling plots for the measured IP traces DIAL1 and DIAL2 shown in the right plot of Figure 6. Recall that this match has been achieved without explicitly modeling any specific aspects of the underlying traces. Instead, we have relied exclusively on the physical understanding of the impacts that certain aspects of user/session- and network-related variability have on the scaling behavior of network traffic. By accounting qualitatively for the different aspects as well as for the proper "shades" of variability, we have done away with conventional statistical inference approaches and have nevertheless succeeded in roughly matching the second-order properties of the measured traces. Next we address the question whether we can do even better.

3.4 Network-related variability: Congestion II

In the previous subsection, we alluded to the observation that on a congested link, there seems to be in general more energy in the packet fluctuations at time scales below the the "typical" RTT as compared to a non- or low-congested link. We argued that this feature is due to the fact that the underlying TCP protocol is faced with more losses when there is congestion, which in turn causes the packet density fluctuations to exhibit more "interesting" local burstiness structure than when there is little congestion. In view of recent findings reported in [8], this local burstiness structure can be observed in its clearest form at the level of individual TCP connections where it has been shown to conform to multifractal scaling. At the aggregate level, multifractal scaling has been observed in a number of measured Internet traces (e.g., see [24, 16, 8]).

Given this empirical connection with multifractal scaling behavior over fine time scales, our aim in this subsection is to provide initial insights into and a first physical understanding for what aspects of variability contribute to what features of the fine time scale behavior of network traffic. Because we have shown that user- or session-related variability is almost exclusively responsible for how IP traffic behaves over large time scales but has no appreciable impact on the dynamics of network traffic over fine time scales, we consider in the following network-related aspects of variability such as delay, bottleneck links, loads, etc. To start with, Figure 7 shows the results of our local scaling analysis for the FLEXBELL configurations with 400 high-speed access clients (left) and 400 modem users (middle), respectively (these are the same configurations for which the global scaling plots are shown in Figure 6). To compare, the right plot in Figure 7 depicts the results of our local scaling analysis for the measured IP trace DIAL2. Thus, while we have seen that the global scaling of the FLEXBELL configuration with modem clients qualitatively fits that of the measured IP traces, the corresponding structure functions in Figure 7 show some obvious differences. Most prominently, we observe in the measured trace a pronounced dip around scales on the order of 1 sec, reminiscent of the toy examples discussed in Section 2.2. Also note that across the medium to small time scales in the local scaling plot of the IP trace DIAL2 (right plot), the variability in the packet density fluctuations is consistently higher than in the corresponding plots on the left in Figure 7 (e.g., compare the values of the corresponding $\log S(q, j)$-functions for scales 12 and larger).

To identify which aspect of variability inherent in the FLEXBELL configuration is primarily responsible for the observed differences in the local scaling plots, we go back to the SINGLEBELL environment with its single 2.5 Mbps bottleneck link and corresponding delay of $z = 640$ msec. We find that using both a low load and a high load scenario, we are able to replicate the pronounced dips in the low load scenario around time scales related to the "typical" RTT. At the same time, for the congested high load scenario, we observe a much wider dip due to an increased RTT variability. Moreover, the whole dip moves to the left; i.e., the packets experience in general longer RTTs, and the variability in the packet density fluctuations over time scales associated with the wide dip are diminished.

Using this understanding, we can now perform a local scaling analysis of the traffic from the FLEXBELL environment on a per-server basis; i.e., for each of the four servers, we record the packets coming from or destined for this server. Packets from the resulting traces go over the same bottleneck link and experience more or less the same amount of congestion. The local scaling plots for two of the four different traces corresponding to the four different servers are given in Figure 8 and show some familiar features. The 45 Mbps link (top plot) provides essentially unlimited capacity in this modem environment and the previous observation about a pronounced RTT behavior applies directly (a similar observation holds for the 1.2 Mbps link). In contrast, the 0.5 Mbps link (bottom plot) is highly congested, hence shows a significantly wider dip which is, in addition, located further to the left of, for example, the corresponding 0.8 Mbps-link dip; also, the variability in the packet counts over time scales associated with this wide dip is significantly smaller than in the 0.8Mbps case implying the presence of a highly regular traffic pattern over those time scales caused by a close-to full pipe. Putting it all together, we have that the local scaling plot of the full trace combines the different effects seen at the different bottleneck links. While the variability due to different RTTs reflects itself through a relatively smooth (as compared to very pronounced) dip, bottleneck-related variability shows up in terms of an appreciable amount of variability in the packet density fluctuations over time scales on the order of the location of the dip.

