A Cascaded On-Off Model for TCP Connection Traces

V. G. Kulkarni
Department of Operations Research
University of North Carolina
Chapel Hill, NC 27599-3180

J. S. Marron
Department of Statistics
University of North Carolina
Chapel Hill, NC 27599-3260

F. D. Smith
Department of Computer Science
University of North Carolina
Chapel Hill, NC 27599-3175

December 22, 2000

Abstract

A stochastic model for the detailed structure of individual internet session traces is proposed. The model combines multiplicative"cascading" ideas with an on-off process structure. A parameter estimation method is developed. This produces simulated traces with the same type of qualitative features as real traces. These parameters are proposed as "shape summary statistics" for characterizing individual traces. It is also seen that the cascaded on off process provides a better individual connection model than conservative cascades.

1 Introduction

A TCP connection trace is a plot of its cumulative number of byes transmitted, as a function of time. The population of such traces is very diverse. TCP traffic on a link is a superposition of many such connections. Figure 1 gives an impression of the diversity of the population of traces, by showing some examples, collected from the link connecting a university to its Internet service provider. These traces are records of the packets flowing from web servers to web browsers.

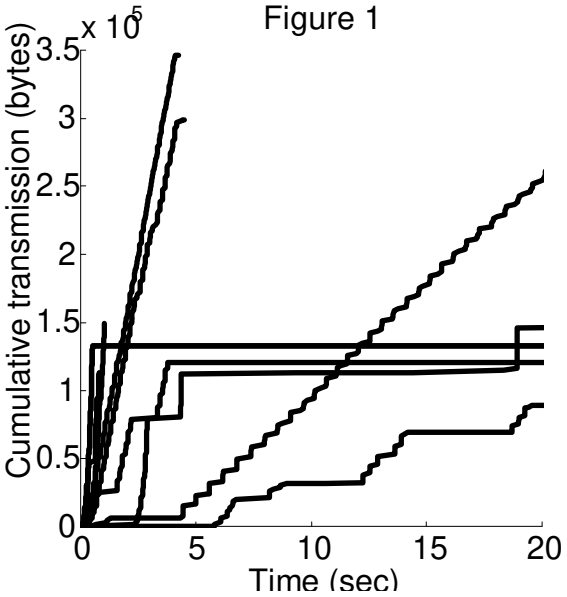

FIGURE 1: *TCP connection traces (some only partial for easy comparison), showing cumulative transmission as a function of time. This illustrates the diversity of this population.*

The goal of this paper is to develop a stochastic model for such traces. More precisely we want to model the arrival process of packets in a TCP connection as observed at some arbitrary link in the Internet. We approach this problem at the individual connection level, meaning that given a trace (e.g. one of those shown in Figure 1) we aim to construct a model which produces qualitatively similar realizations. For different approaches to this problem, see Melamed and Sengupta (1992), Garrerr and Willinger (1994), Leland, et. al. (1994), Willinger, et. al. (1995), Melamed (1997), Robert and Leboudec (1997), Willinger, Taqqu and Sherman (1997), Riedi, Crouse and Ribeiro (1999) and Sahinoglu and Tekinay (1999). The wide variety of shapes in the population of traces, e.g. Figure 1, is intended to be modeled via differing parameter values. In particular, parameter estimates for individual connections are viewed as trace summary statistics.

There are many potential approaches to construction of summary statistics. To help guide the choice among them, we suggest a list of criteria:

1. Reconstructions based on the summaries should "look qualitatively right".

2. Traces should aggregate correctly, meaning that when they are aggregated, the result should have the long range dependence and multifractal properties that have been observed in real traffic flows.

3. Allow straightforward queueing theory analysis, i.e. admit tractable calculations for predicting network demands.

4. Make physical sense, i.e. be explainable by driving forces of the type that is usual in Internet traffic.

An added benefit of getting the connection by connection structure right, is that our model could provide a useful building block for aggregation into a full scale simulation of Internet traffic, and for queueing theory analysis to predict future network demands.

Our basic model is developed in Section 2. Analysis of several properties of the model is done in Section 3. Parameter estimation is explored in Section 4. In Section 5 we show that both our model, and our parameter estimation method, give good performance, in the sense that traces simulated from the estimated model have very similar visual characteristics to the original trace. These simulations motivate some extensions and variation on our model, which are discussed in Section 6. In Section 7 we show why the Conservative Cascade model gives less effective simulated versions of TCP connection traces. ??? FDS: Section with conclusions? ???

2 Cascaded On-Off Model

The model proposed here is based on the idea that at most of the core (thus high capacity) Internet links, each packet is transmitted as a single fast burst. Thus for the packets in a given connection, the time period during which a single packet is transmitted is called the "on time", and these are quite homogeneous (in particular on times are proportional to packet sizes). On the other hand, the "off times", i.e. the periods between packets for the given connection are very diverse, and can frequently be orders of magnitude different. Some causes of this include TCP's windowing, congestion control, and loss recovery mechanisms.

Such diverse off times can be generated in a simple way, by starting with a set of independent standard On-Off processes, $X_1(t), ..., X_n(t)$, where the ith process on times are exponentially distributed with rate (i.e. 1/mean) parameter $2^{i-1}\lambda$, and the off times ore exponentially distributed with rate $2^{i-1}\mu$. More precisely, let $\{X_n(t) : t \geq 0, n = 1, 2, 3...\}$ be a sequence of independent stationary Continuous Time Markov Chains (CTMC) on the state space $\{0, 1\}$ with rate matrix

$$A = \begin{bmatrix} -2^{n-1}\mu & 2^{n-1}\mu \\ 2^{n-1}\lambda & -2^{n-1}\lambda \end{bmatrix}.$$

Our physical view of these processes is that they represent various "levels" or "scales" at which TCP traffic can be delayed, discussed in detail below. The overall flow is in the on state only when all of the components (i.e. the processes at each level) are on. A simple combination with this property is the product. In particular, a process with very heterogeneous off times is generated as:

$$Y_n(t) = \prod_{i=1}^{n} X_i(t), \quad t \geq 0, \quad n = 1, 2, 3, ...$$

Since multiplication means the zero state is achieved more and more often for larger n, a sensible normalization is to rescale to maintain a constant (over n) overall expected rate:

$$Z_n(t) = m\left(\frac{\lambda+\mu}{\mu}\right)^n Y_n(t), \quad t \geq 0, \quad n = 1, 2, 3, ...$$

Here m is the long term rate at which data are transferred. Figure 2 gives and indication of how the normalized product, $Z_n(t)$, gives the desired properties. Figures 2a, b and c show respective realizations of the processes X_1, X_2 and X_3. The normalized products Z_1, Z_2 and Z_3 are shown in Figure 2d. Note that for larger n, the constant height of Z_n is much larger, as needed to keep the expected rate constant. Figure 2e shows the cumulatives corresponding to the intensities in Figure 2d. For larger n, these are steeper during the on time, and the flat off times are longer, and also are very diverse.