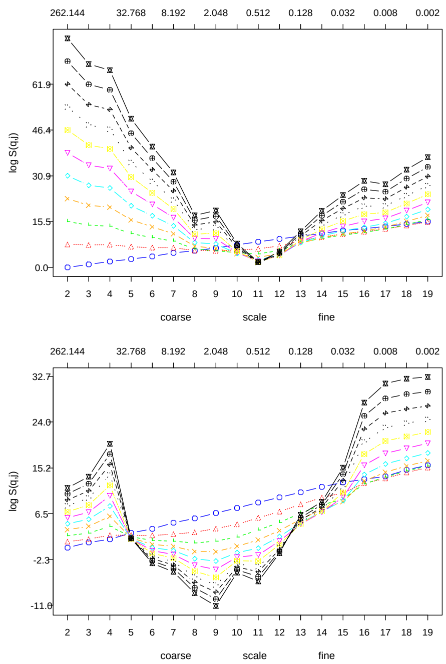

Figure 8: Impact of network load on local scaling – separating traffic according to servers. Top: Server 1 with 45 Mbps link; bottom: Server 4 with 0.5 Mbps link); (FLEXBELL, 400 MODEM NODES, PARETO 2).

However, compared to the local scaling plot of the measured trace DIAL2 (right plot in Figure 7), the corresponding local scaling analysis of the FLEXBELL configuration with 400 modem users shown in Figure 7 (middle plot) falls short of matching at least qualitatively the high variability in the packet density fluctuations across the medium to small time scales in the measured traffic as seen through the local scaling plots. Even the FLEXBELL configuration with the 400 high-speed access clients fails to match this variability over the small time scales (see left plot in Figure 7). This problem remains even if we add yet another aspect of variability; i.e., we replace the FLEXBELL with the CROSSBELL configuration, thereby introducing two-way or cross-traffic (not shown here).

To get a better understanding for how this mismatch in local scaling behavior can occur and to point out a possible approach for tackling this problem, we consider once again the FLEXBELL configuration with the 400 modem clients and focus on the traffic that traverses the 1.2Mbps link associated with server 2. We perform a local scaling analysis of the resulting trace and of its two components consisting, respectively, only of TCP data packets and only of the ACKs. The results are given in Figure 9 and comparing the left and middle plots shows that any non-trivial packet density fluctuations on this link are almost exclusively due to non-trivial fluctuations in the time series of ACK counts. In fact, over small to medium time scales, the wavelet coefficients associated with the time series of number of TCP data packets per msec (and hence the values of the structure function) are extremely small (see right plot in Figure 9), implying an essentially regular stream of TCP packets when viewed over those time scales. This should come as no surprise, though, since a close-to-saturated link is not likely to see significant traffic fluctuations. Thus, in order to increase the variability of the packet density fluctuations over the small to medium time scales of the aggregate traffic, one has to allow for significant fluctuations not only in the ACKs but also in the TCP data packet streams. To accomplish this task, a basic understanding of the interactions between the dynamics of ACK packet and TCP

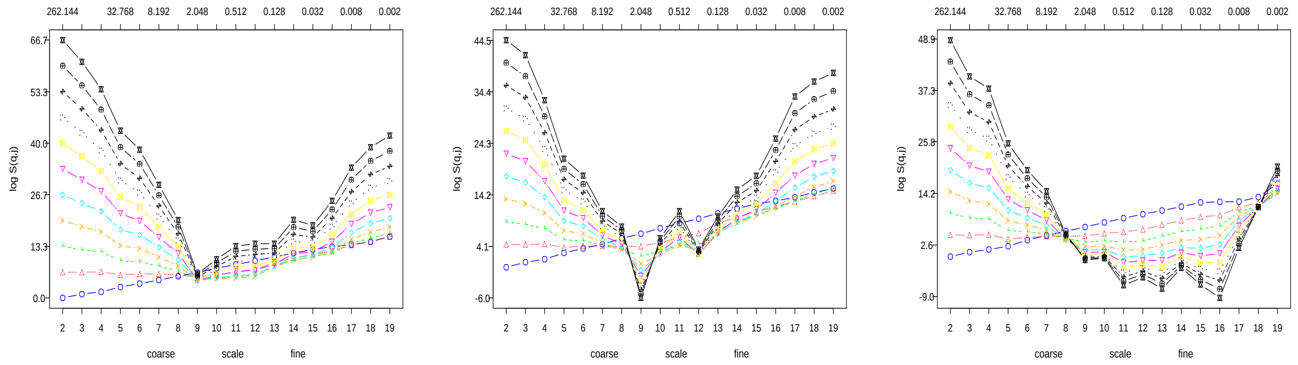

Figure 9: Local scaling analysis for traffic to and from server 2: all packets (left), ACK only (middle), TCP data only (right); (FLEXBELL, 400 MODEM NODES, PARETO 2).

data packet streams in a not-too-simplistic networking environment is needed. Thus, to gain insight into the dynamics of measured IP traffic, it is not sufficient to understand how the various aspects of user- and network-related variability impacts the traffic. It also requires knowing (at least qualitatively) which features of what protocols contribute to what aspects of measured network traffic. In the context of today's Internet and in view of the findings reported earlier in this paper, this means gaining insight into what features of TCP impact what aspects of the multifractal scaling behavior over small time scales.