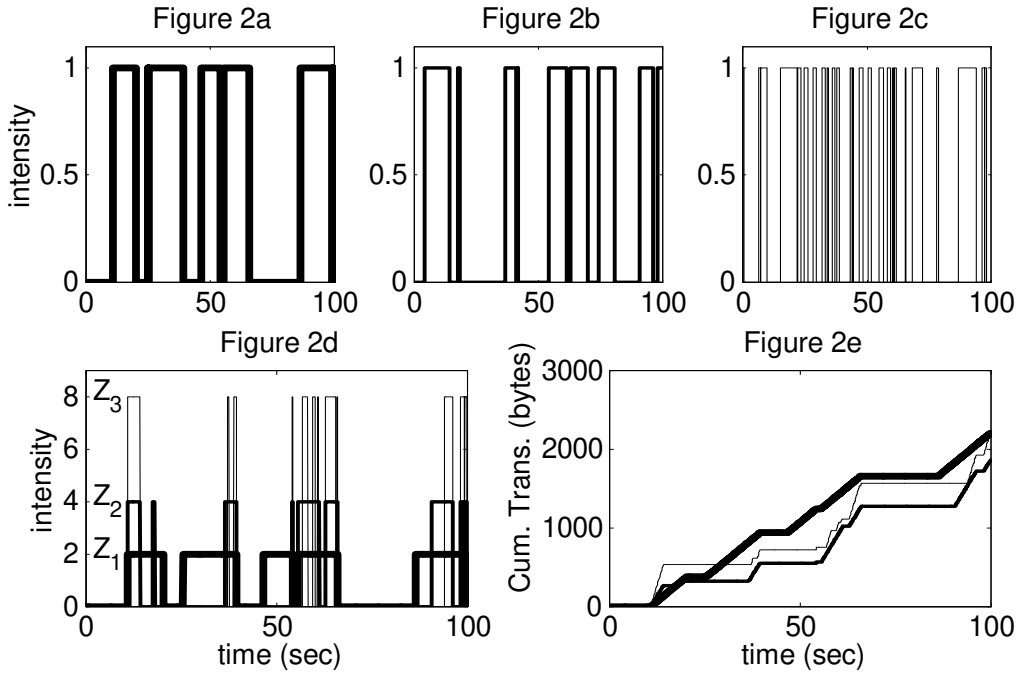

FIGURE 2: *Toy Example showing how multiplying On-Off processes produces diverse off times. Figures 2a, b and c show one realization of X_1, X_2 and X_3 respectively. Their normalized products, Z_1, Z_2 and Z_3 are shown in Figure 2d, and the corresponding cumulatives in Figure 2e.*

These levels appearing in our process have physical interpretations in terms of the delays encountered by TCP traffic. The finest scale component, modeled by the largest n and visually represented here as Figure 2c, models the the

spacings between individual packet transmissions determined by hardware and software latencies, etc. The next scale, represented here as Figure 2b, is a crude model for the TCP windowing and congestion control. The next scale, e.g. Figure 2a, could be capturing phenomena such as server induced latencies. Figure 2 doesn't allow it, but we intend to include further scales of occasional delays, including long off times generated by the actions of various Internet switches and routers, for example via packet loss from buffer overflow.

Again the key idea is that since the resulting traffic is in the on state only when all of the model components are on, it is natural to combine component effects by multiplying the processes $X_i(t)$. An issue is the choice of scaling accross levels, our factor of 2^{i-1} in the matrix A. We are suggesting this as only a crude first attempt, and believe this choice could be substantially refined.

A related model is proposed in Mannersalo, Norros and Riedi (1999), that uses products of stochastic processes to produce multifractal models of traffic. They start with n stationary, unit mean, non-negative real-valued stochastic processes and study their product. In particular they study the case where the factors are just time-scale changed versions of a single stochastic process. Their main emphasis is on the study of the convergence of the product as the number of factors goes to infinity. In particular they apply the results to a stow-state ??? VGK: what is "stow state"? ??? CTMC, which is very closely related to ours. Since they assume that the mean rate is 1 for each factor, they do not need to explictly scale the peak rate as we do in our model. Also, they do not consider the estimation aspects that we investigate in Sections 4 and 5. Our estimation procedure relates the cascaded on-off model to the physical trace in a less ambiguous fashion that makes our model more appealing than many others.

Another product based model can be found in Misra, V. and Gong, W. (1998) ??? Vidydhar: care to check this out further? ???

3 Analysis

In this section we provide mathematical analysis of the cascaded on-off rate process $Z_n = \{Z_n(t), t \geq 0\}$ defined in the previous section. First of all note that the state space of Z_n is $\{0, m(\frac{\lambda+\mu}{\mu})^n\}$. Furthermore if the processes $X_i = \{X_i(t), t \geq 0\}$ $(i = 1, 2, ..., n)$ are taken to be stationary, the Z_n process is also stationary. We construct an n-dimensional process $X^n(t) = \{(X_1(t), X_2(t), ..., X_n(t)), t \geq 0\}$ to study the properties of the Z_n process in more detail below. The X^n process itself is a Continuous Time Markov Chain (CTMC) with state space $S^n = \{0, 1\}^n$. Let Q be its generator matrix.

Example 1 *The X^2 process is a CTMC on state space $S^2 = \{(0,0), (0,1), (1,0), (1,1)\}$*

and has the following generator matrix:

$$Q = \begin{bmatrix} -3\mu & 2\mu & \mu & 0 \\ 2\lambda & -(\mu+2\lambda) & 0 & \mu \\ \lambda & 0 & -(\lambda+2\mu) & 2\mu \\ 0 & \lambda & 2\lambda & -3\lambda \end{bmatrix}.$$

3.1 The On Times

The theorem below describes the on times in the Z_n process (i.e. intervals where the Z_n process takes the value $m(\frac{\lambda+\mu}{\mu})^n$).

Theorem 2 *The successive on times in the Z_n process are iid random variables with common cdf*

$$F_{on}(t) = 1 - \exp\{-(2^n-1)\lambda t\}, \quad t \geq 0, \tag{1}$$

and mean

$$\tau_{on} = \frac{1}{(2^n-1)\lambda}. \tag{2}$$

Proof: Let e be a n-vector of all ones. From the definition it is clear that the Z_n process is "on" if and only if the X^n process is in state e. The rate at which it exits state e is

$$-Q(e,e) = \lambda + 2\lambda + 4\lambda + ... + 2^{n-1}\lambda = (2^n-1)\lambda.$$

Hence the on times are exponential random variables with rate parameter $(2^n - 1)\lambda$. This gives the cumulative distribution function and the mean given in (1) and (2). The i. i. d. nature of successive on times follows since X^n is a CTMC. □

3.2 The Off Times

Next we study the structure of the off times in the Z_n process (i. e., intervals where the Z_n process is zero). First we need some notation. Let e_i be an n-vector whose ith coordinate is zero and all other coordinates are 1 $(i = 1, 2, ..., n)$. Let $\alpha = [\alpha(a)]_{a \in S^n}$ be a row vector of size 2^n, defined as follows:

$$\alpha(a) = \begin{cases} 2^{i-1}/(2^n-1) & \text{if } a = e_i \\ 0 & \text{otherwise.} \end{cases}$$

Let $\hat{Q}$ be a modified matrix obtained from Q by deleting the row and column corresponding to the state e. Let $\hat{\alpha}$ be the row vector obtained from α by deleting the element corresponding to the state e. These constructions are illustrated by an example below:

Example 3 *The state space and the generator matrix of the X^2 process is described in Example 1. The $\hat{Q}$ matrix is as given below:*

$$\hat{Q} = \begin{bmatrix} -3\mu & 2\mu & \mu \\ 2\lambda & -(\mu+2\lambda) & 0 \\ \lambda & 0 & -(\lambda+2\mu) \end{bmatrix}.$$

The α vector is given by

$$\alpha = [0 \ \frac{1}{3} \ \frac{2}{3} \ 0]$$

and the $\hat{\alpha}$ vector is given by

$$\hat{\alpha} = [0 \ \frac{1}{3} \ \frac{2}{3}].$$

The next theorem describes the off times in the Z_n process.

Theorem 4 *The successive off times in the Z_n process are i.i.d. with common cdf*

$$F_{off}(t) = 1 - \hat{\alpha} \exp(\hat{Q}t)\mathbf{1}, \tag{3}$$

*(where **1** is column vector of all ones) and mean*

$$\tau_{off} = \frac{(\frac{\lambda+\mu}{\mu})^n - 1}{(2^n - 1)\lambda}. \tag{4}$$

Proof: We shall use the fact that an off time starts when the X^n process leaves state e and terminates when it re-enters the state e. When the X^n process leaves state e, it enters state e_i with probability $\alpha(e_i) = 2^{i-1}/(2^n - 1)$. Thus the distribution of the X^n process when an off-time starts is given by α. The off-time is thus the first passage time of the CTMC X^n into state e if its initial distribution is given by α. Hence (see Kulkarni (1995), Section 6.7) the off-time is a phase type random variable with parameters $(\hat{\alpha}, \hat{Q})$, and its cumulative distribution function is as given (Kulkarni (1995), Theorem 6.22) in (3) above. The successive off times are i. i. d. since X^n is a CTMC.

One could compute the mean as

$$\tau_{off} = -\hat{\alpha}\hat{Q}^{-1}\mathbf{1} \tag{5}$$

from Corollary 6.4 of Kulkarni(1995). However, a more compact expression can be obtained by using the fact that Z_n is an alternating renewal process. (See Kulkarni (1995) Chapter 8 for definitions). From the theory of renewal processes (see Kulkarni (1995) Theorem 8.23)we get

$$\lim_{t\to\infty} P\{Z_n(t) = m(\frac{\lambda+\mu}{\mu})^n\} = \frac{E(\text{On Time})}{E(\text{Off Time}) + E(\text{On Time})} = \frac{\tau_{on}}{\tau_{off} + \tau_{on}} \tag{6}$$

where τ_{on} is given in (2), and τ_{off} is to be determined. Using the results about the limiting distribution of a two-state CTMC (Kulkarni (1995), Example 6.29), we get

$$\begin{aligned}\lim_{t\to\infty} P\{Z_n(t) = m(\frac{\lambda+\mu}{\mu})^n\} &= \lim_{t\to\infty}\prod_{i=1}^{n} P\{X_i(t)=1\} \\ &= \prod_{i=1}^{n}\frac{2^{i-1}\mu}{2^{i-1}\lambda+2^{i-1}\mu} \\ &= (\frac{\mu}{\lambda+\mu})^n. \end{aligned} \tag{7}$$

Using (2) and (7) in (6), we can solve for τ_{off}, which yields (4).□

A simple extension to Cor. 6.24 of Kulkarni (1995) gives

$$E((\text{Off Time})^2) = \hat{\alpha}(\hat{Q}^{-1})^2\mathbf{1}. \tag{8}$$

This can be used to compute the variance of the off times. This calculation has to be done numerically, and we do not yet have a closed form expression for the variance of the off times ??? VGK: want to say something here about the faster formula? ???

3.3 The Autocovariance Function

Assume that X_i $(i = 1, 2, ..., n)$ are stationary, so that Z^n is also stationary. In this section we compute the autocovariance function of the Z_n process. The main result is given in the following theorem.

Theorem 5

$$Cov(Z_n(t), Z_n(0)) = m^2\left[\prod_{i=1}^{n}\left(1+\rho e^{-2^{i-1}(\lambda+\mu)t}\right)-1\right]. \tag{9}$$

Proof: We know that (See Kulkarni (1995), equation 6.38)

$$P(X_i(t)=1|X_i(0)=1) = \frac{\mu}{\lambda+\mu}[1+\frac{\lambda}{\mu}e^{-2^{i-1}(\lambda+\mu)t}], \quad t\geq 0.$$

Now,

$$\begin{aligned} E(Z_n(t)Z_n(0)) &= m^2(\frac{\lambda+\mu}{\mu})^{2n}E(\prod_{i=1}^{n}X_i(t)\prod_{i=1}^{n}X_i(0)) \\ &= m^2(\frac{\lambda+\mu}{\mu})^{2n}\prod_{i=1}^{n}E(X_i(t)X_i(0)) \\ &= m^2(\frac{\lambda+\mu}{\mu})^{2n}\prod_{i=1}^{n}P(X_i(t)=1|X_i(0)=1)\frac{\mu}{\lambda+\mu} \\ &= m^2\prod_{i=1}^{n}[1+\frac{\lambda}{\mu}e^{-2^{i-1}(\lambda+\mu)t}]. \end{aligned} \tag{10}$$

Using stationarity we get

$$\begin{aligned} E(Z_n(t))E(Z_n(0)) &= m^2(\frac{\lambda+\mu}{\mu})^{2n}E(\prod_{i=1}^{n}X_i(t))E(\prod_{i=1}^{n}X_i(0)) \\ &= m^2(\frac{\lambda+\mu}{\mu})^{2n}(\frac{\mu}{\lambda+\mu})^{2n} \\ &= m^2. \end{aligned} \tag{11}$$

Using Eqs. 10 and 11, we get Eq. 9.□

Thus the autocovariance dies off exponentially, dominated by the exponential decay rate of $\lambda + \mu$. Thus, clearly the process will not display long range dependence.

3.4 Queueing Analysis

Suppose the input to a queue is modeled by a cascaded on-off process Z_n with mean rate m and peak rate $r_n = m\left(\frac{\lambda+\mu}{\mu}\right)^n$. The buffer size is infinite, and the output from the queue is at a constant rate c whenever the buffer is non-empty. Let $W(t)$ be the buffer content at time t. The dynamics of the buffer content process $\{W(t), t \geq 0\}$ are given by

$$\frac{dW(t)}{dt} = \begin{cases} Z_n(t) - c & \text{if } W(t) > 0, \\ (Z_n(t) - c)^+ & \text{if } W(t) = 0. \end{cases}$$

If $m \geq c$ the queue is unstable and the buffer content is infinite in the limit as $t \to \infty$. If $r_n \leq c$, the buffer content is always zero. Hence we consider the case $m < c < r_n$, where the buffer content process has a bonafide limiting distribution, see Kulkarni(1997). Since the input is on-off with exponential on times and general off times, the tail behavior of the buffer content process has been well studied in the literaure. For example, Gautam et al. (1999) study a fluid queue driven by a semi-Markov process. We apply their results to our case (a two-state semi-Markov process) and state the results below.

Let $\phi(s)$ be the LST (Laplace Stieltjes Transform) of the off time distribution. From Equation 3 we get

$$\phi(s) = \int_0^\infty e^{-st}dF_{off}(t) = \hat{\alpha}\hat{Q}(sI - \hat{Q})^{-1}\mathbf{1}. \tag{12}$$

Define $\lambda_n = (2^n - 1)\lambda$ and let η be the smallest solution to

$$\phi(\eta c) = 1 - \frac{\eta(r_n - c)}{\lambda_n}.$$

Then, from the results of Gautam et al. we get

Theorem 6 *Suppose $m < c < r_n$. Then*

$$\lim_{t\to\infty} P(W(t) > x) \leq e^{-\eta x}.$$

. This provides the exponential decay bound on the tail of the steady state buffer content. Thus the traffic losses at a finite buffer of size B can be approximated by $e^{-\eta B}$.