4 On the impact of feedback control

Intuitively, the results of the previous section show that there is more to IP traffic dynamics than understanding the different aspects and shades of user- and network-related variability. More formally, our task is to explore how flow control impacts the local scaling behavior of measured IP traffic; i.e., the empirically observed multifractal structure over small time scales. In networking terminology, the objective of this section is to gain insight into the effects that the different components of a full-blown TCP implementation have on the nature of the local burstiness or clustering of packets of measured IP traffic. To this end, we follow a similar approach as in Section 3 and report on a set of *ns-2*-based experiments using the FLEXBELL configuration with the 400 modem users, where we systematically manipulate the various components of TCP. Although this scenario leads to appreciable loss rates on the order of 6%, we have observed similar results as the ones described below in lower load scenarios. Note that the results of our local scaling analysis for the scenario that uses "genuine" TCP is shown in the right plot in Figure 7.

4.1 Open-loop or UDP-like controls

To start our investigation of the impact of flow control on the local scaling behavior of IP traffic, we consider the extreme case of no flow control. In particular, we use UDP which injects packets into the network at a constant rate without feedback. More precisely, our modem clients use TCP to send a request to the Web server reliably in order to receive data; once the session is established, the server uses UDP to transmit the data. We investigate two scenarios which differ by how fast the Web servers are permitted to inject data into the network; in the first case, the server can send a packet every 100 milliseconds, in the second case every 10 milliseconds. The results of the local scaling analysis for the second case are depicted in the top plot in Figure 10. As expected, we observe some dipping effect because of the periodic nature with which UDP injects packets into the network. Because of significant loss rates in both settings (24.5% in the 100 msec case and 69.86% in the 10 msec case), the locations of the dips do not exactly coincide with the periodicities of 100 and 10 msec, respectively, but occur at some slightly larger time scale. In addition, the local scaling behavior over time scales 14–19 is approximately linear, with a slope that can be shown to be roughly linear in q. Similar results hold for other network configurations and load scenarios, which we take as strong indication that open-loop flow controls such as UDP have little impact on the observed fine-time scaling behavior of measured Internet traffic. In the case of large-time scaling behavior, similar findings have been reported in [19].

4.2 Closed-loop control: Stop and Wait

Moving from UDP that does not adjust its sending rate in the face of network congestion to a closed-loop control with some UDP-like flavor, we can proceed in two ways. In this subsection we consider a version of TCP that has the retransmission component enabled (i.e., reliable transfer) but uses a modification of TCP's congestion control algorithm where the window size is set to a fixed value, namely window=1. Compared to a full-blown TCP implementation, limiting the window size means more work when transmitting data, especially when the load is non-trivial (as is the case here). The losses are lower than for full TCP, simply because with a window size of 1 the source can generally not take advantage of the available bandwidth. The local scaling behavior of the resulting trace is depicted in the middle plot in Figure 10 and shows a clear dip around time scales on the order of the expected RTT, which in this case is about 1 second. More importantly, when compared to the top plot, we observe the emergence of non-trivial local scaling behavior over the smaller time scales, which can be shown to be consistent with multifractal scaling (i.e., the partition functions are approximately linear for scales 13–19, with slopes that change in a non-linear fashion as q changes from small to medium to large. Thus, even a very bare-bones implementation of TCP's windowing mechanism causes complex local clustering of packets, which demonstrates the importance of closed-loop flow control for understanding local scaling behavior in measured Internet traffic.

4.3 Closed-loop controls and reliable transfer

Instead of modifying, as we have just done, TCP's congestion control algorithm and keeping the retransmission feature of TCP intact, we can also consider a version of TCP where we keep the full congestion control component intact but where we disable TCP's retransmission feature. If this version of TCP detects a loss, e.g., by receiving multiple ACKs or by timeout, it will adjust its congestion window but will assume that the packet has been delivered successfully. In this sense, this version of TCP can “move on” even if a lot of the packets are dropped, while “genuine” TCP will have to deal with the losses. The resulting local scaling behavior is similar to the one obtained in the previous subsection, with the exception that the variability in the packet density fluctuations is somewhat reduced when we permit the full range of window sizes as compared to when we set the window size to 1. Intuitively, reduced variability means that TCP with its genuine congestion control algorithm has to work “less hard” as compared to TCP with the “broken” window=1 algorithm. Moreover, we also observe a much more pronounced dip around scales 11–12 than for the full TCP or the window=1 version. What makes this dip so pronounced in the present case is the fact that this version of TCP is not required to wait for the retransmitted packets so the impact of the timeouts is not as severe as in the full TCP case. Instead it will just advance its window. As a result, there is significantly less variability in the RTT behavior for this version of TCP as compared to, for example, full TCP with its timeouts and rules about waiting for the ACKs of the retransmitted packets.