A related approach to queueing analysis was taken by Carlsson and Fiedler (2000), who base their work on the Mannersalo, et. al. (1999) model. Although they don't explicitly mention it, they assume that the factors are independent 2-state CTMCs. Then they do the explicit quqeing analysis by using the $2^n \times 2^n$ matrix of the n-dimensional process to obtain the exact stationary probability density function of the buffer content process using well established results on Markov modulated fluid queues. We only report the exponential bound that is sufficient in many instances.

3.5 Limiting Behavior: Preliminary Results

A question of interest is: what happens to the Z_n process as $n \to \infty$. We have not studied this question fully as yet, but have a preliminary understanding summarized in the theorem below.

Theorem 7

$$\lim_{n\to\infty} \tau_{on} = 0, \tag{13}$$

$$\lim_{n\to\infty} \tau_{off} = \begin{cases} 0 & \text{if } \lambda < \mu, \\ \infty & \text{if } \lambda > \mu, \\ 1/\lambda & \text{if } \lambda = \mu. \end{cases} \tag{14}$$

$$\lim_{n\to\infty} \frac{\tau_{off}}{\tau_{on}} = \infty. \tag{15}$$

Proof: Follows from the explicit expressions for τ_{on} and τ_{off}.□

Thus the behavior of the mean off time depends on the relative magnitudes of λ and μ in a critical fashion. We expect that the limiting behavior of the Z_n process will also depend on this in a similar fashion.

4 Parameter Estimation

In this section, we propose estimates of the model parameters λ, μ and n, based on a single TCP connection trace. Successful estimation should give a fitted stochastic process whose realizations share the qualitative features of the trace.

More precisely, the estimation is based on the following data:

1. The peak rate r_{peak} is the true transmission rate at the location where the measurements are made, in bits per second. In all examples in this paper, measurement was made on an OC-3 link, so $r_{peak} = 1.55 \times 10^8 (bits/\text{sec})/(8 bits/byte)$.

2. N = total number of packets in the trace

3. T_i = time stamp of the ith packet, in secs. $(i = 1, 2, ..., N)$

4. S_i = size of the ith packet, in bytes. $(i = 1, 2, ..., N)$

Figure 3 shows how these pieces of data are used to construct sets of sample on times and off times to use in the estimation. The raw data is a set of packet time stamps, and cumulative sizes (i.e. the total size number of bytes transmitted up to the given time), which are shown as symbols in Figure 3. The simplest version of the cumulative is shown as the medium thickness broken line, simple linear interpolation. A refined version of the cumulative, shown as the medium solid line, takes the peak rate (generally much faster than the rate represented by the slope of the linear interpolant) into account, alternating constant off times, with on times at the rate r_{peak}. Note that r_{peak}, together with the time stamps and packet sizes, determines the respective on (shown as the thick solid line) and off (shown as the thick dashed line) times.

Figure 3

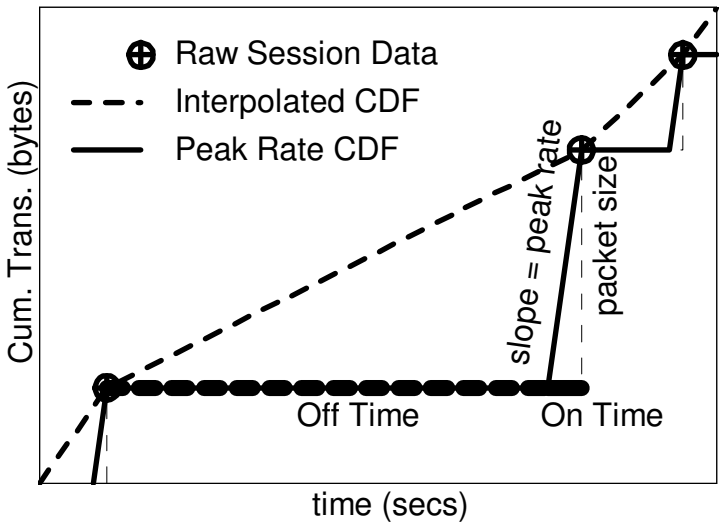

FIGURE 3: *Zoomed in plot of a TCP connection trace, showing raw data, and how the peak rate is used to determines on and off times.*

Here are the proposed steps for parameter estimation based on simple Method of Moment ideas. First for a fixed value of the level parameter n:

1. Get the total size right: i.e. estimate the overall mean rate r_{mean} of transmission for the given connection by

$$\hat{r}_{mean} = \frac{\sum_i S_i}{T_N} = \frac{total\ size}{total\ time}.$$

2. Make the jumps right: i.e. estimate the mean on time τ_{on} by

$$\hat{\tau}_{on} = \frac{\sum_i S_i}{N r_{peak}} = \frac{\hat{r}_{mean}}{r_{peak}} \cdot \frac{T_N}{N},$$

which is the average of the on times shown as the thick solid line in Figure 3.

3. Estimate the mean off time τ_{off} by

$$\hat{\tau}_{off} = \frac{T_N}{N} - \hat{\tau}_{on},$$

which is the average of the off times shown as the thick dashed line in Figure 3.

4. Solve the mean equations (2) and (4) to get estimates of λ and μ:

$$\hat{\lambda}_n = \frac{1}{\hat{\tau}_{on}(2^n - 1)},$$

$$\hat{\mu}_n = \frac{\hat{\lambda}_n}{\left[\frac{\hat{\tau}_{off}}{\hat{\tau}_{on}} + 1\right]^{1/n} - 1}.$$

Using this method, for several choices of the level n, on some sample traces gave good results (in terms of qualitative appearance), however different values of n were needed. Hence it seems important to also estimate the parameter n. Again we appeal to a Method of Moments approach, focussing now on 2nd moments, i.e. variances (since the first moments have already been used). Since the on time distribution is exponential, the variance is determined by the first moment τ_{on}, and contains no additional parameter estimation information. But some investigation of the off time distribution showed that n is directly related to the variance. There does not appear to be a closed form for the off time variance, but a numerical version, based on (8) is straighforward to compute. This took a reasonable to large amount of time (up to an hour on a 400 MHz PC) for the values that we encountered, $n = 1, 2, ..., 12$. It seems likely that this computation can be speeded up very considerably, but we have not attempted this yet.

To estimate n, we propose considering a grid of n values. For each n, estimate $\hat{\lambda}_n$ and $\hat{\mu}_n$ as above. Then calculate the theoretical off time variances $\sigma^2_{off}\left(\hat{\lambda}_n, \hat{\mu}_n, n\right)$, $n = 1, 2, ...$ Choose n which makes this closest to the sample off time variance $\widehat{\sigma}^2_{off}$. The effectiveness of this method is demonstrated in the next section.

5 Simulation Results

This section shows the effectiveness of the estimation method developed in Section 4. For a given real trace, the parameters are estimated, and the effectiveness is assessed in terms of visual impression.