4.4 Closed-loop controls and TCP-type congestion avoidance

Finally, we consider a version of TCP that is the more aggressive than the window=1 case. While the window=1 version of TCP is network-friendly it does not work very efficiently (cannot in general fill the pipe), by setting window=10 and leaving everything else the same as in the window=1 case (in particular, we do have retransmission in place), we deal with an aggressive version of TCP that can burst many back-to-back packets into the network and avoids many aspects of slow start and congestion avoidance. In fact, setting the congestion window size to 10 allows this version of TCP to send up to 10 packets, a full window, back to back before it has to receive any acknowledgments, thereby potentially swamping the network. In effect, in a relatively uncongested environment, ACK clocking is now done for the transmission of a whole window instead of on a packet-by-packet basis. The impact of using a version of TCP that eliminates a major component of TCP flow control, can be seen in the bottom plot in Figure 10. As expected, the ability to send back-to-back packets creates local scaling behavior over small time scales that is distinctly different from all the other versions of TCP or, for that case, UDP. A precise reasoning for why this version of TCP generates such distinctive local scaling needs however further investigations,

4.5 Putting the pieces together, not quite(!)

By experimenting with various versions of full TCP, we provide initial empirical evidence that in a reasonably heterogeneous network environment, TCP-like flow control (with or without retransmission, with fixed but small window size or with the “real” dynamic windowing mechanism) is a major reason for the emergence of complex local scaling phenomena (i.e., multifractal scaling) in measured IP traffic over fine time scales. This observation suggests that to gain a physical or networking-related understanding of the mathematical concept of multifractals, it is necessary to gain insight into the intricate interactions between the ACK packets and TCP data packets within one and the same connection in a heterogeneous network environment and across the different connections that share a common link in that same environment.

To illustrate that the ACK/TCP data packet interactions may be related to the complex local scaling phenomenon that can be mathematically described using multifractals, we return to the example of Figure 9, where we looked at the ACK-only and TCP data-only traffic associated with server 2 in the FLEXBELL configuration with 400 modem clients. Here, we slightly modify this setup by mov-

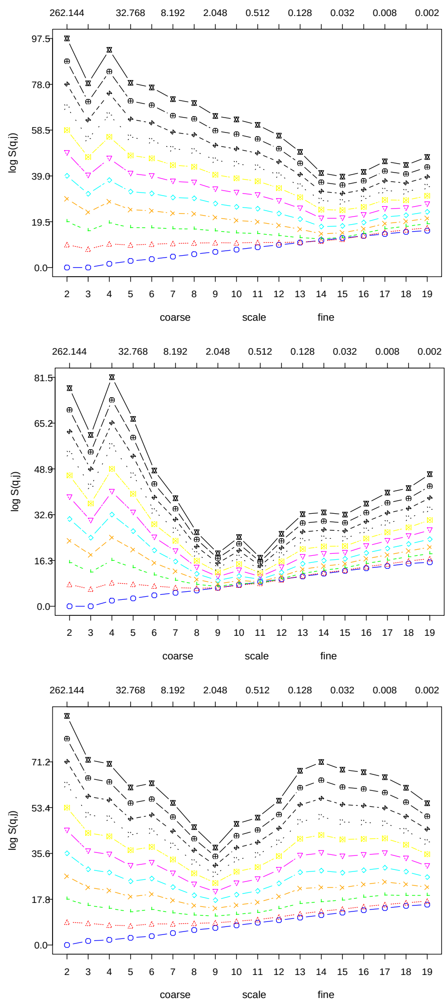

Figure 10: Impact of feedback control on local scaling: UDP with interpacket spacing of 10 msec (top, loss rate: 63.86%); TCP with window size fixed to 1 (middle, loss rate: 5.63%); TCP with flow control disabled, i.e., fixed window size of 10 (bottom, loss rate: 10.38%). (FLEXBELL, 400 MODEM NODES, PARETO 2).

ing to a version of the CROSSBELL configuration that is identical to the 400 modem FLEXBELL environment, except that we introduce cross-traffic on links labeled B and C in Figure 3 that specifically interferes with the ACK packet stream on the 0.9 and 0.6 Mbps links. The resulting local scaling plots for link B (0.9 Mbps) are shown in Figure 11, left plot for the time series of total number of packets per 1 msec, middle plot for ACK-only time series and right plot for the trace consisting of TCP data-only packets. Due to the presence of cross-traffic that interferes with the ACKs on the link connecting server 2 to the rest of the network, the local scaling plot for the ACKs looks more “interesting” (i.e., shows higher variability on the medium to small scales) than the corresponding local scaling plot for the ACK-only trace in Figure 9 which does not see any interfering cross traffic. Consequently, because ACK packets trigger TCP packets, the characteristics of the spacing of the TCP packets changes as well. This observation of a more interesting local burstiness of packet clustering behavior in the presence of cross-traffic as compared to a one-way traffic environment is known as *ACK-compression phenomenon* and agrees with findings reported, for example, in [26, 32, 17]. A more recent study [20] found ACK compression to be fairly common in measured IP

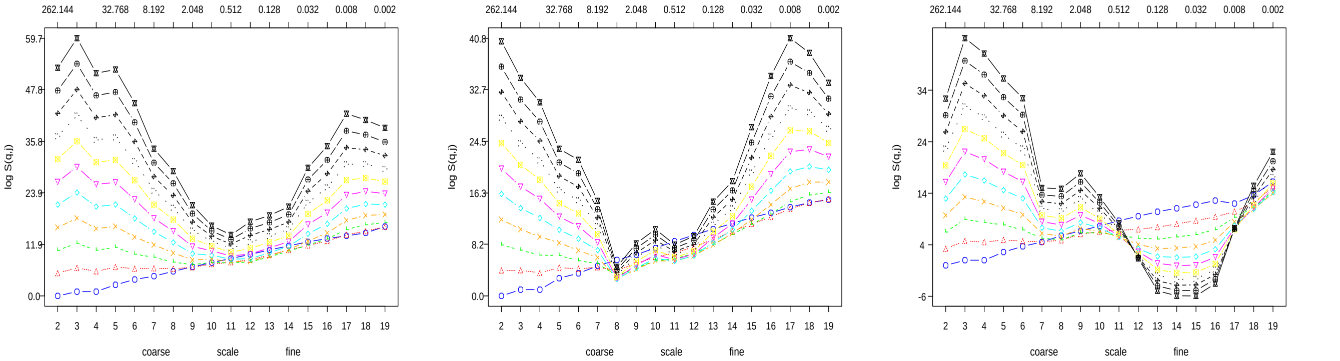

Figure 11: Impact of ACK compression on local scaling – traffic to and from server 2 according to its protocol components. From left: all packets, ACKs only, and TCP data only; (CROSSBELL, 400 MODEM NODES, PARETO 2).

traffic traces but concluded that it has no real effect on network performance. Without contradicting this latter conclusion, we contribute here to the existing literature on ACK-compression by suggesting that the ACK-compression phenomenon may be directly related to the observed highly complex scaling behavior of measured IP traffic over fine time scales and hence may be a potential candidate for explaining and understanding multifractality in terms of observed networking-related mechanisms. Rather than having a direct impact on performance, such an explanation could point toward properties of actual IP networks that have not been considered in the past and may in turn lead to an improved understanding of the dynamics of IP networks.

While our conjecture does not do much at this stage to demystify either the ACK-compression or the multifractal scaling phenomenon, it is interesting to note that the intuition behind both features (i.e., complex "clustering" and "burstiness," respectively) agrees – at least heuristically – with the visual appearance of the time series of ACK's resulting from a purely one-way traffic scenario or from a scenario where cross-traffic is present. In fact, there is in general an appreciable difference between the two time series and little disagreement about their visual effects; namely that the one-way traffic time series is "less bursty" (or, depending on one's background, "shows less ACK-compression" or "exhibits a less interesting multifractal behavior") than the two-way traffic time series. Further investigation into the one-way delay times of ACK packets from individual connections confirms that ACK compression is indeed taking place.

We conclude this section with a reminder and warning about replacing the empirically validated hierarchical and variable session structure employed in our simulations by simpler versions which in essence equate a session with an infinite file transfer. While such simplifications are often convenient for analytical studies of TCP dynamics (e.g., see [14] and references therein), they lead in general to very different behaviors of the resulting traffic, especially in a reasonably heterogeneous network environment. To illustrate, Figure 12 shows the local scaling analysis at link B for the FLEXBELL configuration under two comparable load scenarios. In one case (top plot), the clients exert high variability in terms of their session structure (in the sense discussed in Section 3.1); in the other case (bottom plot), the clients exhibit no variability; that is, sessions are infinite file transfers. Despite keeping all other components of the network environment constant, the differences in the local scaling behavior between the two resulting traffic traces are extreme and so is their global scaling behavior (not shown here). This example should serve as a reminder that mathematically convenient models do not necessarily reflect reality and should undergo more scrutiny, especially if the differences are as drastic as observed here.