Figure 4 shows the full estimation process. Figure 4a is the original raw data trace, showing both TCP congestion control and windowing effects, and also two long delays apparently caused by packet loss and timeout before retransmission of the lost data. The x-axis in Figure 4b shows the range of n values considered, $n = 1, ..., 10$, and the corresponding estimated values $\hat{\lambda}_n$, $\hat{\mu}_n$. Figure 4c shows the corresponding model variance $\sigma^2_{off}\left(\hat{\lambda}_n, \hat{\mu}_n, n\right)$, as the thick curve, and how it compares with the off time sample variance, $\widehat{\sigma}^2_{off}$. These are closest at $n = 9$, as indicated by the vertical dashed line (also shown in Figure 4b). Figure 4d shows 5 realizations from the estimated model. Note that these do a good job of reproducing the visual impression of the raw data trace, with about the right number and sizes of flat spots.

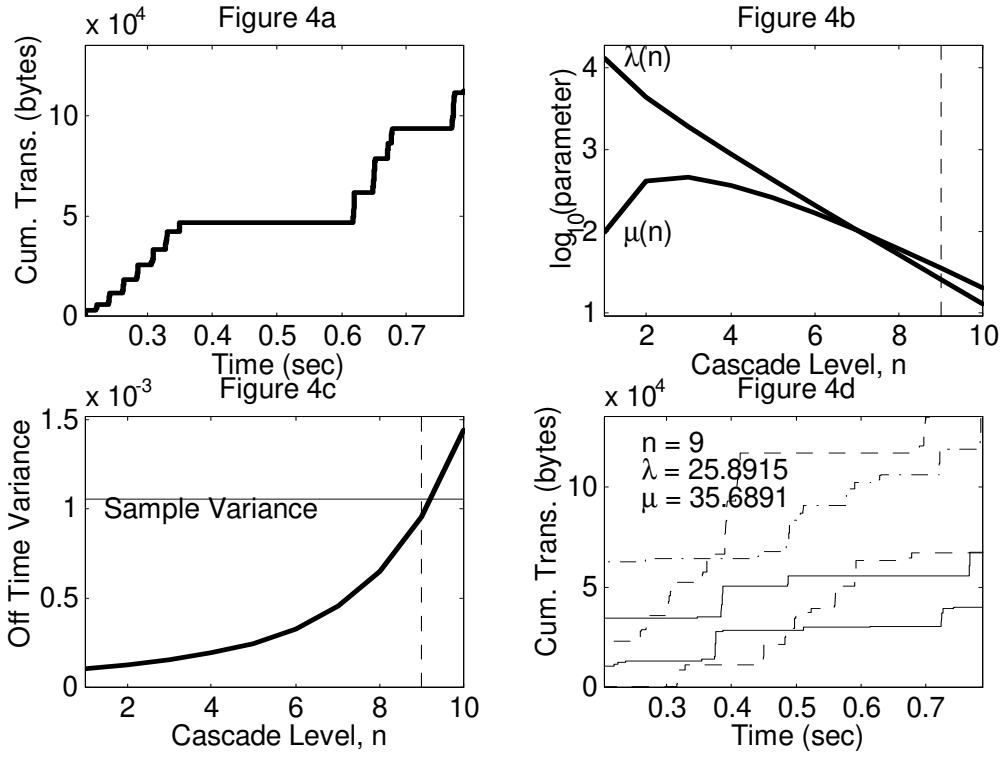

FIGURE 4: *Raw data trace is in Figure 4a, with the estimation process illustrated in Figures 4b and 4c. Figure 4d shows 5 simulated traces from the model with estimated parameters, showing similar qualitative structure.*

Another example is shown in Figure 5. This time the estimation parts are omitted to save space (these parts look similar to Figures 4b and 4c), and only the raw data is shown in Figure 5a, and the simulated realizations from the estimated model are shown in Figure 5b. Once again the visual impression of the simulated traces is similar to the real one.

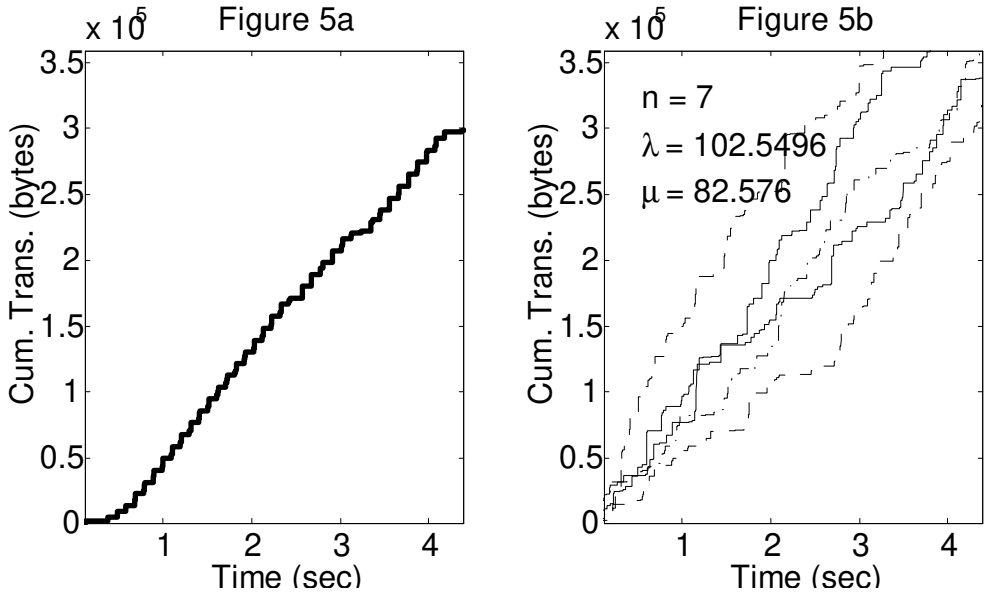

FIGURE 5: *Another trace, Figure 5a, together with 5 traces simulated from the estimated model. This again shows good visual fit of the model.*

We considered several other examples, where the visual impression was as good as the above. The worst cases that we observed follow.

Figure 6a shows a real data trace with TCP "slow start" structure. The simulated traces from the estimated model, shown in Figure 6b, have about the right numbers and types of flat pieces, but they appear in random locations, not carefully ordered at the beginning, as in the real data trace. Thus slow start is a trace characteristic that is not well captured in our model.

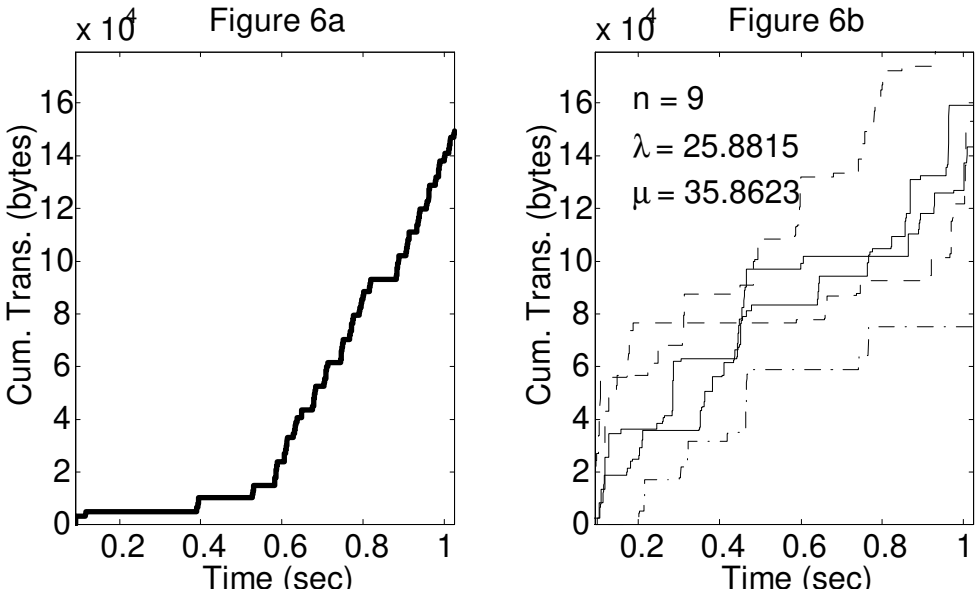

FIGURE 6: *Another estimation - simulation example showing that while our model gives the right number and size of flat spots, it does not effectively model TCP slow start.*