5 Conclusions and outlook

By presenting a set of wavelet-based scaling techniques for analyzing and understanding network-related measurements, we have identified in this paper various user- and network-related aspects and the effects that they have on the dynamics of measured IP traffic. In other words, we have illustrated how these analysis techniques can be used for detecting and identifying "fingerprints" in measured IP traffic traces that provide relevant information about user- and network-specific behaviors. In particular, we have gained new insights into how various aspects of user- and network-related variability contribute to the observed scaling phenomena (e.g., self-similar scaling over large time scales, multifractal scaling over small time scales) in measured Internet traffic. On the one hand, we have shown how and why self-similar scaling over large time scales is almost exclusively due to user-related variability and is essentially oblivious to underlying, network-specific aspects. On the other hand, we have also explained how and why multifractal scaling

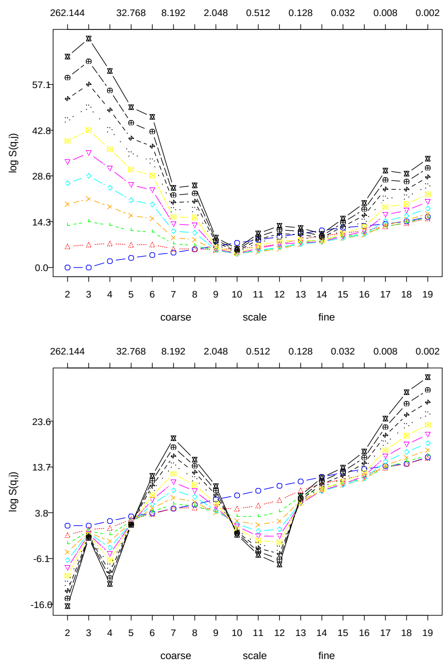

Figure 12: Impact of infinite sources on local scaling: top (FLEXBELL, 400 MODEM NODES, PARETO 2); bottom (FLEXBELL, 100 MODEM NODES, CONSTANT).

over small time scales cannot be solely explained in terms of the various aspects of network-related variability, but is impacted in a major way by the presence of TCP-like flow control algorithms which give rise to a surprisingly rich burstiness or clustering structure over small time scales of IP packets as they traverse the Internet.

Potential practical applications of this "detective" aspect of our work are numerous. To illustrate this, consider the local scaling analysis of the traces associated with the three networks shown in Figure 2 in Section 2.3. By isolating the traffic that is destined to the same part of the Internet and by applying scaling analysis to the resulting time series, one can gain insight into the performance of the paths between the measurement point and the network. For example, by comparing the left and middle plots, we can conclude that the expected performance when going to the second network is substantially worse as compared to the first network – TCP is likely to see more congestion which in turn shows up in the local scaling plot as a pronounced "folding" effect. The folding in the middle plot also indicates that somewhere along the path to and from this network, there is a bottleneck link of fairly limited capacity. A real-time implementation of this feature of the "fingerprinting" capability would have the advantage of using purely passive measurements for uncovering aspects of Internet performance that are of current interest. Such an implementation also begs for a full-blown exploitation of the local scaling analysis techniques; that is, providing the capability of localizing in time "interesting" features in a set of network-related measurements. Another potential application that is motivated by Figure 2 in Section 2.3 is using local scaling analysis techniques for detecting and identifying non-TCP-friendly connections. However, the feasibility and actual implementation of this idea remains an open problem.

On a different note, the detective nature of our investigations also has an impact on the problem related to simulation of "realistic" Internet scenarios. The challenges associated with simulating Internet-like environments are clearly spelled out in [22] but our empirical work points towards an approach that does away with traditional simulation modeling and coincides with a number of arguments put forward in [22]. In particular, we have demonstrated in this paper that by relying almost exclusively on the physical or networking-related understanding of the impacts of the various user- and network-related aspects of variability and of such basic concepts as closed-loop flow control, it appears to be possible to end up with a full-blown networking environment that is in the right "ball park" when compared to real networks. Note that this has been achieved by replacing traditional statistical inference and estimation methodologies by a qualitative understanding of which aspects impact the different scaling phenomena associated with networks, but much work is left to achieve this goal and to feel comfortable with the proposed method.

Clearly, exploring the parameter space relevant to our empirical approach is non-trivial and at times overwhelming. While the present work explores some dimensions of this space (e.g., user variability, network-related variability), others remain untouched or sufficiently obscure. For example, we have not yet systematically explored issues related to traffic synchronization (see for example [31, 10]). Although we have observed a significant amount of synchronization effects in simulations that assume infinite sources (see Section 4.5), very little of this phenomenon seems to show up when assuming our hierarchical and variable session structure for web users or when analyzing measured traces from our ISP environment. We conjecture that this lack of observed synchronization is due to the realistic variability structure of a typical web session. This conjecture is supported by our findings that in our simulation environments, we are typically able to reproduce the self-similar property of observed flow arrivals and the infinite variance or high-variability of the flows' sizes or durations. Another dimension of the parameter space that has been left unexplored but appears to play a crucial role in advancing our understanding of the spatio-temporal dynamics of IP networks is the impact or "fingerprint" of network topology-related variability. Also, while we conjecture that our findings are generic and not TCP-specific, the problem remains open as well. Finally, one of the most intriguing open issues that remains is how precisely TCP-like congestion control algorithms give rise to multifractal scaling. While we have obtained initial empirical evidence that seems to relate multifractal scaling of IP traffic with phenomena such as ACK-compression, a mathematical rigorous and intuitively appealing construction and explanation that makes sense in the networking context still eludes us. However, the experimental studies using different versions of closed-loop TCP-like or open-loop UDP-like controls shed some light on how one may want to proceed.

Acknowledgments

We would like to acknowledge David Clark for his original suggestion to look into ACK compression as a possible cause for multifractal scaling. We are also grateful to John Heidemann for valuable input concerning the presentation of our *ns-2*-based simulation results and to the anonymous Sigcomm reviewers for their constructive criticisms and suggestions for improving the presentation of the material.

References

[1] P. Abry and D. Veitch. Wavelet analysis of long-range dependent traffic. *IEEE Transactions on Information Theory*, 44:2–15, 1998.

[2] A. Arneodo. *Wavelets: Theory and Application*, Wavelet analysis of fractals: From the mathematical concept to experimental reality, pages 349–502. Oxford University Press, New York, 1996.

[3] S. Bajaj, L. Breslau, D. Estrin, K. Fall, S. Floyd, P. Haldar, M. Handley, A. Helmy, J. Heidemann, P. Huang, S. Kumar, S. McCanne, R. Rejaie, P. Sharma, S. Shenker, K. Varadhan, H. Yu, Y. Xu, and D. Zappala. Improving Simulation for Network Research Technical Report 99-702, University of Southern California, March 1999.

[4] P. Barford and M. E. Crovella. Generating representative web workloads for network and server performance evaluation. In *Proceedings of Performance'98/ACM Sigmetrics'98*, pages 151–160, 1998.

[5] M. E. Crovella and A. Bestavros. Self-similarity in world wide web traffic - evidence and possible causes. In *Proceedings of ACM Sigmetrics'96*, pages 160–169, 1996.

[6] I. Daubechies. *Ten lectures on wavelets*. SIAM, Philadelphia, 1992.

[7] A. Feldmann, R. Caceres, F. Douglis, and M. Rabinovich. Performance of web proxy caching in heterogeneous bandwidth environments. In *Proc. IEEE INFOCOM*, 1999.

[8] A. Feldmann, A. C. Gilbert, and W. Willinger. Data networks as cascades: Investigating the multifractal nature of Internet WAN traffic. In *Proc. of the ACM/SIGCOMM'98*, pages 25–38, 1998.

[9] A. Feldmann, A. C. Gilbert, W. Willinger, and T. G. Kurtz. The changing nature of network traffic: Scaling phenomena. *Computer Communication Review*, 28(2):5–29, 1998.

[10] S. Floyd and V. Jacobson. The Synchronization of Periodic Routing Messages. In *Proc. of the ACM/SIGCOMM'98*, pages 33–44, 1993.

[11] A. C. Gilbert, W. Willinger, and A. Feldmann. Scaling analysis of conservative cascades, with applications to network traffic. *IEEE Transactions on Information Theory*, 45(3):971–991, 1999.

[12] V. Jacobson. Congestion Avoidance and Control. In *Proc. of the ACM/SIGCOMM'88*, pages 314–329, 1988.

[13] R. Jain, K. K. Ramakrishnan, and D.-M. Chiu. Congestion Avoidance in Computer Networks with a Connectionless Network Layer. In Craig Partridge, editor, *Innovations in Networking*. Artech House, 1988.

[14] T.V. Lakshman and U. Madhow. The performance of TCP/IP for networks with high bandwidth-delay products and random loss. *IEEE/ACM Transactions on Networking*, 3, 1997.

[15] W. E. Leland, M. S. Taqqu, W. Willinger, and D. V. Wilson. On the self-similar nature of Ethernet traffic (extended version). *IEEE/ACM Transactions on Networking*, 2:1–15, 1994.

[16] P. Mannersalo and I. Norros. Multifractal analysis of real ATM traffic: A first look. Technical Report COST257TD, VTT Information Technology, 1997.

[17] J. Mogul. Observing TCP Dynamics in Real Networks. In *Proc. of the ACM/SIGCOMM'92*, pages 305–317, 1992.

[18] J. C. Mogul, F. Douglis, A. Feldmann, and B. Krishnamurthy. Potential benefits of delta encoding and data compression for HTTP. In *Proc. ACM/SIGCOMM'97*, pages 181–194, 1997.