The worst fit of our model to a real data trace is shown in Figure 7. This trace has bursts of data, of similar size, arriving at 15 seconds interval, of the type that is common in automatically refreshed web pages ??? FDS: this OK now? ???, resulting in a trace with large chunky steps. These regular steps are not well captured by our model.

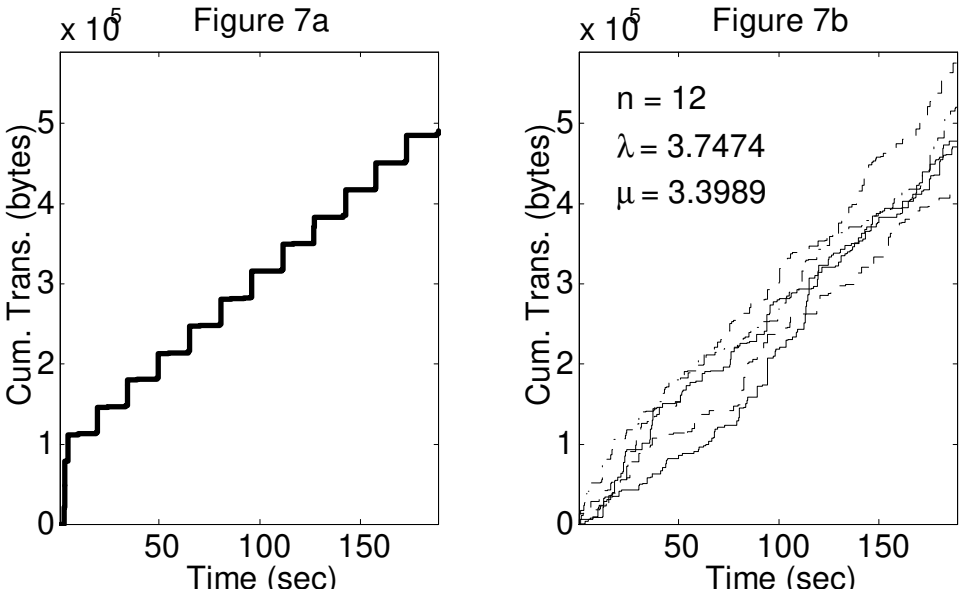

FIGURE 7: *The worst case estimation - simulation example that we observed, where the TCP window is very large, giving a structure that is not effectively represented by our models.*

"Visual impression" is a simple measure of goodness of fit of a model. More sophisticated measures are based on statistical summaries. An important statistical summary is the time varying correlation structure. An important, and perhaps questionable, assumption of our Cascaded On Off model is that the off times are independent. This is investigated in Figure 8, which shows the autocorrelation function of the time series of off times from the trace shown in Figure 4a, as the thick solid line.

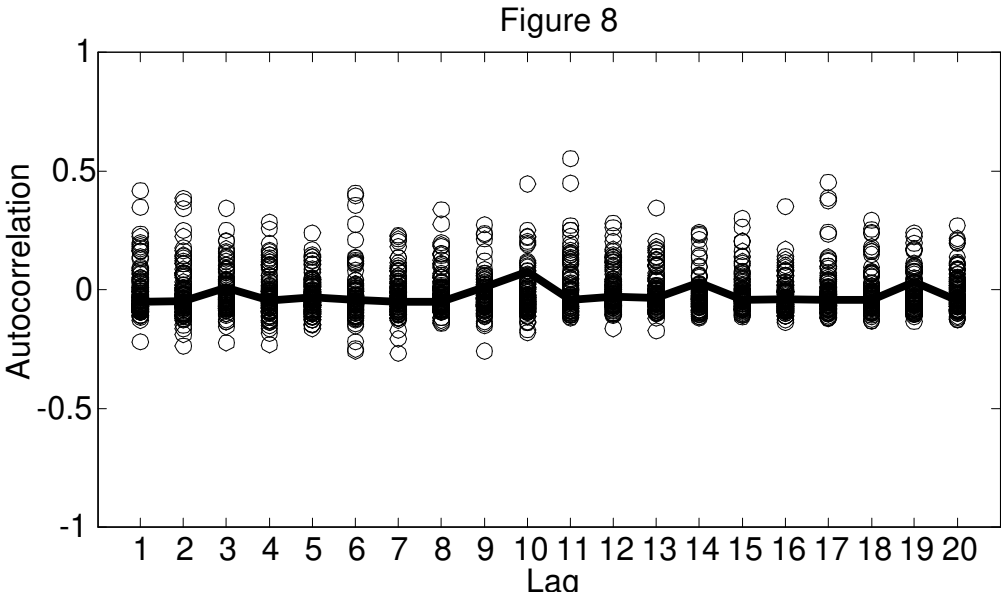

FIGURE 9: *Sample autocorrelation function for the off time series, from the trace in Figure 4a, shown as the solid line. The circles are 100 simulated autocorrolations from the fit model, to assess the variability.*

The autocorrelation function is not identically zero, as expected, because of sampling variability. The overlaid circles assess whether these departures from zero are statistically significant. They are the sample autocorrelations based on 100 simulated draws from the fit distribution, illustrated in Figure 4d. These show that the solid curve is well within the expected variability, and thus the our independent model holds up well.

We obtained very similar results for most of the other traces we have studied. Exceptions were all caused by periodicities. For example the repeated bursts of data in the trace of Figure 7a gave substantial autocorrelation. A different type of non-zero autocorrelation was caused by TCP windowing effects.

6 Possible Extensions and Future Work

Many variations of the construction of our model given in Section 2 are possible. The above was chosen as the simplest with good visual properties, and is intended only as a first crude approximation.

One point where our model clearly differs from the traces we have studied, is in the on time distributions. We have modelled this as exponential, but in the actual data there is a strong tendency for packet sizes to be nearly constant. This is because common TCP segment sizes are around 1500 bytes. For investigations at the scale of the analysis in Section 5, this issue is not important

because it does not affect the visual impression, but it could be for other considerations. A simple modification of our model, which could handle this, is to allow the on times for the component $X_n(t)$ to be constant.

Our model also does a poor job of describing TCP slow start, as shown in Figure 6. A simple means of handling this is to fit an exponential curve to the trace (or to a suitably defined "beginning part"). Then a deterministic 0-1 factor, with appropriate exponentially decreasing off times, could be added to the model. VGK: work "semi Markov" into this?

Also, as shown in Figure 7, our model does not effectively handle peiodicities. This problem is more challenging, because it requires estimation of the size and frequency of these varying steps. But if this can be done effectively, then an appropriate factor could again be added to our model.