[19] K. Park, G. Kim and M. E. Crovella. On the relationship between file sizes, transport protocols, and self-similar network traffic. In *Proc. IEEE International Conference on Network Protocols*, pages 171–180, 1996.

[20] V. Paxson. End-to-end Internet packet dynamics. In *Proc. ACM/SIGCOMM'97*, pages 139–152, 1997.

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

[22] V. Paxson and S. Floyd. Why we don't know how to simulate the Internet. In *Proc. of the 1997 Winter Simulation Conference*, 1997.

[23] R. Riedi, M. Crouse, V. Ribeiro, and R. Baraniuk. A Multifractal Wavelet Model with Application to TCP Network Traffic. *IEEE Transactions on Information Theory*, 45(3):992–1018, 1999.

[24] R. H. Riedi and J. Levy-Vehel. TCP traffic is multifractal: A numerical study. Preprint, 1997.

[25] S. Shenker. A Theoretical Analysis of Feedback Flow Control. In *Proc. of the ACM/SIGCOMM'90*, pages 156–165, 1990.

[26] S. Shenker, L. Zhang, and D. Clark. Some Observations on the Dynamics of a Congestion Control Algorithm. *ACM Computer Communication Review*, 20(4):30–39, October, 1990.

[27] W.R. Stevens. *TCP/IP Illustrated Volume 1*. Addison-Wesley, 1994.

[28] W3C, 1998. Web Characterization Working Group.

[29] W. Willinger, V. Paxson, and M. S. Taqqu. *A Practical Guide to Heavy Tails: Statistical Techniques for Analyzing Heavy Tailed Distributions*, Self-similarity and heavy tails: Structural modeling of network Traffic. Birkhauser Verlag, Boston, pages 27–53, 1998.

[30] W. Willinger, M. S. Taqqu, R. Sherman, and D. V. Wilson. Self-similarity through high-variability: Statistical analysis of Ethernet LAN traffic at the source level. *IEEE/ACM Transactions on Networking*, 5:71–86, 1997.

[31] L. Zhang and D. Clark. Oscillating Behavior of Network Traffic: A Case Study Simulation. *Internetworking: Research and Experience*, 1(2):101–112, 1990.

[32] L. Zhang, S. Shenker, and D. Clark. Observations on the Dynamics of a Congestion Control Algorithm: The Effects of Two-Way Traffic. In *Proc. of the ACM/SIGCOMM'91*, pages 133–147, 1991.

Appendix

A.1 Detailed description of user/session attributes

The following provides a detailed description of the user/session attributes for which one has to specify a probability distribution:

Inter–session time: Time between sessions from different users

Pages per session: Number of Web pages accessed within a session by the same user.

Inter–page time: Time between consecutive pages downloaded by the same user. We experienced with two different variances of inter–page times. In one case, the inter–page time is the time between when the page download was initiate and when the next page download is initiated. In the second case, the download of the current Web page has to complete (including the download of all of the inlined object) before the interpage time is applied. For each download of a Web page the user picks one of the available Web server at random.

Objects per page: Number of inlined objects within a Web page. All inlined objects are retrieved from the same Web server as the original Web page.

Inter–object time: Time between requests to the inlined objects.

Object size: Size of an object in KB (equals number of packets required to transfer the object).

A.2 Probability distributions for user/session attributes

Name	inter–page	objs. per page	inter–object	obj. size
PARETO 1	Pareto mean 50 shape 2	Pareto mean 4 shape 1.2	Pareto mean 0.5 shape 1.5	Pareto mean 12 shape 1.2
PARETO 2	Pareto mean 10 shape 2	Pareto mean 3 shape 1.5	Pareto mean 0.5 shape 1.5	Pareto mean 12 shape 1.2
EXP 1	Pareto mean 25 shape 2	Constant 1	—	Exp 12
EXP 2	Exp mean 10	Constant 1	—	Exp 12
CONSTANT	Constant mean 10	Constant 1	—	Constant 1000000

A.3 Description of the data sets:

Throughout this paper we use the following high-quality data sets (i.e., packet drops reported by *tcpdump* were negligible and other causes for drops have been identified to be negligible as well; high time stamp accuracy of about 10-100 μsec). The trace DIAL1 was gathered from an FDDI ring (with typical utilization levels of 5-10%) that connects about 420 modems to the rest of the Internet. Although we collect every packet seen on the FDDI ring on July 22, 1997 between 22:00 and 23:00, DIAL1 contains (bidirectional) *modem user traffic only*. This amounts to 2,752,779 packets. This is the same dataset that has been used in a previous study [8] of the multifractal scaling behavior of Internet traffic. A second trace DIAL2 was collected in the same location as DIAL1, on January 21, 1998 again between 22:00 and 23:00, and contains 2,882,859 packets.