As noted in Section 2 our choice of rescaling the rates by a factor of 2 is completely arbitrary. More generally a scale factor of r_n could be used. It could be worth using a simple parametric form, and doing parameter estimation.

Additional important future work is in the direction of point 2, in Section 1: correct aggregation. We plan to investigate this by comparing statistical properties (including long range dependence and fractality measures) of aggregated simulated traces, with that of actual aggregate traffic.

7 Why Not Conservative Cascades?

In this section, conservative cascades are investigated as an alternative model for generating simulated traces. See for example Section 1.3.2 of Riedi and Willinger (1998) for a careful description of the construction. Figure 9 illustrates the main idea.

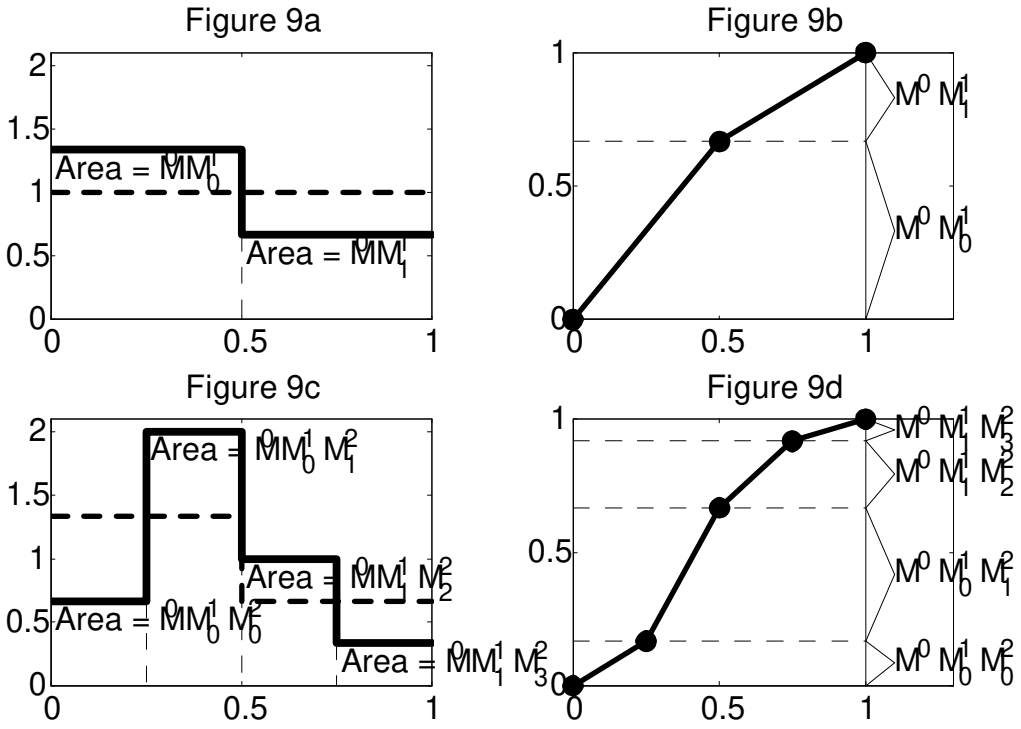

FIGURE 9: *Toy example illustrating the conservative cascade construction. Figure 9a shows the rate function r_1 and Figure 9b shows the corresponding cumulative for $n = 1$. Figure 9c shows the rate function r_2 and Figure 9d shows the corresponding cumulative for $n = 2$.*

The conservative cascade construction is iterative, with two such steps shown in Figure 9. The curves on the left side are"rate functions" $R_n(t)$ defined on $[0, 1]$, with the corresponding "cumulatives" shown on the right side. A simple view is that at each level the existing subintervals are split in half, and the centerpoint of the cumulative is moved vertically by a random amount (as shown in Figure 9d). At the level n this results in intervals with endpoints at the dyadic points $\frac{i}{2^n}$, $i = 1, 2, 3, ..., 2^n - 1$, and with piecewise jumps in the cumulatives of the form: $M^0 M_{i_1}^1 M_{i_2}^2 \cdots M_{i_n}^n$, where the "generators of the cascade" are random variables

$$\left\{ M_i^j : i = 0, 2, ..., 2^j - 2, \ j = 1, 2, ..., n \right\}, \tag{16}$$

that are assumed to be independent. As shown in Riedi and Willinger (1999), the resulting stochastic process has some multifractal properties.

While there is a lot of flexibility in terms of generator distributions, we were not successful in finding ones that gave simulated traces that looked visually

similar to those in Figure 1. To understand why, we "backsolved for the generators", and then did a statistical analysis. In particular, for $n \approx \log_2(\#\text{ packets})$, we interpolate the real data trace to the dyadic grid $\left\{\frac{i}{2^n} : i = 0, 1, ..., 2^n\right\}$. Now the mass assigned to the subinterval indexed by k (even) is $M^0 M^1_{i_1} M^2_{i_2} \cdots M^n_k$ and the mass for the next subinterval indexed by $k+1$ is $M^0 M^1_{i_1} M^2_{i_2} \cdots M^n_{k+1}$. The ratio of these masses is $R^n_k = \frac{M^n_k}{M^n_{k+1}}$. Thus solving the mass conservation equation $M^n_k + M^n_{k+1} = 1$ gives $M^n_k = \frac{R^n_k}{R^n_k+1}$ and $M^n_{k+1} = \frac{1}{R^n_k+1}$. This procedure is quickly and simply iterated back through the entire cascade, to give empirical values of the full set of generators (16).

Figure 9 shows some statistical analysis of the M^n_k, for the TCP connection trace in Figure 4a. We repeated the analysis for a number of other different looking traces, but the main lessons were similar. These were computed using $n = 10$, i.e. interpolation was done to a grid of $2^{10} = 1024$.

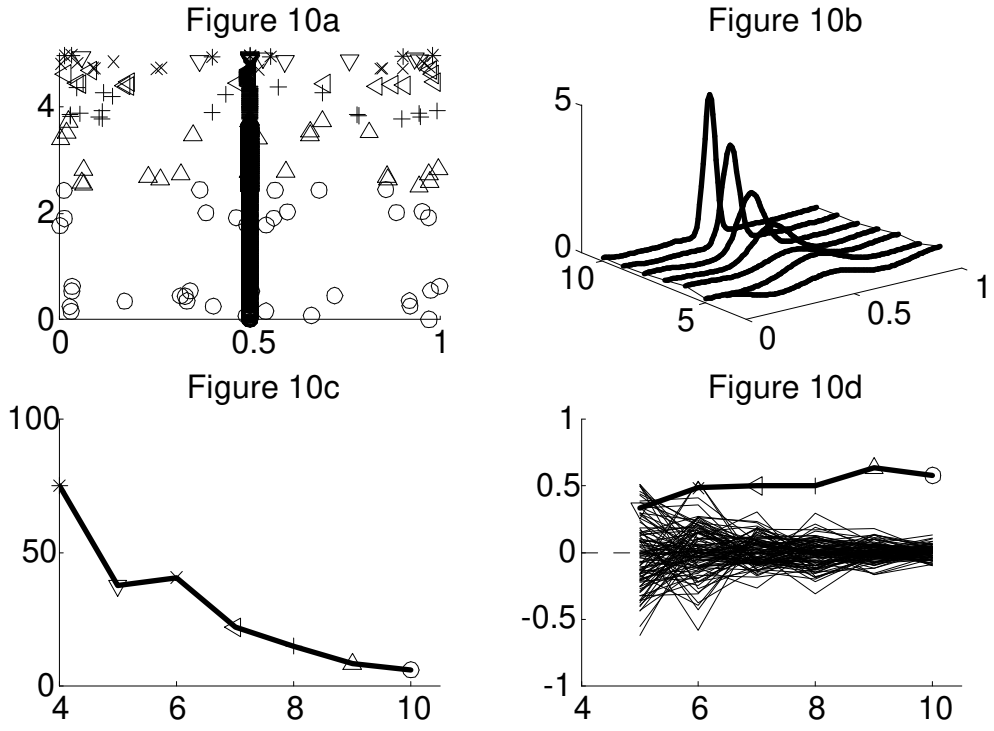

FIGURE 10: *Statistical analyses of decomposed conservative cascade generators for trace of Figure 4a. Figure 10a is an order plot of the raw generators. Figure 10b shows level wise kernel smoothed histograms. Figure 10c show the levelwise percent inside the interval* $[0.45, 0.55]$. *Figure 10d shows the correlation between level* j *and* $j-1$.

Figure 10a is a graphical display of the M_i^j. Only even i are shown because the even and odd pairs sum to 1, so including the odd pairs does not add useful information (they would just be mirror images of the shown points). The numerical value appears on the horizontal axis, and a lexicographical ordering of the M_i^j appears on the vertical axis. The level $j = 4, ..., 10$ (smaller j levels could be computed, but less than 8 data points does not admit useful analysis of the type done here) is indicated with different symbol types. The highest level values, for $j = 10$, are shown at the bottom as circles, and they are ordered depending on the index i. This fills half the plot, because essentially half of the full set appear at this level. The next level, $j = 9$, are shown as triangles, and occupy the next quarter of the plot, and so on. The symbol types are connected with the level j in Figure 10c. Figure 10a shows that a very large number of the M_i^j take on the value 0.5, so this should be built into any model that attempts to reconstruct traces of the type shown in Figure 1.

A more precise view of the level-wise distributions is shown in Figure 10b, where each curve is a kernel density estimate of the generators at the given level j. This shows that it would be inappropriate to consider a conservative cascade model with a generator distribution that is constant across the level j. It also suggests that some simple level-varying models could be appropriate.

A large part of the level varying structure in Figure 10b is explainable in terms of the number of M_i^j that take on values near 0.5, which is shown in Figure 10c. For each level $j = 4, ..., 10$, shown on the horizontal axis, the percent of the M_i^j, for even $i = 0, 2, ..., 2^j - 2$, in the range $[0.45, 0.55]$ are shown. This is quite large at the high levels, and falls away in a fashion that could be simply modelled.

These plots suggest ways in which simple, sensible, level-wise conservative cascade models could be constructed. We tried this, but the simulation results did not look like the driving traces. In particular, the visual impression was much worse than the results shown in Section 5.

The main reason for this is shown in Figure 10d. In this graph, again the level j is on the horizontal axis (again using the same level symbol types), but this time the across row empirical correlation, $\rho\left(M_i^j, M_{i/2}^{j-1}\right)$, based on the sample of even $i = 0, 2, ..., 2^j - 2$, between each generator and its "parent" in the next level, is shown. Note that this is systematically quite large, indicating a strong positive correlation across levels. The envelope of thin lines provides a check that this correlation is systematic, and not due to sampling variability. It shows 100 realizations of the correponding correlations, from the generators of a simulated cascade.

This correlation is a clear violation of the usual independence assumptions made for conservative cascades. It seems to be mostly caused by the long (and also the shorter) flat spots in the trace, seen in Figure 4a. The dyadic interpolation slices the flat segments in rather arbitrary ways, leading to this dependence. One could attempt to build across row correlation into the conservative cascade model, but we view our cascaded on off model as a more natural way to model the type of structure that is visible in the data.

Other modifications of the conservative cascade idea are also worth consideration. For example we tried applying the cascade construction to the inverse cumulative function. This corresponds to dyadically interpolating the vertical size axis. Then instead of studying "vertical random shifts at dyadic time points", one studies "horizontal random shifts at dyadic size points". The same analysis as above was applied, and gave some substantial improvement in visual quality of simulated traces, however it was still unsatisfactory, especially compared to the cascaded on off model. Again the main problem seems to be that the conservative cascade model has a rigidity, that makes it poor at modelling the type of structure present in the data.

Another variation, studied in detail by Dubins, L. E. and Freedman, D. A. (1967), is to allow both vertical an horizontal perturbations of the cumulative. This is intuitively attractive, and may also solve some of the problems with conventional conservative cascades. We have not followed this direction, because it looks less tractable for subsequent queueing theory type analysis than our cascaded on off model.

References

[1] Dubins, L. E. and Freedman, D. A. (1967) Random distribution functions, *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Vol. II: Contributions to Probability Theory*, University of California Press, Berkeley, CA, 183-214.

[2] Garrerr, M. W. and Willinger, W. (1994). Analysis, Modeling and Generation of Self-Similar Video Traffic, *Proc. of the ACM Sigcom '94*, London, UK, 269-280.

[3] Gautam, N., Kulkarni, V. G., Palmowski, Z. and Rolski, T. (1999). Bounds for fluid models driven by semi-Markov inputs, *PEIS*, **13**, No. 4, 429-475.

[4] Kulkarni, V. G. (1995) *Modeling and Analysis of Stochastic Systems*, CRC Press, London.

[5] Kulkarni. V. G. (1997). Fluid models for single buffer systems, *Frontiers in Queueing: Models and Applications in Science and Engineering*, 321-338, Ed. J. H. Dshalalow, CRC Press.

[6] Leland, W. E., Taqqu, M. S., Willinger, W. and Wilson, D. V. (1994). On the Self-Similar Nature of Ethernet Traffic (Extended Version), *IEEE/ACM Trans. on Networking*, 2, 1-15.

[7] Mannersalo, P., Norros, I. and Riedi, R. (1999) Multifractal products of stochastic processes: a preview, unpublished manuscript.

[8] Melamed, B. and Sengupta, B. (1992). TES Modeling of Video Traffic, *IEICE Transactions on Communications.*, E75-B, 1292-1300.

[9] Melamed, B. (1997) The Empirical TES Methodology: Modeling Empirical Time Series, *Journal of Applied Mathematics and Stochastic Analysis*, 10, 333–353.

[10] Misra, V. and Gong, W. (1998) A Hierarchical Model for Teletraffic, *Proceedings of 37th IEEE Conference of Decision and Control*, Tampa, Fl.

[11] Riedi, R. H. and Willinger W. (1998) Toward an improved understanding of network traffic dynamics, in *Self Similar Network Traffic and Performance Evaluation*, Park, K. and Willinger, W. Eds., Wiley, New York.

[12] Riedi, R. H., Crouse, M. S. and Ribeiro, V. J. (1999). A multifractal wavelet model with application to network traffic, *IEEE Transactions on Information Theory*, 45, 992-1018.

[13] Robert, S. and Leboudec, J. Y. (1997). New models of pseudo self-similar traffic, *Performance Evaluation*, 30, 57-68.

[14] Sahinoglu Z and S. Tekinay, S. (1999). On multimedia networks: Self-similar traffic and network performance, IEEE Communications Magazine, 37, 48-52.